



**FACULTAD DE CIENCIAS HUMANAS Y SOCIALES  
DEPARTAMENTO DE TRADUCCIÓN Y COMUNICACIÓN**

## **Tecnologías libres para la traducción y su evaluación**

Presentado por:

Silvia Andrea Flórez Giraldo

Dirigido por:

Dra. Amparo Alcina Caudet

Universitat Jaume I

Castellón de la Plana, diciembre de 2012



## **AGRADECIMIENTOS**

Quiero agradecer muy especialmente a la Dra. Amparo Alcina, directora de esta tesis, en primer lugar por haberme acogido en el máster Tecnoloc y el grupo de investigación TecnoLeTTra y por haberme animado luego a continuar con mi investigación como proyecto de doctorado. Sus sugerencias y comentarios fueron fundamentales para el desarrollo de esta tesis.

Agradezco también al Dr. Gabriel Quiroz, quien como profesor durante mi último año en la Licenciatura en Traducción en la Universidad de Antioquia (Medellín, Colombia) despertó mi interés por la informática aplicada a la traducción. De igual manera, agradezco a mis estudiantes de Traducción Asistida por Computador en la misma universidad por interesarse en el software libre y por motivarme a buscar herramientas alternativas que pudiéramos utilizar en clase sin tener que depender de versiones de demostración ni recurrir a la piratería. A mi colega Pedro, que comparte conmigo el interés por la informática aplicada a la traducción y por el software libre, le agradezco la oportunidad de llevar la teoría a la práctica profesional durante todos estos años.

Quisiera agradecer a Esperanza, Anna, Verónica y Ewelina, compañeras de aventuras en la UJI, por haber sido mi grupo de apoyo y estar siempre ahí para escucharme en los momentos más difíciles. Mis más sinceros agradecimientos también a María por ser esa voz de aliento y cordura que necesitaba escuchar para seguir adelante y llegar a feliz término con este proyecto.

Gracias a mi familia por su apoyo siempre incondicional y especialmente a mis padres por preocuparse en todo momento por mi bienestar, así como por inculcarme la curiosidad y el amor por los libros desde pequeña. Por último, gracias a Santa por su paciencia para escuchar mis divagaciones monotemáticas durante estos años, por soportar mi mal genio en los momentos de desespero y sobre todo por seguir siendo mi dulce hogar a pesar del tiempo y la distancia.



# TABLA DE CONTENIDO

0	Introducción.....	1
1	Tecnologías y traducción.....	8
1.1	El software libre como alternativa al software privativo.....	8
1.1.1	Oportunidades.....	14
1.1.2	Barreras y riesgos.....	22
1.2	Clasificación de las tecnologías de la traducción.....	26
1.3	Las necesidades de los traductores.....	40
1.3.1	Escenarios de uso.....	40
1.3.2	Los sistemas de memorias de traducción.....	41
1.3.2.1	Filtros de conversión.....	43
1.3.2.2	Coincidencias.....	45
1.3.2.3	El editor.....	46
1.3.2.4	El proceso de traducción en un sistema de memorias de traducción.....	47
1.4	Tecnologías libres para la traducción.....	52
1.5	Recapitulación.....	56
2	Calidad de software.....	59
2.1	Generalidades.....	59
2.2	Evaluación de las tecnologías de la traducción.....	67
2.2.1	La propuesta metodológica del grupo EAGLES.....	69
2.2.2	Adaptaciones de las normas ISO y la propuesta de EAGLES.....	74
2.2.2.1	Höge (2002).....	75
2.2.2.2	Rico (2001).....	76
2.2.2.3	Mašlanko (2004).....	77
2.2.2.4	Filatova (2010).....	78
2.2.2.5	Guillardeau (2009).....	80
2.3	Evaluación de software libre.....	80
2.3.1	Modelos de calidad de primera generación.....	81
2.3.1.1	El proyecto Qualification and Selection of Open Source Software (QSOS).....	82
2.3.1.2	El proyecto Business Readiness Rating (BRR).....	86
2.3.2	Modelos de calidad de segunda generación.....	90
2.3.2.1	El proyecto Quality Platform for Open Source Software (QualiPSo).....	91
2.3.2.2	El proyecto Software Quality Observatory for Open Source Software (SQO-OSS).....	94
2.3.3	Otras propuestas.....	96
2.3.3.1	Wheeler (2008).....	96
2.3.3.2	Van den Berg (2005).....	98
2.3.3.3	La Certificación de calidad del Centro de Excelencia de Software Libre de Castilla-La Mancha.....	100
2.3.3.4	El proyecto Open Source Software Management Framework.....	104
2.3.3.5	Modelos de éxito usados en el área de Sistemas de Información.....	106
2.4	Recapitulación.....	110

3 Trabajo empírico.....	123
3.1 Objetivos.....	123
3.2 Fase 1: Elaboración del catálogo de software libre para traductores.....	125
3.2.1 Criterios de selección de los programas.....	125
3.2.2 Diseño de la ficha para almacenar los datos.....	126
3.2.3 Clasificación de los programas.....	127
3.2.4 Fuentes utilizadas para la búsqueda.....	128
3.2.5 Recolección de datos.....	132
3.2.6 Resultados de la búsqueda, selección y clasificación de los programas.....	133
3.2.7 Difusión del catálogo.....	135
3.2.7.1 El catálogo en Internet.....	135
3.2.7.2 Artículos publicados.....	137
3.3 Fase 2: Definición del método de evaluación.....	141
3.3.1 Objetivo.....	141
3.3.2 Usuarios previstos.....	141
3.3.3 Perfil del evaluador.....	141
3.3.4 Actividades y pasos.....	142
3.3.5 El modelo de calidad.....	143
3.3.5.1 Categorías y criterios.....	144
3.3.5.1.1 Calidad del proyecto.....	144
3.3.5.1.2 Calidad del producto.....	145
3.3.5.2 Atributos y métricas.....	147
3.3.5.3 Procedimientos para la consolidación de resultados.....	157
3.3.6 Instrumentos.....	159
3.3.6.1 Instrumento de evaluación.....	159
3.3.6.2 Interfaz de selección.....	162
3.4 Fase 3: Evaluación de las tecnologías libres para la traducción.....	164
3.4.1 Descripción de la muestra.....	164
3.4.2 Recolección de datos.....	166
3.4.2.1 Caracterización de los proyectos de desarrollo.....	166
3.4.2.2 Evaluación del software como producto.....	167
3.4.3 Resultados de la evaluación.....	169
3.4.3.1 Caracterización de los proyectos.....	170
3.4.3.1.1 Anaphraseus.....	170
3.4.3.1.2 Autshumato ITE.....	174
3.4.3.1.3 EsperantiloTM.....	178
3.4.3.1.4 ForeignDesk.....	180
3.4.3.1.5 Frankenstein Translation Memory Suite.....	183
3.4.3.1.6 OmegaT.....	185
3.4.3.1.7 OmegaT+.....	190
3.4.3.1.8 Open Language Tools.....	193
3.4.3.1.9 OpenTM2.....	197
3.4.3.1.10 Transolution.....	200
3.4.3.1.11 Virtaal.....	202
3.4.3.2 Evaluación del software como producto.....	207
3.4.3.2.1 Anaphraseus.....	207
3.4.3.2.2 Autshumato ITE.....	210
3.4.3.2.3 OmegaT.....	213

3.4.3.2.4 OmegaT+.....	217
3.4.3.2.5 Open Language Tools.....	220
3.4.3.2.6 OpenTM2.....	223
3.4.3.2.7 Virtaal.....	226
3.4.3.2.8 Comparativa de funcionalidad.....	229
3.4.4 Discusión.....	233
3.5 Resumen y conclusiones.....	239
Bibliografía citada.....	243
Cibergrafía.....	253
Anexo 1: Listado de programas incluidos en el catálogo de software libre para traductores	255
Anexo 2: Instrumento de evaluación.....	261
Anexo 3: Protocolo de aplicación del instrumento de evaluación.....	278
Anexo 4: Propiedades de los proyectos evaluados.....	281

## Índice de figuras

Figura 1.1: Principales categorías de software libre y no libre (fuente: Free Software Foundation, s. f.-b) [sic].....	15
Figura 1.2: Anuncio de PangeaMT en la edición de abril-mayo de 2012 de la revista Multilingual.....	21
Figura 1.3: Categorización de las tecnologías de la traducción según Lagoudaki (2008).....	29
Figura 1.4: Los ocho tipos de tecnologías de la traducción según Melby (1998).....	31
Figura 1.5: Las herramientas y el proceso de traducción según Auster Mühl (2001).....	32
Figura 1.6: Actividades de los traductores independientes y TIC según Fulford y Granell Zafra (2005).....	34
Figura 2.1: Calidad interna y externa del software según el estándar ISO 9126 (2001).....	64
Figura 2.2: Calidad en uso según el estándar ISO 9126 (2001).....	65
Figura 2.3: Atributos no funcionales del modelo de calidad del proyecto QSOS (fuente: Deprez & Alexandre, 2008, p. 3).....	83
Figura 2.4: Ejemplo de puntuaciones posibles para un atributo binario tomado de la plantilla del proyecto BRR.....	89
Figura 2.5: Ejemplo de puntuaciones propuestas para una atributo que permite un rango pero no utiliza los 5 valores de la escala.....	90
Figura 2.6: Esquema de la metodología GQM (objetivo-pregunta-métrica) (fuente: Petrinja et al., 2008, p. 13).....	93
Figura 2.7: Modelo de calidad propuesto por el observatorio de calidad del software libre (SQO-OSS).....	95
Figura 2.8: Diagrama de los actores y posibles casos de uso de un sistema (fuente: CESLCAM, s. f.-b, p. 31).....	103
Figura 2.9: Proceso de evaluación propuesto por Helander et al. (2007, p. 36).....	106
Figura 2.10: Modelo propuesto por Lee et al. (2009, p. 430).....	108
Figura 3.1: Programas del catálogo desglosados por categoría.....	134
Figura 3.2: Formulario de consulta del Centro Virtual del Observatorio de Tecnologías de la Traducción. ....	136
Figura 3.3: Formato utilizado en la wiki para presentar las herramientas.....	137
Figura 3.4: Búsqueda en el catálogo por propiedades semánticas y valores.....	138
Figura 3.5: A la izquierda: edición de una página del catálogo mediante sintaxis wiki; a la derecha, edición de la misma página mediante formularios. ....	140
Figura 3.6: Características y subcaracterísticas de la calidad del proyecto.....	145



Figura 3.7: Características y subcaracterísticas de la calidad del producto.....	146
Figura 3.8: Instrumento de evaluación – Estrategia del proyecto.....	161
Figura 3.9: Instrumento de evaluación – Portabilidad.....	162
Figura 3.10: Captura de pantalla de la interfaz de selección de herramientas.....	163
Figura 3.10: Ejemplo de ficha de resultados en la wiki.....	170
Figura 3.11: Acerca de Anaphraseus.....	171
Figura 3.12: Página web del proyecto Anaphraseus. ....	172
Figura 3.13: Puntuación de Anaphraseus en SourceForge según sus usuarios.....	173
Figura 3.14: Comentarios sobre Anaphraseus en Twitter.....	174
Figura 3.15: Acerca de Autshumato ITE.....	175
Figura 3.16: Página web del proyecto Autshumato.....	176
Figura 3.17: Puntuación de Autshumato ITE en SourceForge según sus usuarios.....	178
Figura 3.18: Puntuación de ForeignDesk en SourceForge según sus usuarios.....	182
Figura 3.19: Perfil de ForeignDesk en Ohloh.net.....	182
Figura 3.20: Acerca de OmegaT.....	186
Figura 3.21: OmegaT – filosofía del proyecto.....	187
Figura 3.22: Página web del proyecto OmegaT.....	187
Figura 3.23: OmegaT – versiones lanzadas en 2011.....	188
Figura 3.24: Puntuaciones de OmegaT en SourceForge según sus usuarios.....	189
Figura 3.25: Acerca de OmegaT+.....	191
Figura 3.26: Puntuaciones de OmegaT+ en SourceForge según sus usuarios.....	193
Figura 3.27: Acerca de Open Language Tools.....	194
Figura 3.28: Open Language Tools – filosofía del proyecto.....	195
Figura 3.29: Open Language Tools – página del proyecto en la forja Java.net.....	195
Figura 3.30: Open Language Tools – historial de versiones.....	196
Figura 3.31: Acerca de OpenTM2.....	198
Figura 3.32: OpenTM2 – página web del proyecto.....	199
Figura 3.33: Acerca de Virtaal.....	203
Figura 3.34: Virtaal – página web.....	205

## Índice de tablas

Tabla 1.1: Clasificación de las principales licencias libres según su permisividad.....	10
Tabla 1.2: Tipos de formatos que se suelen procesar en los sistemas de memorias de traducción.....	44
Tabla 2.1: Tipos de pruebas en la evaluación orientada a los usuarios (fuente: Höge, 2002, p. 114).....	74
Tabla 2.2: Categorías utilizadas por Mašlanko (2004) para la evaluación de los módulos de gestión terminológica.....	78
Tabla 2.3: Métricas utilizadas por Filatova (2010) para la evaluación cuantitativa de las herramientas para el traductor.....	79
Tabla 2.4: Atributos funcionales para evaluar los procesadores de texto según el proyecto QSOS (fuente: <a href="http://www.qsos.org">http://www.qsos.org</a> ).....	85
Tabla 2.5: Posibles puntuaciones para un atributo no funcional según la propuesta del proyecto QSOS (fuente: Atos Origin, 2006, p. 19).....	85
Tabla 2.6: Perfiles para los atributos del criterio de Analizabilidad en el proyecto SQO-OSS. ....	96
Tabla 2.7: Aspectos específicos a tener en cuenta para la evaluación de software libre según Wheeler (2008).....	98
Tabla 2.8: Resumen de los criterios utilizados por van den Berg (2005) para la evaluación de software libre.....	100
Tabla 3.1: Criterios para la selección de los programas para el catálogo.....	126
Tabla 3.2: Modelo de ficha que se utilizó para el catálogo.....	127
Tabla 3.3: Categorías utilizadas para clasificar los programas seleccionados y tipos de programas que corresponden a cada una de ellas.....	128
Tabla 3.4: Ficha de uno de los programas del catálogo.....	132
Tabla 3.5: Programas del catálogo desglosados según el tipo de licencia.....	133
Tabla 3.6: Programas del catálogo desglosados por sistema operativo soportado.....	135
Tabla 3.7: Atributos para determinar la estrategia del proyecto.....	149
Tabla 3.8: Atributos para caracterizar la comunidad del proyecto.....	150
Tabla 3.9: Atributos para determinar la madurez del proyecto.....	151
Tabla 3.10: Atributos para determinar la reputación del proyecto.....	151
Tabla 3.11: Atributos para evaluar la portabilidad del producto.....	153
Tabla 3.12: Atributos para evaluar la usabilidad del producto.....	155
Tabla 3.13: Atributos para evaluar la funcionalidad del producto.....	157

Tabla 3.14: Indicadores de calidad de la comunidad y madurez del proyecto.....	158
Tabla 3.15: Indicadores de calidad para la portabilidad y la usabilidad.....	159
Tabla 3.16: Valoración general del proyecto Anaphraseus.....	174
Tabla 3.17: Valoración general del proyecto Autshumato.....	178
Tabla 3.18: Valoración general del proyecto EsperantiloTM.....	180
Tabla 3.19: Valoración general del proyecto ForeignDesk.....	183
Tabla 3.20: Valoración general del proyecto Frankestein Translation Memory Suite.....	185
Tabla 3.21: Valoración general del proyecto OmegaT.....	190
Tabla 3.22: Valoración general del proyecto OmegaT+.....	193
Tabla 3.23: Valoración general del proyecto Open Language Tools.....	197
Tabla 3.24: Valoración general del proyecto OpenTM2.....	200
Tabla 3.25: Valoración general de Transolution.....	202
Tabla 3.26: Valoración general del proyecto Virtaal.....	206
Tabla 3.27: Anaphraseus – portabilidad.....	208
Tabla 3.28: Anaphraseus – usabilidad.....	209
Tabla 3.29: Anaphraseus – funcionalidad.....	210
Tabla 3.30: Valoración general de la herramienta Anaphraseus.....	210
Tabla 3.31: Autshumato ITE – portabilidad.....	211
Tabla 3.32: Autshumato ITE – usabilidad.....	212
Tabla 3.33: Autshumato ITE – funcionalidad.....	213
Tabla 3.34: Valoración general de la herramienta Autshumato ITE.....	213
Tabla 3.35: OmegaT – portabilidad.....	214
Tabla 3.36: OmegaT – usabilidad.....	215
Tabla 3.37: OmegaT – funcionalidad.....	216
Tabla 3.38: Valoración general de la herramienta OmegaT.....	217
Tabla 3.39: OmegaT+ – portabilidad.....	218
Tabla 3.40: OmegaT+ – usabilidad.....	218
Tabla 3.41: OmegaT+ – funcionalidad.....	219
Tabla 3.42: Valoración general de la herramienta OmegaT+.....	220
Tabla 3.43: Open Language Tools – portabilidad.....	221
Tabla 3.44: Open Language Tools – usabilidad.....	221
Tabla 3.45: Open Language Tool – funcionalidad.....	222
Tabla 3.46: Valoración general de la herramienta Open Language Tools.....	223
Tabla 3.47: OpenTM2 – portabilidad.....	223

Tabla 3.48: OpenTM2 – usabilidad.....	224
Tabla 3.49: OpenTM2 – funcionalidad.....	225
Tabla 3.50: Valoración general de la herramienta OpenTM2.....	226
Tabla 3.51: Virtaal – portabilidad.....	226
Tabla 3.52: Virtaal – usabilidad.....	227
Tabla 3.53: Virtaal – funcionalidad.....	228
Tabla 3.54: Valoración general de la herramienta Virtaal.....	229
Tabla 3.55: Comparativa de funciones incluidas en las herramientas evaluadas.....	231
Tabla 3.56: Comparativa de formatos soportados por las herramientas evaluadas.....	232

## **CONVENCIONES UTILIZADAS EN ESTE TRABAJO**

Dada la naturaleza técnica del contenido de este trabajo, es frecuente la aparición de siglas y acrónimos. La primera vez que se mencionen se introducirá la forma completa y en las siguientes ocurrencias se utilizarán las siglas. Como referencia para el lector, se incluye en la siguiente página el listado completo de abreviaturas utilizadas.

Cuando se mencionan por primera vez en el texto instituciones, empresas o proyectos que cuentan con una página web oficial, la dirección se incluye como nota al pie de página. Si se trata de una fuente de consulta, la referencia completa aparecerá también en la sección correspondiente a la bibliografía citada. Además, se incluye al final una cibergrafía que recoge las páginas web de interés especial por su relación con el contenido de esta tesis. El listado está organizado en tres secciones: software libre en general, evaluación de software libre y software libre para traductores.



## LISTA DE ABREVIATURAS

ALPAC	Automatic Language Processing Advisory Committee
API	Interfaz de programación de aplicaciones
BRR	Business Readiness Rating
CESLCAM	Centro de Excelencia de Software Libre de Castilla-La Mancha
CMMI	Capability Maturity Model Integration
EAGLES	Expert Advisory Group on Language Engineering Standards
EGW	Evaluation Working Group
FEMTI	Framework for the Evaluation of Machine Translation in ISLE
FOSS	Free/Open-Source Software
FSF	Free Software Foundation
GPL	General Public License
GQM	objetivo-pregunta-métrica ( <i>goal-question-metric</i> )
I+D	Investigación y Desarrollo
ISLE	International Standards for Language Engineering
ISO	International Organisation for Standardization
LETRAC	Language Engineering for Translators Curricula
OMM	OpenSource Maturity Model
OPSOA	OPen SOurce Assessment
OSI	Open Source Initiative
OSMM	Open Source Maturity Model
QSOS	Qualification and Selection of Open Source Software
QualipPSo	Quality Platform for Open Source Software
QualOSS	Quality in Open Source Software
SQO-OSS	Software Quality Observatory for Open Source Software
TEMAA	Test-bed Study of Evaluation Methodologies: Authoring Aids
TI	Tecnologías de la Información
TIC	Tecnologías de la Información y la Comunicación





# PRIMERA PARTE – MARCO TEÓRICO

## 0 INTRODUCCIÓN

- *Justificación*

En los últimos años, el oficio de la traducción como muchos otros ha sido objeto de las transformaciones que las tecnologías de la información y la comunicación han traído consigo. Desde principios de los años noventa el uso de herramientas informáticas por parte de los traductores ha venido aumentando, cosa que también ha sucedido con la cantidad y variedad de herramientas disponibles, que van desde programas generales como editores o procesadores de texto hasta herramientas específicas para traductores como los sistemas de memorias de traducción. Ante el siempre creciente número de herramientas disponibles, el traductor no puede más que preguntarse cuál de ellas se adaptaría mejor a sus necesidades, muchas veces sin contar con parámetros de comparación que le permitan tomar una decisión informada.

Ahora bien, aunque el área de las tecnologías aplicadas a la traducción sin duda ha recibido mucha atención en la literatura científica y profesional, también es cierto que hay un tipo de programas que ha sido relegado a un segundo plano sin recibir la atención que merece: el software libre y de código abierto. A pesar de que este tipo de software en general ha ganado mucho terreno en años recientes, según sugieren los resultados de un estudio realizado por García (2008) para determinar la situación del mercado de las tecnologías de la traducción entre los traductores parece predominar el desconocimiento y la falta de interés por este tipo de

programas. Si bien el estudio de García mostró que un buen número de traductores utilizan software libre para tareas no relacionadas con la traducción, los sistemas de memorias de traducción libres apenas si se empiezan a considerar como opciones viables. En un gremio en el que las herramientas que por años han liderado el mercado cuestan varios cientos de euros parece primar la concepción popular de que algo gratuito difícilmente podrá ser de buena calidad.

Surge entonces la pregunta de cómo facilitar a los traductores la labor de identificar los programas libres que de verdad respondan a sus necesidades. Para obtener una posible respuesta a tal pregunta podemos recurrir a los criterios que se han planteado desde los campos de la ingeniería de software y los sistemas de información, campos en que durante los últimos años se han adaptado metodologías de evaluación que tienen en cuenta las particularidades del software libre y su paradigma de desarrollo.

En el campo de las tecnologías de la traducción la evaluación de software tiene una larga historia que se remonta al informe ALPAC de 1966 sobre el estado de la traducción automática, pero dada la abundancia y diversidad de herramientas y la variedad de grupos de interés y posibles escenarios de uso (industria, administración pública, investigadores, desarrolladores, agencias, traductores independientes, estudiantes, etc.) hacen falta métodos estándar de evaluación que sean fiables, aceptables y reproducibles (Quah, 2006; Rico, 2001; Höge, 2002). Como resalta Quah (2006), en el caso de los sistemas de memorias de traducción muchas veces la evaluación hace parte del proceso de desarrollo de los programas y se lleva a cabo desde el punto de vista de investigadores y desarrolladores, no de los usuarios finales. Además, en muchos casos son las mismas empresas desarrolladoras las que evalúan los programas y, por aquello de la feroz competencia que se da en este campo, por lo general los resultados se consideran confidenciales.

En este sentido, el software libre presenta un escenario diferente, puesto que el modelo de desarrollo basado en la comunidad y en la libre disponibilidad del código fuente hace que el tema de la confidencialidad sea irrelevante. Hay también

otras características que son distintivas del modelo de desarrollo y distribución del software libre, incluyendo aspectos como la naturaleza de la comunidad de usuarios y desarrolladores y el marco ideológico de los proyectos, el nivel de actividad de la comunidad o el tipo de asistencia y servicios disponibles para los usuarios (Wheeler, 2008; van den Berg, 2005).

En este contexto, una evaluación objetiva y detallada de las herramientas libres para el traductor disponibles en la actualidad puede ser una buena manera de difundir el concepto de software libre en nuestro gremio y de promover su uso. Ahora bien, teniendo en cuenta que los métodos de evaluación tradicionalmente utilizados para las tecnologías de la lengua responden a procesos de diseño y ciclos de desarrollo secuenciales o iterativos e incrementales y no continuos como los del software libre (Gasser, Scacchi, Ripoche, & Penne, 2003), se hace necesario plantear una metodología de evaluación integral en la que además del software como producto final se consideren también aspectos del proyecto de desarrollo, como la gestión de la propiedad intelectual, la planificación a futuro, las dinámicas de las comunidades y las tecnologías que las soportan.

- ***Preguntas de investigación***

En este contexto, las preguntas de investigación que motivan el presente trabajo son las siguientes:

- ¿Cuáles son las herramientas libres disponibles en la actualidad que pueden resultar útiles para el trabajo del traductor?
- ¿Cómo caracterizar los proyectos de desarrollo de herramientas libres para traductores?
  - ¿Cómo evaluar la estrategia de los proyectos de desarrollo, las comunidades que los respaldan, la madurez de los procesos y la reputación de estas herramientas?
- ¿Cómo evaluar las herramientas libres para traductores como producto?

- ¿Cómo evaluar las características de funcionalidad, portabilidad y usabilidad de estas herramientas?

- **Objetivos**

El objetivo del presente trabajo es proponer una metodología e instrumento para la evaluación de las tecnologías libres para la traducción que permita a los traductores seleccionar programas con base en sus necesidades particulares. Para ello, se elabora un inventario de herramientas libres que pueden resultar útiles para los traductores. La metodología de evaluación propuesta incluirá un análisis de las características del software como producto, así como de la comunidad de usuarios y desarrolladores que soporta cada proyecto como indicador de la madurez del proceso de desarrollo y la sostenibilidad del proyecto.

Este objetivo general se descompone a su vez en distintas fases, que dan lugar a resultados específicos:

- a) Elaborar un inventario de programas libres útiles para el traductor.
- b) Preparar un instrumento de evaluación para los programas libres de ayuda a la traducción y una interfaz de selección que sirva para facilitar la toma de decisiones de los traductores a la hora de escoger programas libres.
- c) Validar el instrumento mediante la caracterización de una muestra de proyectos de desarrollo de tecnologías libres para la traducción y la evaluación de los programas.

Los resultados serán pues:

- a) Un catálogo de software libre para el traductor.
- b) Un instrumento de evaluación validado mediante su aplicación a una serie de proyectos y herramientas libres.
- c) Una muestra de evaluación de once proyectos y siete herramientas libres.

- *Enfoque metodológico*

Para lograr estos objetivos se hace necesario empezar por realizar una reseña de la literatura sobre tecnologías de la traducción para encontrar una manera de limitar y clasificar los programas a considerar. Con el mismo fin, se recoge también información sobre el software libre y otros tipos de software según sus licencias y se exploran las implicaciones éticas y sociales de estas últimas. Finalmente, se examina la literatura sobre evaluación de software para conocer los criterios y procedimientos que los investigadores en el área han determinado, con especial atención a la evaluación de las tecnologías de la traducción y del software libre.

A partir de los criterios de evaluación encontrados en la literatura reseñada se consideran aquellos que sean aplicables a la evaluación de software libre con miras a la selección de herramientas para su utilización en el entorno de trabajo del traductor. En la primera fase del trabajo se determinan los criterios mínimos que deben cumplir los programas para ser incluidos en el inventario de candidatos a evaluación, así como la información que debe recoger la ficha de cada programa, y luego se procede con la búsqueda, selección y clasificación de los programas. El catálogo resultante está disponible en Internet y se ha difundido en publicaciones especializadas.

En la segunda fase se deciden los criterios para la evaluación teniendo en cuenta la motivación para la evaluación y el contexto específico en el que se plantea llevarla a cabo, lo que incluye el tipo de herramientas en cuestión, los usuarios finales y el entorno de uso previsto para las herramientas. Dadas las características particulares del software libre, la evaluación que aquí se plantea abarcará tanto las herramientas en sí mismas (evaluación del producto) como los proyectos que las respaldan (caracterización de las comunidades y sus procesos de desarrollo). A partir de los criterios establecidos se crean los instrumentos de evaluación y selección con base en los modelos y recomendaciones encontrados en la literatura y se define el procedimiento para llevar a cabo la evaluación.

La tercera y última fase consiste en establecer la muestra de proyectos a ser evaluados para luego aplicar el instrumento a dicha muestra según el procedimiento

definido en la fase anterior. Una vez recogidos los datos se analizan los resultados de cada programa, se totalizan las puntuaciones correspondientes a la evaluación del producto, se resumen las características de los proyectos en términos de sus fortalezas y debilidades y se publican los resultados. Para cada proyecto evaluado se prepara una ficha con los datos que se agrega al catálogo creado en la primera fase.

- ***Estructura***

El presente trabajo se divide en dos partes. La primera parte proporciona los antecedentes teóricos necesarios mediante una reseña de la literatura relevante organizada en dos capítulos. El capítulo 1, que aborda el área de las tecnologías de la traducción, se divide en cinco subapartados. En el primero de ellos (1.1) se aclaran los conceptos básicos relacionados con los tipos de software según sus licencias, así como las oportunidades y riesgos asociados al uso de software libre. En el subapartado 1.2 se presentan diferentes perspectivas para la clasificación de las tecnologías de la traducción, mientras que en el subapartado 1.3 se recogen las principales necesidades de los traductores en términos de los escenarios de uso previstos y las características generales de los sistemas de memorias de traducción. En el subapartado 1.4 se presentan los principales recursos disponibles en la actualidad sobre las tecnologías libres para la traducción, algunos surgidos desde el ámbito profesional y otros desde el ámbito académico. Para terminar, el subapartado 1.5 presenta la recapitulación de los aspectos tratados en el capítulo.

El segundo capítulo de la primera parte describe los principales enfoques que se han utilizado para la evaluación de software, empezando con las generalidades (2.1) para pasar luego a la evaluación de las tecnologías de la traducción (2.2) y finalmente de software libre (2.3); para cerrar el capítulo, en el subapartado 2.4 se presenta un resumen donde se retoman los aspectos más relevantes a la hora de definir la metodología para la evaluación que se plantea en este trabajo.

La segunda parte corresponde a un capítulo que detalla el trabajo empírico y que se organiza en cinco subapartados. En el primero de ellos (3.1) se presenta una introducción al trabajo realizado. En el segundo subapartado (3.2), se detalla el

procedimiento para la búsqueda, selección y clasificación de los programas para el catálogo de software libre para el traductor. El tercer subapartado (3.3) corresponde a la preparación de la evaluación; mientras que en el cuarto subapartado (3.4) se describe el procedimiento seguido para llevar a cabo la evaluación. En el último subapartado (3.5) se presentan el resumen y las conclusiones del proceso adelantado y se esbozan recomendaciones para trabajos futuros.

# 1 TECNOLOGÍAS Y TRADUCCIÓN

En este capítulo se aborda el tema de las tecnologías aplicadas a la traducción. En el primer apartado se definen los conceptos básicos en relación con las licencias de software y se consideran las oportunidades y riesgos asociados al software libre. En el segundo apartado consideraremos algunas de las propuestas de clasificación de las tecnologías aplicadas a la traducción. En el tercer apartado exploraremos la funcionalidad que ofrecen la mayoría de sistemas de memorias de traducción disponibles en la actualidad. A continuación, en el cuarto apartado presentaremos algunos esfuerzos emprendidos desde las dimensiones profesional y académica de la traducción para promover el uso de software libre en nuestro gremio. Finalmente, en el quinto apartado haremos la recapitulación de los aspectos abordados en este primer capítulo.

## 1.1 El software libre como alternativa al software privativo

El movimiento del **software libre** (en inglés *free software*) se inició a principios de la década de los ochenta como una iniciativa basada en la idea de que el usuario de software debe tener garantizadas cuatro libertades fundamentales: a) la libertad de usar el programa con cualquier fin; b) la libertad de entender cómo trabaja el programa y poder adaptarlo a sus necesidades; c) la libertad de distribuir copias del programa; y d) la libertad de mejorar el programa y distribuir copias de la versión mejorada (Fernández, 2003; James, 2003; Kooths, Langenfurth, & Kalwey, 2003; McKay, 2005). Se entiende, pues, que tener acceso al programa escrito en lenguaje de programación (lo que se conoce como *código fuente*) es un prerrequisito para b) y d). Hoy por hoy, el software libre es un importante movimiento social y un fenómeno comercial en el que participan cientos de miles de programadores y una gran diversidad de empresas de desarrollo de software en todo el mundo (Crowston, Wei, Howison, & Wiggins, 2011).



La filosofía que promueve el movimiento del software libre está basada en el libre acceso al conocimiento. Esta filosofía está en directa oposición a la del software privativo comercial, cuya instalación, uso, copia y distribución están considerablemente restringidas según se estipula en sus licencias (Free Software Foundation, s. f.-a), llegando por ejemplo al extremo de prohibir la instalación en más de un ordenador, tener que pagar por las actualizaciones o a obligar a los usuarios a someterse a auditorías de licencias si el fabricante lo considera necesario.

A principios de la década de los ochenta, con la implantación de las leyes de propiedad intelectual en el ámbito del software y su creciente comercialización la única posibilidad legal si se quería lograr la libertad promovida por Richard Stallman (líder del movimiento de software libre) era declarar el programa como de **dominio público**, lo que básicamente quiere decir que el autor renuncia a todos sus derechos y el usuario adquiere derechos ilimitados de explotación (Kooths et al., 2003). Sin embargo, si el código fuente pasa a ser de dominio público cualquiera puede apropiarse de él para continuar su desarrollo como software privativo. Como solución a este dilema, a mediados de los años ochenta Stallman creó la *Free Software Foundation*<sup>1</sup> (en adelante FSF) y redactó la Licencia Pública General (GPL, del inglés *General Public License*), que garantiza a los usuarios las cuatro libertades arriba mencionadas sin que el autor renuncie a sus derechos, permitiendo a la vez garantizar que las libertades de los usuarios no sean restringidas en las versiones modificadas. La GPL, sin embargo, no es la única licencia libre y la FSF, como veremos más adelante, no es la única entidad dedicada a la promoción del software libre.

Según aclara la FSF, un aspecto que tiende a crear confusión es que las licencias libres no necesariamente implican que los programas tengan que ser gratuitos, pero sí en muchos casos que las copias y redistribuciones de los programas deben estar disponibles para el público con el mismo tipo de licencia del programa original. De esta manera se pretende impedir que un programa desarrollado originalmente de manera libre sea convertido posteriormente en un programa privativo. Esto es lo que se conoce como *copyleft*, en contraposición al

---

<sup>1</sup> <http://fsf.org>

modelo tradicional de derecho de autor, *copyright*<sup>2</sup>. Sin embargo, no todas las licencias libres incluyen cláusulas de *copyleft* y las que sí lo hacen se pueden clasificar en términos de su permisividad como *copyleft* fuerte o *copyleft* débil (Helander et al., 2007).

La GPL, por ejemplo, es una licencia con *copyleft* fuerte, lo que quiere decir que las versiones derivadas de los programas heredan siempre la misma licencia. Las licencias con *copyleft* débil, por otro lado, permiten que ciertas versiones derivadas se puedan redistribuir con limitaciones adicionales a las de la licencia original pero siempre incluyen acceso al código fuente; es el caso de la *GNU Lesser General Public License* o la *Mozilla Public License*. Las licencias sin cláusula de *copyleft*, como la licencia BSD (*Berkeley Software Distribution License*), son las más permisivas de las licencias libres ya que para las versiones modificadas no se incluye de manera explícita la condición de publicar el código fuente (Kooths et al., 2003). En la tabla 1.1 se presentan las licencias libres más comunes clasificadas según su permisividad<sup>3</sup>.

Permisividad	Licencia
<b>Sin <i>copyleft</i></b>	Berkeley Software Distribution License (BSD) Apache License Academic Free License MIT License Creative Commons Attribution (CC-BY)
<b><i>Copyleft</i> débil</b>	Common Development and Distribution License (CDDL) Mozilla Public License GNU Lesser General Public License (LGPL) Microsoft Public License Common Public Attribution License
<b><i>Copyleft</i> fuerte</b>	Affero General Public License (AGPL) GNU General Public License (GPL) IBM Public License Open Software License (OSL) Eclipse Public License Creative Commons Attribution-Share Alike (CC BY-SA)

Tabla 1.1: Clasificación de las principales licencias libres según su permisividad.

Pasamos ahora a aclarar la distinción entre software libre y software de **código abierto** u *open source*. En términos generales, *código abierto* hace énfasis

2 Cabe mencionar que aunque en español es posible utilizar *izquierdos de copia* o *izquierdos de autor*, es más común hacer uso del término en inglés (ver por ejemplo <http://fundacioncopyleft.org/es/9/que-es-copyleft>).

3 Para una lista más completa de licencias libres clasificadas según su permisividad ver [http://www.ifross.org/ifross\\_html/lizenzcenter-en.html](http://www.ifross.org/ifross_html/lizenzcenter-en.html).

en la metodología de desarrollo, mientras que *software libre* hace referencia a un movimiento ideológico y social. Según la misma FSF, “el software libre es un movimiento político, el código abierto es un modelo de programación” (Free Software Foundation, s. f.-a). Así pues, la Open Source Initiative<sup>4</sup> (en adelante OSI), surgida en 1997, es una versión matizada ideológicamente con el objeto de evitar entrar en consideraciones éticas y así atraer a los empresarios para que utilicen programas libres haciendo énfasis en que no están obligados a compartir las versiones mejoradas mientras no las distribuyan. En el fondo, entonces, se trata de una diferencia filosófica (May, 2006), ya que en la práctica es la misma idea y de hecho los dos movimientos tienen aprobadas básicamente las mismas licencias con unas pocas excepciones (Free Software Foundation, s. f.-a; Kooths et al., 2003).

De ahí que en inglés haya surgido el acrónimo *FOSS* (*Free/Open Source Software*) para evitar ambigüedades y abarcar las dos propuestas, aunque algunos autores prefieren utilizar *OSS/FS* (*Open Source Software/Free Software*), *FLOSS* (*Free/Libre/Open Source Software*) o simplemente utilizar uno de los dos calificativos, libre o de código abierto, para referirse indistintamente a ambas iniciativas.

De modo, pues, que otra de las particularidades del software libre y de código abierto, además de las licencias, es su modelo de desarrollo llevado a cabo por la comunidad y para la comunidad. Pero no todos los proyectos son iguales y de hecho existe bastante variación en cuanto a las características y número de miembros involucrados, así como a los tipos de procesos de toma de decisiones y formas de organización.

En algunos casos, la comunidad de desarrollo está compuesta principalmente por voluntarios que se rigen por lo que se denomina *ética hacker*. Este término hace referencia al hecho de que los colaboradores trabajan por diversión, así como al hecho de que muchos proyectos surgen como soluciones a una necesidad particular de un desarrollador, sin que exista una remuneración económica por el trabajo. En el otro extremo del continuo encontramos los proyectos liderados por empresas,

---

4 <http://opensource.org>

donde por lo menos algunos de los desarrolladores se rigen por la ética empresarial tradicional basada en la remuneración económica.

En cuanto al número de participantes, existen proyectos que cuentan con un solo desarrollador, mientras que en el otro extremo encontramos los megaproyectos como LibreOffice que cuentan con miles de colaboradores distribuidos por todo el mundo. Independiente del número de colaboradores, se habla de distintos tipos de liderazgo y formas de organización, que van desde las dictaduras benevolentes<sup>5</sup>, en las que un solo desarrollador o grupo cerrado de desarrolladores principales lidera el proyecto y se encarga de tomar las decisiones de manera centralizada, hasta los proyectos meritocráticos o democráticos sin una jerarquía predefinida donde las decisiones se toman de forma descentralizada.

Otra aclaración pertinente tiene que ver con la tendencia a asociar el concepto de *software libre* con el sistema operativo basado en el núcleo o *kernel* Linux. Ya que el núcleo es solo una parte del sistema operativo, la FSF recomienda utilizar la denominación *GNU/Linux* para referirnos al sistema operativo, que incluye también aplicaciones desarrolladas como parte del proyecto GNU. Este sistema operativo, disponible en diferentes variantes conocidas como **distribuciones** o **distros**, está compuesto fundamentalmente por software libre, aunque algunas distribuciones incluyen también software privativo preinstalado. Pero no todo el software libre es exclusivamente para este sistema operativo y naturalmente el usuario final también puede instalar software privativo incluso en las distribuciones de GNU/Linux que son 100% libres (Mas, 2003; Rosa & Heinz, 2007). Para promover, proteger y fomentar el desarrollo del kernel Linux en el año 2000 se creó la Linux Foundation<sup>6</sup>, una organización sin ánimo de lucro que además organiza eventos y ofrece formación especializada.

Pasamos ahora al otro extremo, es decir, al *software no libre* o *software privativo*. El primer aspecto a aclarar tiene que ver con la abundancia de términos que se utilizan para referirse a este concepto. Encontramos, pues, las

---

5 En este caso el término *dictadura benevolente* hace referencia al hecho de que aquellos que se someten al régimen del dictador lo hacen de manera voluntaria.

6 <http://www.linuxfoundation.org/>

denominaciones de *software privativo*, *propietario*, *privado*, *de marca registrada*, *con propietario* o *de propiedad*, llegando incluso a aparecer en un mismo texto varios de estos términos de manera indiferente. Algunas veces de manera errónea, incluso en la literatura especializada, se utiliza también *software comercial* para referirse al mismo concepto. Para los intereses del presente trabajo, se utilizará el término *software privativo*, por ser este el preferido por la FSF y porque de los adjetivos antes mencionados *privativo* es el que más sentido tiene en español ya que según el diccionario de la Real Academia Española su significado es “que causa privación o la significa”, privación que en este caso se refiere a las libertades de los usuarios.

También es importante aclarar el concepto de *software comercial* ya que la creencia general es que esto es sinónimo de software privativo y, como se mencionó antes, que el software libre o de código abierto es por definición no comercial. En realidad, el adjetivo *comercial* es aplicable a cualquier programa desarrollado por una empresa que espera obtener ganancias económicas (Free Software Foundation, s. f.-a; Fuggetta, 2003). Así, aunque parezca contradictorio, también existe el software libre comercial y el software privativo no comercial. En el caso del software libre comercial el acceso al código fuente está garantizado, pero se pueden obtener ganancias, por ejemplo, por la distribución de copias, por la asistencia técnica profesional o por ofrecer el software como servicio (Mas, 2003).

Por otro lado, el software privativo no comercial, que se conoce generalmente como *freeware* (aunque las definiciones de este término pueden variar), incluye aquellos programas que el usuario puede instalar y muchas veces también copiar y distribuir gratuitamente. No obstante, las condiciones varían según lo estipulado en cada licencia y en general no se cuenta con acceso al código fuente ni se pueden hacer modificaciones al programa (Fuggetta, 2003; Kooths et al., 2003).

Existen también otras variedades como el *adware*, el *shareware* o el *donationware*, casos en los que tampoco se cuenta con acceso al código fuente. El primero, por ejemplo, se distribuye gratuitamente para los usuarios pero se financia

mediante la publicidad que el programa muestra continuamente y su licencia impone restricciones.

El *shareware*, por su parte, se descarga de manera gratuita y generalmente también se puede copiar y distribuir, pero no se tiene acceso al código fuente y se considera solamente una versión de prueba (Kooths et al., 2003). Si el usuario quiere conservar el programa debe adquirir una licencia (Fuggetta, 2003); sin embargo, en muchos casos el programa sigue funcionando indefinidamente sin más limitación que un cuadro de diálogo que aparece al iniciar el programa recordando al usuario que su copia no es legal. Así pues, aunque el *shareware* se puede conseguir gratuitamente, técnicamente no lo es y también responde a un fin comercial, si bien sus licencias generalmente no son muy costosas y en la práctica muchos de sus usuarios no llegan a pagar por la licencia (Free Software Foundation, s. f.-a), como en el caso de los conocidos programas de compresión de archivos WinZip y WinRar.

En cuanto al *donationware*, constituye una variante en la que el programa se puede descargar, copiar y distribuir gratuitamente, pero se sugiere al usuario que haga una donación a los desarrolladores, si bien esta no es de carácter obligatorio (Kooths et al., 2003). Algunos proyectos de software libre recurren a este modelo de financiación, por lo que en este caso el factor decisivo para saber si el programa es libre o no es el tipo de licencia y el acceso al código fuente. En la figura 1.1 se presentan de manera esquemática las principales categorías de software libre y no libre o privativo.

### 1.1.1 Oportunidades

Una de las principales oportunidades que ofrece el software libre es la **independencia estratégica** que permite al evitar depender de un único proveedor de software (Ciolkowski & Soto, 2008). Ya que el código fuente está disponible públicamente, la oferta de servicios basados en el producto, como la asistencia técnica o la adaptación del software a nuestras necesidades, no está limitada a lo que ofrezca el proveedor original.

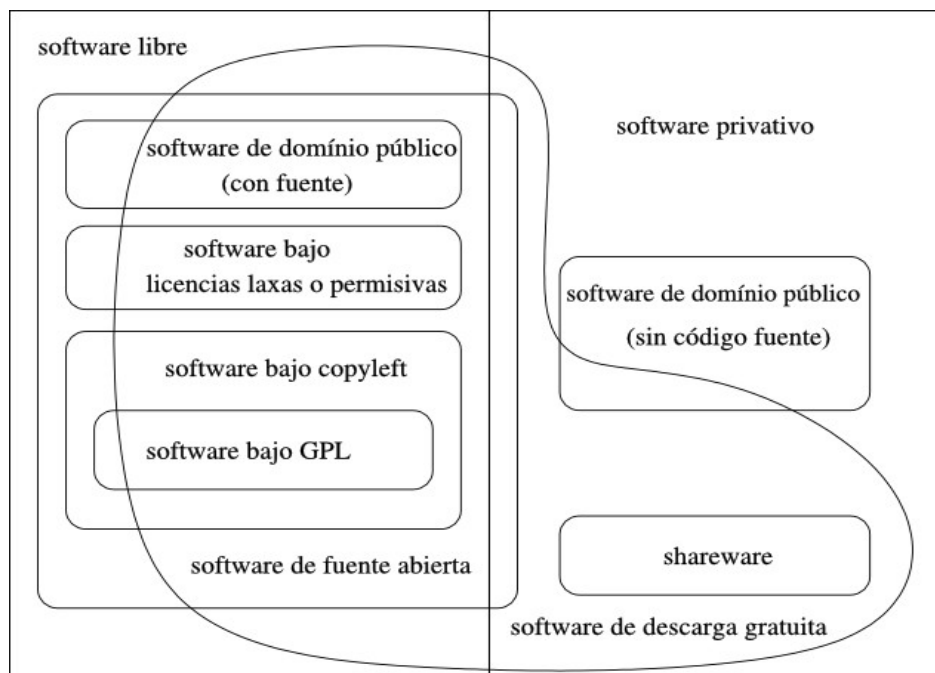


Figura 1.1: Principales categorías de software libre y no libre (fuente: Free Software Foundation, s. f.-b) [sic].

Por su modelo de desarrollo basado en la colaboración, el software libre normalmente parte de un diseño modular que permite a los colaboradores aportar complementos que se desarrollan de manera independiente y se integran luego al sistema. Esto implica, en general, una **mayor flexibilidad** y hace más fácil la adaptación de los programas a las necesidades específicas de los usuarios.

Otra característica del software libre es su tendencia a basar sus formatos nativos en estándares abiertos, mientras que en el software privativo es más común encontrar formatos binarios cuyas especificaciones no siempre están disponibles públicamente, cosa que dificulta su implementación en otras herramientas. Así pues, el software libre **favorece la interoperabilidad** y facilita la posibilidad de acceder a los datos incluso con herramientas desarrolladas por otros proveedores.

Pero la ventaja tal vez más evidente de utilizar software libre tiene que ver con la **reducción de costes** al no tener que pagar por las licencias. No obstante, existen otros gastos asociados a la implementación de software, entre los que se cuentan los recursos humanos necesarios para la instalación de los programas, la capacitación inicial, el soporte técnico, las actualizaciones y el mantenimiento,

además de los costes relacionados con la adquisición y actualización de hardware (McKay, 2005). Con todo, al no tener que pagar por las licencias, el *coste total de propiedad*, que es como se denominan dichos costes asociados, tiende a ser menor, especialmente si se considera que las actualizaciones del software privativo comercial exigen por lo general hardware cada vez más potente. Esta obsolescencia programada reduce prematuramente la vida útil de los ordenadores, lo que a su vez incide en los costes debido a las frecuentes actualizaciones de hardware que se hacen necesarias (James, 2003).

En cuanto a los aspectos **éticos** y **legales**, aunque sin duda son pocos los usuarios promedio que leen las licencias de los programas que instalan en sus ordenadores, estos textos constituyen contratos con implicaciones legales que no porque las desconozcamos dejarán de tener efecto. En el caso del software privativo comercial las licencias individuales por lo general estipulan que el programa no se puede copiar, modificar ni distribuir y que solo se puede instalar en un determinado número de ordenadores.

Es cierto que en muchos casos es posible adquirir paquetes de licencias especialmente concebidos para instituciones, que pueden llegar a ser incluso gratuitos en el caso de las instituciones educativas, pero siguen existiendo limitaciones sobre el número de copias instaladas y la distribución de los programas a terceros. Si bien a nivel individual el incumplimiento de las licencias privativas rara vez representa consecuencias, a nivel empresarial e institucional se ejerce mucho más control y se corre el riesgo de enfrentar demandas y multas por infringir los derechos de propiedad intelectual.

En este sentido, el uso de software libre no tiene esas limitaciones y permite a la vez promover modelos de aprendizaje más autónomos, ya que es posible, por ejemplo, la creación de material didáctico con el que se pueden incluir copias de los programas que los estudiantes serán libres de instalar en sus ordenadores para prácticas fuera de las clases. Con esto se evita también el problema ético que representa para las instituciones educativas el hecho de que muchas veces sea desde las mismas aulas desde donde se promueva la instalación de copias ilegales (Rosa



& Heinz, 2007). En algunos casos esto pasa precisamente por desconocimiento sobre las implicaciones de las licencias pero en otros con plena conciencia de que se trata de copias ilegales. En este sentido, cabe también mencionar que la piratería de software privativo, paradójicamente, es uno de los factores que ha limitado una mayor difusión del software libre al anular su ventaja económica, especialmente en los países en vías de desarrollo donde los índices de piratería suelen ser más altos (James, 2003).

Otras implicaciones del uso de software libre tienen que ver con aspectos de carácter más **social**. La filosofía de la FSF se basa en el libre acceso al conocimiento y el software libre favorece dicho acceso. Primero, la ausencia de costes por licencias permite que los programas estén al alcance de todos. Segundo, el acceso al código fuente permite una mirada más analítica y reflexiva hacia las tecnologías en general, ya que el software deja de verse como un producto terminado para convertirse, si se quiere, en un objeto de experimentación, puesto que cualquiera puede involucrarse en el proceso de desarrollo, sea como usuario o como desarrollador (Rosa & Heinz, 2007). En el caso de instituciones universitarias o empresas, por ejemplo, es posible destinar parte del presupuesto que se ahorra en licencias a la adaptación de un determinado programa a las necesidades específicas.

El desarrollo de los proyectos libres normalmente se lleva a cabo en plataformas en línea denominadas *forjas* (en inglés *forges*)<sup>7</sup>. Estos sitios ofrecen una serie de herramientas para facilitar el desarrollo colaborativo y la gestión de los proyectos. Entre la funcionalidad que normalmente ofrecen las forjas se encuentran los espacios de comunicación (por ejemplo, listas de distribución y foros de usuarios), herramientas colaborativas (como sistemas de control de versiones o gestión de fallos) y estadísticas sobre los proyectos (por ejemplo, número de descargas o de visitas a la página o nivel de actividad del proyecto) (Squire & Williams, 2012).

Este tipo de forjas públicas y la posibilidad de participar en el desarrollo de los proyectos podría ayudar a reducir la brecha digital global ya que permitiría salir

---

<sup>7</sup> Squire y Williams (2012) presentan un estudio descriptivo del ecosistema de forjas de desarrollo de software libre y sus principales características.

del círculo vicioso de dependencia tecnológica en el que tradicionalmente la falta de capacidad tecnológica (en lo que se refiere tanto a uso como a producción) en los países en vías de desarrollo hace que los recursos tecnológicos y el personal calificado para su implementación provengan de países desarrollados, lo que a su vez perpetúa la dependencia (James, 2003; Rosa & Heinz, 2007).

El uso de software libre, en cambio, facilita y a su vez promueve la capacitación de recursos humanos locales y constituye una oportunidad de **desarrollo tecnológico local** adaptado a las necesidades particulares de las comunidades (May, 2006), lo que generalmente ha probado tener mejores resultados que la importación de tecnologías pensadas para funcionar en otros contextos socio-económicos (James, 2003). El software libre puede entonces resultar especialmente atractivo en el contexto de las instituciones universitarias, ya que además de permitir reducir gastos y de ayudar a promover un pensamiento crítico entre los estudiantes también se ajusta fácilmente a sus agendas de investigación y desarrollo (Rosa & Heinz, 2007). De hecho, no es fortuito que desde el año 2001 la UNESCO venga apoyando el movimiento del software libre y se haya unido a la FSF para crear un directorio de programas libres<sup>8</sup>, entre otros proyectos para promover su difusión, una tarea con la que también la Unión Europea se ha comprometido (Díaz Fouces, 2011).

La posibilidad de adaptación a contextos locales, además, está asociada directamente con la traducción, ya que el **aspecto lingüístico** es clave para el libre acceso al conocimiento. A diferencia de los programas privativos, otra de las ventajas de los programas libres es que en muchos casos están disponibles en una gran variedad de lenguas, incluyendo lenguas minoritarias. Incluso cuando no es así cualquiera puede darse a la tarea de localizar un programa libre ya que no se necesita autorización de nadie ni se tiene que considerar la viabilidad económica, que en el caso del software privativo es muchas veces la justificación para que los programas solo se localicen a unas pocas lenguas (Mas, 2003). Así pues, en el mundo del software libre, paralela a las comunidades de usuarios y desarrolladores, existe también la comunidad de traductores.

---

8 <http://directory.fsf.org>

Otra implicación de este fenómeno es que para facilitar los proyectos colaborativos necesarios para localizar los programas libres y su documentación la misma comunidad de software libre se ha encargado de desarrollar herramientas y portales de colaboración especiales para los traductores que, naturalmente, son de libre acceso. Para colaborar en alguno de los proyectos activos, normalmente solo es necesario aceptar que las traducciones pasen a estar cubiertas por la misma licencia del proyecto correspondiente o por otra licencia libre específica para documentación, según sea el caso (Fernández, 2003; Mas, 2003).

Este fenómeno, de manera similar a los movimientos de *fansubs* y traducción de cómics japoneses (*manga*) mencionados por O'Hagan y Ashworth (2002), implica que en la mayoría de los casos los traductores son voluntarios sin formación o experiencia en traducción pero con un interés en promover el software libre y facilitar su difusión. En el aula de traducción, estos entornos colaborativos se pueden aprovechar para proporcionar a los estudiantes ejercicios reales en los cuales su participación, como expertos en traducción entre tantos voluntarios sin conocimientos especializados, además de ser beneficiosa para la comunidad en general puede ayudarles a captar la importancia de la traducción en su dimensión social.

- ***Modelos de negocio basados en software libre***

En el contexto empresarial, el software libre ofrece también interesantes oportunidades para explorar nuevos modelos de negocio. Según explican Helander et al. (2007), en años recientes se ha visto un cambio en el panorama del software libre en el sentido de que cada vez son más las compañías que participan en proyectos activos o inician nuevos proyectos libres, bien liberando código que antes era privativo o involucrándose en el desarrollo de proyectos desde su inicio.

En un estudio reciente, Daffara (2009) identificó nueve estrategias de negocio utilizadas por empresas que basan parte de sus ingresos económicos en el software libre. De estos modelos, que no son mutuamente excluyentes, dos se basan directamente en la distribución de software, otros seis se basan en la oferta de servicios y el último se basa en el ahorro en costes de ingeniería. A continuación

describiremos brevemente estos modelos de negocio con base en la explicación que ofrece Daffara (2009).

**Licenciamiento dual:** este modelo de negocio, basado en la distribución de software y por tanto limitado a las empresas del área de las tecnologías de la información (TI), consiste en distribuir el mismo programa bajo dos licencias diferentes, una libre (normalmente la GPL) y otra privativa. De este modo, los clientes que quieran utilizar el programa para producir software derivado sin comprometerse a liberar el código resultante deben adquirir la licencia privativa.

**Núcleo abierto:** en este modelo, también limitado a las empresas del sector TI, el producto principal está disponible bajo una licencia libre, pero se ofrecen también una serie de extensiones o plugins solo disponibles bajo una licencia privativa. Un ejemplo relevante en nuestro contexto es la empresa germanoespañola ]project-open[, desarrolladora del sistema de gestión de proyectos libre del mismo nombre. Dicha herramienta incluye módulos especializados para gestionar el flujo de trabajo de los proyectos de traducción y localización<sup>9</sup> y la empresa ofrece además tres extensiones privativas diseñadas para satisfacer las necesidades de las grandes empresas de traducción.

**Especialistas en un producto:** en este caso se trata de empresas que han creado o participan en el desarrollo de un programa libre y que obtienen ingresos mediante la oferta de servicios como capacitación, asesoría, instalación, configuración, integración, desarrollo personalizado, mantenimiento o software como servicio (SaaS, por sus siglas en inglés). La empresa ]project-open[, mencionada en el apartado anterior, complementa su modelo de núcleo abierto con servicios de formación, asistencia técnica y software como servicio. Otros dos ejemplos en el área de la traducción son Spartan Consulting<sup>10</sup> y PangeaMT<sup>11</sup>. La primera es una empresa estadounidense que basa su modelo de negocio en el programa libre GlobalSight, un sistema de gestión de traducciones (TMS) basado en web que ofrecen en la modalidad de software como servicio. PangeaMT, por otro

---

9 <http://project-open.com/en/solutions/project-open-translation-mangement-system.html>

10 <http://www.globalsightsaas.com>

11 <http://www.pangea.com.mt/es>

lado, es una división de la empresa española Pangeanic que ofrece servicios de instalación, configuración, adaptación, integración y entrenamiento de soluciones personalizadas de traducción automática basadas en Moses, un sistema libre de base estadística (ver anuncio en la figura 1.2).

**From the leaders of Open Source MT:**

- Build your own engines
- **Update at will**
- Retrain with new material
- **Create new engines**
- Control Panel & Stats
- **Customize your own SMT**
- Integrate MT in your workflow
- **SaaS**
- MT API developments for CAT & Web

[pangea.com.mt](http://pangea.com.mt)  
[pangeanic.com](http://pangeanic.com)

April/May 2012 MultiLingual 15

Figura 1.2: Anuncio de PangeaMT en la edición de abril-mayo de 2012 de la revista Multilingual.

**Proveedores de plataforma:** al igual que en el caso anterior las empresas que aplican este modelo de negocio también ofrecen servicios, pero en este caso no se limitan a un solo producto sino que abarcan una serie de productos relacionados que conforman una plataforma. Un ejemplo ampliamente conocido es la empresa RedHat, que obtiene ingresos económicos a partir de los servicios asociados a sus distribuciones de GNU/Linux para empresas. En nuestro campo, ha surgido recientemente un proyecto que pretende proporcionar toda una plataforma de herramientas para la traducción en la nube. Establecido como fundación en Suiza, el proyecto Olanto<sup>12</sup> ofrecerá además capacitación y certificación a una red de socios que luego podrán a su vez ofrecer servicios asociados en sus respectivos países.

**Selección/asesoría:** las empresas que utilizan este modelo de negocio se dedican a ofrecer servicios de evaluación, selección y asesoría sobre una serie de productos de software libre, aunque no siempre participan en su desarrollo.

---

<sup>12</sup> <http://olanto.org>

**Proveedores de asistencia técnica para un conjunto de soluciones:** las empresas que utilizan este modelo de negocio se especializan en ofrecer el servicio de asistencia técnica para diversos programas libres relacionados, bien sea de manera directa o sirviendo de intermediarios entre clientes y empresas especialistas en un producto.

**Certificación y asesoría legal:** este modelo de negocio, que no implica participación en el desarrollo de los programas, consiste en ofrecer servicios de asesoría legal para asegurar que se cumplen los términos de las licencias libres y evitar posibles pleitos.

**Formación y documentación:** las empresas que aplican este modelo de negocio obtienen ingresos ofreciendo cursos y formación especializada sobre varios productos libres, así como documentación y manuales que complementan aquellos ofrecidos por la comunidad. En el campo de la traducción, la empresa española Qabiria ha lanzado recientemente un libro sobre OmegaT que venden en formato impreso.

**División de los costes de investigación y desarrollo:** este modelo de negocio, que es tal vez el más extendido, se basa en aprovechar el hecho de que según algunos estudios los costes en investigación y desarrollo se pueden reducir hasta en un 36 % cuando se utiliza software libre producido por una comunidad grande y activa. Esta es una oportunidad especialmente interesante para las universidades y centros de investigación.

### **1.1.2 Barreras y riesgos**

Diversos autores han identificado también ciertos aspectos particulares del software libre que pueden dificultar su adopción y que los usuarios potenciales deben considerar a la hora de seleccionar los programas apropiados para sus necesidades. Entre estos aspectos, en el contexto empresarial se destaca el desconocimiento general, que tiene a su vez como consecuencia la falta de recursos humanos capacitados en este tipo de herramientas (Skok, 2011).

Uno de los aspectos a considerar por parte de los usuarios potenciales es que en el software libre, a diferencia del software privativo, normalmente no existe una especificación de requerimientos funcionales de carácter formal y contractual previa al desarrollo (Scacchi, 2009). En otras palabras, a menudo los mismos desarrolladores, que a su vez suelen ser usuarios, van implementando nuevas funciones con base en sus necesidades y conocimientos y en las posibles discusiones en foros y grupos de usuarios sin que exista un control centralizado. De ahí que Scacchi (2009) proponga abordar la elicitación y análisis de requerimientos en el software libre como un proceso continuo de adaptación que tiene lugar después del desarrollo, y no antes, y que por tanto convendría denominar “especificación de capacidades del software” en lugar de la especificación de requerimientos inicial que establece la visión tradicional de la ingeniería de software y requerimientos.

El vasto número de proyectos de software libre es otro aspecto que puede representar una barrera inicial para los usuarios interesados (BRR, 2005). A la fecha (marzo de 2012), el portal SourceForge<sup>13</sup>, la mayor forja de proyectos de software libre en el mundo, cuenta con unos 324.000 proyectos registrados, por lo que para algunas categorías de programas el número de opciones disponibles es bastante elevado. De hecho, en el contexto empresarial, el proceso de búsqueda e identificación de alternativas libres adecuadas puede llegar a representar un gasto bastante importante, incluso mayor que el gasto asociado al proceso de formación del personal en el caso de migraciones a gran escala (Daffara, 2009).

Otro aspecto que los usuarios potenciales de software libre deben considerar es que la mayoría de proyectos no ofrece asistencia técnica de carácter profesional, sino que esta se basa en los aportes de la comunidad de usuarios y desarrolladores (van den Berg, 2005; BRR, 2005; Wheeler, 2008). La manera como se gestiona la asistencia en un proyecto de software libre constituye una de las principales diferencias entre el modelo de desarrollo del software libre y el del software privativo (Wheeler, 2008) y por tanto este es un aspecto importante que se debe tener en cuenta por sus posibles implicaciones.

---

13 <http://sourceforge.net>

Tenemos además la volatilidad característica de los proyectos de software libre, que se refiere al constante lanzamiento de actualizaciones propio del paradigma de desarrollo basado en una comunidad de desarrolladores (BRR, 2005). Este aspecto puede dificultar la implementación de los programas dados los continuos cambios, aunque también puede representar una ventaja en tanto que los problemas reportados o funciones solicitadas en algunos casos se pueden corregir e implementar de manera más rápida (van den Berg, 2005).

Otro aspecto relacionado con el anterior es la longevidad de los proyectos de software libre, es decir, su tiempo de existencia (van den Berg, 2005; BRR, 2005; Wheeler, 2008). Este aspecto depende directamente del interés de la comunidad de usuarios y desarrolladores, al no existir un respaldo comercial en la mayoría de los casos (BRR, 2005). Según Lee, Kim y Gupta (2009), las estadísticas de SourceForge en el año 2005 mostraban que la mayoría de los proyectos libres registrados habían fracasado al no lograr superar las fases de planeación (22 %), prealfa (17 %) o alfa (58 %) o no mostrar signos de actividad de desarrollo. En la misma línea, en un estudio empírico llevado a cabo en 2006 por Schweik y English (2007) sobre los proyectos registrados en SourceForge, de los 110.993 proyectos analizados el 63 % fueron clasificados como abandonados por no haber llegado a lanzar ninguna versión del software (35 %) o llevar un año o más sin lanzar versiones (28 %).

Especialmente en el caso de algunos proyectos jóvenes, otro riesgo a tener en cuenta tiene que ver con la baja calidad del código y la ausencia de procesos de ingeniería y pruebas formales. Dado que cualquiera puede crear un proyecto de software libre, es común que en sus etapas iniciales los programas tengan problemas y los creadores son programadores aficionados pueden no contar con los recursos y experiencia necesarios para lanzar productos completos (BRR, 2005).

#### • *Recapitulación*

En este apartado hemos presentado los aspectos básicos del software libre como alternativa al software privativo. Como vimos, el software libre es un movimiento ideológico que aboga por garantizar las libertades de los usuarios de software, de



modo que las licencias libres nos permiten el uso, estudio, modificación, copia y distribución de los programas. Las licencias privativas, por el contrario, establecen restricciones que el usuario debe aceptar si quiere utilizar el programa. Así pues, este es un asunto que va más allá del factor económico, puesto que no todo el software gratuito es libre y el software libre también puede ser comercial. Vimos también que el software libre se basa en un modelo de desarrollo abierto, transparente y colaborativo en el que el acceso público al código fuente es una precondición y en el que cualquier interesado puede participar en la medida de sus posibilidades.

Después de explicar los conceptos fundamentales, exploramos las oportunidades que ofrece el software libre, así como las principales barreras y riesgos para su implementación. En cuanto a las oportunidades que ofrece, vimos que el software libre promueve la independencia tecnológica al evitar depender de un único proveedor de software y que la oportunidad más obvia es la reducción del coste total de propiedad del software al no tener que pagar por la licencia. Además, el software libre promueve el autoaprendizaje y una actitud crítica frente el software, que es visto como un objeto de creación y no solo de uso. En este sentido, se favorece también el acceso a las tecnologías en la lengua materna de los usuarios, ya que las comunidades locales pueden contribuir a los proyectos con la localización de los programas a sus lenguas. Por otra parte, la posibilidad de reutilizar el código es una oportunidad de especial interés para los programas de I+D de universidades y empresas. Vimos también que el software libre es en general más flexible, ya que su modelo de desarrollo favorece la creación de plugins, extensiones o módulos independientes que luego se integran al sistema principal. Finalmente, el software libre tiende a favorecer la interoperabilidad, pues se da preferencia a los formatos basados en estándares abiertos. Así pues, no es de extrañar que desde el ámbito empresarial se estén explorando diferentes modelos de negocio basados en el software libre.

Pasando a las barreras y riesgos a considerar a la hora de implementar software libre, vimos que el desconocimiento general sobre este tipo de alternativas es tal vez el factor más determinante. Otros factores que hay que tener en cuenta

son la ausencia de asistencia técnica formal en muchos casos, el lanzamiento constante de actualizaciones propio de su paradigma de desarrollo y la posibilidad de que el código sea de baja calidad en el caso de los proyectos creados por aficionados. Finalmente, la longevidad de los proyectos depende en gran medida del interés de la comunidad de usuarios y desarrolladores, ya que en muchos casos no se cuenta con respaldo institucional.

## **1.2 Clasificación de las tecnologías de la traducción**

Only when translators are critically aware of the available tools can they hope to be in control of their work.  
(Biau-Gil & Pym, 2006, p. 19)

En el apartado anterior abordamos las generalidades del software libre y las distintas tipologías de software según sus licencias. En este apartado pasaremos a considerar el panorama general de las tecnologías informáticas desde el punto de vista de la traducción, panorama que nos servirá luego como base para la selección de herramientas para el entorno de trabajo del traductor.

En la era digital en que vivimos el ordenador ha pasado de ser una herramienta sofisticada para uso exclusivo de científicos y estudiosos a ser parte esencial de la cotidianidad de profesionales en todas las áreas, fuente de entretenimiento y puerta de acceso al mundo para todos aquellos que tienen el privilegio de tener acceso a un ordenador personal y una conexión a Internet. La traducción como profesión no ha sido ajena a este cambio ya que los nuevos modos de comunicación como el correo electrónico, la mensajería instantánea, la videoconferencia o la voz por IP, sumados a los nuevos tipos de contenido digital de los archivos multimedia, los hipertextos o el código de páginas web y software, más las nuevas herramientas especializadas a nuestra disposición exigen de parte de los traductores un alfabetismo digital cercano cada vez más al nivel de usuario avanzado en una amplia gama de tecnologías (O'Hagan & Ashworth, 2002).

Así pues, dado el amplio número de herramientas y recursos informáticos que pueden facilitar la labor del traductor en sus diferentes etapas, a la hora de abordar

el tema la primera pregunta que surge es cómo clasificar las tecnologías disponibles. En este sentido, encontramos en la literatura diversidad de propuestas de clasificación según diferentes criterios que Alcina (2008) resume en las siguientes cinco perspectivas: 1) según el grado de automatización del proceso de traducción; 2) según la fase del proceso de traducción en que se utilizan las herramientas; 3) según el nivel de conocimientos informáticos necesarios; 4) según su relación con la traducción; y 5) según la dimensión de la traducción a la que es aplicable la herramienta. También resulta relevante la distinción que establece la misma autora entre herramientas y recursos, donde las primeras se refieren al software y los segundos a conjuntos de datos lingüísticos disponibles en formato electrónico, como diccionarios, corpora o bases de datos terminológicas (Alcina, 2008). En el presente trabajo solo nos ocuparemos de las herramientas.

Según el grado de automatización del proceso de traducción, encontramos por ejemplo la clasificación propuesta por Hutchins y Somers (1992), que distingue entre traducción humana tradicional, traducción humana asistida por ordenador, traducción automática con participación humana y traducción totalmente automática de alta calidad.

Según esta clasificación, en la primera categoría no hay ningún tipo de automatización y por tanto, como bien plantea Austerlühl (2001), no constituye un modelo válido para la traducción en nuestros días; en la segunda categoría encontramos herramientas como correctores ortográficos y de gramática, enciclopedias, tesauros y diccionarios electrónicos y las estaciones de trabajo del traductor; de la tercera categoría hacen parte los sistemas de traducción automática interactivos y la traducción automática con pre y postedición; y la cuarta categoría, que implica la ausencia total de intervención humana y un resultado de alta calidad, aún hoy en día sigue siendo una utopía (Alcina, 2008; Austerlühl, 2001). Así pues, las categorías dos y tres constituyen el área de interés en la actualidad y autores como Austerlühl (2001) las agrupan en una misma categoría.

Otra clasificación que parte de la misma perspectiva es la ofrecida por Melby (1996). En este caso, el autor distingue tres niveles. En el primero se encuentran las

herramientas necesarias para traducir textos recibidos en papel o por fax incluyendo, entre otros, sistemas de reconocimiento de voz, procesadores de texto, faxes, correo electrónico y herramientas de gestión terminológica. En el segundo nivel, por otro lado, se encuentran las herramientas necesarias para procesar textos en formato electrónico, incluyendo memorias de traducción, herramientas de búsqueda terminológica y de control de calidad. Finalmente, el tercer nivel corresponde a la traducción automática.

Además de las dos clasificaciones mencionadas, encontramos otras propuestas más recientes que también parten de la distinción según el grado de automatización y que por tanto se basan en la dicotomía traducción automática – traducción asistida. En su libro *Computer-Aided Translation Technology*, Bowker (2002), por ejemplo, distingue tres tipos básicos de tecnologías utilizadas para la traducción. En la primera categoría, que denomina *traducción humana*, encontramos los procesadores de texto, correctores de ortografía y gramática, los recursos electrónicos e Internet. En la segunda categoría, denominada *traducción asistida por ordenador*, se incluyen las herramientas de captura de datos como los sistemas de reconocimiento óptico de caracteres o de reconocimiento de voz, las herramientas de análisis de corpus, los sistemas de gestión terminológica, las memorias de traducción y las herramientas de localización de software y páginas web. La tercera y última categoría, al igual que en la clasificación de Melby, corresponde a los sistemas de traducción automática.

Partiendo de la misma perspectiva del grado de automatización, Oliver, Moré, y Climent, por su parte, establecen dos categorías básicas en su libro *Traducción y tecnologías* (2008). La primera corresponde a los sistemas de traducción automática, mientras que la segunda, denominada *traducción asistida por ordenador*, se divide a su vez en dos subcategorías: herramientas de memorias de traducción, y otras herramientas de traducción asistida por ordenador, donde se incluyen, entre otros, editores de páginas web, correctores ortográficos y extractores automáticos de terminología.

Una propuesta similar es la de Lagoudaki (2008), quien también basa su clasificación en la dicotomía traducción automática – traducción asistida. Como se puede ver en la figura 1.3, en este caso la autora descompone las tecnologías de la traducción en sistemas de traducción automática, por un lado, y sistemas interactivos de apoyo a la traducción, por el otro. Estos últimos incluyen, según la autora, los sistemas de memorias de traducción, las herramientas de gestión terminológica, las herramientas de concordancias, las herramientas de localización y, por último, los sistemas de gestión de la traducción (TMS, por sus siglas en inglés).

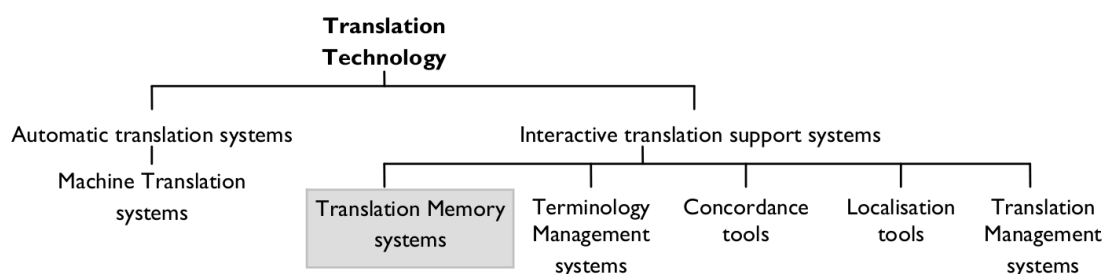


Figura 1.3: Categorización de las tecnologías de la traducción según Lagoudaki (2008).

Otro ejemplo de esta perspectiva de clasificación lo ofrece Hutchins en su Compendium of Translation Software (2008). Este compendio, que pretende ofrecer un listado exhaustivo y detallado de los sistemas de traducción automática y herramientas de traducción asistida por ordenador disponibles en el mercado, incluye para todos los programas listados la categoría a la que pertenecen según las siguientes distinciones.

En primer término encontramos los sistemas de traducción automática, que a su vez se dividen en uso doméstico, para Internet y páginas web, uso profesional, uso empresarial y para lengua hablada. La segunda categoría comprende las herramientas de apoyo a la traducción y se divide en diccionarios electrónicos, herramientas para localización, sistemas de memorias de traducción, herramientas de alineación, sistemas de gestión terminológica, herramientas de pre-edición y estaciones de trabajo del traductor. La última categoría no corresponde en realidad a herramientas sino a servicios de traducción y se divide en portales de traducción automática, que proporcionan acceso o información sobre varios sistemas, y

servicios de traducción automática ofrecidos vía Internet o teléfono móvil que utilizan sistemas de traducción automática y que pueden incluir postedición humana.

Al considerar estas propuestas de clasificación de las herramientas, nos damos cuenta de que al asumir esta perspectiva en última instancia todo parece quedar reducido a dos posibilidades: traducción automática y traducción asistida por ordenador, siendo esta última categoría una especie de comodín en el que encajan el resto de herramientas. En el mejor de los casos, como en las propuestas de Melby (1996) y Bowker (2002), se propone una tercera categoría para la traducción humana más tradicional, si bien los criterios para distinguir esta categoría de la traducción asistida no son del todo claros, ya que para Melby se trata del medio original del documento, mientras que para Bowker la traducción humana implica simplemente el uso de procesadores de texto y recursos electrónicos. Además, de entre estas propuestas solo la de Hutchins y Somers (1992) tiene en cuenta el hecho de que el uso de sistemas de traducción automática no excluye la intervención humana, por lo menos cuando el usuario es un traductor profesional, aunque como resultado de esto la clasificación de estos autores queda reducida a una sola categoría válida, traducción asistida por ordenador, ya que como lo plantea Auster Mühl (2001) los dos extremos del continuo resultan siendo inaplicables, uno por ser cosa del pasado (traducción humana sin automatización) y el otro por utópico (traducción totalmente automática de alta calidad).

Pasamos ahora a las clasificaciones que se basan en la fase del proceso de traducción en que se utilizan las herramientas. En esta línea, Melby (1998) propone una versión más actualizada de sus clasificaciones anteriores en la que distingue tres etapas del proceso (antes, durante y después de la traducción) y dos niveles de aplicación de las herramientas (términos y segmentos). Esta matriz da como resultado seis tipos de tecnologías, a los que se suman otros dos que abarcan todo el proceso y que corresponden a la infraestructura necesaria, que incluye los sistemas de creación y gestión de documentos, bases de datos terminológicas y las herramientas relacionadas con las comunicaciones (intranet/Internet, correo electrónico, clientes FTP, navegadores web, etc.), y a las herramientas de gestión de

proyectos y de contabilidad. En la figura 1.4 se presenta el esquema de la propuesta de Melby (1998).

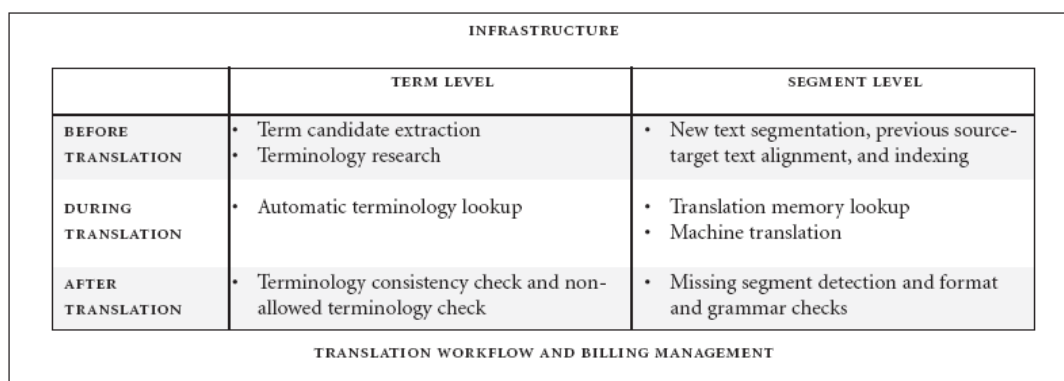


Figura 1.4: Los ocho tipos de tecnologías de la traducción según Melby (1998).

En esta perspectiva, al incluir como niveles solo términos y segmentos, se dejan de lado las actividades que implican el tratamiento de los textos como unidad, como es el caso de la creación y explotación de corpóra o el reconocimiento óptico de caracteres. Además, se incluye la alineación de traducciones previas como una actividad a nivel de segmentos, sin considerar que se parte de textos completos que en el proceso son segmentados. Esta clasificación también ignora la posibilidad de realizar las traducciones sin recurrir a sistemas de memorias de traducción pero haciendo uso de otras herramientas, como ocurre cuando se trabaja sobrescribiendo el original con su traducción en procesadores de texto, herramientas de maquetación o de edición de imágenes.

También en esta línea encontramos la clasificación de Austermühl, quien se basó en el modelo de proceso propuesto por Holmes en 1988 para la evaluación de traducciones literarias. En esta clasificación se distinguen tres grandes fases del proceso: recepción del texto de partida, transferencia y formulación del texto de llegada (ver figura 1.5). Según el autor, en la fase de recepción para entender el texto original el traductor debe recurrir a herramientas que le permitan documentarse en la lengua de partida incluyendo, entre otras, bases de datos terminológicas y de conocimiento, herramientas de análisis de textos y recursos electrónicos como diccionarios y enciclopedias, así como herramientas que faciliten la comunicación con expertos y colegas.

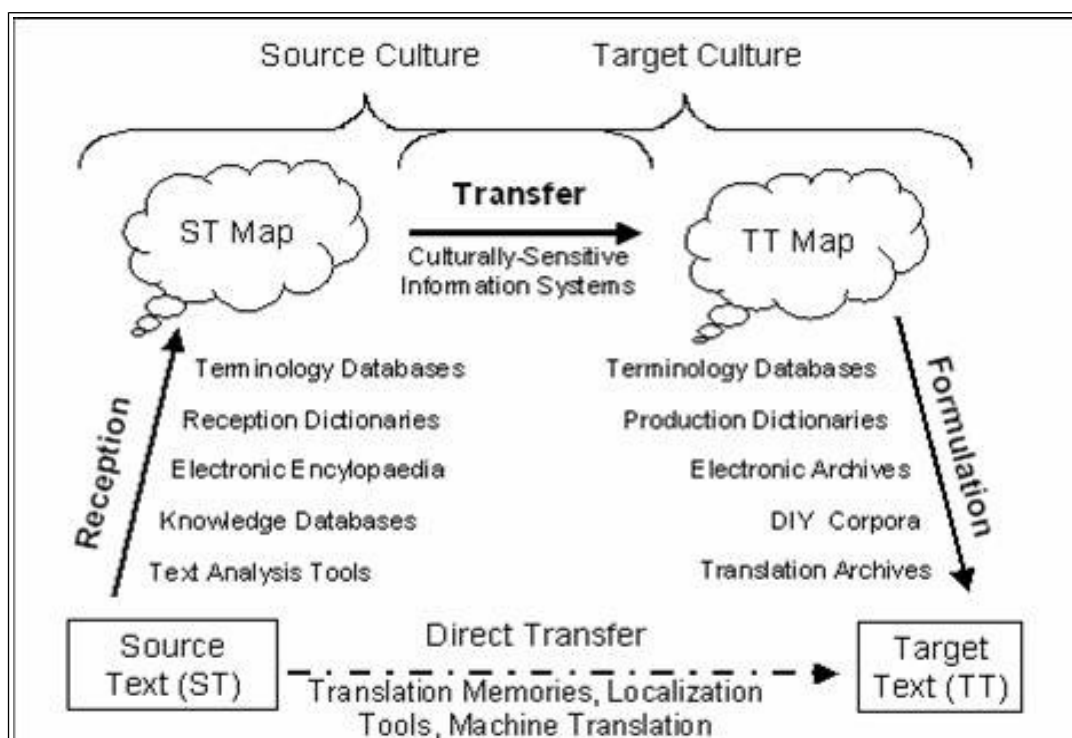


Figura 1.5: Las herramientas y el proceso de traducción según Austermühl (2001).

A diferencia de la fase de recepción que se comparte con los lectores y la de formulación que se comparte con los redactores, la fase de transferencia es exclusiva de la traducción. Curiosamente, según Austermühl, las herramientas necesarias para esta fase serían las más escasas y corresponderían más bien a recursos, incluyendo por ejemplo diccionarios, bases de datos terminológicas y sistemas hipermedia con audio y vídeo específicos para las culturas implicadas en la traducción en cuestión.

En la fase de formulación se recurre a herramientas similares a las de la fase de recepción, pero esta vez en lengua de llegada, así como a sistemas de gestión y almacenamiento de los documentos y bases de datos terminológicas. Pero además, el autor incluye en el diagrama una cuarta posibilidad, denominada *transferencia directa*, que corresponde a los sistemas desarrollados para abarcar prácticamente todo el proceso de traducción en un solo programa, a saber, sistemas de memorias de traducción, herramientas de localización y sistemas de traducción automática.

Esta clasificación, que parte de un modelo teórico del proceso de traducción, podría complementarse con otras herramientas más generales y de infraestructura que también son importantes para aumentar la eficiencia y productividad de los



traductores, como es el caso de los sistemas de reconocimiento óptico de caracteres o de voz, o las herramientas para la gestión financiera y de proyectos. Tampoco resulta clara la diferenciación entre las fases de transferencia y de transferencia directa, primero porque en la primera solo se incluyen recursos idealizados y no herramientas, ya que como el mismo autor reconoce se trataría de recursos diseñados especialmente para proporcionar al traductor información comparativa detallada de las culturas de partida y de llegada, con lo que serían demasiado específicos y demandarían demasiado esfuerzo para su desarrollo considerando que variarían para cada par de culturas involucradas; y segundo porque si las herramientas de la fase de transferencia directa abarcan prácticamente todo el proceso, dada la tendencia actual del mercado hacia herramientas todo en uno, al final solo tendríamos una categoría.

Fulford y Granell Zafra (2005), por su parte, llevaron a cabo un estudio empírico sobre la implantación de las tecnologías de la información y comunicación (TIC) por parte de los traductores independientes en el Reino Unido. Para clasificar las tecnologías partieron del modelo de Austerlühl (2001), pero adaptándolo específicamente a las actividades de los traductores independientes teniendo en cuenta en el proceso aspectos como el mercadeo y la búsqueda de clientes, así como la gestión administrativa y financiera, que aunque no hacen parte del proceso de transferencia cultural y lingüística como tal sí constituyen elementos importantes de la traducción como actividad empresarial. El resultado de su clasificación, que se presenta en la figura 1.6, comprende 6 categorías: creación de documentación, búsqueda y recuperación de información, creación de traducción, comunicación, mercadeo y consecución de trabajos y administración del negocio.

A esta clasificación haría falta agregarle actividades como la alineación de traducciones previas, la digitalización de textos en papel mediante software de reconocimiento óptico de caracteres o la gestión y actualización de memorias de traducción. También se podría argumentar que las actividades de mercadeo y consecución de trabajos hacen parte de la administración del negocio.

<b>Activity</b>	<b>ICT Support</b>
<b>Document production</b> <i>e.g. creating and formatting target texts; overtyping sources texts with target texts</i>	Word processing software (e.g. MS Word, Wordperfect); Graphical / presentation software (e.g. MS PowerPoint); Web publishing software (e.g. MS FrontPage, Dreamweaver); Desktop publishing software (e.g. QuarkXpress, PageMaker)
<b>Information search &amp; retrieval</b> <i>e.g. locating background and reference materials; locating client company information; identifying terminology; locating definitions of terms; finding examples of terminology usage; managing personal terminology collections</i>	Internet search engine (e.g. Google, Altavista); Electronic encyclopaedia / reference work (e.g. Encyclopaedia Britannica, Encarta); Terminology databank (e.g. EuroDicAutom, CILF); Text corpus / document archive (e.g. British National Corpus, New Scientist Archive); Electronic library (e.g. The British Library, Biblioteca Nacional de España); Electronic dictionary and / or glossary (e.g. yourDictionary.com, Lexicool); Database software (e.g. MS Access, FileMaker); Terminology management software (e.g. MultiTerm, Lingo)
<b>Translation creation</b> <i>formulating translation</i>	Translation memory (e.g. Trados, Déjà Vu, SDLX, Transit); Machine translation (e.g. Reverso Pro, Systran)
<b>Communication</b> <i>e.g. liaising with clients; networking with colleagues</i>	Electronic mail (e.g. Webmail, MS Outlook, Thunderbird); Electronic mailing lists (e.g. LANTRA-L, The LINGUIST List); Online discussion groups (e.g. Proz.com, TranslatorsCafe.com)
<b>Marketing &amp; work procurement</b> <i>e.g. promoting translation services; searching for clients; bidding for translation contracts</i>	Having own web site; Online marketplaces (e.g. Foreignword.biz, Proz.com)
<b>Business management</b> <i>e.g. client &amp; contact data management; contract quotations; billing / invoicing; financial management</i>	Database software (e.g. MS Access, FileMaker) Spreadsheet software (e.g. MS Excel, Lotus 1-2-3); Accounting / bookkeeping package (e.g. Sage, QuickBooks)

Figura 1.6: Actividades de los traductores independientes y TIC según Fulford y Granell Zafra (2005).

La siguiente perspectiva de clasificación toma como punto de partida el nivel de conocimientos informáticos requeridos y corresponde a la propuesta del proyecto LETRAC (Language Engineering for Translators Curricula) (Badia et al., 1999). Este proyecto tenía como objetivo facilitar la unificación de los programas

universitarios en traducción en Europa en lo concerniente a los elementos tecnológicos que se deberían incluir en el currículo (Alcina, 2008). Así pues, como resultado del proyecto se definieron tres módulos divididos a su vez en unidades, algunas de ellas obligatorias y otras opcionales, con lo que se pretendía abarcar todos los conocimientos y herramientas de la informática aplicada a la traducción.

El Módulo A, denominado *Introducción a las ciencias de la computación*, está compuesto de una unidad de carácter obligatorio que se considera prerrequisito para los demás módulos. Esta unidad abarca los aspectos generales de la entrada, procesamiento, almacenamiento y salida de datos desde el punto de vista del hardware y el software involucrado en el proceso (Badia et al., 1999). El Módulo B, *Tecnologías de la información y maquetación para traductores*, está dividido en tres unidades, dos de ellas obligatorias. La primera unidad comprende las herramientas de maquetación, manipulación de imágenes y presentación de diapositivas. La segunda unidad abarca los principios generales de las telecomunicaciones e Internet, incluyendo aspectos como protocolos de red, estructura cliente-servidor, correo electrónico, FTP, telnet, navegadores y editores de HTML, así como la gran variedad de recursos en línea disponibles para los traductores. La tercera unidad del módulo B, de carácter opcional, permite profundizar en los aspectos tratados en las otras dos unidades del módulo y además puede servir como introducción avanzada a temas como lenguajes de programación o sistemas operativos.

El Módulo C, denominado *Ingeniería lingüística*, constituye el componente más importante del currículo y está organizado en 6 unidades. La unidad C1, de carácter obligatorio, comprende herramientas y aspectos generales como procesadores de texto y correctores, lenguaje controlado, gestión de proyectos y documentos mediante bases de datos, hojas de cálculo y administración de archivos, y sistemas de gestión de terminología. La unidad C2, también obligatoria, abarca los sistemas de memorias de traducción y traducción automática, con especial atención a la integración de estas dos tecnologías. Las cuatro unidades restantes, de carácter opcional y con un importante componente teórico, se ocupan de la formalización lingüística, la traducción automática, la lingüística de corpus y análisis básico, y de estudios de caso en las áreas de gestión terminológica y de

proyectos, lenguaje controlado en entornos industriales y proyectos de localización (Badia et al., 1999).

Así pues, en la organización de estos módulos es posible apreciar una clasificación de las herramientas que tiene como base cuatro grandes categorías: software general, software de creación de documentación, software para comunicaciones y trabajo en red, y software para el procesamiento de datos lingüísticos, que se divide a su vez en herramientas generales, software específico para traducción y software para análisis de corpus. En esta clasificación, sin embargo, no está muy claro el lugar que ocupan las herramientas necesarias para la dimensión de la traducción como negocio, ya que las herramientas para gestión de proyectos solo se mencionan brevemente en la misma categoría que las herramientas generales de procesamiento de datos lingüísticos y las herramientas de gestión financiera no se incluyen.

Otra perspectiva de clasificación que considera la relación de las tecnologías con la traducción es la de Vilarnau (2001, en Alcina, 2008), quien distingue 5 categorías de herramientas. En la primera categoría, la más relacionada con la traducción y por tanto denominada *programas de traducción*, encontramos entre otros los procesadores de texto, herramientas de traducción asistida, sistemas de traducción automática, software de maquetación, editores HTML y herramientas de localización. Un segundo grupo, denominado *software de ayuda a la traducción*, incluye recursos en CD o DVD, navegadores web, hojas de cálculo y correctores de gramática y ortografía. La tercera categoría comprende lo necesario para enviar y recibir documentos, por ejemplo, correo electrónico, FTP, reconocimiento óptico de caracteres y reconocimiento de voz. La cuarta categoría, que el autor denomina *software accesorio para la traducción*, abarca programas de compresión, chat, motores de búsqueda, software de gestión de descargas, entre otros, mientras que la última categoría, la menos relacionada con la traducción, incluye programas generales como antivirus, cortafuegos, software para el mantenimiento del sistema operativo, etc. (Alcina, 2008).

Esta propuesta tiene la ventaja de que toma como punto de partida las necesidades del traductor profesional y por lo tanto pretende incluir todas las herramientas que puedan servir para facilitar su labor; sin embargo, los nombres asignados a las categorías, así como los criterios utilizados para decidir qué herramientas se incluyen en cada grupo, no son muy claros (Alcina, 2008). Llama la atención, por ejemplo, que los procesadores de texto, herramientas de maquetación y editores HTML pertenezcan a la categoría de programas de traducción, o que los compresores y gestores de descargas pertenezcan a la categoría de programas accesorios a la traducción y no a la de programas generales; más aún, con los ejemplos incluidos es difícil entender el porqué estas dos últimas deben ser categorías separadas.

Por otro lado, en la edición 123 de su boletín para profesionales de la traducción, Zetzsche (2008) clasifica las herramientas desarrolladas específicamente para traductores en tres grandes grupos. En el primer grupo encontramos las herramientas que proporcionan de manera independiente funciones específicas para el traductor. Las funciones comprendidas en este grupo incluyen la búsqueda en recursos lingüísticos, la gestión terminológica, la gestión de proyectos, el recuento de palabras y la localización de software.

En el segundo grupo encontramos las herramientas que proporcionan funciones que sirven de complemento a los sistemas de memorias de traducción, como las herramientas de extracción terminológica, los filtros de conversión para formatos específicos, las herramientas de alineación y las herramientas de control de calidad. El tercer y último grupo corresponde a los sistemas de memorias de traducción propiamente dichos, es decir, aquellas herramientas que proporcionan un entorno integral que abarca una gran variedad de funciones relacionadas con la traducción.

De esta propuesta resulta útil su identificación de módulos funcionales relevantes para los traductores, aunque el criterio del autor de considerar solo herramientas desarrolladas especialmente para traductores no es del todo claro. Las herramientas de gestión de proyectos, por ejemplo, no responden necesariamente a

dicho criterio y lo mismo podría decirse de las herramientas para búsquedas en recursos lingüísticos y extracción terminológica. Además, desde esta perspectiva se dejan sin clasificar otras herramientas útiles para los traductores de ciertas especialidades, como las herramientas para edición y traducción de subtítulos.

Neunzig (2001), por su parte, sugiere que las herramientas se pueden aplicar a la traducción en tres dimensiones diferentes: enseñanza de la traducción, traducción profesional, e investigación en traducción. Como bien señala Alcina (2008), esta perspectiva permite distinguir tres perfiles de usuarios de las herramientas pero no clasificarlas, ya que lo que varía es el propósito para el que se utilizan y no las herramientas utilizadas, aunque se debe tener en cuenta que la dimensión de investigación en traducción sí puede abarcar un abanico más variado de herramientas que no son necesarias para el traductor profesional o para la enseñanza de la traducción incluyendo, por ejemplo, las herramientas de registro de actividades (necesarias cuando se utilizan protocolos de pensar en voz alta), herramientas para la creación de cuestionarios en línea o paquetes estadísticos avanzados.

Hemos considerado hasta este punto las cinco perspectivas de clasificación de las tecnologías de la traducción identificadas por Alcina (2008) en su revisión de la literatura; pasaremos ahora a considerar la clasificación que propone la misma autora con base en dicha revisión y su experiencia en el campo de la investigación y enseñanza en esta área. Como punto de partida, Alcina describe las tecnologías de la traducción como

el área de estudio que se ocupa del diseño y adaptación de estrategias, herramientas y recursos tecnológicos que sirven para hacer más fácil la labor del traductor, así como para facilitar la investigación y enseñanza de estas actividades (2008, p. 90, traducción propia).

Según esta descripción, entonces, como criterio básico para determinar qué componentes se incluyen en este campo se deben tener en cuenta las actividades y el proceso de trabajo del traductor partiendo de una visión actualizada de las herramientas disponibles. Como criterios de clasificación de las herramientas, la autora sugiere que se deben considerar las convenciones de clasificación utilizadas

en otros campos partiendo de las similitudes entre herramientas con base en aspectos como los tipos de datos que manipulan, el tipo de procesamiento que realizan, sus funciones y el hecho de que manipulen o no información lingüística. De igual manera, la clasificación debe ser flexible y expansible para que se pueda adaptar a los nuevos desarrollos en el área y debe permitir su adaptación al currículo de los programas en traducción.

Con estas consideraciones en mente, Alcina (2008) establece una clasificación con cinco grupos. En el primero de ellos se ubican los elementos más generales incluyendo, componentes físicos y periféricos, sistemas operativos, programas de mantenimiento, tipos de medios extraíbles y utilidades generales. En el segundo grupo, denominado *herramientas de comunicación y documentación*, encontramos los componentes que permiten la comunicación con clientes, colegas y expertos (correo electrónico, chat, listas de correo, entornos colaborativos de trabajo, etc.), así como la utilización de redes, incluyendo Internet, para obtener información especializada. El tercer grupo comprende las herramientas para la creación de textos en diferentes formatos (procesadores de texto, programas de maquetación, edición de HTML/XML), así como las herramientas para su revisión y corrección y los programas de reconocimiento óptico de caracteres y conversión o validación de formatos (ej. PDF a TXT, validación de HTML/XML). En un cuarto grupo se incluyen las herramientas y recursos lingüísticos, que incluyen diccionarios, bases de datos y córpora en formato electrónico y las herramientas para su creación y gestión. El quinto grupo, finalmente, comprende las herramientas de traducción, que son las que permiten trabajar a la vez con el texto de partida y el texto de llegada, sea en segmentos o como textos completos.

Luego de haber considerado las propuestas de los diferentes autores, para el presente trabajo se tomó como base la clasificación de Alcina (2008) con una categoría adicional para recoger las herramientas para las tareas de gestión de proyectos y finanzas, actividades que también constituyen parte importante del trabajo del traductor y que otros autores incluyen en sus clasificaciones (cf. Fulford & Granell Zafra, 2005; Melby, 1998; Zetzsche, 2008).

### **1.3 Las necesidades de los traductores**

En el apartado anterior consideramos las diferentes propuestas de clasificación de las tecnologías aplicadas a la traducción. En este apartado veremos los diferentes escenarios de uso de las tecnologías de la traducción (1.3.1) y describiremos de manera general las principales características de los sistemas de memorias de traducción disponibles en la actualidad (1.3.2), por ser este el tipo de herramienta que vamos a evaluar en el presente trabajo.

#### **1.3.1 Escenarios de uso**

Según Höge (2002), “las necesidades de los traductores (...) pueden variar considerablemente en diferentes situaciones” (p. 90, traducción propia) y dichas necesidades dependerán en gran medida del contexto profesional en el que se lleve a cabo la traducción en relación con tres factores: a) el número de personas que participan en el proyecto; b) los recursos compartidos disponibles; y c) la manera en que se asignan las tareas. Según la misma autora, se distinguen entonces dos grandes contextos profesionales: los traductores independientes, por un lado, y la traducción en el contexto de medianas y grandes empresas, por el otro.

Si bien esta división en dos grandes grupos es un buen punto de partida, es posible ser aún más específico. En el caso de los traductores independientes, en primer término, es necesario tener en cuenta que gracias a las tecnologías de la información y la comunicación (TIC) cada vez es más común que el trabajo se organice en “equipos globales descentralizados” (Rico, 2001, p. 4), con lo que la figura del traductor completamente aislado cada vez va quedando más relegada al pasado. En el contexto de las medianas y grandes empresas, por otro lado, es posible distinguir entre agencias de traducción, empresas con departamento de traducción interno, organizaciones bilingües, organizaciones internacionales e instituciones públicas (EAGLES, 1996). Estos diferentes contextos profesionales implican diferentes tipos de relaciones cliente-traductor que a su vez repercuten en aspectos como el uso de distintos medios de comunicación, los plazos de entrega, el



volumen de textos a traducir, la calidad requerida y las tarifas aplicables (Rico, 2001).

En relación con las tecnologías, una diferencia importante entre los dos grandes grupos distinguidos por Höge (2002) es que en el caso de los traductores independientes y pequeños equipos de traductores son ellos mismos quienes suelen estar a cargo de seleccionar e instalar los programas, resolver los posibles problemas técnicos y llevar a cabo el mantenimiento general de los ordenadores. En las empresas y organizaciones, en cambio, normalmente existe un técnico o un departamento informático especializado que se encarga de realizar dichas actividades.

En cuanto a las tareas asignadas, es necesario considerar que las necesidades del usuario pueden variar según el papel que desempeñe en un proyecto dado (Taravella, 2011). Entre las principales tareas que se llevan a cabo en un proyecto de traducción podemos distinguir la gestión del proyecto, la preparación de presupuestos, la gestión terminológica y de recursos lingüísticos, la traducción, la postedición, la maquetación y la revisión lingüística y funcional, entre otras. En el caso de los traductores independientes, normalmente es el mismo traductor quien asume los diferentes papeles según las características de cada proyecto.

### **1.3.2 Los sistemas de memorias de traducción**

En el apartado 1.2 abordamos el tema de la clasificación de las tecnologías aplicadas a la traducción en un sentido general. Ya que en este trabajo nos limitaremos a evaluar sistemas de memorias de traducción (que forman parte de la categoría de *herramientas de traducción*), nos concentraremos ahora en este tipo de programas. En la actualidad, las alternativas privativas más conocidas son SDL Trados Studio, WordFast, Déjà Vu, SDLX, STAR Transit y memoQ, aunque la oferta es cada día más amplia y difícil de abarcar<sup>14</sup>.

---

<sup>14</sup> Lagoudaki (2006) ofrece un panorama detallado sobre la penetración de los sistemas de memorias de traducción en nuestra industria y los sistemas más utilizados en diferentes contextos según los resultados de una encuesta realizada a 874 traductores en todo el mundo. Hutchins (2008), por su parte, proporciona un catálogo bastante completo de sistemas de memorias de traducción y otras herramientas para el traductor, si bien la versión actualizada de dicho catálogo solo está disponible para los miembros de la Asociación Europea para la Traducción Automática

Como principal herramienta de trabajo de los traductores, los sistemas de memorias de traducción han recibido considerable atención en los últimos años. Muchas de las discusiones se centran en analizar las ventajas y desventajas de su utilización y su impacto en el proceso de traducción como tal. Entre las ventajas más destacadas encontramos el aumento de la productividad, una mejor coherencia y consistencia terminológica y de estilo y una mayor flexibilidad tanto en términos de los formatos de archivo que el traductor puede procesar como en la coordinación de proyectos complejos (Azzano, 2011).

Naturalmente, no todo son ventajas y también se ha dedicado bastante atención a las desventajas y posibles problemas asociados al uso de este tipo de herramientas. Algunos de los inconvenientes mencionados por varios autores son el coste de las licencias, la curva de aprendizaje, la incompatibilidad entre sistemas, la fragmentación de los textos, la presión del mercado hacia la reducción de tarifas, el tiempo necesario para el mantenimiento de las bases de datos y los derechos de propiedad intelectual sobre las mismas (Azzano, 2011; Bowker, 2002).

En cuanto a su diseño, si bien existen diferencias fundamentales entre los distintos sistemas, todos tienen como propósito final la reutilización de traducciones basándose en el concepto de *memoria de traducción*, es decir, el almacenamiento e indexación de traducciones previas para su posterior recuperación (Lagoudaki, 2006; Zerfaß, 2002). En este punto es pertinente señalar que el término *memoria de traducción* se utiliza para referirse tanto al software (sistema de memorias de traducción), como a la base de datos de traducciones previas (Somers, 2003). Sin embargo, no todos los sistemas utilizan una base de datos para almacenar las traducciones. Transit, por ejemplo, genera una especie de corpus temporal con los originales y traducciones indicados por el usuario para un proyecto dado (Lagoudaki, 2006).

Existen además dos enfoques diferentes para el almacenamiento e indexación de las traducciones. El enfoque más común es el basado en oraciones, en el que los textos se dividen en segmentos y cada uno se almacena con su traducción correspondiente; el otro enfoque, más reciente, consiste en almacenar pares de

---

(EAMT, por sus siglas en inglés).

textos completos, lo que permite buscar y recuperar cadenas de caracteres de cualquier longitud (Gow, 2003). Este último enfoque, conocido como CSB (*character-string-within-a-bitext*), es el utilizado por herramientas como MultiTrans, TransSearch y LogiTerm.

A pesar de las diferencias, como bien afirma Höge (2002), en términos generales las herramientas disponibles en la actualidad ofrecen funciones similares. Así pues, un sistema de memorias de traducción incluye tres componentes principales: los filtros de conversión de formatos, el motor de coincidencias y el editor (Azzano, 2011). En los apartados a continuación veremos en más detalle cada uno de estos aspectos, para cerrar luego con una descripción del proceso de traducción en un sistema de memorias de traducción. La descripción que se presenta se basa principalmente en algunas descripciones funcionales de sistemas privativos, así como en la experiencia propia con este tipo de herramientas, y pretende ser genérica, de modo que nos concentraremos sobre todo en las similitudes y no en las diferencias entre sistemas.

### ***1.3.2.1 Filtros de conversión***

Para procesar los archivos originales cada sistema incluye una serie de filtros predefinidos que extraen los elementos textuales de los archivos permitiéndonos trabajar en una misma interfaz con archivos en diferentes formatos (Lagoudaki, 2009). Para representar el formato en el texto a traducir se utilizan unas etiquetas o códigos internos cuyo nivel de protección normalmente se puede configurar, de modo que el traductor puede hacer los ajustes necesarios sin tener que preocuparse de alterar el formato del documento.

Los sistemas actuales normalmente permiten traducir archivos en formatos de uno o varios de los siguientes cinco grupos principales: formatos de texto y ofimática, formatos de maquetación, formatos multimedia, formatos de localización de software y formatos de localización web (ver tabla 1.2). Según el estudio sobre el uso de sistemas de memorias de traducción llevado a cabo por Lagoudaki (2006), en el que se realizó una encuesta a 874 profesionales de la traducción en 54 países sobre las herramientas que utilizan y su grado de satisfacción con las mismas, los formatos que los traductores encuestados reciben con más frecuencia son los de la

suite ofimática de Microsoft y los archivos de texto (96 %), seguidos por el formato PDF (43 %), los formatos XML/HTML (26 %) y los formatos de maquetación (9 % los del programa FrameMaker y 3 % los demás).

Los filtros de conversión pueden además incluir algunas opciones configurables según el tipo de formato. Para los archivos en formato HTML o XML, por ejemplo, la manera de procesar los distintos atributos según nos interese traducirlos o no puede ser una opción configurable por el usuario. Además, en algunos sistemas también es posible crear filtros nuevos desde cero o a partir de los ya existentes, aunque esta es una opción que por lo general solo pueden aprovechar los usuarios más avanzados.

Al terminar la traducción, los mismos filtros utilizados para la importación inicial de los archivos de partida nos permiten realizar el proceso inverso, es decir, la generación de los archivos finales insertando los textos traducidos en lugar de los originales conservando el formato.

Tipo de formato	Ejemplos
Formatos de texto y ofimática	TXT, CSV, TAB, DOC, DOT, RTF, XLS, XLT, PPT, PPS, DOCX, DOTX, XLSX, XLTX, XLSM, PPTX, PPSX, POTX, ODT, ODS, ODP, SRT.
Formatos de maquetación	MIF (FrameMaker), XML (FrameMaker), INX (InDesign), IDML (InDesign), TXT etiquetado (Pagemaker, Ventura), QSC (QuarkXPress), XTG (QuarkXPress), TTG (QuarkXPress), TAG (QuarkXPress), IASCI (Interleaf/QuickSliver), PDF (Acrobat Reader).
Formatos multimedia	PSD (Photoshop), SVG (Photoshop, Illustrator, CorelDraw, genérico), DXF (AutoCAD), TXT (AutoCAD).
Formatos de localización web	HTML, XML, ASP, PHP, JSP, INC, NET, RESX, PPSM, XAML, SGM.
Formatos de localización de software	RC, DLG, EXE, DLL, MO, PO(T), Java Resource Bundles, XML (Android resource), XIB (iOS App resource), TS (Qt Linguist), QPH (Qt Phrase Book), DTD (Mozilla)

Tabla 1.2: Tipos de formatos que se suelen procesar en los sistemas de memorias de traducción.

Además de los archivos originales, los sistemas de memorias de traducción también deben permitir importar y exportar las memorias de traducción y glosarios o bases de datos terminológicas que se utilizan como referencia durante la traducción. En la actualidad, para permitir el intercambio de memorias con otras herramientas, además de su formato nativo, la mayoría de sistemas implementa el

estándar TMX (Zerfaß, 2002; Zetzsche, 2011). Algunas herramientas ofrecen como alternativa para exportar las memorias un formato abierto basado en XML (por ejemplo, TTX en el caso de SDL Trados o TXML en el caso de WordFast Pro).

En cuanto a los glosarios, entre los formatos más comunes encontramos los archivos de texto delimitado por comas (CSV) o tabulaciones (TAB, TXT), las hojas de cálculo y el estándar TBX, que tiene la ventaja de permitir intercambiar información entre bases de datos con modelos de datos diferentes (Mašlanko, 2004).

### ***1.3.2.2 Coincidencias***

Como se mencionó antes, los sistemas de memorias de traducción se basan en la reutilización de traducciones previas. Así, cuando se procesa un nuevo segmento el sistema lo compara con los segmentos almacenados en la base de datos en busca de coincidencias y en caso de encontrarlas se determina entonces el grado de similitud entre el segmento nuevo y el preexistente. Dicho grado de similitud se expresa normalmente como porcentaje y permite distinguir entre coincidencias exactas, totales y parciales (en inglés *fuzzy*) (Austermühl, 2001; Bowker, 2002).

Se habla de **coincidencias exactas**, o del 100%, cuando el segmento nuevo y el segmento recuperado coinciden carácter por carácter, mientras que cuando se presentan diferencias solo en cuanto a la puntuación, las etiquetas de formato u otros elementos transferibles (como números, fechas, etc.) se habla de **coincidencias totales** (Bowker, 2002). Por otro lado, cuando solo una parte del segmento nuevo coincide con el segmento almacenado se habla de **coincidencias parciales**.

En los últimos años, algunos sistemas han empezado a ofrecer otro tipo de coincidencias, conocidas como coincidencias contextuales y coincidencias subsegmentarias. Las **coincidencias contextuales** son coincidencias exactas que además ocurren en el mismo contexto; en otras palabras, en este caso coincide tanto el segmento que se va a traducir como el segmento anterior y el posterior, por lo que se les suele asignar un porcentaje de similitud del 101% (Azzano, 2011). En el caso de las **coincidencias subsegmentarias**, por otro lado, la comparación con los

textos almacenados en la base de datos no se realiza con base en los segmentos completos sino en fragmentos o trozos de los mismos, de modo que reutilizando fragmentos provenientes de diferentes segmentos es posible ensamblar la traducción para el segmento nuevo (Bowker, 2002).

### ***1.3.2.3 El editor***

Una diferencia importante entre los sistemas de memorias de traducción existentes tiene que ver con la interfaz de edición en la que se basan. Algunas herramientas, como la versión tradicional de WordFast o su equivalente libre Anaphraseus, se integran como *plugins* de un procesador de palabras, mientras que otras proporcionan un editor independiente que por lo general presenta el texto y su traducción en formato de tabla (Lagoudaki, 2009). En este último caso, la interfaz incluye además otros paneles para mostrar las coincidencias de memorias y glosarios y dichos paneles por lo general se pueden redimensionar y reubicar según las preferencias del usuario. En el caso de las coincidencias de las memorias, por lo general los sistemas indican las diferencias entre el original almacenado en la memoria y el original del segmento activo mediante distintos colores para facilitar su visualización.

Muchos sistemas de memorias de traducción incluyen en el editor las funciones de **corrección ortográfica, gramatical o de estilo** habituales en los procesadores de texto. En algunos casos, el usuario puede incluso personalizar los diccionarios del corrector ortográfico o las reglas de estilo. Otros sistemas incluyen también **controles de calidad** más especializados y configurables que permiten verificar que se hayan utilizado correctamente las etiquetas de formato, la puntuación, los espacios al inicio y al final de los segmentos, los números o los términos del glosario, entre otros. Para este tipo de revisiones existen dos modos posibles: en tiempo real, es decir, mientras escribimos, o bajo demanda, iniciando la revisión mediante una opción de menú o un atajo de teclado.

También resultan muy útiles las funciones de búsqueda integradas en los editores de la mayoría de los sistemas de memorias de traducción, que normalmente permiten la utilización de comodines o expresiones regulares para realizar **búsquedas avanzadas**. De este modo, es posible recuperar **concordancias** para

una palabra o expresión en los archivos originales y en algunos casos también en archivos de referencia (archivos PDF, otras memorias de traducción, etc.), así como llevar a cabo **búsquedas y reemplazos globales** en los archivos de trabajo. Algunos sistemas también permiten realizar **búsquedas en recursos lingüísticos externos**, como diccionarios locales o en la web, desde la misma interfaz del editor.

Por último, en algunos sistemas es posible utilizar **filtros** (por estatus o por fecha de modificación, por ejemplo) para facilitar la navegación entre segmentos en el editor.

#### ***1.3.2.4 El proceso de traducción en un sistema de memorias de traducción***

Para empezar a trabajar en un sistema de memorias de traducción normalmente es necesario crear primero un proyecto de traducción al que luego agregaremos los recursos de apoyo disponibles (que pueden ser una o varias memorias de traducción y uno o varios glosarios terminológicos, según el sistema) y los documentos de partida.

Después de importar los archivos originales se lleva a cabo un análisis de los mismos. Dicho análisis implica normalmente la **tokenización** y **segmentación** del texto, donde la primera se refiere a la división del texto en *tokens* (típicamente palabras) y la segunda a la delimitación en unidades de traducción, como se explicará más adelante. Esto se puede complementar con la **lematización**, es decir, la reducción de las palabras a sus raíces, o algún tipo de análisis sintáctico, procesos que permiten mejorar el reconocimiento de términos compuestos y fraseología, así como las coincidencias de glosarios y memorias.

Para la segmentación de los textos se utilizan una serie de reglas formuladas mediante expresiones regulares o patrones que permiten dividir el texto en unidades, que según la especificación del estándar TMX pueden ser párrafos, oraciones, frases o bloques. Así pues, se entiende por *unidad de traducción* un segmento en lengua de partida alineado con el segmento correspondiente en la lengua de llegada (Bowker & Barlow, 2004). Para almacenar e intercambiar las reglas de segmentación, que normalmente pueden ser adaptadas por el usuario, se

desarrolló el estándar SRX, que lamentablemente todavía no es tan ampliamente utilizado como el TMX (Zetzsche, 2011).

Este análisis inicial permite determinar el **recuento de palabras** a traducir, así como las **estadísticas de repeticiones y coincidencias** encontradas en las memorias existentes, si es el caso. Estas estadísticas del proyecto luego se actualizarán automáticamente a medida que se avanza en el proyecto y nos permiten saber en todo momento el número de palabras y segmentos traducidos y pendientes de traducir.

Luego de la segmentación y análisis, y antes de empezar el proceso de traducción como tal, es posible realizar una **pretraducción** de los textos a partir de recursos preexistentes. Esta función se basa en la búsqueda de coincidencias en diferentes fuentes para insertarlas como texto de llegada en aquellos segmentos para los que se encuentren resultados. Las fuentes pueden ser glosarios, memorias de traducción o motores de traducción automática (locales o remotos). En algunos casos, el usuario puede especificar las fuentes a utilizar entre aquellas disponibles o establecer prioridades. Posteriormente, durante la traducción propiamente dicha, el traductor deberá revisar los segmentos pretraducidos y traducir los segmentos que no se pretradujeron.

Otra característica que ofrecen algunos sistemas de memorias de traducción es que para la importación y análisis permiten procesar varios archivos a la vez, lo que se conoce como **procesamiento por lotes**. Después del análisis y eventual pretraducción de uno o varios archivos, algunas herramientas permiten también generar un archivo en el formato de intercambio XLIFF para traducirlo en otra herramienta. Dicho archivo incluirá todos los segmentos fuente, con o sin traducción, y toda la información necesaria para recrear los archivos traducidos en su formato original.

En cuanto al proceso de traducción propiamente dicho, denominado por algunos autores como **traducción interactiva**, los sistemas de memorias de traducción presentan el texto original segmento por segmento y el usuario debe ingresar la traducción en el campo correspondiente. En algunos sistemas, como se



mencionó antes, se utiliza el formato de tabla, de modo que los segmentos originales se muestran en una columna y las traducciones en otra. En otros casos, la traducción se ingresa en un campo ubicado debajo del segmento original y es posible configurar si se quieren visualizar todos los segmentos originales o solo el original del segmento activo.

Al abrir o activar un segmento para edición, el sistema lleva a cabo la búsqueda de segmentos similares en los recursos disponibles para el proyecto. En algunos casos, además de los recursos locales, los sistemas permiten también el uso simultáneo de recursos (memorias y glosarios) compartidos a través de la red. Al igual que para la pretraducción, en el modo interactivo algunos sistemas también permiten obtener una primera versión del segmento activo a partir de un motor de traducción automática local o disponible como servicio web, sugerencia que se presenta por lo general en un panel independiente del editor.

Por lo general, tanto para las coincidencias de las memorias como para las sugerencias provenientes de un motor de traducción automática, el usuario tiene la opción de configurar el sistema para que realice la **inserción automática** al pasar a un segmento sin traducir. En el caso de las memorias, también es posible indicar un **porcentaje mínimo de coincidencia**, de modo que solo se presentarán o insertarán las coincidencias parciales cuyo grado de similitud sea igual o mayor al porcentaje señalado. Si el usuario prefiere trabajar a partir del segmento vacío, normalmente se tiene la opción de utilizar atajos de teclado para seleccionar entre las sugerencias disponibles e insertar las correspondencias deseadas.

En el caso de las equivalencias de los glosarios y de las etiquetas de formato también es común contar con **atajos de teclado** que permiten insertarlas fácilmente en el texto de llegada mientras escribimos. Otros elementos que normalmente se pueden configurar para facilitar su inserción mediante atajos de teclado al traducir son los elementos **colocables y localizables** (en inglés *placeables and localizables*<sup>15</sup>). Estos comprenden elementos como fechas, números o medidas que

---

15 Azzano (2011, p. 38-42) presenta una discusión sobre la variedad de términos utilizados en los sistemas de memorias de traducción y por diferentes autores para referirse a este tipo de elementos: colocables, transferibles, intraducibles, constantes, expresiones paralingüísticas, elementos extralingüísticos, entre otros. Como señala el autor, algunos elementos pueden ser

siguen diferentes convenciones según el idioma (por ejemplo, el separador de miles en el caso de los números) u otros que se transfieren tal cual de una lengua a otra, como nombres propios, direcciones de correo electrónico o elementos de una interfaz gráfica, por ejemplo (Azzano, 2011).

Una vez finalizada la traducción de un segmento, según la configuración establecida pueden realizarse varias acciones al avanzar al segmento siguiente, como el cambio de estatus del segmento (p. ej. de pendiente a traducido) o su actualización en la base de datos. También es posible configurar el programa para que de manera automática utilice la misma traducción en todos los segmentos repetidos en el mismo proyecto. Esto es lo que se conoce como **propagación automática** de las repeticiones internas. No obstante, en algunos casos no conviene utilizar la misma traducción en segmentos que aunque sean idénticos en el idioma original pueden requerir traducciones diferentes según el contexto. En estos casos algunos sistemas ofrecen la posibilidad de incluir en la memoria **traducciones alternativas** para el mismo segmento original.

Otras funciones que tienen que ver con la manipulación de los segmentos incluyen la **validación** mediante diferentes estatus (sin traducir, traducido, aprobado, entre otros), y la posibilidad de **añadir comentarios** a las unidades de traducción o de **cambiar la segmentación** mientras traducimos (dividir un segmento en dos o unir segmentos consecutivos).

Existen otras funciones que pretenden facilitar el trabajo de revisión. Algunos sistemas, por ejemplo, permiten hacer comparaciones entre diferentes versiones e implementan opciones similares al **control de cambios** de los procesadores de texto. En otros casos, existe la posibilidad de exportar las traducciones en formato de tabla, bien como documento de texto o como hoja de cálculo, para llevar a cabo

---

invariables entre un par de lenguas dado mientras que para otro par de lenguas pueden requerir ciertas adaptaciones. Es el caso, por ejemplo, de un número con decimales: si traducimos del italiano al español el separador de decimales será el mismo, por lo que en este caso el número sería un elemento colocable; cuando traducimos del inglés al español, por otro lado, es necesario cambiar el separador de decimales, por lo que en este caso el mismo número sería un elemento localizable. De ahí que el autor concluya, acertadamente en nuestra opinión, que el término más adecuado para agrupar este tipo de elementos es "colocables y localizables".

la revisión en herramientas externas. Después de la revisión, el sistema importa de nuevo la tabla y actualiza la memoria con los cambios realizados externamente.

Finalmente, otra opción muy valorada por los traductores es la posibilidad de **previsualizar el formato** de la traducción en cualquier momento desde la interfaz del editor (Lagoudaki, 2009). De este modo, durante la traducción se pueden tener en cuenta los elementos visuales que acompañan al texto y se puede verificar que la disposición de los elementos no se haya alterado.

Al terminar la traducción, el paso final consiste en crear los archivos traducidos en el formato del texto original. Una vez exportados los archivos finales, es recomendable verificar el formato en la aplicación externa que se utilizó para la creación de los originales, teniendo en cuenta que si se realizan cambios directamente en los archivos finales será necesario actualizar la memoria de traducción para que refleje dichos cambios.

## 1.4 Tecnologías libres para la traducción

A pesar de lo mucho que se ha escrito sobre las tecnologías aplicadas a la traducción y la abundancia de herramientas disponibles, la mayoría de los autores limitan sus consideraciones al software privativo comercial que lidera el mercado, dejando de lado los programas libres disponibles y las ventajas que estos pueden tener para los traductores independientes, pequeñas empresas e instituciones de enseñanza de la traducción, especialmente en los países en vías de desarrollo. En este apartado consideraremos los principales aportes desde las dimensiones profesional y académica de la traducción a favor del software libre.

Desde la dimensión profesional de la traducción encontramos algunos traductores que vienen promoviendo el uso de software libre en nuestro gremio. McKey, por ejemplo, es autora de un artículo titulado *Free and Open Source Software for Translators* (2005), ha realizado presentaciones sobre el tema en conferencias de la *American Translators Association* y reservaba un espacio en su sitio web para incluir enlaces e información general sobre el software libre, aunque dicha información ahora está desactualizada. Además, uno de los episodios de su *podcast "Speaking of Translation"* (Bodeux & McKay, 2010), emitido el 5 de febrero de 2010, estuvo dedicado al software libre para traductores.

Un recurso útil para los traductores que quieren saber más sobre el software libre es la recopilación de enlaces disponible en la página web del traductor alemán Wunderlich<sup>16</sup>. En una sección de su web, ahora en proceso de renovación, se presenta un listado de herramientas libres para traductores, sin incluir herramientas generales como editores de texto o procesadores de texto.

Otro sitio web enfocado a los traductores interesados en el software libre es *Linux for Translators*<sup>17</sup> (Prior, s. f.). Como su nombre lo indica, la información allí contenida tiene que ver especialmente con la utilización de programas para traductores en el sistema operativo libre GNU/Linux. Como vimos antes, también es posible utilizar programas privativos en este sistema operativo, por lo que no

---

16 [http://www.martinwunderlich.com/links\\_en.php4](http://www.martinwunderlich.com/links_en.php4)

17 <http://www.linuxfortranslators.org>

todas las herramientas incluidas en el sitio web son necesariamente libres. Prior ha creado también un foro de usuarios<sup>18</sup> como complemento del sitio web. Este foro, que el 16 de mayo de 2012 celebró sus 10 años, cuenta ya con casi 600 miembros registrados. En Prior (2003), el autor presenta las principales características del sistema de memorias de traducción libre OmegaT, destacando las posibilidades que ofrece el software libre en general.

Finalmente, cabe mencionar la distribución de GNU/Linux especialmente para traductores denominada *Linguas OS*, creada en el año 2007 por Baldwin, traductor e intérprete estadounidense. El objetivo de dicha distribución, ya discontinuada, era mostrar a los traductores de manera práctica la posibilidad de utilizar software libre para su trabajo profesional. Baldwin ha desarrollado también varias utilidades que están disponibles como software libre en su página web, en la que también dedica una sección<sup>19</sup> a recoger enlaces a los principales programas libres de interés para los traductores y a explicar la filosofía de este movimiento.

También desde la dimensión académica de la traducción se han explorado las posibilidades del software libre, en algunos casos para su uso en el proceso enseñanza-aprendizaje (ver por ejemplo Bowker, McBride, & Marshman [2008]) y en otros casos participando directamente en el desarrollo de aplicaciones, ya que su modelo de desarrollo y distribución lo hace especialmente interesante para los programas de investigación y desarrollo de las instituciones de educación superior (Rosa & Heinz, 2007). De hecho, en el campo del procesamiento del lenguaje natural la tendencia general de liberar el código de las herramientas desarrolladas viene siendo cada vez más evidente (Koehn & Hoang, 2010).

Algunos ejemplos de proyectos libres surgidos desde la dimensión académica de la traducción son los sistemas de traducción automática Moses (Koehn & Hoang, 2010) y Apertium (Ramírez-Sánchez, Sánchez-Martínez, Ortiz-Rojas, Pérez-Ortiz, & Forcada, 2006), las herramientas de extracción terminológica TES [antes llamado LexTerm] (Oliver, Vázquez, & Moré, 2007) y TTC Term Suite (Blancafort et al.,

---

18 <http://tech.groups.yahoo.com/group/linuxfortranslators/>

19 <http://tonybaldwin.me/hax/doku.php?id=freetran>

2011), la herramienta de alineación Bitext2tmx y la herramienta de revisión de traducciones PetraREV (Perea Sardón, 2010).

Otro ejemplo es el trabajo de Sandrini, profesor del Departamento de Traducción de la Universidad de Innsbruck, quien ha creado dos colecciones de software libre para traductores que se pueden descargar desde su página web<sup>20</sup>, que cuenta con una sección especialmente dedicada al tema (Sandrini, s. f.). La primera de las dos colecciones es una suite de aplicaciones portátiles<sup>21</sup> (libres y *freeware*), denominada *USBTrans*. Esta suite se puede ejecutar desde un dispositivo USB con un mínimo de 512 MB e incluye, además de aplicaciones generales, programas específicos para el traductor. La segunda colección es una distribución del sistema operativo GNU/Linux denominada *Tuxtrans*, también específicamente pensada para las labores de traducción (Sandrini, 2012a). También en este caso se ha creado el grupo de discusión *Tuxtrans*<sup>22</sup> para los usuarios de la distribución. Además, el autor ha publicado recientemente un artículo sobre las tecnologías libres para la traducción en la revista en línea InTRAlínea (Sandrini, 2012b).

Otro esfuerzo en este sentido surgido desde el ámbito académico es el libro Traducción y tecnologías de Oliver et al. (2008), publicado por la Universitat Oberta de Catalunya (UOC), que incluye en sus capítulos información sobre programas libres para diferentes tareas. El profesor Oliver también ha desarrollado varias utilidades para traductores que están disponibles bajo licencias libres. Además, en el posgrado en Traducción y Tecnologías que se ofrece en la UOC se incluyen algunas herramientas libres para traductores y, como asignatura opcional, una introducción al sistema operativo libre GNU/Linux.

El Grupo de Estudio de las Tecnologías Libres de la Traducción de la Facultad de Filología y Traducción en la Universidad de Vigo también cuenta con su propia distribución de GNU/Linux para la enseñanza de la traducción, cuyo proceso de creación se presenta en Díaz Fouces (2007). Esta distribución,

---

20 <http://homepage.uibk.ac.at/~c61302/en-fsfrans.html>

21 Las aplicaciones portátiles son programas que se pueden ejecutar sin necesidad de instalación y sin hacer cambios ni dejar datos en el ordenador huésped (ver por ejemplo <http://portableapps.com> o <http://www.programasportables.com>).

22 <http://groups.google.de/group/tuxtrans>

denominada MinTrad, se puede probar como Live DVD/USB, es decir, sin necesidad de instalación y sin afectar los datos del ordenador huésped.

Las posibilidades que brinda el software libre para la docencia de la traducción en el contexto universitario ha sido un tema explorado ampliamente por Díaz Fouces en diversos artículos (2005a, 2005b, 2006). Otro resultado del trabajo de este grupo de investigación es la edición del libro Traducir (con) software libre (Díaz Fouces & García González, 2008) en el que se aborda el tema desde diferentes perspectivas, incluyendo ejemplos prácticos para la utilización de herramientas libres en el aula de traducción (Cánovas & Samson, 2008a).

La revista especializada *Tradumática*<sup>23</sup>, dedicada a las tecnologías de la traducción, ha dedicado recientemente un número especial al software libre y su relación con la traducción. Los artículos incluidos surgen desde las perspectivas académica, profesional, de investigación y de las comunidades de software libre y abordan diversos aspectos que van desde los conceptos básicos del movimiento, los procesos, herramientas y estándares utilizados en la localización de software libre hasta el papel del software libre en la formación de traductores y en la caja de herramientas del traductor profesional (Díaz Fouces, 2011).

En el ámbito de la investigación en traducción automática el software libre ha generado especial interés, ya que facilita la reproducibilidad de los experimentos y permite la reutilización del código. Son muestra de este creciente interés el número de sistemas de traducción automática y herramientas relacionadas que se han lanzado bajo licencias libres en los últimos años<sup>24</sup> y la abundancia de eventos académicos dedicados especialmente a los sistemas de traducción automática libre<sup>25</sup>, así como la publicación en junio de 2011 de un número de la revista

---

23 <http://revistes.uab.cat/tradumatica>

24 Forcada (s. f.) ha compilado una lista bastante completa que está disponible en <http://computing.dcu.ie/~mforcada/fosmt.html>

25 Por ejemplo, el Taller sobre Traducción Automática Libre (OSMaTran) llevado a cabo como parte del 10º Encuentro sobre Traducción Automática (<http://www.torsimany.ua.es/OSMaTran>); las maratones de traducción automática que han tenido lugar como parte de los proyectos de la Unión Europea EuroMatrix y EuroMatrixPlus (en <http://lium3.univ-lemans.fr/mtmarathon2010> se puede encontrar información sobre la 5ª edición); el 3º Taller sobre Traducción Automática Basada en Ejemplos titulado ‘*Going open-source to revive EBMT*’ (<http://computing.dcu.ie/~mforcada/ebmt3>); y los dos Talleres Internacionales sobre Sistemas de Traducción Automática Libres Basados en Reglas (en <http://www.uoc.edu/freerbmt11> se puede

especializada *Machine Translation* dedicado exclusivamente al tema (Sánchez-Martínez & Forcada, 2011).

## 1.5 Recapitulación

En los apartados anteriores hemos abordado los aspectos fundamentales que nos permitirán una mejor comprensión de las tecnologías aplicadas a la traducción. En el apartado 1.1 presentamos las generalidades sobre el movimiento del software libre y los conceptos básicos sobre los diferentes tipos de software según la licencia. Además, abordamos las principales oportunidades que ofrece el software libre, así como las barreras y riesgos que pueden dificultar su implementación.

En el apartado 1.2 exploramos las diferentes perspectivas de clasificación de las herramientas informáticas que pueden facilitar la labor del traductor. Como vimos, algunas propuestas de clasificación se basan en el grado de automatización del proceso de traducción, mientras que otras consideran la fase del proceso de traducción en que se utilizan las herramientas, el nivel de conocimientos informáticos necesarios para utilizarlas, su relación con la traducción o la dimensión de la traducción en la que se utilizan (didáctica, profesional o investigación). Después de considerar las diversas propuestas, para este trabajo adoptamos la clasificación de Alcina (2008). A las cinco categorías de dicha clasificación (utilidades generales, herramientas de comunicación, herramientas de edición y maquetación, herramientas lingüísticas y herramientas de traducción) añadimos una más (herramientas de gestión).

En el apartado 1.3 abordamos las necesidades de los traductores en términos de los diferentes escenarios de uso de las herramientas. Por un lado, vimos que es posible distinguir entre dos grandes contextos profesionales: traductores independientes y organizaciones, que a su vez pueden ser agencias de traducción, empresas con departamento de traducción interno, organizaciones bilingües, organizaciones internacionales o instituciones públicas. Por otro lado, encontramos que las necesidades también varían en relación con el papel que desempeña el

---

encontrar información sobre la edición de 2011).



traductor en un proyecto dado, que puede ser gestor del proyecto, gestor de recursos lingüísticos, traductor, posteditor, maquetador o revisor, entre otros. En este apartado exploramos también las principales características que ofrecen los sistemas de memorias de traducción en la actualidad, ya que la evaluación que nos proponemos realizar abarcará específicamente este tipo de herramientas.

Finalmente, en el apartado 1.4 presentamos las principales iniciativas enfocadas a promover el uso de las tecnologías libres para la traducción, tanto en la dimensión profesional como en la académica. Como vimos, las tecnologías libres han venido ganando terreno en la última década, algo que se evidencia tanto en el aumento del número de programas disponibles como en el creciente interés entre investigadores, académicos y profesionales de la traducción. En este contexto, resulta entonces relevante llevar a cabo un inventario de las alternativas libres existentes en la actualidad que sirva de punto de partida a los interesados en explorar estas alternativas. Esta es, pues, la tarea que nos propusimos realizar en la primera fase de nuestro trabajo empírico (ver apartado 3.2). Al favorecer una mayor difusión de este tipo de herramientas entre los traductores presentes y futuros, con este trabajo esperamos contribuir a contrarrestar el desconocimiento general, una de las principales barreras para la adopción de software libre.



## 2 CALIDAD DE SOFTWARE

Cuando abordamos el tema de la calidad nos enfrentamos a un concepto multidimensional y subjetivo que dependerá en cierta medida de aquello que nos interese evaluar, de la perspectiva particular sobre lo evaluado y de las propiedades que nos interesen según el contexto (Kan, 2002). En el caso del software libre para traductores, parece que por ser en general gratuito primara cierta concepción popular de calidad que podríamos resumir con el dicho "lo barato sale caro"; en otras palabras, lo que cuesta poco (o nada), suele considerarse de dudosa calidad.

De ahí que resulte relevante llevar a cabo una evaluación de las tecnologías libres para la traducción que nos permita corroborar o rebatir esta percepción general. La pregunta es, entonces, ¿cómo podemos evaluar las tecnologías libres para la traducción desde el punto de vista del traductor como usuario final? Buscando responder a esta pregunta, en este capítulo consideraremos los modelos de evaluación y calidad que se han propuesto desde la ingeniería de software y desde otros campos relacionados, como los sistemas de información y, naturalmente, la ingeniería lingüística. En el apartado 2.1 abordaremos las generalidades sobre la calidad de software para pasar luego a considerar la evaluación de las tecnologías de la traducción (2.2) y del software libre (2.3). Finalmente, en el apartado 2.4 se hará una recapitulación de los temas tratados en el capítulo.

### 2.1 Generalidades

Para empezar, encontramos que en la ingeniería de software se define la calidad como “el grado en el que un objeto (entidad) (ej. proceso, producto o servicio) satisface una serie de atributos o requerimientos especificados” (Schulmeyer, 2006, p. 6). En cuanto a la delimitación del objeto o entidad, existen dos concepciones diferentes: una más limitada, conocida como  $q$ , que comprende solo la calidad

intrínseca del producto, y otra más general, conocida como  $Q$ , que además de tener en cuenta el producto se ocupa también del proceso de desarrollo y de la satisfacción de los usuarios (Kan, 2002).

Groven et al. (2011) resumen estas tres perspectivas de calidad (producto, proceso y satisfacción de los usuarios) de la siguiente manera: la perspectiva del producto, basada en la concepción de que “las características de los productos se definen mediante las características (...) de sus partes”; en la perspectiva de fabricación, por otro lado, “la calidad del producto se deriva del cumplimiento de las especificaciones y de la capacidad de las organizaciones para producir software según procesos de software definidos”; y, finalmente, la perspectiva del usuario, según la cual la calidad del software equivale a “satisface[r] las necesidades de los usuarios, donde la confiabilidad, desempeño/eficiencia, mantenibilidad y usabilidad son aspectos centrales” (p. 74).

Garvin (1984), por su parte, consideraba otras dos perspectivas posibles para describir la calidad de un producto: la perspectiva trascendental, según la cual la calidad se puede reconocer pero no definir; y la perspectiva basada en el valor (que ya mencionábamos al empezar el capítulo) según la cual hay una relación directa entre la calidad y lo que el usuario esté dispuesto a pagar por el producto.

Ya en la práctica, en las últimas cuatro décadas se han seguido dos enfoques principales para la comprensión y estudio de la calidad de software: uno diacrónico basado en la gestión de la calidad, en el que se asume una postura flexible y cualitativa y una metodología correctiva (normalmente utilizado de manera interna en la organización que desarrolla el software); y otro basado en modelos de calidad, en el que se sigue una metodología descriptiva con una postura más rígida desde la que se entiende la calidad como un concepto cuantificable, bien sea en términos de la adherencia a los procesos o con base en la medición o valoración de una serie de atributos (Groven et al., 2011).

En esta última línea se enmarca el estándar ISO 9126 («ISO/IEC 9126. Software engineering. Product quality.», 2001), que establece un modelo de calidad de software y las directrices para el uso de dicho modelo. El modelo de calidad

propuesto comprende dos partes: en la primera parte se especifican las características que permiten determinar la calidad interna y externa del software, mientras que la segunda parte se ocupa del concepto de calidad en uso.

- ***Calidad interna y externa***

La calidad interna y externa del software como producto se refiere a las propiedades del software en sí y según el estándar comprende seis características: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. Pasaremos ahora a explicar cada una de estas características con base en el estándar ISO 9126.

La **funcionalidad**, en primer término, abarca aquellos aspectos relacionados con el logro de los propósitos para los cuales se desarrolla el software y que incluyen: a) la adecuación, que se refiere a la presencia e idoneidad de las funciones incluidas en el programa para realizar ciertas tareas; b) la exactitud, que se entiende como la capacidad del programa para proporcionar los resultados correctos o acordados; c) la interoperabilidad, es decir, la capacidad del software para interactuar con otros sistemas; d) la seguridad, que se refiere a la capacidad del producto de evitar que usuarios o sistemas no autorizados accedan al programa o a los datos de manera accidental o deliberada; y, finalmente, e) el cumplimiento funcional, que se entiende como la adhesión a los estándares, convenciones, legislación o protocolos relacionados con la aplicación.

La **fiabilidad**, por su parte, comprende aquellos aspectos que tienen que ver con la capacidad del programa para mantener su nivel de desempeño bajo ciertas condiciones y por un determinado periodo de tiempo. Así pues, son subcaracterísticas de la fiabilidad: a) la madurez, que se refiere a la frecuencia con la que falla el software por errores en el programa; b) la tolerancia a fallos, que se entiende como la capacidad del programa de mantener un nivel de desempeño en caso de fallos o de entradas inesperadas; c) la recuperabilidad, es decir, la posibilidad de restablecer un nivel de desempeño especificado y de recuperar los datos después de un fallo, así como el esfuerzo necesario para hacerlo; y d) el cumplimiento de la fiabilidad, es decir, la adhesión a los estándares, convenciones, legislación o protocolos de fiabilidad aplicables.

La **usabilidad**, por otro lado, abarca los aspectos que tienen que ver con el esfuerzo necesario para utilizar el programa y con la manera como los usuarios lo perciben. Según el estándar de calidad, esta característica se descompone en las siguientes subcaracterísticas: a) la comprensibilidad, que se refiere al grado de esfuerzo necesario para entender el concepto lógico del programa y su aplicabilidad; b) la capacidad para ser aprendido, que se entiende como el esfuerzo necesario por parte del usuario para aprender la aplicación del programa; c) la operabilidad, es decir, la facilidad para usar y controlar el programa; d) la capacidad del programa de resultar atractivo para el usuario; y e) el cumplimiento de la usabilidad, que se refiere a la adhesión a los estándares, convenciones, legislación o protocolos de usabilidad pertinentes según el tipo de aplicación.

La **eficiencia** engloba aspectos que tienen que ver con el rendimiento del programa en términos de tiempos de respuesta en relación con los recursos que utiliza en determinadas condiciones. Son subcaracterísticas de la eficiencia: a) el comportamiento temporal, que se refiere a los tiempos de procesamiento y respuesta del programa en unas condiciones dadas; b) la utilización de recursos, que se entiende como la cantidad de recursos utilizados y el tiempo que toma ejecutar alguna función; y c) el cumplimiento y adhesión a los estándares, convenciones, legislación o protocolos de eficiencia pertinentes.

La **mantenibilidad** comprende los aspectos relacionados con el esfuerzo necesario para modificar (corregir, mejorar o adaptar) el programa. Se compone, pues, de las siguientes subcaracterísticas: a) la analizabilidad, entendida como el esfuerzo necesario para diagnosticar e identificar las causas de los fallos y las modificaciones necesarias para corregirlos; b) la modificabilidad, es decir, el esfuerzo necesario para realizar modificaciones, bien se para la corrección de fallos o para la adaptación a cambios en el entorno; c) la estabilidad, que se refiere al riesgo de que las modificaciones causen efectos inesperados; d) el esfuerzo necesario para someter el programa a las pruebas requeridas para validar las modificaciones; y e) la adhesión a los estándares, convenciones, legislación o protocolos de mantenibilidad pertinentes.

La sexta y última característica de calidad interna y externa del producto según la norma ISO 9126 (2001) es la **portabilidad**, que abarca los aspectos relacionados con la posibilidad de transferir el programa entre diferentes entornos de hardware y software. Se consideran subcaracterísticas de la portabilidad: a) la adaptabilidad, entendida como la facilidad para adaptar el programa a diferentes entornos específicos; b) la instalabilidad, es decir, el esfuerzo necesario para instalar el programa en un entorno dado; c) la coexistencia, entendida como la capacidad del programa de coexistir con otros programas en el mismo entorno; d) la capacidad de reemplazar, que se refiere a la posibilidad de utilizar el programa en lugar de otro en un entorno determinado; y e) el cumplimiento de las normativas aplicables que tengan que ver con la portabilidad. En la figura 2.1 se presenta la descomposición de la calidad interna y externa del software según la norma ISO 9126.

- **Calidad en uso**

Pasamos ahora de la calidad interna y externa del software a la calidad en uso, que se refiere al grado en que un usuario determinado puede lograr sus objetivos en unas condiciones específicas de uso. Según la norma ISO 9126 (2001), la calidad en uso se descompone en cuatro características: la efectividad, la productividad, la seguridad física y la satisfacción.

La **efectividad** se refiere a la exactitud y completud de los resultados que se pueden obtener con el programa en un escenario de uso dado. La productividad, por su parte, se define según los recursos utilizados en relación con la efectividad obtenida en unas condiciones de utilización específicas. La **seguridad física** tiene que ver con el riesgo de que el programa cause algún daño a las personas, los datos, el medio ambiente o al programa mismo en un contexto determinado. La **satisfacción**, por último, se entiende como la capacidad del programa de satisfacer a los usuarios en un escenario de uso dado. En la figura 2.2 se presenta la descomposición de la calidad en uso según la norma ISO 9126.

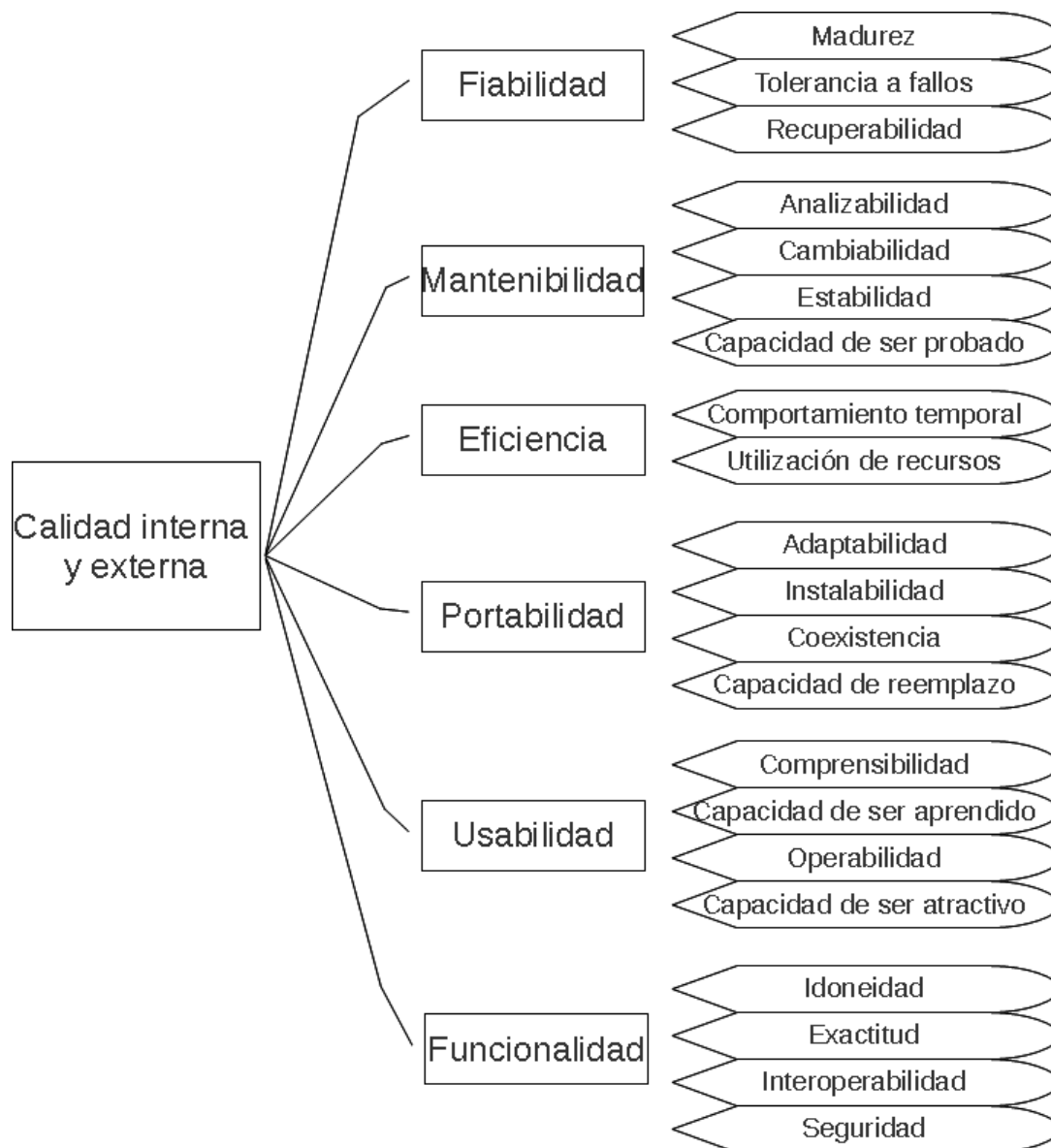


Figura 2.1: Calidad interna y externa del software según el estándar ISO 9126 (2001).

Cada una de las características y subcaracterísticas de calidad mencionadas hasta ahora se descompone a su vez en una serie de **atributos** que son los que realmente se miden utilizando ciertas escalas o valores definidos para este propósito (Cataldi, 2000). En el contexto de la evaluación de software estas escalas o valores se conocen como **métricas**. Es importante tener en cuenta que las métricas, al ser medidas sólo de las manifestaciones de la calidad, constituyen siempre medidas indirectas de esta (Ibid.). Se entiende, pues, que toda evaluación, a diferencia de la



medición en sí, “es siempre parte de un proceso de toma de decisiones y representa su dimensión subjetiva” (Morisio, Stamelos, & Tsoukias, 2003, p. 98); en otras palabras, diferentes evaluadores pueden expresar diferentes preferencias para un mismo atributo según su contexto y necesidades.

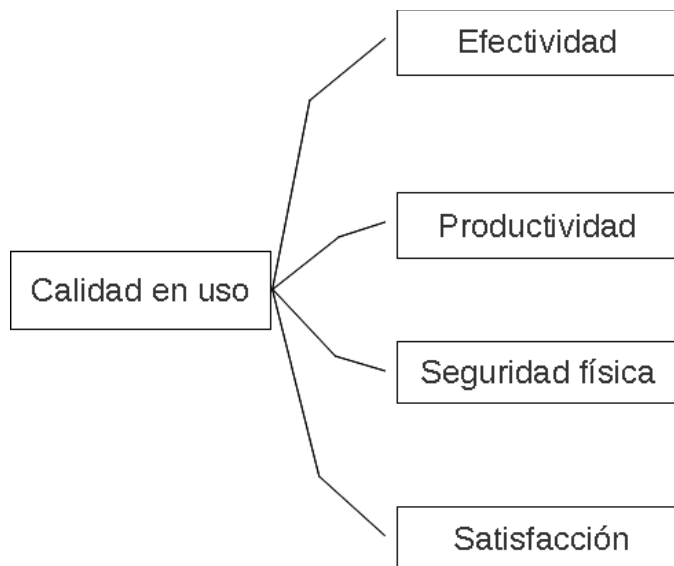


Figura 2.2: Calidad en uso según el estándar ISO 9126 (2001).

- **El proceso de evaluación**

En el estándar ISO 9126 (2001) se incluyen también las directrices para llevar a cabo el proceso de evaluación, en el que se distinguen tres fases. En primer lugar se establecen los requisitos de calidad que se considerarán, para lo cual se tienen en cuenta las necesidades de los usuarios, las especificaciones del programa y las seis características de calidad; la segunda fase es la preparación de la evaluación, que implica la selección de las métricas, la definición de las categorías en que se agruparán las puntuaciones (por ejemplo, satisfactorio e insatisfactorio) y la definición de los criterios que se utilizarán para resumir los resultados y presentar la valoración general del software. En la tercera y última fase se aplican las métricas, se asignan las puntuaciones y se resumen y presentan los resultados generales de la evaluación (EAGLES, 1996; Quah, 2006).

Existe otra norma que también se ocupa del proceso de evaluación: el estándar ISO 14598 («UNE-ISO/IEC 14598. Information Technology. Software Product Evaluation», 1998). En este estándar encontramos que el proceso de

evaluación se divide en cinco etapas. En la primera, se identifican los requisitos de la evaluación según el propósito, es decir, se decide si la evaluación se centrará en algunos componentes o en los programas como un todo, y se establecen además los criterios y parámetros de calidad que se utilizarán, que suelen ser los especificados en ISO 9126.

En la segunda etapa, teniendo en cuenta los criterios de la primera, se establece el tipo de métricas que se utilizarán según la descripción de los productos que se quieren evaluar. Además, se deben establecer categorías según las puntuaciones para cada atributo y se debe decidir qué procedimiento se utilizará para resumir las puntuaciones de todos los atributos en una sola valoración general.

A continuación, en la etapa de diseño de la evaluación, se establece el cronograma y se deciden los métodos que se utilizarán, para lo cual es importante considerar el grupo de interés que motiva la evaluación (investigadores, desarrolladores, usuarios finales, etc.).

La cuarta etapa constituye la evaluación en sí, que comprende la aplicación de las métricas, la clasificación según las puntuaciones, su interpretación y el resumen por escrito de dicha interpretación.

En la etapa final se redacta el informe completo de la evaluación, que incluye los resultados, las conclusiones y una descripción de los problemas encontrados durante el proceso (EAGLES, 1996; Quah, 2006).

En los siguientes apartados veremos cómo se han utilizado estas recomendaciones recogidas en las normas ISO y otros enfoques a la hora de evaluar las tecnologías de la traducción (apartado 2.2) y el software libre y de código abierto (apartado 2.3). Para cerrar el capítulo se presentará un resumen y conclusiones sobre los aspectos más relevantes aplicables a nuestro contexto de evaluación (apartado 2.4).

## 2.2 Evaluación de las tecnologías de la traducción

For anyone interested in evaluation, it is important to bear in mind that there is still no standard method of evaluation that is reliable and acceptable (...) Nevertheless, it is generally agreed that evaluation is important for translation tools. (Quah, 2006, p. 129)

Como bien resaltan Rico (2001), Höge (2002), Daelemans y Hoste (2010) y Quah (2006), el primer aspecto que se debe mencionar con respecto a la evaluación de las tecnologías de la traducción es la ausencia de métodos estándar que sean fiables, aceptables y reproducibles, hecho que según este último autor se debe en gran medida a la abundancia de herramientas existentes y a la variedad de grupos de interés, que incluyen investigadores, desarrolladores, patrocinadores y usuarios finales, que a su vez pueden ser proveedores de servicios lingüísticos, traductores independientes, estudiantes o profesores de traducción, entre otros.

Otra posible razón para esta ausencia es el hecho de que las tecnologías de la traducción son un avance relativamente reciente (Gow, 2003), con una excepción: la traducción automática. En este último campo en particular existe una gran trayectoria de evaluación que se remonta al informe de 1966 del *Automatic Language Processing Advisory Committee* (ALPAC). Desde ese entonces, son numerosos los proyectos de evaluación que se han emprendido en esta área. Si bien se han abordado aspectos como la misma metodología de evaluación, la relación coste-beneficio, y los aspectos técnicos de interés para los desarrolladores, la tendencia general ha sido concentrarse en dos aspectos: la inteligibilidad y la fidelidad de los resultados obtenidos con los sistemas de traducción automática evaluados (Quah, 2006).

Como señalan Rico (2001) y Höge (2002), durante los últimos veinte años también se han llevado a cabo numerosos proyectos de evaluación de herramientas de traducción, pero en su mayoría se trata de evaluaciones que no son generalizables pues “se concentran en productos particulares utilizados en un contexto en particular y bajo ciertas condiciones específicas que, a su vez, limitan la evaluación del sistema a una serie de necesidades predefinidas y específicas” (Rico,

2001, p. 2, traducción propia). En otros casos, no se establece una metodología de evaluación de software propiamente dicha sino que se limitan a presentar una comparación de dos o más sistemas en términos de su funcionalidad (Höge, 2002).

En el campo de la traducción se distinguen tres posibles contextos de evaluación: en el contexto de la investigación y desarrollo de tecnologías de la traducción; desde el punto de vista de los usuarios potenciales; o en el contexto de la enseñanza de la traducción (Höge, 2002). Se distinguen también tres posibles perspectivas de evaluación de herramientas según se evalúe a) el resultado producido por la herramienta (evaluación declarativa), b) el proceso de interacción entre traductor y herramienta (evaluación operacional), o c) el sistema, es decir, la herramienta (evaluación de software) (Cerezo, 2003). Esta última perspectiva es la que nos ocupa en este trabajo y es en la que nos concentraremos a continuación.

Como tentativa de solución al problema de la falta de criterios estandarizados de evaluación mencionado al principio de este apartado, en años más recientes se han hecho varios esfuerzos por establecer un marco general o serie de directrices de referencia para la evaluación de las tecnologías lingüísticas (Quah, 2006), categoría que engloba a las tecnologías de la traducción. La primera de estas iniciativas fue emprendida en 1993 por el *Expert Advisory Group on Language Engineering Standards* (EAGLES), financiado por la Unión Europea. Tomando como base las seis características propuestas por el estándar ISO 9126, el propósito del proyecto era:

Identificar y especificar los componentes de un compendio de criterios de evaluación y técnicas asociadas, junto con las directrices para su utilización, de entre las cuales el usuario individual de la evaluación pueda seleccionar aquellas técnicas que sean relevantes para su propósito (EAGLES, 1996, traducción propia).

En resumen, el proyecto buscaba desarrollar un marco general que sirviera de punto de partida para la evaluación de las tecnologías de la lengua (King, 1998).

Como continuación del trabajo emprendido por EAGLES, en el año 2000 se dio inicio a un proyecto cooperado entre Europa y Estados Unidos denominado *International Standards for Language Engineering* (ISLE) con tres grupos de

trabajo, uno de ellos dedicado al tema de la evaluación (*Evaluation Working Group*, EGW) (Calzolari, McNaught, Palmer, & Zampolli, 2003). El trabajo de este grupo se concentró en el área de la traducción automática, por ser esta una de las tecnologías más difíciles de evaluar, aunque la idea a largo plazo era que los resultados obtenidos se pudieran generalizar para la evaluación de otras tecnologías de la lengua (Calzolari et al., 2003).

Como resultado del trabajo de este grupo se desarrolló el *Framework for the Evaluation of Machine Translation in ISLE* (FEMTI), que constituye una colección estructurada de métodos para la evaluación de sistemas de traducción automática (Calzolari et al., 2003; Quah, 2006). Otro trabajo derivado de la iniciativa EAGLES fue el *Test-bed Study of Evaluation Methodologies: Authoring Aids* (TEMAA), que tenía como objetivos principales favorecer la reflexión sobre el proceso de evaluación de las herramientas de procesamiento de lenguaje natural y trabajar en la creación de una herramienta que llevara a cabo dicho proceso de manera automatizada (Quah, 2006; TEMAA, s. f.). En el marco de dicho proyecto se llevaron a cabo estudios de caso sobre la evaluación de correctores ortográficos y gramaticales, así como de herramientas de recuperación de información.

Ya que ISLE y FEMTI se centran en la evaluación de sistemas de traducción automática y TEMAA en la automatización del proceso de evaluación, nos concentraremos ahora en el marco propuesto por EAGLES por ser el que mejor se adapta a los objetivos del presente trabajo.

### **2.2.1 La propuesta metodológica del grupo EAGLES**

El primer aspecto importante de la iniciativa EAGLES es que se establece una distinción entre los tipos de evaluación según el propósito por el que se lleva a cabo y el público objetivo de los resultados de la misma (Höge, 2002). Se distinguen, pues, tres tipos de evaluación:

- Evaluación de adecuación: cuando el objetivo es determinar si un sistema es adecuado para un propósito dado (enfocada a los usuarios).

- Evaluación de progreso (también conocida como evaluación de rendimiento): cuando el objetivo es evaluar el estado y desempeño de un sistema con respecto al desempeño ideal esperado (enfocada a los desarrolladores o a gestores de investigación y desarrollo).
- Evaluación diagnóstica: cuando el objetivo es estudiar por qué o cuándo falla un sistema (enfocada a los desarrolladores).

El trabajo de EAGLES se limitó al primero de estos tres tipos de evaluación y asumió un enfoque orientado hacia el usuario final en el que se da especial relevancia al contexto específico de cada proyecto de evaluación según el tipo de herramienta a evaluar y el entorno de uso previsto.

Según EAGLES, los métodos de evaluación deben ser fiables, válidos y fácilmente aplicables. Fiable, se refiere a que se deben obtener resultados idénticos o similares cuando la evaluación se lleva a cabo en el mismo contexto aun con diferentes evaluadores. Válido, por su parte, se refiere al hecho de que los usuarios finales puedan hacer inferencias sobre las herramientas con base en los resultados de la evaluación. Fácilmente aplicable, por último, implica que llevar a cabo la evaluación debe requerir el menor esfuerzo posible, especialmente para los usuarios finales.

El marco general resultante de la iniciativa de EAGLES se basa en las características de calidad y en las etapas del proceso de evaluación propuestas en los estándares ISO 9126 y 14598 y tiene tres componentes principales: atributos, requisitos y métodos. Los **atributos** son las características que se considerarán en la evaluación y se miden mediante escalas y valores; por lo tanto es posible clasificar los atributos según el tipo de valores que aceptan (binarios, clasificatorios, comparativos, numéricos o métricos). Gran parte del trabajo del grupo EAGLES se concentró precisamente en la formalización de los estándares relacionados con la calidad de software (ISO 9126 y 14598) mediante la creación de modelos de calidad con atributos específicos para las tecnologías del lenguaje y la definición y validación de las métricas para medir dichos atributos (King, 1998).

Los **requisitos** son las necesidades, explícitas o implícitas, que un programa pretende satisfacer. En el marco de evaluación de EAGLES se expresan como atributos y se dividen en dos categorías: funcionales y no funcionales. Los primeros están relacionados con la característica de funcionalidad del estándar ISO 9126 y por tanto varían según el tipo de programa, mientras que los segundos corresponden a las demás características (fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) y son igualmente aplicables a cualquier tipo de programa (Höge, 2002).

Para responder al hecho de que las necesidades de los traductores varían según la situación de uso, como parte de la funcionalidad el grupo EAGLES agregó la subcaracterística *configurabilidad*, que se refiere a la posibilidad de que el usuario ajuste el sistema a diferentes situaciones según sus necesidades (Höge, 2002; Rico, 2001) (por ejemplo, la posibilidad de configurar los colocables o la utilización de filtros de búsqueda según los atributos *changeid* o *changedate* en las memorias de traducción).

Los **métodos**, finalmente, determinan la manera como se establecerán los valores de los atributos y están divididos en tres subcomponentes: tipos de pruebas, instrumentos y materiales. Los **tipos de pruebas** están determinados por las motivaciones subyacentes a la evaluación, que según el mismo grupo EAGLES se pueden categorizar en tres tipos principales: a) valorar la idoneidad de un programa para el trabajo diario; b) examinar el comportamiento de un programa en condiciones específicas; y c) comprobar la funcionalidad general de un programa. Así pues, a cada una de estas motivaciones le corresponde un tipo de prueba distinto, a saber, pruebas de escenario, pruebas sistemáticas e inspección de características, respectivamente (Höge, 2002).

Las **pruebas de escenario** son las más apropiadas para evaluar la usabilidad, ya que implican la puesta en funcionamiento del sistema para el propósito que fue diseñado y se tiene en cuenta el perfil y entorno de trabajo de un usuario final típico. Este tipo de pruebas se puede llevar a cabo en el entorno real del usuario o en laboratorio y los instrumentos utilizados incluyen listas de comprobación,

entrevistas antes y después de usar el programa, protocolos de pensar en voz alta, archivos de registro y programas de grabación de actividades.

En las **pruebas sistemáticas**, por otro lado, se parte de unas condiciones específicas de utilización del programa y los resultados suelen ser los esperados. Estas pruebas suelen ser llevadas a cabo por ingenieros de software, desarrolladores o representantes de los usuarios, pero no por los usuarios mismos. La sistematicidad de estas pruebas está dada por uno de tres enfoques: orientado a las tareas, orientado a los menús o comparativas de rendimiento (*benchmarking*).

En las **pruebas sistemáticas orientadas a las tareas** se predefine una serie de tareas con el fin de comprobar si el sistema se comporta según lo esperado. Este tipo de pruebas se puede llevar a cabo en cualquier fase del ciclo de vida del software y es especialmente indicada para evaluar los atributos correspondientes a la funcionalidad de los programas. Las **pruebas sistemáticas orientadas a los menús**, por su parte, son bastante detalladas ya que se utilizan los menús como punto de partida para examinar secuencialmente todas las funciones ofrecidas por el programa. Al igual que las pruebas orientadas a las tareas, este tipo de enfoque también es aplicable en cualquier fase del ciclo de vida de los programas. Finalmente, las **pruebas tipo benchmarking**, o comparativas de rendimiento, permiten examinar el desempeño de ciertas funciones individuales, ciertos módulos o el sistema completo de manera que, en teoría, el factor humano no incide en los resultados.

El tercer tipo de pruebas, la **inspección de características**, tiene como objetivo describir las características técnicas de un programa detalladamente, de manera que el usuario final tenga parámetros para comparar sistemas similares y pueda así decidir cuál es el más apropiado según sus necesidades. Para este tipo de pruebas no siempre es necesario ejecutar el programa, ya que se basa en la comparación de las características del programa con una lista de comprobación predefinida, de manera que la información necesaria puede muchas veces obtenerse de la documentación del programa. Este proceso tiene un carácter más bien descriptivo y es bastante objetivo, ya que generalmente se limita a establecer la



ausencia o presencia de las características de la lista de comprobación y no se ocupa de identificar problemas o errores del programa.

Naturalmente, el éxito de este último tipo de pruebas depende en gran medida de la lista de comprobación que se utilice, que debe incluir los aspectos que son importantes desde el punto de vista del usuario y lo que es técnicamente posible. Para elaborar las listas de comprobación el evaluador debe estar familiarizado con una amplia gama de sistemas del mismo tipo de manera que comprenda su filosofía subyacente. Además, se recomienda utilizar un enfoque de arriba hacia abajo, es decir, empezar por las características principales para luego abordar los detalles. Con este tipo de pruebas es posible examinar las 6 características de calidad, si bien el énfasis se hace en la funcionalidad. Otra característica de este tipo de pruebas es que a medida que se va realizando la evaluación va quedando listo el informe de los resultados, y la fase de análisis de los datos se reduce al mínimo ya que estos prácticamente no dan lugar a interpretación.

Como recomendación general, cada ítem de la lista de comprobación se debe poder evaluar con base en una escala bueno/malo, según la presencia o ausencia de una determinada característica o mediante un valor numérico de la función o característica. Según Höge (2002), quien también explica en detalle los tipos de escala más convenientes para cada característica de calidad, en las listas de comprobación se utiliza especialmente la escala nominal binaria de presencia o ausencia, lo que resulta mucho menos complejo y subjetivo que utilizar escalas ordinales para medir lo apropiadas que son las funciones ofrecidas. La tabla 2.1 resume los tipos de pruebas para la evaluación orientada a los usuarios según el marco metodológico definido por el proyecto EAGLES.

El segundo subcomponente de los métodos de evaluación según el marco general de EAGLES son los **instrumentos**, que se clasifican en dos tipos: instrumentos para la recolección de datos e instrumentos para la presentación de los resultados obtenidos. Los primeros pueden a su vez clasificarse en **instrumentos para recolección manual de datos**, entre los que se encuentran por ejemplo los cuestionarios, listas de comprobación, entrevistas y protocolos de pensar en voz

alta, e **instrumentos para recolección automática de datos**, que incluyen los programas que graban las interacciones entre el usuario y el sistema y los programas diseñados específicamente para llevar a cabo pruebas tipo *benchmarking*.

Tipo de prueba	Subtipos	Objetivo
Pruebas de escenario	Pruebas de campo Pruebas de laboratorio	Valorar la idoneidad de un programa para el trabajo diario.
Pruebas sistemáticas	Pruebas orientadas a las tareas Pruebas orientadas a los menús Pruebas tipo <i>benchmarking</i>	Examinar el comportamiento de un programa en condiciones específicas.
Inspección de características	--	Comprobar la funcionalidad general de un programa.

Tabla 2.1: Tipos de pruebas en la evaluación orientada a los usuarios (fuente: Höge, 2002, p. 114).

Los **instrumentos para presentación de resultados**, por otro lado, incluyen la descripción de la evaluación, informes de problemas, informes de resultados e informes de valoración general.

El tercer y último subcomponente de los métodos de evaluación son los **materiales** utilizados en las pruebas, que pueden ser textos auténticos en formato electrónico o conjuntos de textos especialmente preparados para que sirvan de entrada durante la evaluación.

## 2.2.2 Adaptaciones de las normas ISO y la propuesta de EAGLES

Como hemos visto hasta ahora, las normas ISO 9126 y 41598 y el grupo EAGLES proponen una metodología de evaluación bastante detallada, si bien se limitan sobre todo a establecer un modelo teórico. A continuación veremos algunos proyectos en los que se plantea algún tipo de evaluación de las tecnologías de la traducción con base en la norma ISO 9126 y en el trabajo de EAGLES. Dado el auge de las tecnologías en los últimos años, existen numerosos trabajos que abordan también la evaluación pero que no se incluyen aquí por limitarse a problemas de evaluación muy específicos (como Cerezo, 2003; Gow, 2003; y Lagoudaki, 2007) o por constituir simples comparativas de funcionalidad de las herramientas (como, por

ejemplo, los trabajos de Zerfaß, 2002; Bowker & Barlow, 2004; Eisele, Federmann, & Hodson, 2009; y Wiechmann & Fuhs, 2006).

### **2.2.2.1 Höge (2002)**

En su tesis doctoral, Höge (2002) presenta sus reflexiones después de diez años de trabajo en el campo de la evaluación de las tecnologías de la traducción desde el punto de vista del usuario. Su trabajo complementa y aplica el marco teórico del grupo EAGLES en la evaluación de distintos sistemas de memorias de traducción como parte del proyecto ESPRIT II (1987-1992), financiado por la Comisión Europea. Para aplicar su propuesta metodológica, la autora evalúa dos sistemas de traducción: Trados Translator's Workbench e IBM TM/2.

La metodología que propone Höge (2002) abarca dos situaciones típicas de evaluación: a) la evaluación que se hace como parte de un proceso de adquisición de software; y b) la evaluación que se hace como parte del proyecto de desarrollo con el fin de mejorar el software. La autora parte de una perspectiva interdisciplinaria que toma elementos de la teoría de la traducción, de la ingeniería de software y de requerimientos, y de la teoría de decisiones. El resultado es un enfoque orientado a las tareas en el que para cada tarea en el proceso de traducción se detalla un árbol de atributos de calidad y un peso asignado según su importancia.

Höge (2002) también afirma que hay tres aspectos que pueden tener especial relevancia a la hora de evaluar las tecnologías de la traducción: a) conocer las necesidades de los usuarios; b) entender la funcionalidad de los sistemas que se evalúa; y c) conocer las técnicas necesarias para realizar pruebas de software.

Höge (2002) empieza pues por establecer los requerimientos funcionales de los sistemas de traducción a partir de las tareas que se llevan a cabo durante el proceso de traducción, los tipos de problemas que se enfrentan y los tipos de estrategias que se emplean para resolver esos problemas. A continuación, establece una tabla de correspondencias entre las tareas y las características de calidad de la norma ISO 9126. Luego, la autora crea una lista con los aspectos cualitativos relacionados con las acciones, objetos, actores, casos de uso, flujos de datos y datos involucrados en el proceso de traducción y las características de calidad pertinentes

y a partir de esos datos define las métricas y las escalas para evaluar los atributos. Finalmente, se indica la opción preferida en cada caso, lo que permite luego resumir los resultados de la evaluación en una valoración global que permite a su vez comparar los sistemas evaluados.

Luego de aplicar su metodología de evaluación, la autora concluye que su propuesta teórica resulta viable en la práctica, si bien es un proceso de evaluación que supone demasiado tiempo y recursos y que por tanto queda limitado a las posibilidades de grandes empresas o proyectos internacionales. Como posibles alternativas para reducir el esfuerzo necesario para llevar a cabo la evaluación, Höge (2002) propone la elaboración de perfiles de usuarios según las tareas que realizan, así como la utilización de sitios web que permitan la elaboración, recopilación, presentación y reutilización de las métricas, resultados y herramientas creados para la evaluación.

#### ***2.2.2.2 Rico (2001)***

También Rico (2001) propone un modelo de evaluación orientado al usuario final preparado con base en la metodología propuesta por EAGLES y las características de calidad definidas por la norma ISO 9126. Su objetivo era definir un modelo de carácter general que pudiera ser reutilizado y aplicado en diferentes contextos de traducción. Para lograrlo, la autora empezó por definir los posibles contextos de uso de las herramientas (industria, administración pública, agencias o traductores autónomos) y por analizar qué características son relevantes para la evaluación teniendo en cuenta los elementos involucrados en todo proyecto de traducción: el cliente, la traducción como producto, la traducción como proceso y el sistema, es decir, la herramienta. Este último elemento se descompone a su vez en los atributos de precisión, interoperabilidad, cumplimiento de estándares, seguridad, portabilidad, usabilidad, eficiencia, mantenibilidad, respaldo y servicio, política de precios, inversión, personalización y política de actualizaciones.

A cada atributo se le asigna un peso según su importancia en el contexto de uso previsto y para medir cada atributo la autora propone utilizar tres categorías de puntuación: óptimo, parcialmente satisfactorio y pobre. Si un atributo es óptimo, se le asigna la puntuación máxima ponderada, si es parcialmente satisfactorio se le

asigna la mitad y si es pobre el valor asignado es 0. De este modo, sumando los valores asignados a cada atributo es posible obtener una puntuación ponderada como valoración global de cada herramienta, con una puntuación máxima posible de 100 puntos.

A modo de ejemplo de aplicación del modelo la autora define los pesos para dos escenarios simulados: el departamento de traducción en una organización internacional (donde parte del trabajo lo hacen los traductores en plantilla mientras un pequeño porcentaje se subcontrata) y la gestión de traducciones en una empresa de telecomunicaciones (donde todo el trabajo se hace en plantilla). No obstante, esta propuesta se queda en el plano teórico ya que al parecer solo se ha probado en escenarios simulados y no se han hecho estudios empíricos que apliquen la metodología definida por la autora.

### **2.2.2.3 Maślanko (2004)**

Maślanko (2004), por su parte, lleva a cabo un estudio comparativo de los módulos de gestión terminológica integrados en diversos sistemas de memorias de traducción (Multiterm iX de Trados, Déjà vu X de Atril y SDLX 2004 de SDL International). Su objetivo era crear una metodología de evaluación objetiva y detallada que pudiera ser utilizada para la selección de herramientas por parte de traductores y empresas unipersonales de traducción en Polonia, su país de origen. Así pues, según explica, el perfil del evaluador en la metodología propuesta es el de un usuario que tal vez no esté familiarizado con el tipo de herramienta a evaluar.

Con base en su objetivo y su contexto particular, para la evaluación el autor utilizó el tipo de prueba denominado *inspección de características* según la metodología propuesta por EAGLES. La lista de comprobación utilizada se basa sobre todo en los criterios de evaluación propuestos en el informe de la Asociación para la Terminología y la Transferencia del Conocimiento (GTW, por sus siglas en alemán) de evaluación de software de gestión terminológica y en el proceso recomendado por EAGLES. Para la recolección de datos, el autor recurrió a la documentación de los programas, a las páginas web oficiales y en algunos casos fue necesario preguntar directamente a los creadores o distribuidores de los programas. En la tabla 2.2 se presentan las categorías utilizadas por Maślanko (2004) en la lista

de comprobación. Para cada categoría y subcategoría se establecieron una serie de preguntas y sus posibles respuestas, que eran de tipo si/no, valores numéricos, descripciones o listas.

Como resultado de la evaluación, el autor presenta una tabla comparativa con los datos obtenidos para los tres programas evaluados y algunos comentarios aclaratorios, pero no consolida los resultados en puntuaciones globales ni indica preferencias. Así pues, a partir de esos datos factuales la fase de análisis y selección queda como tarea para los traductores interesados en estas herramientas.

<b>Categorías</b>	<b>Subcategorías</b>
Descripción técnica	Requisitos de hardware Requisitos de software
Compatibilidad	
Interfaz de usuario	Rutina de instalación Tipo de interfaz de usuario Idiomas de la interfaz Documentación, formación y ayuda disponibles para el usuario Elementos de la interfaz de usuario Opciones de visualización configurables por el usuario
Aspectos terminológicos	Gestión de datos Modelo de entrada y estructura
Recuperación de información	Acceso a la información Respuesta del sistema Seguridad de la información
Entrada de información	Edición Extracción de términos Validación/control
Intercambio de información	Impresión Importación/exportación
Interacción con otras herramientas	Procesadores de texto Memorias de traducción
Fuentes y juegos de caracteres	
Operaciones de mantenimiento	
Aspectos comerciales	

Tabla 2.2: Categorías utilizadas por Mašlanko (2004) para la evaluación de los módulos de gestión terminológica.

#### **2.2.2.4 Filatova (2010)**

En su tesis doctoral, Filatova (2010) se propone adaptar un modelo científico de evaluación a las necesidades prácticas de los traductores. Este proyecto es más amplio en términos de los tipos de herramientas evaluados, pues abarca no solo herramientas según la autora específicas para traductores (diccionarios electrónicos

multilingües, recuento de palabras y caracteres, análisis de corpus, suites de memorias de traducción) sino también herramientas que clasifica como ofimáticas (compresores de archivos, navegadores web, clientes de correo electrónico, suites de ofimática, lectores PDF y editores de páginas web). Su propuesta metodológica se basa sobre todo en las normas ISO 9126 y 14598 y tiene en cuenta las seis características de calidad (funcionalidad, fiabilidad, usabilidad, mantenibilidad, portabilidad y eficiencia) para evaluar las herramientas cuantitativa y cualitativamente con miras a su integración en el entorno de trabajo del traductor autónomo. En concreto, la autora optó por realizar pruebas funcionales y para la evaluación cuantitativa utilizó las métricas que se presentan en la tabla 2.3.

Característica	Subcaracterística	Métrica aplicable
Usabilidad	Sencillez en operación	Número de comandos claros frente al número total de mensajes revisados
	Satisfacción	Ratio de comentarios positivos a negativos durante el uso
Funcionalidad	Precisión	Número de resultados correctos frente al número total de resultados precalculados
	Adecuación	Ratio de las funciones ejecutables con éxito al total de funciones utilizadas
Fiabilidad	Madurez	Número de errores frente al número total de funciones utilizadas
	Eficiencia	Tiempo de ejecución útil frente al tiempo total de ejecución
Eficiencia	Tiempos de respuesta	Tiempo transcurrido entre el inicio de la acción del sistema y la recepción de la respuesta del sistema
	Eficiencia	Coste de los recursos y del equipo utilizado
Portabilidad	Instalabilidad	Número de intentos correctos frente al número total de intentos de instalación

*Tabla 2.3: Métricas utilizadas por Filatova (2010) para la evaluación cuantitativa de las herramientas para el traductor.*

Con fines ilustrativos, Filatova (2010) aplica su propuesta metodológica a una muestra que comprende tres herramientas de cada una de las tipologías indicadas en el párrafo anterior, para un total de 30 programas (finalmente se evaluaron 28). Como dato destacado, en las tipologías correspondientes a la ofimática las herramientas que obtuvieron mejores resultados, tanto en el análisis cuantitativo como en el cualitativo, fueron casi todas libres (7-Zip, Mozilla Firefox, Mozilla Thunderbird, OpenOffice, Mozilla Kompozer), mientras que de las tipologías

específicas para traductores solo se evaluó una herramienta libre (OmegaT), que no obtuvo los mejores resultados.

En nuestra opinión, tratándose de una evaluación cuantitativa, la descomposición de la calidad que propone la autora resulta poco detallada, ya que solo se tienen en cuenta dos subcaracterísticas de cada característica (una en el caso de la portabilidad) y la evaluación de cada subcaracterística se basa en un solo atributo, para un total de apenas nueve atributos. Como consecuencia, es posible que los resultados obtenidos no presenten una imagen completa de las características de los programas evaluados. Otro problema encontrado es la ausencia de información precisa sobre las versiones de cada programa evaluado, ya que solo se indica que “[s]e emplean las últimas versiones disponibles para el momento de evaluación” (p. 102). Este hecho impide la reproducción del experimento y la verificación de los resultados.

#### ***2.2.2.5 Guillardau (2009)***

El trabajo de Guillardau (2009) es, según él mismo y por lo que se ha podido comprobar, el primer estudio en ocuparse exclusivamente de la evaluación comparativa de sistemas de memorias de traducción libres. El autor se basa en los criterios de calidad propuestos por ISO, EAGLES y en la tesis doctoral de Lagoudaki (2008) para llevar a cabo una comparación de carácter cualitativo de dos sistemas libres (OmegaT y Anaphraseus) en términos de su funcionalidad, eficiencia y usabilidad.

Aunque Guillardau (2009) no propone un modelo de evaluación que se pueda reutilizar para evaluar herramientas de otras tipologías, su trabajo sin duda ayuda a difundir las posibilidades de las tecnologías libres para la traducción y evidencia el creciente interés que estas han generado en los últimos años.

### **2.3 Evaluación de software libre**

Como hemos visto en el apartado 1.1, el modelo de desarrollo y distribución del software libre y de código abierto presenta ciertas características que lo distinguen de los modelos tradicionales del software privativo. Teniendo en cuenta estas



diferencias, en los últimos años han surgido diversas propuestas de modelos de calidad específicos para el software libre que además de evaluar el software como producto abarcan también aspectos relacionados con las comunidades que respaldan los proyectos (Samoladas, Gousios, Spinellis, & Stamelos, 2008).

Stol y Ali Babar (2010) identificaron 20 proyectos de evaluación de software libre surgidos entre los años 2003 y 2009. De estos proyectos, 15 surgen en el entorno académico como parte de proyectos de investigación, 4 surgen de las necesidades de la industria y solo uno cuenta con la participación de la academia y la industria conjuntamente. Según los mismos autores, de los 20 proyectos identificados, solo la mitad especifican las actividades, tareas, materiales y resultados requeridos para ser considerados métodos propiamente dichos, mientras que los demás se limitan a establecer una serie de criterios de evaluación.

Los primeros modelos de calidad específicos, surgidos entre 2003 y 2005, se conocen como modelos de primera generación y parten de los modelos de calidad tradicionales del software privativo, adaptándolos y complementándolos de manera que sean aplicables al software libre (Groven et al., 2011). Los modelos de calidad surgidos a partir de 2006 se conocen como modelos de segunda generación y se basan tanto en los modelos tradicionales como en los de primera generación. En los apartados a continuación abordaremos las particularidades de algunos de estos modelos de calidad específicos para el software libre para pasar luego a considerar otras propuestas de evaluación basadas a su vez en estos modelos.

### **2.3.1 Modelos de calidad de primera generación**

Según Groven et al. (2011), los modelos de calidad de primera generación surgieron entre 2003 y 2005 y tienen como característica común su simplicidad y el hecho de que el proceso de recolección de datos y la evaluación en sí se llevan a cabo de manera manual con la ayuda de plantillas de hojas de cálculo como posible herramienta. Entre los modelos de primera generación se destacan el Open Source Maturity Model (OSMM) desarrollado por Capgemini en el año 2003, el OSMM desarrollado por Navica en el año 2004, el desarrollado por el proyecto

Qualification and Selection of Open Source Software (QSOS) (Atos Origin, 2006), iniciado por la empresa Atos Origin también en el año 2004, y el del proyecto Business Readiness Rating (BRR) (BRR, 2005; Wasserman, Murugan, & Chan, 2006) iniciado por el Carnegie Mellon West Center for Open Source Investigation e Intel, entre otros, en el año 2005 (Groven et al., 2011). Ya que los modelos de Capgemini y Navica fueron utilizados como base para el proyecto BRR no los incluiremos en esta reseña y en cambio nos concentraremos en los proyectos QSOS y BRR.

### ***2.3.1.1 El proyecto Qualification and Selection of Open Source Software (QSOS)***

Iniciado en 2004 por la empresa Atos Origin, una compañía internacional de servicios relacionados con las tecnologías de la información, el proyecto QSOS<sup>26</sup> (cualificación y selección de software libre) tiene por objetivo facilitar a las empresas interesadas en alternativas libres el proceso de evaluación necesario para cualquier adquisición de software, concentrándose en identificar los riesgos específicos de este tipo de software.

El método de evaluación resultante se compone de cuatro fases que comprenden a su vez una serie de actividades. La primera fase comprende la definición de los marcos de referencia necesarios para las demás fases (tipologías de software, clasificación de las licencias libres según sus características y clasificación de las comunidades) y de los criterios de calidad y métricas que se utilizarán. Estos últimos se organizan en dos partes: una parte genérica, que es aplicable a cualquier tipología de software (ver figura 2.3), y una parte específica para cada tipología (ver ejemplo en la tabla 2.4), que corresponde a la característica de funcionalidad en ISO 9126.

En la segunda fase se lleva a cabo la evaluación, es decir, se aplican las métricas definidas. A cada atributo se le asigna una puntuación en una escala de 0 a 2. En el caso de los atributos funcionales, 0 corresponde a función no incluida, 1 a incluida pero limitada y 2 a incluida de manera satisfactoria; para los atributos no funcionales, las puntuaciones 0, 1 y 2 corresponden a tres posibles situaciones (ver

---

26 <http://www.qsos.org>

ejemplo en la tabla 2.5). De esta fase se obtienen como resultado las fichas identificatorias de los programas evaluados (con la información factual de cada uno) y las plantillas con los resultados de la evaluación.

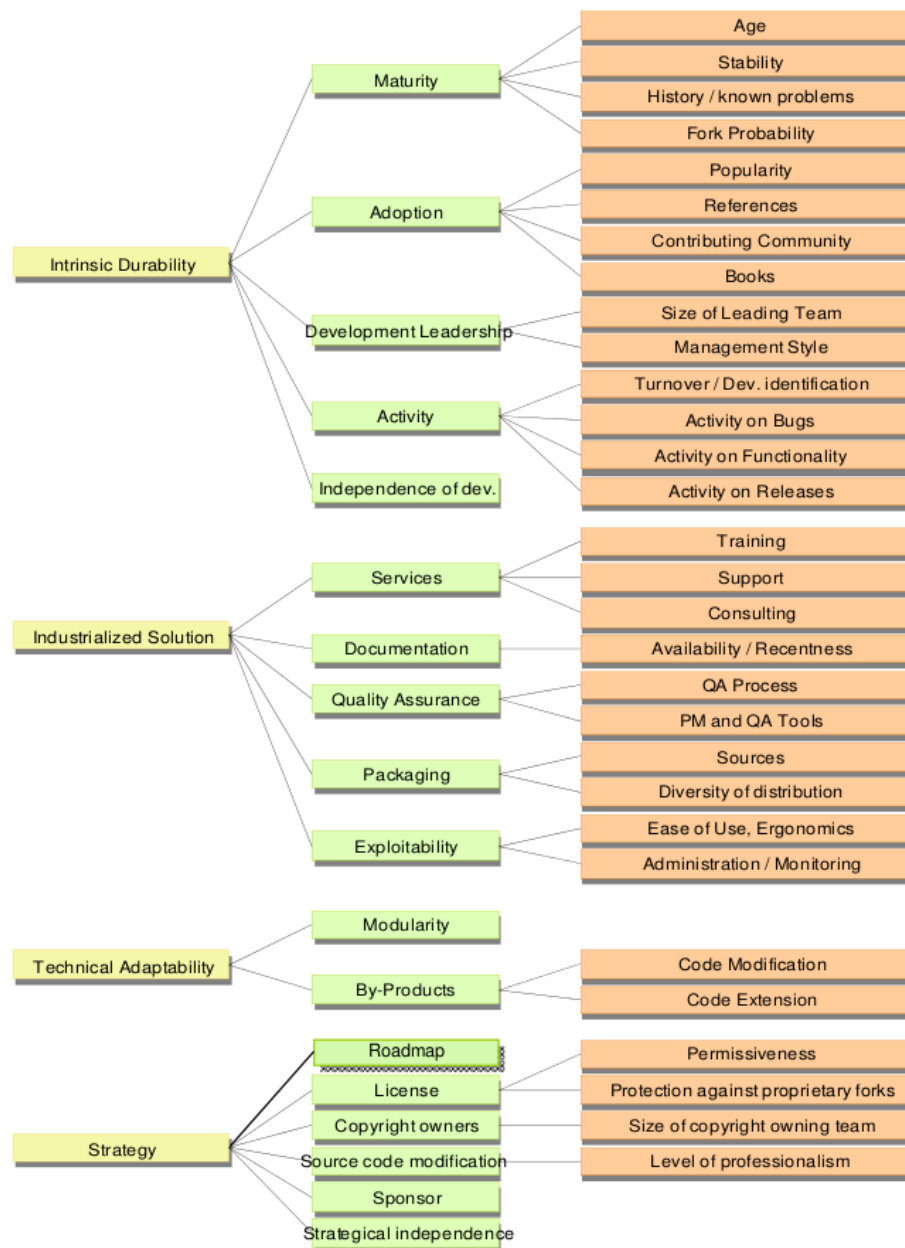


Figura 2.3: Atributos no funcionales del modelo de calidad del proyecto QSOS (fuente: Deprez & Alexandre, 2008, p. 3).

<b>Funcionalidad de los procesadores de texto</b>		
Funcionalidad estándar Funciones básicas de un procesador de texto	General	Integración en una suite ofimática
		Otros componentes de la suite ofimática
		Integración de los componentes
		Internacionalización
		Accesibilidad
		Recuperación de documentos
		Personalización
		Ayuda
	Procesamiento de textos	Control de cambios
		Versiones del documento
		Comparación de documentos
		Buscar y reemplazar
		Navegador de objetos
		Modos de visualización
		Inserción de campos
		Atajos de referencia
		Hiperenlaces
		Comentarios
		Encabezados y pies de página
		Secciones
		Índices
		Gestión de estilos
		Plantillas de documento
		Corrector ortográfico
		Corrector gramatical
		Diccionario integrado
		Tesaurus
Autocompletar		
Autocorregir		
Tablas anidadas		
Autoformato de tablas		
Ordenar tablas y fórmulas		
Conversión texto/tabla		
Seguridad del documento		
Formatos, macros y API Formatos de documento soportados, funcionalidad de macros y API soportadas	Formatos	Formatos soportados
		Soporte para el formato OpenDocument
		Exportación a PDF
		Exportación a DocBook
		Exportación a LaTeX
		Soporte para XSLT
	Macros y API	Soporte para macros

		Lenguajes de macros soportados
		Grabación de macros
		IDE
		Utilización de macros
		Seguridad de las macros
		API soportadas
Funcionalidad avanzada	Envío de correo	Motor para el envío de correo
		Fuentes de datos
		Envío de imágenes
		Fórmulas matemáticas
		Vistas de documentos
		Copiar y pegar estilos
		Texto condicional
		Composición de documentos
		Herramientas de dibujo
		Funciones de trabajo colaborativo

Tabla 2.4: Atributos funcionales para evaluar los procesadores de texto según el proyecto QSOS (fuente: <http://www.qsos.org>)

Solución industrializada		Puntuación		
		0	1	2
Servicios	Formación	No hay oferta de formación	Existe oferta pero está restringida geográficamente y a un solo idioma o es proporcionada por un único proveedor	Existe una amplia oferta de servicios proporcionados por varios proveedores, en varios idiomas y organizada en módulos de niveles progresivos
	Asistencia técnica	No hay oferta de asistencia técnica aparte de los foros públicos y listas de correo	Existe oferta pero es proporcionada por un único proveedor sin compromiso de calidad del servicio	Varios proveedores del servicio con un compromiso claro de calidad (p. ej. tiempo de respuesta garantizado)
	Consultoría	No hay oferta de servicios de consultoría	Existe oferta pero está restringida geográficamente y a un solo idioma o es proporcionada por un único proveedor	Servicio de consultoría proporcionado por diferentes proveedores en varios idiomas

Tabla 2.5: Posibles puntuaciones para un atributo no funcional según la propuesta del proyecto QSOS (fuente: Atos Origin, 2006, p. 19).

En la tercera fase, denominada *cualificación*, entran en consideración el contexto y las necesidades específicas del usuario. Para esto, se definen ciertos filtros según las características de las fichas identificatorias de los programas y

según la relevancia de los atributos, funcionales y no funcionales, en el contexto en cuestión. En el caso de los atributos funcionales, para cada función se debe indicar si se trata de una función requerida, opcional o no requerida, mientras que para los atributos no funcionales se debe indicar si se trata de un atributo irrelevante, relevante o decisivo.

Para terminar, en la etapa de selección se asignan valores numéricos a las preferencias expresadas en la fase de cualificación y se obtienen puntuaciones totales para los programas. Para seleccionar entre herramientas similares, el método propone dos modos. Si se opta por el modo estricto, las herramientas se descartan en cuanto no cumplan alguna de las preferencias indicadas en la fase de cualificación, empezando con características de las fichas identificatorias (por ejemplo, según los sistemas operativos soportados), siguiendo con los atributos funcionales y terminando con los no funcionales.

En el modo menos estricto, por otro lado, no se descartan los programas, sino que se comparan todas las alternativas asignando pesos a las preferencias según el contexto de uso de la siguiente manera: función requerida +3, función opcional +1 y función no requerida 0; atributo irrelevante 0, atributo relevante +1 o -1, y atributo decisivo +3 o -3. La selección se realiza con base en la puntuación total ponderada.

El proyecto QSOS se basa en una herramienta web y tanto las fichas de los programas como los resultados de evaluación están disponibles en un repositorio público disponible en línea. El proyecto sigue activo y ha lanzado recientemente (25 de octubre de 2011) la versión 1.7 del modelo. En la actualidad, están preparando la versión 2.0.

### **2.3.1.2 El proyecto *Business Readiness Rating (BRR)***

El proyecto Business Readiness Rating (BRR)<sup>27</sup> fue iniciado por el Carnegie Mellon West Center for Open Source Investigation e Intel, entre otros, en el año 2005. Este proyecto propone un modelo estándar y abierto de evaluación que facilite el proceso de toma de decisiones a la hora de adoptar software libre en una organización. El

---

<sup>27</sup> <http://openbr.org>

objetivo del proyecto era lograr un modelo completo, simple, adaptable y consistente, basándose en la orientación funcional del software, es decir, la combinación de dos aspectos: el tipo de aplicación a evaluar (navegador web, paquete ofimático, etc.) y el entorno de uso. Este último aspecto implica determinar el papel que tendrá el programa según su uso previsto en la organización: imprescindible, normal, de desarrollo interno o de experimentación, conceptos que se explicarán a continuación.

Según el proyecto BRR, se habla de un papel *imprescindible* cuando el software debe estar en constante funcionamiento y el flujo normal de trabajo de la compañía depende de este o cuando el software hace parte de los productos o servicios ofrecidos por la compañía. Un entorno de uso *normal* es aquel en el que el software es utilizado solo internamente en la compañía y la implementación de soluciones provisionales para los problemas no tiene mayores consecuencias. Un entorno de uso *de desarrollo interno* implica que dentro de la compañía existe un grupo de desarrolladores que evalúan el software con miras a integrarlo en los sistemas utilizados o incluirlo en los productos o servicios ofrecidos, lo que implica que posiblemente se modificará el código y el grupo en cuestión será responsable de ofrecer soporte. Finalmente, un entorno de uso *de experimentación* es aquel en el que se investiga el software con fines comparativos o para una futura implementación, sin que haya objetivos de productividad inmediatos, si bien tal investigación puede dar como resultado la implementación del software en uno de los entornos de uso anteriormente mencionados.

La primera fase del proceso propuesto por el proyecto BRR consiste en una preselección que permite reducir la lista de programas descartando los que no son viables. Para esto se recomienda utilizar una serie de indicadores de viabilidad que puede incluir, entre otros, aspectos como el tipo de licencia, la posibilidad de obtener soporte de una organización estable, idioma(s) de implementación, los usuarios referenciables que utilicen el programa, la posibilidad de internacionalización y localización del programa al idioma deseado, la existencia de

reseñas de terceros sobre el programa y el hecho de que analistas de la industria, como Gartner o IDC<sup>28</sup>, hagan un seguimiento del programa.

Esta lista de criterios, sin embargo, no es exhaustiva y los autores recomiendan modificar los indicadores según el contexto específico. Una vez se haya determinado el uso que se le dará a la aplicación y decidido la lista completa de indicadores de viabilidad, es necesario establecer una política de resultados aceptables para pasar luego a determinar qué programas satisfacen los criterios para ser incluidos en la evaluación detallada.

Para la evaluación como tal, el proyecto BRR propone un modelo de atributos ponderados, en el que a cada atributo se le asigna un peso según su importancia. Cada programa recibe una calificación para cada uno de los atributos predefinidos y los valores obtenidos se multiplican por el porcentaje de importancia asignado a cada atributo para obtener así la puntuación final. Para calificar cada atributo, se establece una serie de subatributos a los que también se asigna un peso para su ponderación.

El proyecto en cuestión propone una lista de doce criterios, de los que recomiendan escoger un máximo de siete según las necesidades específicas en el contexto de evaluación. Los doce criterios que proponen son: funcionalidad, usabilidad, calidad, seguridad, desempeño, escalabilidad, arquitectura, soporte, documentación, adopción, comunidad y profesionalismo. Para decidir cuáles escoger, los autores recomiendan ordenarlos por prioridades y distribuir los pesos entre los criterios seleccionados de manera que sumen 100.

El proyecto BRR propone además normalizar las métricas para todos los criterios, excepto la funcionalidad, que se computa de manera diferente. Dicha normalización implica utilizar una escala de 1 a 5 (que se traduce verbalmente a las categorías de inaceptable, pobre, aceptable, muy bueno y excelente) como correspondencia de los posibles valores para un determinado aspecto. Así por

---

28 Las empresas Gartner (<http://gartner.com>) e IDC [*International Data Corporation*] (<http://idc.com>) son líderes mundiales en investigación, consultoría y organización de eventos en el campo de las tecnologías de la información. Los resultados de sus estudios sirven de referente a miles de clientes y consumidores en el mundo a la hora de tomar decisiones estratégicas relacionadas con las tecnologías.



ejemplo, para puntuar la edad de un proyecto se podría utilizar una equivalencia del tipo:

- 1 año o menos – inaceptable (1)
- 1-2 años – pobre (2)
- 2-3 años – aceptable (3)
- 3-4 años – muy bueno (4)
- más de 4 años – excelente (5)

Para las métricas que requieran un valor binario (si o no), a la respuesta negativa se la asigna un 1 y a la afirmativa un 3 o un 5 según la diferencia que a juicio del evaluador implique la presencia o ausencia del aspecto en cuestión. La figura 2.4 a continuación muestra un ejemplo tomado de la plantilla disponible en la página web del proyecto.

<b>Metric Name</b>	
Is there a dedicated information (web page, wiki, etc) for security?	
<b>Test Description</b>	
This measures how aware of and seriously the project takes security issues.	
<b>Test Score Specification</b>	<b>Score</b>
Yes, well maintained	5
Yes	3
No	1

*Figura 2.4: Ejemplo de puntuaciones posibles para un atributo binario tomado de la plantilla del proyecto BRR.*

Para las métricas que se puedan medir en un rango de posibilidades que no sean fácilmente traducibles a los 5 valores de la escala normalizada los autores recomiendan dejar algunos valores sin utilizar. La figura 2.5 muestra un ejemplo tomado de la misma plantilla.

Para evaluar el criterio de funcionalidad, el proyecto BRR propone la utilización de una lista de comprobación de características estándar según el tipo de software. Tal lista se puede diseñar específicamente para la evaluación según las necesidades del usuario o bien se puede adaptar alguna lista existente. En primer término, a todos los ítems de la lista de comprobación se les asigna un valor en una escala de 1 a 3 según su importancia. A continuación, se comparan las características del programa evaluado con la lista de comprobación. Para las características que estén presentes se suma el valor de su importancia al acumulado,

mientras que para las que no estén presentes se resta el valor. Si el programa tiene características adicionales que no estén incluidas en la lista de comprobación también se les asigna un valor de importancia y se suma al acumulado. A continuación, se divide el resultado obtenido entre la puntuación máxima posible según la lista de funciones básicas. El valor que se obtiene está dado en porcentaje y puede llegar a ser mayor que 100 para programas ricos en funciones adicionales o menor que 0 para programas que carecen de las funciones básicas. Finalmente, el porcentaje obtenido se traduce a una escala de 1 a 5 utilizando las siguientes correspondencias:

menos de 65% = 1 (inaceptable)  
 65%-80% = 2 (pobre)  
 80%-90% = 3 (aceptable)  
 90%-96% = 4 (muy bueno)  
 más de 96% = 5 (excelente)

**Metric Name**

Difficulty to enter the core developer team

**Test Description**

To ensure software quality, mature projects must be selective in accepting committers. New projects often has no choice

**Test Score Specification**

	<b>Score</b>
Only after being active outside committer for a while	5
Rather difficult, must contribute accepted patches for some time	3
Anyone can enter	1

*Figura 2.5: Ejemplo de puntuaciones propuestas para una atributo que permite un rango pero no utiliza los 5 valores de la escala.*

### 2.3.2 Modelos de calidad de segunda generación

Los modelos de calidad de software libre surgidos a partir de 2006 se caracterizan por estar basados tanto en los modelos tradicionales del software privativo como en los modelos de primera generación; además, están más enfocados a la automatización del proceso de evaluación y a proporcionar métricas más avanzadas y herramientas para la evaluación disponibles como aplicaciones web o plugins para los entornos de desarrollo (Groven et al., 2011).

Entre los modelos de calidad de segunda generación más conocidos encontramos los desarrollados por los proyectos Quality in Open Source Software (QualOSS) (Ciolkowski & Soto, 2008; Deprez, 2009; Izquierdo-Cortazar,

González-Barahona, Robles, Deprez, & Auvray, 2010; Izquierdo Cortázar, Robles, González Barahona, & Deprez, 2009; Soto et al., 2009), Quality Platform for Open Source Software (QualiPSo) (Petrinja et al., 2008; Wittmann & Nambakam, 2010) y Software Quality Observatory for Open Source Software (SQO-OSS) (Samoladas et al., 2008; Spinellis et al., 2009), todos ellos con financiación de la Comunidad Europea (Deprez & Alexandre, 2008). A continuación nos limitaremos a reseñar estas dos últimas propuestas, ya que la primera de ellas (QualOSS) se basa en gran medida en las características de calidad del producto establecidas en el estándar 9126 y en los proyectos QSOS y BRR antes reseñados (Ciolkowski & Soto, 2008).

### ***2.3.2.1 El proyecto Quality Platform for Open Source Software (QualiPSo)***

El proyecto QualiPSo<sup>29</sup> fue desarrollado por un consorcio de empresas, entidades gubernamentales y universidades de Europa, Brasil y China con el objetivo de promover una mayor adopción de software libre en todos los sectores. El proyecto, que cuenta con financiación de la Unión Europea, trabaja en siete áreas: simplificación de los aspectos legales, definición de modelos de negocio viables, mejora de la interoperabilidad, mejora de la gestión de la información, mejora de la calidad y confiabilidad de los procesos, creación de forjas de última generación y creación de centros especializados de apoyo al desarrollo y la adopción de software libre.

De estas siete líneas de trabajo nos interesa la de mejora de la calidad y confiabilidad, que se centra en definir un modelo de calidad con base en encuestas a desarrolladores y empresas europeas sobre sus percepciones sobre el software libre y en el Capability Maturity Model Integration (CMMI), un modelo tradicional de calidad de los procesos. Su propuesta se centra, pues, en la calidad y mejora de los procesos y solo de manera indirecta en la calidad del producto resultante. El objetivo es servir de ayuda a creadores y comunidades de software libre y a las compañías que integran programas libres en sus productos a aumentar su nivel de confiabilidad y a la vez proporcionar a los usuarios finales una manera objetiva de evaluar a los proveedores de software libre.

---

<sup>29</sup> <http://www.qualipso.org>

Para su modelo de madurez, denominado Open Source Maturity Model (OMM), identificaron 12 características de calidad, o “elementos de confianza”, que según las comunidades y empresas encuestadas tienen el siguiente orden de importancia (Petrinja et al., 2008, p. 24):

- Calidad de la documentación: exhaustiva y fácil de entender
- Popularidad del producto
- Utilización de estándares establecidos y ampliamente reconocidos
- Disponibilidad y uso de una hoja de ruta
- Calidad del plan de pruebas
- Relación entre las partes vinculadas al proyecto (usuarios, desarrolladores, etc.)
- Existencia de un proceso de toma de decisiones para seleccionar las licencias
- Existencia de una serie de herramientas de desarrollo bien establecidas
- Vitalidad del proceso de desarrollo e idoneidad del proceso de corrección de fallos
- Reforzar la mantenibilidad y estabilidad del producto/proceso
- Contribución al producto por parte de compañías de software
- Disponibilidad de resultados de evaluación relacionados con el proceso

En relación con los perfiles de usuario, en las encuestas a empresas europeas los autores encontraron que:

los entrevistados interesados solo en el uso de la herramienta libre no estaban interesados en el proceso de desarrollo (aproximadamente la mitad de la muestra encuestada) ... [mientras que] los usuarios que también planean mejorar la herramienta o integrarla con otros componentes de software estaban interesados en el proceso de desarrollo seguido por la comunidad libre y en su documentación (Wittmann & Nambakam, 2010, p. 7, traducción propia).

Para definir las métricas para cada uno de estos elementos los autores utilizaron una técnica conocida como objetivo-pregunta-métrica (GQM, por sus siglas en inglés) que permite “cuantificar los objetivos relevantes para los proyectos de desarrollo de software” (Petrinja et al., 2008, p. 12) (ver esquema en la figura 2.6). El proceso empieza en el plano conceptual con la definición de los objetivos que se espera alcanzar con respecto a cada uno de los elementos identificados. A continuación se pasa al plano operativo formulando para cada objetivo una serie de preguntas cuyas respuestas permiten determinar si se ha logrado el objetivo en cuestión. Finalmente, para cada una de las preguntas formuladas se identifican los datos asociados que permitan obtener un resultado cuantitativo, teniendo en cuenta que los datos en sí pueden ser tanto objetivos, cuando solo dependen del objeto que

se mide, como subjetivos, cuando se incluye también el punto de vista de alguien (Petrinja et al., 2008, p. 14).

Para definir el modelo OMM se hizo énfasis en lograr un modelo simple y fácil de usar, procurando evitar la complejidad inherente al modelo CMMI. Los autores organizaron los elementos de confianza en tres niveles: básico, intermedio y avanzado. Así, para alcanzar el nivel de madurez avanzado es necesario que el proyecto cumpla todos los elementos correspondientes a los niveles inferiores y por lo menos el 90 % de los elementos correspondientes al nivel avanzado.

El proceso de evaluación propuesto por el proyecto QualiPSo está dividido en tres fases. En la fase de planeación se seleccionan los proyectos a evaluar y se prepara un protocolo donde se definen las responsabilidades de los participantes y el alcance de la evaluación.

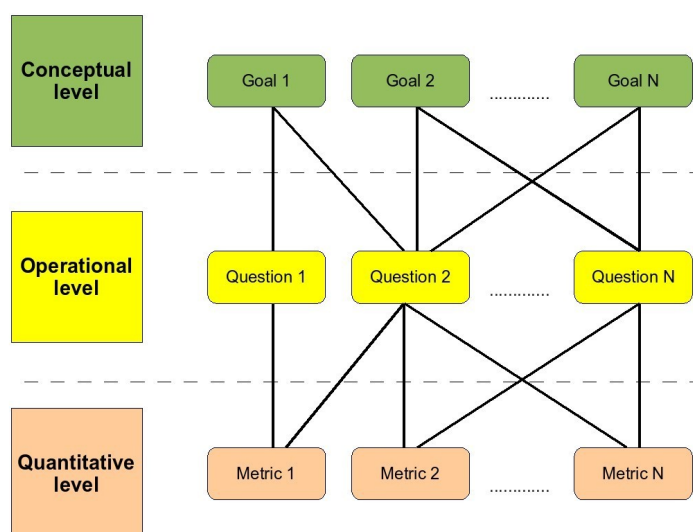


Figura 2.6: Esquema de la metodología GQM (objetivo-pregunta-métrica) (fuente: Petrinja et al., 2008, p. 13).

La segunda fase corresponde a la aplicación del cuestionario de evaluación y la obtención de las puntuaciones. La aplicación del cuestionario se puede hacer de forma manual o mediante una herramienta denominada Spago4Q. Para asignar las puntuaciones para cada métrica se utilizó una escala de 0 a 3, donde 0 corresponde a no aplicable, 1 a no implementado, 2 a parcialmente implementado y 3 a

totalmente implementado. Para consolidar las puntuaciones se calcularon promedios.

La tercera y última etapa corresponde a la presentación de resultados. En el informe final se presentan las puntuaciones para cada elemento de confianza y la puntuación total y se incluye además un informe de debilidades y fortalezas del proyecto.

### ***2.3.2.2 El proyecto Software Quality Observatory for Open Source Software (SQO-OSS)***

El proyecto Software Quality Observatory for Open Source Software (SQO-OSS)<sup>30</sup> fue creado por un consorcio de proyectos de código abierto, empresas de consultoría y universidades de Grecia, Alemania, Reino Unido y Suecia bajo la coordinación de la Athens University of Economics and Business y financiado parcialmente por el VI Programa Marco de Investigación de la Unión Europea.

Según explican Samoladas et al. (2008, p. 3), un aspecto destacado del proyecto SQO-OSS es que se pretendía desarrollar un “sistema de monitoreo continuo de calidad” para el software libre. El objetivo principal de este proyecto era lograr la automatización del proceso de evaluación obteniendo los datos directamente desde las plataformas web donde se gestiona el desarrollo de los proyectos libres. Así pues, seleccionaron atributos que se pudieran medir con mínima intervención humana y no tuvieron en cuenta ni la usabilidad ni la funcionalidad.

En esta propuesta se evalúa tanto el producto como la comunidad. Para la evaluación del producto se centran en el análisis del código fuente, mientras que para la evaluación de la comunidad solo consideran atributos que se puedan medir directamente (como el número de suscriptores en la lista de correo o el número de mensajes por mes).

El proceso de evaluación definido se divide en dos fases, cada una con dos pasos. En la primera fase se define el modelo de calidad, lo que implica definir los

<sup>30</sup> La página web del proyecto (<http://www.sqo-oss.eu>) ya no está disponible, pero aún se puede descargar la presentación del proyecto desde el siguiente enlace:  
[ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate\\_d/st-ds/sqo-oss-project-story\\_en.pdf](ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/st-ds/sqo-oss-project-story_en.pdf).

atributos y subatributos a considerar (ver figura 2.7) y las métricas correspondientes. Para esta fase se utiliza una versión simplificada de la metodología GQM (objetivo-pregunta-métrica) (ver figura 2.6 en la página 93).

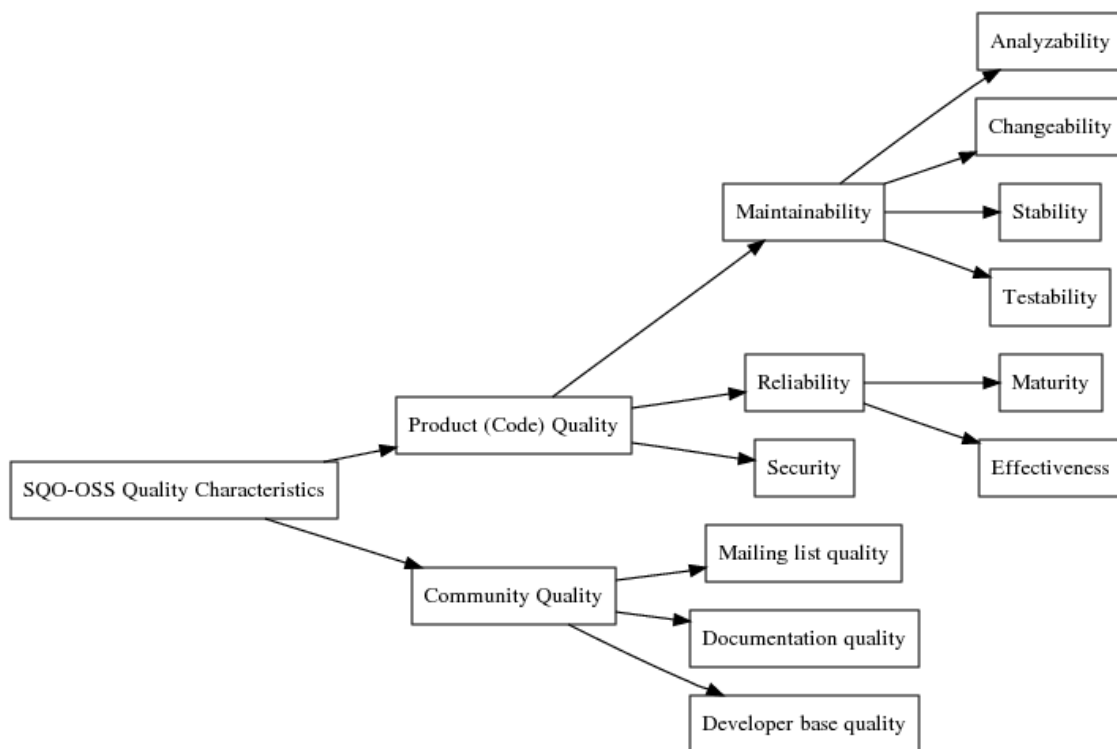


Figura 2.7: Modelo de calidad propuesto por el observatorio de calidad del software libre (SQO-OSS).

En la segunda fase se define el proceso para computar una valoración general a partir de los valores obtenidos por atributo. Para esto, en lugar de utilizar promedios ponderados según importancia como la mayoría de propuestas de evaluación, los autores decidieron utilizar una escala ordinal de cuatro categorías (excelente > bueno > aceptable > malo). Aunque el uso de pesos para ponderar la importancia de los atributos también es posible, los autores no lo recomiendan pues asumen que todas las métricas tienen igual importancia.

Los autores definieron entonces las puntuaciones necesarias para categorizar un programa en los tres primeros perfiles: excelente (E), bueno (B) y aceptable (A) (ver tabla 2.6), de modo que si el valor obtenido para un atributo no corresponde a ninguno de esos tres perfiles se entiende que el resultado es malo. En esta fase

puede intervenir el usuario final modificando los umbrales de los perfiles según sus necesidades y contexto de uso.

<b>Criterio compuesto</b>	<b>Criterio</b>	<b>Perfil E</b>	<b>Perfil B</b>	<b>Perfil A</b>	<b>Escala</b>
<b>Analizabilidad</b>	Número ciclomático	4	6	8	Menor es mejor
	Número de declaraciones	10	25	50	Menor es mejor
	Frecuencia de los comentarios	0,5	0,3	0,1	Mayor es mejor
	Longitud promedio de las afirmaciones	2	3	4	Menor es mejor

Tabla 2.6: Perfiles para los atributos del criterio de Analizabilidad en el proyecto SQO-OSS.

La propuesta inicial fue revisada por desarrolladores y usuarios de software libre, académicos y empresas especializadas en desarrollo y soporte de software libre, que aportaron comentarios para mejorarla.

### 2.3.3 Otras propuestas

En los apartados a continuación veremos otras propuestas de evaluación de software libre surgidas en los últimos años.

#### 2.3.3.1 Wheeler (2008)

Wheeler (2008) propone un proceso general de evaluación que aunque es aplicable también a software privativo ofrece información específica para la evaluación de software libre y de código abierto. Aunque su propuesta no constituye un método por ser menos detallada en cuanto a los procedimientos para la evaluación y no proponer un sistema formal para puntuar los programas, proporciona indicaciones sobre los aspectos a los que se debe prestar más atención cuando se trata de software libre.

El proceso de evaluación que propone consta de cuatro pasos. Primero, se deben identificar los candidatos, recurriendo principalmente a búsquedas en Internet pero sin dejar de lado otras fuentes. Como segundo paso, el autor recomienda leer las reseñas existentes y de ser posible preguntar a quienes tengan experiencia con los programas. Tercero, comparar los atributos básicos de los programas con nuestras necesidades para así reducir la lista de candidatos a evaluar. Para este paso



el autor recomienda concentrarse en los siguientes aspectos: funcionalidad, coste, cuota del mercado, soporte, mantenimiento/longevidad, fiabilidad, desempeño, escalabilidad, usabilidad, seguridad, flexibilidad/adaptabilidad, interoperabilidad y aspectos legales/de licencias. En la tabla 2.7 se presentan estos criterios con un resumen de los factores que para cada uno de ellos el autor considera especialmente relevantes para el software libre. Como último paso, se realiza el análisis detallado de los mejores candidatos, con especial atención a la facilidad para añadir funciones y a la seguridad del programa, para lo cual es necesario analizar el código fuente.

<b>Criterio</b>	<b>Aspectos a tener en cuenta cuando se evalúa software libre</b>
Funcionalidad	Considerar los sistemas operativos soportados y otros requisitos de sistema. Considerar la ventaja que representa la posibilidad de modificar el código para añadir funcionalidad según necesidades específicas.
Coste	Considerar todos los costes asociados a la implementación (instalación, actualizaciones, personal, capacitación, soporte y mantenimiento, etc.), no solo el de la licencia.
Cuota del mercado	Es difícil de valorar para los programas libres. Se pueden considerar el número de descargas o índices de popularidad (ofrecidos por repositorios como SourceForge), el número de páginas con enlaces al proyecto o la inclusión del programa en cuestión en distribuciones de GNU/Linux.
Soporte	Constituye una de las diferencias más importantes entre software libre y privativo. Tener en cuenta cuáles son las opciones disponibles y valorar las listas de distribución asociadas al proyecto.
Mantenimiento/longevidad	Examinar las listas de distribución de los desarrolladores, el sistema para reportar problemas y la existencia de <i>wikis</i> . Tener en cuenta que la disponibilidad del código fuente permite 'revivir' un proyecto inactivo si el usuario lo considera apropiado para sus necesidades.
Fiabilidad	Considerar el estado de desarrollo del proyecto según los desarrolladores. Determinar si el programa está incluido en alguna distribución de GNU/Linux. Si se planea modificar el código, considerar si hay algún paquete de pruebas ( <i>test suite</i> ) disponible con el programa.
Desempeño	No es recomendable confiar únicamente en la información presentada en las páginas del proyecto; es mejor hacer pruebas específicas según el contexto específico de uso previsto.
Escalabilidad	Se refiere a la cantidad de datos que puede manejar el software. Considerar la disponibilidad de un paquete de pruebas específico con el programa.
Usabilidad	Considerar el tipo de interfaz (línea de comandos, gráfica, basada en web).
Seguridad	Considerar evaluaciones independientes existentes.
Flexibilidad/adaptabilidad	Una de las ventajas más significativas del software libre frente al software privativo. Si se piensa hacer modificaciones al código, considerar si existen plantillas, complementos, extensiones o API (interfaces de programación de aplicaciones) para programadores.
Interoperabilidad	Los programas libres por lo general implementan estándares abiertos. Sin embargo, se debe verificar según los estándares que sean relevantes.
Aspectos legales/de licencias	Esta es la diferencia fundamental entre el software libre y el privativo. Verificar si la licencia está en la lista de licencias de la OSI o la FSF. Si se piensa hacer modificaciones al código se debe considerar si la licencia incluye cláusulas de <i>copyleft</i> o licencias especiales para las bibliotecas en las que se basa el programa.

*Tabla 2.7: Aspectos específicos a tener en cuenta para la evaluación de software libre según Wheeler (2008).*

### **2.3.3.2 Van den Berg (2005)**

Van den Berg (2005), por otra parte, basa su modelo de evaluación en las diferentes propuestas que encontró en la literatura sobre evaluación de software libre, entre las que se incluyen los modelos de calidad de Capgemini y Navicasoft y la versión de 2005 de las recomendaciones de Wheeler. Su propuesta implica dos pasos: la selección de candidatos y la evaluación detallada. En primer término, para elaborar la lista inicial de candidatos la autora sugiere recurrir a portales de software libre como SourceForge o Freshmeat, así como a búsquedas en Internet, y consultas a desarrolladores de software libre, listas de correos relacionadas y distribuidores de software. A continuación, para determinar los programas que se evaluarán en detalle la autora propone un método de selección que combina dos modelos. Por un lado, el modelo de “eliminación por aspectos”, que consiste en determinar unos criterios mínimos que deben cumplir los programas. Estos criterios se organizan en orden descendente según su importancia y se determina un valor mínimo aceptable para cada uno de ellos. Así, los programas que no obtengan el valor mínimo para el primer criterio son descartados. Los programas restantes se califican para el segundo criterio y así sucesivamente se van eliminando los programas que no obtengan el valor mínimo para el criterio en cuestión.

El segundo modelo, denominado modelo linear de atributos ponderados, también aplicable para la evaluación en sí, consiste en utilizar una lista de criterios a los que se les asignan pesos según su importancia relativa. Cada programa es entonces calificado según estos criterios, las puntuaciones se multiplican por los pesos y los valores resultantes se suman para obtener un puntuación ponderada para cada programa.

Según van den Berg (2005), la combinación de estos dos modelos permite, por una parte, ahorrar tiempo en el proceso de selección ya que los programas eliminados no tienen que ser puntuados para el resto de los criterios y, por otra parte, obtener una lista reducida que se podrá organizar según las puntuaciones obtenidas con el modelo de atributos ponderados. Los criterios para esta etapa de

selección deben permitir hacerse una impresión general de los programas y a la vez ser fáciles de medir.

La autora llevó a cabo un estudio de caso con software libre para gestión de cursos en el que utilizó los criterios de funcionalidad y actividad de lanzamientos para la eliminación por aspectos; los programas que cumplieron los requisitos mínimos para estos dos criterios fueron calificados con el modelo linear de atributos ponderados para dichos criterios más los de comunidad y longevidad. Los dos primeros programas del listado resultante fueron entonces sometidos a la evaluación detallada.

Como criterios para la evaluación, van den Berg (2005) seleccionó nueve atributos que habían sido utilizados en por lo menos cuatro de las fuentes sobre software libre que analizó, si bien en algunos casos aparecían bajo nombres diferentes. Los criterios adoptados fueron comunidad, actividad de lanzamiento, longevidad, licencia, asistencia técnica, documentación, seguridad, funcionalidad e integración.

<b>Criterio</b>	<b>Explicación</b>
<b>Comunidad</b>	La comunidad de usuarios y desarrolladores constituye el motor y principal recurso de los proyectos de software libre. Se consideran el número de usuarios y desarrolladores, el número de mensajes en las listas (por periodo y en total), la velocidad y calidad de las respuestas y el tipo de contribuciones de la comunidad (código, documentación, etc.)
<b>Actividad de lanzamiento</b>	El lanzamiento de nuevas versiones del programa es un indicador del nivel de actividad del proyecto y del progreso del mismo. Se consideran los lanzamientos de nuevas versiones por periodo de tiempo, así como el <i>roadmap</i> , o guía del desarrollo proyectado para el programa.
<b>Longevidad</b>	La longevidad de un proyecto constituye un indicador de su estabilidad y oportunidades de supervivencia. Se consideran aspectos como la edad del proyecto (según fecha del primer lanzamiento), el número de versión y si el programa ha pasado por ciclos de rediseño.
<b>Licencia</b>	El tipo de licencia refleja la filosofía detrás del proyecto e indica las condiciones en las que se puede usar, distribuir y modificar el programa. Se debe considerar si es una licencia pública (aprobada por la OSI o la FSF) y si se ajusta al objetivo del programa.
<b>Asistencia técnica</b>	El tipo de asistencia técnica disponible constituye un indicador de la seriedad de los desarrolladores frente al proyecto. Se consideran las opciones disponibles, que pueden incluir asistencia ofrecida por la comunidad y asistencia de pago ofrecida por el distribuidor o un tercero.
<b>Documentación</b>	La existencia de un buen paquete de documentación facilita la puesta en funcionamiento del programa. Se consideran aspectos como la existencia de diferentes tipos de guías (instalación, mantenimiento, guía de usuario, guía de desarrollador, solución de problemas, etc.), su calidad y actualización.

<b>Seguridad</b>	La seguridad también es un indicador de la seriedad de los desarrolladores. Se considera la existencia de advertencias de seguridad, de un sistema para reportar problemas ( <i>bugtracker</i> ) y el tiempo de respuesta para resolver los agujeros de seguridad.
<b>Funcionalidad</b>	La funcionalidad del programa permite determinar si este se ajusta a las necesidades del usuario. Se considera la lista de funciones incluidas comparada con las necesidades y se considera también la existencia de funciones adicionales.
<b>Integración</b>	La integración es importante a la hora de implementar un programa en un entorno específico. Se consideran aspectos como la modularidad del diseño, el uso de estándares abiertos, la colaboración con otros productos y los requisitos de software, incluyendo los sistemas operativos soportados.

Tabla 2.8: Resumen de los criterios utilizados por van den Berg (2005) para la evaluación de software libre.

La tabla 2.8 presenta los nueve criterios con la explicación de los aspectos que se consideraron para cada uno de ellos. Además de estos nueve criterios, la autora incluyó también, si bien con menor peso, la motivación y origen del proyecto como indicador del uso previsto del programa y de la seriedad de sus desarrolladores. En la etapa de la evaluación detallada, la autora también utilizó el modelo lineal de atributos ponderados para calcular las puntuaciones.

### ***2.3.3.3 La Certificación de calidad del Centro de Excelencia de Software Libre de Castilla-La Mancha***

En el contexto español, encontramos la metodología de evaluación propuesta por el Centro de Excelencia de Software Libre de Castilla-La Mancha (CESLCAM) en su proyecto de certificación de software libre OPSOA (OPen SOurce Assessment) (Navarro, Montero, & JCCM, 2009). Este proyecto surge con el objetivo de “apoyar a través de un 'SELLO DE CONFIANZA', el despliegue total del Software Libre o FOSS (Free and Open Source Software), generando la confianza necesaria para que los usuarios aprovechen las ventajas que ofrece este software” (CESLCAM, s. f.-a, p. 3). Como beneficios de la acreditación de calidad los autores destacan la mejora de los procesos de desarrollo, la mejora de la calidad del producto y el aumento de confianza de los usuarios.

Antes de empezar el proceso de certificación del producto, CESLCAM propone aplicar una metodología de análisis de confianza para los proyectos de software libre. Para este análisis se consideran tres aspectos, en el siguiente orden: la gestión y mantenimiento del proyecto, su madurez y los recursos y servicios disponibles para los usuarios del producto (CESLCAM, 2011).

Para cada sección se utiliza un cuestionario compuesto por preguntas de tipo si/no/no disponible y se definen de dos a cuatro items que se deben responder afirmativamente para continuar con el siguiente cuestionario; si el proyecto no cumple con esos requisitos básicos finaliza el análisis, se descarta el proyecto como *No confiable* y se genera el informe de resultados.

En el cuestionario sobre la gestión y mantenimiento del proyecto se incluyen 15 preguntas sobre la existencia y características de la comunidad que respalda el proyecto y las tecnologías en las que se basa la gestión. Como requisitos para continuar el análisis, el proyecto debe contar con una comunidad activa que publique al menos una versión del producto por año y con un sitio web donde se recoja la información destacada; además, el código fuente debe estar disponible públicamente bajo una licencia libre.

A continuación, en el cuestionario sobre la madurez del proyecto se evalúan aspectos como el tiempo de vida del proyecto, el número de versiones estables, el modelo de negocio, los idiomas disponibles y la gestión de los diferentes artefactos del proyecto (código fuente, foros de usuarios, informes de errores, etc.). Como requisitos básicos, el proyecto debe tener más de dos años y debe haber lanzado por lo menos tres versiones estables del producto.

En el tercer y último cuestionario sobre el proyecto se analizan los recursos y servicios disponibles para los usuarios. Como requisitos básicos para certificar el proyecto, se especifica que debe existir algún tipo de documentación creada por la comunidad (por ejemplo una *wiki*, manuales de uso o una lista de preguntas frecuentes) y deben existir varias formas de instalar o ejecutar la aplicación (código fuente, autoinstalables, software como servicio, etc.).

Si el proyecto cumple los requisitos básicos de las tres secciones se genera el informe como proyecto *Confiable* y se suman las puntuaciones de todas las preguntas para obtener una puntuación final de confianza. El informe de resultados puede incluir también el resultado de la evaluación del producto, si esta se lleva a cabo.

La certificación del producto, por otro lado, puede ser solicitada directamente por el autor del programa y está basada en ISO 9126. Comprende el análisis de la calidad en función de cuatro características: usabilidad, mantenibilidad, portabilidad y funcionalidad del producto. El proceso de evaluación se realiza en tres fases: en la primera fase se lleva a cabo la caracterización del software y la determinación de requisitos; la segunda fase comprende el diseño y ejecución de pruebas; y en la tercera y última fase se lleva a cabo el análisis de los resultados. En la metodología se definen también los tres participantes involucrados en el proceso (analista, autor y *tester*) y su papel en cada fase, y todo el proceso de evaluación se basa en una herramienta creada para tal propósito denominada eOPSOA.

En la primera fase, para la caracterización del producto el analista evalúa los atributos no funcionales (es decir, los correspondientes a la usabilidad, mantenimiento y portabilidad) mediante una lista de comprobación con preguntas de tipo si/no/no disponible. La lista de comprobación está dividida en once secciones, cada una con una serie de preguntas asociadas que permiten analizar los siguientes aspectos (CESLCAM, s. f.-b, pp. 63-70):

- La visibilidad del estado en que se encuentra el sistema.
- La correspondencia entre el producto software y el mundo real.
- El control y la libertad del usuario.
- La consistencia y el cumplimiento de estándares.
- La interacción basada más en el reconocimiento que en el recuerdo.
- La flexibilidad y eficiencia de uso.
- El diseño estético y minimalista.
- La ayuda y documentación que ofrece el producto software.
- El tratamiento de la privacidad que se hace en el producto software.
- La portabilidad, extensibilidad y mantenimiento.
- El proyecto, soporte, comunidad y licencias.

Para la caracterización del producto se pueden también llevar a cabo entrevistas con los desarrolladores (autores), siempre que sea posible y que el analista lo considere necesario. Además, esta actividad se puede complementar con sesiones de lluvia de ideas grupales (con desarrolladores, usuarios o interesados) y con el análisis de la documentación disponible. El último paso de la caracterización del producto es la descripción del sistema, que queda plasmada en dos documentos preparados por el analista: Visión del producto y Glosario de términos.

Estos documentos se utilizan luego en la segunda actividad de la primera fase, la determinación de los requisitos funcionales. Esta actividad es responsabilidad del analista, con posible colaboración del autor, y consiste en identificar una serie de casos de uso con base en el conocimiento acumulado sobre el programa, los usuarios previstos y las posibles secuencias de interacciones entre ellos (ver diagrama ilustrativo en la figura 2.8). Además, cada caso de uso identificado se clasifica en función de su importancia como de prioridad alta o baja.

En la segunda fase de la evaluación, el *tester* analiza el código fuente del programa y diseña y ejecuta las pruebas de escenario para evaluar los atributos funcionales con base en los casos de uso identificados en la fase anterior. Durante el proceso de ejecución de las pruebas se genera de forma automática un *log* o registro que el *tester* debe examinar al terminar para identificar los fallos o problemas detectados en las pruebas y calcular los resultados finales de las pruebas.

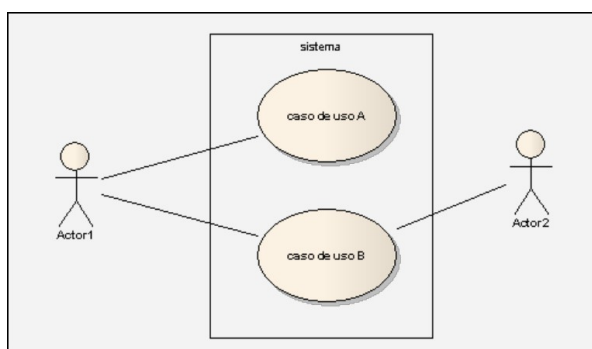


Figura 2.8: Diagrama de los actores y posibles casos de uso de un sistema (fuente: (CESLCAM, s. f.-b, p. 31).

En la tercera fase, con base en los resultados obtenidos en las fases anteriores, el analista determina si el producto obtiene o no la certificación positiva y elabora el informe final. Para recibir la certificación de nivel 1 o SILVER el programa debe haber superado el 100% de los casos de prueba con prioridad alta y el 90% de los casos de prueba con prioridad baja. Para obtener la certificación de nivel 2 o GOLD se requieren esos mismos resultados en las pruebas de escenario y por lo menos un 80% de respuestas afirmativas en la lista de comprobación de la primera fase.

Cuando el producto no obtiene la certificación de calidad, en el informe de resultados se incluyen los detalles sobre los problemas y errores encontrados, de modo que los autores puedan tomar las medidas correctivas necesarias.

#### ***2.3.3.4 El proyecto Open Source Software Management Framework***

Helander et al. (2007) presentan los resultados del proyecto Managing Open Source Software as an Integrated Part of Business (OSSSI)<sup>31</sup>, un proyecto llevado a cabo entre 2005 y 2007 por dos universidades finlandesas (Tampere University of Technology y University of Tampere) dedicado a estudiar las posibilidades que ofrece el software libre en el contexto empresarial. Su trabajo se centraba, pues, en buscar la mejor manera de gestionar la interacción entre las empresas y las comunidades de desarrollo de software libre. Para su propuesta de evaluación de software resultan entonces fundamentales dos aspectos: la identificación de los posibles niveles de participación de las empresas en los proyectos y la definición de una tipología que les permitiera clasificar las comunidades de desarrollo de software libre.

En cuanto al papel de las empresas, los autores afirman que “identificar y analizar el tipo de usuario es esencial para formular las preguntas adecuadas y para identificar los posibles cuellos de botella y riesgos a largo plazo” (p. 36, traducción propia). Así pues, distinguen cinco posibles niveles según el grado de contribución (de menor a mayor): usuario de la aplicación, utilización en I+D, integración de componentes, participación activa y gestión de la comunidad, y lanzamiento de nuevas comunidades. Naturalmente, los niveles de mayor participación incrementan la complejidad del marco de gestión necesario para la empresa en cuestión.

En cuanto a los tipos de comunidades, los autores llevaron a cabo un estudio empírico con base en encuestas y entrevistas a los desarrolladores de cuatro comunidades (Debian, Gnome, Eclipse y MySQL). Dicho estudio les llevó a concluir que es posible clasificar las comunidades con base en cuatro aspectos: su ética laboral, su madurez, su tamaño y el tipo de licencia utilizada para el código. En cuanto a la ética laboral de las comunidades, los autores distinguen a su vez tres tipos: comunidad de voluntarios que trabajan por diversión o para satisfacer sus

---

31 <http://coss.fi/ossi/>



propias necesidades (ética *hacker*); comunidad mixta liderada normalmente por una fundación sin ánimo de lucro (ética híbrida); y comunidad liderada por una o varias empresas (ética empresarial). Respecto a la madurez de las comunidades, se distinguen también tres tipos: comunidad joven, comunidad en desarrollo y comunidad ya establecida. En relación con el tamaño de las comunidades, los autores utilizan tres categorías: comunidad pequeña, mediana o grande. En cuanto a las estructuras de comunicación y toma de decisiones, los autores hablan de comunidades descentralizadas, balanceadas o centralizadas; de este aspecto se infiere también el sistema de gobierno de la comunidad, que según los autores será democrático, meritocrático o dictatorial, respectivamente. Finalmente, en cuanto al tipo de licencia que se utiliza para el código los autores utilizan la clasificación según el grado de permisividad: sin *copyleft*, con *copyleft* débil o con *copyleft* fuerte (ver tabla 1.1 en la página 10).

Para la evaluación, los autores proponen un enfoque holístico e interdisciplinario que implica dos fases: análisis del contexto y evaluación como tal (ver figura 2.9). En la primera fase, se deben responder tres preguntas básicas:

i) what software or softwares exist for the task in question and what communities are behind these softwares, ii) how will the software be used and what is the user role of the company (...) and, finally, iii) what kind of added value is sought by the OSS use (time-to-market, outsourcing, cost savings, etc.) (p. 36).

Una vez definido el contexto, en la segunda fase se lleva a cabo la evaluación mediante un análisis cualitativo y cuantitativo de los proyectos. Para la parte cualitativa se utiliza un cuestionario que permite determinar los riesgos desde los puntos de vista tecnológico, sociológico, legal y económico.

Para el análisis cuantitativo, por otro lado, los autores decidieron utilizar como base la propuesta del proyecto BRR (ver apartado 2.3.1.2), pero para los casos de mayor interacción con las comunidades recomiendan ampliar el modelo para cubrir mejor los aspectos socioculturales y de sostenibilidad tecnológica.

Para los dos niveles que implican menor participación de las empresas (usuario de la aplicación, utilización en I+D), los autores sugieren que el informe de

resultados es suficiente, mientras que para los tres niveles restantes (integración de componentes, participación activa y gestión de la comunidad, y lanzamiento de nuevas comunidades) el informe se debe complementar con una guía sobre gestión de comunidades que incluya recomendaciones prácticas que faciliten la cooperación empresa-comunidad.

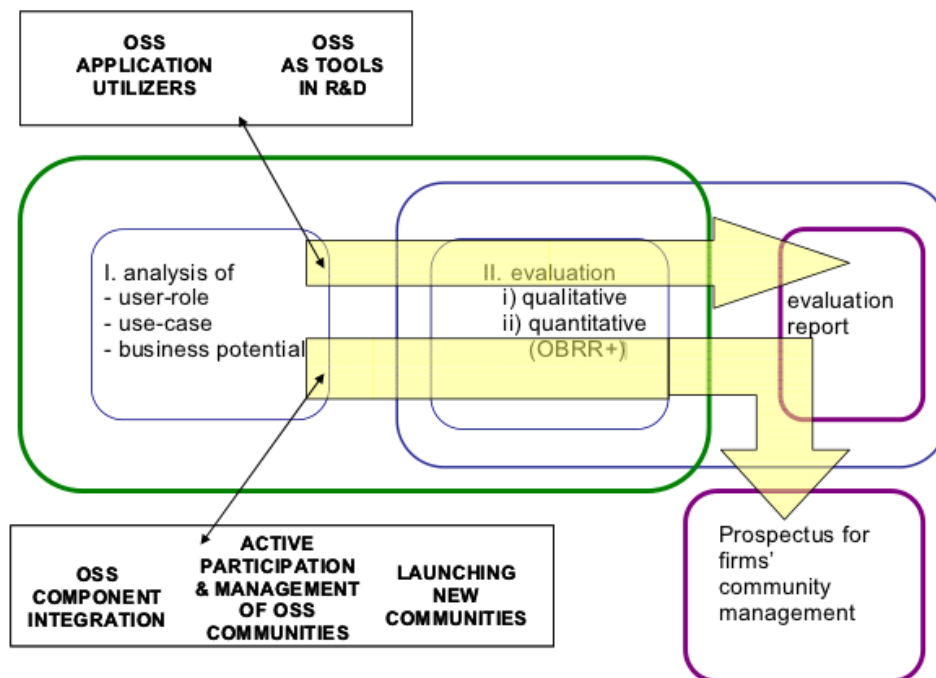


Figura 2.9: Proceso de evaluación propuesto por Helander et al. (2007, p. 36).

### 2.3.3.5 Modelos de éxito usados en el área de Sistemas de Información

Un sistema de información comprende las tecnologías (hardware, software y redes), individuos, datos y procesos involucrados en el funcionamiento y gestión de una organización. En un sentido más amplio, mientras que las ciencias de la computación se ocupan del *diseño y creación* de las tecnologías informáticas, los sistemas de información como disciplina tienen sus bases teóricas en las ciencias sociales y administrativas y la economía y se centran en el uso de dichas tecnologías y su impacto en individuos, grupos y organizaciones o en la sociedad en general (March, 2000). De ahí que en los últimos años esta disciplina haya dedicado

considerable atención al fenómeno del software libre y resulte relevante considerar sus aportes.

En el campo de los sistemas de información, el modelo de éxito más conocido es el de DeLone y McLean, propuesto originalmente en 1992 y actualizado en el año 2002 (DeLone y McLean, 2002; 2003). Se trata de un modelo que parte de la visión de desarrollo de sistemas en el contexto de una organización con base en tres aspectos: la creación del sistema, el uso del sistema y las consecuencias del uso de dicho sistema (Crowston, Howison, & Annabi, 2006). Así pues, según este modelo el éxito de un sistema de información depende tanto de la calidad del software, de la información (o datos) y de los servicios, como del uso (o intención de uso), de la satisfacción de los usuarios y del impacto o beneficios generados por el uso del sistema. Este modelo ha sido utilizado por autores como Crowston, Howison y Annabi (2003), Lee et al. (2009) y Wiggins, Howison, y Crowston (2009), entre otros, como base para la evaluación de software libre.

Crowston et al. (2003), por un lado, hacen énfasis en el carácter continuo del proceso de desarrollo del software libre y reestructuran el modelo inicial de DeLone y McLean concentrándose solo en tres aspectos: el producto software, el proceso de desarrollo y los efectos del proyecto para sus miembros. Los autores identifican una serie de métricas que permiten complementar el modelo inicial teniendo en cuenta las particularidades del software libre.

En cuanto al producto, los autores sugieren considerar aspectos como el paso de versión alfa a versión estable (según los desarrolladores), el cumplimiento de los objetivos planteados y la satisfacción de los desarrolladores. Para evaluar el proceso, destacan la importancia de aspectos como el número de desarrolladores que participan, el nivel de actividad del proyecto, el tiempo entre versiones y el tiempo necesario para corregir errores e implementar nuevas funciones. Finalmente, según los autores, aspectos como las nuevas oportunidades laborales para los desarrolladores, una mejor reputación o el conocimiento adquirido sirven como indicadores de los efectos o beneficios para los miembros del proyecto.

Lee et al. (2009), por otro lado, resaltan el hecho de que más del 50 % de los proyectos de software libre registrados en SourceForge no pasan de la versión alfa, por lo que se proponen llevar a cabo un estudio empírico con el que buscan definir y validar métricas que permitan medir el éxito o fracaso de un proyecto de software libre. Su objetivo es, pues, plantear un modelo teórico que permita una mejor comprensión del software libre y ofrecer unos resultados que permitan a los desarrolladores mejorar la gestión de los proyectos.

Partiendo del modelo de DeLone y McLean (2003), el modelo propuesto por los autores distingue cinco aspectos: la calidad del software, la calidad de los servicios ofrecidos por la comunidad, el uso del software, la satisfacción de los usuarios y los beneficios netos individuales (ver figura 2.10). Además, buscan explicar y predecir cómo influyen estos aspectos unos en otros.

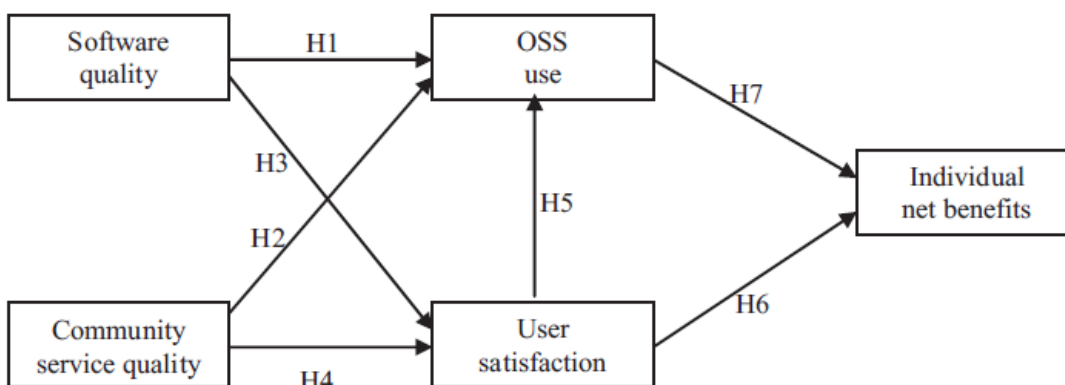


Figura 2.10: Modelo propuesto por Lee et al. (2009, p. 430).

Los autores analizan las percepciones de 157 informantes mediante un cuestionario de 24 afirmaciones que se deben responder en una escala de Likert de 1 a 7 (totalmente en desacuerdo a totalmente de acuerdo). El cuestionario fue administrado vía web y en sesiones presenciales con usuarios y desarrolladores de software libre. Entre las conclusiones del estudio cabe destacar que según sus resultados la calidad de los servicios no parece tener un efecto significativo en el uso del software a nivel individual, aunque sí parece tener un efecto positivo en la satisfacción de los usuarios. Según los autores, “information quality and system quality are the most important quality components for an individual user, whereas

service quality may become the most important variable in an organizational context” (Lee et al., 2009, p. 434).

Otra propuesta que se basa en el modelo de éxito de DeLone y McLean es el trabajo de Wiggins et al. (2009). Estos autores se centran en uno de los aspectos que resultan más problemáticos en el contexto del software libre: medir el posible interés en el software y el número de usuarios activos. A falta de una manera directa de medir el uso de un programa libre, lo más común, según los autores, es utilizar el número de descargas como indicador, asumiendo que para usar el software antes es necesario conseguirlo y que la manera más común de obtener un programa libre es descargándolo, normalmente desde la forja de desarrollo.

No obstante, como señalan los autores, esta suposición presenta varios problemas. En primer lugar, el número de descargas es insuficiente como indicador de los usuarios activos del programa ya que es imposible saber cuántos de los paquetes descargados efectivamente se instalan y utilizan regularmente, pues en muchos casos un usuario potencial puede descargar la aplicación para luego descartarla. En segundo lugar, utilizar una sola fuente para el número de descargas tampoco es suficiente, ya que el software libre puede distribuirse por varios canales de forma simultánea (repositorios de diferentes distribuciones de GNU/Linux y forjas de desarrollo, por ejemplo). En tercer lugar, si se utiliza el número de descargas totales del programa no hay manera de discriminar las descargas realizadas por usuarios activos que actualizan el programa, con lo que estos usuarios se estarían contando más de una vez. Además, el número de descargas totales también dependerá en gran medida del ciclo de lanzamientos del proyecto, por lo que un proyecto con lanzamientos frecuentes posiblemente tendrá un número mayor de descargas que un proyecto con un ciclo de lanzamiento más lento sin que esto se pueda interpretar necesariamente como un mayor número de usuarios regulares.

Teniendo en cuenta lo anterior y después de llevar a cabo un estudio detallado sobre el número de descargas diarias de dos proyectos libres (BibDesk y Skim-app) a lo largo del tiempo, los autores sugieren que es razonable asumir que el número

de descargas diarias de un programa permanecerá relativamente constante a lo largo del tiempo pues corresponde mayormente a las descargas realizadas por usuarios potenciales; por otro lado, el número de descargas en los días siguientes al lanzamiento de una nueva versión será mucho mayor por corresponder a las descargas realizadas por los usuarios regulares para actualizar el programa. En ambos casos, la tendencia general al alza o a la baja a lo largo del tiempo se puede interpretar como un indicador de que la popularidad del programa está aumentando o disminuyendo, respectivamente.

## 2.4 Recapitulación

The mere fact that so many evaluation initiatives were called into life, and research work of so many scientists was directed towards the development and improvement of evaluation methodologies shows that the need for such an evaluation methodology is, in fact, striking. (Höge, 2002, p. 41)

Como afirma Höge (2002) y como hemos visto en los apartados anteriores, el gran número de proyectos dedicados a la evaluación de software, desde diferentes perspectivas, muestra que este sigue siendo un campo de investigación muy fértil. Para resumir y analizar la literatura presentada en este capítulo empezaremos por retomar algunas recomendaciones generales sobre los métodos de evaluación y luego pasaremos a comparar las metodologías reseñadas tomando como punto de partida el marco de análisis propuesto por Stol y Ali Babar (2010). A partir de este análisis podremos pasar entonces a definir la metodología para el desarrollo del trabajo empírico, aplicando y adaptando las propuestas que resulten más relevantes teniendo en cuenta el contexto y alcance del presente trabajo.

En primer término es pertinente, pues, retomar algunas recomendaciones generales para tener en cuenta a la hora de preparar una evaluación de software. EAGLES (1996), por un lado, recomienda procurar que los métodos de evaluación cumplan tres propiedades básicas: **validez**, es decir, que los resultados permitan a los usuarios hacer inferencias sobre los programas; **fiabilidad**, en el sentido de que se deben obtener resultados similares o idénticos siempre que la evaluación se

realice en el mismo contexto aunque con distintos evaluadores; y **eficiencia de aplicación**, o la posibilidad de llevar a cabo la evaluación con el menor esfuerzo posible.

Según el proyecto BRR (2005), por otro lado, un buen método de evaluación debe ser completo, simple, adaptable y consistente. Un método **completo** debe abarcar todas las características de los programas de modo que los resultados no sean engañosos. **Simple** se refiere al hecho de que sea fácil de entender y fácil de usar para que el método tenga buena aceptación. Un método **adaptable** debe ser abierto y debe poder ampliarse para considerar desarrollos futuros. **Consistente**, finalmente, se refiere a que las escalas y puntuaciones del modelo sean comparables.

Es de destacar también la recomendación de Deprez (2009) de definir una metodología general que se pueda adaptar a diferentes contextos de manera estructurada, estableciendo una serie de perfiles de uso y especificando en cada caso las actividades, tareas, criterios de calidad, materiales y resultados requeridos. Helander et al. (2007), por su parte, nos recuerdan que “durante la etapa de preparación [de la evaluación], es imposible capturar todos los usos potenciales del modelo” (p. 30, traducción propia). En este sentido, parecen coincidir con la recomendación de Rico (2001) y Deprez (2009) de lograr un método general, riguroso y sistemático, que sea reutilizable y que se adapte fácilmente a diferentes casos de uso.

Para comparar diferentes metodologías de evaluación, Stol y Ali Babar (2010) proponen un marco de análisis en el que plantean, entre otras, las siguientes preguntas:

- ¿Cuáles son los objetivos?
- ¿Quiénes son los beneficiarios previstos?
- ¿Cuál es el perfil del evaluador?
- ¿Cuáles son las actividades y pasos?
- ¿Cuáles son las categorías y criterios de calidad utilizados?
- ¿Se incluye la funcionalidad?
- ¿Cuáles son los resultados esperados?
- ¿Existe un repositorio público de recursos y resultados?
- ¿Se basa el proceso en alguna herramienta? (p. 3).

Siguiendo su propuesta, utilizaremos dichas preguntas para estructurar el resumen que se presenta a continuación.

- ***¿Cuáles son los objetivos y quiénes son los beneficiarios previstos de los resultados?***

La mayoría de los métodos de evaluación reseñados tienen por objetivo ayudar a los usuarios a seleccionar entre las opciones disponibles la herramienta que mejor se adapte a su contexto específico de uso y a sus necesidades. En este caso, los beneficiarios previstos de los resultados de la evaluación son los usuarios finales de las herramientas.

En otros casos, como en el modelo CMMI, el modelo del proyecto QualiPSo y el del marco de gestión de software de código abierto propuesto en Helander et al. (2007), el objetivo es ayudar a mejorar la gestión del proyecto de desarrollo; en otras palabras, se busca mejorar los procesos mediante la implementación de ciertas prácticas recomendadas y el planteamiento de ciertos objetivos a largo plazo. En este caso, los beneficiarios previstos de los resultados son los desarrolladores y otras partes vinculadas al proyecto.

Otros autores, como Rico (2001), Höge (2002), (CESLCAM, s. f.-b) y Filatova (2010), por su parte, asumen un enfoque orientado al usuario final pero contemplando también la posibilidad de que los resultados de la evaluación sirvan a los desarrolladores para mejorar las herramientas.

Cuando los beneficiarios previstos de los resultados de la evaluación son a la vez los usuarios finales de las herramientas es importante identificar y analizar los posibles perfiles de uso y el marco de colaboración previsto con las comunidades de software libre (Helander et al., 2007). En este sentido, como parte del proyecto QualiPSo en las encuestas realizadas a empresarios europeos se encontró que los usuarios que solo planean utilizar la herramienta no muestran interés en el proceso de desarrollo, a diferencia de aquellos que planean también mejorarla o integrarla con otros componentes de software.



- ***¿Cuál es el perfil del evaluador?***

Según Höge (2002), para llevar a cabo un proceso de evaluación de software son importantes tres aspectos: a) conocer las necesidades de los usuarios; b) entender la funcionalidad de los sistemas que se evalúa; y c) conocer las técnicas necesarias para realizar pruebas de software. No obstante, en algunas de las propuestas reseñadas, como Rico (2001), BRR (2005) o Filatova (2010), se busca lograr precisamente que sean los usuarios finales los que lleven a cabo la evaluación, con lo cual se asume que el evaluador no está familiarizado con las técnicas de evaluación de software.

En la propuesta de CESLCAM, por otro lado, en la evaluación participan tres actores con diferentes perfiles: el autor o desarrollador del programa, que conoce las características del programa; el analista, que debe tener conocimientos básicos relacionados con la Ingeniería de Requisitos y de Software; y el *tester*, que debe estar familiarizado con las técnicas de pruebas de software.

En otros casos, como en las propuestas de QSOS (Atos Origin, 2006) y Mašlanko (2004), algunas actividades del proceso son llevadas a cabo por el evaluador, pero la fase final o de selección es llevada a cabo directamente por el usuario final.

- ***¿Cuáles son las actividades y pasos para llevar a cabo la evaluación?***

Encontramos que en las propuestas reseñadas las actividades para llevar a cabo la evaluación en general tienden a ser las mismas (si bien se pueden encontrar agrupadas en fases de diferentes maneras):

- Definición del contexto de evaluación
- Definición de los criterios de calidad
- Definición de los atributos y métricas
- Definición de los procedimientos para obtener las puntuaciones globales
- Identificación de programas candidatos a evaluación
- Evaluación de los candidatos
- Análisis de resultados
- Presentación de resultados
- Selección de herramienta y/o mejora de procesos y producto

Una característica común de los modelos de evaluación propuestos para el software libre es que una de las etapas del proceso consiste en hacer una preselección de programas para pasar luego a evaluar los mejores candidatos de manera más detallada. Dado el amplio número de proyectos libres existentes en la actualidad, esta preselección permite descartar de manera rápida los programas que no cumplen ciertos requisitos básicos, de modo que se puede optimizar el tiempo y los recursos disponibles para la evaluación.

En cuanto a la definición del contexto, a la hora de preparar la evaluación se deben considerar factores como el grupo de interés que motiva la evaluación, el tipo de herramienta a evaluar, el uso previsto que tendrá la herramienta y los recursos humanos, técnicos y económicos disponibles para la evaluación (EAGLES, 1996). Según estos factores se procede entonces a decidir el tipo de pruebas que se realizarán, así como los instrumentos y materiales necesarios (Ibid.). Mašlanko (2004) resalta que la mayoría de propuestas suponen una evaluación a gran escala y por tanto son difíciles de aplicar para usuarios independientes, como puede ser el caso de los traductores. Su propuesta y la de Filatova (2010), en cambio, están pensadas precisamente para ser llevadas a cabo a pequeña escala, bien sea por traductores independientes o por pequeñas empresas.

En relación con los tipos de pruebas, cuando la motivación para la evaluación es comprobar la funcionalidad general de los programas el tipo de pruebas más recomendable según el grupo EAGLES (1996) es la inspección de características. Höge (2002), por su parte, también sugiere que este tipo de pruebas es el que mejor se adapta al objetivo de facilitar la selección de herramientas, mientras que las pruebas sistemáticas orientadas a las tareas y de tipo *benchmarking* solo parecen recomendables para complementar los resultados obtenidos mediante inspección de características; las pruebas sistemáticas orientadas a los menús parecen ser aún menos relevantes en este contexto y las pruebas de escenario parecen dificultar la obtención de resultados válidos, fiables y relevantes a la hora de seleccionar herramientas, razón por la cual la misma autora recomienda este tipo de pruebas solo para validación de resultados.

Otra ventaja de la inspección de características en el contexto del presente trabajo es que gran parte de la información necesaria para este tipo de prueba se puede obtener muchas veces de la documentación de los programas, con lo que se ahorra tiempo y esfuerzo, aspecto importante a la hora de considerar los recursos humanos y técnicos disponibles para el proceso de evaluación. Por último, la inspección de características permite resultados más objetivos, ya que se basa fundamentalmente en identificar la ausencia o presencia de ciertas características en los programas, sin ocuparse de identificar problemas o errores (EAGLES, 1996; Höge, 2002).

- ***¿Cuáles son las categorías y criterios de calidad utilizados? ¿Se incluye la funcionalidad?***

En la mayoría de las propuestas reseñadas al preparar la evaluación se debe decidir qué criterios tener en cuenta, dejando fuera aquellos que resulten menos relevantes para el contexto en cuestión. Para definir los criterios de calidad se utiliza normalmente un enfoque de lo general a lo específico en el que primero se definen las características de más alto nivel y luego se detallan los atributos en los que se descompone cada característica. Como hemos visto en este capítulo, la evaluación de software se puede abordar desde diversas perspectivas teniendo en cuenta en primer término si se evaluará solo el software como producto o si se tendrá también en cuenta el proceso de desarrollo.

Para delimitar la evaluación y seleccionar los criterios a tener en cuenta, la mayoría de propuestas coinciden en su recomendación de considerar el tipo de herramientas a evaluar y el entorno previsto de uso, aspectos que el proyecto BRR agrupa en el concepto de *orientación funcional de los programas*. A partir de los tipos de usuarios del software y sus necesidades podemos establecer unos escenarios de uso que permiten crear perfiles típicos con el fin de reducir el esfuerzo necesario para llevar a cabo la evaluación (Höge, 2002).

En cuanto a la evaluación del software como producto, el estándar ISO 9126 constituye quizá el modelo de calidad más ampliamente reconocido (Filatova, 2010; Samoladas et al., 2008). Este estándar de calidad establece seis características

básicas (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad) que comprenden una serie de subcaracterísticas (ver figura 2.1 en la página 64). De estas seis características, las últimas cinco se pueden agrupar bajo la denominación de “no funcionales” y son igualmente aplicables para la evaluación de cualquier tipo de software. La funcionalidad, por su parte, está relacionada con el tipo de tareas que se realizan con el programa (por ejemplo, edición de textos, búsqueda de concordancias o compresión de archivos), por lo que la evaluación de esta característica no es generalizable a diferentes tipos de programas.

Cuando la evaluación se hace desde el punto de vista del usuario final con el fin de facilitarle la comparación de herramientas similares, varios de los autores reseñados coinciden en que las funciones que ofrece cada tipo de herramientas constituyen un elemento clave a considerar (BRR, 2005; EAGLES, 1996; Höge, 2002; van den Berg, 2005; Wheeler, 2008). Además, como sugieren Wheeler (2008) y van den Berg (2005), es necesario comprobar que las herramientas soporten estándares y formatos abiertos que garanticen su interoperabilidad con otras herramientas. Según el estándar ISO 9126 estos aspectos se consideran subcaracterísticas de la funcionalidad.

Teniendo en cuenta que la funcionalidad, a diferencia de las características no funcionales, es específica según el tipo de programa, para un modelo de evaluación en el que se quieran abarcar programas de diferentes tipologías sería necesaria una lista de funciones para cada tipología. Delimitar las funciones que pueden ser de interés en la evaluación de cierto tipo de herramienta, como bien afirman King (1998) y BRR (2005), puede representar demasiado tiempo y esfuerzo en la etapa de preparación de la evaluación. En este sentido, resulta pertinente la sugerencia del proyecto BRR de utilizar listas predefinidas tomadas de otras fuentes, si bien dichas listas se deben adaptar ligeramente según el contexto específico de evaluación.

Otro aspecto en el que coinciden varios autores es que desde el punto de vista del usuario final es importante que las herramientas sean amigables o *user-friendly*, lo que implica considerar aspectos como la documentación incluida, la interfaz de usuario y el tipo de asistencia técnica disponible (van den Berg, 2005; BRR Project,

2005; Cerezo, 2003; Wheeler, 2008), aspectos que según ISO 9126 corresponden a la característica de usabilidad. No obstante, es necesario mencionar el hecho de que evaluar esta característica implica invariablemente juicios subjetivos, por lo que algunas de las propuestas de evaluación reseñadas (por ejemplo Mašlanko [2004] y Samoladas et al. [2008]) no la consideran.

Al evaluar programas para seleccionar los que más se ajusten a las necesidades de un entorno particular, como lo plantean la mayoría de autores reseñados, es importante tener en cuenta los sistemas operativos para los que están disponibles los programas, pues esto puede representar una limitación para su implementación en entornos existentes. Además, es también importante considerar el tiempo, el esfuerzo y los requisitos técnicos necesarios para la instalación del programa, pues de ellos dependerá que el usuario efectivamente logre poner el sistema en funcionamiento para su utilización, y por tanto su aceptación (van den Berg, 2005; BRR Project, 2005; Wheeler, 2008; Höge, 2002). Estos aspectos encajan en lo que la norma ISO 9126 define como portabilidad.

Evaluar las tres características restantes que plantea ISO 9126 (eficiencia, fiabilidad y mantenibilidad) implica en la mayoría de los casos la utilización de métricas basadas en el análisis del código fuente de los programas (Samoladas et al., 2008). Además, como nos recuerda Höge: “Existe un gran riesgo de que el procedimiento de evaluación resulte inmanejable si en el proceso de decisión final se consideran demasiadas características no funcionales que tengan un papel menor [en el proceso de análisis específico]” (2002, p. 92, traducción propia).

La fiabilidad, en particular, implica considerar atributos directamente relacionados con la frecuencia en que ocurren errores por fallos en el sistema, la tolerancia a los fallos (es decir, la habilidad del programa para mantener un nivel de desempeño especificado) y la capacidad de recuperación luego de que ocurre un fallo. Estos aspectos resultan especialmente relevantes cuando la motivación de la evaluación es examinar el comportamiento de un programa en condiciones específicas, un tipo de evaluación que normalmente se lleva a cabo desde el punto de vista de los desarrolladores y para el que según EAGLES (1996) el tipo de

pruebas más indicado son las pruebas sistemáticas. Además, como bien indica Filatova (2010), un usuario final rara vez estará en capacidad de realizar las pruebas necesarias para evaluar esta característica.

La eficiencia, por su parte, implica determinar el desempeño de los programas en relación con los recursos de hardware y el tiempo necesario para los procesos. Gracias a los grandes avances en la capacidad de procesamiento de los ordenadores personales, hoy en día los recursos de hardware ya no son un factor tan determinante como hace unos años (Höge, 2002) y, en términos generales, los programas libres tienden a ser menos exigentes en este sentido que sus equivalentes privativos, por lo que este aspecto pierde relevancia en el contexto del presente trabajo. En cuanto al tiempo necesario para los procesos, Höge (2002) resalta la dificultad de definir maneras objetivas de medir este aspecto para los sistemas interactivos, como es el caso de las herramientas de traducción. Además, tanto los recursos como el tiempo necesarios dependerán en gran medida del volumen de datos que se procese, razón por la cual no parece apropiado incluir estos aspectos cuando el tipo de prueba que se va a utilizar es la inspección de características.

La mantenibilidad, finalmente, abarca los aspectos que tienen que ver con el esfuerzo necesario para implementar modificaciones en los programas, incluyendo el diagnóstico de los problemas, la modificación del código para resolverlos y las pruebas necesarias para verificar que dichos cambios tengan los efectos esperados. Como concluye Höge (2002), estos aspectos resultan especialmente relevantes para los desarrolladores, pero no tanto cuando la evaluación se lleva a cabo desde el punto de vista del usuario final, como en el presente trabajo.

Con el surgimiento y posterior expansión del software libre, los modelos de calidad tradicionales han sido adaptados para responder a las características distintivas del modelo de desarrollo de este tipo de software (Samoladas et al., 2008). Así pues, además de las características de calidad del producto establecidas en el estándar ISO 9126, los modelos de calidad del software libre incluyen aspectos que corresponden al proceso de desarrollo y a la comunidad que respalda

al proyecto, al ser estos los principales factores que diferencian al software libre del software privativo.

Al igual que los modelos tradicionales, los modelos de calidad específicos para software libre también definen una jerarquía de características que van de lo general a lo específico y una serie de métricas y procedimientos para asignar las puntuaciones para cada atributo. Como hemos visto, para la evaluación de los proyectos de software libre por lo general se incluyen aspectos como el marco ideológico del proyecto, el tipo de licencia que utiliza, la planificación, las estructuras de comunicación y toma de decisiones y la colaboración con otros proyectos libres, aspectos que en conjunto nos permiten valorar la estrategia y seriedad de los proyectos.

Otros aspectos como la capacidad de mantenimiento del proyecto, su sostenibilidad o posibilidad de supervivencia a largo plazo, los servicios disponibles para los usuarios y la existencia de proyectos derivados dependen directamente de las personas y entidades involucradas y nos permiten hacernos una idea de la comunidad que da soporte al proyecto.

Los modelos de calidad adaptados al software libre también incluyen aspectos que permiten determinar la madurez del proceso de desarrollo, como la fecha de inicio del proyecto, las características de la página web oficial, la utilización de tecnologías específicas para la gestión del código, de los informes de fallos y solicitudes de nuevas funciones, así como las tecnologías utilizadas por los miembros de la comunidad para comunicarse y proporcionar asistencia técnica a otros usuarios. Otros indicadores utilizados para valorar la madurez del proceso de desarrollo son el número de versiones que se han lanzado y la existencia de un ciclo de lanzamiento de versiones predefinido.

Finalmente, encontramos que varias propuestas de evaluación de software libre incluyen aspectos que tienen que ver con la reputación de los proyectos, como, por ejemplo, la existencia de implementaciones de referencia y usuarios reconocidos, la disponibilidad de libros o reseñas especializadas, el número de

descargas en determinado periodo de tiempo y los comentarios de los usuarios en los foros.

- ***¿Cuáles son los resultados esperados?***

Las metodologías de evaluación reseñadas se basan casi todas en un análisis cuantitativo del software, que en algunos casos se complementa con un análisis cualitativo. Para el análisis cuantitativo es necesario definir las métricas o procedimientos para obtener las puntuaciones. Para esto es común recurrir a una técnica conocida como GQM (objetivo-pregunta-métrica), que permite pasar del plano conceptual (objetivo) al plano operativo (pregunta) para llegar finalmente al plano cuantitativo (métrica).

Como resultado del análisis cuantitativo se obtiene una puntuación final que consolida las puntuaciones obtenidas para cada atributo considerado. En la mayoría de los casos se utiliza el promedio ponderado como método para consolidar las puntuaciones (Samoladas et al., 2008), lo que implica la asignación de pesos según la importancia relativa de cada atributo en el contexto.

En el caso de las propuestas de QualiPSO y CESLCAM, enfocadas a ayudar a mejorar los procesos, los resultados de la evaluación incluyen también objetivos y prácticas que los miembros del proyecto pueden utilizar para reducir los riesgos identificados. La propuesta de Helander et al. (2007), por otro lado, contempla la creación de guías que permitan a las empresas interesadas en colaborar en los proyectos lograr una participación fructífera.

- ***¿Existe un repositorio público para compartir recursos y resultados? ¿Se basa el proceso en alguna herramienta?***

En el caso de la evaluación de las tecnologías de la traducción, los instrumentos utilizados para la evaluación suelen estar disponibles como listas o tablas en documentos de texto. En este sentido, Höge (2002) resalta la relevancia de las posibilidades que ofrece Internet para desarrollar plataformas que permitan



compartir recursos y herramientas para la evaluación y así reducir el esfuerzo necesario para futuras evaluaciones.

En el caso del software libre, como hemos visto, varios de los proyectos de evaluación, como QSOS, QualiPSo y CESLCAM, sí disponen de repositorios públicos en línea donde se almacenan y comparten recursos y resultados de evaluación. En algunos casos, además, el proceso de evaluación en sí está basado en una herramienta creada para automatizar el proceso, que puede estar disponible como una plantilla para descargar (QSOS y BRR), como aplicación web (QualiPSo) o como *plugin* para los entornos de desarrollo (SQQ-OSS).



## **SEGUNDA PARTE**

### **3 TRABAJO EMPÍRICO**

En este capítulo se presentan las tres fases del trabajo empírico: selección de programas candidatos a evaluación; adaptación de una metodología para la evaluación de las tecnologías libres para la traducción; y evaluación como tal. Como resultado de la primera fase se obtuvo un catálogo de programas libres para el traductor; de la segunda fase se obtuvieron un instrumento de evaluación, un protocolo para su aplicación y una interfaz de selección de programas; de la tercera fase, finalmente, se obtuvieron las fichas descriptivas con toda la información recogida sobre los proyectos y programas evaluados.

Este capítulo está organizado en cinco apartados. En el primer apartado se presentan los objetivos del trabajo. En los tres apartados siguientes se describe cada una de las tres fases del desarrollo del trabajo empírico. Por último, en el quinto apartado se presentan las conclusiones del proceso adelantado, los problemas encontrados y los aspectos a tener en cuenta para futuros trabajos.

#### **3.1 Objetivos**

- a) Elaborar un catálogo de programas libres útiles para el traductor.
- b) Preparar un método de evaluación para las tecnologías libres para la traducción a partir de métricas que permitan una evaluación rápida y fiable.
- c) Caracterizar una muestra de proyectos de desarrollo de tecnologías libres

para la traducción y evaluar las herramientas creadas por dichos proyectos desde el punto de vista del traductor como usuario final.

- d) Poner a disposición de los traductores interesados el catálogo de programas y los instrumentos y resultados de la evaluación de modo que tengan bases para la toma de decisiones a la hora de escoger las herramientas para su entorno de trabajo.

En el contexto del presente trabajo nos interesa principalmente definir una metodología de evaluación que facilite a los traductores la selección de herramientas libres para su entorno de trabajo, de modo que la evaluación que se plantea se debe llevar a cabo desde el punto de vista de los usuarios finales. Nos proponemos definir una metodología que resulte útil para traductores profesionales (tanto individuos como equipos), pequeñas empresas, profesores y estudiantes de traducción, así como investigadores. No obstante, sin ser un objetivo directo, consideramos que los resultados obtenidos también pueden ser útiles como punto de partida para identificar algunas de las debilidades de los proyectos de desarrollo de modo que los desarrolladores de los proyectos estudiados puedan trabajar para mejorar esos aspectos.

Este trabajo se desarrolló en tres etapas. En la primera etapa se llevó a cabo la búsqueda, selección y clasificación de los programas candidatos a evaluación que dio como resultado un catálogo de software libre para traductores (apartado 3.2). La segunda etapa consistió en preparar la evaluación, para lo cual se delimitó el contexto de la misma, se decidieron los criterios a considerar con base en lo encontrado en la literatura reseñada y se crearon los materiales necesarios para la evaluación (apartado 3.3). La tercera y última etapa consistió en evaluar una selección de programas, analizar los datos obtenidos y presentar los resultados de la evaluación (apartado 3.4).

## **3.2 Fase 1: Elaboración del catálogo de software libre para traductores**

Para adelantar esta etapa del proceso el primer paso consistió en establecer los criterios que se utilizarían para seleccionar los programas libres para el traductor candidatos a evaluación. Además, se decidió qué información incluir en la ficha que se rellenaría para cada programa, así como el programa que se utilizaría para almacenar los datos. El siguiente paso fue decidir la manera de clasificar los programas seleccionados por categorías según tipologías. Finalmente, se identificaron las principales fuentes que se utilizarían para buscar los programas y se decidió el procedimiento a seguir para la recolección de los datos.

### **3.2.1 Criterios de selección de los programas**

Para la elaboración del catálogo de programas libres para el traductor candidatos a evaluación se establecieron como criterios de selección el tipo de licencia, el estado de desarrollo del proyecto y la categoría a la que pertenece el programa según la clasificación de las tecnologías de la traducción adoptada para este trabajo (ver apartado 1.2).

En cuanto a los tipos de licencia, para que un programa fuera seleccionado la licencia utilizada debía ser una de las aprobadas por la FSF o por la OSI, o en su defecto debía ser gratuita y permitir la libre copia, distribución y modificación del programa. Según el estado de desarrollo del proyecto se descartaron los programas que aún estaban en etapa de planeación o que solo estaban disponibles en versión pre-alfa o alfa según los mismos desarrolladores. En cuanto a las categorías de herramientas (ver apartado 3.2.3), se excluyeron los programas de las categorías equipo del traductor y herramientas de comunicación y documentación dado que por ser tan generales el número de opciones disponibles es demasiado amplio e imposible de abarcar en este proyecto. La tabla 3.1 a continuación resume los criterios utilizados para la selección de los programas para el catálogo de software libre para el traductor.

Aspecto	Criterios de inclusión
Licencia	Aprobada por la FSF o por la OSI, o que permita libre copia, distribución y modificación.
Estado de desarrollo	Beta, producción/estable o maduro.
Categoría	Edición y maquetación, herramientas lingüísticas, herramientas de traducción o herramientas de gestión.

*Tabla 3.1: Criterios para la selección de los programas para el catálogo.*

### **3.2.2 Diseño de la ficha para almacenar los datos**

Para decidir la información que se iba a incluir en la ficha de cada programa se consideraron los datos que se incluyen en catálogos de software como el de Hutchins (2008), el de Sokoli (2002) o el del Centro Virtual desarrollado por el grupo Tecnolettra (2008) de la Universitat Jaume I, coordinado por la Dra. Amparo Alcina, en el marco del Observatorio de Tecnologías de la Traducción, un proyecto con la participación de varias universidades españolas y coordinado por la Dra. Celia Rico, investigadora de la Universidad Europea de Madrid.

Así pues, se decidió que para cada programa la ficha incluiría una descripción tomada de la página web de cada proyecto, la indicación de la última versión disponible, los sistemas operativos soportados, el URL de la página web del proyecto y la categoría a la que pertenece el programa. También se decidió incluir información sobre los requisitos de instalación en términos de software adicional necesario, si fuera el caso; como información opcional, de estar disponible, se incluyó el lenguaje de programación y el nombre de los programadores, principales contribuidores o de la empresa que desarrolla el programa.

El otro dato que se incluyó en la ficha, y al que se prestó especial atención dados los objetivos del presente trabajo, fue el tipo de licencia. Para los programas libres esto implica normalmente indicar el nombre de la licencia o licencias utilizadas por el programa (ej. GNU GPL, GNU LGPL, Apache License, etc.), aunque en algunos casos, cuando se utilizan licencias personalizadas, fue necesario

indicarlo de manera genérica como “licencia libre” o “licencia de código abierto” según la inclinación ideológica del proyecto. En la tabla 3.2 se presenta el modelo de ficha que se utilizó para la recolección de datos.

	<b>Nombre del programa</b>
<b>Programador</b>	Información opcional. Puede ser el nombre del programador, principales contribuidores o empresa que desarrolla el programa.
<b>Descripción</b>	Breve descripción del programa tomada de la página web del proyecto.
<b>Versión</b>	Indicación del número de la última versión estable disponible.
<b>Sistemas operativos</b>	Sistemas operativos para los que está disponible el programa.
<b>Lenguaje de programación</b>	Información opcional. Suele estar disponible para los programas libres.
<b>URL</b>	Página principal del programa. Si el programa no cuenta con una página web dedicada, dirección de la página donde se describe el programa o desde donde se puede descargar.
<b>Licencia</b>	Nombre de la licencia o licencias (ej. GNU General Public License, GNU Lesser General Public License, etc.).
<b>Categoría</b>	Según la descripción de las funciones del programa (ver apartado 5.2.1.3. Clasificación de los programas).
<b>Requisitos</b>	Se indica si es necesario instalar software adicional.

*Tabla 3.2: Modelo de ficha que se utilizó para el catálogo.*

Para almacenar los datos se decidió utilizar el programa libre Zotero, gestor de referencias bibliográficas que tiene la ventaja de instalarse como extensión del navegador Firefox, con lo cual se facilita la labor de llenar la información sobre los programas a partir de sus páginas web sin cambiar de aplicación. Además, Zotero permite mantener una copia sincronizada en sus servidores, a la que también se puede acceder vía web desde cualquier ordenador con conexión a Internet.

### **3.2.3 Clasificación de los programas**

Como se mencionó al finalizar el apartado 1.2, luego de considerar las propuestas de clasificación encontradas en la literatura reseñada, para el presente trabajo se decidió utilizar como base la clasificación propuesta por Alcina (2008). Esta propuesta está compuesta por cinco categorías (equipo del traductor, herramientas de comunicación y documentación, herramientas de edición y maquetación, herramientas lingüísticas y herramientas de traducción), a las que se agregó una categoría más (herramientas de gestión). Dado el alcance de este proyecto, y como ya se explicó en el apartado 3.2.1, no se incluyeron en el catálogo los programas de

las categorías *equipo del traductor* y *herramientas de comunicación y documentación*.

Así pues, los programas seleccionados debían clasificarse en una de las cuatro categorías restantes. Para clasificar cada programa se tomó como base la descripción presentada en su página web y se utilizó una guía de referencia (ver tabla 3.3) en la que para cada categoría se detallan los tipos de programas que esta abarca. No obstante, la tabla no se considera cerrada, ya que se contaba con la posibilidad de encontrar programas de otros tipos que encajaran en las categorías consideradas.

<b>Categoría</b>	<b>Tipos de programas</b>
Herramientas de edición y maquetación	Editores de texto, suites ofimáticas, procesadores de texto, herramientas de maquetación, edición avanzada de imágenes, editores de HTML y XML, herramientas de reconocimiento óptico de caracteres, utilidades para conversión y validación de formatos.
Herramientas lingüísticas	Herramientas para extracción terminológica, herramientas para creación y explotación de corpus, herramientas para gestión de bases de datos terminológicas, herramientas para búsquedas en recursos lingüísticos.
Herramientas de traducción	Programas de traducción asistida por ordenador, programas de traducción automática, herramientas para localización, herramientas para gestión de memorias de traducción, herramientas para alineación de textos, utilidades para conversión y validación de formatos específicos de traducción.
Herramientas de gestión	Programas para gestión de proyectos, herramientas para creación de presupuestos y facturas, herramientas para gestión financiera, herramientas para gestión de recursos, herramientas para control de calidad.

*Tabla 3.3: Categorías utilizadas para clasificar los programas seleccionados y tipos de programas que corresponden a cada una de ellas.*

### **3.2.4 Fuentes utilizadas para la búsqueda**

Para buscar los programas libres para el traductor se partió de la experiencia propia con algunos programas de este tipo, de varios catálogos de software para traductores, de recopilaciones de enlaces a programas libres para el traductor y de dos distribuciones de GNU/Linux especialmente para traductores. También se realizaron búsquedas en la forja de software libre SourceForge y búsquedas generales en Internet.



Una de las fuentes consultadas fue el Compendium of Translation Software, de Hutchins (2008), disponible en formato pdf en la página web del autor (<http://www.hutchinsweb.me.uk/Compendium-14.pdf>). Este directorio de los sistemas de traducción automática y herramientas de traducción asistida por ordenador disponibles en el mercado incluye, además del nombre y versión de los programas, información sobre la compañía, el tipo de sistema, los idiomas, los requisitos de sistema y el precio, entre otros. La búsqueda en esta fuente se llevó a cabo consultando solo las opciones gratuitas, pero se encontró que en la mayoría de los casos se trataba de servicios de traducción automática o diccionarios electrónicos y no de herramientas, con lo que finalmente fueron muy pocos los programas encontrados en esta fuente.

Otra fuente utilizada fue el Centro Virtual del Observatorio de Tecnologías de la Traducción (Tecnolettra, 2008). Este sitio permite consultar una base de datos de “software relacionado directa o indirectamente con tareas relacionadas con la traducción o que puedan facilitar el trabajo al traductor en alguna de las fases del proceso de traducción.” Las búsquedas se pueden especificar por nombre del programa, categoría, descripción, precio (gratuito o no gratuito), empresa, distribuidor y requisitos técnicos (Windows, GNU/Linux o Mac). Las categorías que se distinguen son: equipo, listas de distribución y redes, edición, diccionarios/bancos de datos, corpus, traducción asistida por ordenador, traducción automática y herramientas de Internet. Así pues, se buscaron las herramientas gratuitas para cualquier sistema operativo pertenecientes a las categorías de edición, traducción asistida por ordenador y traducción automática. La búsqueda, sin embargo, no arrojó resultados útiles, ya que solo apareció un programa: WordFast, que en realidad no es gratuito, sino que ofrece un modo demo muy “generoso” en palabras de su desarrollador.<sup>32</sup>

Se recurrió también a la página *Free and open source link list*, del sitio web del traductor Wunderlich disponible en <http://www.martinwunderlich.com/>. Se listan en la página los proyectos de software libre y de código abierto de las áreas de traducción y localización, sin incluir proyectos más generales como editores de

---

<sup>32</sup> Posteriormente, como colaboradora del grupo Tecnolettra incorporé los programas libres recopilados en el catálogo a la base de datos del Centro Virtual (ver apartado 3.2.7.1).

texto o procesadores de palabra. Además del software libre y de código abierto, hay una sección aparte en la que se incluyen algunos programas *freeware*. El visitante tiene también la posibilidad de suscribirse para recibir las actualizaciones de la lista. En este listado se incluye una breve descripción de los programas, los requisitos de sistema, el tipo de licencia y el enlace a la página web del programa. Aunque no se ha actualizado recientemente, esta fuente, que incluye unos cincuenta programas, fue uno de los recursos más importantes para el presente trabajo.

Mediante un enlace en la página anterior, se llegó a la Open Source Translation Technology Initiative (<http://ostti.net/>), un proyecto creado con el objetivo de agrupar los proyectos de software libre en las áreas de traducción automática y asistida por ordenador. Aunque inicialmente se pensó que este sería un buen repositorio de herramientas, la realidad es que solo hay 7 proyectos registrados de los cuales solo 2 aparecen con estado de desarrollo estable y solo uno tiene archivos para descargar. Al parecer la iniciativa no tuvo mucha acogida, ya que las últimas noticias que aparecen en la página son de finales de 2006.

Se visitó también la página web *Linux for translators* (<http://www.linuxfortranslators.org/linux/linux.html>), del traductor Prior. Como lo indica el título, esta página está dirigida a los traductores que utilicen o quieran utilizar GNU/Linux para sus actividades diarias. Así pues, en esta página, además de información general y preguntas frecuentes sobre GNU/Linux, se encuentra también un listado de las herramientas para traductores disponibles para este sistema operativo. El listado incluye las categorías de procesadores de texto, memorias de traducción, herramientas para alineación, herramientas para gestión de diccionarios y bases de datos, utilidades para búsqueda y recuperación de información, editores de texto, herramientas de gestión y herramientas de traducción automática. Para cada programa se incluye una breve descripción y el enlace a su página web.

Se utilizó también la página web *Free and Open Source Software for Translators* (<http://www.translatewrite.com/foss/index.php?s=foss&p=main>), de la traductora McKay, que incluye información sobre software libre para traductores.

Sin embargo, en cuanto a herramientas, se encontró que la página no ha sido actualizada y solo se incluyen enlaces a cuatro programas libres de memorias de traducción, a saber, OmegaT, ForeignDesk, Open Language Tools y Transolution.

Otra fuente utilizada fue la publicación quincenal *The Tool Kit – A computer newsletter for translation professionals* (Zetzsche, s. f.). En esta publicación se presenta información de actualidad sobre las tecnologías aplicadas a la traducción y con frecuencia se incluyen reseñas de herramientas con su descripción y el enlace a su página web. Aunque en muchos casos se trata de programas privativos comerciales, de cuando en cuando también se incluye alguna herramienta libre. Para la primera fase de este trabajo se revisaron las ediciones publicadas de diciembre de 2007 a diciembre de 2010.

Por otro lado, está la página de Sandrini (<http://homepage.uibk.ac.at/~c61302/en-index.html>), profesor de traducción de la Universidad de Innsbruck y creador de la distribución de GNU/Linux *Tuxtrans*, una distribución basada en Ubuntu y creada específicamente para traductores. En la página web se encuentra la lista de las aplicaciones incluidas en dicha distribución. Sandrini también ofrece un paquete de aplicaciones gratuitas y portátiles pensado para el traductor, denominado *USBTrans*, que incluye, entre otras aplicaciones más generales, un paquete ofimático, un sistema de memorias de traducción, una herramienta de alineación, dos programas de análisis textual, una herramienta para gestión de proyectos y un gestor de bases de datos terminológicas. Tanto de la distribución *Tuxtrans* como del paquete de aplicaciones portátiles se incluyeron las herramientas que cumplen con los criterios establecidos para la selección.

El libro *Traducir (con) software libre* (Díaz Fouces & García González, 2008), editado por miembros del Grupo de Estudios das Tecnoloxías Libres da Tradución – GETLT – de la Universidad de Vigo, fue otra fuente utilizada para este trabajo, pues proporciona un panorama general sobre las diferentes posibilidades que ofrece el software libre para la traducción profesional y la enseñanza de la traducción. En especial, dos de los capítulos presentan una selección de software libre para traductores, uno sobre herramientas para el sistema operativo Windows

(Cánovas & Samson, 2008b), y otro sobre herramientas para GNU/Linux y Mac OS X (Díaz Fouces, 2008).

Finalmente, también se hicieron búsquedas en el portal SourceForge, el mayor repositorio de software libre en la actualidad. Aunque muchos de los programas de software libre encontrados en las otras fuentes están alojados en SourceForge, a la hora de hacer las búsquedas directamente en el buscador disponible en el sitio el problema que se encontró fue precisamente la abundancia de proyectos y la dificultad para filtrar los resultados según categorías, ya que en muchos casos las herramientas para traductores no encajan en las clasificaciones tradicionales de software y aparecen clasificadas simplemente como “otros”. Como filtro para las búsquedas se utilizó el estado de desarrollo del proyecto, limitando los resultados a beta, producción/estable y maduro.

### 3.2.5 Recolección de datos

Para el proceso de recolección de datos, se decidió que a medida que se fueran encontrando los programas, se visitaría su página web y se verificaría que cumplieran con los requisitos mínimos antes mencionados según el tipo de licencia, el estado de desarrollo y la categoría a la que corresponden. Para los programas seleccionados, se buscó entonces en la página web la información necesaria para rellenar la ficha. En la tabla 3.4 se incluye la ficha de uno de los programas a modo de ejemplo.

<b>Anaphraseus</b>	
<b>Descripción</b>	Anaphraseus is a CAT (Computer Aided Translation) tool, OpenOffice.org 2 macro set similar to famous Wordfast. Works with Wordfast Translation Memory format (*.TXT). Supports text segmentation. Features: Term Recognition. Fuzzy Search. Unicode support.
<b>Versión</b>	2.04 beta
<b>Sistemas operativos</b>	Windows, GNU/Linux, Mac OS X
<b>URL</b>	<a href="http://anaphraseus.sourceforge.net/">http://anaphraseus.sourceforge.net/</a>
<b>Licencia</b>	GNU General Public License
<b>Categoría</b>	Herramientas de traducción
<b>Requisitos</b>	OpenOffice.org o LibreOffice

Tabla 3.4: Ficha de uno de los programas del catálogo.

### 3.2.6 Resultados de la búsqueda, selección y clasificación de los programas

El catálogo de programas libres para el traductor incluye a la fecha un total de 188 programas. El listado completo se presenta en el Anexo 1 (p. 255) en cuatro tablas organizadas alfabéticamente, donde cada tabla corresponde a una de las cuatro categorías consideradas. Por motivos de espacio, y ya que el catálogo está disponible en Internet (ver apartado 3.2.7.1), en este listado solo se incluyeron los nombres de los programas, las tipologías a las que corresponden y las páginas web de los proyectos.

En cuanto a los tipos de licencias, la mayoría de los programas encontrados (122) utilizan la licencia GNU General Public License y 63 utilizan otra licencia libre (ej. GNU LGPL, Mozilla Public License, Apache License, etc.) o una combinación de varias de ellas. Finalmente, 3 de los programas encontrados no incluyen información sobre la licencia, tal vez por tratarse de *scripts*<sup>33</sup>; estas utilidades se incluyeron en el catálogo puesto que aparecen en la página de recursos del proyecto libre OmegaT y fueron desarrolladas como contribuciones para el mismo, por lo que también son de libre distribución. La tabla 3.5 a continuación presenta el desglose de los programas del catálogo según el tipo de licencia.

Tipo de licencia	Número de programas
GNU General Public License	122
Otra licencia libre o una combinación de licencias libres	63
Sin información sobre licencia	3
<b>Total</b>	<b>188</b>

Tabla 3.5: Programas del catálogo desglosados según el tipo de licencia.

En cuanto a su clasificación, de los 188 programas seleccionados, 46 pertenecen a la categoría de herramientas de edición y maquetación, 29 a herramientas lingüísticas, 84 a herramientas de traducción y 29 a herramientas de

---

<sup>33</sup> En términos generales, el término *script* se refiere a una serie de comandos incluidos en un archivo de texto que no requiere compilación y que se ejecuta de principio a fin sin interacción con el usuario. Los *scripts* son creados y utilizados principalmente por desarrolladores o usuarios con ciertos conocimientos de programación y normalmente no están destinados a usuarios finales.

gestión. En la figura 3.1 se presenta el desglose de los programas según la categoría a que corresponden.

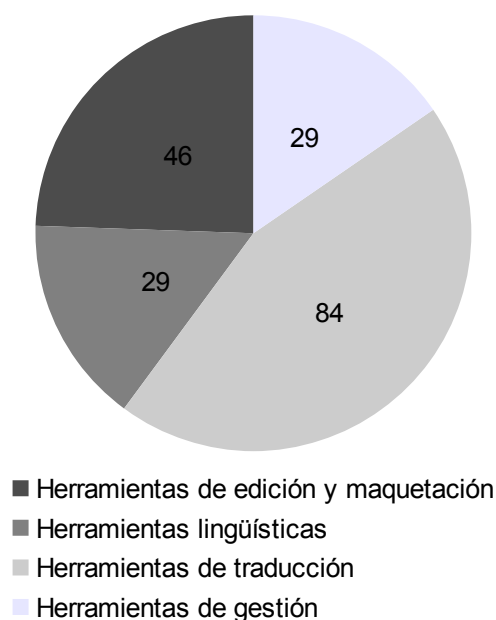


Figura 3.1: Programas del catálogo desglosados por categoría.

Es importante aclarar que el catálogo no se considera exhaustivo, ya que de los programas más generales como editores de texto o herramientas de gestión de proyectos existen muchas herramientas disponibles. En vista de lo anterior, se procuró más exhaustividad para las categorías relacionadas directamente con la traducción (herramientas lingüísticas y herramientas de traducción).

En cuanto a los sistemas operativos soportados, 153 de los programas del catálogo son multiplataforma, 19 están disponibles solo para sistemas operativos Windows, 16 solo para GNU/Linux y ninguno de los programas es exclusivo para Mac OS X. La tabla 3.6 presenta el desglose de los programas del catálogo según los sistemas operativos soportados.

<b>Sistema operativo</b>	<b>Número de programas</b>
Multiplataforma	153
Solo Windows	19
Solo GNU/Linux	16
Solo Mac OS X	0
<b>Total</b>	<b>188</b>

Tabla 3.6: Programas del catálogo desglosados por sistema operativo soportado.

### 3.2.7 Difusión del catálogo

El catálogo de software libre para traductores está disponible en Internet y ha sido presentado en dos artículos publicados en revistas especializadas. En este apartado explicamos la difusión que se ha dado a los resultados de esta primera fase del trabajo.

#### 3.2.7.1 El catálogo en Internet

Con la idea de favorecer la difusión del software libre entre los traductores, las fichas del catálogo se incorporaron a la base de datos del Centro Virtual (Tecnolettra, 2008). El formulario de consulta del Centro Virtual, disponible en <http://tecnolettra.uji.es/ott/virta.html>, permite delimitar las búsquedas por varios criterios: nombre del producto, tipo (herramienta o recurso), acceso (local o en línea), categoría, descripción, precio (gratuito o no gratuito), licencia (libre o no libre), empresa, distribuidor y requisitos técnicos. En la figura 3.2 se muestra una captura de pantalla del formulario de consulta.

En consonancia con los principios del movimiento del software libre, en abril de 2011 creamos también una *wiki*, donde de manera abierta y colaborativa se seguirá actualizando el catálogo con nuevos programas y con más información sobre los programas ya recopilados. Inicialmente en inglés para llegar a un mayor público, la wiki está disponible en <http://traduccionmundolibre.com/wiki>. La wiki se basa en el programa libre MediaWiki, el mismo en el que se basa la Wikipedia. Se escogió este programa por su gran comunidad de usuarios y desarrolladores, por los numerosos módulos y extensiones disponibles para extender su funcionalidad y por ser relativamente fácil de instalar y gestionar.

Formulario de Consulta

Nombre del producto:

Tipo:

Acceso:

Categoría:  Equipo básico  
 Comunicación y documentación  
 Edición y maquetación  
 Gestión de formatos  
 Herramientas lingüísticas  
 Herramientas de traducción: TAO  
 Herramientas de traducción: TA  
 Gestión de proyectos y finanzas

Descripción:

Precio:

Licencia:

Empresa:

Distribuidor:

Requisitos técnicos:  Cualquiera  
 Windows  
 Linux  
 Mac

Figura 3.2: Formulario de consulta del Centro Virtual del Observatorio de Tecnologías de la Traducción.

Para facilitar la búsqueda y navegación en el catálogo se instaló la extensión Semantic MediaWiki que además de permitir el ingreso y presentación de la información mediante plantillas estandarizadas, permite etiquetar los datos con anotaciones semánticas. Así pues, al importar los datos desde el gestor de referencias Zotero, cada uno de los campos se convirtió en una propiedad semántica<sup>34</sup>, de modo que es posible explorar el catálogo por categorías (edición y maquetación, herramientas lingüísticas, herramientas de traducción, herramientas de gestión), tipologías y propiedades (sistema operativo, licencia, etc.). En la figura 3.3 se muestra el formato que se utiliza para presentar las herramientas, en este caso OmegaT, mientras que la figura 3.4 presenta los resultados de una búsqueda por propiedades. Para agregar y editar los datos se puede utilizar la sintaxis *wiki*

<sup>34</sup> Para establecer las propiedades semánticas se utilizaron como guía los esquemas de metadatos sobre proyectos de software desarrollados por Dillon & Simmons (2010), Dumbill (s. f.), y la Unión Europea (European Union, s. f.).



tradicional o un formulario creado para facilitar el trabajo de los posibles colaboradores (ver figura 3.5).

Figura 3.3: Formato utilizado en la wiki para presentar las herramientas.

### 3.2.7.2 Artículos publicados

Los resultados de esta primera fase del trabajo empírico han sido publicados en dos artículos. En un primer artículo, titulado *Free/Open Source Software for the Translation Classroom* (Flórez & Alcina, 2011a) y publicado en la revista *The Interpreter and Translator Trainer*, se aborda el tema de la integración de las tecnologías en los programas de formación de traductores y la relevancia del software libre en este contexto. Además, se explica la metodología utilizada para la

recopilación del catálogo y se incluye el listado de los programas con indicación de la tipología a la que corresponden y la dirección de la página web de cada proyecto. Finalmente, en el artículo se presenta también la suite de aplicaciones libres y portátiles *PortableCAT*<sup>35</sup>, un proyecto derivado de este trabajo con el que se pretende promover el uso de software libre entre docentes y estudiantes de traducción. Creada como parte del material docente del módulo de Software libre para la traducción del Máster en Tecnologías de la Traducción y Localización, esta suite portátil se puede utilizar en el aula de traducción, adaptándola según la asignatura, y a la vez constituye una manera de animar a los estudiantes a explorar las tecnologías por fuera del aula de manera gratuita y legal.

## Typology Translation environment

---

A list of all pages that have property "Typology" with value "Translation environment"

- [Anaphraseus](#) +
- [Autshumato ITE](#) +
- [EsperantiloTM](#) +
- [ForeignDesk](#) +
- [Frankenstein Translation Memory Suite](#) +
- [OmegaT](#) +
- [OmegaTplus](#) +
- [Open Language Tools](#) +
- [OpenTM2](#) +
- [Transolution](#) +
- [Virtaal](#) +

Property:  Value:

Figura 3.4: Búsqueda en el catálogo por propiedades semánticas y valores.

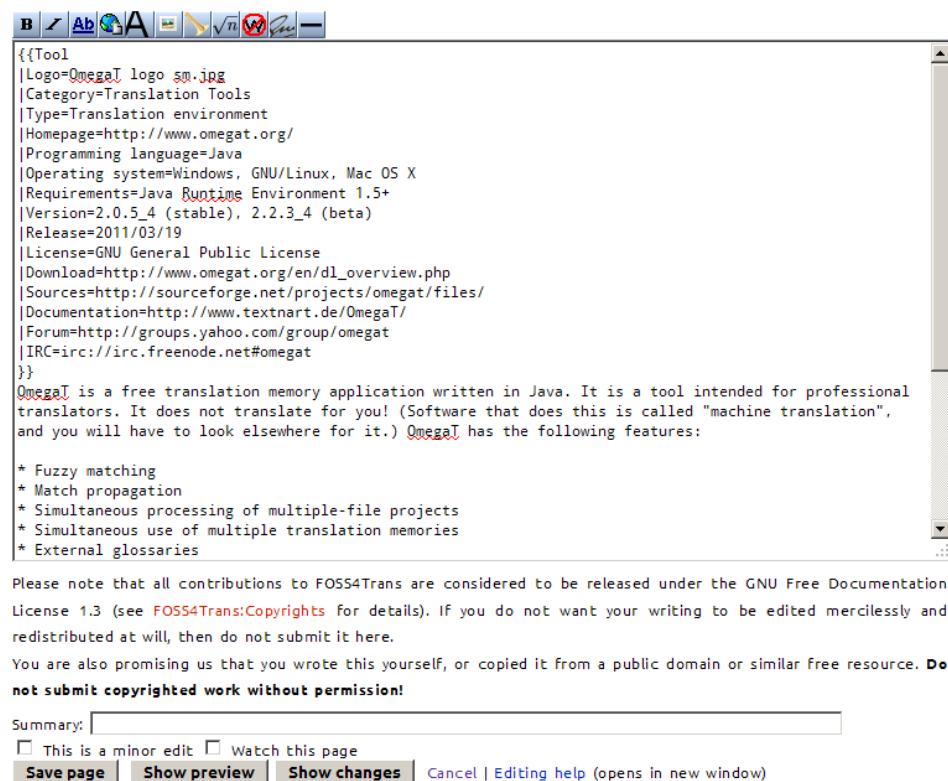
El segundo artículo, titulado *Catálogo de software libre para la traducción* (Flórez & Alcina, 2011b), fue publicado en la revista Tradumática y está enfocado a los profesionales de la traducción. En el artículo se presenta una selección de veintisiete herramientas libres y se remite a los lectores a visitar la *wiki* donde se encuentra disponible el catálogo completo. A modo de introducción, se exponen ciertas características distintivas del modelo de desarrollo y distribución del software libre, como el marco ideológico de los proyectos y la importancia de la

---

35 Disponible en <http://traduccionmundolibre.com/wiki/PortableCAT.html>

comunidad de usuarios y desarrolladores que los respalda, con el objetivo de que los traductores profesionales se familiaricen con las diferencias entre el software privativo y el software libre y sepan qué aspectos deben tener en cuenta a la hora de considerar este último.

## OmegaT



The screenshot shows a wiki editor interface with a toolbar at the top. The main editing area contains the following content:

```
{{Tool
|Logo=OmegaT logo sm.jpg
|Category=Translation Tools
|Type=Translation environment
|Homepage=http://www.omegat.org/
|Programming language=Java
|Operating system=Windows, GNU/Linux, Mac OS X
|Requirements=Java Runtime Environment 1.5+
|Version=2.0.5.4 (stable), 2.2.3.4 (beta)
|Release=2011/03/19
|License=GNU General Public License
|Download=http://www.omegat.org/en/dl_overview.php
|Sources=http://sourceforge.net/projects/omegat/files/
|Documentation=http://www.textnart.de/OmegaT/
|Forum=http://groups.yahoo.com/group/omegat
|IRC=irc://irc.freenode.net#omegat
}}
```

OmegaT is a free translation memory application written in Java. It is a tool intended for professional translators. It does not translate for you! (Software that does this is called "machine translation", and you will have to look elsewhere for it.) OmegaT has the following features:

- \* Fuzzy matching
- \* Match propagation
- \* Simultaneous processing of multiple-file projects
- \* Simultaneous use of multiple translation memories
- \* External glossaries

Please note that all contributions to FOSS4Trans are considered to be released under the GNU Free Documentation License 1.3 (see FOSS4Trans:Copyrights for details). If you do not want your writing to be edited mercilessly and redistributed at will, then do not submit it here.

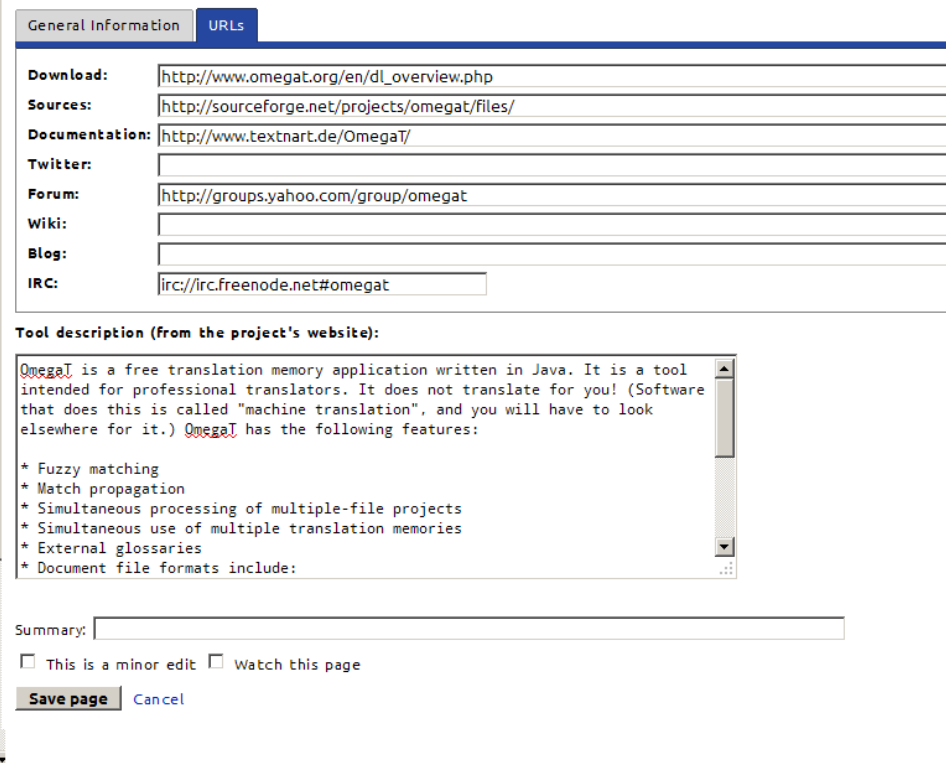
You are also promising us that you wrote this yourself, or copied it from a public domain or similar free resource. **Do not submit copyrighted work without permission!**

Summary:

This is a minor edit  Watch this page

**Save page** **Show preview** **Show changes** [Cancel](#) | [Editing help](#) (opens in new window)

## Edit Tool: OmegaT



The screenshot shows a form-based editor interface with two tabs: "General Information" and "URLs". The "URLs" tab is active and contains the following fields:

Download:	<input type="text" value="http://www.omegat.org/en/dl_overview.php"/>
Sources:	<input type="text" value="http://sourceforge.net/projects/omegat/files/"/>
Documentation:	<input type="text" value="http://www.textnart.de/OmegaT/"/>
Twitter:	<input type="text"/>
Forum:	<input type="text" value="http://groups.yahoo.com/group/omegat"/>
Wiki:	<input type="text"/>
Blog:	<input type="text"/>
IRC:	<input type="text" value="irc://irc.freenode.net#omegat"/>

**Tool description (from the project's website):**

OmegaT is a free translation memory application written in Java. It is a tool intended for professional translators. It does not translate for you! (Software that does this is called "machine translation", and you will have to look elsewhere for it.) OmegaT has the following features:

- \* Fuzzy matching
- \* Match propagation
- \* Simultaneous processing of multiple-file projects
- \* Simultaneous use of multiple translation memories
- \* External glossaries
- \* Document file formats include:

Summary:

This is a minor edit  Watch this page

**Save page** **Cancel**

Figura 3.5: A la izquierda: edición de una página del catálogo mediante sintaxis wiki; a la derecha, edición de la misma página mediante formularios.

### **3.3 Fase 2: Definición del método de evaluación**

La segunda fase del trabajo empírico consistió en definir un método de evaluación para las tecnologías libres para la traducción. Se entiende que un método de evaluación comprende un modelo de calidad y las directrices para su utilización, es decir, las actividades, tareas y participantes en el proceso de evaluación y el uso esperado de los resultados.

Partiendo de una perspectiva interdisciplinar que incluye aspectos tecnológicos, sociológicos y de negocios, para la evaluación se adoptó un enfoque cualitativo, ya que el principal interés era describir las características del ecosistema de las tecnologías libres para la traducción y explorar la viabilidad de los programas disponibles en la actualidad, más que llegar a generalizaciones sobre este tipo de software.

#### **3.3.1 Objetivo**

El objetivo del método de evaluación desarrollado es facilitar a los traductores la selección de herramientas libres para integrarlas en su entorno de trabajo.

#### **3.3.2 Usuarios previstos**

Los usuarios previstos de los resultados de la evaluación y de la interfaz de selección de tecnologías libres para la traducción son los traductores independientes, equipos de traducción, pequeñas empresas, investigadores, estudiantes y docentes de traducción.

#### **3.3.3 Perfil del evaluador**

El evaluador debe estar medianamente familiarizado con las tecnologías de la traducción y con las principales plataformas de gestión y desarrollo de software libre. Para la recolección de datos es necesario visitar las páginas web de los

proyectos, inspeccionar los foros, grupos de usuarios, wikis y demás repositorios públicos de los proyectos, utilizar motores de búsqueda y rellenar formularios web.

Para utilizar la interfaz de selección de programas el traductor como usuario final debe estar en capacidad de indicar sus requisitos básicos en términos del entorno previsto de uso, los formatos de trabajo y las tareas que necesita realizar.

### **3.3.4 Actividades y pasos**

El método de evaluación comprende tres actividades principales que se dividen a su vez en una serie de pasos según se detalla a continuación:

**Preparación de la evaluación:** consiste en definir el tipo de pruebas y el modelo de calidad (las categorías y criterios a considerar y las métricas y procedimientos de consolidación de resultados) y en diseñar e implementar los instrumentos. Preparar la evaluación ha sido uno de los objetivos del presente trabajo y se explica en lo que resta del apartado 3.3.

**Evaluación:** consiste en determinar la muestra de proyectos a evaluar y recolectar los datos mediante la aplicación del cuestionario, con lo que se generan automáticamente las fichas con los resultados. Esta actividad corresponde a otro de los objetivos del trabajo y se detalla en el apartado 3.4.

**Selección:** consiste en especificar los requerimientos del usuario (entorno existente, formatos de trabajo y módulos funcionales según las tareas a realizar), comparar los programas que cumplen dichos requerimientos y escoger el más apropiado.

Como vimos en el capítulo 2, en una evaluación de calidad pueden participar diferentes actores con distintos perfiles. En el presente trabajo, la investigadora estará encargada de preparar la evaluación y llevarla a cabo, pero no se contempla la selección de herramientas como tal. Para esta última actividad se implementa una interfaz de búsqueda para facilitar a los usuarios finales la selección de herramientas. Aunque el hecho de que la misma persona prepare la evaluación y la

lleve a cabo se podría ver como una limitación del presente trabajo, esta decisión estuvo fundamentada en la necesidad de que el evaluador esté familiarizado con el software libre y las tecnologías utilizadas para su desarrollo de modo que pueda recoger los datos necesarios para la caracterización de los proyectos.

### **3.3.5 El modelo de calidad**

Para definir el modelo de calidad de software se empezó por definir el tipo de prueba a utilizar y por delimitar el contexto de la evaluación. Teniendo en cuenta que la motivación para la evaluación del software en este caso era comprobar las características generales de los programas para su posible implementación en el entorno de trabajo del traductor, se decidió utilizar el tipo de pruebas denominado *inspección de características*, que se ocupa solamente de indicar la presencia o ausencia de determinadas características y no de identificar errores en los programas. Se optó por este tipo de pruebas por su carácter descriptivo y por su sencillez, rapidez y facilidad de aplicación, ya que los datos necesarios se pueden obtener en gran medida a partir de la documentación de los programas y las páginas web de los proyectos. Esto facilitaría luego la integración tanto del instrumento como de los resultados de la evaluación en la *wiki* creada en la primera fase de este trabajo.

En cuanto a los escenarios de uso de los programas, el método de evaluación abarca el uso tanto en la dimensión profesional de la traducción como para investigación y enseñanza/aprendizaje de la traducción y tanto para uso individual como para trabajo colaborativo. Además, se entiende que en cada una de estas dimensiones los usuarios pueden asumir diferentes roles, como traductor, revisor, terminólogo, gestor de recursos lingüísticos o gestor del proyecto, entre otros.

Desde el punto de vista tecnológico, el entorno de uso previsto para los programas es en este caso un entorno normal, pues se asume que el software solo se utiliza internamente y la implementación de soluciones provisionales para los posibles problemas no tiene mayores consecuencias. Se asume también que por lo

general es el mismo usuario quien se encarga de la instalación, actualización y mantenimiento de los programas.

Desde el punto de vista sociológico y de negocios, se consideraron los perfiles que no implican necesariamente participación en la comunidad ni un modelo de negocio basado directamente en el software libre, es decir, usuario de la aplicación e integración de componentes en la infraestructura informática del negocio. En el caso de la traducción, esto se refiere al uso por parte de profesionales, docentes y estudiantes o investigadores; en el caso de pequeñas empresas se considera solamente la utilización del software a nivel interno. La integración en productos o servicios basados en el software, la participación activa y la gestión o lanzamiento de comunidades, por otro lado, son casos de uso que no abordaremos en este trabajo, ya que corresponden a modelos de negocio basados directamente en software libre y requieren por tanto un nivel de evaluación más detallado.

### ***3.3.5.1 Categorías y criterios***

En la jerarquía para definir los criterios de evaluación partimos de la distinción entre proyecto y producto. El modelo de calidad se compone pues de dos partes: la primera permite la caracterización de los proyectos de desarrollo para tener una mejor comprensión de las prácticas y procesos involucrados y de los recursos y servicios disponibles para la comunidad de usuarios (apartado 3.3.5.1.1); la segunda parte corresponde a la calidad del software como producto y permite determinar las funciones y características técnicas de los programas (apartado 3.3.5.1.2).

#### **3.3.5.1.1 Calidad del proyecto**

Con el fin de caracterizar los proyectos de desarrollo de las tecnologías libres para la traducción, con base en lo encontrado en la literatura y siguiendo la recomendación de ir de lo general a lo específico, se definieron cuatro características a tener en cuenta: la estrategia, la comunidad, la madurez y la reputación del proyecto.



Para determinar la *estrategia del proyecto* se tuvieron en cuenta aspectos como el marco ideológico, la gestión de los derechos de propiedad intelectual, la planeación a futuro, las estructuras de comunicación y toma de decisiones y la colaboración con otros proyectos libres. En relación con la *comunidad del proyecto*, por otro lado, se consideraron aspectos como la capacidad de mantenimiento, su sostenibilidad y los recursos y servicios disponibles. Para la *madurez del proyecto*, se tuvieron en cuenta el estado actual de cada proyecto, el uso de tecnologías apropiadas para la gestión del proyecto y la manera de gestionar las versiones. Finalmente, para determinar la *reputación del proyecto* se tuvieron en cuenta el grado de adopción, la popularidad del proyecto y la satisfacción de los usuarios. La descomposición de la calidad del proyecto en características y subcaracterísticas se presenta en la figura 3.6.

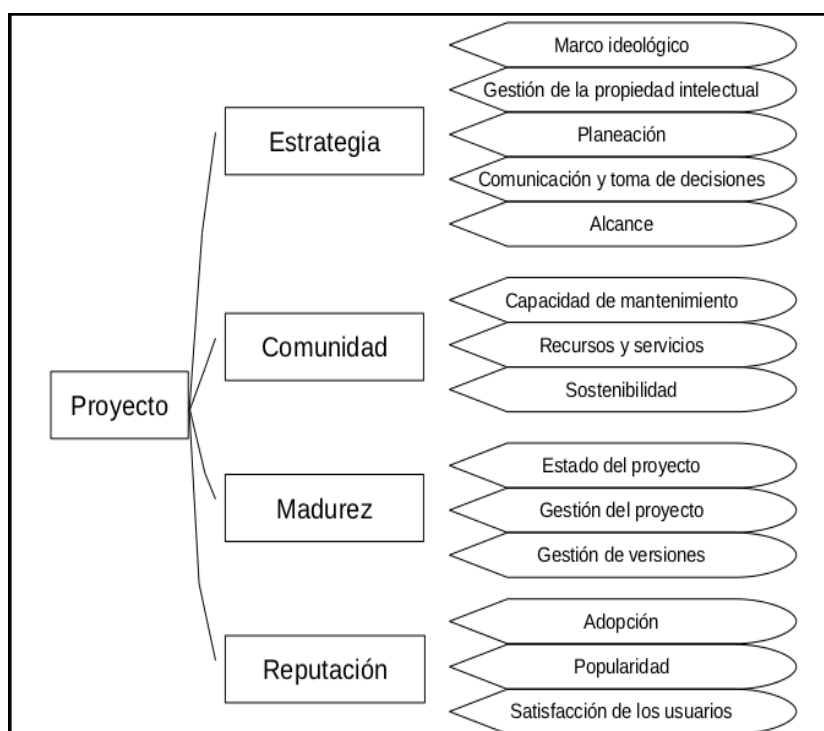


Figura 3.6: Características y subcaracterísticas de la calidad del proyecto.

### 3.3.5.1.2 Calidad del producto

Teniendo en cuenta la motivación de la evaluación y la orientación funcional de los programas, como criterios para la evaluación del software como producto se

utilizaron tres de las seis características propuestas en el estándar ISO 9126, a saber, funcionalidad, usabilidad y portabilidad.

Se incluyó la portabilidad puesto que a la hora de seleccionar programas para un entorno específico es necesario considerar su compatibilidad con dicho entorno y la facilidad para implementar los programas. Se decidió incluir la usabilidad puesto que la evaluación se realiza desde el punto de vista de los usuarios finales, en este caso traductores, por lo que los aspectos relacionados con la facilidad para comprender, aprender y utilizar el programa resultan relevantes. Se tuvo también en cuenta la funcionalidad puesto que las funciones que ofrece un programa determinan en gran medida su idoneidad para realizar tareas específicas; de igual manera, la configurabilidad de las funciones y la interoperabilidad que ofrece un programa, también subcaracterísticas de la funcionalidad, pueden resultar determinantes en ciertos contextos de uso. En la figura 3.7 se presentan las características y subcaracterísticas de la calidad del producto que se tuvieron en cuenta.

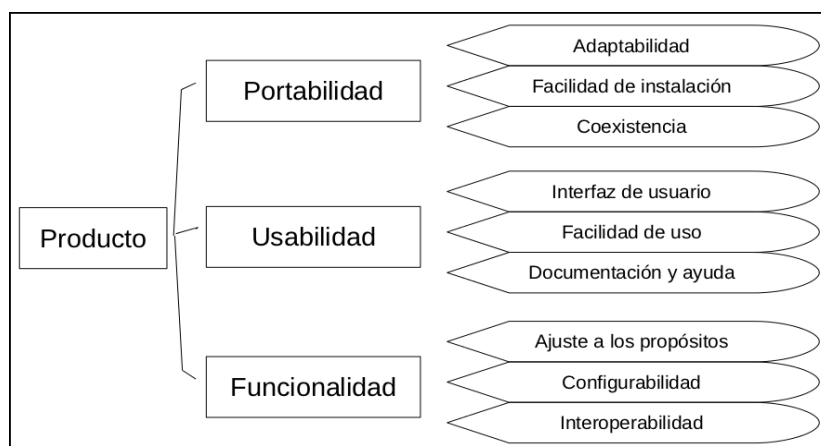


Figura 3.7: Características y subcaracterísticas de la calidad del producto.

Dado el alcance de este proyecto, las tres características restantes propuestas en el estándar ISO 9126 (fiabilidad, mantenibilidad y eficiencia) no se incluyeron en el modelo de evaluación. Como se explicó en el apartado 2.4, evaluar dichas características implica conocimientos y pruebas técnicas avanzadas que sobrepasan los recursos humanos, técnicos y económicos que se presuponen para el tipo de evaluación planteado en este trabajo.

Así pues, para evaluar la portabilidad de un programa se tuvieron en cuenta su arquitectura y escalabilidad, el grado de dificultad de su instalación y la compatibilidad con el entorno existente. Para la evaluación de la usabilidad, por otro lado, se consideraron aspectos relacionados con la interfaz de usuario, la facilidad de uso y la documentación y ayuda de los programas.

En este punto es importante recordar que los atributos correspondientes a la portabilidad y la usabilidad son igualmente relevantes para cualquier tipo de herramienta. En otras palabras, son criterios no funcionales que se pueden aplicar tanto a un navegador web como a un programa de ofimática o a una herramienta de traducción. Los atributos de la característica de funcionalidad, en cambio, varían según el tipo de herramienta a evaluar y las tareas que se pueden realizar con ella (alineación, traducción, revisión, facturación, etc.). Para evaluar la funcionalidad de las herramientas se tuvieron en cuenta las subcaracterísticas de ajuste a los propósitos, configurabilidad e interoperabilidad.

Como vimos en el apartado 3.2.3, en el catálogo de software libre para traductores recopilado en la primera fase se incluyeron cuatro categorías de programas: edición y publicación, herramientas lingüísticas, herramientas de traducción y herramientas de gestión. El modelo de calidad preparado para este trabajo se limita al análisis de la funcionalidad de los sistemas de memorias de traducción de escritorio, pertenecientes a la categoría *herramientas de traducción* (ver apartado 1.2 sobre la clasificación de las herramientas).

#### **3.3.5.2 Atributos y métricas**

El siguiente paso consistió en descomponer cada una de las características y subcaracterísticas de calidad en uno o más atributos. En el caso de las características de calidad del proyecto, se optó por una evaluación cualitativa. Esto quiere decir que para estos atributos no se definieron puntuaciones cuantitativas; en cambio, la información factual se presenta directamente en las fichas de resultados de modo que los usuarios puedan ampliar su conocimiento sobre cada proyecto. Para las características no funcionales de calidad del producto (portabilidad y usabilidad), por otro lado, se definieron los atributos y las métricas correspondientes, es decir, la forma como se obtienen las puntuaciones cuantitativas

y las escalas a utilizar en cada caso. Para la funcionalidad, finalmente, se definió una lista de comprobación que permite indicar las características presentes, pero no se utilizaron puntuaciones ni se hacen valoraciones sobre las funciones implementadas.

Como vimos en el apartado 3.3.5.1.1, para caracterizar los proyectos se tuvieron en cuenta cuatro características de calidad (estrategia, comunidad, madurez y reputación) que se descomponen a su vez en una serie de subcaracterísticas. Así, la estrategia del proyecto incluye aspectos como el marco ideológico del proyecto, la gestión de la propiedad intelectual, la planeación a futuro, las estructuras de comunicación y toma de decisiones y el alcance del proyecto.

Para cada una de estas subcaracterísticas se definieron entonces los atributos a considerar y las posibles opciones para cada uno de ellos. Para la subcaracterística *gestión de la propiedad intelectual*, por ejemplo, se definieron los siguientes atributos: estrategia general de licenciamiento, permisividad de la licencia, existencia de guías y acuerdos de cesión de derechos para los miembros colaboradores y titularidad de los derechos de autor. Para valorar el atributo *estrategia general de licenciamiento* se definieron las siguientes opciones de respuesta:

- a) Una licencia libre: la herramienta y todas las funciones asociadas están disponibles bajo una sola licencia libre.
- b) Varias licencias libres: la herramienta y todas las funciones asociadas están disponibles utilizando una combinación de licencias libres.
- c) Licenciamiento dual: la herramienta está disponible bajo una licencia libre y una privativa.
- d) Núcleo abierto: el producto central es libre, pero existe una versión con funciones adicionales disponible bajo una licencia privativa.

En las tablas a continuación se presentan los atributos definidos para evaluar la **estrategia** (tabla 3.7), la **comunidad** (tabla 3.8), la **madurez** (tabla 3.9) y la **reputación** de los proyectos (tabla 3.10) y las posibles opciones establecidas para cada atributo. Como se aprecia en las tablas, algunos atributos son de tipo binario (presencia/ausencia), otros clasificatorios y otros numéricos. Para definir los atributos se tuvo en cuenta la disponibilidad de los datos, de modo que fuera posible

obtener la información necesaria directamente desde las páginas web, forjas y documentación de los proyectos.

<b>Estrategia del proyecto</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Opciones</b>
Marco ideológico del proyecto	Origen del proyecto	Proyecto independiente Proyecto con financiación pública Proyecto con financiación privada Proyecto con financiación mixta
	Tipo de ética que rige el proyecto	Ética <i>hacker</i> Ética híbrida Ética empresarial
Gestión de la propiedad intelectual	Estrategia general de licenciamiento	Una licencia libre Varias licencias libres Licenciamiento dual (libre/privativo) Núcleo abierto
	Permisividad de la licencia	Sin <i>copyleft</i> Con <i>copyleft</i> débil Con <i>copyleft</i> fuerte
	Guías o acuerdos de cesión de derechos para los colaboradores	Presencia Ausencia
	Titularidad de los derechos de autor	El titular es un único desarrollador Titularidad asignada a una entidad legal Titularidad distribuida
Planeación a futuro	Especificación de requerimientos	Presencia Ausencia
	Hoja de ruta	Presencia Ausencia
	Descripción de nuevas funciones planeadas	Presencia Ausencia
	Plan de versiones	Presencia Ausencia
Estructuras de comunicación y toma de decisiones	Tipo de proceso para la toma de decisiones	Descentralizado Balanceado Centralizado
	Sistema de gobierno	Dictadura benevolente Meritocracia Democracia Anarquía
	Mecanismo de representación utilizado por el proyecto para comunicarse y ser identificado	Desarrollador original Líderes reconocidos Fundación Comité directivo Empresa o institución patrocinadora
Alcance	Integración de código de otros proyectos libres	Sí No
	Proyecto derivado de otro proyecto libre	Sí No
	Desarrollo de otras herramientas	Sí No

Tabla 3.7: Atributos para determinar la estrategia del proyecto.

<b>Comunidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Opciones</b>
Capacidad de mantenimiento	Tipo de comunidad de desarrollo	Desarrollador independiente Grupo de desarrolladores Desarrolladores organizados formalmente Entidad legal Entidad comercial
	Proyectos bifurcados ( <i>forks</i> ) o herramientas derivadas	Presencia Ausencia
	Instituciones vinculadas al proyecto	Presencia Ausencia
	Número de desarrolladores activos	Valor numérico
	Número de suscriptores en las listas de usuarios	Valor numérico
Sostenibilidad	Número de usuarios que participaron en las discusiones en el último mes	Valor numérico
	Promedio de mensajes por mes en el foro de usuarios en 2011	Valor numérico
	Tiempo promedio de respuesta en los foros (últimas 5 preguntas)	Valor numérico
Recursos y servicios disponibles	Portal web que recoja la información destacada del proyecto	Presencia Ausencia
	Espacios de comunicación utilizados activamente en el último año (listas de correo, <i>wiki</i> , blog, chat IRC, redes sociales)	Presencia Ausencia
	Asistencia técnica personalizada	Presencia Ausencia
	Suscripciones de valor añadido	Presencia Ausencia
	Formación (tutoriales, canal de video, webinars, etc.)	Presencia Ausencia
	Desarrollo personalizado	Presencia Ausencia
	Asesoría	Presencia Ausencia
	Software como servicio	Presencia Ausencia

Tabla 3.8: Atributos para caracterizar la comunidad del proyecto.

<b>Madurez del proyecto</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Opciones</b>
Estado del proyecto	Fecha de inicio del proyecto	Valor numérico
	Estado actual de desarrollo	Beta Estable Maduro Inactivo

<b>Madurez del proyecto</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Opciones</b>
Gestión del proyecto	Gestión del proyecto en una de las principales forjas públicas	Presencia Ausencia
	Repositorio de código fuente con sistema de control de revisiones	Presencia Ausencia
	Sistema de gestión de informes de posibles errores	Presencia Ausencia
	Sistema de gestión de solicitudes de nuevas funciones	Presencia Ausencia
	Existencia de procesos documentados para contribuir al proyecto	Presencia Ausencia
	Plataforma para la gestión de la localización del programa y la documentación	Presencia Ausencia
	Proceso documentado de elicitación y gestión de requisitos	Presencia Ausencia
Gestión de versiones	Ciclo de lanzamientos definido	Presencia Ausencia
	Versiones lanzadas en 2011	Valor numérico
	Actualizaciones menores lanzadas en 2011	Valor numérico
	Fecha de la última versión liberada	Valor numérico

Tabla 3.9: Atributos para determinar la madurez del proyecto.

<b>Reputación del proyecto</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Opciones</b>
Adopción	Libros, publicaciones, reseñas o entradas en blogs sobre el proyecto	Presencia Ausencia
	Implementación de referencia/casos de éxito documentados en la web del proyecto	Presencia Ausencia
	Promedio de descargas en la semana siguiente al lanzamiento de las tres últimas versiones	Valor numérico
Popularidad	Número de descargas en el último mes	Valor numérico
	Discusiones en foros de traductores (ProZ, LinkedIn, etc.)	Presencia Ausencia
	Paquetes incluidos en repositorios de GNU/Linux	Presencia Ausencia
	Proyecto incluido en catálogos o directorios de software	Presencia Ausencia
	Perfil del proyecto en Ohloh.net	Presencia Ausencia
Satisfacción de los usuarios	Reseñas y puntuaciones en la forja utilizada	Presencia Ausencia
	Comentarios sobre el proyecto en redes sociales	Presencia Ausencia

Tabla 3.10: Atributos para determinar la reputación del proyecto.

En cuanto a la calidad del producto, para las características no funcionales (**portabilidad** y **usabilidad**), cada subcaracterística se descompuso en una serie de atributos para los que se formularon una serie de posibles respuestas y las puntuaciones asociadas. Para estas dos características se decidió utilizar una escala homogénea de 1 a 3, donde 1 corresponde a inaceptable, 2 a aceptable y 3 a satisfactorio. Al redactar las posibles respuestas se procuró dar cuenta de las situaciones que se encuentran en casos reales de uso y se prestó especial atención a evitar la ambigüedad, de modo que se redujera la posibilidad de dar lugar a diferentes interpretaciones por parte de diferentes evaluadores en diferentes contextos. Las tablas 3.11 y 3.12 detallan los atributos para la evaluación de la portabilidad y la usabilidad del producto, respectivamente, con las tres opciones definidas para cada uno de ellos y las puntuaciones correspondientes.

Portabilidad		Puntuación		
Subcaracterística	Atributo	1	2	3
Adaptabilidad	Modularidad	El diseño de la herramienta no permite el desarrollo de componentes independientes.	El diseño de la herramienta permite el desarrollo de componentes independientes que se pueden integrar al sistema pero no hay documentación disponible.	El diseño de la herramienta permite el desarrollo de componentes independientes mediante una arquitectura de plugins o una API pública bien documentada.
	Escalabilidad	El sistema no está pensado para implementaciones a gran escala ni incluye modo multiusuario.	El sistema puede ser implementado a gran escala, pero no está pensado para entornos multiusuario o viceversa.	El sistema puede ser implementado a gran escala y en entornos multiusuario.
Facilidad de instalación	Tiempo necesario para la instalación	Más de 30 minutos.	Entre 15 y 30 minutos.	Menos de 15 minutos.
	Requisitos de sistema	Es necesario instalar otro(s) programa(s) para poder utilizar la herramienta.	Los programas necesarios se incluyen en el paquete de instalación.	El programa no depende de software de terceros.
	Conocimientos técnicos necesarios para la instalación	Es necesario realizar procedimientos avanzados para la instalación o configuración (ej. compilación de fuente, configuración de	La instalación del programa es sencilla, pero es necesario realizar otras configuraciones adicionales que requieren algunos conocimientos	La instalación es sencilla y no hay que realizar configuraciones adicionales.



Portabilidad		Puntuación		
Subcaracterística	Atributo	1	2	3
		servidores o bases de datos, etc.).	informáticos (ej. configurar PATH o asignar memoria).	
Coexistencia	Compatibilidad con el entorno existente	El programa no puede ser utilizado en el entorno existente.	El programa puede ser utilizado en el entorno existente realizando algunas configuraciones especiales.	El programa puede ser utilizado en el entorno existente.
	Posibilidad de integración	El programa difícilmente se puede integrar al flujo de trabajo actual.	El programa se puede integrar al flujo de trabajo actual implementando algunas soluciones provisionales.	El programa se puede integrar fácilmente al flujo de trabajo actual.

Tabla 3.11: Atributos para evaluar la portabilidad del producto.

Usabilidad		Puntuación		
Subcaracterística	Atributo	1	2	3
Interfaz de usuario	Distribución de la interfaz de usuario	La interfaz es compleja con demasiada información sin una organización clara; es necesario utilizar el manual.	Toma un poco de tiempo entender la interfaz, la información está más o menos organizada; hay que usar un poco el manual.	La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.
	Disponibilidad en el idioma requerido	El programa y su documentación y ayuda solo están disponibles en un idioma distinto al requerido.	La localización es parcial (interfaz en el idioma requerido pero documentación sin traducir o viceversa).	El programa está totalmente localizado al idioma requerido, tanto la interfaz de usuario como las ayudas y otra documentación incluida.
	Comprensibilidad de los iconos y nombres de funciones	Algunos iconos o nombres de funciones y menús son difíciles de relacionar con la acción que realizan.	Es fácil relacionar los iconos y nombres de funciones y menús con las acciones que realizan.	Es fácil relacionar los nombres de funciones y menús con las acciones que realizan y se incluyen textos alternativos ( <i>tooltips</i> ) para explicar las funciones de los iconos.
	Configurabilidad de la apariencia	No es posible cambiar la configuración de la apariencia del programa.	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente,	Es posible realizar cambios en la configuración de apariencia (ej. tipo y tamaño de fuente,

Usabilidad		Puntuación		
Subcaracterística	Atributo	1	2	3
			colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	colores, etc.) y dichos cambios se pueden exportar.
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	La interacción depende en gran medida del uso del ratón.	El programa permite la navegación y operación con el teclado aunque para algunas acciones es necesario el ratón.	La interfaz está diseñada para permitir la navegación y operación solo con el teclado sin depender del ratón.
	Existencia de ayuda contextual	El programa no cuenta con ayuda accesible dentro de la interfaz; es necesario abrir un manual de usuario en otra aplicación o recurrir a otras fuentes (ej. página web o wiki).	El programa incluye una ayuda básica en la interfaz, pero para funciones avanzadas es necesario recurrir a otras fuentes (ej. manual o comunidad de usuarios).	El programa incluye ayuda sobre la mayoría de las funciones y es posible acceder a ella sin salir de la aplicación.
	Indicadores de progreso y mensajes de error	No se incluyen barras de progreso ni mensajes informativos, por lo que el usuario en ocasiones no sabe si el programa está ocupado realizando algún proceso o si ha ocurrido un fallo.	Se incluyen mensajes informativos o barras de progreso, pero en algunos casos el usuario no sabe si el programa está ocupado o si ha ocurrido un fallo.	Se incluyen barras de progreso, mensajes de error u otros mensajes informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.
Documentación	Existencia de varios tipos de documentación	Solo se dispone de un archivo de texto tipo Léame.txt o no hay documentación disponible.	Se dispone de un archivo Léame y por lo menos otra guía o manual de usuario.	El programa incluye varios tipos de documentación (guía rápida o de instalación, manual de usuario, manual para desarrolladores, preguntas frecuentes, solución de problemas, etc.).
	Actualización de la documentación	Toda la documentación se refiere a versiones anteriores del programa o no hay documentación.	Algunas partes de la documentación no corresponden a la versión actual del programa.	Toda la documentación está actualizada con la versión actual del programa.
	Exhaustividad de la documentación	La documentación es muy limitada o no hay documentación disponible.	La documentación no incluye información sobre algunas funciones importantes.	La documentación cubre de manera clara todas las funciones del programa.

Usabilidad		Puntuación		
Subcaracterística	Atributo	1	2	3
	Existencia de recursos multimedia	No se incluyen ayudas visuales ni recursos multimedia como parte de la documentación o no hay documentación.	La documentación incluye algunas capturas de pantalla u otros gráficos.	La documentación incluye capturas de pantalla y se dispone de videotutoriales o demos en línea.

Tabla 3.12: Atributos para evaluar la usabilidad del producto.

Para evaluar la **funcionalidad**, por otro lado, se consideraron las funciones incluidas, las posibilidades de configuración, la capacidad de procesar diferentes formatos de entrada y la interoperabilidad. Para las subcaracterísticas de *ajuste a los propósitos* y *configurabilidad*, se estableció una lista de comprobación (ver tabla 3.13) con las principales características que se espera encontrar en este tipo de herramientas con base en las descripciones funcionales de los principales sistemas privativos comerciales y en el conocimiento previo sobre los sistemas de memorias de traducción (ver apartado 1.3). Siguiendo esta misma línea, la lista de funciones y formatos soportados se puede ampliar fácilmente para abarcar otras tipologías de programas.

En cuanto a la subcaracterística de *interoperabilidad*, se consideró en primer término la posibilidad de importar y exportar datos en los formatos basados en los estándares abiertos de intercambio utilizados en la industria (TMX, TBX, SRX, etc.), así como la capacidad de procesar archivos en formatos abiertos creados por otras herramientas de traducción.

Para cada una de los atributos de la funcionalidad se indica la presencia o ausencia de la característica en cuestión, pero no se calculan puntuaciones ni se hacen valoraciones sobre la adecuada implementación de las funciones. En cambio, la lista completa de características presentes se incluirá en la ficha de resultados. La tabla 3.13 presenta en detalle los atributos que se utilizaron para evaluar la funcionalidad de los programas de la tipología *sistemas de memorias de traducción*.

Funcionalidad		Puntuación
Subcaracterística	Atributo	Presencia o ausencia
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de programa	<p>Opciones del proyecto:</p> <p>Análisis de originales (recuento de palabras, coincidencias, repeticiones)</p> <p>Procesamiento por lotes</p> <p>Pretraducción de documentos</p> <p>Pretraducción con priorización de fuentes utilizadas</p> <p>Pseudotraducción</p> <p>Creación de proyectos con múltiples documentos de partida</p> <p>Posibilidad de utilizar las memorias en ambas direcciones</p> <p>Múltiples memorias por proyecto</p> <p>Múltiples glosarios por proyecto</p> <p>Múltiples traducciones para el mismo segmento original</p> <p>Memorias multilingües (más de dos idiomas)</p> <p>Uso simultáneo de glosarios/memorias compartidas por red</p> <p>Coincidencias parciales</p> <p>Coincidencias basadas en contexto</p> <p>Coincidencias de glosario</p> <p>Inserción automática de coincidencias exactas</p> <p>Inserción automática de coincidencias parciales</p> <p>Propagación automática de segmentos repetidos</p>
		<p>Opciones del editor:</p> <p>Visualización de metadatos de las coincidencias (fecha, ID de usuario, proyecto, etc.)</p> <p>Validación de segmentos mediante diferentes estatus</p> <p>Opción de navegación en el editor mediante filtros</p> <p>Posibilidad de añadir comentarios a los segmentos</p> <p>Estadísticas del proyecto (segmentos traducidos/sin traducir)</p> <p>Búsqueda y reemplazo global</p> <p>Búsqueda de concordancias en archivos originales</p> <p>Búsqueda de concordancias en archivos de referencia</p> <p>Autocompletar al escribir</p> <p>Corrector ortográfico al escribir</p> <p>Corrector ortográfico bajo demanda</p> <p>Corrector gramatical / de estilo al escribir</p> <p>Corrector gramatical / de estilo bajo demanda</p> <p>Vista previa del formato</p> <p>Modo de revisión (control de cambios, comentarios, exportación a tabla)</p> <p>Controles de calidad al escribir</p> <p>Controles de calidad bajo demanda</p>
		<p>Integración con aplicaciones externas:</p> <p>Integración con motores de traducción automática locales o vía web</p> <p>Búsqueda en recursos externos (locales o vía servicios web)</p> <p>Integración con software de reconocimiento de voz (comandos y/o dictado)</p>
	Filtros de archivo implementados (ver listado detallado de formatos en la tabla 1.2 en la página 44)	<p>Formatos de texto y ofimática</p> <p>Formatos de maquetación</p> <p>Formatos multimedia</p> <p>Formatos de localización web</p> <p>Formatos de localización de software</p>
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	<p>Filtros configurables</p> <p>Reglas de segmentación configurables</p> <p>Posibilidad de cambiar la segmentación durante la traducción</p> <p>Porcentaje mínimo de coincidencias configurable</p> <p>Diccionarios del corrector ortográfico personalizables</p>

Funcionalidad		Puntuación
Subcaracterística	Atributo	Presencia o ausencia
		Reglas del corrector lingüístico personalizables Búsquedas y reemplazos basados en expresiones regulares Colocables y localizables configurables (fechas, variables, etc.) Controles de calidad configurables (etiquetas, puntuación, espacios, números, términos, etc.) Control de acceso al sistema mediante usuarios y permisos Atajos de teclado configurables
Interoperabilidad	Soporte para estándares de intercambio de datos	Codificación Unicode Reglas de segmentación SRX Memorias TMX Bases de datos TBX Glosarios como texto delimitado (CSV, TAB o TXT) Archivos pretraducidos XLIFF
	Soporte para formatos abiertos generados por otras herramientas de traducción	TTX (SDL Trados) TXT (WordFast) TXML (WordFast Pro) NXT (STAR Transit)

Tabla 3.13: Atributos para evaluar la funcionalidad del producto.

### 3.3.5.3 Procedimientos para la consolidación de resultados

Se definieron luego los procedimientos para resumir los datos de los atributos en puntuaciones globales por subcaracterística. Por tratarse de una evaluación exploratoria y general, todos los atributos y características se consideran igualmente importantes de modo que decidimos no utilizar pesos para ponderar los resultados pues no se parte de un contexto específico de evaluación que justifique la asignación de determinados valores. Además, el uso de diferentes escalas (binarias, clasificatorias y ordinales) hace que los promedios ponderados no resulten apropiados para la consolidación de resultados.

En cuanto a la calidad del proyecto, para las características *estrategia del proyecto* y *reputación* se decidió no resumir los resultados mediante indicadores por considerarse que estos aspectos no son determinantes para la selección de herramientas. En cambio, la información correspondiente a la estrategia del proyecto se presenta en las fichas de resultados a modo de párrafo descriptivo sobre los proyectos, mientras que los datos encontrados sobre la reputación de los mismos se incluyen en las fichas como enlaces de referencia para los interesados.

Para resumir los resultados de las otras dos características de calidad del proyecto (comunidad y madurez) se definieron los criterios de aceptación que se

presentan en la tabla 3.14. Si el proyecto cumple los criterios establecidos se le asigna una estrella para la subcaracterística correspondiente, de modo que el proyecto puede obtener un máximo de tres estrellas por característica. El número de estrellas obtenido se interpreta de la siguiente manera: 3 estrellas = satisfactorio, 2 estrellas = aceptable, 1 estrella = pobre, 0 estrellas = inaceptable. Además, se decidió que para los proyectos que no obtuvieran ninguna estrella para las características de comunidad y madurez no se llevaría a cabo la evaluación del software como producto.

<b>Característica</b>	<b>Subcaracterística</b>	<b>Criterios de aceptación</b>
Comunidad	Capacidad de mantenimiento	Por lo menos un desarrollador activo y un foro de usuarios con suscriptores.
	Sostenibilidad	Existencia de discusiones activas en el último mes y un promedio de mensajes por mes en el último año no inferior a cuatro.
	Recursos y servicios disponibles	Portal web con información relevante del proyecto, por lo menos dos espacios de comunicación donde los usuarios puedan obtener respuestas a sus inquietudes.
Madurez	Estado del proyecto	El proyecto debe tener por lo menos dos años de vida y su estado actual de desarrollo debe ser estable o maduro.
	Gestión del proyecto	El código debe ser gestionado en una forja pública con un sistema de control de revisiones y de gestión de informes de fallos.
	Gestión de versiones	El proyecto debe haber lanzado por lo menos una versión o actualización en 2011 y la última versión disponible debe ser de 2011 o 2012.

*Tabla 3.14: Indicadores de calidad de la comunidad y madurez del proyecto.*

En cuanto a la calidad del producto, para las características no funcionales (portabilidad y usabilidad) también se asignan estrellas por subcaracterística, pero en este caso para obtener las puntuaciones globales el procedimiento consiste simplemente en sumar las puntuaciones individuales de los atributos de cada subcaracterística y clasificar los resultados según la tabla 3.15.

Característica	Subcaracterística	Criterios de aceptación
Portabilidad	Adaptabilidad	Puntuación mínima igual o mayor que cuatro.
	Facilidad de instalación	Puntuación mínima igual o mayor que seis.
	Coexistencia	Puntuación mínima igual o mayor que cuatro.
Usabilidad	Interfaz de usuario	Puntuación mínima igual o mayor que ocho.
	Documentación	Puntuación mínima igual o mayor que seis.
	Facilidad de uso	Puntuación mínima igual o mayor que ocho.

Tabla 3.15: Indicadores de calidad para la portabilidad y la usabilidad.

Para la funcionalidad, finalmente, no se consolida la información sino que, como ya se explicó en el apartado anterior, en la ficha de resultados se presenta el listado de las funciones disponibles y los formatos de archivo soportados. Será en la fase de selección, mediante la interfaz diseñada para tal propósito, cuando cada usuario determine las funciones imprescindibles según su contexto de uso.

### 3.3.6 Instrumentos

El método de evaluación se implementó en la *wiki* como complemento del catálogo de software libre compilado en la primera fase del trabajo. De este modo, se cuenta a la vez con un repositorio en el que tanto los instrumentos como los resultados de la evaluación están disponibles públicamente. El instrumento para la recolección de datos por parte del evaluador se presenta en el apartado 3.3.6.1 y consiste en una serie de formularios web (uno para cada característica de calidad). La interfaz de selección, que se presenta en el apartado 3.3.6.2, consiste en un formulario que permite realizar búsquedas utilizando propiedades semánticas para recuperar los datos según los requisitos del usuario.

#### 3.3.6.1 Instrumento de evaluación

Como se explicó en el apartado anterior, la parte que se ocupa de la caracterización de los proyectos de desarrollo permite recoger datos cualitativos que nos darán una idea general de cada proyecto y sus posibilidades a mediano y largo plazo, mientras que para la evaluación de dos de las tres características del software como producto a cada posible respuesta se le asigna una puntuación que se va sumando para obtener al final un resultado total por subcaracterística. Con esta combinación se esperaba lograr una evaluación integral de las tecnologías libres que permitiera a los

usuarios finales hacerse una idea general sobre cada proyecto y tener parámetros para comparar sistemas similares.

A la hora de preparar el instrumento de evaluación se tuvo en cuenta la disponibilidad de los datos. En otras palabras, se incluyó el tipo de información que normalmente está disponible en las páginas web de los proyectos y en las forjas utilizadas por la mayoría de proyectos libres, de modo que para la recolección de los datos no fuera necesario preguntar en foros o contactar con los desarrolladores o usuarios de los programas.

Los atributos correspondientes a las cuatro características de los proyectos de desarrollo (estrategia del proyecto, comunidad, madurez y reputación) y a la funcionalidad del producto se implementaron como propiedades semánticas en la *wiki* al ser datos objetivos que no dependen del punto de vista del evaluador. En otras palabras, a cada uno de estos atributos solo puede corresponderle un valor o serie de valores en el momento de recoger los datos, aunque dichos valores se pueden actualizar según la evolución de los proyectos.

Para facilitar al máximo el trabajo del evaluador, se utilizó la extensión de MediaWiki denominada Semantic Forms. Esta extensión permite crear formularios web en los que a cada campo se le asigna una propiedad semántica y al que corresponde un tipo de entrada según el tipo de dato. En este caso se creó un formulario dividido en varias pestañas (ver figura 3.8). Los datos obtenidos se presentan luego como información complementaria en las fichas del catálogo elaborado en la primera fase del trabajo.

La evaluación de algunos atributos de la portabilidad y la usabilidad de los programas, por otro lado, se basa en datos que dependen del punto de vista del evaluador. Para la portabilidad, en el caso del atributo *compatibilidad*, por ejemplo, la respuesta dependerá del sistema operativo en que el usuario tenga previsto instalar el programa. En la misma línea, en el caso de los atributos de la usabilidad, aunque se procuró favorecer la objetividad definiendo tres posibles respuestas para cada atributo, aspectos como la comprensibilidad de la interfaz gráfica dependerán de los conocimientos previos del usuario y su grado de familiaridad con el tipo de



herramienta evaluado, mientras que el atributo *disponibilidad en el idioma requerido* dependerá de la lengua del usuario. Por esto, el instrumento para recoger los datos sobre la portabilidad y la usabilidad de las herramientas se implementó mediante una extensión adicional de MediaWiki denominada Qpoll. Esta extensión permite preparar formularios tipo encuesta que pueden ser rellenos por diferentes evaluadores (ver figura 3.9).

The screenshot shows a web-based evaluation form titled 'Project Strategy'. The form is part of a larger system with tabs for 'General Information', 'Functionality', 'Project Strategy', 'Community', 'Maturity', and 'Reputation'. The 'Project Strategy' tab is currently selected. The form contains the following questions and input fields:

- Evaluation date:** A date picker showing 'April' and '2012'.
- How did the project originate?** A dropdown menu.
- What is the underlying ethics?** A dropdown menu.
- What is the licensing strategy?** A dropdown menu.
- What is the license(s) permissiveness?** Three checkboxes: 'no copyleft', 'weak copyleft', and 'strong copyleft'.
- Are there explicit contributor agreements? (URL)** A text input field.
- What is the intellectual rights management policy?** A dropdown menu.
- Is there a product roadmap? (URL)** A text input field.
- Is there a requirements specification? (URL)** A text input field.
- Is there a description of new functions to be implemented? (URL)** A text input field.
- Is there a versioning plan?** A text input field.
- How is the project represented?** A dropdown menu.
- What is the system of governance?** A dropdown menu.
- How can the decision-making process be described?** A dropdown menu.
- Is the project a derivative or fork of another free/open-source project?** A text input field.
- Does the project integrate code from other free/open-source projects?** A text input field.
- Does the project develop other tools?** A text input field.

Figura 3.8: Instrumento de evaluación – Estrategia del proyecto.

Una vez rellenos los formularios, los resultados se pueden visualizar en una página especial de la *wiki*, a la que se enlaza desde la ficha principal de cada programa, y además se pueden exportar en formato de hoja de cálculo. Para facilitar la consulta del instrumento de evaluación, se incluyó la versión completa en español en formato de texto en el Anexo 2 (p. 261).

**OpenTM2-Evaluation**

**Portability**

**1 Modularity**  
 Check if the software design allows for the development of separate components that can be connected to the main system (e.g. existence of plugins or APIs).  1  2  3

**2 Scalability**  
 Check whether the system is suitable for large-scale implementations.  1  2  3

**3 Time For installation**  
 Check the amount of time needed to install the program (excluding system preparation).  1  2  3

**4 System requirements**  
 Check whether it is necessary to install other software to make the program work.  1  2  3

**5 Technical skills required for installation**  
 Check whether advanced technical skills are necessary in order to install the software.  1  2  3

**6 Compatibility**  
 Check whether the program can be used in the existing environment.  1  2  3

**7 Integrability**  
 Check whether the system can be integrated in your current workflow, i.e. whether it is compatible with other software you use.  1  2  3

**Legend:**  
 1 = More than 30 minutes; 2 = Between 15 and 30 minutes; 3 = Less than 15 minutes.

Figura 3.9: Instrumento de evaluación – Portabilidad.

### 3.3.6.2 Interfaz de selección

La interfaz de selección de tecnologías libres para la traducción permite la definición de una serie de filtros que recuperan la información de los programas con base en las necesidades del usuario. La simplicidad y facilidad de uso fueron factores clave a la hora de preparar la interfaz de selección ya que el objetivo era que pudiera ser utilizada por traductores en formación, profesionales, equipos de traductores, pequeñas empresas o docentes de traducción. A diferencia de los ingenieros o desarrolladores de software, este tipo de usuarios normalmente no está familiarizado con los procedimientos de evaluación de software, por lo que se trató de lograr una interfaz sencilla y clara.

Al igual que parte del instrumento de evaluación, la interfaz de selección se basa también en la extensión Semantic Forms, pero en este caso el formulario

creado en lugar de servir para introducir datos permite realizar búsquedas (*queries*) con base en las anotaciones semánticas (ver figura 3.10).

**Personal tools**

- Safg.admin
- My talk
- My preferences
- My watchlist
- My contributions
- Log out

**Navigation**

- About FOSS4Trans
- Editing and Publishing Tools
- Language Tools
- Translation Tools
- Management Tools
- Tools for Windows
- Tools for GNU/Linux
- Tools for Mac OS X
- Add/Edit Tool
- Blog [ES]

**Toolbox**

- Upload file
- Special pages

## Run query: Selector

Use the filters below to specify your needs; all fields are optional.

**Typology:**

**Operating system:**

**Architecture:**

**Implemented standards:** Unicode, TXT, TAB, CSV

**Office formats:** TXT, CSV, TAB, DOC

**DTP formats:** MIF (FrameMaker), XML (FrameMaker), INX (InDesign), IDML (InDesign)

**Multimedia formats:** PSD (Photoshop), SVG, SVG (Photoshop), SVG (Illustrator)

**Software localization formats:** RC, DLG, EXE, DLL

**Web localization formats:** HTML, XML, ASP, PHP

Figura 3.10: Captura de pantalla de la interfaz de selección de herramientas.

### **3.4 Fase 3: Evaluación de las tecnologías libres para la traducción**

La tercera fase del trabajo empírico consistió en llevar a cabo la evaluación preparada en la fase anterior. En el apartado 3.4.1 se presentan los proyectos evaluados, mientras que en el apartado 3.4.2 se describe el procedimiento de recolección de datos para la caracterización de los proyectos (3.4.2.1) y la evaluación de las herramientas (3.4.2.2). Para terminar, en el apartado 3.4.3 se presentan los resultados de la evaluación y en el 3.4.4 la discusión de los mismos.

#### ***3.4.1 Descripción de la muestra***

Para la evaluación, se identificaron en el catálogo de tecnologías libres para la traducción los proyectos que desarrollan herramientas correspondientes a la tipología *sistemas de memorias de traducción*, de la categoría *herramientas de traducción*, disponibles como aplicaciones de escritorio. Se excluyeron los sistemas basados en web por requerir conocimientos técnicos más avanzados para llevar a cabo la instalación y puesta en marcha en un servidor. La muestra incluye proyectos en diferentes fases de evolución y con diferentes tipos de vinculación institucional y empresarial. Los proyectos identificados para evaluación fueron aquellos encargados del desarrollo de las siguientes herramientas:

- **Anaphraseus:** consiste en una serie de macros para el procesador de palabras de la suite OpenOffice.org (también compatibles con LibreOffice). Está disponible como extensión del programa y como documento ODT independiente y permite utilizar el procesador de palabras como entorno de traducción, a la manera de la versión original de WordFast® o Trados. Al tratarse de una extensión para OpenOffice.org, Anaphraseus se puede utilizar en equipos con cualquier sistema operativo mientras se cuente con dicha suite ofimática y con el entorno de ejecución de Java.
- **Autshumato ITE:** un sistema de memorias de traducción que integra un procesador de textos (OpenOffice.org), un entorno de traducción con memorias y glosarios (OmegaT) y un motor de traducción automática

(Moses). La herramienta está basada en Java y por tanto se puede utilizar en equipos con sistema operativo Windows, GNU/Linux o Mac OS X, teniendo como requisito una versión actualizada del entorno de ejecución de Java.

- **EsperantiloTM:** un sistema de memorias de traducción escrito en el lenguaje Tcl y disponible para los sistemas operativos Windows y GNU/Linux.

- **ForeignDesk:** una suite de herramientas de traducción que incluye un sistema de memorias de traducción y un gestor de terminología disponibles únicamente para equipos con el sistema operativo Windows.

- **Frankenstein Translation Memory Suite:** un sistema de memorias de traducción basado en Java y por tanto disponible para cualquier sistema operativo con el entorno de ejecución de Java.

- **OmegaT:** un sistema de memorias de traducción basado en Java y por tanto multiplataforma (Windows, GNU/Linux, Mac OS).

- **OmegaT+:** un sistema de memorias de traducción escrito en Java y por tanto disponible para cualquier sistema operativo que cuente con un entorno de ejecución de Java actualizado.

- **Open Language Tools (OLT):** un conjunto de herramientas para la traducción escrito en Java y por tanto compatible con cualquier sistema operativo que cuente con el entorno de ejecución de Java. OLT se compone de una interfaz de conversión de formatos hacia y desde XLIFF y un entorno de edición específico para este tipo de archivos.

- **OpenTM2:** se trata de la versión libre del sistema de memorias de traducción TM/2, desarrollado originalmente por IBM y disponible por ahora solo para el sistema operativo Windows.

- **Transolution:** un sistema de memorias de traducción escrito en Python y disponible para equipos con sistema operativo Windows, GNU/Linux y Mac OS.

- **Virtaal:** un sistema de memorias de traducción escrito en Python y disponible para equipos con sistema operativo Windows, GNU/Linux o Mac OS X.

### **3.4.2 Recolección de datos**

En el Anexo 3 (ver página 278) se presenta el protocolo de aplicación del instrumento de evaluación que sirve como guía durante la recolección de los datos. Tanto para la caracterización de los proyectos como para la evaluación de las herramientas la recolección de datos se llevó a cabo de manera manual mediante los formularios web creados para tal propósito. La información recogida en los formularios se complementó con capturas de pantalla de las páginas de los proyectos y de la interfaz de las herramientas. En los subapartados a continuación se explica en términos generales el procedimiento de recolección de datos para la caracterización de los proyectos (3.4.2.1) y la evaluación de las herramientas (3.4.2.2).

#### **3.4.2.1 Caracterización de los proyectos de desarrollo**

Para recoger los datos sobre los proyectos de la muestra se exploraron sus páginas web y las páginas en las forjas donde se gestiona el desarrollo de los programas. En el instrumento de evaluación, los campos correspondientes a cada característica de calidad de los proyectos están disponibles en un único formulario con varias pestañas (una por característica), de modo que durante la exploración de las páginas web y las forjas de desarrollo a medida que se iba encontrando la información pertinente era posible cambiar entre pestañas para rellenar los campos correspondientes (ver figura 3.8 en la página 161).

Para determinar la estrategia de los proyectos se revisaron las páginas web en busca de información concerniente al origen del proyecto, su filosofía y su alcance. Además, se revisaron aquellas secciones dedicadas a la información sobre la licencia o licencias utilizadas para distribuir el software. Una vez identificada la licencia, se utilizó la tabla 1.1 presentada en la página 10 para clasificarla según el tipo de permisividad (con *copyleft* fuerte, con *copyleft* débil o sin *copyleft*). Finalmente, se buscó una hoja de ruta o información referente a nuevas funciones que se planea implementar.

Para obtener los datos necesarios para caracterizar la comunidad se empezó por explorar las páginas web para determinar el tipo de comunidad y las posibles

organizaciones vinculadas al proyecto. También en las páginas web se buscó información sobre los espacios de comunicación utilizados, los servicios disponibles para la comunidad y la existencia de herramientas derivadas. Dado que cada proyecto implementa diferentes herramientas de comunicación, en cada caso se rellenaron solo los campos pertinentes. Finalmente, se exploraron también los repositorios de código fuente para determinar el número de desarrolladores activos y se visitaron luego los archivos de los foros de usuarios para comprobar el número total de suscriptores, de participantes activos, el promedio de mensajes por mes durante 2011 y el tiempo promedio de respuesta para las últimas cinco preguntas formuladas.

La información necesaria para completar la parte del instrumento correspondiente a la madurez de los proyectos se obtuvo principalmente a partir de las forjas, donde se registra la fecha de inicio de cada proyecto y su estado de desarrollo. Además, se revisaron las herramientas de gestión utilizadas, la información sobre las versiones y actualizaciones lanzadas durante 2011 y la fecha de liberación de la última versión disponible. Finalmente, en las páginas web se buscó información sobre los procedimientos que se deben seguir para contribuir al proyecto.

Para determinar la reputación de los proyectos se realizaron búsquedas en la web, en directorios de software, en foros especializados para traductores y en el buscador especializado para redes sociales *Topsy*. Los datos sobre el número de descargas y sobre reseñas y puntuaciones de los usuarios se obtuvieron de las forjas donde se gestionan los proyectos. Además, se visitó la plataforma Ohloh<sup>36</sup> para revisar si los proyectos están registrados allí.

#### ***3.4.2.2 Evaluación del software como producto***

La evaluación del software como producto solo se llevó a cabo para aquellos proyectos que obtuvieron por lo menos una estrella para las características de

---

<sup>36</sup> Ohloh (<http://ohloh.net>) es una plataforma para desarrolladores y proyectos de software libre en la que se analizan los repositorios de código fuente de los programas y se ofrecen resúmenes de las estadísticas, incluyendo las líneas de código, licencias y lenguajes de programación utilizados, nivel de actividad de los proyectos y valor estimado de los mismos.

comunidad y maduras. En otras palabras, no se evaluaron aquellas herramientas que son producto de proyectos que puedan considerarse inactivos o abandonados.

Para evaluar los atributos correspondientes a la calidad de las herramientas se exploró la documentación disponible y se descargaron e instalaron los programas. En el caso de la portabilidad, los atributos correspondientes a la subcaracterística *adaptabilidad* se evaluaron con base en la documentación de cada programa. A continuación, se procedió con la instalación de los programas según sus instrucciones para evaluar el tiempo necesario para la instalación, los requisitos técnicos y los conocimientos técnicos necesarios, atributos correspondientes a la subcaracterística *instalabilidad*. Para evaluar los atributos correspondientes a la subcaracterística *coexistencia*, como entorno de instalación se utilizó un ordenador con doble procesador de 1,60GHz, 1 GB de RAM y el sistema operativo Windows XP SP2; además, se asumió que el flujo de trabajo implica la traducción de archivos DOC y DOCX principalmente y la utilización de memorias en formato TMX.

Una vez instalados los programas, para evaluar las subcaracterísticas de la usabilidad se exploraron las interfaces gráficas y la documentación incluida, así como cualquier documentación adicional disponible en las páginas web de los proyectos. Para los atributos correspondientes a la interfaz de usuario, se comprobó la distribución predeterminada de los elementos, así como los idiomas disponibles. Luego, se revisaron los iconos y nombres de funciones y menús y se buscó en las opciones de configuración la posibilidad de cambiar la apariencia y exportar los cambios. A continuación, luego de determinar los tipos de documentación disponible, se procedió a revisar los manuales para valorar su actualización y exhaustividad, así como la existencia de ayudas visuales u otro material multimedia complementario.

Finalmente, para detallar la funcionalidad de los programas se utilizó la información incluida en la página web y en la documentación de cada herramienta y se exploró la interfaz de usuario en busca de las opciones correspondientes. Ya que para esta característica la evaluación consistía únicamente en indicar la presencia o ausencia de las funciones, no se llevaron a cabo acciones de prueba; la exploración



de la interfaz se llevó a cabo solo con el fin de comprobar que las funciones descritas en la documentación estuvieran implementadas y no para comprobar posibles errores en su ejecución.

### **3.4.3 Resultados de la evaluación**

En este apartado se presentan los resultados obtenidos luego de aplicar el instrumento de evaluación a cada uno de los proyectos y herramientas seleccionados según se detalla en los apartados 3.4.2.1 y 3.4.2.2. Esta misma información se presenta en inglés y de manera más esquemática en las fichas diseñadas para presentar los proyectos en la *wiki* disponible en línea (ver figura 3.10). Como se mencionó antes, los datos se etiquetaron como propiedades semánticas y la *wiki* permite también visualizar la información en crudo como listado de propiedades semánticas con sus valores (ver Anexo 4, p. 281), así como exportarla en el formato de intercambio RDF (del inglés *Resource Description Framework*).

A continuación, en el apartado 3.4.3.1 se presentan los resultados correspondientes a la caracterización de los proyectos y en el apartado 3.4.3.2 se presentan los resultados de la evaluación de las herramientas. Tanto para los proyectos como para el software se presentan primero los resultados por subcaracterística y luego la valoración general obtenida mediante el procedimiento de consolidación de resultados presentado en el apartado 3.3.5.3.

Cabe recordar que para cuatro de las siete características (madurez, comunidad, portabilidad y usabilidad) se definieron unos criterios de aceptación y se utilizaron estrellas como indicadores de su cumplimiento. Para interpretar el número de estrellas obtenido se utilizó la siguiente tabla de correspondencias: 3 estrellas = satisfactorio, 2 estrellas = aceptable, 1 estrella = pobre, 0 estrellas = inaceptable. Para las tres características restantes (estrategia del proyecto, reputación y funcionalidad) no se utilizaron indicadores para resumir los resultados sino que se incluyen los datos completos en las fichas del catálogo. Además, como ya se explicó antes, en el caso de los proyectos con resultados inaceptables para las

características de comunidad y madurez no se llevó a cabo la evaluación de la herramienta.

EsperantiloTM	
<b>EsperantiloTM</b>	<b>Project Details</b>
Category: Translation Tools	Launched as an independent initiative, the project is led by the original developer. Code is developed on a volunteer basis. The software and all associated features are available under a single free/open-source license (strong copyleft). The copyright is owned by an individual.
Typology: Translation environment	
EsperantiloTM is a program for translators which facilitates the translation of papers by using a translation memory.	
<a href="http://www.esperantilo.org/tm/">http://www.esperantilo.org/tm/</a>	
Application Type: Desktop	<b>Community</b>
Programming language: Tcl	★ ★ ★
Operating systems: Windows, GNU/Linux	The software is developed and managed by an isolated developer.
Last release: 0.992 (2008/07/25)	⊖ No active user communication venues.
License: GNU General Public License	<b>Maturity</b>
<b>Available Resources</b>	
Documentation: <a href="http://www.esperantilo.org/tm/esperantilotm_en.pdf">http://www.esperantilo.org/tm/esperantilotm_en.pdf</a>	★ ★ ★
	The project was started on 2008/06/12 .
	⊖ No recent releases available.

Figura 3.10: Ejemplo de ficha de resultados en la wiki.

### 3.4.3.1 Caracterización de los proyectos

En este apartado presentaremos los resultados obtenidos luego de aplicar el instrumento de evaluación a los once proyectos seleccionados. En algunos casos se incluyen capturas de pantalla que tienen como fin el servir de ejemplo de algunas de las fuentes utilizadas para recoger los datos. Las fichas con los datos crudos registrados en la *wiki* se presentan en el Anexo 4.

#### 3.4.3.1.1 Anaphraseus

- **Estrategia del proyecto**

Anaphraseus surgió como un proyecto independiente y el titular de los derechos es el desarrollador original, Ole Yansen, que sigue liderando el proyecto (ver figura 3.11). El desarrollo es llevado a cabo de manera voluntaria y el software está disponible bajo una sola licencia libre (la GNU GPL, con *copyleft* fuerte). El

sistema de gobierno es una dictadura benevolente donde los objetivos del proyecto y la mayor parte de las decisiones dependen del desarrollador principal.

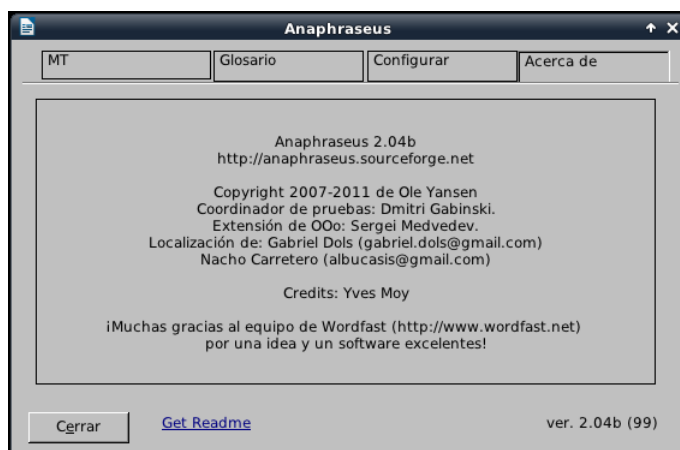


Figura 3.11: Acerca de Anaphraseus.

- **Comunidad**

El proyecto cuenta con una página web de diseño simple donde se puede encontrar fácilmente la información más relevante sobre el proyecto (ver figura 3.12). En marzo de 2012 había un solo desarrollador activo, un colaborador encargado de la documentación y un grupo de usuarios con 40 suscriptores y un promedio de dos mensajes por mes durante 2011. Para la comunicación entre miembros de la comunidad se utiliza también el chat IRC y el proyecto cuenta con perfil oficial en las redes sociales Twitter y Facebook.

Se valora positivamente que el proyecto cuente con una página web oficial y que se registre actividad de desarrollo reciente, así como la existencia de diversos espacios de comunicación para los usuarios; por el contrario, se valora negativamente el bajo promedio de mensajes en el foro de usuarios en 2011 y la ausencia de mensajes en el mismo durante el último mes.

- **Madurez**

El proyecto está registrado en SourceForge desde el 8 de octubre de 2007 y según los datos allí registrados su estado actual de desarrollo es beta. En 2011 se lanzaron 3 versiones del programa y la última versión disponible (2.04 beta) fue lanzada el

12 de diciembre de 2011. El código se gestiona en un repositorio público con control de versiones (SVN) y los informes de fallos y las solicitudes de nuevas funciones se gestionan mediante las herramientas que ofrece la forja SourceForge. En la página web se incluye un formulario de contacto para los interesados en contribuir al proyecto.

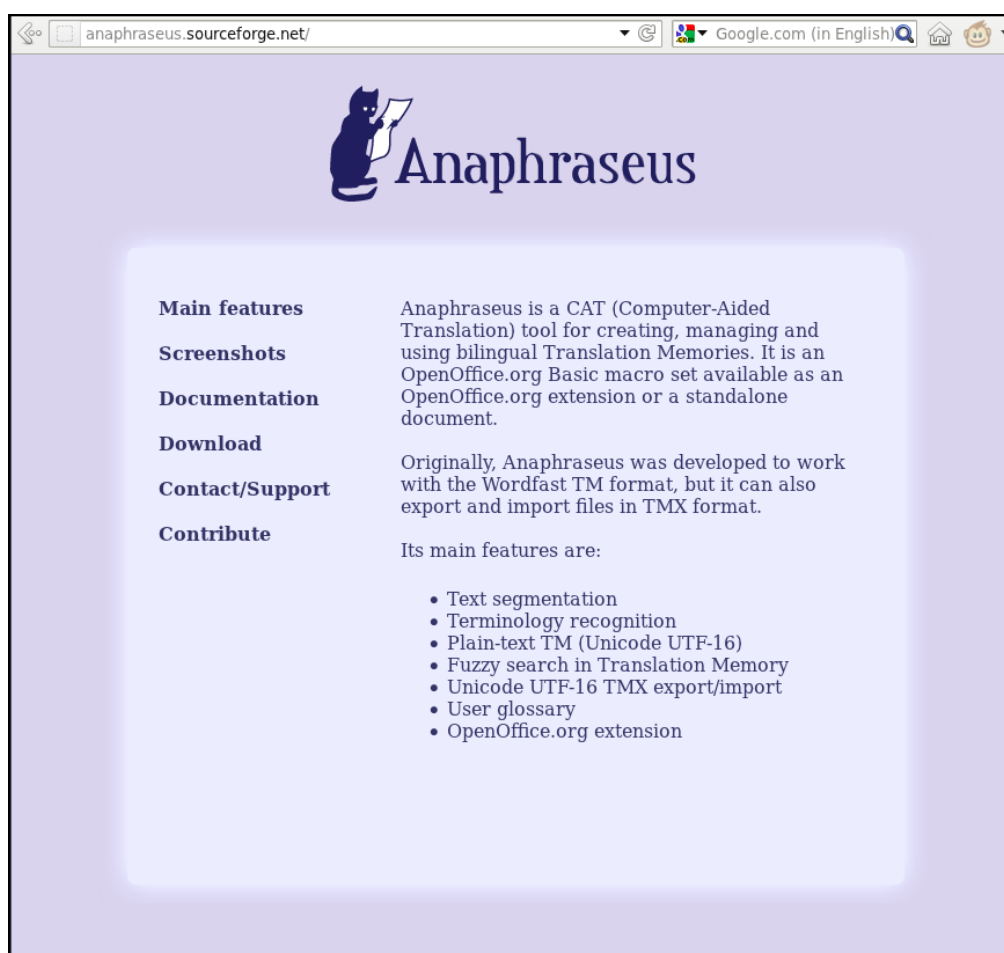


Figura 3.12: Página web del proyecto Anaphraseus.

En este caso, se valora positivamente la edad del proyecto, el lanzamiento reciente de una versión actualizada y la utilización de herramientas de gestión. En cambio, se valora negativamente el hecho de que después de casi cinco años de existencia el estado de desarrollo no sea estable o maduro.

#### • **Reputación**

En marzo de 2012 el programa fue descargado 655 veces y según el número de descargas durante la semana siguiente al lanzamiento de las tres últimas versiones

los usuarios regulares podrían ser unos 170. En los últimos años se ha escrito sobre el proyecto en diversos blogs sobre traducción y también existen discusiones sobre la herramienta en los foros de traductores ProZ y TranslatorsCafe. Además, las puntuaciones de los usuarios en SourceForge son en general positivas, con 238 recomendaciones y 2 valoraciones no favorables (ver figura 3.13). También hay comentarios recientes sobre la herramienta en Twitter (ver figura 3.14) y el proyecto cuenta con un perfil en la plataforma Ohloh.

Se valora positivamente la existencia de varias publicaciones y discusiones sobre el proyecto en blogs y foros para traductores, al igual que la existencia de recomendaciones en la forja, comentarios en Twitter y un perfil en Ohloh.



Figura 3.13: Puntuación de Anaphraseus en SourceForge según sus usuarios.

• **Valoración general del proyecto**

La tabla 3.16 muestra la valoración general de Anaphraseus una vez consolidados los resultados. Según estos datos, la comunidad y la madurez del proyecto se consideran aceptables (dos estrellas en cada caso), de modo que se llevó a cabo la evaluación de la herramienta (ver apartado 3.4.3.2.1).

Comunidad		
★ Capacidad de mantenimiento	★ Sostenibilidad	★ Recursos y servicios disponibles



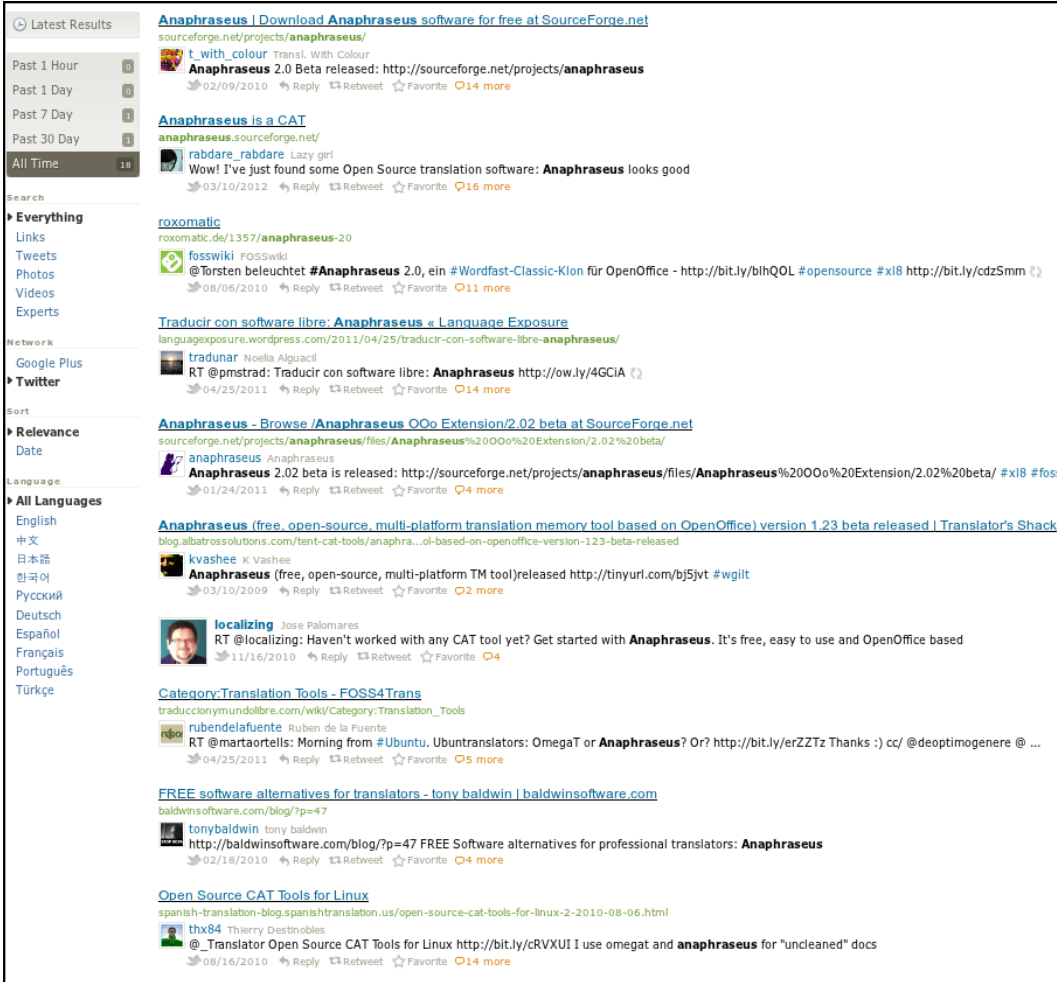
Madurez		
		
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.16: Valoración general del proyecto Anaphraseus.



The screenshot shows a Twitter search interface for the term 'Anaphraseus'. The search results are filtered by 'Latest Results'. The left sidebar includes filters for time (Past 1 Hour, Past 1 Day, Past 7 Day, Past 30 Day, All Time), search, and language. The main content area displays several tweets and links related to the project, including announcements of beta releases, translations, and user feedback. Key tweets include:
 

- A tweet from 't\_with\_colour' announcing 'Anaphraseus 2.0 Beta released' with a link to the SourceForge project page.
- A tweet from 'rabdare\_rabdare' stating 'Wow! I've just found some Open Source translation software: Anaphraseus looks good'.
- A tweet from 'roxomatic' linking to a SourceForge page for 'Anaphraseus-20'.
- A tweet from 'fosswiki' mentioning 'Anaphraseus 2.0' in the context of OpenOffice.
- A tweet from 'tradunar' retweeting information about translating with free software.
- A tweet from 'anaphraseus' announcing 'Anaphraseus 2.02 beta is released'.
- A tweet from 'kvashee' announcing 'Anaphraseus (free, open-source, multi-platform TM tool) released'.
- A tweet from 'localizing' retweeting a question about CAT tools.
- A tweet from 'rubendela Fuente' mentioning 'Anaphraseus' in the context of Ubuntu.
- A tweet from 'tonybaldwin' discussing 'FREE Software alternatives for professional translators: Anaphraseus'.
- A tweet from 'thx84' mentioning 'Anaphraseus' as a tool for 'uncleaned' docs.

Figura 3.14: Comentarios sobre Anaphraseus en Twitter.

### 3.4.3.1.2 Autshumato ITE

- *Estrategia del proyecto*

El proyecto surgió como una iniciativa financiada con fondos públicos. En concreto, se trata de un proyecto promovido por el Departamento Sudafricano de Arte y Cultura y desarrollado en el Centre for Text Technology (CTexT) de la North-West University con la colaboración de la Universidad de Pretoria. La ética

que rige el proyecto es la ética tradicional basada en la retribución económica al trabajo. El software está disponible bajo una sola licencia libre (la GNU-GPL, con *copyleft* fuerte) y el titular de los derechos es una entidad legal (CtexT) (ver figura 3.15).

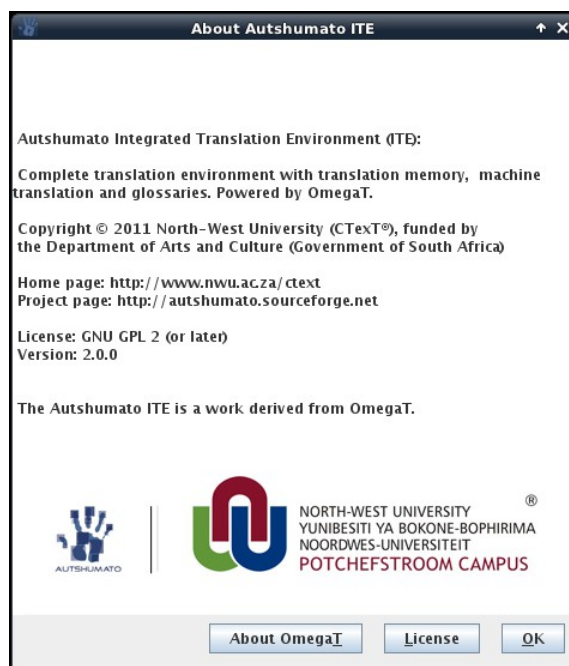


Figura 3.15: Acerca de Autshumato ITE.

El proyecto se lleva a cabo con base en una serie de objetivos predefinidos y se puede describir como una dictadura benevolente con un proceso de toma de decisiones centralizado. El proyecto integra código de otros proyectos libres (OpenOffice.org, OmegaT y Moses) y desarrolla otras herramientas libres, entre las que destacan un sistema de gestión terminológica (Autshumato TMS), una utilidad para integración de memorias de traducción (TMX Integrator), un programa de alineación de textos (CTeXt Alignment Interface Pro) y una herramienta para extraer texto de archivos PDF (PDF Text Extractor).

En este caso, se valora positivamente que el proyecto reutilice el código de otros proyectos libres y que trabajen en el desarrollo de varias herramientas, así como el hecho de que se documente el origen del proyecto.

- **Comunidad**

El proyecto cuenta con una página web en la que se incluye una breve descripción de cada herramienta y un enlace a la página en SourceForge para acceder a las descargas disponibles (ver figura 3.16). La comunidad es liderada por un centro de investigación público. En marzo de 2012 había un desarrollador activo y el foro de usuarios no registraba actividad reciente. No se encontró evidencia de oferta de servicios profesionales basados en el software.

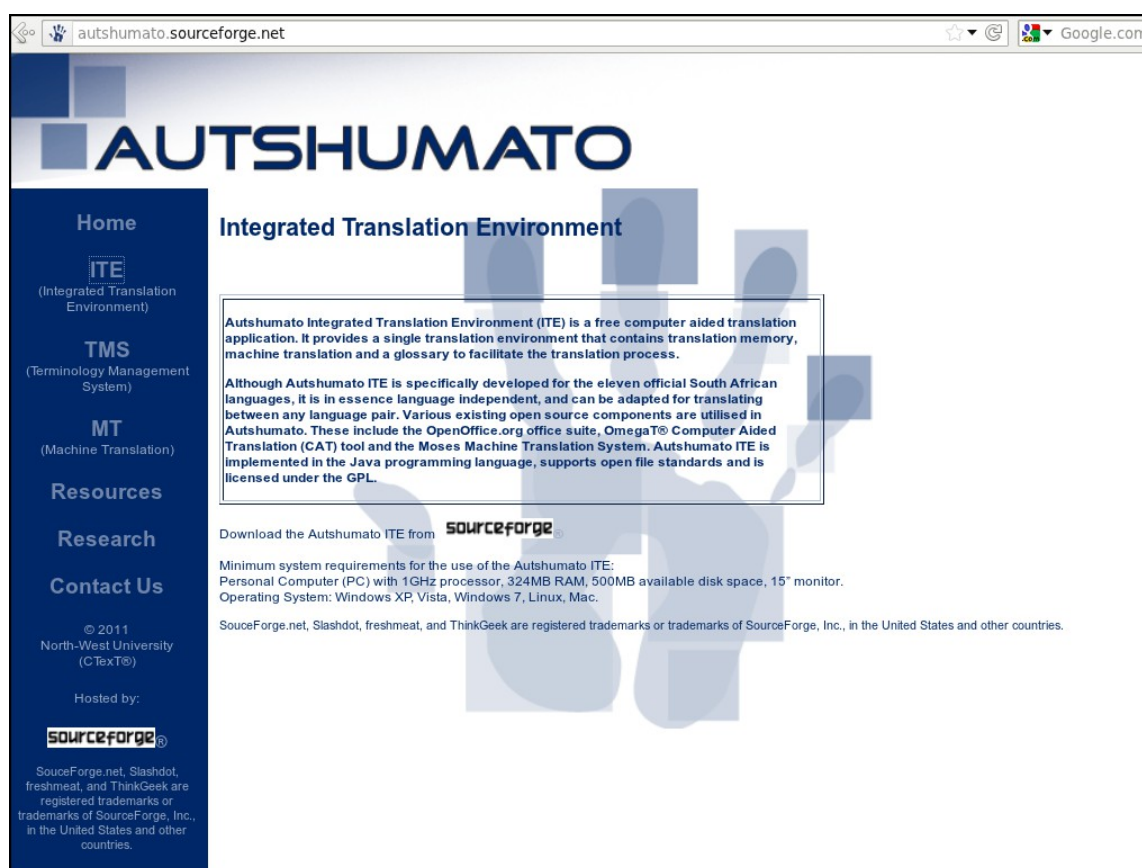


Figura 3.16: Página web del proyecto Autshumato.

En este caso se valora positivamente la actividad reciente de desarrollo, así como la existencia de un portal web con información sobre el proyecto y las herramientas que desarrollan. Por el contrario, se valora negativamente que solo exista un espacio de comunicación para los usuarios y que este no se use activamente.



- ***Madurez***

El proyecto fue registrado en SourceForge el 4 de febrero de 2009 y su estado de desarrollo actual es beta. En 2011 solo se lanzó una versión del programa y no se lanzaron actualizaciones. La última versión estable disponible (2.0.0) fue lanzada el 20 de marzo de 2012. El código se gestiona en un repositorio público con control de revisiones (SVN) y el proyecto utiliza el sistema de gestión de fallos proporcionado por SourceForge.

Para esta característica se valora positivamente que el proyecto tenga más de dos años, que se utilice una forja pública para la gestión del código y que se utilice un sistema de gestión de informes de fallos. Por otro lado, se valora negativamente que el estado de desarrollo aún no sea estable o maduro.

- ***Reputación***

En marzo de 2012 el programa fue descargado 78 veces y según el número de descargas en la semana siguiente al lanzamiento de las 3 últimas versiones el número de usuarios regulares se puede estimar en unos 36. El proyecto fue presentado en la decimotercera edición de la Conferencia Anual de la Asociación Europea para la Traducción Automática (EAMT, por sus siglas en inglés) (Groenewald & Fourier, 2009). Las puntuaciones de los usuarios en SourceForge son positivas en su gran mayoría (13 recomendaciones frente a 1 valoración negativa, ver figura 3.17). El proyecto aparece listado en el directorio de software Softpedia y en la plataforma Ohloh. Además, se encontraron comentarios sobre el proyecto en Twitter.

Se valora positivamente la presentación del proyecto en un evento académico del área de la traducción, así como la existencia de puntuaciones y comentarios de los usuarios, la inclusión de la herramienta en un directorio de software y que el proyecto cuente con un perfil en la plataforma Ohloh.

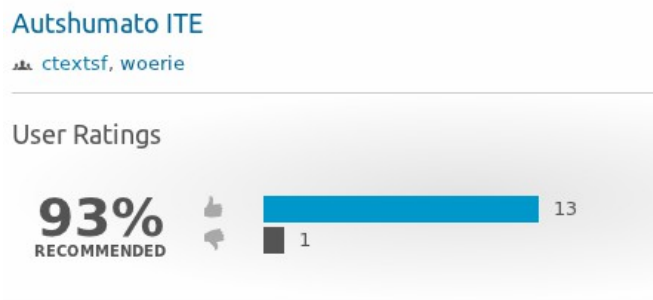


Figura 3.17: Puntuación de Autshumato ITE en SourceForge según sus usuarios.

- **Valoración general del proyecto**

La tabla 3.17 presenta la valoración general del proyecto Autshumato. Según estos resultados, el proyecto no cuenta con una comunidad de usuarios y desarrolladores que garantice la capacidad de mantenimiento y sostenibilidad del proyecto y no existe oferta de recursos y servicios para los usuarios (cero estrellas). No obstante, la madurez del proyecto es aceptable (dos estrellas), de modo que se llevó a cabo la evaluación de la herramienta (ver apartado 3.4.3.2.2).

Comunidad		
★	★	★
Capacidad de mantenimiento	Sostenibilidad	Recursos y servicios disponibles

Madurez		
★	★	★
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.17: Valoración general del proyecto Autshumato.

### 3.4.3.1.3 EsperantiloTM

- **Estrategia del proyecto**

Este proyecto surgió como una iniciativa liderada por un desarrollador independiente que figura como titular de los derechos. El desarrollo se llevaba a cabo de manera voluntaria y el software está disponible bajo una sola licencia libre con *copyleft* fuerte (GNU GPL). El proyecto reutiliza el código de otro proyecto

libre (Hunspell) y se deriva de otro proyecto libre (Esperantilo text editor), aspectos que se valoran positivamente.

- ***Comunidad***

El proyecto cuenta con una página web donde se recoge la información básica del proyecto. Un solo desarrollador se encargaba de escribir el código y no hay un foro u otro espacio de comunicación para los usuarios ni oferta de servicios basados en el software.

En este caso, se valora negativamente que no haya actividad de desarrollo reciente, así como la inexistencia de espacios de comunicación y servicios para los usuarios. Se valora positivamente la existencia de una página web con información sobre el proyecto.

- ***Madurez***

La primera versión del proyecto fue lanzada el 12 de junio de 2008. El proyecto no está registrado en una forja pública ni utiliza herramientas de gestión de código o informes de fallos. Tampoco existen procesos documentados para contribuir al proyecto. La última versión disponible (0.992) fue lanzada el 25 de julio de 2008 y el estado de desarrollo es inactivo.

En este caso se valora negativamente que no se utilicen herramientas específicas para la gestión del proyecto, que la última versión disponible tenga ya cuatro años y que el proyecto esté inactivo.

- ***Reputación***

Por no estar registrado en ninguna forja de desarrollo no se pudieron obtener datos sobre el número de descargas. Tampoco se encontraron publicaciones sobre el programa ni discusiones en foros de traductores o comentarios en las redes sociales. El proyecto no está registrado en Ohloh y tampoco se encontró evidencia de que esté incluido en algún directorio de software.

Se valora negativamente la ausencia de registros públicos sobre el número de descargas, así como la aparente inexistencia de publicaciones, discusiones y comentarios sobre el proyecto.

- **Valoración general**

La tabla 3.18 presenta la valoración general del proyecto según los datos recogidos. Como se aprecia en la tabla, tanto la comunidad como la madurez del proyecto son inaceptables (cero estrellas), de modo que no se llevó a cabo la evaluación de la herramienta.

Comunidad		
 Capacidad de mantenimiento	 Sostenibilidad	 Recursos y servicios disponibles

Madurez		
 Estado del proyecto	 Gestión del proyecto	 Gestión de versiones

Tabla 3.18: Valoración general del proyecto EsperantiloTM.

### 3.4.3.1.4 ForeignDesk

- **Estrategia del proyecto**

Este proyecto surgió como una iniciativa privada y fue desarrollado originalmente como software privativo por la empresa estadounidense LionBridge, que conserva la titularidad de los derechos. El software está disponible bajo dos licencias libres: la BSD (sin copyleft) y la IBM Public License (con *copyleft* fuerte).

- **Comunidad**

El proyecto no cuenta con una página web oficial y la única información disponible se encuentra en la página de la forja de desarrollo. Según la información encontrada, después de liberar el código la empresa LionBridge no continuó

colaborando con el proyecto y a la fecha no hay desarrolladores activos. Aunque existe un foro de usuarios, los últimos mensajes allí registrados son de diciembre de 2006. Tampoco se encontró oferta de servicios basados en el software.

Se valora negativamente que el proyecto no cuente con una página web y que no haya desarrolladores activos. Igualmente, se valora negativamente que solo exista un espacio de comunicación para los usuarios y que este tampoco registre actividad reciente, así como la ausencia de servicios basados en el software.

- ***Madurez***

El proyecto fue registrado en la forja pública SourceForge el 25 de septiembre de 2001 y la última versión disponible (5.7.3) fue lanzada el 2 de mayo de 2002. El proyecto utiliza un sistema de control de versiones para gestionar el código (CSV) y la última contribución data de diciembre de 2003. El estado de desarrollo es inactivo.

En este caso se valora positivamente la edad del proyecto, que esté registrado en una forja de desarrollo y que se utilice un sistema de control de versiones. No obstante, se valora negativamente que el estado del proyecto sea inactivo y que no existan versiones recientes del programa.

- ***Reputación***

En marzo de 2012 el software fue descargado 110 veces. Como se aprecia en la figura 3.18, en SourceForge se han registrado 5 recomendaciones y 2 valoraciones negativas. El proyecto está incluido en un directorio de software y está registrado en Ohloh, pero nadie gestiona el perfil (ver figura 3.19), lo que es señal de que el proyecto no cuenta con colaboradores activos. Se encontró una discusión sobre el proyecto en un foro para traductores y no se encontraron comentarios en Twitter.

En este caso se valora positivamente la existencia de recomendaciones, así como la inclusión en un directorio de software y en la plataforma Ohloh. Por el

contrario, se valora negativamente la ausencia de comentarios sobre el proyecto en Twitter y que solo se haya encontrado una discusión en un foro para traductores.

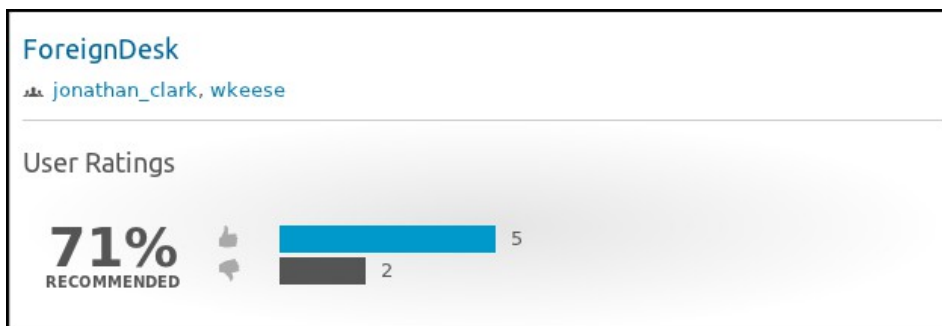


Figura 3.18: Puntuación de ForeignDesk en SourceForge según sus usuarios.

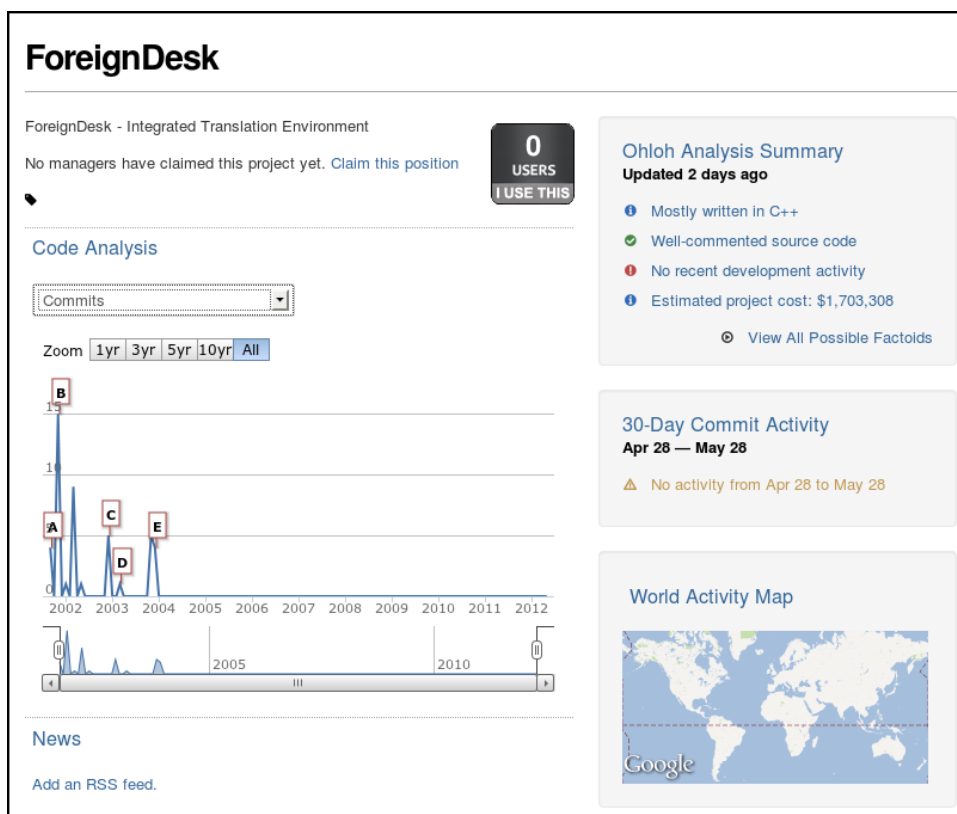


Figura 3.19: Perfil de ForeignDesk en Ohloh.net.

#### • **Valoración general del proyecto**

La tabla 3.19 presenta la valoración general de ForeignDesk. Según los datos encontrados, la comunidad y madurez del proyecto son inaceptables (cero estrellas), de modo que no se llevó a cabo la evaluación de la herramienta.

Comunidad		
		
Capacidad de mantenimiento	Sostenibilidad	Recursos y servicios disponibles

Madurez		
		
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.19: Valoración general del proyecto ForeignDesk.

### 3.4.3.1.5 Frankenstein Translation Memory Suite

- **Estrategia del proyecto**

Este proyecto surgió como una iniciativa del desarrollador independiente Hans Sussenburger, que es el único titular de los derechos. El desarrollo se llevaba a cabo de manera voluntaria. Según la página del proyecto en SourceForge, el software y sus funciones asociadas están disponibles bajo la licencia GNU GPL (con *copyleft* fuerte). No obstante, según la información disponible en la documentación, al parecer no todo el código es libre; de hecho, según una fuente el programa sería *freeware* y según otra sería *shareware* con un coste de 100 €. Esta falta de claridad sobre la estrategia de licenciamiento se valora negativamente.

- **Comunidad**

La página web oficial del proyecto ya no está disponible y la información existente se limita a la incluida en la forja de desarrollo. Ningún desarrollador participa actualmente en el proyecto y tampoco hay un foro o espacio de discusión para los usuarios. Tampoco se encontró evidencia de oferta de servicios basados en el software.

En este caso se valora negativamente que el proyecto no tenga una página web y que no haya desarrolladores activos. De igual manera, se valora negativamente la ausencia de espacios de comunicación y servicios para los usuarios.

- ***Madurez***

El proyecto fue registrado en SourceForge el 30 de octubre de 2003. No se utilizaba un sistema de control de versiones ni de gestión de informes de fallos. La única versión disponible en la forja (1.6 build 15) data del 31 de octubre de 2003 y desde entonces no se ha lanzado ninguna actualización. El estado actual de desarrollo es inactivo.

Se valoran positivamente la edad del proyecto y que esté registrado en una forja pública. Por el contrario, se valora negativamente que no se utilizaran herramientas específicas para la gestión del proyecto, la ausencia de versiones actualizadas y que el estado de desarrollo sea inactivo.

- ***Reputación***


En marzo de 2012 se registraron 8 descargas de los archivos del proyecto. Existen dos discusiones sobre el proyecto en ProZ, una de abril del año 2003 y otra de octubre del año 2005. También se encontró el proyecto en un directorio de herramientas para traductores. No se encontraron artículos ni entradas en blogs sobre el proyecto. Tampoco se encontraron reseñas en la forja ni comentarios en Twitter y el proyecto no está registrado en la plataforma Ohloh.

En este caso se valora positivamente la existencia de discusiones sobre el proyecto en un foro para traductores y en un directorio de herramientas, mientras que la ausencia de publicaciones, reseñas y comentarios se valora negativamente. De igual manera, se valora negativamente que el proyecto no esté registrado en Ohloh.

- ***Valoración general***

La tabla 3.20 presenta la valoración general del proyecto Frankenstein Translation Memory Suite. Ya que según los datos encontrados tanto la comunidad como la madurez del proyecto son inaceptables (cero estrellas) no se llevó a cabo la evaluación de la herramienta.



Comunidad		
		
Capacidad de mantenimiento	Sostenibilidad	Recursos y servicios disponibles

Madurez		
		
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.20: Valoración general del proyecto Frankenstein Translation Memory Suite.

### 3.4.3.1.6 OmegaT

- **Estrategia del proyecto**

El proyecto surgió como una iniciativa independiente del desarrollador Keith Godfrey y ahora cuenta con un grupo de líderes reconocidos. El trabajo se lleva a cabo de manera voluntaria. El software y sus funciones están disponibles bajo la licencia GNU GPL (*copyleft* fuerte) y la titularidad es distribuida entre sus desarrolladores (ver figura 3.20). Según la filosofía del proyecto (ver figura 3.21), se trata de una “anarquía delegada” donde cualquiera es libre de contribuir al proyecto y existe un equipo central de desarrolladores que decide qué contribuciones se integran al código que se distribuye a la comunidad. El proyecto integra código desarrollado por otros proyectos libres (Hunspell, LanguageTool, Lucene Tokenizers, Okapi filters).

- **Comunidad**

El proyecto cuenta con una página web donde se recoge la información pertinente (ver figura 3.22). El desarrollo es llevado a cabo por un grupo de desarrolladores de manera colaborativa e informal. En marzo de 2012 había cinco desarrolladores activos y el grupo de usuarios contaba con 1720 suscriptores, de los cuales 39 habían participado activamente el último mes. Además, el proyecto cuenta con un coordinador general, un coordinador del desarrollo, un coordinador de la documentación y un coordinador de la localización.

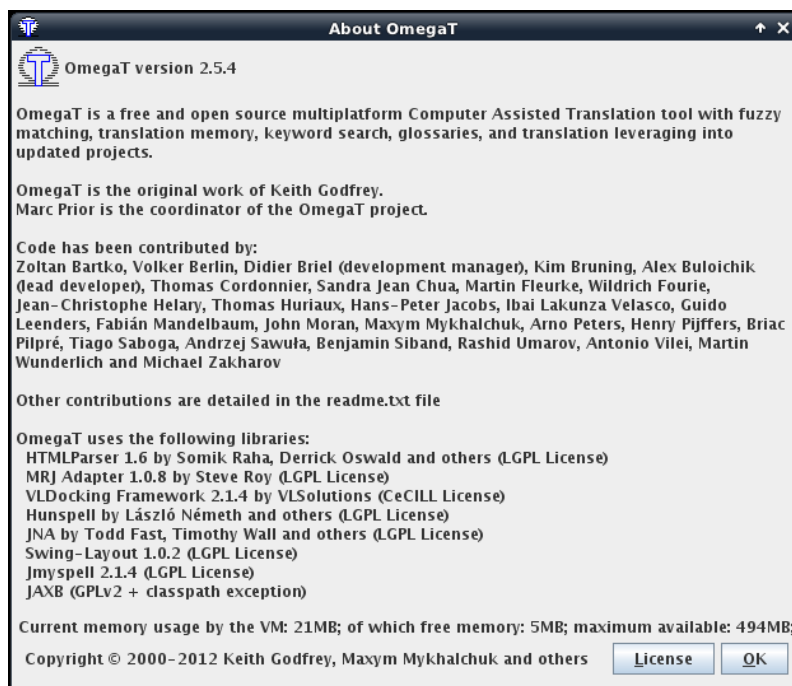


Figura 3.20: Acerca de OmegaT.

En 2011, el promedio de mensajes por mes en el grupo de usuarios fue 304 y el tiempo promedio de respuesta para las últimas cinco preguntas fue 0,3 horas; no es necesario ser miembro del grupo para consultar el archivo de mensajes. El proyecto cuenta también con una lista de distribución para desarrolladores y otra para la gestión de la localización. Además, se dispone de un canal de chat IRC. En cuanto a oferta de servicios, es posible patrocinar el desarrollo de nuevas funciones poniéndose en contacto directamente con los desarrolladores para acordar el valor de la contribución monetaria necesaria.

Existen varios proyectos derivados de OmegaT: **OmegaT+**, un *fork* o proyecto bifurcado por uno de los desarrolladores por desacuerdos (a la fecha siguen existiendo conflictos entre los dos proyectos por el nombre *OmegaT* como marca registrada por el proyecto original); **Boltran**, una versión de OmegaT basada en web; y **Autshumato ITE**, un sistema de memorias de traducción que integra OmegaT, OpenOffice.org y el motor de traducción automática Moses (en este caso existe cierto grado de colaboración entre los proyectos).

**The OmegaT project philosophy**

The same spirit can be found within the OmegaT project. Users are free to download and use the program. In turn, they are free, and indeed encouraged (but not obliged in any way) to contribute back to the project. How they do so is very much up to them. For example, there is nothing to stop a user writing their own user manual or a demonstration video for OmegaT, and if the project considers it remotely useful, it is likely to be made available to OmegaT's community of users. Where routine tasks are concerned, volunteers working on the OmegaT project are generally given as much freedom as possible in determining the form of their contribution. This hands-off approach is described by the current OmegaT project co-ordinator as "delegated anarchy".

In some areas, some degree of co-ordination is required. For example, anyone is free to make changes to OmegaT's program code and to submit these changes to the development team. Only if they are considered beneficial for OmegaT's users, however, will the development team include them in OmegaT. This ensures that the users' interests are protected, and has also resulted in OmegaT being awarded the Softpedia "100% clean" guarantee.

Figura 3.21: OmegaT – filosofía del proyecto.

**OmegaT**  
The free (GPL) translation memory tool

**Introducing OmegaT**

OmegaT is a free translation memory application written in Java. It is a tool intended for professional translators. It does not translate for you! (Software that does this is called "machine translation", and you will have to look elsewhere for it.) OmegaT has the following features:

- Fuzzy matching
- Match propagation
- Simultaneous processing of multiple-file projects
- Simultaneous use of multiple translation memories
- User glossaries with recognition of inflected forms
- Document file formats include:
  - XHTML and HTML
  - Microsoft Office Open XML (Office 2007/2010: .docx, .xlsx, .pptx)
  - OpenOffice.org
  - XLIFF (Okapi)
  - MediaWiki (Wikipedia)
  - Plain text
- Unicode (UTF-8) support: can be used with non-Latin alphabets
- Support for right-to-left languages

Figura 3.22: Página web del proyecto OmegaT.

En este caso se valora positivamente la existencia de una página web bien organizada y con información detallada sobre el proyecto, así como el número de colaboradores activos y la existencia de proyectos derivados. Además, se considera también positiva la existencia de varios espacios de comunicación para los miembros del proyecto, así como el nivel de actividad y tiempo de respuesta en el foro de usuarios. En cuanto a la oferta de servicios profesionales, aunque se valora

positivamente la posibilidad de patrocinar el desarrollo de nuevas funciones, teniendo en cuenta las características del proyecto se echa en falta una mayor oferta de servicios profesionales.

- **Madurez**

El proyecto se inició en el año 2000 según la declaración de *copyright* (ver figura 3.20) y fue registrado en SourceForge el 28 de noviembre de 2002. El estado de desarrollo actual es estable y se mantienen dos versiones principales paralelas: una denominada *estándar*, con todas las funciones debidamente documentadas, y otra denominada “*más reciente*” (en inglés *latest*), antes denominada *beta*, que según los desarrolladores es igualmente estable pero se diferencia de la anterior en que las funciones más recientes no están documentadas aún y la localización puede no estar totalmente actualizada. En 2011, se lanzaron 2 versiones principales y 13 actualizaciones (ver figura 3.23) y a la fecha (mayo de 2012) la versión estándar más reciente (2.5.4) es del 9 de mayo de 2012.



Subject	Author	Post Date
OmegaT standard version 2.3.0 released	didierbr	2011-07-09
OmegaT latest version 2.5.0 released	didierbr	2011-09-27
OmegaT beta version 2.2.3 update 2 released	didierbr	2011-01-22
OmegaT version 2.3.0 update 3 released	didierbr	2011-10-21
OmegaT latest version 2.5.0 update 1 released	didierbr	2011-10-25
OmegaT latest version 2.5.0 update 1 released	didierbr	2011-10-25
OmegaT beta version 2.2.3 update 1 released	didierbr	2011-01-12
OmegaT beta version 2.2.3 update 3 released	didierbr	2011-03-03
OmegaT beta version 2.2.3 update 4 released	didierbr	2011-03-19
OmegaT version 2.3.0 update 1 released	didierbr	2011-07-29
OmegaT version 2.3.0 update 2 released	didierbr	2011-09-27
OmegaT latest version 2.5.0 update 2 released	didierbr	2011-11-02
OmegaT latest version 2.5.0 update 3 released	didierbr	2011-11-16
OmegaT latest version 2.5.0 update 4 released	didierbr	2011-11-17
OmegaT version 2.3.0 update 4 released	didierbr	2011-12-27

Figura 3.23: OmegaT – versiones lanzadas en 2011.

El proyecto utiliza un repositorio con control de revisiones (SVN) para la gestión del código y las herramientas proporcionadas por SourceForge para la

gestión de fallos y solicitudes de nuevas funciones. Existe un proceso documentado para contribuir con la localización de la interfaz y la documentación del programa.

Se valora positivamente la edad del proyecto y su estado de desarrollo actual, así como la utilización de una forja pública y de herramientas específicas para la gestión del código, informes de fallos y solicitudes de nuevas funciones. Además, aunque no hay un ciclo de lanzamientos predefinido, se valora positivamente el lanzamiento regular de actualizaciones y la disponibilidad de una versión reciente.

- **Reputación**

En marzo de 2012 el software fue descargado 5033 veces. El promedio de descargas realizadas durante la semana siguiente al lanzamiento de las tres últimas versiones es 1344, valor que puede servir como indicador del número de usuarios regulares de la herramienta. Se encontraron numerosas publicaciones sobre OmegaT y también se encontraron discusiones específicas en foros de traductores. Además, OmegaT está incluido en los repositorios de varias distribuciones de GNU/Linux y aparece listado en varios directorios de software. Según las puntuaciones en SourceForge, el 88 % de los usuarios recomienda la herramienta (170 recomendaciones frente a 23 valoraciones negativas, ver figura 3.24). También se encontraron comentarios recientes en Twitter y el proyecto cuenta con un perfil actualizado en Ohloh.

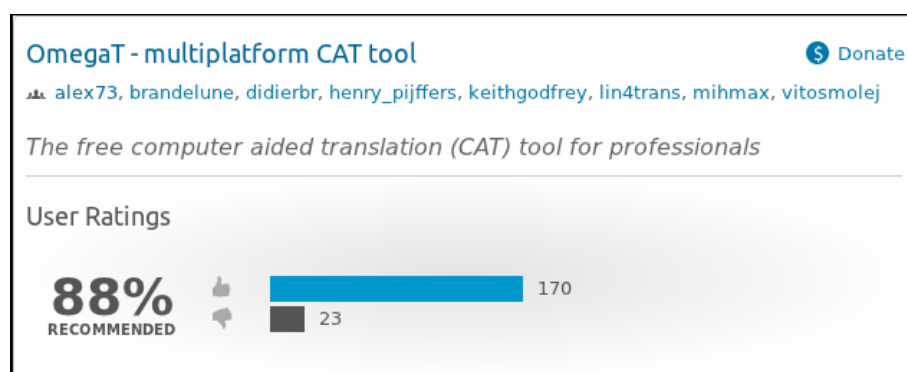


Figura 3.24: Puntuaciones de OmegaT en SourceForge según sus usuarios.

Se valora positivamente la existencia de publicaciones sobre el proyecto y el alto número de descargas. Además, se considera positiva la existencia de discusiones sobre la herramienta en foros de traductores y su inclusión en directorios de software y distribuciones de GNU/Linux. De igual manera, se valora positivamente la existencia de recomendaciones en la forja y comentarios en Twitter, así como el perfil actualizado en Ohloh.

- **Valoración general del proyecto**

La tabla 3.21 presenta la valoración general del proyecto OmegaT. Como se aprecia en la tabla, luego de consolidar los resultados se considera que tanto la comunidad y madurez del proyecto son satisfactorias (tres estrellas en cada caso), de modo que se llevó a cabo la evaluación de la herramienta (ver apartado 3.4.3.2.3).

Comunidad		
★	★	★
Capacidad de mantenimiento	Sostenibilidad	Recursos y servicios disponibles

Madurez		
★	★	★
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.21: Valoración general del proyecto OmegaT.

### 3.4.3.1.7 OmegaT+

- **Estrategia del proyecto**

El proyecto surgió a raíz de un conflicto en la comunidad de OmegaT. En el año 2005, uno de los colaboradores decidió bifurcar el proyecto y seguir desarrollando la herramienta de manera independiente. El software y sus funciones asociadas están disponibles bajo la licencia GNU GPL (*copyleft* fuerte). La titularidad de los derechos es distribuida (ver figura 3.25). El proyecto se rige por la ética *hacker* y se puede describir como una dictadura benevolente donde las decisiones las toma el desarrollador principal.

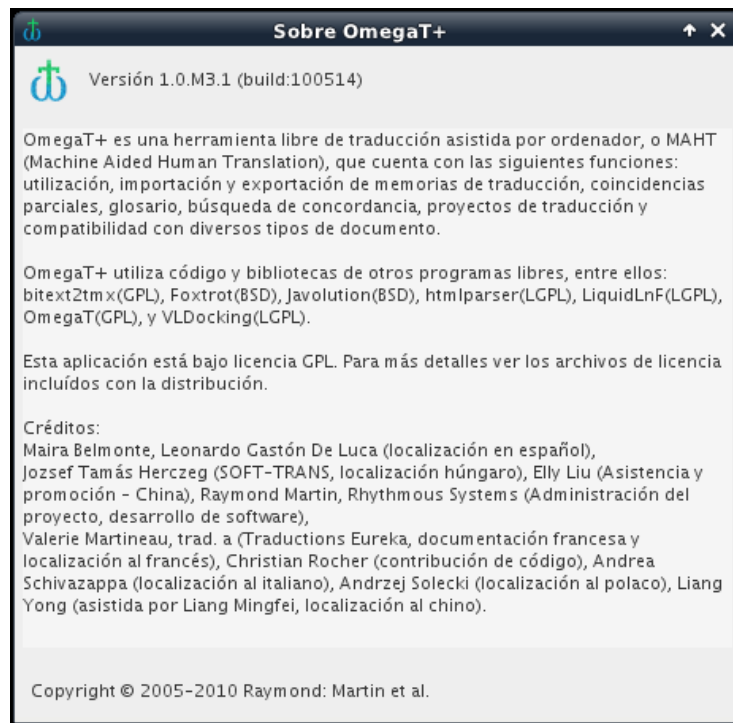


Figura 3.25: Acerca de OmegaT+.

- **Comunidad**

El proyecto cuenta con una página web donde se recoge la información relevante. La comunidad es gestionada por un desarrollador independiente. No se encontró evidencia de actividad de desarrollo reciente, pero sí existe un foro de usuarios con mensajes recientes (2011 y 2012) y el proyecto cuenta con un perfil en Facebook. El desarrollador ofrece servicios profesionales de pago (desarrollo personalizado y suscripciones de valor añadido).

En este caso se valora positivamente la existencia de una página web oficial y de dos espacios de comunicación para los usuarios. Igualmente, se considera positiva la existencia de una oferta de servicios profesionales basados en el software. Por el contrario, se valora negativamente que no haya desarrolladores activos y el bajo nivel de actividad reciente en el foro de usuarios.

- ***Madurez***

El proyecto fue registrado en SourceForge el 11 de junio de 2005 y su estado de desarrollo actual es estable. La versión más reciente (1.0M3.1) fue lanzada el 14 de mayo de 2010. El proyecto no utiliza un repositorio con control de revisiones para el código fuente pero sí un sistema de gestión de fallos y solicitudes de nuevas funciones.

Se valora positivamente la edad del proyecto y su estado de desarrollo actual, así como el hecho de que esté registrado en una forja pública y utilice herramientas específicas para la gestión de los informes de fallos y solicitudes de nuevas funciones. Por el contrario, se valora negativamente que no se utilice un repositorio con control de versiones para gestionar el código del proyecto y que la versión más reciente del programa tenga ya dos años.

- ***Reputación***

En marzo de 2012 el programa fue descargado 233 veces. Las puntuaciones de los usuarios en SourceForge son positivas en su mayoría (40 recomendaciones frente a 2 valoraciones negativas, ver figura 3.26). El proyecto aparece listado en directorios de software y está registrado en Ohloh, aunque no se presentan las estadísticas del proyecto ya que el código no se gestiona en un repositorio público. No se encontraron comentarios en Twitter ni publicaciones sobre el proyecto y se encontró una discusión en un foro para traductores.

En este caso se valora positivamente el número de descargas, así como la existencia de recomendaciones en la forja y la inclusión del proyecto en directorios de software. Por otro lado, se valora negativamente el perfil incompleto en Ohloh debido a la imposibilidad de acceder al código, así como la inexistencia de publicaciones y comentarios recientes en Twitter y el haber encontrado solo una discusión en un foro para traductores. No obstante, es necesario mencionar que las búsquedas en Internet se vieron complicadas por el hecho de que el nombre del proyecto se diferencia del nombre de otro proyecto libre solo por el carácter “+”,



que en los buscadores se utiliza como carácter especial, de modo que es posible que no se hayan recuperado todos los resultados pertinentes.

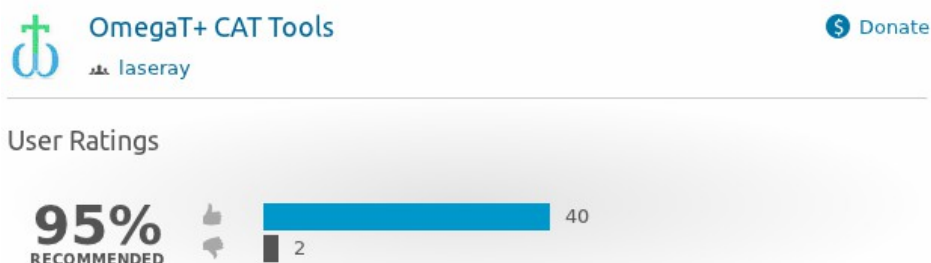


Figura 3.26: Puntuaciones de OmegaT+ en SourceForge según sus usuarios.

- **Valoración general del proyecto**

La tabla 3.22 presenta la valoración general del proyecto OmegaT+. Luego de consolidar los resultados, se considera que la comunidad y madurez del proyecto son pobres (una estrella). También en este caso se llevó a cabo la evaluación de la herramienta (ver apartado 3.4.3.2.4).

Comunidad		
★ Capacidad de mantenimiento	★ Sostenibilidad	★ Recursos y servicios disponibles
Madurez		
★ Estado del proyecto	★ Gestión del proyecto	★ Gestión de versiones

Tabla 3.22: Valoración general del proyecto OmegaT+.

### 3.4.3.1.8 Open Language Tools

- **Estrategia del proyecto**

Creado originalmente por Sun Microsystems para la localización de su documentación y software, desde que el código de Open Language Tools fue liberado el proyecto cuenta con la participación de voluntarios. Con la reciente adquisición de Sun Microsystems por parte de Oracle su futuro parece depender

principalmente de los miembros de la comunidad de OpenOffice.org (que también era de Sun Microsystems) que siguen contribuyendo al proyecto; los objetivos se establecen de manera informal y anárquica y las decisiones se toman de manera descentralizada. La titularidad de los derechos sigue apareciendo a nombre de Sun Microsystems, pero hay que tener en cuenta que esta información parece no haber sido actualizada desde el año 2005 (ver figura 3.27). El software y todas sus funciones están disponibles bajo la licencia CDDL (Common Development and Distribution License), con *copyleft* débil. Según la filosofía del proyecto (ver figura 3.28), los objetivos son ayudar a los traductores a mejorar la calidad y productividad, promover los estándares abiertos y ayudar a cerrar la brecha digital proporcionando las herramientas necesarias para que los usuarios finales puedan utilizar los programas y acceder a la documentación en su lengua materna.



Figura 3.27: Acerca de Open Language Tools.

- **Comunidad**

El proyecto no cuenta con una página web oficial y la única información disponible se encuentra en la forja Java.net (ver figura 3.29), donde el proyecto cuenta con 167 miembros registrados. En marzo de 2012 había un desarrollador activo; la lista de usuarios contaba con 40 suscriptores y se enviaron dos mensajes durante 2011 y ninguno en lo que va de 2012.

Se valora positivamente que el proyecto cuente con un desarrollador activo, con miembros registrados y con un espacio de comunicación para los usuarios. Por el contrario, se valora negativamente que el proyecto no cuente con una página web oficial y el bajo nivel de actividad en el foro de usuarios.

## Philosophy

We believe that computers are tools that can help people - in our case, we want computers to help translators. Anything that can be done to help translators improve the quality of their work, or reduce the amount of time it takes to do translation is definitely within the scope of this project. We strongly believe in the value of shared open standards - everything we do should be based on open standards, and should interoperate via those standards.

Lastly writing translation tools is a noble cause : we believe that we should be doing all we can to break down the *Digital divide*, and language is one aspect of that. If you can't use a computer because it's not translated to your native language, then we want to provide tools to fix that problem.

Figura 3.28: Open Language Tools – filosofía del proyecto.

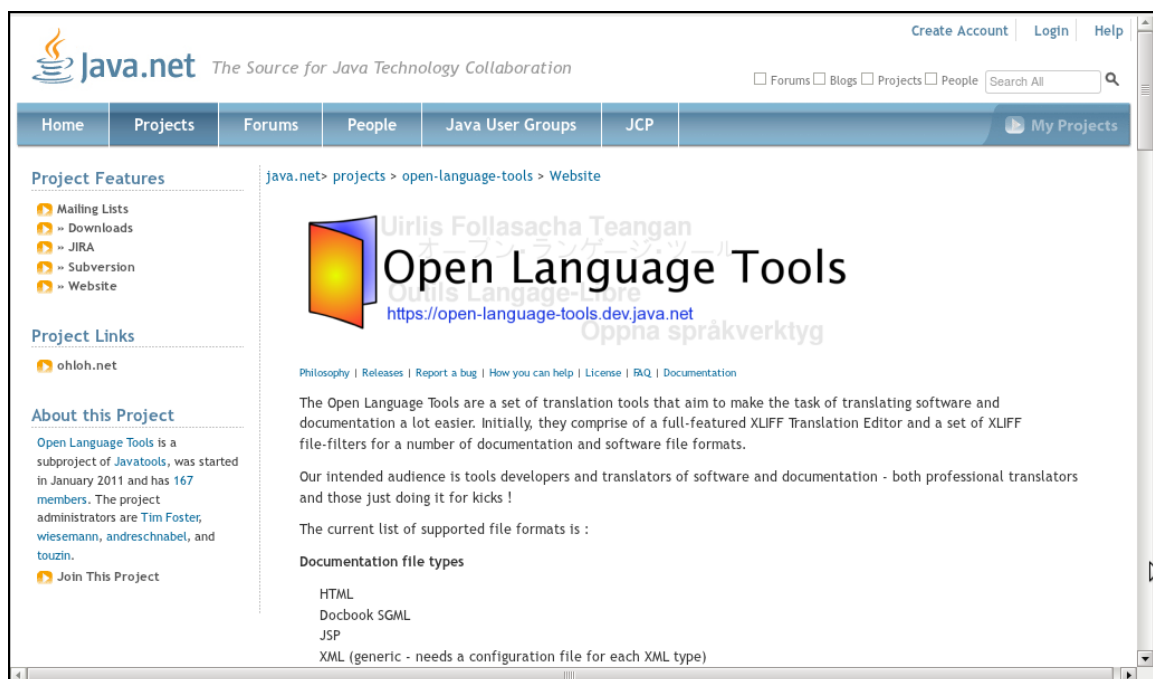


Figura 3.29: Open Language Tools – página del proyecto en la forja Java.net.

### • *Madurez*

El código fue liberado en junio de 2004 y se encuentra registrado en una forja pública. En el 2011 no se lanzó ninguna versión y la versión disponible en la actualidad es una versión beta (1.4.0\_dev) del 19 de enero de 2012. El proyecto utiliza un repositorio con control de revisiones (SVN) para la gestión del código fuente y para la gestión de fallos se utiliza la lista de correo de usuarios. Los procedimientos para contribuir a la localización de la herramienta y el historial de versiones (ver figura 3.30) están documentados. También existe un proceso documentado para obtener asistencia técnica.

open-language-tools			
Summary	<b>Versions</b>		
Issues	<a href="#">current</a>		Editor and Filters
Road Map	<a href="#">1.4.0</a>	Release Date: 29/Sep/10	Editor
Change Log	<a href="#">1.3.1</a>	Release Date: 15/Mar/10	Editor
Popular Issues	<a href="#">1.3.0</a>	Release Date: 20/Sep/09	Editor
Labels	<a href="#">1.2.9</a>		
<b>Versions</b>	<a href="#">1.2.8</a>		
Components	<a href="#">1.2.7</a>	Release Date: 22/Dec/06	Editor and Filters
Agile	<a href="#">1.2.6</a>		
	<a href="#">1.2.5</a>		
	<a href="#">1.2.4</a>	Release Date: 19/Apr/06	Editor and Filters

Figura 3.30: Open Language Tools – historial de versiones

En este caso se valora positivamente la edad del proyecto y que se gestione en una forja pública con un sistema de control de versiones. Por otro lado, se valora negativamente que no esté disponible una versión estable de la herramienta y que no se haya lanzado ninguna actualización durante todo un año.

- **Reputación**

Se encontró una publicación sobre el proyecto. No se encontró información sobre el número de descargas y tampoco se encontraron comentarios recientes en Twitter ni discusiones en foros para traductores. El proyecto no está listado en directorios de software pero sí está registrado en Ohloh.

Se valora negativamente la ausencia de información sobre el número de descargas, así como el hecho de que el proyecto no aparezca en directorios de software y que no se hayan encontrado comentarios en Twitter ni en foros para traductores. Por otro lado, se valora positivamente haber encontrado una publicación sobre el proyecto y que este cuente con un perfil en Ohloh, aunque no se ha actualizado recientemente.

- **Valoración general del proyecto**

En la tabla 3.23 se presenta la valoración general de Open Language Tools. Según los resultados obtenidos, la comunidad y la madurez del proyecto se consideran pobres (una estrella), razón por la cual se procedió con la evaluación de la herramienta (ver apartado 3.4.3.2.5).

Comunidad		
		
Capacidad de mantenimiento	Sostenibilidad	Recursos y servicios disponibles

Madurez		
		
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.23: Valoración general del proyecto Open Language Tools.

### 3.4.3.1.9 OpenTM2

- **Estrategia del proyecto**

El proyecto era desarrollado originalmente por IBM como software privativo. Desde la liberación del código fuente, el proyecto ha sido liderado por un Comité Directivo integrado por representantes de las organizaciones vinculadas al grupo de interés formado por IBM Corporation, LISA, Linux Solution Group, Welocalize y Cisco. La mayor parte del desarrollo se ha llevado a cabo con base en la ética laboral tradicional. El software y todas sus funciones asociadas están disponibles bajo la licencia EPL (Eclipse Public License, sin *copyleft*). La titularidad de los derechos está a nombre de IBM y otros (ver figura 3.31). El proyecto trabaja con base en objetivos formales predefinidos y se puede describir como una dictadura benevolente. Las principales decisiones en lo que tiene que ver con el plan de desarrollo, la solución de cuellos de botella, la resolución de conflictos y el éxito técnico en general recaen en el Comité Directivo y el proyecto cuenta con una hoja de ruta predefinida.

- **Comunidad**

El proyecto cuenta con una página web donde se recoge la información pertinente (ver figura 3.32). Varias organizaciones respaldan el proyecto (ver apartado anterior) y el desarrollo es llevado a cabo por un grupo de desarrolladores que gestiona el ciclo del software de manera formal con base en la asignación de roles. En marzo de 2012, había 3 desarrolladores activos y el grupo de usuarios contaba con 48 suscriptores. El número promedio de mensajes por mes durante 2011 fue de 4. El proyecto también cuenta con una *wiki* y un foro para desarrolladores.

Se valora positivamente que el proyecto cuente con respaldo institucional y que exista una página web oficial donde se presenta la información de manera organizada y exhaustiva. También se valora positivamente que el proyecto cuente con desarrolladores activos y con un grupo de usuarios y otros espacios de comunicación. No obstante, se valora negativamente el bajo nivel de actividad en el grupo de usuarios.

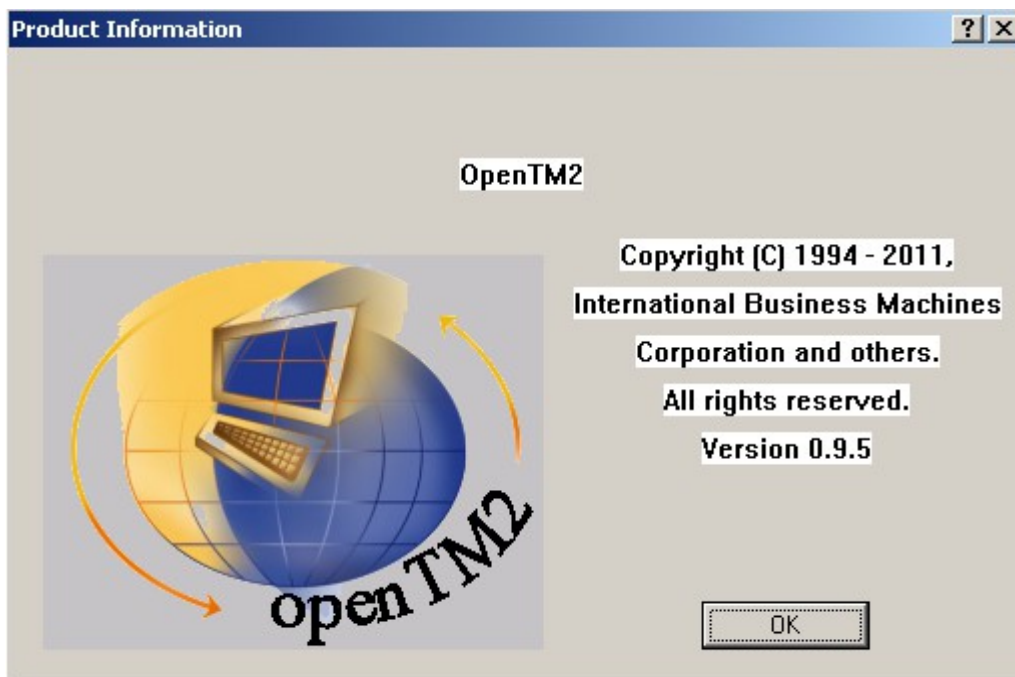


Figura 3.31: Acerca de OpenTM2.

- **Madurez**

El proyecto fue registrado en SourceForge el 4 de junio de 2010, aunque su desarrollo de forma privativa en IBM Corporation data de principios de la década

de los noventa. La política de numeración de versiones está documentada y se lleva también un registro con el historial de los cambios incluidos en cada versión. El estado actual de desarrollo es estable. En 2011 lanzaron una versión estable y una actualización del programa. La última versión disponible (0.9.5) fue lanzada el 29 de julio de 2011. El proyecto gestiona el código en un repositorio con control de revisiones (SVN) y utiliza un sistema de gestión de fallos y solicitudes de nuevas funciones.

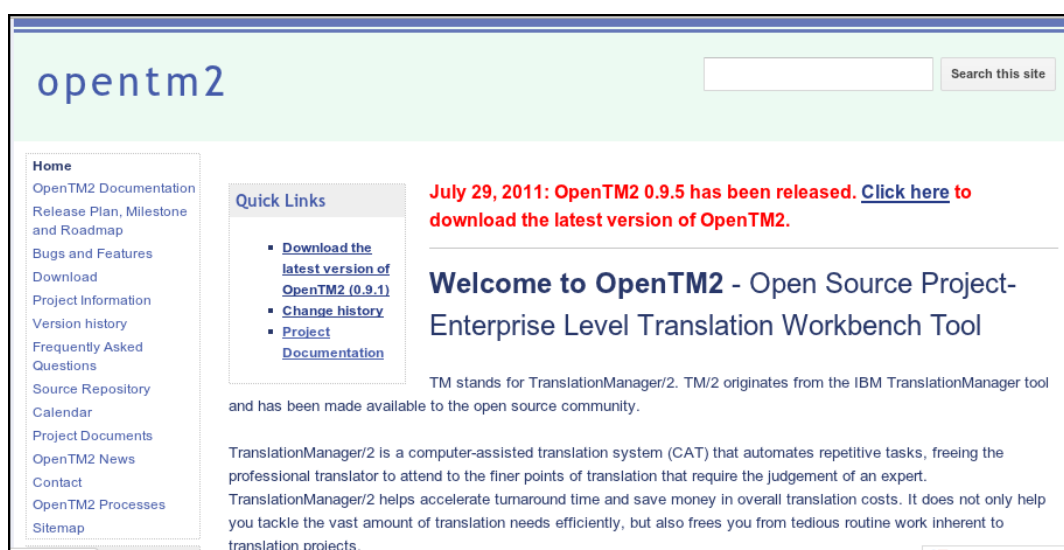


Figura 3.32: OpenTM2 – página web del proyecto.

Se valora positivamente la edad del proyecto y su estado actual de desarrollo, así como el hecho de que esté registrado en una forja pública y que se utilicen herramientas específicas para la gestión del código y de los informes de fallos y solicitudes de nuevas funciones. También se valora positivamente que en el 2011 se hayan lanzado una versión y una actualización de la herramienta, así como el hecho de que la versión más reciente del programa no tenga más de un año.

- **Reputación**

Se encontraron tres entradas sobre el programa en blogs de traductores. También se encontraron discusiones sobre la herramienta en foros para traductores (ProZ y CATMT). El proyecto aparece listado en el directorio de software de TAUS, cuenta con un perfil en Ohloh y también se encontraron comentarios en Twitter. No se

pudo encontró información sobre el número de descargas ni se encontraron recomendaciones en la forja.

En este caso se valora positivamente la existencia de entradas sobre el proyecto, de discusiones en foros de traductores y comentarios en Twitter, así como el hecho de que el proyecto esté incluido en un directorio de software para traductores y cuente con un perfil en la plataforma Ohloh. Se valora negativamente la ausencia de información pública sobre el número de descargas y de recomendaciones en la forja.

- **Valoración general**

En la tabla 3.24 se presenta la valoración general de OpenTM2. Según los resultados obtenidos, la comunidad del proyecto se considera aceptable (dos estrellas), mientras que la madurez del proyecto se considera satisfactoria (tres estrellas), de modo que se llevó a cabo la evaluación de la herramienta (ver apartado 3.4.3.2.6).

Comunidad		
★	★	★
Capacidad de mantenimiento	Sostenibilidad	Recursos y servicios disponibles

Madurez		
★	★	★
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.24: Valoración general del proyecto OpenTM2.

### 3.4.3.1.10 Transolution

- **Estrategia del proyecto**

El proyecto surgió como una iniciativa independiente del desarrollador Fredrik Corneliusson, titular de los derechos. El desarrollo se llevaba a cabo de manera voluntaria en consonancia con la ética *hacker*. El software está disponible bajo una



licencia libre (GNU GPL, con *copyleft* fuerte). No se encontró información sobre integración del código de otros proyectos libres.

- ***Comunidad***

La página oficial del proyecto ya no está disponible y la información existente se limita a la incluida en la forja de desarrollo. El proyecto estaba a cargo de un desarrollador independiente, pero no se encontró evidencia de actividad de desarrollo reciente y tampoco existe un foro u otro espacio de comunicación para los usuarios. Tampoco se encontró información sobre la posible oferta de servicios basados en el software.

Se valora negativamente la ausencia de una página web oficial, así como el hecho de que no haya desarrolladores activos ni espacios de comunicación para los usuarios. De igual manera, se valora negativamente la ausencia de servicios profesionales para los usuarios.

- ***Madurez***

El proyecto fue registrado en la forja SourceForge el 24 de febrero de 2005 y la última versión disponible (0.4b5) fue lanzada el 21 de agosto del mismo año. Según su creador, el estado de desarrollo era beta, aunque en SourceForge aparece clasificado como prealfa; el estado actual es inactivo. No se utilizaba un repositorio con sistema de control de versiones para la gestión del código, ni herramientas específicas para la gestión de los informes de fallos y solicitudes de nuevas versiones.

En este caso se valora positivamente la edad del proyecto, así como el hecho de que esté registrado en una forja pública. Por el contrario, se valora negativamente que no se registre actividad de desarrollo reciente y que no se utilizaran herramientas específicas para la gestión del proyecto.

- **Reputación**

En marzo de 2012 el programa fue descargado 275 veces. El proyecto aparece incluido en dos directorios de software. No se encontraron publicaciones ni discusiones sobre el proyecto en foros para traductores o comentarios en Twitter. El proyecto tampoco está registrado en Ohloh.

Se valora positivamente el número de descargas y que el proyecto esté incluido en directorios de software. Por otro lado, se valora negativamente que no existan publicaciones, discusiones o comentarios sobre el proyecto y que no esté registrado en Ohloh.

- **Valoración general**

En la tabla 3.25 se presenta la valoración general del proyecto Transolution. Ya que según los datos encontrados tanto la comunidad como la madurez del proyecto son inaceptables (cero estrellas), no se llevó a cabo la evaluación de la herramienta.

Comunidad		
 Capacidad de mantenimiento	 Sostenibilidad	 Recursos y servicios disponibles

Madurez		
 Estado del proyecto	 Gestión del proyecto	 Gestión de versiones

Tabla 3.25: Valoración general de Transolution.

### 3.4.3.1.11 Virtaal

- **Estrategia del proyecto**

El proyecto, surgido gracias a financiación pública, es liderado por Translate.org.za, parte de una entidad sin ánimo de lucro (Zuza Software Foundation) creada con el objetivo de localizar software libre a las once lenguas oficiales de Sudáfrica. Se trata de una comunidad híbrida, donde parte del desarrollo es llevado a cabo por

miembros de una organización pero también se cuenta con la participación de voluntarios. El software y todas sus funciones adicionales está disponible bajo la licencia GNU GPL (con copyleft fuerte). El titular de los derechos de autor es la Zuza Software Foundation (ver figura 3.33). El proyecto trabaja con base en objetivos formales y se puede describir como una dictadura benevolente donde las decisiones se toman de forma centralizada. Existe una hoja de ruta y el proyecto desarrolla otras herramientas libres (Pootle, Translate Toolkit, Corpus Catcher y Spelt).



Figura 3.33: Acerca de Virtaal.

- **Comunidad**

El proyecto cuenta con una página web donde se recoge la información de las herramientas que desarrollan (ver figura 3.34). La comunidad es liderada por una entidad sin ánimo de lucro. En marzo de 2012 había 4 desarrolladores activos. No existe un foro de usuarios específico para la herramienta, pero el proyecto sí utiliza una lista de correo para desarrolladores que registra actividad en los últimos meses. El proyecto tiene un perfil en Facebook y también utiliza un canal de chat IRC para

la comunicación, especialmente entre desarrolladores. No se encontró información sobre oferta de servicios basados en el software.

En este caso se valora positivamente la existencia de una página web oficial, así como el número de desarrolladores activos. También se valora positivamente la existencia de varios espacios de comunicación, aunque estos, con excepción del perfil en Facebook, están más enfocados a los desarrolladores que a los usuarios.

- ***Madurez***

El proyecto fue registrado en la forja SourceForge el 16 de octubre de 2008. El estado de desarrollo actual es estable. En el año 2011 lanzaron una versión estable y una actualización de la herramienta. La última versión disponible (0.7.1) fue lanzada el 9 de enero de 2012. El código fuente se gestiona en un repositorio público con control de revisiones (SVN) y el proyecto utiliza las herramientas de gestión de fallos que ofrece SourceForge. Existen procedimientos documentados para contribuir con la localización y con las pruebas (*testing*).

Se valora positivamente la edad del proyecto y su estado actual de desarrollo, así como el hecho de que se hayan lanzado dos versiones en 2011 y que la última versión disponible tenga menos de un año. Además, se valora positivamente la utilización de una forja de desarrollo y de herramientas específicas para la gestión del proyecto.

- ***Reputación***

En marzo de 2012 el programa fue descargado 1152 veces. El promedio de descargas durante la semana siguiente al lanzamiento de las tres últimas versiones fue 271, lo que sirve como indicador del número de usuarios regulares de la herramienta. Se encontró una entrada sobre el proyecto en un blog y aparece listado en varios directorios de software. Además, el programa se incluye en los repositorios de varias distribuciones de GNU/Linux. Las puntuaciones de los usuarios en SourceForge son positivas en su totalidad (38 recomendaciones). El

proyecto está registrado en Ohloh y se encontraron comentarios recientes en Twitter.

**Virtaal**

Virtaal is a graphical translation tool. It is meant to be easy to use and powerful at the same time. Although the initial focus is on software translation (localisation or l10n), we definitely intend it to be useful for several purposes.

Virtaal is built on the powerful API of the [Translate Toolkit](#). "Virtaal" is an Afrikaans play on words meaning "For Language", but also refers to translation.

Read more about the [features](#) in Virtaal, or view the [screenshots](#). You can also download a [screencast](#) (33MB in Ogg Theora format) to see some of these features in action.

Learn more about [using Virtaal](#), available [shortcuts](#) and some extra [tips](#) for people who want to customise their installation.

**Installation**

Platform	Instructions	Notes
Windows	<a href="#">Download Virtaal setup.exe</a>	Includes all dependencies
Mac OS X	<a href="#">Download Virtaal .dmg</a>	Beta release. OS X 10.5 and greater
Ubuntu	<a href="#">Translate.org.za PPA</a> <a href="#">Tools</a>	Also available in software center
Fedora	yum install virtaal	Or use the graphical package manager. For older Fedora releases use the <a href="#">custom repo</a> .
Mandriva	urpmi virtaal	Or simply use the graphical package manager
Debian Squeeze	apt-get install virtaal	Or simply use the graphical package manager
openSUSE	Read <a href="#">openSUSE installation</a> instructions	
Other	<a href="#">Download source .zip or .tar.bz2</a>	Make sure you have all the dependencies including the latest Translate Toolkit

**Contact**

- Chat in our IRC channel [#pootle](#)
- [Report bugs](#)
- Join the [Translate-devel mailing list](#)

**Contributing**

There are many ways of contributing to Virtaal. Join the mailing list or IRC channel to join our effort. You can join our effort to distribute Virtaal by sharing informing with people, writing documentation or packaging for more platforms.

If you would like to contribute to the Virtaal software, you can start by reading the instructions on the following pages:

- [Localising Virtaal](#)
- [Building](#)
- [Testing](#)
- [Development Plans](#)
- [Suggestions](#)

Figura 3.34: Virtaal – página web.

Se valora positivamente el número de descargas, así como la existencia de publicaciones, discusiones, recomendaciones y comentarios sobre el proyecto. De igual manera, se valora positivamente que se incluya el programa en algunas distribuciones de GNU/Linux y que el proyecto cuente con un perfil actualizado en Ohloh.

- **Valoración general del proyecto**

La tabla 3.26 presenta la valoración general de Virtaal luego de consolidar los resultados. Según lo encontrado, la comunidad se considera pobre (una estrella), mientras que la madurez del proyecto se considera satisfactoria (tres estrellas), de modo que se llevó a cabo la evaluación de la herramienta (ver apartado 3.4.3.2.7).

Comunidad		
		
Capacidad de mantenimiento	Sostenibilidad	Recursos y servicios disponibles

Madurez		
		
Estado del proyecto	Gestión del proyecto	Gestión de versiones

Tabla 3.26: Valoración general del proyecto Virtaal.

- **Conclusión**

En total, de los once proyectos evaluados, se encontraron cuatro cuyas comunidad y madurez se calificaron como inacceptables (EsperantiloTM, ForeignDesk, Frankenstein Translation Memory Suite y Transolution). Por consiguiente, las herramientas desarrolladas por estos proyectos no se evaluaron.

### 3.4.3.2 Evaluación del software como producto

En este apartado se presentan los resultados correspondientes a las siete herramientas evaluadas: Anaphraseus, Autshumato ITE, OmegaT, OmegaT+, Open Language Tools, OpenTM2 y Virtaal.

Tal como se explicó en los apartados 3.3.5.2 y 3.3.5.3, en la evaluación del software como producto se valoran las características de portabilidad, usabilidad y funcionalidad. Para las características de portabilidad y usabilidad se definieron unas puntuaciones cuantitativas por atributo que luego se consolidan en una puntuación total por subcaracterística. Si las puntuaciones totales superan los criterios de aceptación definidos para cada subcaracterística se asignan las estrellas correspondientes. Para la funcionalidad, en cambio, no se utilizan puntuaciones y solo se presenta la lista de características ofrecidas por la herramienta sin entrar a valorar si la implementación de las funciones es más o menos adecuada.

Para presentar los resultados de la evaluación de las herramientas, hemos organizado la información correspondiente a cada característica en forma de tabla, pues entendemos que de este modo se facilita la visualización de los datos.

#### 3.4.3.2.1 Anaphraseus

Se evaluó la versión 2.04b (99) de la herramienta.

##### • *Portabilidad*

Los resultados de la evaluación de la portabilidad de Anaphraseus se presentan en la tabla 3.27.

<b>Portabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Adaptabilidad	Modularidad	El diseño de la herramienta no permite el desarrollo de componentes independientes.	1
	Escalabilidad	El sistema puede ser implementado a gran escala, pero no está pensado para entornos multiusuario.	2
Facilidad de instalación	Tiempo necesario para la instalación	Menos de 15 minutos.	3

<b>Portabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
	Requisitos de sistema	Es necesario instalar otro programa para poder utilizar la herramienta.	1
	Conocimientos técnicos necesarios para la instalación	La instalación es sencilla y no hay que realizar configuraciones adicionales.	3
Coexistencia	Compatibilidad con el entorno existente	El programa puede ser utilizado en el entorno existente.	3
	Posibilidad de integración	El programa se puede integrar fácilmente al flujo de trabajo actual.	3

Tabla 3.27: Anaphraseus – portabilidad.

• **Usabilidad**

La tabla 3.28 presenta los resultados correspondientes a la usabilidad de Anaphraseus.

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Interfaz de usuario	Distribución de la interfaz de usuario	La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.	3
	Disponibilidad en el idioma requerido	La localización es parcial (interfaz en el idioma requerido pero documentación sin traducir).	2
	Comprensibilidad de los iconos y nombres de funciones	Es fácil relacionar los nombres de funciones y menús con las acciones que realizan y se incluyen textos alternativos ( <i>tooltips</i> ) para explicar las funciones de los iconos.	3
	Configurabilidad de la apariencia	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	2
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	La interfaz está diseñada para permitir la navegación y operación solo con el teclado sin depender del ratón.	3
	Existencia de ayuda contextual	El programa no cuenta con ayuda accesible dentro de la interfaz; es necesario abrir un manual de usuario en otra aplicación o recurrir a otras fuentes (ej. página web o <i>wiki</i> ).	1
	Indicadores de progreso y mensajes de error	Se incluyen barras de progreso, mensajes de error u otros mensajes informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.	3



<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Documentación	Existencia de varios tipos de documentación	Se dispone de un archivo Léame y por lo menos otra guía o manual de usuario.	2
	Actualización de la documentación	Algunas partes de la documentación no corresponden a la versión actual del programa.	1
	Exhaustividad de la documentación	La documentación cubre de manera clara todas las funciones del programa.	3
	Existencia de recursos multimedia	La documentación incluye algunas capturas de pantalla u otros gráficos.	2

Tabla 3.28: Anaphraseus – usabilidad.

• **Funcionalidad**

La tabla 3.29 presenta los resultados correspondientes a la funcionalidad ofrecida por Anaphraseus.




<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de programa	Opciones del proyecto: Pretraducción de documentos Coincidencias parciales Inserción automática de coincidencias exactas Inserción automática de coincidencias parciales Coincidencias de glosarios
		Opciones del editor: Búsqueda y reemplazo global Búsqueda de concordancias en archivos originales Autocompletar al escribir Corrector ortográfico al escribir Corrector ortográfico bajo demanda Corrector gramatical / de estilo al escribir Corrector gramatical / de estilo bajo demanda Vista previa del formato
	Filtros de archivo implementados	Integración con aplicaciones externas: Integración con motores de traducción automática locales o vía web Búsqueda en recursos externos (locales o vía web)
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	Formatos de texto y ofimática: TXT, DOC, DOT, RTF, DOCX, DOTX, ODT, SXW
	Posibilidad de cambiar la segmentación durante la traducción Porcentaje mínimo de coincidencias configurable Diccionarios del corrector ortográfico personalizables Búsquedas y reemplazos basados en expresiones regulares	
Interoperabilidad	Soporte para estándares de intercambio de datos	Codificación Unicode Memorias TMX Glosarios como texto delimitado (CSV, TAB o TXT)
	Soporte para formatos abiertos generados por	TXT (WordFast)

Funcionalidad		
Subcaracterística	Atributo	Características presentes
	otras herramientas de traducción	

Tabla 3.29: Anaphraseus – funcionalidad.

- **Valoración general de la herramienta**

Según los resultados obtenidos, la portabilidad de Anaphraseus se considera aceptable (dos estrellas), mientras que la usabilidad de la herramienta se considera satisfactoria (tres estrellas). En la tabla 3.30 se resume la valoración general de la portabilidad y la usabilidad de Anaphraseus.

Portabilidad		
		
Adaptabilidad	Facilidad de instalación	Coexistencia




Usabilidad		
		
Interfaz de usuario	Documentación	Facilidad de uso

Tabla 3.30: Valoración general de la herramienta Anaphraseus.

### 3.4.3.2.2 Autshumato ITE

Se evaluó la versión 2.0.0 de la herramienta.

- **Portabilidad**

Los resultados de la evaluación de la portabilidad de Autshumato ITE se presentan en la tabla 3.31.

Portabilidad			
Subcaracterística	Atributo	Descripción	Puntuación
Adaptabilidad	Modularidad	El diseño de la herramienta permite el desarrollo de componentes independientes mediante una arquitectura de plugins.	3
	Escalabilidad	El sistema puede ser implementado a gran escala, pero no está pensado para entornos multiusuario.	2

<b>Portabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Facilidad de instalación	Tiempo necesario para la instalación	Menos de 15 minutos.	3
	Requisitos de sistema	Es necesario instalar otro programa para poder utilizar la herramienta.	1
	Conocimientos técnicos necesarios para la instalación	La instalación es sencilla y no hay que realizar configuraciones adicionales.	3
Coexistencia	Compatibilidad con el entorno existente	El programa puede ser utilizado en el entorno existente.	3
	Posibilidad de integración	El programa se puede integrar fácilmente al flujo de trabajo actual.	3

Tabla 3.31: Autshumato ITE – portabilidad.

- **Usabilidad**

La tabla 3.32 presenta los resultados de la usabilidad de la herramienta.

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Interfaz de usuario	Distribución de la interfaz de usuario	La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.	3
	Disponibilidad en el idioma requerido	La localización es parcial (interfaz en el idioma requerido pero documentación sin traducir).	2
	Comprensibilidad de los iconos y nombres de funciones	Es fácil relacionar los iconos y nombres de funciones y menús con las acciones que realizan.	2
	Configurabilidad de la apariencia	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	2
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	El programa permite la navegación y operación con el teclado aunque para algunas acciones es necesario el ratón.	2
	Existencia de ayuda contextual	El programa incluye ayuda sobre la mayoría de las funciones y es posible acceder a ella sin salir de la aplicación.	3
	Indicadores de progreso y mensajes de error	Se incluyen barras de progreso, mensajes de error u otros mensajes informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.	3
Documentación	Existencia de varios tipos de documentación	Se dispone de un archivo Léame y por lo menos otra guía o manual de usuario.	2

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
	Actualización de la documentación	Toda la documentación está actualizada con la versión actual del programa.	3
	Exhaustividad de la documentación	La documentación cubre de manera clara todas las funciones del programa.	3
	Existencia de recursos multimedia	La documentación incluye algunas capturas de pantalla u otros gráficos.	2

Tabla 3.32: Autshumato ITE – usabilidad.

• **Funcionalidad**

La tabla 3.33 presenta los resultados correspondientes a la funcionalidad de Autshumato ITE.

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de programa	<p>Opciones del proyecto:</p> <p>Análisis de originales (recuento de palabras, coincidencias, repeticiones)</p> <p>Procesamiento por lotes</p> <p>Pretraducción de documentos</p> <p>Pretraducción con priorización de fuentes utilizadas</p> <p>Creación de proyectos con múltiples documentos de partida</p> <p>Coincidencias parciales</p> <p>Inserción automática de coincidencias exactas</p> <p>Inserción automática de coincidencias parciales</p> <p>Propagación automática de segmentos repetidos</p> <p>Coincidencias de glosario</p> <p>Múltiples glosarios por proyecto</p> <p>Posibilidad de utilizar las memorias en ambas direcciones</p> <p>Múltiples memorias por proyecto</p> <p>Múltiples traducciones para el mismo segmento original</p> <p>Memorias multilingües (más de dos idiomas)</p>
		<p>Opciones del editor:</p> <p>Visualización de metadatos de las coincidencias (fecha, ID de usuario, proyecto, etc.)</p> <p>Opción de navegación en el editor mediante filtros</p> <p>Posibilidad de añadir comentarios a los segmentos</p> <p>Estadísticas del proyecto (segmentos traducidos/sin traducir)</p> <p>Búsqueda de concordancias en archivos originales</p> <p>Búsqueda de concordancias en archivos de referencia</p> <p>Corrector ortográfico al escribir</p> <p>Corrector gramatical / de estilo al escribir</p> <p>Vista previa del formato</p> <p>Controles de calidad bajo demanda</p>
	Filtros de archivo implementados	<p>Integración con aplicaciones externas:</p> <p>Integración con motores de traducción automática locales o vía web</p> <p>Formatos de texto y ofimática: TXT, CSV, TAB, DOCX, DOTX, XLSX, PPTX, ODT, ODS, ODP, SXW, SRT, DocBook, LaTeX</p>

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
		Formatos de maquetación: XML (Infix), XTG (QuarkXPress), TAG (QuarkXPress) Formatos multimedia: SVG, XML (Flash export) Formatos de localización web: HTML, XML, RESX Formatos de localización de software: PO, Java Resource Bundles, TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	Filtros configurables Reglas de segmentación configurables Posibilidad de cambiar la segmentación durante la traducción Porcentaje mínimo de coincidencias configurable Diccionarios del corrector ortográfico personalizables Reglas del corrector lingüístico personalizables Búsquedas basadas en expresiones regulares Atajos de teclado configurables
Interoperabilidad	Soporte para estándares de intercambio de datos	Codificación Unicode Memorias TMX Bases de datos TBX Glosarios como texto delimitado (CSV, TAB o TXT)
	Soporte para formatos abiertos generados por otras herramientas de traducción	TXML (WordFast Pro)

Tabla 3.33: Autshumato ITE – funcionalidad.

• **Valoración general de la herramienta**

La portabilidad y usabilidad de Autshumato ITE resultaron ser satisfactorias (tres estrellas en cada caso). La tabla 3.34 presenta la consolidación de los resultados de la portabilidad y la usabilidad de la herramienta.

<b>Portabilidad</b>		
★	★	★
Adaptabilidad	Facilidad de instalación	Coexistencia

<b>Usabilidad</b>		
★	★	★
Interfaz de usuario	Documentación	Facilidad de uso

Tabla 3.34: Valoración general de la herramienta Autshumato ITE.

**3.4.3.2.3 OmegaT**

Como se explicó en el apartado 3.4.3.1.6, el proyecto OmegaT mantiene dos versiones paralelas: la estándar y la más reciente. Para la evaluación del producto se utilizó la versión estándar (2.5.4) por ser la recomendada para los usuarios que empiezan a usar la herramienta.

- **Portabilidad**

Los resultados de la evaluación de la portabilidad de OmegaT se presentan en la tabla 3.35.

<b>Portabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Adaptabilidad	Modularidad	El diseño de la herramienta permite el desarrollo de componentes independientes mediante una arquitectura de plugins.	3
	Escalabilidad	El sistema puede ser implementado a gran escala, pero no está pensado para entornos multiusuario.	2
Facilidad de instalación	Tiempo necesario para la instalación	Menos de 15 minutos.	3
	Requisitos de sistema	Los programas necesarios se incluyen en el paquete de instalación.	2
	Conocimientos técnicos necesarios para la instalación	La instalación es sencilla y no hay que realizar configuraciones adicionales.	3
Coexistencia	Compatibilidad con el entorno existente	El programa puede ser utilizado en el entorno existente.	3
	Posibilidad de integración	El programa se puede integrar fácilmente al flujo de trabajo actual.	3

Tabla 3.35: OmegaT – portabilidad.

- **Usabilidad**

La tabla 3.36 presenta los resultados correspondientes a la usabilidad de la herramienta.

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Interfaz de usuario	Distribución de la interfaz de usuario	La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.	3

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
	Disponibilidad en el idioma requerido	El programa está totalmente localizado al idioma requerido, tanto la interfaz de usuario como las ayudas y otra documentación incluida.	3
	Comprensibilidad de los iconos y nombres de funciones	Es fácil relacionar los iconos y nombres de funciones y menús con las acciones que realizan.	2
	Configurabilidad de la apariencia	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	2
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	El programa permite la navegación y operación con el teclado aunque para algunas acciones es necesario el ratón.	2
	Existencia de ayuda contextual	El programa incluye ayuda sobre la mayoría de las funciones y es posible acceder a ella sin salir de la aplicación.	3
	Indicadores de progreso y mensajes de error	Se incluyen barras de progreso, mensajes de error u otros mensajes informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.	3
Documentación	Existencia de varios tipos de documentación	El programa incluye varios tipos de documentación (guía rápida, manual de usuario, preguntas frecuentes, tutoriales).	3
	Actualización de la documentación	Toda la documentación está actualizada con la versión actual del programa.	3
	Exhaustividad de la documentación	La documentación cubre de manera clara todas las funciones del programa.	3
	Existencia de recursos multimedia	La documentación incluye algunas capturas de pantalla u otros gráficos.	2

Tabla 3.36: OmegaT – usabilidad.

• **Funcionalidad**

La tabla 3.37 presenta los resultados correspondientes a la funcionalidad de OmegaT.

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de programa	Opciones del proyecto: Análisis de originales (recuento de palabras, coincidencias, repeticiones) Procesamiento por lotes Pretraducción de documentos Pretraducción con priorización de fuentes utilizadas Pseudotraducción Creación de proyectos con múltiples documentos de partida Coincidencias parciales

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
		<p>Coincidencias basadas en contexto  Inserción automática de coincidencias exactas  Inserción automática de coincidencias parciales  Propagación automática de segmentos repetidos  Coincidencias de glosario  Múltiples glosarios por proyecto  Posibilidad de utilizar las memorias en ambas direcciones  Múltiples memorias por proyecto  Múltiples traducciones para el mismo segmento original  Memorias multilingües (más de dos idiomas)</p> <p>Opciones del editor:  Visualización de metadatos de las coincidencias (fecha, ID de usuario, proyecto, etc.)  Opción de navegación en el editor mediante filtros  Posibilidad de añadir comentarios a los segmentos  Estadísticas del proyecto (segmentos traducidos/sin traducir)  Búsqueda de concordancias en archivos originales  Búsqueda de concordancias en archivos de referencia  Corrector ortográfico al escribir  Corrector gramatical / de estilo al escribir  Controles de calidad bajo demanda</p> <p>Integración con aplicaciones externas:  Integración con motores de traducción automática locales o vía web</p>
	Filtros de archivo implementados	<p>Formatos de texto y ofimática: TXT, CSV, TAB, DOCX, DOTX, XLSX, XLTX, XLSM, PPTX, PPSX, POTX, ODT, ODS, ODP, SXW, SXC, SXI, SRT, DocBook, LaTeX  Formatos de maquetación: XML (Infix), IDML (InDesign), XTG (QuarkXPress), TAG (QuarkXPress)  Formatos multimedia: SVG, XML (Flash export), CAMPROJ (Camstasia Studio)  Formatos de localización web: HTML, XML, RESX, JSON  Formatos de localización de software: RC, POT, PO, Java Resource Bundles, XML (Android resource), TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)</p>
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	<p>Filtros configurables  Reglas de segmentación configurables  Porcentaje mínimo de coincidencias configurable  Diccionarios del corrector ortográfico personalizables  Reglas del corrector lingüístico personalizables  Búsquedas basadas en expresiones regulares  Controles de calidad configurables  Atajos de teclado configurables</p>
Interoperabilidad	Soporte para estándares de intercambio de datos	<p>Codificación Unicode  Memorias TMX  Bases de datos TBX  Glosarios como texto delimitado (CSV, TAB o TXT)  Archivos pretraducidos XLIFF</p>
	Soporte para formatos abiertos generados por otras herramientas de traducción	TXML (WordFast Pro)

Tabla 3.37: OmegaT – funcionalidad.



- **Valoración general de la herramienta**

Según los datos obtenidos, tanto la portabilidad como la usabilidad de la herramienta se consideran satisfactorias (tres estrellas). La tabla 3.38 presenta la consolidación de los resultados para la portabilidad y la usabilidad de OmegaT.

Portabilidad		
★	★	★
Adaptabilidad	Facilidad de instalación	Coexistencia

Usabilidad		
★	★	★
Interfaz de usuario	Documentación	Facilidad de uso

Tabla 3.38: Valoración general de la herramienta OmegaT.

### 3.4.3.2.4 OmegaT+

Se evaluó la versión 1.0M3.1 de la herramienta.

- **Portabilidad**

Los resultados de la evaluación de la portabilidad de OmegaT+ se presentan en la tabla 3.39.

Portabilidad			
Subcaracterística	Atributo	Descripción	Puntuación
Adaptabilidad	Modularidad	El diseño de la herramienta permite el desarrollo de componentes independientes que se pueden integrar al sistema pero no hay documentación disponible.	2
	Escalabilidad	El sistema puede ser implementado a gran escala, pero no está pensado para entornos multiusuario.	2
Facilidad de instalación	Tiempo necesario para la instalación	Menos de 15 minutos.	3
	Requisitos de sistema	Es necesario instalar otro programa para poder utilizar la herramienta.	1
	Conocimientos técnicos necesarios para la instalación	La instalación es sencilla y no hay que realizar configuraciones adicionales.	3
Coexistencia	Compatibilidad con el entorno existente	El programa puede ser utilizado en el entorno existente.	3

<b>Portabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
	Posibilidad de integración	El programa se puede integrar fácilmente al flujo de trabajo actual.	3

Tabla 3.39: OmegaT+ – portabilidad.

• **Usabilidad**

La tabla 3.40 presenta los resultados correspondientes a la usabilidad de OmegaT+.

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Interfaz de usuario	Distribución de la interfaz de usuario	La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.	3
	Disponibilidad en el idioma requerido	La localización es parcial (interfaz en el idioma requerido pero documentación sin traducir).	2
	Comprensibilidad de los iconos y nombres de funciones	Es fácil relacionar los iconos y nombres de funciones y menús con las acciones que realizan.	2
	Configurabilidad de la apariencia	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	2
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	El programa permite la navegación y operación con el teclado aunque para algunas acciones es necesario el ratón.	2
	Existencia de ayuda contextual	El programa no cuenta con ayuda accesible dentro de la interfaz; es necesario abrir un manual de usuario en otra aplicación o recurrir a otras fuentes (ej. página web o wiki).	1
	Indicadores de progreso y mensajes de error	Se incluyen barras de progreso, mensajes de error u otros mensajes informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.	3
Documentación	Existencia de varios tipos de documentación	El programa incluye varios tipos de documentación (guía rápida, guía de instalación, manual de usuario).	3
	Actualización de la documentación	Toda la documentación está actualizada con la versión actual del programa.	3
	Exhaustividad de la documentación	La documentación cubre de manera clara todas las funciones del programa.	3
	Existencia de recursos multimedia	La documentación incluye algunas capturas de pantalla u otros gráficos.	2

Tabla 3.40: OmegaT+ – usabilidad.

• **Funcionalidad**

La tabla 3.41 presenta los resultados correspondientes a la funcionalidad de OmegaT+.

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de programa	Opciones del proyecto: Análisis de originales (recuento de palabras, coincidencias, repeticiones) Creación de proyectos con múltiples documentos de partida Coincidencias parciales Inserción automática de coincidencias exactas Inserción automática de coincidencias parciales Propagación automática de segmentos repetidos Coincidencias de glosario Múltiples glosarios por proyecto Posibilidad de utilizar las memorias en ambas direcciones Múltiples memorias por proyecto Memorias multilingües (más de dos idiomas)
		Opciones del editor: Estadísticas del proyecto (segmentos traducidos/sin traducir) Búsqueda de concordancias en archivos originales Búsqueda de concordancias en archivos de referencia Controles de calidad bajo demanda
		Integración con aplicaciones externas: Integración con motores de traducción automática locales o vía web
	Filtros de archivo implementados	Formatos de texto y ofimática: TXT, CSV, TAB, DOCX, XLSX, PPTX, ODT, ODS, ODP, SXW, SXC, SXI, SRT, DocBook, LaTeX Formatos de maquetación: XTG (QuarkXPress), TAG (QuarkXPress) Formatos de localización web: HTML, XML, RESX Formatos de localización de software: RC, POT, PO, Java Resource Bundles, XML (Android resource), TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	Filtros configurables Reglas de segmentación configurables Posibilidad de cambiar la segmentación durante la traducción Porcentaje mínimo de coincidencias configurable Búsquedas basadas en expresiones regulares
Interoperabilidad	Soporte para estándares de intercambio de datos	Codificación Unicode Memorias TMX Glosarios como texto delimitado (CSV, TSV) Archivos pretraducidos XLIFF
	Soporte para formatos abiertos generados por otras herramientas de traducción	-

Tabla 3.41: OmegaT+ – funcionalidad.

- **Valoración general de la herramienta**

Según los resultados obtenidos, la portabilidad y la usabilidad de la herramienta son satisfactorias (tres estrellas). En la tabla 3.42 se presenta la consolidación de los resultados correspondientes a la portabilidad y la usabilidad de OmegaT+.

Portabilidad		
★	★	★
Adaptabilidad	Facilidad de instalación	Coexistencia

Usabilidad		
★	★	★
Interfaz de usuario	Documentación	Facilidad de uso

Tabla 3.42: Valoración general de la herramienta OmegaT+.

### 3.4.3.2.5 Open Language Tools

Se evaluó la versión 1.4.0\_dev.

- **Portabilidad**

Los resultados de la evaluación de la portabilidad de Open Language Tools se presentan en la tabla 3.43.

Portabilidad			
Subcaracterística	Atributo	Descripción	Puntuación
Adaptabilidad	Modularidad	El diseño de la herramienta no permite el desarrollo de componentes independientes.	1
	Escalabilidad	El sistema puede ser implementado a gran escala, pero no está pensado para entornos multiusuario.	2
Facilidad de instalación	Tiempo necesario para la instalación	Menos de 15 minutos.	3
	Requisitos de sistema	Es necesario instalar otro programa para poder utilizar la herramienta.	1
	Conocimientos técnicos necesarios para la instalación	La instalación es sencilla y no hay que realizar configuraciones adicionales.	3
Coexistencia	Compatibilidad con el entorno existente	El programa puede ser utilizado en el entorno existente.	3

<b>Portabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
	Posibilidad de integración	El programa se puede integrar al flujo de trabajo actual implementando algunas soluciones provisionales.	2

Tabla 3.43: *Open Language Tools* – portabilidad.

• **Usabilidad**

La tabla 3.44 presenta los resultados sobre la usabilidad de la herramienta.

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Interfaz de usuario	Distribución de la interfaz de usuario	La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.	3
	Disponibilidad en el idioma requerido	La localización es parcial (interfaz en el idioma requerido pero documentación sin traducir).	2
	Comprensibilidad de los iconos y nombres de funciones	Es fácil relacionar los nombres de funciones y menús con las acciones que realizan y se incluyen textos alternativos ( <i>tooltips</i> ) para explicar las funciones de los iconos.	3
	Configurabilidad de la apariencia	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	2
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	El programa permite la navegación y operación con el teclado aunque para algunas acciones es necesario el ratón.	2
	Existencia de ayuda contextual	El programa no cuenta con ayuda accesible dentro de la interfaz; es necesario abrir un manual de usuario en otra aplicación o recurrir a otras fuentes (ej. página web o wiki).	1
	Indicadores de progreso y mensajes de error	Se incluyen mensajes informativos o barras de progreso, pero en algunos casos el usuario no sabe si el programa está ocupado o si ha ocurrido un fallo.	2
Documentación	Existencia de varios tipos de documentación	Se dispone de un archivo Léame y por lo menos otra guía o manual de usuario.	2
	Actualización de la documentación	Toda la documentación se refiere a versiones anteriores del programa.	1
	Exhaustividad de la documentación	La documentación cubre de manera clara todas las funciones del programa.	3
	Existencia de recursos multimedia	La documentación incluye algunas capturas de pantalla u otros gráficos.	2

Tabla 3.44: *Open Language Tools* – usabilidad.

- **Funcionalidad**




La tabla 3.45 presenta los resultados correspondientes a la funcionalidad de Open Language Tools.

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de programa	<p>Opciones del proyecto:            Pretraducción de documentos            Coincidencias parciales            Inserción automática de coincidencias exactas            Inserción automática de coincidencias parciales            Múltiples traducciones para el mismo segmento original            Propagación automática de segmentos repetidos</p> <p>Opciones del editor:            Estadísticas del proyecto (segmentos traducidos/sin traducir)            Visualización de metadatos de las coincidencias (fecha, ID de usuario, proyecto, etc.)            Validación de segmentos mediante diferentes estatus            Opción de navegación en el editor mediante filtros            Posibilidad de añadir comentarios a los segmentos            Búsqueda y reemplazo global            Búsqueda de concordancias en archivos originales            Corrector ortográfico bajo demanda            Controles de calidad bajo demanda</p>
	Filtros de archivo implementados	<p>Formatos de texto y ofimática: TXT, ODT, ODS, ODP, SXW, SXC, SXI, DocBook            Formatos de localización web: HTML, XML, JSP            Formatos de localización de software: PO, Java Resource Bundles, DTD (Mozilla)</p>
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	<p>Diccionarios del corrector ortográfico personalizables            Controles de calidad configurables (etiquetas)            Atajos de teclado configurables</p>
Interoperabilidad	Soporte para estándares de intercambio de datos	<p>Codificación Unicode            Memorias TMX            Archivos pretraducidos XLIFF</p>
	Soporte para formatos abiertos generados por otras herramientas de traducción	-

Tabla 3.45: Open Language Tool – funcionalidad.

- **Valoración general de la herramienta**

Según los resultados, la portabilidad y la usabilidad de Open Language Tools se consideran aceptables (dos estrellas). La consolidación de los resultados para estas dos características se presenta en la tabla 3.46.

Portabilidad		
		
Adaptabilidad	Facilidad de instalación	Coexistencia




Usabilidad		
		
Interfaz de usuario	Documentación	Facilidad de uso

Tabla 3.46: Valoración general de la herramienta Open Language Tools.

### 3.4.3.2.6 OpenTM2

La versión evaluada fue la 0.9.5.

- **Portabilidad**

Los resultados de la evaluación de la portabilidad de OpenTM2 se presentan en la tabla 3.47.

Portabilidad			
Subcaracterística	Atributo	Descripción	Puntuación
Adaptabilidad	Modularidad	El diseño de la herramienta permite el desarrollo de componentes independientes mediante una arquitectura de plugins o una API pública bien documentada.	3
	Escalabilidad	El sistema puede ser implementado a gran escala y en entornos multiusuario.	3
Facilidad de instalación	Tiempo necesario para la instalación	Menos de 15 minutos.	3
	Requisitos de sistema	El programa no depende de software de terceros.	3
	Conocimientos técnicos necesarios para la instalación	La instalación es sencilla y no hay que realizar configuraciones adicionales.	3
Coexistencia	Compatibilidad con el entorno existente	El programa puede ser utilizado en el entorno existente.	3
	Posibilidad de integración	El programa se puede integrar al flujo de trabajo actual implementando algunas soluciones provisionales.	2

Tabla 3.47: OpenTM2 – portabilidad.

• **Usabilidad**

La tabla 3.48 presenta los resultados correspondientes a la usabilidad de la herramienta.

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Interfaz de usuario	Distribución de la interfaz de usuario	Toma un poco de tiempo entender la interfaz, la información está más o menos organizada; hay que usar un poco el manual.	2
	Disponibilidad en el idioma requerido	El programa y su documentación y ayuda solo están disponibles en un idioma distinto al requerido.	1
	Comprensibilidad de los iconos y nombres de funciones	Es fácil relacionar los nombres de funciones y menús con las acciones que realizan y se incluyen textos alternativos ( <i>tooltips</i> ) para explicar las funciones de los iconos.	3
	Configurabilidad de la apariencia	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	2
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	El programa permite la navegación y operación con el teclado aunque para algunas acciones es necesario el ratón.	2
	Existencia de ayuda contextual	El programa incluye una ayuda básica en la interfaz, pero para funciones avanzadas es necesario recurrir a otras fuentes (ej. manual o comunidad de usuarios).	2
	Indicadores de progreso y mensajes de error	Se incluyen barras de progreso, mensajes de error u otros mensajes informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.	3
Documentación	Existencia de varios tipos de documentación	El programa incluye varios tipos de documentación (guía rápida, guía de instalación, manual de usuario, manual para desarrolladores, preguntas frecuentes).	3
	Actualización de la documentación	Toda la documentación está actualizada con la versión actual del programa.	3
	Exhaustividad de la documentación	La documentación cubre de manera clara todas las funciones del programa.	3
	Existencia de recursos multimedia	La documentación incluye algunas capturas de pantalla u otros gráficos.	2

Tabla 3.48: OpenTM2 – usabilidad.

• **Funcionalidad**

La tabla 3.49 presenta los resultados correspondientes a la funcionalidad de OpenTM2.



<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de programa	<p>Opciones del proyecto:  Análisis de originales (recuento de palabras, coincidencias, repeticiones)  Pretraducción de documentos  Creación de proyectos con múltiples documentos de partida  Coincidencias parciales  Inserción automática de coincidencias exactas  Inserción automática de coincidencias parciales  Coincidencias de glosario  Múltiples glosarios por proyecto  Múltiples memorias por proyecto  Uso simultáneo de glosarios/memorias compartidas por red</p> <p>Opciones del editor:  Visualización de metadatos de las coincidencias (fecha, ID de usuario, proyecto, etc.)  Posibilidad de añadir comentarios a los segmentos  Estadísticas del proyecto (segmentos traducidos/sin traducir)  Búsqueda y reemplazo global  Búsqueda de concordancias en archivos originales  Búsqueda de concordancias en archivos de referencia  Corrector ortográfico al escribir  Corrector ortográfico bajo demanda  Vista previa del formato</p>
	Filtros de archivo implementados	Formatos de texto y ofimática: TXT, DOC, PPT, RTF, WordPerfect, Ami Pro, BookMaster® Formatos de localización web: HTML, SGML
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	Porcentaje mínimo de coincidencias configurable Posibilidad de cambiar la segmentación durante la traducción Diccionarios del corrector ortográfico personalizables Atajos de teclado configurables
Interoperabilidad	Soporte para estándares de intercambio de datos	Codificación Unicode

Tabla 3.49: OpenTM2 – funcionalidad.

• **Valoración general de la herramienta**

Luego de consolidar los resultados por subcaracterística, la portabilidad y usabilidad de OpenTM2 se consideran satisfactorias (ver tabla 3.50).



Usabilidad		
		
Interfaz de usuario	Documentación	Facilidad de uso

Tabla 3.50: Valoración general de la herramienta OpenTM2.

### 3.4.3.2.7 Virtaal

Se evaluó la versión 0.7.1 de la herramienta.

#### • *Portabilidad*

Los resultados de la evaluación de la portabilidad de Virtaal se presentan en la tabla 3.51.

Portabilidad			
Subcaracterística	Atributo	Descripción	Puntuación
Adaptabilidad	Modularidad	El diseño de la herramienta permite el desarrollo de componentes independientes mediante una API pública bien documentada.	3
	Escalabilidad	El sistema puede ser implementado a gran escala, pero no está pensado para entornos multiusuario.	2
Facilidad de instalación	Tiempo necesario para la instalación	Menos de 15 minutos.	3
	Requisitos de sistema	Los programas necesarios se incluyen en el paquete de instalación.	2
	Conocimientos técnicos necesarios para la instalación	La instalación es sencilla y no hay que realizar configuraciones adicionales.	3
Coexistencia	Compatibilidad con el entorno existente	El programa puede ser utilizado en el entorno existente.	3
	Posibilidad de integración	El programa se puede integrar al flujo de trabajo actual implementando algunas soluciones provisionales.	2

Tabla 3.51: Virtaal – portabilidad.

#### • *Usabilidad*

La tabla 3.52 presenta los resultados correspondientes a la usabilidad de la herramienta.

<b>Usabilidad</b>			
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Descripción</b>	<b>Puntuación</b>
Interfaz de usuario	Distribución de la interfaz de usuario	La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.	3
	Disponibilidad en el idioma requerido	La localización es parcial (interfaz en el idioma requerido pero documentación sin traducir).	2
	Comprensibilidad de los iconos y nombres de funciones	Es fácil relacionar los iconos y nombres de funciones y menús con las acciones que realizan.	2
	Configurabilidad de la apariencia	Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.	2
Facilidad de uso	Posibilidad de navegación y operación solo con teclado	La interfaz está diseñada para permitir la navegación y operación solo con el teclado sin depender del ratón.	3
	Existencia de ayuda contextual	El programa no cuenta con ayuda accesible dentro de la interfaz; es necesario abrir un manual de usuario en otra aplicación o recurrir a otras fuentes (ej. página web o wiki).	1
	Indicadores de progreso y mensajes de error	Se incluyen barras de progreso, mensajes de error u otros mensajes informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.	3
Documentación	Existencia de varios tipos de documentación	El programa incluye varios tipos de documentación (manual de usuario, trucos y sugerencias).	3
	Actualización de la documentación	Toda la documentación está actualizada con la versión actual del programa.	3
	Exhaustividad de la documentación	La documentación no incluye información sobre algunas funciones importantes.	2
	Existencia de recursos multimedia	La documentación incluye algunas capturas de pantalla u otros gráficos.	2

Tabla 3.52: *Virtaal – usabilidad.*

- ***Funcionalidad***

La tabla 3.53 presenta los resultados correspondientes a la funcionalidad de Virtaal.

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
Ajuste a los propósitos	Correspondencia entre las funciones incluidas y las funciones esperadas según el tipo de	Opciones del proyecto: Coincidencias parciales Inserción automática de coincidencias exactas Inserción automática de coincidencias parciales

<b>Funcionalidad</b>		
<b>Subcaracterística</b>	<b>Atributo</b>	<b>Características presentes</b>
	programa	Visualización de metadatos de las coincidencias (fecha, ID de usuario, proyecto, etc.) Propagación automática de segmentos repetidos Coincidencias de glosario Múltiples glosarios por proyecto
		Opciones del editor: Validación de segmentos mediante diferentes estatus Opción de navegación en el editor mediante filtros Estadísticas del proyecto (segmentos traducidos/sin traducir) Búsqueda y reemplazo global Búsqueda de concordancias en archivos originales Autocompletar al escribir Corrector ortográfico al escribir Controles de calidad al escribir
		Integración con aplicaciones externas: Integración con motores de traducción automática locales o vía web Búsqueda en recursos externos (locales o vía servicios web)
	Filtros de archivo implementados	Formatos de localización de software: MO, PO, TS (Qt Linguist), QPH (Qt Phrase Book)
Configurabilidad	Posibilidad de configurar el sistema según diferentes necesidades	Diccionarios del corrector ortográfico personalizables Búsquedas y reemplazos basados en expresiones regulares Colocables configurables Controles de calidad configurables
Interoperabilidad	Soporte para estándares de intercambio de datos	Codificación Unicode Memorias TMX Bases de datos TBX Glosarios como texto delimitado (CSV, TAB o TXT) Archivos pretraducidos XLIFF
	Soporte para formatos abiertos generados por otras herramientas de traducción	TXT (WordFast)

Tabla 3.53: *Virtaal – funcionalidad.*

• **Valoración general de la herramienta**

Según los resultados obtenidos, la portabilidad y la usabilidad de la herramienta se consideran satisfactorias (tres estrellas). En la tabla 3.54 se presenta la consolidación de los datos para estas dos características de calidad.

Portabilidad		
★	★	★
Adaptabilidad	Facilidad de instalación	Coexistencia

Usabilidad		
★	★	★
Interfaz de usuario	Documentación	Facilidad de uso

*Tabla 3.54: Valoración general de la herramienta Virtaal.*

### 3.4.3.2.8 Comparativa de funcionalidad

En los apartados anteriores hemos presentado los resultados obtenidos para cada uno de los proyectos y herramientas evaluados de forma individual. Para complementar estos resultados individuales, en las tablas 3.55 y 3.56 se presenta la comparación de las siete herramientas evaluadas en términos de las funciones incluidas y los formatos soportados, respectivamente.

<b>Funcionalidad</b>	<b>Anaphraseus</b>	<b>Autshumato ITE</b>	<b>OmegaT</b>	<b>OmegaT+</b>	<b>Open Language Tools</b>	<b>OpenTM2</b>	<b>Virtaal</b>
Análisis de originales (coincidencias, repeticiones)		X	X	X		X	
Procesamiento por lotes		X	X				
Pretraducción de documentos	X	X	X		X	X	
Pretraducción con priorización de fuentes utilizadas		X	X				
Pseudotraducción			X				
Creación de proyectos con múltiples documentos de partida		X	X	X		X	
Coincidencias parciales	X	X	X	X	X	X	X
Coincidencias basadas en contexto			X				
Propagación automática de segmentos repetidos		X	X	X	X		X
Coincidencias de glosario	X	X	X	X		X	X
Múltiples glosarios por proyecto		X	X	X		X	X
Inserción automática de coincidencias exactas	X	X	X	X	X	X	X
Inserción automática de coincidencias parciales	X	X	X	X	X	X	X
Posibilidad de utilizar las memorias en ambas direcciones		X	X	X			
Múltiples memorias por proyecto		X	X	X		X	
Múltiples traducciones para el mismo segmento original		X	X		X		
Memorias multilingües (más de dos idiomas)		X	X	X			
Uso simultáneo de glosarios/memorias compartidas por red						X	
Visualización de metadatos de las coincidencias		X	X		X	X	X
Validación de segmentos mediante diferentes estatus					X		X
Opción de navegación en el editor mediante filtros		X	X		X		X
Posibilidad de añadir comentarios a los segmentos		X	X		X	X	
Estadísticas del proyecto (segmentos traducidos/sin traducir)		X	X	X	X	X	X
Búsqueda y reemplazo global	X				X	X	X
Búsqueda de concordancias en archivos originales	X	X	X	X	X	X	X
Búsqueda de concordancias en archivos de referencia		X	X	X		X	

<b>Funcionalidad</b>	<b>Anaphraseus</b>	<b>Autshumato ITE</b>	<b>OmegaT</b>	<b>OmegaT+</b>	<b>Open Language Tools</b>	<b>OpenTM2</b>	<b>Virtual</b>
Autocompletar al escribir	X						X
Corrector ortográfico al escribir	X	X	X			X	X
Corrector ortográfico bajo demanda	X				X	X	
Corrector gramatical / de estilo al escribir	X	X	X				
Corrector gramatical / de estilo bajo demanda	X						
Vista previa del formato	X	X				X	
Modo de revisión (control de cambios, comentarios, tabla)							
Controles de calidad al escribir							X
Controles de calidad bajo demanda		X	X	X	X		
Integración con motores de trad. automática locales o vía web	X	X	X	X			X
Búsqueda en recursos externos (locales o vía servicios web)	X						X
Filtros configurables		X	X	X			
Reglas de segmentación configurables		X	X	X			
Posibilidad de cambiar la segmentación durante la traducción	X	X		X		X	
Porcentaje mínimo de coincidencias configurable	X	X	X	X		X	
Diccionarios del corrector ortográfico personalizables	X	X	X		X	X	X
Reglas del corrector lingüístico personalizables		X	X				
Búsquedas y reemplazos basados en expresiones regulares	X	Solo búsquedas	Solo búsquedas	Solo búsquedas			X
Colocables y localizables configurables (fechas, variables, etc.)							X
Controles de calidad configurables					X		X
Control de acceso al sistema mediante usuarios y permisos							
Atajos de teclado configurables		X	X		X	X	

Tabla 3.55: Comparativa de funciones incluidas en las herramientas evaluadas.

Formatos soportados	Anaphraseus	Autshumato ITE	OmegaT	OmegaT+	Open Language Tools	OpenTM2	Virtaal
Formatos de texto y ofimática	TXT, DOC, DOT, RTF, DOCX, DOTX, ODT, SXW	TXT, CSV, TAB, DOCX, DOTX, XLSX, PPTX, ODT, ODS, ODP, SXW, SRT, DocBook, LaTeX	TXT, CSV, TAB, DOCX, DOTX, XLSX, XLTX, XLSM, PPTX, PPSX, POTX, ODT, ODS, ODP, SXW, SXC, SXI, SRT, DocBook, LaTeX	TXT, CSV, TAB, DOCX, XLSX, PPTX, ODT, ODS, ODP, SXW, SXC, SXI, SRT, DocBook, LaTeX	TXT, ODT, ODS, ODP, SXW, SXC, SXI, DocBook	TXT, DOC, PPT, RTF, WordPerfect, Ami Pro, BookMaster	
Formatos de maquetación	-	XML (Infix), XTG (QuarkXPress), TAG (QuarkXPress)	XML (Infix), IDML (InDesign), XTG (QuarkXPress), TAG (QuarkXPress)	XTG (QuarkXPress), TAG (QuarkXPress)	-	-	
Formatos multimedia	-	SVG, XML (Flash export)	SVG, XML (Flash export), CAMPROJ (Camstasia Studio)	-	-	-	
Formatos de localización web	-	HTML, XML, RESX	HTML, XML, RESX, JSON	HTML, XML, RESX	HTML, XML, JSP	HTML, SGML	
Formatos de localización de software	-	RC, PO, Java Resource Bundles, TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)	RC, POT, PO, Java Resource Bundles, XML (Android resource), TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)	RC, POT, PO, Java Resource Bundles, XML (Android resource), TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)	PO, Java Resource Bundles, DTD (Mozilla)	-	MO, PO, TS (Qt Linguist), QPH (Qt Phrase Book)
Formatos basados en estándares de intercambio de datos	Codificación Unicode, TMX, glosarios como texto delimitado (CSV, TAB o TXT)	Codificación Unicode, TMX, TBX, glosarios como texto delimitado (CSV, TAB o TXT)	Codificación Unicode, TMX, TBX, glosarios como texto delimitado (CSV, TAB o TXT), XLIFF	Codificación Unicode, TMX, glosarios como texto delimitado (CSV, TSV), XLIFF	Codificación Unicode, TMX, XLIFF	Codificación Unicode	Codificación Unicode, TMX, TBX, glosarios como texto delimitado (CSV, TAB o TXT), XLIFF
Formatos de otras herramientas de traducción	TXT (WordFast)	TXML (WordFast Pro)	TXML (WordFast Pro)	-	-	-	TXT (WordFast)

Tabla 3.56: Comparativa de formatos soportados por las herramientas evaluadas.



#### 3.4.4 Discusión

Como vimos en el capítulo 2, existen numerosos enfoques posibles a la hora de evaluar software, así como diversas maneras de descomponer la calidad como concepto multidimensional. Para la evaluación planteada en este trabajo se partió de una descomposición en dos grandes componentes: calidad del proyecto y calidad del producto. De este modo, logramos abarcar las prácticas y procesos involucrados en los proyectos de desarrollo, así como los recursos y servicios disponibles para la comunidad de usuarios, por un lado, y las características técnicas de las herramientas, por el otro.

En total, se consideraron 55 atributos para evaluar la calidad de los proyectos y 23 atributos para evaluar las herramientas. Al incluir las características particulares del paradigma de desarrollo y distribución del software libre, los traductores interesados en estas alternativas, como beneficiarios principales de los resultados, las podrán tener en cuenta en el momento de seleccionar una herramienta para su entorno de trabajo.

En cuanto a los once proyectos evaluados, se encontraron cuatro proyectos cuyas comunidades y madurez se calificaron como inaceptables (EsperantiloTM, ForeignDesk, Frankenstein Translation Memory Suite y Transolution), razón por la cual no se llevó a cabo la evaluación de dichas herramientas. En el caso de los siete proyectos restantes, solo una de las comunidades se calificó como inaceptable (AutshumatoITE); las comunidades de OmegaT+, Open Language Tools y Virtaal, por otro lado, se consideran pobres, mientras que las comunidades de Anaphraseus y OpenTM2 se consideran aceptables; finalmente, solo la comunidad de OmegaT se calificó como satisfactoria. En cuanto a la madurez de los siete proyectos activos, en dos casos se considera pobre (OmegaT+ y Open Language Tools), en dos casos se considera aceptable (Anaphraseus y AutshumatoITE) y en los tres casos restantes se considera satisfactoria (OmegaT, OpenTM2 y Virtaal).

En cuanto a las siete herramientas evaluadas, solo en dos casos se calificó la portabilidad como aceptable (Anaphraseus y Open Language Tools), mientras que en los cinco casos restantes la calificación fue satisfactoria (AutshumatoITE, OmegaT, OmegaT+, OpenTM2 y Virtaal). Respecto a la usabilidad de las herramientas, solo en un caso se considera aceptable (Open Language Tools), mientras que en los seis casos restantes la usabilidad se considera satisfactoria (Anaphraseus, AutshumatoITE, OmegaT, OmegaT+, OpenTM2 y Virtaal).

El método de evaluación presentado en este trabajo es adaptable en cuanto se puede ampliar fácilmente para incluir otras características de calidad o profundizar en la descomposición de las características ya incluidas en el modelo. Para evaluar otras tipologías de programas, por ejemplo, se debe definir una lista de funciones según el tipo de herramienta partiendo de descripciones funcionales y reseñas existentes. Otra adaptación posible es la modificación de los criterios de aceptación establecidos para cada subcaracterística según las necesidades particulares para adecuar la evaluación a contextos específicos.

Al definir los atributos y métricas y preparar el instrumento de evaluación se procuró evitar ambigüedades y se dio preferencia a los atributos binarios de presencia/ausencia por su objetividad inherente; para facilitar la labor del evaluador, en el caso de los atributos cuantitativos con una escala clasificatoria (1, 2 o 3) se especificaron las posibles respuestas con base en las situaciones que se encuentran en casos reales de uso. Para llevar a cabo la evaluación con el menor esfuerzo posible, la disponibilidad de los datos fue uno de los aspectos que se tuvo en cuenta, buscando que la información necesaria se encontrara en las páginas web y documentación de los proyectos sin tener que recurrir a los desarrolladores y usuarios como fuentes de la información. Además, el instrumento de evaluación se implementó en la *wiki* de modo que se pudiera rellenar directamente en línea y las fichas de resultados se obtuvieran automáticamente.

Para lograr que los resultados obtenidos fueran verificables, el instrumento permite recoger directamente el URL de los artefactos disponibles para cada proyecto (listas de correos, sistema de gestión de fallos, perfil de Facebook, etc.) y

de las publicaciones y discusiones encontradas, que a la vez pueden servir de referencia para los usuarios interesados en saber más sobre los proyectos. Además, a modo de ejemplo, se incluyen capturas de pantalla de las páginas web y forjas de donde se obtuvo la información correspondiente a otros atributos.

- ***Problemas encontrados y aspectos a mejorar***

Teniendo en cuenta el enfoque exploratorio del presente trabajo la experiencia de evaluación ha sido positiva en general, aunque durante el proceso también se detectaron algunos posibles problemas y aspectos que se podrían mejorar para lograr una evaluación más rigurosa y detallada.

Al evaluar la estrategia de los proyectos, por ejemplo, dos de los tres atributos definidos para la subcaracterística *estructuras de comunicación y toma de decisiones* resultaron problemáticos. Para los atributos *tipo de proceso para la toma de decisiones* (descentralizado, balanceado o centralizado) y *sistema de gobierno* (dictadura benevolente, meritocracia, democracia o anarquía) solo en un caso se encontró en las páginas de los proyectos la información explícita necesaria para decidir entre las opciones, por lo que en el caso de los proyectos inactivos se optó por dejar estos atributos sin evaluar. En el caso de los proyectos activos, por otro lado, se tomaron como base los atributos *mecanismo de representación y tipo de comunidad de desarrollo* y los conocimientos previos sobre los proyectos para inferir la respuesta adecuada. Resulta evidente, pues, que estos dos atributos son más complejos de lo que se había anticipado, por lo que sería recomendable utilizar otras técnicas para su evaluación, como un análisis detallado de los archivos de las listas de correo o entrevistas a los desarrolladores.

Un aspecto de la estrategia de los proyectos que no se tuvo en cuenta y que podría contribuir a mejorar la comprensión del alcance de los proyectos es el público al que está dirigido. Algunos proyectos, especialmente en el campo del procesamiento del lenguaje natural, están dirigidos a usuarios con conocimientos informáticos avanzados y desarrolladores acostumbrados a trabajar en la línea de comandos, es decir, sin interfaces gráficas. En otros casos, las herramientas están basadas en web y no se ofrecen como servicio, lo que implica que su instalación y

mantenimiento están fuera de las posibilidades de los usuarios con conocimientos técnicos limitados al entorno de escritorio. Así pues, sería útil agregar el atributo *público objetivo* como parte de la subcaracterística *alcance del proyecto*, de modo que con base en estos datos los interesados pudieran filtrar las herramientas según los conocimientos técnicos necesarios para su utilización.

En cuanto a la caracterización de las comunidades, la descomposición de la subcaracterística *sostenibilidad* se podría mejorar. En el método propuesto se utilizaron tres atributos: número de participantes en las listas de usuarios en el último mes, promedio de mensajes por mes en 2011 y tiempo promedio de respuesta para las últimas 5 preguntas realizadas en los foros. No obstante, solo para dos proyectos se encontraron los datos necesarios para evaluar este último atributo. Además, como criterio de aceptación para obtener la estrella correspondiente a la *sostenibilidad* se requería la existencia de discusiones activas en el último mes y un promedio de mensajes por mes en 2011 no inferior a cuatro, sin que este último valor esté fundamentado objetivamente. Así pues, para la *sostenibilidad* se podrían precisar tanto los atributos utilizados como el criterio de aceptación establecido.

Al evaluar la madurez de los proyectos, como parte de la subcaracterística *estado del proyecto* se consideraron dos atributos: la fecha de inicio del proyecto y el estado actual de desarrollo. En ambos casos los datos se obtuvieron a partir de las forjas de desarrollo, pero en algunos casos se encontraron discrepancias entre la autoclasificación de los proyectos y la clasificación de la forja. Además, también es necesario prever que es posible que los proyectos libres cambien de forja de desarrollo, con lo que la fecha inicial de registro del proyecto puede quedar desfasada, por lo que es recomendable confirmar este dato utilizando otras fuentes, como la información proporcionada en las páginas web y blogs de los proyectos o el registro de cambios que en algunos casos se incluye en las descargas.

La evaluación de la reputación de los proyectos es otro de los aspectos en los que se podría profundizar más, para lo cual se podría recurrir a técnicas cualitativas como el análisis de los contenidos publicados en foros de traductores y redes

sociales o encuestas a los usuarios para determinar su grado de satisfacción con las herramientas.

En cuanto a la portabilidad de las herramientas, para calcular el tiempo necesario para la instalación, correspondiente a la subcaracterística *facilidad de instalación*, se podría mejorar el instrumento precisando que se trata de la instalación básica de la herramienta, sin incluir las dependencias ni plugins o extensiones. Además, para evaluar la posibilidad de integración de las herramientas en el flujo de trabajo existente, atributo correspondiente a la subcaracterística *coexistencia*, el tipo de prueba utilizado (inspección de características) podría no ser suficiente y sería recomendable profundizar la evaluación de este aspecto mediante pruebas de escenario en el entorno previsto de uso.

Según se pudo comprobar, la usabilidad de las herramientas es tal vez la característica que supone un mayor riesgo de subjetividad. Aspectos de la interfaz de usuario como lo intuitivo de su distribución o la comprensibilidad de iconos y funciones dependen en gran medida del punto de vista del evaluador y posiblemente también de su grado de familiaridad con el tipo de herramientas evaluadas. Para la subcaracterística *facilidad de uso*, por otro lado, aunque los atributos valorados son de carácter más objetivo (*posibilidad de navegación y operación solo con el teclado, existencia de ayuda contextual y existencia de indicadores de progreso y mensajes de error*) se podrían lograr resultados más rigurosos utilizando pruebas sistemáticas orientadas a los menús, diseñadas para examinar secuencialmente todas las funciones ofrecidas por un programa.

Finalmente, durante la evaluación de la característica de *funcionalidad* no se encontraron datos en la documentación de los programas sobre la posibilidad de integración con software de reconocimiento de voz. En consecuencia, este aspecto de las herramientas no se pudo evaluar y habría de comprobarse mediante pruebas de escenario.



### **3.5 Resumen y conclusiones**

Con este trabajo se esperaba contribuir a la difusión de los programas libres que pueden representar una alternativa para el entorno de trabajo del traductor, partiendo de la base de que una de las principales barreras para la utilización de software libre es el desconocimiento de este tipo de alternativas.

La primera parte del trabajo, correspondiente al marco teórico, se divide en dos capítulos. En el capítulo 1 se abordaron las generalidades de las tecnologías aplicadas a la traducción. En un primer subapartado (1.1) se aclararon los conceptos básicos sobre las licencias de software y se exploraron las oportunidades y riesgos asociados específicamente al uso de software libre. En el subapartado 1.2 se presentaron diversas clasificaciones de las tecnologías de la traducción desde diferentes enfoques, mientras que en el subapartado 1.3 se presentaron los diferentes escenarios de uso previstos para las tecnologías de la traducción y las principales características de los sistemas de memorias de traducción disponibles en la actualidad. En el subapartado 1.4 se presentaron algunos recursos sobre las tecnologías libres para la traducción surgidos desde las dimensiones profesional y académica y, para terminar, en el subapartado 1.5 se presentó una breve recapitulación de los temas abordados en el capítulo.

En el segundo capítulo de la primera parte se presentó una reseña de la literatura sobre calidad y evaluación de software. En el primer subapartado (2.1) se presentaron las generalidades para pasar luego a la evaluación de las tecnologías de la traducción (2.2) y finalmente a la evaluación de software libre (2.3); en el subapartado 2.4 se retomaron y discutieron los aspectos más relevantes para definir la metodología para la evaluación que se plantea en este trabajo.

En la segunda parte del trabajo, correspondiente al trabajo empírico, se describieron los objetivos (apartado 3.1) y las actividades llevadas a cabo, así como los resultados obtenidos. En la primera fase, presentada en el apartado 3.2, se recopiló un catálogo de programas cuyas licencias permiten su libre uso, copia, modificación y distribución. Los programas se agruparon en las categorías de

herramientas de edición y publicación, herramientas lingüísticas, herramientas de traducción y herramientas de gestión. A la hora de seleccionar los programas para el catálogo se tuvo en cuenta el estado de desarrollo de los mismos, excluyendo los programas que aún estuvieran en etapa de diseño o en versión prealfa o alfa. Se considera que el catálogo de programas elaborado, con un total de 188 programas a la fecha (noviembre de 2012), puede servir de referencia tanto para traductores independientes y equipos de traductores como para investigadores, profesores y estudiantes de traducción que estén interesados en conocer alternativas a los programas privativos comerciales.

En la segunda fase del trabajo (apartado 3.3), se exploró el tema de los criterios y metodologías de evaluación de software aplicables para programas libres y se definió un instrumento de evaluación para facilitar a los interesados la selección de programas del catálogo con miras a su implementación en el entorno de trabajo del traductor. No obstante, esta propuesta deberá ser validada mediante otras pruebas antes de que se puedan sacar conclusiones generalizables, especialmente si se tienen en cuenta la abundancia de enfoques posibles y los múltiples aspectos que se pueden considerar a la hora de evaluar software.

En la tercera fase, presentada en el apartado 3.4, para llevar a cabo la evaluación se estableció la muestra de proyectos a evaluar, se recolectaron los datos y se configuró un repositorio público que permite almacenar la información recogida y posteriormente agregar información adicional sobre los proyectos y las herramientas. Además, se diseñó la ficha para presentar los resultados de la evaluación en la *wiki* y se crearon las consultas necesarias para recuperar los datos de los programas mediante filtros. Tanto los resultados de la evaluación como la interfaz de selección de herramientas están disponibles en línea en la web gestionada por la autora de esta investigación<sup>37</sup>.

Para la descripción de los proyectos de desarrollo se tuvieron en cuenta cuatro características: la estrategia del proyecto, la comunidad, la madurez y la reputación. Para la evaluación del software como producto se seleccionaron tres características a considerar: portabilidad, usabilidad y funcionalidad. Los atributos definidos para

<sup>37</sup> <http://traduccionmundolibre.com/wiki>



las dos primeras de estas tres características son directamente aplicables a cualquier tipo de programa, mientras que para evaluar la última, es decir, la funcionalidad, se parte de una lista de funciones que varía según el tipo de programa.

Teniendo en cuenta que en el presente trabajo se quería que el instrumento fuera aplicable a las diferentes categorías de programas para el traductor con mínimas adaptaciones, para la evaluación de los atributos funcionales se consideró solo la presencia o ausencia de las características. Con base en las funciones incluidas en las herramientas privativas más comunes se definió una lista de funciones para la tipología de *sistemas de memorias de traducción*. Siguiendo la misma línea, esta lista se puede ampliar fácilmente para abarcar otras tipologías de programas.

Después de probar el instrumento de evaluación con una muestra de proyectos libres, en nuestra opinión los resultados obtenidos permiten a los posibles usuarios hacer inferencias sobre los proyectos evaluados, compararlos y seleccionar la herramienta más apropiada según sus necesidades, que en definitiva era lo que se esperaba lograr. Además, en términos generales, se considera que los resultados obtenidos reflejan las características de los proyectos evaluados y permiten a los traductores interesados familiarizarse con los aspectos característicos del software libre que deberían tener en cuenta al seleccionar una herramienta para su entorno de trabajo. Con todo, es necesario tener en cuenta que, dadas las características de la evaluación llevada a cabo, la validez de los resultados es solo interna en cuanto no se buscaba hacer generalizaciones a partir de los datos obtenidos.

El panorama general presentado en el marco teórico complementado con el catálogo de programas y la evaluación de una pequeña muestra de proyectos y herramientas contribuye a ampliar la discusión sobre el software libre aplicado a la traducción y a generar más interés hacia las posibilidades que brindan este tipo de herramientas.

- ***Futuros trabajos***

Dados los continuos avances y el surgimiento de nuevos proyectos, la catalogación y evaluación de las tecnologías libres que se propone en esta tesis es un trabajo que no debe terminar aquí. En este sentido, la *wiki* creada podría llegar a convertirse en un espacio de encuentro e intercambio de conocimientos para los traductores interesados en estas alternativas. Eventualmente, si se genera el suficiente interés, podría entonces hablarse de una especie de observatorio permanente de las tecnologías libres para la traducción. En este contexto, sería pertinente seguir con la evaluación de un mayor número de proyectos y herramientas libres y probar la interfaz de selección con una muestra de usuarios representativa.

Sería también interesante en el futuro probar el instrumento de evaluación con distintos evaluadores para comprobar si se obtienen resultados similares y poder sacar conclusiones sobre la fiabilidad del método. De igual manera, para sacar conclusiones definitivas sobre la simplicidad y eficiencia de aplicación de la evaluación también sería conveniente contar con la opinión de distintos evaluadores. Otro aspecto que sería interesante abordar en más detalle es la evaluación directa de la satisfacción de los usuarios de las tecnologías libres para la traducción. Para esto se podría instalar en la *wiki* una extensión complementaria que permitiera a los visitantes asignar puntuaciones a los programas sobre aspectos generales como la facilidad de instalación, facilidad de uso, documentación, asistencia técnica y funcionalidad. Estos datos subjetivos permitirían contrastar los resultados obtenidos en la presente evaluación.

Después de considerar las diferentes estrategias de negocio basadas en software libre y el estado actual de las tecnologías libres para la traducción, creemos que existe un segmento de mercado interesante para una empresa dedicada a ofrecer a pequeñas empresas de traducción y traductores independientes diversos servicios basados en software libre: asesoría y selección de herramientas según las necesidades particulares, asistencia técnica, formación y documentación especializada. En la situación actual de crisis económica, una iniciativa de este tipo presenta varias ventajas, entre las que destacan el autoempleo, la posibilidad de teletrabajo y la baja inversión económica inicial necesaria.

## BIBLIOGRAFÍA CITADA

- Alcina, A. (2008). Translation Technologies: Scope, tools and Resources. *Target: International Journal on Translation Studies*, 20(1), 79-102.
- Atos Origin. (2006). Method for Qualification and Selection of Open Source Software (QSOS) version 1.6. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.qsos.org/download/qsos-1.6-en.pdf>
- Austermühl, F. (2001). *Electronic Tools for Translators*. Manchester: St. Jerome.
- Azzano, D. (2011). *Placeable and localizable elements in translation memory systems: A comparative study* (PhD). University of Munich, Munich.
- Badia, T., Freigang, K.-H., Haller, J., Horschmann, D. H., Maia, B., Reuther, U., & Schmidt, P. (1999). LETRAC Curriculum Modules. Deliverable D3. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.iai-sb.de/docs/D3.pdf>
- Biau-Gil, J. R., & Pym, A. (2006). Technology and translation (a pedagogical overview). En *Translation Technology and its Teaching* (pp. 5-19). Tarragona: Intercultural Studies Group, Universitat Rovira i Virgili.
- Blancafort, H., Heid, U., Gornostay, T., Méchoulam, C., Daille, B., & Sharoff, S. (2011). User-centred Views on Terminology Extraction Tools □ Usage Scenarios and Integration into MT and CAT Tools. En *TRALOGY 2011*. TRALOGY Conference «Translation Careers and Technologies: Convergence Points for the Future». Recuperado por última vez el 15 de marzo de 2013 a partir de [http://www.ttc-project.eu/images/stories/TTC\\_Tralogy\\_2011.pdf](http://www.ttc-project.eu/images/stories/TTC_Tralogy_2011.pdf)
- Bodeux, E., & McKay, C. (2010). Free and Open Source Software for Translators. *Speaking of Translation*. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://speakingoftranslation.com/listen/>
- Bowker, L. (2002). *Computer-Aided Translation Technology: A Practical Introduction*. Ottawa: University of Ottawa Press.
- Bowker, L., & Barlow, M. (2004). Bilingual concordancers and translation memories: A comparative evaluation. En *Proceedings of the 20th International Conference of Computational Linguistics COLING-2004* (pp. 70-83). Presentado en Second International Workshop on Language Resources for Translation Work, Research & Training, Ginebra, Suiza.
- Bowker, L., McBride, C., & Marshman, E. (2008). Getting More than You Paid for? Considerations in Integrating Free and Low-Cost Technologies into Translator Training Programs. *Revista Electrónica de Didáctica de la Traducción y la*

*Interpretación*, (1), 26-47.

BRR. (2005). BRR Whitepaper 2005 RFC 1.

Calzolari, N., McNaught, J., Palmer, M., & Zampolli, A. (2003). ISLE Final Report. ISLE Deliverable D14.2. ISLE. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://www.ilc.cnr.it/EAGLES96/isle/ISLE\\_D14.2.zip](http://www.ilc.cnr.it/EAGLES96/isle/ISLE_D14.2.zip)

Cánovas, M., & Samson, R. (2008a). Dos ejemplos de aplicación del software libre en la docencia de la traducción. En O. Díaz Fouces & M. García González (eds.), *Traducir (con) software libre* (pp. 193-210). Granada: Comares.

Cánovas, M., & Samson, R. (2008b). Herramientas libres para la traducción en entorno MS Windows. En O. Díaz Fouces & M. García González (eds.), *Traducir (con) software libre* (pp. 33-55). Granada: Comares.

Cataldi, Z. (2000). *Metodología de diseño, desarrollo y evaluación de software educativo* (Magíster en Informática). Universidad Nacional de La Plata. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.fi.uba.ar/laboratorios/lisi/cataldi-tesisdemagistereninformatica.pdf>

Cerezo, L. (2003). Hacia la evaluación de dos sistemas comerciales de memorias de traducción. En *Entornos informáticos de la traducción profesional: las memorias de traducción* (pp. 193-213). Granada: Editorial Atrio.

CESLCAM. (2011). Metodología análisis de confianza para proyectos de software libre. Centro de Excelencia de Software Libre de Castilla-la Mancha. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://www.ceslcam.com/fileadmin/user\\_upload/PDFs/metodologia-analisis-confianza-proyectos.pdf](http://www.ceslcam.com/fileadmin/user_upload/PDFs/metodologia-analisis-confianza-proyectos.pdf)

CESLCAM. (s. f.-a). Sello de Confianza OPSOA: Información para usuarios.

CESLCAM. (s. f.-b). Programa de Acreditación de Confianza para Aplicaciones de Software Libre. *Centro de Excelencia de Software Libre de Castilla-La Mancha*. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.ceslcam.com/acreditate/aplicaciones/>

Ciolkowski, M., & Soto, M. (2008). Towards a Comprehensive Approach for Assessing Open Source Projects. En R. Dumke, R. Braungarten, G. Büren, A. Abran, & J. Cuadrado-Gallego (eds.), *Software Process and Product Measurement* (Vol. 5338, pp. 316-330). Springer Berlin / Heidelberg. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://dx.doi.org/10.1007/978-3-540-89403-2\\_26](http://dx.doi.org/10.1007/978-3-540-89403-2_26)

Crowston, K., Howison, J., & Annabi, H. (2003). Defining Open Source Software Project Success. En *ICIS* (pp. 327-340).

Crowston, K., Howison, J., & Annabi, H. (2006). Information Systems Success in Free and Open Source Software Development: Theory and Measures. *SOFTWARE PROCESS: IMPROVEMENT AND PRACTICE*, 123-148.

- Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2011). Free/Libre Open Source Software Development: What we know and what we do not know. *ACM Computing Surveys*.
- Daelemans, W., & Hoste, V. (Eds.). (2010). *Evaluation of Translation Technology*. ASP - Academic & Scientific Publishers.
- Daffara, C. (2009). *The Small/Medium Enterprise guide to Open Source Software*. Conecta. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://guide.conecta.it/FLOSSguide.pdf>
- DeLone, W.H., & McLean, E. R. (2003). The DeLone and McLean model of information systems success: a ten-year update. *Journal of Management Information Systems*, 19(4), 9-30.
- DeLone, William H, & McLean, E. R. (2002). Information Systems Success Revisited. En *Proceedings of the 35th Hawaii International Conference on System Sciences*. Presentado en 35th Hawaii International Conference on System Sciences, Hawaii.
- Deprez, J.-C. (2009). QualOSS Assessment Methodology v. 1.1.
- Deprez, J.-C., & Alexandre, S. (2008). Comparing Assessment Methodologies for Free/Open Source Software: OpenBRR and QSOS. En A. Jedlitschka & O. Salo (eds.), *Product-Focused Software Process Improvement* (Vol. 5089, pp. 189-203). Springer Berlin / Heidelberg. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://dx.doi.org/10.1007/978-3-540-69566-0\\_17](http://dx.doi.org/10.1007/978-3-540-69566-0_17)
- Díaz Fouces, O. (2005a). Software libre y software propietario: algunas preguntas y algunas respuestas. En D. Reineke (ed.), *Traducción y localización: mercado, gestión y tecnologías* (pp. 317-346). Las Palmas de Gran Canaria: Anroart Ediciones.
- Díaz Fouces, O. (2005b). Anar a Babel sense passar per Redmond. Migració a PL en la formació universitària de traductors. Presentado en IV Jornades de Programari Lliure, Universitat Politècnica de Catalunya. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.jornadespl.org/biblioteca/iv-jornades/ponencies/odiaz.pdf>
- Díaz Fouces, O. (2006). Software Libre en la formación de traductores: entre el pragmatismo y la utopía 1 (Vol. 2). Presentado en Congreso AIETI.
- Díaz Fouces, O. (2007). Creación de un entorno de trabajo para la formación de profesionales de los servicios lingüísticos con GNU/Linux. *Hermeneus*, (9), 21-38.
- Díaz Fouces, O. (2008). Ferramentas lliures para traduir com GNU/Linux e Mac OS X. En O. Díaz Fouces & M. García González (eds.), *Traducir (con) software libre* (pp. 57-73). Granada: Comares.
- Díaz Fouces, O. (2011). El programari lliure com a objectiu i com a instrument per a la traducció. *Tradumàtica: tecnologies de la traducció*, (9), 1-4.

- Díaz Fouces, O., & García González, M. (Eds.). (2008). *Traducir (con) software libre*. Granada: Comares.
- Dillon, T. S., & Simmons, G. L. (2010). Towards an Ontology for Open Source Software Development. *International Federation for Information Processing Digital Library*, 203(1). Recuperado por última vez el 15 de marzo de 2013 a partir de <http://cs.anu.edu.au/iojs/index.php/ifip/article/view/12255>
- Dumbill, E. (s. f.). Description of a Project (DOAP) vocabulary. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://usefulinc.com/ns/doap#>
- EAGLES. (1996). Evaluation of Translators' Aids. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.issco.unige.ch/en/research/projects/ewg95//node140.html>
- Eisele, A., Federmann, C., & Hodson, J. (2009). Towards an effective toolkit for translators. En *Proceedings of the ASLIB International Conference Translating and the Computer 31*. London: ASLIB. Recuperado a partir de [http://www.dfki.de/lt/publication\\_show.php?id=4586](http://www.dfki.de/lt/publication_show.php?id=4586)
- European Union. (s. f.). Asset Description Metadata Schema for Software. Recuperado por última vez el 15 de marzo de 2013 a partir de [https://joinup.ec.europa.eu/asset/adms\\_foss/home](https://joinup.ec.europa.eu/asset/adms_foss/home)
- Fernández, J. R. (2003). La Traducción en el mundo del Software Libre: Análisis del estado de las herramientas lingüísticas, proyectos actuales y necesidades de la comunidad del software libre. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://es.tldp.org/Articulos/0000otras/doc-traducion-libre/>
- Filatova, I. (2010). *Evaluación de herramientas y recursos informáticos (TAO y ofimática) para la traducción profesional: hacia la configuración de un entorno óptimo de trabajo para el traductor autónomo* (Tesis doctoral). Universidad de Málaga.
- Flórez, S., & Alcina, A. (2011a). Free/open source software for translators: a catalogue of available tools. *The Interpreter and Translator Trainer*, 5(2), 325-57.
- Flórez, S., & Alcina, A. (2011b). Catálogo de software libre para la traducción. *Tradumàtica: tecnologies de la traducció*, (9), 57-73.
- Forcada, M. (s. f.). Free/open-source machine translation systems and tools. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://computing.dcu.ie/~mforcada/fosmt.html>
- Free Software Foundation. (s. f.-a). Licenses. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.fsf.org/licensing/licenses/>
- Free Software Foundation. (s. f.-b). Categorías de software libre y software que no es libre. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.gnu.org/philosophy/categories.es.html>
- Fuggetta, A. (2003). Open Source Software -an evaluation. *The Journal of Systems and*

*Software*, 66, 77-90.

- Fulford, H., & Granell Zafra, J. (2005). Translation and technology: a study of UK freelance translators. *JoSTrans - The Journal of Specialised Translation*, (04), 2-17.
- García González, M. (2008). Free software for translators: is the market ready for a change? En *Traducir (con) software libre* (pp. 9-31). Granada: Comares.
- Garvin, D. (1984). What Does «Product Quality» Really Mean? *Sloan Management Review*, 26, 25-45.
- Gasser, L., Scacchi, W., Ripoche, G., & Penne, B. (2003). Understanding Continuous Design in F/OSS Projects. Presentado en 16th. Intern. Conf. Software & Systems Engineering and their Applications, Paris. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.ics.uci.edu/%7Ewscacchi/Papers/New/ICSSEA03.pdf>
- Gow, F. (2003). *Metrics for Evaluating Translation Memory Software* (MA Thesis). University of Ottawa. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://isg.urv.es/library/papers/gow.doc>
- Groenewald, H., & Fourier, W. (2009). Introducing the Autshumato Integrated Translation Environment. En *Proceedings of the 13th Annual Conference of the EAMT* (pp. 190-196). Presentado en 13th Annual Conference of the EAMT, Barcelona. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.mt-archive.info/EAMT-2009-Groenewald.pdf>
- Groven, A.-K., Haaland, K., Glott, R., Tannenber, A., & Darbousset-Chong, X. (2011). Quality Assessment of FOSS. En *INF5780 H2011: Open Source, Open Collaboration and Innovation* (pp. 73-91). Recuperado por última vez el 15 de marzo de 2013 a partir de <http://publications.nr.no/Compendium-INF5780H11.pdf>
- Guillardeau, S. (2009). *Freie Translation Memory Systeme für die Übersetzungspraxis*. Universität Wien. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://othes.univie.ac.at/6863/>
- Helander, N., Aaltonen, T., Mikkonen, T., Oksanen, V., Puhakka, M., Seppänen, M., ... Vainio, N. (2007). *Open Source Software Management Framework*. Tampere: Tampere University of Technology. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://www.academia.edu/323439/OPEN\\_SOURCE\\_SOFTWARE\\_MANAGEMENT\\_FRAMEWORK](http://www.academia.edu/323439/OPEN_SOURCE_SOFTWARE_MANAGEMENT_FRAMEWORK)
- Höge, M. (2002). *Towards a Framework for the Evaluation of Translators' Aids Systems* (Phd Thesis). University of Helsinki, Finland. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.3520&rep=rep1&type=pdf>
- Hutchins, J. (2008). Compendium of Translation Software. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.hutchinsweb.me.uk/Compendium-14.pdf>

- Hutchins, J., & Somers, H. (1992). *An Introduction to Machine Translation*. London: Academic Press. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.hutchinsweb.me.uk/IntroMT-TOC.htm>
- ISO/IEC 9126. Software engineering. Product quality. (2001).
- Izquierdo-Cortazar, D., González-Barahona, J., Robles, G., Deprez, J., & Auvray, V. (2010). Floss communities: Analyzing evolvability and robustness from an industrial perspective. *Open Source Software New Horizons*, 336–341.
- Izquierdo Cortázar, D., Robles, G., González Barahona, J., & Deprez, J.-C. (2009). Assessing FLOSS Communities: An Experience Report from the QualOSS Project. En *Open Source Ecosystems Diverse Communities Interacting* (Vol. 299, p. 364). Springer Boston. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://dx.doi.org/10.1007/978-3-642-02032-2\\_38](http://dx.doi.org/10.1007/978-3-642-02032-2_38)
- James, J. (2003). Free software and the digital divide: opportunities and constraints for developing countries. *Journal of Information Science*, 29(1), 25-33.
- Kan, S. H. (2002). *Metrics and Models in Software Quality Engineering* (2nd ed.). Reading, Mass.: Addison-Wesley.
- King, M. (1998). Evaluation Design: The EAGLES framework. En *Evaluation of the Linguistic Performance of Machine Translation Systems*. Presentado en Konvens'98, Bonn: Verlag, St. Augustin. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.issco.unige.ch/pub/king-konvens.ps.gz>
- Koehn, P., & Hoang, H. (2010). Machine Translation with Open Source Software. University of Edinburgh.
- Kooths, S., Langenfurth, M., & Kalwey, N. (2003). Open Source-Software: An Economic Assessment. *MICE Economic Research Studies*, 4. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://flosshub.org/179>
- Lagoudaki, E. (2006). Translation Memory systems: Enlightening users' perspective. Key finding of the TM Survey 2006 carried out during July and August 2006. Imperial College London.
- Lagoudaki, E. (2007). Translators evaluate TM systems - a survey. *Multilingual*, (March), 57-59.
- Lagoudaki, E. (2008). *Expanding the Possibilities of Translation Memory Systems: From the Translator's Wishlist to the Developers Design* (PhD Thesis). Imperial College London.
- Lagoudaki, E. (2009). Translation Editing Environments. Presentado en MT Summit XII, Ottawa, Ontario, Canada. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.mt-archive.info/MTS-2009-Lagoudaki.pdf>
- Lee, S.-Y. T., Kim, H.-W., & Gupta, S. (2009). Measuring open source software success. *Omega*, 37(2), 426 - 438. doi:10.1016/j.omega.2007.05.005



- March, S. (2000). Reflections on Computer Science and Information Systems Research. En A. Laender, S. Liddle, & V. Storey (eds.), *Conceptual Modeling — ER 2000* (Vol. 1920, pp. 16-26). Presentado en 19th International Conference on Conceptual Modeling, Salt Lake City, Utah, USA: Springer Berlin / Heidelberg. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://link.springer.com/chapter/10.1007%2F3-540-45393-8\\_2](http://link.springer.com/chapter/10.1007%2F3-540-45393-8_2)
- Mas, J. (2003). El software libre y las lenguas minoritarias: una oportunidad impagable. *Digithum*, (5). Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.uoc.edu/humfil/articles/esp/mas0303/mas0303.html>
- Maślanko. (2004). *A Comparative Study of Terminology Management Tools in Machine-Assisted Human Translation*. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://www.transsoft.seo.pl/en/translator\\_tools.html](http://www.transsoft.seo.pl/en/translator_tools.html)
- May, C. (2006). Escaping the TRIPs' trap: The political economy of free and open source software in Africa. *Political Studies*, 54(1), 123-146.
- McKay, C. (2005). Software libre y software de código abierto para traductores. (P. Barruetaña, trad.). AbroadLink. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.abroadlink.com/es/Articulos/Software%20de%20codigo%20abierto%20y%20software%20libre%20para%20traductores.pdf>
- Melby, A. (1996). Machine Translation and Other Translation Technologies. *Annual Review of Applied Linguistics*, 16, 86-98.
- Melby, A. (1998). Eight Types of Translation Technology. Presentado en American Translators Association ATA 39th Annual Conference, Hilton Head Island, South Carolina. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.ttt.org/technology/8types.pdf>
- Morisio, M., Stamelos, I., & Tsoukias, A. (2003). Software Product and Process Assessment Through Profile-Based Evaluation. *International Journal of Software Engineering & Knowledge Engineering*, 13(5), 495 - 512.
- Navarro, E., Montero, F., & JCCM. (2009). OPSOA: certificación de software libre. Junta de Comunidades de Castilla-la Mancha (JCCM).
- Neunzig, W. (2001). *La intervención pedagógica en la enseñanza de la traducción on-line – cuestiones de método y estudio empírico* (Tesis doctoral). Universitat Autònoma de Barcelona.
- O'Hagan, M., & Ashworth, D. (2002). *Translation-mediated communication in a digital world facing the challenges of globalization and localization*. Clevedon: Multilingual Matters.
- Oliver, A., Moré, J., & Climent, S. (2008). *Traducción y tecnologías*. Barcelona: Editorial UOC.
- Oliver, A., Vázquez, M., & Moré, J. (2007). Linguoc Lexterm: una herramienta gratuita de extracción automática de terminología. *Translation Journal*, 11(4). Recuperado por

última vez el 15 de marzo de 2013 a partir de <http://www.translationdirectory.com/articles/article1606.php>

Perea Sardón, J. I. (2010). *Revisión asistida por ordenador de traducciones. Aplicación práctica a la revisión del sistema operativo libre Ubuntu como ejemplo* (Phd). Universidad de Granada, Granada. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://opentranslation.es/recursos/PereaSardon2010.pdf>

Petrinja, E., Silliti, A., Nambakam, R., Wittmann, M., Oltolina, S., Ruffati, G., ... Ortega, F. (2008). WD 6.2.1 Trustworthy elements identified in OS processes.

Prior, M. (2003). Close Windows. Open Doors. *Translation Journal*, 7(1). Recuperado por última vez el 15 de marzo de 2013 a partir de <http://translationjournal.net/journal/23linux.htm>

Prior, M. (s. f.). Linux for translators. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.linuxfortranslators.org>

Quah, C. K. (2006). *Translation and Technology*. New York: Palgrave Macmillan Ltd.

Ramírez-Sánchez, G., Sánchez-Martínez, F., Ortiz-Rojas, S., Pérez-Ortiz, J. A., & Forcada, M. L. (2006). Opendrad Apertium open-source machine translation system: an opportunity for business and research. En *Proceeding of Translating and the Computer 28 Conference*. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.dlsi.ua.es/~mlf/docum/ramirezsanchez06p.pdf>

Rico, C. (2001). Reproducible models for CAT tools evaluation: A user-oriented perspective. En *Translating and the Computer 23*. Presentado en Aslib, Londres. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.mt-archive.info/Aslib-2001-Rico.pdf>

Rosa, F. da, & Heinz, F. (2007). Guía práctica sobre software libre: su selección y aplicación local en América Latina y el Caribe. UNESCO. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://unesdoc.unesco.org/images/0015/001560/156096s.pdf>

Samoladas, I., Gousios, G., Spinellis, D., & Stamelos, I. (2008). The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation. En E. Damiani & G. Succi (eds.), *Open Source Development, Communities and Quality — OSS 2008: 4th International Conference on Open Source Systems* (pp. 237–248). Boston: Springer. doi:10.1007/978-0-387-09684-1\_19

Sánchez-Martínez, F., & Forcada, M. L. (2011). Free/open-source machine translation: preface. *Machine Translation*, 25(2), 83-86. doi:10.1007/s10590-011-9113-x

Sandrini, P. (2012a). Todo sobre TuxTrans. *Revista CTPCBA, Ene/Feb*(111), 13-14.

Sandrini, P. (2012b). Tecnologia FLOSS per la traduzione. Disponibilità, applicazione e problematiche. *InTRAlinea online translation journal, Special Issue: Specialised Translation II*. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.intralea.org/specials/article/1796>

- Sandrini, P. (s. f.). Free Software for Translators. Recuperado enero 16, 2011, a partir de <http://homepage.uibk.ac.at/~c61302/en-fsfrans.html>
- Scacchi, W. (2009). Understanding Requirements for Open Source Software. En W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, K. Lyytinen, ... B. Robinson (eds.), *Design Requirements Engineering: A Ten-Year Perspective* (Vol. 14, pp. 467-494). Springer Berlin Heidelberg. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://dx.doi.org/10.1007/978-3-540-92966-6\\_27](http://dx.doi.org/10.1007/978-3-540-92966-6_27)
- Schulmeyer, G. G. (Ed.). (2006). *Handbook of Software Quality Assurance* (4.<sup>a</sup> ed.). Boston, London: Artech House.
- Schweik, C. M., & English, R. (2007). Identifying Success and Abandonment of Free/Libre and Open Source (FLOSS) Commons: A Preliminary Classification of Sourceforge.net projects. ScholarWorks@UMass Amherst. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://scholarworks.umass.edu/opensource/2>
- Skok, M. (2011). Future of Open Source 2011 Survey. Presentado en Open Source Business Conference, San Francisco. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.futureopensource.net/2011-survey-results>
- Sokoli, R. (2002). Catálogo de herramientas para la localización de software y páginas web. *Tradumática, 1*, 13 páginas.
- Somers, H. (2003). Translation memory systems. En *Computers and Translation: A translator's guide* (pp. 31-47). John Benjamins Publishing.
- Soto, M., Ciolkowski, M., Deprez, J.-C., Ruiz, J., Herraiz, I., García Campos, C., & Matulevičius, R. (2009). Metrics and Indicators of the Standard QualOSS Assessment Method, Deliverable 4.2 of the QualOSS project. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://www.academia.edu/817534/QualOSS\\_Methodology\\_and\\_QualOSS\\_Assessment\\_Method](http://www.academia.edu/817534/QualOSS_Methodology_and_QualOSS_Assessment_Method)
- Spinellis, D., Gousios, G., Karakoidas, V., Louridas, P., Adams, P. J., Samoladas, I., & Stamelos, I. (2009). Evaluating the Quality of Open Source Software. *Electronic Notes in Theoretical Computer Science*, 233, 5-28. doi:10.1016/j.entcs.2009.02.058
- Squire, M., & Williams, D. (2012). Describing the software forge ecosystem. Presentado en 45th Hawaii International Conference on System Sciences, Maui, Hawaii.
- Stol, K.-J., & Ali Babar, M. (2010). A Comparison Framework for Open Source Software Evaluation Methods. En P. Ågerfalk, C. Boldyreff, J. M. González-Barahona, G. R. Madey, & J. Noll (eds.), *Open Source Software: New Horizons* (Vol. 319, pp. 389-394). Berlin, Heidelberg: Springer Berlin Heidelberg. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.springerlink.com/content/59485r7074x40277/>
- Taravella, A. (2011). *Preliminary Summary Report on the Results of the Survey Conducted among Users of Language Technologies in April-May 2011*. Canadá: Language Technologies Research Centre. Recuperado por última vez el 15 de marzo de 2013

a partir de <http://www.crtl.ca/display266>

Tecnolettra. (2008). Centro Virtual del Observatorio de Tecnologías de la Traducción. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://tecnolettra.uji.es/ott/index.html>

TEMAA. (s. f.). TEMAA Final Report. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://cst.dk/temaa/D16/d16exp-Contents.html>

UNE-ISO/IEC 14598. Information Technology. Software Product Evaluation. (1998).

Van den Berg, K. (2005). *Finding Open options: An Open Source software evaluation model with a case study on Course Management Systems* (Master Thesis). Tilburg University. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.karinvandenberg.nl/Thesis.pdf>

Wasserman, A., Murugan, P., & Chan, C. (2006). *The Business Readiness Rating Model: an Evaluation Framework for Open Source*.

Wheeler, D. A. (2008). How to Evaluate Open Source Software / Free Software (OSS/FS) Programs. Recuperado por última vez el 15 de marzo de 2013 a partir de [http://www.dwheeler.com/oss\\_fs\\_eval.html](http://www.dwheeler.com/oss_fs_eval.html)

Wiechmann, D., & Fuhs, S. (2006). Corpus linguistics resources: Concordancing software.

Wiggins, A., Howison, J., & Crowston, K. (2009). *Heartbeat: Measuring Active User Base and Potential User Interest in FLOSS Projects*. (C. Boldyreff, K. Crowston, B. Lundell, & T. Wasserman, eds.) (Vol. 299). Skövde, Sweden, 3-6 June: Springer Boston.

Wittmann, M., & Nambakam, R. (2010). OMM: CMM-like model for OSS. Qualipso Project.

Zerfaß, A. (2002). Evaluating Translation Memory Systems. *Language Resources for Translation Work and Research*, 49.

Zetzsche, J. (2008). Taking Inventory. *The Tool Kit. 123rd edition*.

Zetzsche, J. (2011). *Individual translators and data exchange standards*. A TAUS & International Writers Group. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.translationautomation.com/component/rsform/form/7-individual-translators-and-data-exchange-standards>

Zetzsche, J. (s. f.). The Tool Box - A monthly newsletter for people in the translation industry who want to get more out of their computers. Recuperado por última vez el 15 de marzo de 2013 a partir de <http://www.internationalwriters.com/toolkit/>

# CIBERGRAFÍA

- ***Software libre en general***

Free Software Directory: <http://directory.fsf.org/>

Free Software Foundation: <http://fsf.org>

Linux Foundation: <http://www.linuxfoundation.org>

Ohloh: <http://ohloh.net>

Open Source Initiative: <http://opensource.org>

SourceForge: <http://sourceforge.net>

- ***Evaluación de software libre***

Business Readiness Rating: <http://openbrr.org>

How to Evaluate Open Source Software / Free Software (OSS/FS) Programs:  
[http://www.dwheeler.com/oss\\_fs\\_eval.html](http://www.dwheeler.com/oss_fs_eval.html)

Managing Open Source Software as an Integrated Part of Business: <http://coss.fi/ossi/>

Programa de Acreditación de Confianza para Aplicaciones de Software Libre:  
<http://www.ceslcam.com/acreditate/aplicaciones/>

Method for Qualification and Selection of Open Source software (QSOS):  
<http://www.qsos.org/>

Quality Platform for Open Source Software: <http://www.qualipso.org>

- ***Software libre para traductores***

FOSS4Trans Wiki: A Catalogue of Free/Open-Source Software for Translators:  
<http://traduccionmundolibre.com/wiki>

Free and Open Source Software for Translators:  
<http://www.translatewrite.com/foss/index.php?s=foss&p=main>

Free / Open Source link list: [http://www.martinwunderlich.com/links\\_en.php4](http://www.martinwunderlich.com/links_en.php4)

Free/open-source machine translation systems and tools:  
<http://computing.dcu.ie/~mforcada/fosmt.html>

Free Software for Translators: <http://homepage.uibk.ac.at/~c61302/en-fsftans.html>

Free Software for Translators: <http://tonybaldwin.me/hax/doku.php?id=freetran>

Grupo de Estudos das TecnoloXías Libres da Tradución: <http://webs.uvigo.es/getlt/>

Linux for translators: <http://www.linuxfortranslators.org>

Linux for Translators User Group: <http://tech.groups.yahoo.com/group/linuxfortranslators/>

Tuxtrans User Group: <http://groups.google.de/group/tuxtrans>

# ANEXO 1: LISTADO DE PROGRAMAS INCLUIDOS EN EL CATÁLOGO DE SOFTWARE LIBRE PARA TRADUCTORES

## Herramientas de edición y maquetación

Programa	Tipología	Página web
AbiWord	Procesador de palabras	<a href="http://www.abisource.com">http://www.abisource.com</a>
Aegisub	Editor de subtítulos	<a href="http://www.aegisub.org/">http://www.aegisub.org/</a>
Aloha Editor	Editor HTML	<a href="http://aloha-editor.com/index.html">http://aloha-editor.com/index.html</a>
BlueGriffon	Editor HTML	<a href="http://bluegriffon.org/">http://bluegriffon.org/</a>
Bluefish	Editor de texto	<a href="http://bluefish.openoffice.nl/">http://bluefish.openoffice.nl/</a>
CMSwithTMS	Sistema de gestión de contenidos multilingües	<a href="http://cmswithtms.net/">http://cmswithtms.net/</a>
Codepage Converter	Conversión de codificación de caracteres	<a href="http://toffeeapps.co.uk/codepage-converter/">http://toffeeapps.co.uk/codepage-converter/</a>
DiffPDF	Herramienta de diferenciación	<a href="http://www.qtrac.eu/diffpdf.html">http://www.qtrac.eu/diffpdf.html</a>
Diffuse	Herramienta de diferenciación	<a href="http://diffuse.sourceforge.net/">http://diffuse.sourceforge.net/</a>
Extpell	Corrector ortográfico	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
GIMP	Editor avanzado de imágenes	<a href="http://www.gimp.org/">http://www.gimp.org/</a>
GImageReader	Reconocimiento óptico de caracteres	<a href="http://sourceforge.net/projects/gimagereader/">http://sourceforge.net/projects/gimagereader/</a>
GNU Emacs	Editor de texto	<a href="http://www.gnu.org/software/emacs/">http://www.gnu.org/software/emacs/</a>
GOOCR	Reconocimiento óptico de caracteres	<a href="http://jocr.sourceforge.net/">http://jocr.sourceforge.net/</a>
Gaupol Subtitle Editor	Editor de subtítulos	<a href="http://home.gna.org/gaupol/">http://home.gna.org/gaupol/</a>
Gnome Subtitles	Editor de subtítulos	<a href="http://gnome-subtitles.sourceforge.net/about">http://gnome-subtitles.sourceforge.net/about</a>
Inkscape	Editor de gráficos de vectores escalables	<a href="http://www.inkscape.org/">http://www.inkscape.org/</a>
JEdit	Editor de texto	<a href="http://www.jedit.org/">http://www.jedit.org/</a>
Jubler	Editor de subtítulos	<a href="http://jubler.org/">http://jubler.org/</a>
KOffice	Suite ofimática	<a href="http://www.koffice.org/">http://www.koffice.org/</a>
KompoZer	Editor HTML	<a href="http://kompozer.net/">http://kompozer.net/</a>
LibreOffice	Suite ofimática	<a href="http://www.documentfoundation.org/download/">http://www.documentfoundation.org/download/</a>
LyX	Editor LaTeX	<a href="http://www.lyx.org">http://www.lyx.org</a>
Notepad-plus-plus	Editor de texto	<a href="http://notepad-plus.sourceforge.net/es/site.htm">http://notepad-plus.sourceforge.net/es/site.htm</a>
OCRFeeder	Reconocimiento óptico de caracteres	<a href="http://live.gnome.org/OCRFeeder">http://live.gnome.org/OCRFeeder</a>
Ocrad	Reconocimiento óptico de caracteres	<a href="http://www.gnu.org/software/ocrad/ocrad.html">http://www.gnu.org/software/ocrad/ocrad.html</a>
Ocre	Reconocimiento óptico de caracteres	<a href="http://lem.eui.upm.es/ocre.html">http://lem.eui.upm.es/ocre.html</a>

OpenOffice.org	Suite ofimática	<a href="http://www.openoffice.org/">http://www.openoffice.org/</a>
PDF Import	Editor de archivos PDF	<a href="http://extensions.services.openoffice.org/project/pdfimport">http://extensions.services.openoffice.org/project/pdfimport</a>
PDFTOHTML	Conversión de PDF a HTML	<a href="http://pdfhtml.sourceforge.net/">http://pdfhtml.sourceforge.net/</a>
PDFedit	Manipulación de archivos PDF	<a href="http://pdfedit.petricek.net/en/index.html">http://pdfedit.petricek.net/en/index.html</a>
PS BRAMUS.TextConvert	Conversión de formatos	<a href="http://www.bram.us/2008/10/30/ps_bramustextconvert-psd2txt-and-txt2psd-for-the-masses/">http://www.bram.us/2008/10/30/ps_bramustextconvert-psd2txt-and-txt2psd-for-the-masses/</a>
Pandoc	Conversión de formatos	<a href="http://johnmacfarlane.net/pandoc/index.html">http://johnmacfarlane.net/pandoc/index.html</a>
Pdftk	Manipulación de archivos PDF	<a href="http://www.accesspdf.com/pdftk/">http://www.accesspdf.com/pdftk/</a>
Scribus	Maquetación	<a href="http://www.scribus.net/">http://www.scribus.net/</a>
Serna Free XML Editor	Editor XML	<a href="http://www.syntext.com/products/serna-free/">http://www.syntext.com/products/serna-free/</a>
Subtitle Edit	Editor de subtítulos	<a href="http://www.nikse.dk/SubtitleEdit">http://www.nikse.dk/SubtitleEdit</a>
Subtitle Editor	Editor de subtítulos	<a href="http://home.gna.org/subtitleeditor/">http://home.gna.org/subtitleeditor/</a>
Tickle Text	Editor de texto	<a href="http://baldwinsoftware.com/tcltext.html">http://baldwinsoftware.com/tcltext.html</a>
Transcriber	Herramienta de transcripción	<a href="http://trans.sourceforge.net/en/presentation.php">http://trans.sourceforge.net/en/presentation.php</a>
VietOCR	Reconocimiento óptico de caracteres	<a href="http://vietocr.sourceforge.net/">http://vietocr.sourceforge.net/</a>
VisualSubSync	Editor de subtítulos	<a href="http://www.visualsubsync.org/">http://www.visualsubsync.org/</a>
WinMerge	Herramienta de diferenciación	<a href="http://winmerge.org">http://winmerge.org</a>
XML Copy Editor	Editor XML	<a href="http://xml-copy-editor.sourceforge.net/">http://xml-copy-editor.sourceforge.net/</a>
Xerlin	Editor XML	<a href="http://www.xerlin.org/">http://www.xerlin.org/</a>
Xpdf	Visualización y conversión de PDF	<a href="http://www.foolabs.com/xpdf/home.html">http://www.foolabs.com/xpdf/home.html</a>

## Herramientas lingüísticas

Programa	Tipología	Página web
AConCorde	Herramienta de concordancias	<a href="http://www.andy-roberts.net/software/aConCorde/index.html">http://www.andy-roberts.net/software/aConCorde/index.html</a>
BootCaT	Extracción terminológica	<a href="http://bootcat.sslmit.unibo.it">http://bootcat.sslmit.unibo.it</a>
CorpusSearch	Procesamiento de corpus	<a href="http://corpussearch.sourceforge.net/">http://corpussearch.sourceforge.net/</a>
Corpuscatcher	Procesamiento de corpus	<a href="http://translate.sourceforge.net/wiki/corpuscatcher/index">http://translate.sourceforge.net/wiki/corpuscatcher/index</a>
ExtPhr for Java	Extracción terminológica	<a href="http://publish.uwo.ca/~craven/freeware.htm#java">http://publish.uwo.ca/~craven/freeware.htm#java</a>
GesTerm	Gestión terminológica	<a href="http://www.termcat.cat/ca/GesTerm/">http://www.termcat.cat/ca/GesTerm/</a>
GoldenDict	Búsqueda en recursos lingüísticos	<a href="http://goldendict.org/">http://goldendict.org/</a>
IMS Open Corpus Workbench	Procesamiento de corpus	<a href="http://cwb.sourceforge.net/">http://cwb.sourceforge.net/</a>
JBootCat	Procesamiento de corpus	<a href="http://www.andy-roberts.net/software/jbootcat/index.html">http://www.andy-roberts.net/software/jbootcat/index.html</a>
Kea	Extracción de frases clave	<a href="http://www.nzdl.org/Kea">http://www.nzdl.org/Kea</a>
LanguageTool	Revisión lingüística	<a href="http://www.languagetool.org/">http://www.languagetool.org/</a>
Lucon	Herramienta de concordancias	<a href="http://sourceforge.net/projects/lucon/">http://sourceforge.net/projects/lucon/</a>
Manatee and Bonito	Procesamiento de corpus	<a href="http://www.textforge.cz/products">http://www.textforge.cz/products</a>
Maui	Extracción terminológica	<a href="http://code.google.com/p/maui-indexer/">http://code.google.com/p/maui-indexer/</a>
Natural Language Toolkit	Procesamiento de corpus	<a href="http://www.nltk.org/">http://www.nltk.org/</a>
Ngram Statistics Package	Extracción terminológica	<a href="http://ngram.sourceforge.net/">http://ngram.sourceforge.net/</a>
OnDiLoTo	Búsqueda en recursos	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>



Programa	Tipología	Página web
	lingüísticos	
Poliqarp	Herramienta de concordancias	<a href="http://poliqarp.sourceforge.net/">http://poliqarp.sourceforge.net/</a>
StarDict	Búsqueda en recursos lingüísticos	<a href="http://stardict.sourceforge.net/">http://stardict.sourceforge.net/</a>
TES	Extracción terminológica	<a href="http://lpg.uoc.edu/aoliverg/index.php?page=tes">http://lpg.uoc.edu/aoliverg/index.php?page=tes</a>
TTC TermSuite	Extracción terminológica	<a href="http://www.ttc-project.eu">http://www.ttc-project.eu</a>
TextSTAT	Herramienta de concordancias	<a href="http://neon.niederlandistik.fu-berlin.de/en/textstat/">http://neon.niederlandistik.fu-berlin.de/en/textstat/</a>
TheW for Java	Creación de tesauros	<a href="http://publish.uwo.ca/~craven/freeware.htm#java">http://publish.uwo.ca/~craven/freeware.htm#java</a>
Tickle-Dict	Búsqueda en recursos lingüísticos	<a href="http://www.baldwinsoftware.com/tdict.html">http://www.baldwinsoftware.com/tdict.html</a>
Translate.Net	Búsqueda en recursos lingüísticos	<a href="http://translate-net.appspot.com/">http://translate-net.appspot.com/</a>
Ucto	Toquenizador	<a href="http://ilk.uvt.nl/ucto/">http://ilk.uvt.nl/ucto/</a>
Uplug corpus tools	Procesamiento de corpus	<a href="http://www.let.rug.nl/~tiedeman/Uplug/">http://www.let.rug.nl/~tiedeman/Uplug/</a>
WeBoCa	Procesamiento de corpus	<a href="http://code.google.com/p/weboca/">http://code.google.com/p/weboca/</a>
Xaira	Procesamiento de corpus	<a href="http://www.oucs.ox.ac.uk/rts/xaira/">http://www.oucs.ox.ac.uk/rts/xaira/</a>

## Herramientas de traducción

Programa	Tipología	Página web
Aligner.py	Alineación de textos	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
Anaphraseus	Sistema de memorias de traducción	<a href="http://anaphraseus.sourceforge.net">http://anaphraseus.sourceforge.net</a>
Android Localizer	Localización	<a href="http://www.artfulbits.com/Android/Localizer.aspx">http://www.artfulbits.com/Android/Localizer.aspx</a>
Apertium	Sistema de traducción automática	<a href="http://www.apertium.org/">http://www.apertium.org/</a>
Attesoro	Localización	<a href="http://attesor.org/">http://attesor.org/</a>
Autshumato ITE	Sistema de memorias de traducción	<a href="http://autshumato.sourceforge.net/ITE.html">http://autshumato.sourceforge.net/ITE.html</a>
Bitext2tmx	Alineación de textos	<a href="http://bitext2tmx.sourceforge.net/">http://bitext2tmx.sourceforge.net/</a>
Bitextor	Generador de bitextos	<a href="http://bitextor.sourceforge.net/">http://bitextor.sourceforge.net/</a>
Bligner	Alineación de textos	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
Boltran	Sistema de memorias de traducción basado en web	<a href="http://www.boltran.com">http://www.boltran.com</a>
Cunei Machine Translation Platform	Sistema de traducción automática	<a href="http://www.cunei.org/">http://www.cunei.org/</a>
Do Moses Yourself (DoMY™) Community Edition	Sistema de traducción automática	<a href="http://www.precisiontranslationtools.com/index.php?option=com_content&amp;view=article&amp;id=1&amp;Itemid=22">http://www.precisiontranslationtools.com/index.php?option=com_content&amp;view=article&amp;id=1&amp;Itemid=22</a>
DotNetLocalizer	Localización	<a href="http://sourceforge.net/projects/dotnetlocalizer/">http://sourceforge.net/projects/dotnetlocalizer/</a>
Entrans	Sistema de memorias de traducción basado en web	<a href="http://sourceforge.net/projects/entrans/">http://sourceforge.net/projects/entrans/</a>
EsperantiloTM	Sistema de memorias de traducción	<a href="http://www.esperantilo.org/tm/">http://www.esperantilo.org/tm/</a>
Extract-tmx-corpus	Creación de corpus	<a href="http://code.google.com/p/extract-tmx-corpus/">http://code.google.com/p/extract-tmx-corpus/</a>
ForeignDesk	Sistema de memorias de traducción	<a href="http://sourceforge.net/projects/foreigndesk/">http://sourceforge.net/projects/foreigndesk/</a>
Frankenstein Translation Memory Suite	Sistema de memorias de traducción	<a href="http://sourceforge.net/projects/frankenstein/">http://sourceforge.net/projects/frankenstein/</a>
G11n Toolkit	Localización	<a href="http://g11ntoolkit.sourceforge.net/">http://g11ntoolkit.sourceforge.net/</a>

<b>Programa</b>	<b>Tipología</b>	<b>Página web</b>
Game Translator	Localización	<a href="http://www.watto.org/trans/info.html">http://www.watto.org/trans/info.html</a>
Gtranslator	Localización	<a href="http://gtranslator.sourceforge.net/">http://gtranslator.sourceforge.net/</a>
Image Localization Manager	Localización	<a href="http://sourceforge.net/projects/ilmanager/">http://sourceforge.net/projects/ilmanager/</a>
IniTranslator	Localización	<a href="http://initranslator.sourceforge.net/wiki/index.php/Main_Page">http://initranslator.sourceforge.net/wiki/index.php/Main_Page</a>
Joshua	Sistema de traducción automática	<a href="http://joshua.sourceforge.net/Joshua/Welcome.html">http://joshua.sourceforge.net/Joshua/Welcome.html</a>
KBabel	Localización	<a href="http://kbabel.kde.org/">http://kbabel.kde.org/</a>
LF Aligner	Alineación de textos	<a href="http://sourceforge.net/projects/aligner/">http://sourceforge.net/projects/aligner/</a>
Lokalize	Localización	<a href="http://userbase.kde.org/Lokalize">http://userbase.kde.org/Lokalize</a>
Marclator	Sistema de traducción automática	<a href="http://www.computing.dcu.ie/~mforcada/marclator.html">http://www.computing.dcu.ie/~mforcada/marclator.html</a>
Moses	Sistema de traducción automática	<a href="http://www.statmt.org/moses">http://www.statmt.org/moses</a>
Moses for Localization	Sistema de traducción automática	<a href="http://code.google.com/p/m4loc/">http://code.google.com/p/m4loc/</a>
Moses for Mere Mortals	Sistema de traducción automática	<a href="http://code.google.com/p/moses-for-mere-mortals/">http://code.google.com/p/moses-for-mere-mortals/</a>
MozillaTranslator	Localización	<a href="http://sourceforge.net/projects/moztrans/">http://sourceforge.net/projects/moztrans/</a>
Narro	Sistema de memorias de traducción basado en web	<a href="http://code.google.com/p/narro">http://code.google.com/p/narro</a>
NiuTrans	Sistema de traducción automática	<a href="http://www.nlplab.com/NiuPlan/NiuTrans.html">http://www.nlplab.com/NiuPlan/NiuTrans.html</a>
OTranCe	Sistema de memorias de traducción basado en web	<a href="http://otrance.de/en/">http://otrance.de/en/</a>
Olifant	Editor de memorias de traducción	<a href="http://okapi.sourceforge.net/">http://okapi.sourceforge.net/</a>
OmegaT	Sistema de memorias de traducción	<a href="http://www.omegat.org/">http://www.omegat.org/</a>
OmegaTplus	Sistema de memorias de traducción	<a href="http://omegatplus.sourceforge.net/">http://omegatplus.sourceforge.net/</a>
Open Language Tools	Sistema de memorias de traducción	<a href="http://open-language-tools.java.net/">http://open-language-tools.java.net/</a>
OpenLogos	Sistema de traducción automática	<a href="http://logos-os.dfki.de/release">http://logos-os.dfki.de/release</a>
OpenMaTrEx	Sistema de traducción automática	<a href="http://www.openmatrex.org/">http://www.openmatrex.org/</a>
OpenTM2	Sistema de memorias de traducción	<a href="http://www.opentm2.org">http://www.opentm2.org</a>
OpenTMS	Sistema de memorias de traducción basado en web	<a href="http://www.opentms.de/?q=en">http://www.opentms.de/?q=en</a>
Poedit	Localización	<a href="http://www.poedit.net/">http://www.poedit.net/</a>
Pology	Localización	<a href="http://pology.nedohodnik.net/">http://pology.nedohodnik.net/</a>
Pootle	Sistema de memorias de traducción basado en web	<a href="http://translate.sourceforge.net/wiki/pootle/index">http://translate.sourceforge.net/wiki/pootle/index</a>
Qt Linguist	Localización	<a href="http://bit.ly/yIHUTv">http://bit.ly/yIHUTv</a>
RESX Editor	Localización	<a href="http://resx.sourceforge.net/">http://resx.sourceforge.net/</a>
RESX Translator	Localización	<a href="http://resxtranslatorbing.codeplex.com/">http://resxtranslatorbing.codeplex.com/</a>
Ratel	Editor de reglas de segmentación	<a href="http://okapi.opentag.com/">http://okapi.opentag.com/</a>
ResBundleEdit	Localización	<a href="http://sourceforge.net/projects/resbundleedit/">http://sourceforge.net/projects/resbundleedit/</a>
ResourceBlender	Localización	<a href="http://www.resourceblender.com/">http://www.resourceblender.com/</a>
ResxTranslator	Localización	<a href="http://sourceforge.net/projects/resxtranslator/">http://sourceforge.net/projects/resxtranslator/</a>
Stripper	Extracción de texto	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>

<b>Programa</b>	<b>Tipología</b>	<b>Página web</b>
Symeval	Herramienta de evaluación de traducciones	<a href="http://sourceforge.net/apps/mediawiki/symeval/index.php?title=Main_Page">http://sourceforge.net/apps/mediawiki/symeval/index.php?title=Main_Page</a>
TBXCheck	Validación de formato	<a href="http://sourceforge.net/projects/tbxutil/">http://sourceforge.net/projects/tbxutil/</a>
THOT	Herramienta para entrenar motores de traducción automática	<a href="http://thot.sourceforge.net/">http://thot.sourceforge.net/</a>
TMXCleaner	Limpieza de archivos TMX	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
TMXMerger	Combinación de archivos TMX	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
TMXValidator	Validación de formato	<a href="http://www.maxprograms.com/products/tmxvalidator.html">http://www.maxprograms.com/products/tmxvalidator.html</a>
Tag Aligner	Alineación de textos	<a href="http://tag-aligner.sourceforge.net/">http://tag-aligner.sourceforge.net/</a>
Tikal	Localización	<a href="http://okapi.opentag.com/">http://okapi.opentag.com/</a>
Tmx Localization Editor	Localización	<a href="http://tmx-editor.sourceforge.net/">http://tmx-editor.sourceforge.net/</a>
Tmx2txt	Conversión de formatos	<a href="http://www.omegat.org/omegatk/tmx2txt.zip">http://www.omegat.org/omegatk/tmx2txt.zip</a>
Toxic	Conversión de formatos	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
TraduXio	Sistema de memorias de traducción basado en web	<a href="http://traduxio.hypertopic.org/index/about">http://traduxio.hypertopic.org/index/about</a>
Tradubi	Traducción automática	<a href="http://sourceforge.net/projects/tradubi">http://sourceforge.net/projects/tradubi</a>
TransTools Suite	Conversión de formatos	<a href="http://www.translatortools.net/">http://www.translatortools.net/</a>
Transie	Editor de textos de doble panel	<a href="http://sourceforge.net/apps/mediawiki/transie">http://sourceforge.net/apps/mediawiki/transie</a>
Transifex	Sistema de memorias de traducción basado en web	<a href="http://transifex.org/">http://transifex.org/</a>
Translate Toolkit	Localización	<a href="http://translate.sourceforge.net/wiki/toolkit/index">http://translate.sourceforge.net/wiki/toolkit/index</a>
Transolution	Sistema de memorias de traducción	<a href="http://sourceforge.net/projects/eviltrans/">http://sourceforge.net/projects/eviltrans/</a>
TuMatXa	Repositorio de archivos TMX	<a href="http://www.tumatxa.com/">http://www.tumatxa.com/</a>
UncleanifyTMX	Conversión de formatos	<a href="http://leuce.com/tempfile/omtautoit/">http://leuce.com/tempfile/omtautoit/</a>
Validator	Validación de formato	<a href="http://omegatplus.sourceforge.net/applications.html">http://omegatplus.sourceforge.net/applications.html</a>
Virtaal	Sistema de memorias de traducción	<a href="http://translate.sourceforge.net/wiki/virtaal/index">http://translate.sourceforge.net/wiki/virtaal/index</a>
WordFast tmx file converter	Conversión de formatos	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
Wordforge	Localización	<a href="http://sourceforge.net/projects/wordforge2/">http://sourceforge.net/projects/wordforge2/</a>
Worldwide Lexicon	Traducción colaborativa	<a href="http://www.worldwidexicon.org">http://www.worldwidexicon.org</a>
XLIFF Tools Java	Conversión de formatos	<a href="http://xliff-tools.freedesktop.org/wiki/Projects/XliffToolsJava">http://xliff-tools.freedesktop.org/wiki/Projects/XliffToolsJava</a>
XLIFF Translator	Conversión de formatos	<a href="http://felix-cat.com/tools/xliff-translator/">http://felix-cat.com/tools/xliff-translator/</a>
XLIFFChecker	Validación de formato	<a href="http://www.maxprograms.com/products/xliffchecker.html">http://www.maxprograms.com/products/xliffchecker.html</a>
XliffRoundTrip Tool	Conversión de formatos	<a href="http://sourceforge.net/projects/xliffroundtrip/">http://sourceforge.net/projects/xliffroundtrip/</a>
Zaval Java Resource Editor	Localización	<a href="http://www.zaval.org/products/jrc-editor/">http://www.zaval.org/products/jrc-editor/</a>

## Herramientas de gestión

<b>Programa</b>	<b>Tipología</b>	<b>Página web</b>
Achievo	Gestión de proyectos	<a href="http://www.achievo.org/">http://www.achievo.org/</a>
Bibus	Gestión de referencias bibliográficas	<a href="http://bibus-biblio.sourceforge.net/">http://bibus-biblio.sourceforge.net/</a>
CheckMate	Quality assurance	<a href="http://okapi.opentag.com/">http://okapi.opentag.com/</a>
Dotproject	Gestión de proyectos	<a href="http://www.dotproject.net/">http://www.dotproject.net/</a>

<b>Programa</b>	<b>Tipología</b>	<b>Página web</b>
Feng Office	Groupware	<a href="http://www.fengoffice.com/web/community/community_index.php">http://www.fengoffice.com/web/community/community_index.php</a>
GMX-V	Word counting	<a href="http://code.google.com/p/gmx-v/">http://code.google.com/p/gmx-v/</a>
GanttProject	Financial management	<a href="http://ganttproject.biz/">http://ganttproject.biz/</a>
GlobalSight	Translation Management System	<a href="http://www.globalsight.com">http://www.globalsight.com</a>
GnuCash	Financial management	<a href="http://www.gnucash.org/">http://www.gnucash.org/</a>
InfOdasis	Enterprise Resource Planning	<a href="http://www.infodasis.com">http://www.infodasis.com</a>
JabRef	Gestión de referencias bibliográficas	<a href="http://jabref.sourceforge.net/">http://jabref.sourceforge.net/</a>
LinkChecker	Quality assurance	<a href="http://linkchecker.sourceforge.net/">http://linkchecker.sourceforge.net/</a>
Open Workbench	Gestión de proyectos	<a href="http://www.openworkbench.org/">http://www.openworkbench.org/</a>
OpenProj	Gestión de proyectos	<a href="http://www.serena.com/products/openproj/">http://www.serena.com/products/openproj/</a>
ProcessMaker	Gestión de proyectos	<a href="http://www.processmaker.com">http://www.processmaker.com</a>
Project open	Gestión de proyectos	<a href="http://www.project-open.org/">http://www.project-open.org/</a>
Project.net	Gestión de proyectos	<a href="http://www.project.net/index.htm">http://www.project.net/index.htm</a>
Rainbow	Gestión de proyectos	<a href="http://okapi.opentag.com/">http://okapi.opentag.com/</a>
SdSIMPLE	Facturación	<a href="http://www.sdsimple.es/web/index.php/about-joomla">http://www.sdsimple.es/web/index.php/about-joomla</a>
Simple Invoices	Facturación	<a href="http://www.simpleinvoices.org">http://www.simpleinvoices.org</a>
Siwapp	Facturación	<a href="http://www.siwapp.org/">http://www.siwapp.org/</a>
T7e	Sistema de gestión de traducciones	<a href="https://sourceforge.net/projects/t7e/">https://sourceforge.net/projects/t7e/</a>
TKount	Recuento de palabras	<a href="http://www.omegat.org/en/resources.html">http://www.omegat.org/en/resources.html</a>
TaskJuggler	Gestión de proyectos	<a href="http://www.taskjuggler.org">http://www.taskjuggler.org</a>
TeamLab	Trabajo colaborativo	<a href="http://teamlab.com/">http://teamlab.com/</a>
Tiki Wiki CMS Groupware	Trabajo colaborativo	<a href="http://tiki.org/">http://tiki.org/</a>
TransProCalc	Gestión financiera	<a href="http://code.google.com/p/transprocalc">http://code.google.com/p/transprocalc</a>
Web2Project	Gestión de proyectos	<a href="http://web2project.net/">http://web2project.net/</a>
Zotero	Gestión de referencias bibliográficas	<a href="http://www.zotero.org">http://www.zotero.org</a>

## **ANEXO 2: INSTRUMENTO DE EVALUACIÓN**

### **El proyecto de desarrollo**

1. Estrategia del proyecto
  - 1.1. Origen del proyecto
    - a) Proyecto independiente
    - b) Proyecto con financiación pública
    - c) Proyecto con financiación privada
    - d) Proyecto con financiación mixta
  - 1.2. Tipo de ética que rige el proyecto
    - a) Ética hacker
    - b) ética híbrida
    - c) ética empresarial
  - 1.3. Estrategia de licenciamiento del proyecto
    - a) Una licencia libre
    - b) Varias licencias libres
    - c) Licenciamiento dual (libre y privativo)
    - d) Núcleo abierto
  - 1.4. Permisividad de la licencia
    - a) Sin *copyleft*

b) Con *copyleft* débil

c) Con *copyleft* fuerte

1.5. ¿Existen guías y acuerdos explícitos para quienes quieran contribuir?

a) Sí (URL): \_\_\_\_\_

b) No

1.6. Titularidad de los derechos de autor

a) El titular es un único desarrollador

b) Titularidad asignada a una entidad legal

c) Titularidad distribuida

1.7. ¿Existe una especificación de requerimientos predefinida?

a) Sí (URL): \_\_\_\_\_

b) No

1.8. ¿Existe una hoja de ruta?

a) Sí (URL): \_\_\_\_\_

b) No

1.9. ¿Existe una descripción de nuevas funciones planeadas?

a) Sí (URL): \_\_\_\_\_

b) No

1.10. ¿Existe un plan de versiones?

a) Sí (URL): \_\_\_\_\_

b) No

1.11. ¿Cómo describiría el proceso de toma de decisiones en el proyecto?

- a) Descentralizado
- b) Balanceado
- c) Centralizado

1.12. ¿Cuál de las siguientes opciones describe mejor el sistema de gobierno del proyecto?

- a) Dictadura benevolente
- b) Meritocracia
- c) Democracia
- d) Anarquía

1.13. ¿Cuál es el mecanismo de representación utilizado por el proyecto para comunicarse y ser identificado?

- a) Líderes reconocidos
- b) Fundación
- c) Comité directivo
- d) Empresa o institución patrocinadora

1.14. ¿Reutiliza el proyecto código de otros proyectos libres?

- a) Sí (¿de cuáles?) \_\_\_\_\_
- b) No

1.15. ¿Se trata de un proyecto derivado de otro proyecto libre?

- a) Sí (¿de cuál?) \_\_\_\_\_
- b) No

1.16. ¿Se desarrollan otras herramientas como parte del mismo proyecto?

a) Sí (¿cuáles?) \_\_\_\_\_

b) No

## 2. Comunidad

2.1. Tipo de comunidad de desarrollo

a) Desarrollador independiente

b) Grupo de desarrolladores

c) Desarrolladores organizados formalmente

d) Entidad legal

e) Entidad comercial

2.2. ¿Existe algún proyecto bifurcado (fork) o herramienta derivada?

a) Sí (¿cuál?): \_\_\_\_\_

b) No

2.3. ¿Cuenta el proyecto con participación institucional?

a) Sí (¿qué instituciones?) \_\_\_\_\_

b) No

2.4. ¿Cuál es el número de desarrolladores activos? \_\_\_\_\_

2.5. ¿Cuál es el número de suscriptores en la lista de usuarios? \_\_\_\_\_

2.6. ¿Cuál es el número de usuarios que participaron en las discusiones durante el último mes? \_\_\_\_\_

2.7. ¿Cuál es el promedio de mensajes por mes en el foro de usuarios en



2011? \_\_\_\_\_

2.8. ¿Cuál ha sido el tiempo promedio de respuesta en los foros para las últimas cinco preguntas? \_\_\_\_\_

2.9. ¿Existe un portal web que recoja la información destacada del proyecto?

a) Sí (URL) \_\_\_\_\_

b) No

2.10. Indique los URL de los espacios de encuentro disponibles para los usuarios:

a) Grupo de usuarios \_\_\_\_\_

b) Foro \_\_\_\_\_

c) Wiki \_\_\_\_\_

d) Blog \_\_\_\_\_

e) Canal de chat IRC \_\_\_\_\_

f) Perfil en Facebook \_\_\_\_\_

g) Perfil en Twitter \_\_\_\_\_

2.11. Indique cuáles de los siguientes servicios están disponibles para los usuarios:

a) Asistencia técnica personalizada Sí \_\_\_ No \_\_\_

b) Suscripciones de asistencia técnica/valor añadido Sí \_\_\_ No \_\_\_

c) Formación Sí \_\_\_ No \_\_\_

d) Desarrollo personalizado Sí \_\_\_ No \_\_\_

e) Asesoría/Consultoría Sí \_\_\_ No \_\_\_

f) Software como servicio

Sí \_\_\_ No \_\_\_

### 3. Madurez del proyecto

3.1. Fecha de inicio del proyecto: \_\_\_\_\_

3.2. Estado actual de desarrollo

- a) Beta
- b) Estable
- c) Maduro
- d) Inactivo

3.3. ¿El proyecto está registrado en alguna de las principales forjas públicas?

- a) Sí (URL) \_\_\_\_\_
- b) No

3.4. ¿Existe un repositorio de código fuente con sistema de control de revisiones?

- a) Sí (URL) \_\_\_\_\_
- b) No

3.5. ¿Se utiliza un sistema de gestión de informes de posibles errores?

- a) Sí (URL) \_\_\_\_\_
- b) No

3.6. ¿Se utiliza un sistema de gestión de solicitudes de nuevas funciones?

- a) Sí (URL) \_\_\_\_\_
- b) No

3.7. ¿Existen procesos documentados para contribuir al proyecto?

a) Sí (URL) \_\_\_\_\_

b) No

3.8. ¿Se utiliza una plataforma específica para la gestión de la localización del programa?

a) Sí (URL) \_\_\_\_\_

b) No

3.9. ¿Existe un proceso documentado de elicitación y gestión de requisitos?

a) Sí (URL) \_\_\_\_\_

b) No

3.10. ¿Existe un ciclo de lanzamientos predefinido?

a) Sí (URL) \_\_\_\_\_

b) No

3.11. ¿Cuántas versiones lanzó el proyecto en 2011? \_\_\_\_\_

3.12. ¿Cuántas actualizaciones lanzó el proyecto en 2011? \_\_\_\_\_

3.13. ¿En qué fecha fue liberada la última versión disponible? \_\_\_\_\_

#### 4. Reputación del proyecto

4.1. ¿Existen publicaciones (libros, reseñas, artículos, entradas en blogs) sobre el proyecto? Sí \_\_\_ No \_\_\_

En caso afirmativo, indique los URL:

---

---

---

---

4.2. ¿Existe alguna implementación de referencia/caso de éxito documentado en la web del proyecto?

a) Sí (URL) \_\_\_\_\_

b) No

4.2.2. ¿Cuál es el promedio de descargas en la semana siguiente al lanzamiento de las tres últimas versiones? \_\_\_\_\_

4.2.3. ¿Cuál es el número de descargas en el último mes? \_\_\_\_\_

4.2.4. ¿Existen discusiones sobre el proyecto en foros para traductores (ProZ, LinkedIn, etc.)? Sí \_\_\_ No \_\_\_

En caso afirmativo, indique los URL:

---

---

---

---

4.3. ¿Se incluye el programa en los repositorios de alguna de las principales distribuciones de GNU/Linux? Sí \_\_\_ No \_\_\_

En caso afirmativo, indique los URL:

---

---

---

---

4.4. ¿Está incluido el proyecto en catálogos o directorios de software?

Sí \_\_\_ No \_\_\_

En caso afirmativo, indique los URL:

---

---

---

---

4.5. ¿Está registrado el proyecto en la plataforma Ohloh.net?

a) Sí (URL) \_\_\_\_\_

b) No

4.5.2. ¿Existen reseñas y puntuaciones en la forja utilizada por el proyecto?

a) Sí (URL) \_\_\_\_\_

b) No

4.5.3. ¿Existen comentarios recientes sobre el proyecto en redes sociales?

a) Sí (URL) \_\_\_\_\_

b) No

## **La herramienta**

5. Portabilidad

5.1. Modularidad

- a) El diseño de la herramienta no permite el desarrollo de componentes separados.
- b) El diseño de la herramienta permite el desarrollo de componentes separados que se pueden integrar al sistema mediante plugins, pero no hay una API pública.
- c) El diseño de la herramienta permite el desarrollo de componentes separados que se pueden integrar al sistema mediante una API plugins.

#### 5.2. Escalabilidad

- a) El sistema no está pensado para implementaciones a gran escala ni incluye modo multiusuario.
- b) El sistema puede ser implementado a gran escala pero no está pensado para entornos multiusuario o viceversa.
- c) El sistema puede ser implementado a gran escala y en entornos multiusuario.

#### 5.3. Tiempo necesario para la instalación (sin incluir la preparación del sistema)

- a) Más de 30 minutos.
- b) Entre 15 y 30 minutos.
- c) Menos de 15 minutos.

#### 5.4. Requisitos de sistema

- a) Es necesario instalar otro(s) programa(s) para poder utilizar la herramienta.
- b) Los programas necesarios se incluyen en el paquete de instalación.
- c) El programa no depende de software de terceros.

### 5.5. Conocimientos técnicos necesarios para la instalación

- a) Es necesario realizar procedimientos avanzados para la instalación o configuración (ej. compilación de fuente, configuración de servidores o bases de datos, etc.).
- b) La instalación del programa es sencilla, pero es necesario realizar otras configuraciones adicionales que requieren algunos conocimientos informáticos.
- c) La instalación es sencilla y no hay que realizar configuraciones adicionales.

### 5.6. Compatibilidad con el entorno existente

- a) El programa no puede ser utilizado en el entorno existente.
- b) El programa puede ser utilizado en el entorno existente realizando algunas configuraciones especiales.
- c) El programa puede ser utilizado en el entorno existente.

### 5.7. Posibilidades de integración

- a) El programa difícilmente se puede integrar al flujo de trabajo actual.
- b) El programa se puede integrar al flujo de trabajo actual implementando algunas soluciones provisionales.
- c) El programa se puede integrar fácilmente al flujo de trabajo actual.

## 6. Usabilidad

### 6.1. Distribución de la interfaz de usuario

- a) La interfaz es compleja con demasiada información sin una organización clara; es necesario utilizar el manual.

- b) Toma un poco de tiempo, la información está más o menos organizada; hay que usar un poco el manual.
- c) La interfaz es simple e intuitiva, la información está bien organizada; prácticamente no se necesita el manual.

#### 6.2. Disponibilidad en el idioma requerido

- a) El programa y su documentación y ayuda solo están disponibles en un idioma distinto al requerido.
- b) La localización es parcial (interfaz en el idioma requerido pero documentación sin traducir o viceversa).
- c) El programa está totalmente localizado al idioma requerido, tanto la interfaz de usuario como las ayudas y otra documentación incluida.

#### 6.3. Comprensibilidad de los iconos y nombres de funciones

- a) Algunos iconos o nombres de funciones y menús son difíciles de relacionar con la acción que realizan.
- b) Es fácil relacionar los iconos y nombres de funciones y menús con las acciones que realizan.
- c) Es fácil relacionar los nombres de funciones y menús con las acciones que realizan y se incluyen textos alternativos (tooltips) para explicar las funciones de los iconos.

#### 6.4. Configurabilidad de la apariencia

- a) No es posible cambiar la configuración de la apariencia del programa.
- b) Es posible realizar algunos cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.), pero no es posible exportar dichos cambios para recuperarlos posteriormente.



- c) Es posible realizar cambios en la configuración de apariencia (ej. tipo y tamaño de fuente, colores, etc.) y dichos cambios se pueden exportar.

#### 6.5. Posibilidad de navegación y operación solo con el teclado

- a) La interacción depende en gran medida del uso del ratón.
- b) El programa permite la navegación y operación con el teclado aunque para algunas acciones es necesario el ratón.
- c) La interfaz está diseñada para permitir la navegación y operación solo con el teclado sin depender del ratón.

#### 6.6. Existencia de ayuda contextual

- a) El programa no cuenta con ayuda accesible dentro de la interfaz; es necesario abrir un manual de usuario en otra aplicación o recurrir a otras fuentes (ej. página web o wiki).
- b) El programa incluye una ayuda básica en la interfaz, pero para funciones avanzadas es necesario salir de la aplicación para recurrir a otras fuentes (ej. manual o comunidad de usuarios).
- c) El programa incluye ayuda sobre la mayoría de las funciones y es posible acceder a ella sin salir de la aplicación.

#### 6.7. Indicadores de progreso y mensajes de error

- a) No se incluyen barras de progreso ni mensajes informativos, por lo que el usuario en ocasiones no sabe si el programa está ocupado realizando algún proceso o si ha ocurrido un fallo.
- b) Se incluyen mensajes informativos o barras de progreso, pero en algunos casos el usuario no sabe si el programa está ocupado o si ha ocurrido un fallo.
- c) Se incluyen barras de progreso, mensajes de error u otros mensajes

informativos de manera que el usuario está siempre informado sobre la actividad del programa o los posibles fallos.

#### 6.8. Existencia de varios tipos de documentación

- a) Solo se dispone de un archivo de texto tipo Léame.txt o no hay documentación disponible.
- b) Se dispone de un archivo Léame y por lo menos otra guía o manual de usuario.
- c) El programa incluye varios tipos de documentación (guía rápida o de instalación, manual de usuario, manual para desarrolladores, preguntas frecuentes, solución de problemas, etc.).

#### 6.9. Actualización de la documentación

- a) Toda la documentación se refiere a versiones anteriores del programa o no hay documentación.
- b) Algunas partes de la documentación no corresponden a la versión actual del programa.
- c) Toda la documentación está actualizada con la versión actual del programa.

#### 6.10. Exhaustividad de la documentación

- a) La documentación es muy limitada o no hay documentación disponible.
- b) La documentación no incluyen información sobre algunas funciones importantes.
- c) La documentación cubre de manera clara todas las funciones del programa.

#### 6.11. Existencia de recursos multimedia

- a) No se incluyen ayudas visuales ni recursos multimedia como parte de la documentación o no hay documentación.
- b) La documentación incluye algunas capturas de pantalla u otros gráficos.
- c) La documentación incluye capturas de pantalla y se dispone de videotutoriales o demos en línea.

## 7. Funcionalidad

### 7.1. Funciones que ofrece la herramienta.

<b>Característica</b>	<b>Disponible</b>	<b>No disponible</b>
Análisis de originales (coincidencias, repeticiones)		
Procesamiento por lotes		
Pretraducción de documentos		
Pretraducción con priorización de fuentes utilizadas		
Pseudotraducción		
Creación de proyectos con múltiples documentos de partida		
Coincidencias parciales		
Coincidencias basadas en contexto		
Autocompletar al escribir		
Inserción automática de coincidencias exactas		
Inserción automática de coincidencias parciales		
Visualización de metadatos de las coincidencias		
Validación de segmentos mediante diferentes estatus		
Opción de navegación en el editor mediante filtros		
Posibilidad de añadir comentarios a los segmentos		
Propagación automática de segmentos repetidos		
Estadísticas del proyecto (segmentos traducidos/sin traducir)		
Coincidencias de glosario		
Múltiples glosarios por proyecto		
Búsqueda y reemplazo global		
Búsqueda de concordancias en archivos originales		
Búsqueda de concordancias en archivos de referencia		
Corrector ortográfico al escribir		
Corrector ortográfico bajo demanda		
Corrector gramatical / de estilo al escribir		

Corrector gramatical / de estilo bajo demanda		
Vista previa del formato		
Posibilidad de utilizar las memorias en ambas direcciones		
Múltiples memorias por proyecto		
Múltiples traducciones para el mismo segmento original		
Memorias multilingües (más de dos idiomas)		
Uso simultáneo de glosarios/memorias compartidas por red		
Integración con motores de trad. automática locales o vía web		
Búsqueda en recursos externos (locales o vía servicios web)		
Integración con software de reconocimiento de voz		
Modo de revisión (control de cambios, comentarios, exportación a tabla)		
Controles de calidad al escribir		
Controles de calidad bajo demanda		
Filtros configurables		
Reglas de segmentación configurables		
Posibilidad de cambiar la segmentación durante la traducción		
Porcentaje mínimo de coincidencias configurable		
Diccionarios del corrector ortográfico personalizables		
Reglas del corrector lingüístico personalizables		
Búsquedas y reemplazos basados en expresiones regulares		
Colocables y localizables configurables (fechas, variables, etc.)		
Controles de calidad configurables (etiquetas, puntuación, espacios, números, términos, etc.)		
Control de acceso al sistema mediante usuarios y permisos		
Atajos de teclado configurables		

## 7.2. Filtros de archivo implementados

<b>Tipo de formato</b>	<b>Formatos soportados</b>
Formatos de texto y ofimática	
Formatos de maquetación	
Formatos multimedia	
Formatos de localización web	
Formatos de localización de software	

## 7.3. Interoperabilidad

<b>Formato</b>	<b>Soportado</b>	<b>No soportado</b>
Codificación Unicode		
Reglas de segmentación SRX		
Memorias TMX		

Bases de datos TBX		
Glosarios como texto delimitado (CSV, TAB o TXT)		
Archivos XLIFF		
Formato TTX (SDL Trados)		
Formato TXT (WordFast)		
Formato TXML (WordFast Pro)		
Formato NXT (STAR Transit)		

## **ANEXO 3: PROTOCOLO DE APLICACIÓN DEL INSTRUMENTO DE EVALUACIÓN**

### **Caracterización del proyecto**

- Visitar la página web del proyecto en busca de información concerniente al origen del proyecto, su filosofía y su alcance.
- También en la página web, buscar información sobre la licencia o licencias utilizadas. Utilizar la tabla de clasificación según permisividad para clasificar la licencia en una de las tres categorías (con *copyleft* fuerte, con *copyleft* débil o sin *copyleft*).
- Buscar la hoja de ruta o información referente a nuevas funciones que se planea implementar.
- Examinar la página web para determinar si hay alguna institución vinculada al proyecto y si se ofrecen servicios profesionales.
- En la página web del proyecto buscar los espacios de comunicación utilizados por los usuarios para intercambiar información y los procedimientos que se deben seguir para contribuir al proyecto.
- En la forja de desarrollo buscar información sobre el número de desarrolladores activos, la fecha de inicio del proyecto, su estado de desarrollo y las herramientas de gestión utilizadas.
- Visitar el archivo del foro o grupo de usuarios para comprobar el número total de suscriptores y de participantes activos, el promedio de mensajes por mes durante 2011 y el tiempo promedio de respuesta para las últimas cinco preguntas formuladas.

- En la página web del proyecto y en la forja de desarrollo buscar información sobre las versiones y actualizaciones lanzadas durante 2011 y la fecha de liberación de la última versión disponible.
- Realizar búsquedas generales en Internet para determinar las publicaciones disponibles sobre el proyecto, así como su inclusión en directorios de software y la existencia de discusiones sobre la herramienta en foros especializados para traductores.
- En el buscador especializado *Topsy*, revisar si existen comentarios sobre el proyecto en las redes sociales.
- Revisar si el proyecto aparece registrado en la plataforma *Ohloh.net*.
- En la forja de desarrollo buscar los datos sobre el número de descargas y sobre reseñas y puntuaciones de los usuarios.

### **Evaluación de la herramienta**

- Si durante la caracterización del proyecto se comprueba que se trata de un proyecto activo, verificar los requisitos técnicos en la página web del proyecto y descargar la herramienta.
- En la página web del proyecto buscar información sobre la existencia de plugins o de una API pública.
- Instalar el programa según sus instrucciones para evaluar el tiempo necesario para la instalación y los conocimientos técnicos necesarios.
- Explorar la interfaz de usuario del programa y consultar la documentación para evaluar la usabilidad.

- Revisar la documentación disponible para valorar su actualización y exhaustividad, así como la existencia de ayudas visuales u otro material multimedia complementario.
- A partir de la información incluida en la página web y la documentación de cada herramienta definir la lista de funciones disponibles y formatos soportados. A modo de comprobación, explorar la interfaz de usuario en busca de las opciones correspondientes.



## ANEXO 4: PROPIEDADES DE LOS PROYECTOS EVALUADOS

A continuación para cada proyecto evaluado se presentan los datos tal como fueron etiquetados en la *wiki*. Cada proyecto se presenta en una tabla que incluye las propiedades semánticas aplicables en cada caso (en orden alfabético) con sus correspondientes valores.

<b>Anaphraseus</b>	
Active developers	1
Application Type	Desktop
Bug tracking	<a href="http://sourceforge.net/tracker/?group_id=207409&amp;amp%3Batid=1001492">http://sourceforge.net/tracker/?group_id=207409&amp;amp%3Batid=1001492</a>
Contributing procedures	<a href="http://anaphraseus.sourceforge.net/contributionform.html">http://anaphraseus.sourceforge.net/contributionform.html</a>
Copyright ownership	Individual copyright holder
Created	8 October 2007
Decision making	Centralized
Description	Anaphraseus is a CAT (Computer Aided Translation) tool, OpenOffice.org macro set similar to famous Wordfast.
Documentation URL	<a href="http://baldwinsoftware.com/AnaphraseusManual_1.23b.html">http://baldwinsoftware.com/AnaphraseusManual_1.23b.html</a>
Download page	<a href="http://sourceforge.net/projects/anaphraseus/files">http://sourceforge.net/projects/anaphraseus/files</a>
Downloads after release	170
Evaluated	15 April 2012
Facebook	<a href="http://www.facebook.com/Anaphraseus">http://www.facebook.com/Anaphraseus</a>
Feature requests	<a href="http://sourceforge.net/tracker/?group_id=207409&amp;amp%3Batid=1001495">http://sourceforge.net/tracker/?group_id=207409&amp;amp%3Batid=1001495</a>
Forge	<a href="http://sourceforge.net/projects/anaphraseus">http://sourceforge.net/projects/anaphraseus</a>
Formats-text	TXT, DOC, DOT, RTF, DOCX, DOTX, ODT, SXW
Formats-translation	TXT (WordFast TM)
Forum URL	<a href="http://groups.google.com/group/anaphraseus">http://groups.google.com/group/anaphraseus</a>
Governance	Benevolent dictatorship
Has function	Pre-translation, Fuzzy matching, Automatic insertion of exact matches, Automatic insertion of fuzzy matches, Customizable fuzzy match rate, Glossary matches, External dictionaries, Remote MT interface, Non-translatables, Modify segmentation during translation, Auto-complete, Global search and replace, Concordance search in project files, Format preview, On-the-fly spell checking, On-demand spell checking, Customizable spell-checking dictionaries, On-the-fly grammar/style checking, On-demand grammar/style checking
Homepage	<a href="http://anaphraseus.sourceforge.net">http://anaphraseus.sourceforge.net</a>
Implements	Unicode, TMX
Irc	irc://irc.freenode.net#anaphraseus
Is addon of	OpenOffice.org, LibreOffice
License	GNU General Public License
License permissiveness	strong copyleft
License strategy	Single open source
Major releases	3
Messages per month	2
Minor releases	0
Modification date	26 May 2012 10:07:11
Name	Anaphraseus
Number of downloads	655
Ohloh	<a href="http://www.ohloh.net/p/anaphraseus">http://www.ohloh.net/p/anaphraseus</a>
Operating system	Windows, GNU/Linux, Mac OS X

Originator	Self-sustained project
Portability	2
Programming language	OpenOffice.org Basic
Project ethics	Hacker ethics
Project leadership	Original developer <a href="http://melodiadetraduccion.wordpress.com/2010/03/15/anaphraseus">http://melodiadetraduccion.wordpress.com/2010/03/15/anaphraseus</a> , <a href="http://languageexposure.wordpress.com/2011/04/25/traducir-con-software-libre-anaphraseus">http://languageexposure.wordpress.com/2011/04/25/traducir-con-software-libre-anaphraseus</a> , <a href="http://sabinecretella.blogspot.com.es/2008/03/anaphraseus-cat-tool-supporting-iso-639.html">http://sabinecretella.blogspot.com.es/2008/03/anaphraseus-cat-tool-supporting-iso-639.html</a> , <a href="http://blog.albatrossolutions.com/tent-cat-tools/anaphraseus-121-beta">http://blog.albatrossolutions.com/tent-cat-tools/anaphraseus-121-beta</a> ,
Published materials	<a href="http://thoughtsontranslation.com/2008/10/08/anaphraseus-a-free-tent-that-works-within-openofficeorg">http://thoughtsontranslation.com/2008/10/08/anaphraseus-a-free-tent-that-works-within-openofficeorg</a> , <a href="http://www.booki.cc/open-translation-tools/anaphraseus">http://www.booki.cc/open-translation-tools/anaphraseus</a> , <a href="http://www.prlog.org/10427404-free-open-source-translation-memory-software-omegat-vs-anaphraseus-tm.html">http://www.prlog.org/10427404-free-open-source-translation-memory-software-omegat-vs-anaphraseus-tm.html</a> , <a href="http://translations.atranspl.com/working-with-anaphraseus-in-openoffice">http://translations.atranspl.com/working-with-anaphraseus-in-openoffice</a> , <a href="http://tonybaldwin.me/blog/2010/02/20/anaphraseus-2-01-released-2">http://tonybaldwin.me/blog/2010/02/20/anaphraseus-2-01-released-2</a>
Release date	12 December 2011
Repository	<a href="http://sourceforge.net/scm/?type=svn&amp;group_id=207409">http://sourceforge.net/scm/?type=svn&amp;group_id=207409</a>
Response time	301
Reviews	<a href="http://sourceforge.net/projects/anaphraseus/reviews/">http://sourceforge.net/projects/anaphraseus/reviews/</a>
Status	beta
Subscribers	40 <a href="http://www.proz.com/forum/cat_tools_technical_help/216413-is_anaphraseus_a_wordfast_version.html">http://www.proz.com/forum/cat_tools_technical_help/216413-is_anaphraseus_a_wordfast_version.html</a> , <a href="http://www.translatorscafe.com/cafe/MegaBBS/thread-view.asp?threadid=16358">http://www.translatorscafe.com/cafe/MegaBBS/thread-view.asp?threadid=16358</a>
Threads	<a href="http://topsy.com/s?q=anaphraseus">http://topsy.com/s?q=anaphraseus</a> <a href="http://twitter.com/Anaphraseus">http://twitter.com/Anaphraseus</a>
Tweets	
Twitter URL	
Type of community	Insulated developer
Typology	Translation environment
Usability	3
Version	2.04 beta
Categories	Translation Tools
<b>Autshumato ITE</b>	
Active developers	1
Affiliation	South African Department of Arts and Culture, Centre for Text Technology (North-West University), University of Pretoria
Application Type	Desktop
Bug tracking	<a href="http://sourceforge.net/tracker/?group_id=252474&amp;Batid=1229373">http://sourceforge.net/tracker/?group_id=252474&amp;Batid=1229373</a>
Copyright ownership	Legal entity
Created	4 February 2009
Decision making	Centralized
Description	Autshumato Integrated Translation Environment (ITE) provides a single translation environment that contains translation memory, machine translation and a glossary to facilitate the translation process.
Download page	<a href="http://sourceforge.net/projects/autshumatoite/files/autshumatoite/Releases/">http://sourceforge.net/projects/autshumatoite/files/autshumatoite/Releases/</a>
Downloads after release	36
Evaluated	25 April 2012
Forge	<a href="http://sourceforge.net/projects/autshumatoite">http://sourceforge.net/projects/autshumatoite</a>
Formats-DTP	XML (Infix), XTG (QuarkXPress), TAG (QuarkXPress)
Formats-multimedia	SVG, XML (Flash export)
Formats-software localization	PO, Java Resource Bundles, TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)
Formats-text	TXT, CSV, TAB, DOCX, DOTX, XLSX, PPTX, ODT, ODS, ODP, SXW, SXC, SXI, SRT, DocBook, LaTeX
Formats-translation	TXML (WordFast Pro)
Formats-web localization	HTML, XML, RESX
Forum URL	<a href="http://sourceforge.net/projects/autshumatoite/forums/forum/915128">http://sourceforge.net/projects/autshumatoite/forums/forum/915128</a>
Governance	Benevolent dictatorship
Has function	Multiple source documents, Document analysis, Batch processing, Pre-translation, Pre-translation with resource prioritizing, Fuzzy matching, Automatic insertion of exact matches, Automatic insertion of fuzzy matches, TM multi-directionality,

	Multiple TMs per project, Multilingual TMs, Alternative translations (same source), Glossary matches, Multiple glossaries per project, External dictionaries, Local MT interface, Remote MT interface, Display of resource meta-data, Support for segment comments, Filter-based navigation, Auto-propagation, Project statistics, Concordance search in project files, Concordance search in external files, Format preview, On-the-fly spell checking, On-the-fly grammar/style checking, On-demand quality checks, Customizable filters, Customizable segmentation rules, Modify segmentation during translation, Customizable fuzzy match rate, Customizable spell-checking dictionaries, Customizable style checks, Regex-based search, Customizable keyboard shortcuts
Homepage	<a href="http://autshumato.sourceforge.net/ITE.html">http://autshumato.sourceforge.net/ITE.html</a>
Implements	Unicode, TBX, TMX
Integrates	OmegaT, OpenOffice.org, Moses
License	GNU General Public License version 2.0
License permissiveness	strong copyleft
License strategy	Single open source
Listed	<a href="http://softpedia.com/get/Programming/Other-Programming-Files/Autshumato-ITE.shtml">http://softpedia.com/get/Programming/Other-Programming-Files/Autshumato-ITE.shtml</a>
Major releases	1
Minor releases	0
Modification date	29 July 2012 00:58:47
Name	Autshumato ITE
Number of downloads	78
Ohloh	<a href="http://www.ohloh.net/p/autshumatoite">http://www.ohloh.net/p/autshumatoite</a>
Operating system	Windows, GNU/Linux, Mac OS X
Originator	Public funding
Portability	3
Programming language	Java
Project ethics	Salary-based commercial ethics
Project leadership	Sponsoring organization(s)
Published materials	<a href="http://www.mt-archive.info/EAMT-2009-Groenewald.pdf">http://www.mt-archive.info/EAMT-2009-Groenewald.pdf</a> , <a href="http://traduccionmundolibre.com/2010/10/20/autshumato-ite-omegat-itaminado/">http://traduccionmundolibre.com/2010/10/20/autshumato-ite-omegat-itaminado/</a>
Related tools	Autshumato TMS, TMX Integrator, CText Alignment Interface Pro, PDF Text Extractor
Release date	20 March 2012
Repository	<a href="http://sourceforge.net/scm/?type=svn&amp;group_id=252474">http://sourceforge.net/scm/?type=svn&amp;group_id=252474</a>
Requirements	Java Runtime Environment 1.6_17+
Reviews	<a href="http://sourceforge.net/projects/autshumatoite/reviews/">http://sourceforge.net/projects/autshumatoite/reviews/</a>
Status	beta
Tweets	<a href="http://topsy.com/autshumato.sourceforge.net/">http://topsy.com/autshumato.sourceforge.net/</a>
Type of community	Legal entity
Typology	Translation environment
Usability	3
Version	2.0.0
Categories	Translation Tools

### **EsperantiloTM**

Application Type	Desktop
Copyright ownership	Individual copyright holder
Created	12 June 2008
Description	EsperantiloTM is a program for translators which facilitates the translation of papers by using a translation memory.
Documentation URL	<a href="http://www.esperantilo.org/tm/esperantilotm_en.pdf">http://www.esperantilo.org/tm/esperantilotm_en.pdf</a>
Evaluated	25 April 2012
Formats-text	TXT, DOC, ODT
Formats-web localization	HTML, XML
Homepage	<a href="http://www.esperantilo.org/tm/">http://www.esperantilo.org/tm/</a>
Implements	Unicode, TMX, XLIFF
Integrates	Hunspell
Is derivative of	Esperantilo
License	GNU General Public License
License permissiveness	strong copyleft

License strategy	Single open source
Modification date	6 May 2012 18:26:25
Name	EsperantiloTM
Operating system	Windows, GNU/Linux
Originator	Self-sustained project
Programming language	Tcl
Project ethics	Hacker ethics
Project leadership	Original developer
Release date	25 July 2008
Status	inactive
Type of community	Insulated developer
Typology	Translation environment
Version	0.992
Categories	Translation Tools

### ForeignDesk

Affiliation	LionBridge
Application Type	Desktop
Copyright ownership	Legal entity
Created	25 September 2001
Description	ForeignDesk is a CAT/TM tool published by LionBridge under an OS license.
Evaluated	30 April 2012
Forge	<a href="http://sourceforge.net/projects/foreigndesk/">http://sourceforge.net/projects/foreigndesk/</a>
Forum URL	<a href="http://sourceforge.net/projects/foreigndesk/forums/forum/120442">http://sourceforge.net/projects/foreigndesk/forums/forum/120442</a>
Homepage	<a href="http://sourceforge.net/projects/foreigndesk/">http://sourceforge.net/projects/foreigndesk/</a>
License	BSD License, IBM Public License
License permissiveness	no copyleft, strong copyleft
License strategy	Multiple open source
Listed	<a href="http://www.7brands.com/top_translation_software_ef.htm">http://www.7brands.com/top_translation_software_ef.htm</a>
Modification date	30 May 2012 14:15:48
Name	ForeignDesk
Number of downloads	110
Ohloh	<a href="http://www.ohloh.net/p/foreigndesk">http://www.ohloh.net/p/foreigndesk</a>
Operating system	Windows
Originator	Private funding
Programming language	C#, C++, Visual Basic
Project ethics	Salary-based commercial ethics
Release date	2 May 2002
Repository	<a href="http://sourceforge.net/scm/?type=cvs&amp;group_id=36405">http://sourceforge.net/scm/?type=cvs&amp;group_id=36405</a>
Reviews	<a href="http://sourceforge.net/projects/foreigndesk/reviews">http://sourceforge.net/projects/foreigndesk/reviews</a>
Status	inactive
Threads	<a href="http://www.translatorscafe.com/cafe/MegaBBS/thread-view.asp?threadid=11158&amp;start=1">http://www.translatorscafe.com/cafe/MegaBBS/thread-view.asp?threadid=11158&amp;start=1</a>
Typology	Translation environment
Version	5.7.3
Categories	Translation Tools

### Frankenstein Translation Memory Suite

Application Type	Desktop
Copyright ownership	Individual copyright holder
Created	30 October 2003
Description	Translation memory software (TM tool) with enhanced capabilities, like networking/collaboration (http, rpc), encoding conversion, project management capabilities, email capability with attachments, and file tree diff.
Evaluated	25 April 2012
Forge	<a href="http://sourceforge.net/projects/frankenstein">http://sourceforge.net/projects/frankenstein</a>
Homepage	<a href="http://sourceforge.net/projects/frankenstein/">http://sourceforge.net/projects/frankenstein/</a>
License	GNU General Public License
License permissiveness	strong copyleft
License strategy	Single open source

Listed	<a href="http://translationstreet.co.uk/component/jdownloads/viewdownload/3-desktop-applications/1-frankenstein-tool-suite">http://translationstreet.co.uk/component/jdownloads/viewdownload/3-desktop-applications/1-frankenstein-tool-suite</a>
Modification date	31 May 2012 11:50:42
Name	Frankenstein Translation Memory Suite
Number of downloads	8
Operating system	Windows, GNU/Linux, Mac OS X
Originator	Self-sustained project
Programming language	C++, Java
Project ethics	Hacker ethics
Project leadership	Original developer
Release date	31 October 2003
Requirements	Java Runtime Environment
Status	inactive
Threads	<a href="http://www.proz.com/forum/smart_shoppers/10212-another_low_cost_cat_tool_%3A_frankenstein.html">http://www.proz.com/forum/smart_shoppers/10212-another_low_cost_cat_tool_%3A_frankenstein.html</a> , <a href="http://www.proz.com/forum/cat_tools_technical_help/38374-anyone_any_luck_with_frankenstein.html">http://www.proz.com/forum/cat_tools_technical_help/38374-anyone_any_luck_with_frankenstein.html</a>
Type of community	Insulated developer
Typology	Translation environment
Version	1.6 build 15
Categories	Translation Tools

### OmegaT

Active developers	5
Active subscribers	39
Application Type	Desktop
Bug tracking	<a href="http://sourceforge.net/tracker/?group_id=68187&amp;amp%3Batid=520347">http://sourceforge.net/tracker/?group_id=68187&amp;amp%3Batid=520347</a>
Contributing procedures	<a href="http://www.omegat.org/en/howtos/localizing_omegat.php">http://www.omegat.org/en/howtos/localizing_omegat.php</a>
Copyright ownership	Distributed
Created	28 November 2002
Decision making	Balanced
Description	OmegaT® is a free and open source multiplatform Computer Assisted Translation tool with fuzzy matching, translation memory, keyword search, glossaries, and translation leveraging into updated projects.
Dev-forum	<a href="http://sourceforge.net/mailarchive/forum.php?forum_name=omegat-development">http://sourceforge.net/mailarchive/forum.php?forum_name=omegat-development</a>
Documentation URL	<a href="http://www.omegat.org/en/documentation.html">http://www.omegat.org/en/documentation.html</a>
Download page	<a href="http://www.omegat.org/en/dl_overview.php">http://www.omegat.org/en/dl_overview.php</a>
Downloads after release	1,344
Evaluated	15 April 2012
Feature requests	<a href="http://sourceforge.net/tracker/?group_id=68187&amp;amp%3Batid=520350">http://sourceforge.net/tracker/?group_id=68187&amp;amp%3Batid=520350</a>
Forge	<a href="http://sourceforge.net/projects/omegat">http://sourceforge.net/projects/omegat</a>
Formats-DTP	XML (Infix), IDML (InDesign), XTG (QuarkXPress), TAG (QuarkXPress)
Formats-multimedia	SVG, XML (Flash export), CAMPROJ (Camstasia Studio)
Formats-software localization	RC, POT, PO, Java Resource Bundles, XML (Android resource), TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)
Formats-text	TXT, CSV, TAB, DOCX, DOTX, XLSX, XLTX, XLSM, PPTX, PPSX, POTX, ODT, ODS, ODP, SXW, SXC, SXI, SRT, DocBook, LaTeX
Formats-translation	TTX (SDL Trados), TXML (WordFast Pro)
Formats-web localization	HTML, XML, RESX, JSON
Forum URL	<a href="http://groups.yahoo.com/group/omegat">http://groups.yahoo.com/group/omegat</a> <a href="http://packages.ubuntu.com/search?keywords=omegat">http://packages.ubuntu.com/search?keywords=omegat</a> , <a href="http://packages.debian.org/search?keywords=omegat">http://packages.debian.org/search?keywords=omegat</a> , <a href="https://admin.fedoraproject.org/pkgdb/acls/name/OmegaT">https://admin.fedoraproject.org/pkgdb/acls/name/OmegaT</a>
GNU/Linux distros	
Governance	Anarchy
Has derivatives	OmegaTplus, Boltran, Autshumato ITE Multiple source documents, Customizable filters, Customizable segmentation rules, Document analysis, Batch processing, Pre-translation with resource prioritizing, Fuzzy matching, Automatic insertion of exact matches, Automatic insertion of fuzzy matches, Customizable fuzzy match rate, TM multi-directionality, Multiple TMs per project, Multilingual TMs, Alternative translations (same source), Glossary matches, Multiple glossaries per project, External dictionaries, Remote MT interface, Display of resource meta-data, Support for segment comments,
Has function	

	Filter-based navigation, Auto-propagation, Project statistics, Concordance search in project files, Concordance search in external files, On-the-fly spell checking, On-the-fly grammar/style checking, On-demand quality checks, Customizable quality checks
Homepage	<a href="http://www.omegat.org/">http://www.omegat.org/</a>
Implements	Unicode, TXT, TAB, CSV, TBX, TMX, XLIFF
Integrates	Hunspell, LanguageTool, Lucene Tokenizers, Okapi filters
Irc	<a href="irc://irc.freenode.net#omegat">irc://irc.freenode.net#omegat</a>
License	GNU General Public License
License permissiveness	strong copyleft
License strategy	Single open source
Listed	<a href="http://omegat.softpedia.com">http://omegat.softpedia.com</a> , <a href="http://en.kioskea.net/download/download-19967-omegat">http://en.kioskea.net/download/download-19967-omegat</a> , <a href="http://directory.fsf.org/wiki/OmegaT">http://directory.fsf.org/wiki/OmegaT</a> , <a href="http://www.taustracker.com/browse/tms/omegat">http://www.taustracker.com/browse/tms/omegat</a>
Major releases	2
Messages per month	304
Minor releases	13
Modification date	6 June 2012 08:40:54
Name	OmegaT
Number of downloads	5,033
Ohloh	<a href="http://www.ohloh.net/p/omegat">http://www.ohloh.net/p/omegat</a>
Operating system	Windows, GNU/Linux, Mac OS X
Originator	Self-sustained project
Programming language	Java
Project ethics	Hacker ethics
Project leadership	Recognized leaders <a href="http://www.ultimabooks.it/guida-completa-a-omegat-tecniche-trucchi-e-consigli-per-traduttori-e-project-manager-1">http://www.ultimabooks.it/guida-completa-a-omegat-tecniche-trucchi-e-consigli-per-traduttori-e-project-manager-1</a> , <a href="http://www.youtube.com/watch?v=iakqcUvRsEU">http://www.youtube.com/watch?v=iakqcUvRsEU</a> , <a href="http://www.your-translations.com/omegat.php">http://www.your-translations.com/omegat.php</a> , <a href="http://www.leuce.com/translate/omegat.html">http://www.leuce.com/translate/omegat.html</a> , <a href="http://translatorsteacup.lingocode.com/tag/omegat">http://translatorsteacup.lingocode.com/tag/omegat</a> , <a href="http://www.translationdirectory.com/article512.htm">http://www.translationdirectory.com/article512.htm</a> , <a href="http://www.youtube.com/watch?v=3Wv79R9Sp6E">http://www.youtube.com/watch?v=3Wv79R9Sp6E</a> , <a href="http://www.english-spanish-translator.org/omegat">http://www.english-spanish-translator.org/omegat</a> , <a href="http://thoughtsontranslation.com/2008/04/11/omegat-a-free-and-very-useful-tent">http://thoughtsontranslation.com/2008/04/11/omegat-a-free-and-very-useful-tent</a> , <a href="http://pinitostranslation.com/2011/09/15/omega-for-translators">http://pinitostranslation.com/2011/09/15/omega-for-translators</a>
Published materials	
Release date	9 May 2012
Repository	<a href="http://sourceforge.net/scm/?type=svn&amp;group_id=68187">http://sourceforge.net/scm/?type=svn&amp;group_id=68187</a>
Requirements	Java Runtime Environment 1.5+
Response time	0.3
Reviews	<a href="http://sourceforge.net/projects/omegat/reviews/">http://sourceforge.net/projects/omegat/reviews/</a>
Services	Customized development
Status	stable
Subscribers	1,720 <a href="http://www.proz.com/forum/omegat_support-586.html">http://www.proz.com/forum/omegat_support-586.html</a> , <a href="http://www.linkedin.com/groups/OmegaT-Translation-Professionals-1769774">http://www.linkedin.com/groups/OmegaT-Translation-Professionals-1769774</a> , <a href="http://www.translatorscafe.com/cafe/MegaBBS/forum-view.asp?forumid=58&amp;group_id=7">http://www.translatorscafe.com/cafe/MegaBBS/forum-view.asp?forumid=58&amp;group_id=7</a>
Threads	<a href="http://sourceforge.net/projects/omegat/reviews/">http://sourceforge.net/projects/omegat/reviews/</a>
Tweets	<a href="http://topsy.com/s?q=omegat%2B%23x18">http://topsy.com/s?q=omegat%2B%23x18</a>
Type of community	Group of developers
Typology	Translation environment
Version	2.3.0_8 (standard), 2.5.3 (latest)
Categories	Translation Tools
<b>OmegaTplus</b>	
Application Type	Desktop
Bug tracking	<a href="http://sourceforge.net/tracker/?group_id=141182&amp;group_id=141182&amp;atid=748748">http://sourceforge.net/tracker/?group_id=141182&amp;group_id=141182&amp;atid=748748</a>
Copyright ownership	Distributed
Created	11 June 2005
Decision making	Centralized
Description	OmegaT+ has many of the good features that users expect in a translation tool presented in a straightforward manner that is simple and easy to use.

Documentation URL	<a href="http://omegatplus.sourceforge.net/doc/index.html">http://omegatplus.sourceforge.net/doc/index.html</a>
Download page	<a href="http://sourceforge.net/projects/omegatplus/files/OmegaT%2B">http://sourceforge.net/projects/omegatplus/files/OmegaT%2B</a>
Evaluated	25 April 2012
Facebook	<a href="http://www.facebook.com/pages/OmegaT/60193457075">http://www.facebook.com/pages/OmegaT/60193457075</a>
Feature requests	<a href="http://sourceforge.net/tracker/?group_id=141182&amp;amp%3BAtid=748751">http://sourceforge.net/tracker/?group_id=141182&amp;amp%3BAtid=748751</a>
Forge	<a href="http://sourceforge.net/projects/omegatplus">http://sourceforge.net/projects/omegatplus</a>
Formats-DTP	XTG (QuarkXPress), TAG (QuarkXPress)
Formats-software localization	RC, POT, PO, Java Resource Bundles, XML (Android resource), TS (Qt Linguist), DTD (Mozilla), HHC (HTML Help Compiler)
Formats-text	TXT, CSV, DOCX, DOTX, XLSX, PPTX, ODT, ODS, ODP, SXW, SXC, SXI, SRT, DocBook, LaTeX
Formats-web localization	HTML, XML, RESX
Forum URL	<a href="http://groups.google.com/group/omegatplus">http://groups.google.com/group/omegatplus</a>
Governance	Benevolent dictatorship
Has function	Multiple source documents, Document analysis, Fuzzy matching, Automatic insertion of exact matches, Automatic insertion of fuzzy matches, TM multi-directionality, Multiple TMs per project, Multilingual TMs, Glossary matches, Multiple glossaries per project, Remote MT interface, Auto-propagation, Project statistics, Global search and replace, Concordance search in project files, Concordance search in external files, On-demand quality checks, Customizable filters, Customizable segmentation rules, Modify segmentation during translation, Customizable fuzzy match rate, Regex-based search
Homepage	<a href="http://omegatplus.sourceforge.net/">http://omegatplus.sourceforge.net/</a>
Implements	Unicode, TXT, TAB, CSV, TMX, XLIFF
Is derivative of	OmegaT
License	GNU General Public License
License permissiveness	strong copyleft
License strategy	Single open source
Listed	<a href="http://downloads.yahoo.com/software/windows-web-tools-omegat—s161394">http://downloads.yahoo.com/software/windows-web-tools-omegat—s161394</a> , <a href="http://linux.softpedia.com/get/Text-Editing-Processing/Markup/OmegaTplus-26732.shtml">http://linux.softpedia.com/get/Text-Editing-Processing/Markup/OmegaTplus-26732.shtml</a>
Modification date	6 June 2012 09:37:20
Name	OmegaTplus
Number of downloads	233
Operating system	Windows, GNU/Linux, Mac OS X
Originator	Self-sustained project
Portability	3
Programming language	Java
Project ethics	Hacker ethics
Project leadership	Original developer
Release date	14 May 2010
Requirements	Java Runtime Environment
Reviews	<a href="http://sourceforge.net/projects/omegatplus/reviews/">http://sourceforge.net/projects/omegatplus/reviews/</a>
Services	Value-added subscriptions, Customized development
Status	inactive
Subscribers	93
Threads	<a href="http://www.translatorscafe.com/cafe/MegaBBS/thread-view.asp?threadid=11158&amp;start=1">http://www.translatorscafe.com/cafe/MegaBBS/thread-view.asp?threadid=11158&amp;start=1</a>
Type of community	Insulated developer
Typology	Translation environment
Usability	3
Version	1.0M3.1
Categories	Translation Tools

### **Open Language Tools**

Active developers	1
Affiliation	Oracle
Application Type	Desktop
Contributing procedures	<a href="http://open-language-tools.java.net/editor/xliff-editor-110n.html">http://open-language-tools.java.net/editor/xliff-editor-110n.html</a>
Copyright ownership	Legal entity
Created	21 June 2004
Decision making	Decentralized



Description	The Open Language Tools are a set of translation tools that aim to make the task of translating software and documentation a lot easier. Initially, they comprise of a full-featured XLIFF Translation Editor and a set of XLIFF file-filters for a number of documentation and software file formats.
Dev-forum	<a href="http://java.net/projects/open-language-tools/lists/dev/archive">http://java.net/projects/open-language-tools/lists/dev/archive</a>
Documentation URL	<a href="http://java.net/projects/open-language-tools/sources/svn/content/trunk/doc/user_doc/">http://java.net/projects/open-language-tools/sources/svn/content/trunk/doc/user_doc/</a>
Download page	<a href="http://java.net/projects/open-language-tools/downloads">http://java.net/projects/open-language-tools/downloads</a>
Evaluated	25 April 2012
Forge	<a href="http://java.net/projects/open-language-tools">http://java.net/projects/open-language-tools</a>
Formats-software localization	PO, Java Resource Bundles, DTD (Mozilla)
Formats-text	TXT, ODT, ODS, ODP, SXW, SXC, SXI, DocBook
Formats-web localization	HTML, XML, JSP
Forum URL	<a href="http://java.net/projects/open-language-tools/lists/users/archive">http://java.net/projects/open-language-tools/lists/users/archive</a>
Governance	Anarchy
Has function	Pre-translation, Fuzzy matching, Automatic insertion of exact matches, Automatic insertion of fuzzy matches, Alternative translations (same source), Display of resource meta-data, Segment status validation, Support for segment comments, Filter-based navigation, Auto-propagation, Alternative translations (same source), Project statistics, Global search and replace, Concordance search in project files, On-demand spell checking, On-demand quality checks, Customizable spell-checking dictionaries, Customizable quality checks, Customizable keyboard shortcuts
Homepage	<a href="http://open-language-tools.java.net/">http://open-language-tools.java.net/</a>
Implements	Unicode, TMX, XLIFF
Integrates	ASpell, Hunspell
License	Common Development and Distribution License (CDDL)
License permissiveness	weak copyleft
License strategy	Single open source
Major releases	0
Minor releases	0
Modification date	7 August 2012 23:17:09
Name	Open Language Tools
Ohloh	<a href="http://www.ohloh.net/p/open-language-tools">http://www.ohloh.net/p/open-language-tools</a>
Operating system	Windows, GNU/Linux, Mac OS X
Originator	Private funding
Portability	2
Programming language	Java
Project ethics	Hybrid ethics
Project leadership	Sponsoring organization(s)
Published materials	<a href="http://www.translationdirectory.com/article507.htm">http://www.translationdirectory.com/article507.htm</a>
Release date	19 January 2012
Repository	<a href="http://java.net/projects/open-language-tools/sources/svn/show">http://java.net/projects/open-language-tools/sources/svn/show</a>
Requirements	Java Runtime Environment
Status	beta
Subscribers	40
Threads	<a href="http://www.proz.com/forum/translator_resources/33669-open_source_cat_tool_from_sun_and_developer_tim_foster_%3A_open_language_tools.html">http://www.proz.com/forum/translator_resources/33669-open_source_cat_tool_from_sun_and_developer_tim_foster_%3A_open_language_tools.html</a>
Type of community	Group of developers
Typology	Translation environment
Usability	2
Version	1.4.0 (dev.)
Categories	Translation Tools
<b>OpenTM2</b>	
Active developers	3
Affiliation	IBM Corporation, LISA, Linux Solution Group, Welocalize, Cisco
Application Type	Desktop
Bug tracking	<a href="http://source.opentm2.org:8000/opentm2">http://source.opentm2.org:8000/opentm2</a>
Copyright ownership	Legal entity
Created	4 June 2010



Decision making	Centralized
Description	TranslationManager/2 (TM/2) originates from the IBM TranslationManager tool and has been made available to the open source community. TM/2 is a computer-assisted translation system (CAT) that automates repetitive tasks, freeing the professional translator to attend to the finer points of translation that require the judgement of an expert.
Dev-forum	<a href="http://groups.google.com/group/opentm2-design">http://groups.google.com/group/opentm2-design</a>
Documentation URL	<a href="http://www.opentm2.org/frequently-asked-questions">http://www.opentm2.org/frequently-asked-questions</a>
Download page	<a href="http://www.opentm2.org/download">http://www.opentm2.org/download</a>
Evaluated	23 April 2012
Feature requests	<a href="http://source.opentm2.org:8000/opentm2">http://source.opentm2.org:8000/opentm2</a>
Forge	<a href="http://sourceforge.net/projects/opentm2">http://sourceforge.net/projects/opentm2</a>
Formats-text	TXT, DOC, RTF, PPT , Ami Pro, BookMaster®
Formats-web localization	HTML, SGM
Forum URL	<a href="http://groups.google.com/group/opentm2-support">http://groups.google.com/group/opentm2-support</a>
Governance	Benevolent dictatorship
Has function	Multiple source documents, Document analysis, Pre-translation, Fuzzy matching, Automatic insertion of exact matches, Automatic insertion of fuzzy matches, Multiple TMs per project, Glossary matches, Multiple glossaries per project, Access to remote resources, Display of resource meta-data, Support for segment comments, Project statistics, Global search and replace, Concordance search in project files, Concordance search in external files, Format preview, On-the-fly spell checking, On-demand spell checking, Modify segmentation during translation, Customizable fuzzy match rate, Customizable spell-checking dictionaries, Customizable keyboard shortcuts
Homepage	<a href="http://www.opentm2.org">http://www.opentm2.org</a>
Implements	Unicode
License	Eclipse Public License
License permissiveness	no copyleft
License strategy	Single open source
Listed	<a href="http://www.taustracker.com/opentm2">http://www.taustracker.com/opentm2</a>
Major releases	1
Messages per month	4
Minor releases	1
Modification date	7 August 2012 23:30:01
Name	OpenTM2
Ohloh	<a href="http://www.ohloh.net/p/OpenTM2">http://www.ohloh.net/p/OpenTM2</a>
Operating system	Windows
Originator	Private funding
Portability	3
Project ethics	Salary-based commercial ethics
Project leadership	Steering committee <a href="http://www.globalvis.com/joomla-globalsight-and-open-tm2/">http://www.globalvis.com/joomla-globalsight-and-open-tm2/</a> , <a href="http://www.mancomun.org/es/no_cache/actualidade/detalledenova/nova/ibm-libera-opentm2-unha-ferramenta-para-a-xestion-de-proxectos-de-traducion-baseada-en-estandare/">http://www.mancomun.org/es/no_cache/actualidade/detalledenova/nova/ibm-libera-opentm2-unha-ferramenta-para-a-xestion-de-proxectos-de-traducion-baseada-en-estandare/</a> , <a href="http://globalizer.wordpress.com/tag/opentm2/">http://globalizer.wordpress.com/tag/opentm2/</a>
Published materials	
Release date	29 July 2011
Repository	<a href="http://www.opentm2.org/development/source-repository">http://www.opentm2.org/development/source-repository</a>
Roadmap	<a href="http://www.opentm2.org/deliverables">http://www.opentm2.org/deliverables</a>
Status	stable
Subscribers	48 <a href="http://www.proz.com/forum/cat_tools_technical_help/174563-experiences_of_opentm2.html">http://www.proz.com/forum/cat_tools_technical_help/174563-experiences_of_opentm2.html</a> , <a href="http://tech.groups.yahoo.com/group/catmt/message/1207">http://tech.groups.yahoo.com/group/catmt/message/1207</a>
Threads	
Type of community	Organization of developers
Typology	Translation environment
Usability	3
Version	0.9.5
Wiki	<a href="http://www.beo-doc.de/opentm2wiki/">http://www.beo-doc.de/opentm2wiki/</a>
Categories	Translation Tools

## Transolution

Application Type	Desktop
Copyright ownership	Individual copyright holder
Created	24 February 2005
Description	Versatile open source translation editor using the XLIFF standard. It aims to provide translators with professional quality editor for both documentation and software. Features tag protection and interactive Translation Memory.
Download page	<a href="http://sourceforge.net/projects/eviltrans/files/Transolution/">http://sourceforge.net/projects/eviltrans/files/Transolution/</a>
Evaluated	25 April 2012
Forge	<a href="http://sourceforge.net/projects/eviltrans/">http://sourceforge.net/projects/eviltrans/</a>
Homepage	<a href="http://sourceforge.net/projects/eviltrans/">http://sourceforge.net/projects/eviltrans/</a>
License	GNU General Public License
License permissiveness	strong copyleft
License strategy	Single open source
Listed	<a href="http://freecode.com/projects/transolution">http://freecode.com/projects/transolution</a> , <a href="http://pypi.python.org/pypi/Transolution/0.4b5">http://pypi.python.org/pypi/Transolution/0.4b5</a>
Modification date	2 June 2012 12:48:41
Name	Transolution
Operating system	Windows, GNU/Linux, Mac OS X
Originator	Self-sustained project
Programming language	Python
Project ethics	Hacker ethics
Project leadership	Original developer
Release date	21 August 2005
Status	inactive
Type of community	Insulated developer
Typology	Translation environment
Version	0.4b5
Categories	Translation Tools

### **Virtaal**

Active developers	4
Affiliation	Zuza Software Foundation
Application Type	Desktop
Bug tracking	<a href="http://bugs.locamotion.org/enter_bug.cgi?product=Virtaal">http://bugs.locamotion.org/enter_bug.cgi?product=Virtaal</a>
Contributing procedures	<a href="http://translate.sourceforge.net/wiki/virtaal/localising_virtaal">http://translate.sourceforge.net/wiki/virtaal/localising_virtaal</a> , <a href="http://translate.sourceforge.net/wiki/virtaal/testing">http://translate.sourceforge.net/wiki/virtaal/testing</a>
Copyright ownership	Legal entity
Created	16 October 2008
Decision making	Centralized
Description	Virtaal is a graphical translation tool. It is meant to be easy to use and powerful at the same time. Although the initial focus is on software translation (localisation or l10n), we definitely intend it to be useful for several purposes.
Dev-forum	<a href="https://lists.sourceforge.net/lists/listinfo/translate-devel">https://lists.sourceforge.net/lists/listinfo/translate-devel</a>
Documentation URL	<a href="http://translate.sourceforge.net/wiki/virtaal/using_virtaal">http://translate.sourceforge.net/wiki/virtaal/using_virtaal</a>
Download page	<a href="http://sourceforge.net/projects/translate/files/Virtaal">http://sourceforge.net/projects/translate/files/Virtaal</a>
Downloads after release	271
Evaluated	25 April 2012
Facebook	<a href="http://www.facebook.com/pages/Virtaal/107874659235398">http://www.facebook.com/pages/Virtaal/107874659235398</a>
Forge	<a href="http://sourceforge.net/projects/translate">http://sourceforge.net/projects/translate</a>
Formats-software localization	MO, PO, TS (Qt Linguist), QPH (Qt Phrase Book)
Formats-translation	TXT (WordFast TM)
GNU/Linux distros	<a href="https://launchpad.net/ubuntu/%2Bsource/virtaal">https://launchpad.net/ubuntu/%2Bsource/virtaal</a> , <a href="http://packages.debian.org/source/squeeze/virtaal">http://packages.debian.org/source/squeeze/virtaal</a> , <a href="https://admin.fedoraproject.org/pkgdb/acls/name/virtaal">https://admin.fedoraproject.org/pkgdb/acls/name/virtaal</a>
Governance	Benevolent dictatorship Fuzzy matching, Automatic insertion of exact matches, Automatic insertion of fuzzy matches, Glossary matches, Multiple glossaries per project, Access to remote resources, Local MT interface, Remote MT interface, Display of resource meta-data, Segment status validation, Filter-based navigation, Auto-complete, Auto-propagation, Project statistics, Global search and replace, Concordance search in project files, On-the-fly spell checking, On-the-fly quality checks, Customizable spell-checking dictionaries, Regex-based search, Customizable placeables,
Has function	

Homepage	<a href="http://translate.sourceforge.net/wiki/virtaal/index">http://translate.sourceforge.net/wiki/virtaal/index</a>
Implements	Unicode, TXT, TAB, TBX, TMX, XLIFF
Integrates	Translate Toolkit, Hunspell, Myspell
Irc	irc://irc.freenode.net/#pootle
License	GNU General Public License 2.0
License permissiveness	strong copyleft
License strategy	Single open source
Listed	<a href="http://www.softpedia.com/get/Others/Home-Education/Virtaal.shtml">http://www.softpedia.com/get/Others/Home-Education/Virtaal.shtml</a> , <a href="http://virtaal.softonic.com">http://virtaal.softonic.com</a> , <a href="http://en.wikibooks.org/wiki/CAT-Tools/Virtaal">http://en.wikibooks.org/wiki/CAT-Tools/Virtaal</a> , <a href="http://www.taustracker.com/virtaal">http://www.taustracker.com/virtaal</a>
Major releases	1
Minor releases	1
Modification date	7 August 2012 23:50:33
Name	Virtaal
Number of downloads	1,152
Ohloh	<a href="http://www.ohloh.net/p/virtaal">http://www.ohloh.net/p/virtaal</a>
Operating system	Windows, GNU/Linux, Mac OS X
Originator	Private funding
Portability	3
Programming language	Python
Project ethics	Hybrid ethics
Project leadership	Foundation
Published materials	<a href="http://econsultant.co.za/virtaal-simplifies-software-translation">http://econsultant.co.za/virtaal-simplifies-software-translation</a>
Related tools	Pootle, Translate Toolkit, Corpuscatcher, Spelt
Release date	9 January 2012
Repository	<a href="https://github.com/translate/virtaal">https://github.com/translate/virtaal</a>
Requirements	Translate Toolkit, Python (Windows version includes all dependencies)
Reviews	<a href="http://sourceforge.net/projects/translate/reviews">http://sourceforge.net/projects/translate/reviews</a>
Roadmap	<a href="http://translate.sourceforge.net/wiki/virtaal/development_plans">http://translate.sourceforge.net/wiki/virtaal/development_plans</a>
Status	stable
Threads	<a href="http://www.proz.com/forum/localization/118252-simple_localisation_translation_tool_virtaal.html">http://www.proz.com/forum/localization/118252-simple_localisation_translation_tool_virtaal.html</a>
Tweets	<a href="http://topsy.com/s?q=virtaal">http://topsy.com/s?q=virtaal</a>
Type of community	Legal entity
Typology	Translation environment
Usability	3
Version	0.7.1
Categories	Translation Tools