

Estudio teórico y evidencia empírica en la aplicación de técnicas de análisis y modelado al proceso de producción multimedia

Miguel Ángel Brigos Hermida

Director.

Josep M. Monguet Fierro

Barcelona, Agosto de 2001

3.2.1.1.1 Normas para Documentación "ServiCaixa"

Para editar transacciones y aplicaciones adicionales en el entorno Servicaixa es necesaria una estación Windows con los programas:

- Amipro 3.0
- FlowChard

Uso de FlowChard

Se utilizarán objetos linkados OLE, de tal forma que no existen archivos originales FlowChard.

En Amipro se crea un marco vacío del tamaño de una página, y se le inserta un objeto (también de una página) FlowChard. (Edición-Insertar-Objeto-FlowChard).

Los marcos siempre deben estar vinculados al párrafo precedente.

Configuración de Amipro

Directorio por omisión C:\TA\DOCUI

Hoja de Estilos por Omisión MABBTAI.STY

Salvar archivos sin incluir formato, siempre debe utilizarse la Hoja de Estilos, (y la misma en todos los documentos).

No imprimir en segundo plano (si se imprime desde el documento maestro)

Guardar automáticamente, y no utilizar archivos .BAK.

Vista-Preferencias, (Retornos y Tabuladores, Campos, Formato final)

Estructura de directorios, y archivos

C:\TA\DOCU\APLICA\

Residen los archivos que documentan la aplicación a nivel Nodo.

Todos los archivos están vinculados a un documento maestro llamado TAI.STY

El índice de materias se llama ITAI.STY.

C:\TA\DOCU\MODULOS\

Residen los archivos que documentan los módulos ejecutables por la aplicación.

Todos los archivos están vinculados a un documento maestro llamado MODULOS.STY

El índice de materias se llama IMODUL.STY.

3.2.1.1.2 Lista de documentos solicitados por una empresa de ingeniería de software para la realización de un proyecto.

Estos documentos son los que solicita Fujitsu para la realización de proyectos de ingeniería de software. La normativa en cuanto a la existencia de dicha documentación no se limita a los proveedores, sino que se solicita incluso entre departamentos de la misma compañía. (p. ej. dpt1 y dpt2) de tal forma que existe un tercer departamento, centro gestor dpt3, que valida la existencia de los mismos, aunque no entre en su contenido.

Esta documentación se puede considerar parte del Pedido del Cliente.

Deliberables (Entregas) por elemento

- Técnica: (Objetivo, Mantenibilidad futura)
 - Requerimientos Externos (Cliente-Fujitsu)
 - Especificaciones Externas (Proveedor)
 - Diagrama de flujo (Proveedor)
 - Descripción submódulos internos (Proveedor)
 - Interfaces internos entre submódulos (Proveedor)
 - Fuentes Sw (Proveedor)
 - Documentación pruebas realizadas (Proveedor)
 - Ejemplo documentación pruebas (Cliente-Fujitsu)
- Seguimiento (Periodicidad 15 días)
 - Schedule (Plan) en Project v.4 (Papel y Floppy)
 - Recursos
 - Hitos de control
 - Subtareas
 - Grado de progreso en cada revisión
 - Documento (2 Hojas) Resumen

- Actividades realizadas
- Objetivos alcanzados
- Actividades próximo periodo
 - Acciones a realizar para conseguir los objetivos
 - Lista de Acciones

3.2.1.1.3 Requisitos funcionales, billetes y localidades.

En este apartado se presenta el análisis funcional de dos transacciones: la venta de billetes de transportes, y la venta de “entradas” de espectáculos.

En el documento se puede observar la duplicidad de algunos de los aspectos del análisis.

3.2.1.1.3.1 Terminología y nomenclatura

Sistema

Llamaremos Sistema al entorno existente en el que se integra la aplicación de venta de localidades construida por MABB.

Aplicación

La aplicación de venta de localidades construida por MABB

Modulo funcional

Cada una de las funcionalidades principales en que se divide la aplicación. Nos referiremos a ellos como **Módulo**

Host

Genéricamente, el ordenador central.

Espacio funcional

Estructuras de datos, de procesos y de sus relaciones que pertenecen a un módulo.

Un módulo puede tener más de un espacio funcional

Estado de interacción

Intervalo de tiempo en el cual la aplicación espera uno o más eventos desde algún dispositivo, físico o virtual, para resolver una indeterminación.

Un estado está soportado por uno o más espacios funcionales o partes de ellos.

Todos los procesos que llevan a un estado y todos los que son consecuencia de él, están descritos por espacios funcionales precisos. En éste análisis se tratarán estos espacios como tales y desde el punto de vista del estado.

Al estado de interacción lo llamaremos simplemente **Estado**

Proceso

Estado de interacción en que el dispositivo del que se espera el evento es diferente del de punteo y teclado, o no existe este dispositivo, por lo que la aplicación usará un evento conocido de antemano, o estará implícito en la información que posee.

Esquema de funcionamiento de la aplicación de venta de localidades

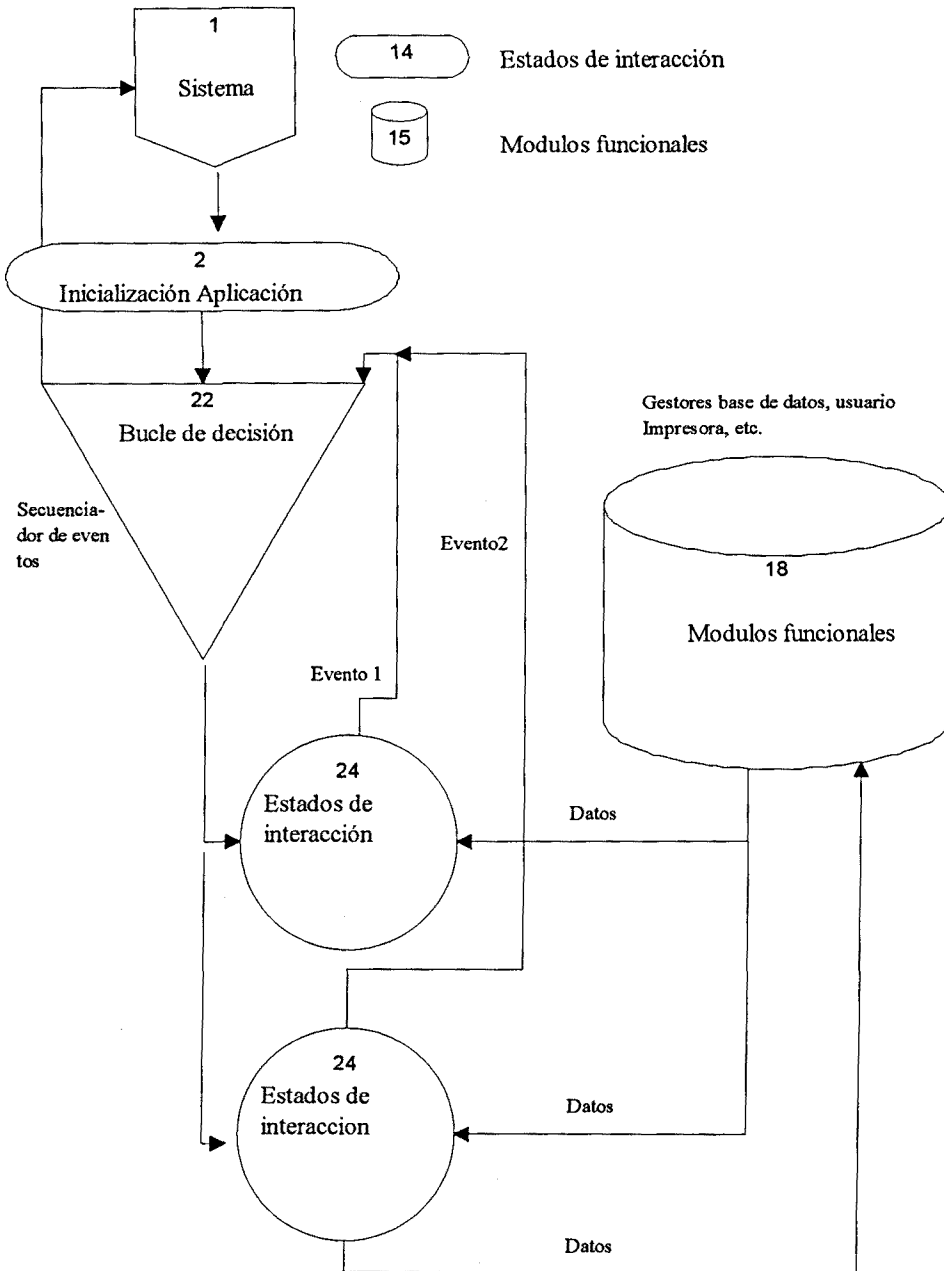


Ilustración 11. Esquema de funcionamiento de venta de localidades. Fujitsu TAI

3.2.1.1.3.2 Venta de Billetes

3.2.1.1.3.2.1 Generalidades

Características

La aplicación de billetes FBILL se integra en un entorno más amplio de prestaciones bancarias caracterizado por :

- Multitarea (sistema multitenhebrado).
- Procesos distribuidos.
- Comunicación a través de colas de mensajes.
- APIs de recubrimiento para gran parte del control de los dispositivos.

Así pues, la aplicación cooperará con otros procesos existentes en la máquina para llevar a término la prestación de venta de billetes y deberá tomar de ellos los datos inherentes al sistema y deberá reportar, así mismo, los datos que se le requieran.

Por otra parte, la aplicación accederá, a través de los APIs, a los dispositivos necesarios.

Estructura

La aplicación tiene, estructuralmente, dos ámbitos distintos :

- Módulos funcionales
- Estados de interacción

Un conjunto de módulos funcionales o partes de ellos, generan un espacio funcional que será donde se desarrollará el estado de interacción. Así, un estado de interacción, la pantalla de billetes, necesita de un gestor de base de datos, un gestor de pantalla, un interfaz con el usuario, etc. para ser generado.

En los esquemas anteriores, se observa como los módulos funcionales alimentan los estados de interacción. Los estados de interacción generan, a la salida, un evento y un mensaje que son tratados por el gestor de estados. Un elemento de este gestor de eventos es el bucle de decisión. Es éste un mecanismo que envía el flujo de la aplicación a un estado de interacción que es función del evento-mensaje encolado por el último estado de interacción.

En el segundo diagrama se observan con más detalle los estados de interacción, aunque agrupados por similitudes funcionales. Es decir, estados cuyo flujo de entrada de datos es básicamente desde la base de datos y desde el host, estados cuyo flujo principal es de salida de datos hacia las impresoras, hacia el sistema, etc.

Funcionalidades

La aplicación tiene como función generar la venta e impresión de tickets de localidades para los espectáculos definidos en las tablas de localidades existentes en un momento determinado en la máquina.

Para ello debe resolver las siguientes funciones :

1. Inicializar datos y procesos
2. Obtener datos de tablas
3. Presentar datos a usuario y esperar su indicación
4. Impresión por pantalla e impresora de aquellos datos de los que se quiere que el usuario tenga conocimiento.
5. Pedir datos y generar transacciones asociadas a la venta de billetes al host.
6. Imprimir los billetes de transporte pedidos.
7. Finalizar

Estados de interacción

Para resolver las funciones anteriormente citadas, la aplicación genera una sucesión de estados donde se desarrolla un dialogo Aplicación-Usuario. Estos estados se listan, genéricamente, a continuación :

1. Inicializaciones. Se inicializan los datos y se genera el siguiente estado.
2. Gestión de billetes. Se resuelve la cantidad y el tipo de billetes pedidos.
3. Presentación de información textual por pantalla o impresora. Se muestra información al usuario para ayudarlo en su decisión.
4. Presentación de mensajes informativos. Se indica al usuario de algún problema, indicación errónea, etc.
5. Venta de Billetes. Se cargan encuentra los importes de los billetes pedidos. Si hay algún problema se genera la anulación con cargo condicional correspondiente, o la anulación total si no se entregan billetes.
6. Generación de los billetes de las localidades. Termina el proceso entregando los billetes al usuario.
7. Finalizaciones. Termina los procesos puestos en marcha por la aplicación, envía datos al sistema retorna a él.

3.2.1.1.3.2.2 Módulos funcionales

Definimos los siguientes módulos funcionales

1. Inicialización de datos y procesos
2. Gestión de Estados(Modulo principal)
3. Impresora de recibos
4. Impresora de billetes de localidades y transporte
5. Gestión de tablas de billetes
6. Tarjeta chip
7. Gestión de Comunicaciones
8. Mensajería
9. Publicidad
10. Avisos Nok
11. Procesos concurrentes
12. Gestión de pantalla
13. El Estado de espera

Inicialización

1. De datos del Sistema. Se obtienen datos del Sistema.
2. De datos de la aplicación. Se inicializan datos de la aplicación :
 - Pilas
 - Lista ejecución
 - Lista de eventos
3. Menú de tipos de billetes. Generamos el menú de tipos de billetes en memoria y lo dejamos listo para presentarlo por pantalla.
4. Espera a la activación por parte del sistema. Esperamos a recibir la orden de continuar con el proceso o terminar.

Gestión de Estados

Constituye el proceso principal de la aplicación. Cubre los siguientes espacios funcionales :

- Bucle de decisión. Es un secuenciador de estados. Toma un evento y sus mensajes asociados para determinar el siguiente estado. El evento y sus mensajes serán generados por los procesos de respuesta a la generación de eventos de cada estado.

- Para cada estado:

- o Lectura y presentación de datos específicos de cada estado.

Cada estado tiene una información concreta y necesaria para el siguiente estado. Una parte de ésta información debe suministrarla el usuario. Así, le debemos mostrar aquella información necesaria y aquella que pueda ayudarle a tomar una decisión.

- o Proceso inherente a cada estado

Cada dato informativo debe ser procesado según su naturaleza y usa y da acceso a determinados recursos que, o bien ponen en marcha una acción, o generan información. Así, el acceso a una base de datos será distinto según el estado.

- o Procesamiento de la selección del usuario y demás eventos externos (comunicaciones).

Cuando se genera un evento, este evento, en general, tiene una cierta significación en la aplicación, y por tanto, es interpretado como tal.

La interpretación del evento, depende así mismo del estado, por lo que, para un mismo evento, cada estado puede generar distintos mensajes.

- Espera de eventos.

- o Genera la información para el secuenciador de estados. La aplicación consulta a los dispositivos físicos o virtuales en espera de un evento. Algunos eventos generan un mensaje asociado preciso como temporizador, botón de salir, botón de retroceder, requerimiento de terminación desde el sistema, etc.

Impresora de recibos

Se aplica a las presentaciones de datos por impresora y generación de comprobantes de la venta realizada.

Gestiona la impresión de texto y su entrega al usuario.

Tiene las siguientes funcionalidades :

- Prepara datos (lee fichero)

Toma los datos de su emplazamiento original. Si es un fichero, lo lee. Si es un recibo, reúne los datos.

- Construye el buffer a enviar a la impresora (formatea)

Formatea el buffer para definir tipos de letra y otros efectos.

- Envía el buffer a la impresora con / sin indicación al usuario

Envía el buffer a imprimir y muestra una pantalla indicando que se está imprimiendo lo solicitado.

- Ofrece el impreso al usuario con mensaje, en pantalla, de entrega

Se saca el impreso y se indica al usuario que lo retire. Si, tras un tiempo previamente definido, el usuario no lo ha retirado, se emite una secuencia de beeps y se expulsa el documento.

Impresora de billetes

Imprime los billetes pedidos. Para ello busca en la tabla TABBUS.DAT los literales a imprimir.

Gestión de la tabla de Billetes

Se aplica sobre la gestión de los tipos de billetes para generar la información necesaria al usuario y a los procesos asociados a los estados citados.

Tiene tres funcionalidades :

- o Generar información para la pantalla de tipos de billetes
- o Generar información para la ayuda y el recibo
- o Generar información para la impresora de billetes.

Tarjeta chip

Cuando se compran localidades con tarjeta chip, debe comprobarse el saldo actual de la tarjeta, y en caso de ser suficiente, descargar el consumo.

Así mismo, los números de prestación y los datos a enviar son diferentes.

Comunicaciones

Se genera un módulo que construye el buffer de las siguientes prestaciones.

- Venta de billetes
 - o Tarjeta propia débito.....0129
 - o Tarjeta propia crédito.....0130

- Libreta.....0131
- Tarjeta ajena.....0132
- Billetes anulados.....0124 (1863)

Recepción

La aplicación estará pendiente de un API del sistema que enviará la respuesta a una transacción, si la recibe.

Una vez reciba la respuesta, El gestor de comunicaciones deberá comprobar que:

- Si el primer byte es un PSS distinto de 02 se desprecia la recepción
- El primer byte debe ser 02.
- Si el resultado de la ejecución es un NOK, entonces deberá mostrar el NOK correspondiente al usuario
- Si llegan avisos al cliente o actualización de banda guardar para devolver al sistema.
- Si llega retener activador, se devuelve al sistema.

Mensajes NOK

Para los mensajes NOK por código se utilizarán las APIs de manejo de cadenas.

La cadena vendrá con caracteres '&' como separadores de línea.

Se hará, por tanto, una función genérica que formatee la línea en la pantalla.

Procesos concurrentes

La aplicación deberá estar pendiente de mensajes provenientes del sistema que la urjan a terminar. En éste caso, se mostrará un mensaje NOK al cliente y se terminará la aplicación si se puede. Es decir, se lanzará un temporizador. Si el temporizador se activa, entonces se abandonará. En todo caso se intentará terminar con el cliente.

Cada estado, en su inicio, así como algunos procesos, lanzarán temporizadores.

Las funciones de temporización y de espera de mensajes del sistema serán tareas paralelas (trheads)

Presentación de archivos de texto con Paginado

Se definen las siguientes funcionalidades :

- Abrir fichero y cargar el texto en memoria.

- Pintar n líneas de texto previo borrado del fondo.
- Gestionar texto adelante o atrás.
- Liberar la memoria ocupada.

El Estado de espera

El estado de espera es único para toda la aplicación. Su función es estar pendiente de la llegada de algún evento. Cuando llega, éste evento se envía al estado actual para su análisis y posterior tratamiento si es necesario.

Utilizaremos las funciones del KBD API para programar el estado de espera. Los siguientes eventos son tratados directamente por el estado de espera :

- EV_SALIR. Botón de salir
- EV_TIMER_USR1
1 Temporizador de menús, mensajes NOK, etc.
- EV_TIMER_USR2 Temporizador de Beeps
- EV_TIMER_ON Temporizador de host.
- EV_TIMER_FIN_APL Temporizador de terminación de aplicación.

Mensajes de Sistema

Estos mensajes son enviados por el sistema central para requerir una acción de la aplicación, Generalmente abandonar.

Se presentará un mensaje NOK al cliente y se abandonará.

Gestión de pantallas

La gestión de pantalla se basa en el API del CrtHandle. Es decir, se usan los mecanismos de acceso y manejo de pantallas definidos para él. Por tanto, se usará los "Templates" para generar zonas activas con o sin iconos.

Se usará, también, el api del kbd para acceder tanto a los eventos generados por el punteo de la pantalla como al teclado.

Estas funciones se recubrirán, generándose una librería de funciones de acceso que tenga en cuenta lo siguiente :

- Carga de templates
- Carga de screens
- Carga de iconos
- Carga de textos
- Cambiar fuente de letra

- Cambiar atributos de escritura
- Posicionar el cursor.
- Modificar un color de la paleta
- Borrar la pantalla
- Establecer color de fondo
- Establecer color de primer plano
- Activar y desactivar zonas activas
- Esperar una cadena desde el teclado.

3.2.1.1.3.2.3 Estados de Interacción

A continuación se describen los diferentes estados que forman la aplicación de localidades. Alguno de estos estados son procesos en los que la interacción es con algún dispositivo diferente de la pantalla (host, impresora, etc.)

Estos estados se dividen en tres grupos. Cada grupo posee su propio espacio funcional.

- De usuario

Está definido por :

- Estado grafico
 - o Titulo
 - o Opciones selectivas
- Barra de navegación (salir, retroceder ,mas opciones, información, ayuda, continuar, índice).
- Espera de selección por el usuario.

- De comunicaciones

Está definido por :

- Procesos de comunicación con el host
 - Procesos de desglose de la información recibida
 - Tratamiento de la información recibida (Gestión de asientos)
- De impresión de información, comprobantes y tickets

- Estado gráfico
 - o Título
 - o Si es impresora Indicador de proceso
 - o Si es pantalla icono contenedor del texto
 - o Barra de navegación
- Espera de terminación de impresora si lo es
- Espera de la selección del usuario

Inicialización

La aplicación se inicia en segundo plano. Construye el menú de billetes y espera la orden de activación por parte del Sistema.

Tipos de billetes

El primer menú se construye en la inicialización.

- TABLAS :

- función OpcBill()
- Función PopcAgru() desapila una opción
- Apila hasta ocho registros en la pila Bill de la tabla de agrupaciones.

- Título : "Venta de Billetes"

Puede tener más de una página.

Cuando pase al siguiente nivel debe conservar la página actual.

- Opciones :

- Generales cortas (8)
- Salir
- info.
- Mas opciones (si hay)
- Campo de entrada de caracteres numéricos para pedir la cantidad

Envío de la compra.

Envío de la prestación 0129 / 0130 / 0131 / 0132 (propia deb, propia cred., libreta, ajena)

Si el resultado de la ejecución es :

- 01 (OK) deberá recibirse un bloque 0330 con el detalle de las localidades y mostrarlas para su aceptación.
- 02 (NOK)
- Bloques 0400 o 0300 de NOK

Confección de las entradas

Se imprimen los billetes solicitados por el cliente.

Si se imprimen todas las entradas :

- Se indicará que se retiren los billetes y se preguntará si se desea comprobante de la operación.
- Si se imprimen menos billetes de los pedidos :
- Mensaje NOK0322
- Generar mensaje informativo de entrega parcial con cargo

Si no se entrega ningún billete

- NOK0321
- Envío de una anulación total

3.2.1.1.3.3 Venta de localidades

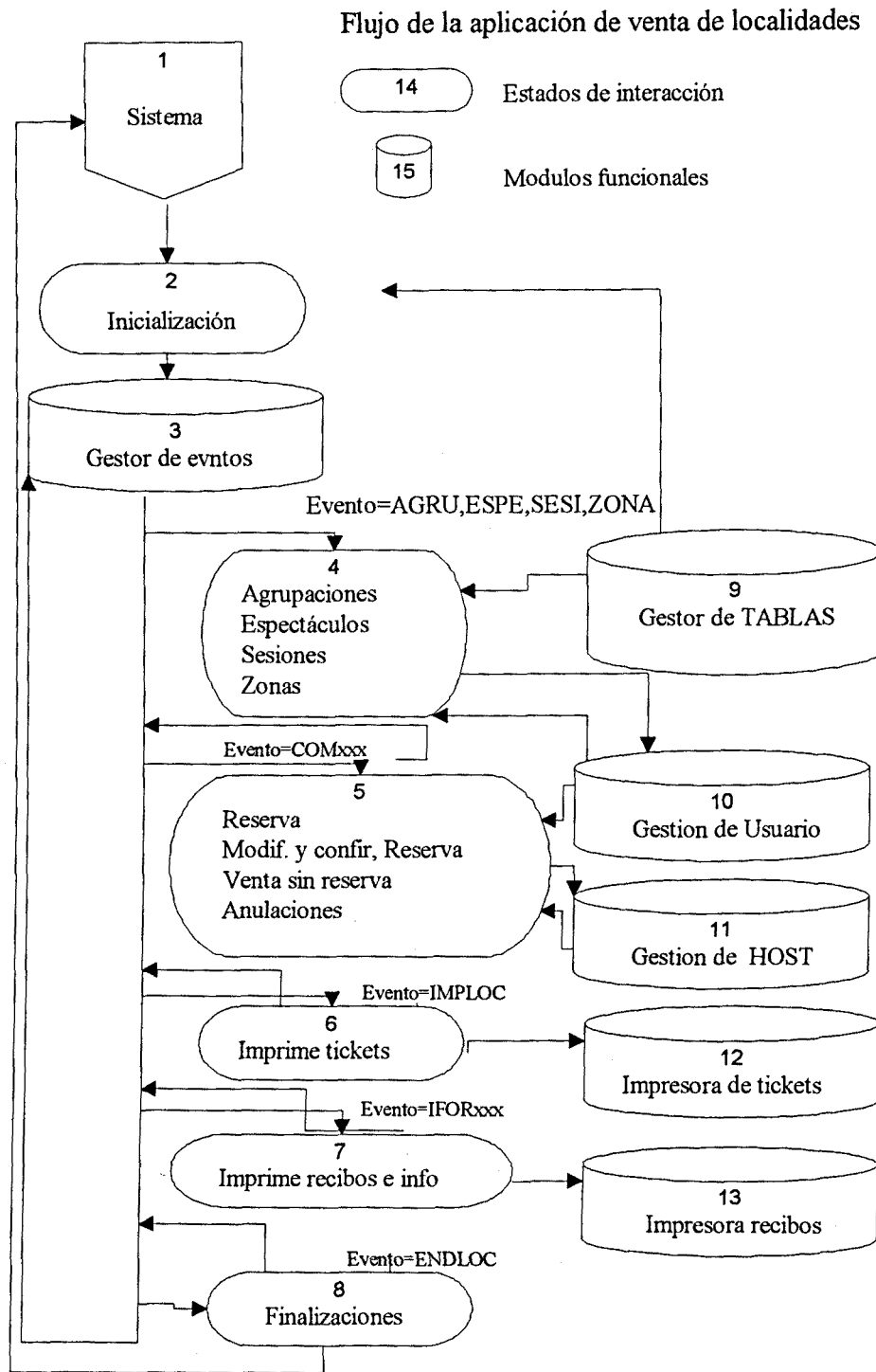


Ilustración 12. Flujo de la aplicación venta localidades. Fujitsu TAI

3.2.1.1.3.3.1 Generalidades

Características

La aplicación de localidades FTAI1 se integra en un entorno más amplio de prestaciones bancarias caracterizado por :

- Multitarea (sistema multitenhebrado).
- Procesos distribuidos.
- Comunicación a través de colas de mensajes.
- APIs de recubrimiento para gran parte del control de los dispositivos.

Así pues, la aplicación cooperará con otros procesos existentes en la máquina para llevar a termino la prestación de venta de localidades y deberá tomar de ellos los datos inherentes al sistema y deberá reportar, así mismo, los datos que se le requieran.

Por otra parte, la aplicación accederá, a través de los APIs, a los dispositivos necesarios.

Estructura

La aplicación tiene, estructuralmente, dos ámbitos distintos :

- Módulos funcionales
- Estados de interacción

Un conjunto de módulos funcionales o partes de ellos, generan un espacio funcional que será donde se desarrollará el estado de interacción. Así, un estado de interacción, la pantalla de agrupaciones, necesita de un gestor de base de datos, un gestor de pantalla, un interfaz con el usuario, etc. para ser generado.

En los esquemas anteriores, se observa como los módulos funcionales alimentan los estados de interacción. Los estados de interacción generan, a la salida, un evento y un mensaje que son tratados por el gestor de estados. Un elemento de este gestor de eventos es el bucle de decisión. Es éste un mecanismo que envía el flujo de la aplicación a un estado de interacción que es función del evento-mensaje encolado por el último estado de interacción.

En el segundo diagrama se observan con más detalle los estados de interacción, aunque agrupados por similitudes funcionales. Es decir, estados cuyo flujo de entrada de datos es básicamente desde la base de datos y desde el host, estados cuyo flujo principal es de salida de datos hacia las impresoras, hacia el sistema, etc.

Funcionalidades

La aplicación tiene como función generar la venta e impresión de tickets de localidades para los espectáculos definidos en las tablas de localidades existentes en un momento determinado en la máquina.

Para ello debe resolver las siguientes funciones :

1. Inicializar datos y procesos
2. Obtener datos de tablas
3. Presentar datos a usuario y esperar su indicación
4. Impresión por pantalla e impresora de aquellos datos de los que se quiere que el usuario tenga conocimiento.
5. Pedir datos y generar transacciones asociadas a la venta de localidades al host.
6. Imprimir los tickets para las localidades pedidas.
7. Finalizar

Estados de interacción

Para resolver las funciones anteriormente citadas, la aplicación genera una sucesión de estados donde se desarrolla un dialogo Aplicación-Usuario. Estos estados se listan, genéricamente, a continuación :

1. Inicializaciones. Se inicializan los datos y se genera el siguiente estado.
2. Gestión de agrupaciones. Se resuelve la agrupación de espectáculos deseada.
3. Gestión de espectáculos. Se resuelve el espectáculo deseado y que cumple lo anterior.
4. Gestión de sesiones. Se resuelve la sesión relativo a la selección anterior que se desea.
5. Gestión de zonas. Se resuelve la zona del recinto que cumple las especificaciones anteriores.
6. Presentación de información textual por pantalla o impresora. Se muestra información al usuario para ayudarlo en su decisión.
7. Presentación de mensajes informativos. Se indica al usuario de algún problema, indicación errónea, etc.
8. Reserva y venta de localidades. Se reservan y adjudican y cargan en cuenta las localidades pedidas. Informando de los detalles al usuario.

9. Cambio de localidades reservadas. Se modifican las localidades adjudicadas por el host
10. Generación de los tickets de las localidades. Termina el proceso entregando las localidades al usuario.
11. Reimpresión de localidades. Entrega de localidades al usuario cuando ya habían sido reservadas o bien en otra sesión de ServiCaixa, o bien por medios distintos al ServiCaixa.
12. Finalizaciones. Termina los procesos puestos en marcha por la aplicación, envía datos al sistema retorna a él.

3.2.1.1.3.3.2 Módulos funcionales

Definimos los siguientes módulos funcionales

1. Inicialización de datos y procesos
2. Gestión de Estados(Modulo principal)
3. Impresora de recibos
4. Impresora de billetes de localidades y transporte
5. Gestión de tablas de localidades, imax y cines
6. Tarjeta chip
7. Gestión de Comunicaciones
8. Mensajería
9. Publicidad
10. Avisos Nok
11. Procesos concurrentes
12. Gestión de pantalla
13. El Estado de espera

Inicialización

1. De datos del Sistema. Se obtienen datos del Sistema.
2. De datos de la aplicación. Se inicializan datos de la aplicación :
 - Pilas
 - Lista ejecución

- Lista de eventos

3. Menú de agrupaciones. Generamos el menú de agrupaciones en memoria y lo dejamos listo para presentarlo por pantalla.
4. Espera a la activación por parte del sistema. Esperamos a recibir la orden de continuar con el proceso o terminar.

Gestión de Estados

Constituye el proceso principal de la aplicación. Cubre los siguientes espacios funcionales :

- Bucle de decisión.

Es un secuenciador de estados. Toma un evento y sus mensajes asociados para determinar el siguiente estado. El evento y sus mensajes serán generados por los procesos de respuesta a la generación de eventos de cada estado.

- Para cada estado :

- Lectura y presentación de datos específicos de cada estado.

Cada estado tiene una información concreta y necesaria para el siguiente estado. Una parte de ésta información debe suministrarla el usuario. Así, le debemos mostrar aquella información necesaria y aquella que pueda ayudarle a tomar una decisión.

- Proceso inherente a cada estado

Cada dato informativo debe ser procesado según su naturaleza y usa y da acceso a determinados recursos que, o bien ponen en marcha una acción, o generan información. Así, el acceso a una base de datos. Será distinto según el estado.

- Procesamiento de la selección del usuario y demás eventos externos (comunicaciones).

Cuando se genera un evento, este evento, en general, tiene una cierta significación en la aplicación, y por tanto, es interpretado como tal.

La interpretación del evento, depende así mismo del estado, por lo que, para un mismo evento, cada estado puede generar distintos mensajes.

- Espera de eventos.

Genera la información para el secuenciador de estados. La aplicación consulta a los dispositivos físicos o virtuales en espera de un evento. Algunos eventos generan un mensaje asociado preciso como temporizador,

botón de salir, botón de retroceder, requerimiento de terminación desde el sistema, etc.

Impresora de recibos

Se aplica a las presentaciones de datos por impresora y generación de comprobantes de la venta realizada.

Gestiona la impresión de texto y su entrega al usuario.

Tiene las siguientes funcionalidades :

- Prepara datos (lee fichero)

Toma los datos de su emplazamiento original. Si es un fichero, lo lee. Si es un recibo, reúne los datos.

- Construye el buffer a enviar a la impresora (formatea)

Formatea el buffer para definir tipos de letra y otros efectos.

- Envía el buffer a la impresora con / sin indicación al usuario

Envía el buffer a imprimir y muestra una pantalla indicando que se está imprimiendo lo solicitado.

- Ofrece el impreso al usuario con mensaje, en pantalla, de entrega

Se saca el impreso y se indica al usuario que lo retire. Si, tras un tiempo previamente definido, el usuario no lo ha retirado, se emite una secuencia de beeps y se expulsa el documento.

Impresora de tickets

Imprime los tickets de las localidades solicitadas. Para ello busca en la tabla TABTICLO.DAT las definiciones de los datos a imprimir.

Gestión de las tablas de localidades, Imax y Cines

Se aplica sobre la gestión de agrupaciones, espectáculos, sesiones y zonas para generar la información necesaria al usuario y a los procesos asociados a los estados citados.

Estas tablas conforman la Base De Datos (BDD) de localidades.

El acceso a la BDD tiene las siguientes funcionalidades :

1. Inicializar tablas y crear índices.
2. Obtener las series de registros que conformarán un menú de opciones.
3. Obtener los datos asociados a un elemento de un menú.

4. Actualizar la disponibilidad según datos recibidos desde el host.
5. Obtención de datos relacionados con la selección de un elemento del menú
6. Obtención de los datos para la confección del ticket.

Tarjeta chip

Cuando se compran localidades con tarjeta chip, debe comprobarse el saldo actual de la tarjeta, y en caso de ser suficiente, descargar el consumo.

Así mismo, los números de prestación y los datos a enviar son diferentes.

Comunicaciones

Se genera un módulo que construye el buffer de las siguientes prestaciones.

Reserva..... 0090

Cambio de reserva.....0158

Confirmación de la reserva

1.Tarjeta propia débito.....0091

2.Tarjeta propia crédito.....0096

3.Libreta.....0094

4.Tarjeta ajena.....0101

Entrega parcial.....0093

Petición de compra sin reserva

1.Tarjeta propia débito.....0092

2.Tarjeta propia crédito.....0097

3.Libreta.....0095

4.Tarjeta ajena.....0102

Anulación On-Line.....

Billetes anulados.....0124 (1863)

Recepción

La aplicación estará pendiente de un API del sistema que enviará la respuesta a una transacción, si la recibe.

Una vez reciba la respuesta, El gestor de comunicaciones deberá comprobar que:

Si el primer byte es un PSS distinto de 02 se desprecia la recepción

El primer byte debe ser 02.

Si el resultado de la ejecución es un NOK, entonces deberá mostrar el NOK correspondiente al usuario

Si llegan avisos al cliente o actualización de banda guardar para devolver al sistema.

Si llega retener activador, se devuelve al sistema.

Mensajes NOK

Para los mensajes NOK por código se utilizarán las APIs de manejo de cadenas.

La cadena vendrá con caracteres '&' como separadores de línea.

Se hará, por tanto, una función genérica que formatee la línea en la pantalla.

Procesos concurrentes

La aplicación deberá estar pendiente de mensajes provenientes del sistema que la urjan a terminar. En éste caso, se mostrará un mensaje NOK al cliente y se terminará la aplicación si se puede. Es decir, se lanzará un temporizador. Si el temporizador se activa, entonces se abandonará. En todo caso se intentará terminar con el cliente.

Cada estado, en su inicio, así como algunos procesos, lanzarán temporizadores.

Las funciones de temporización y de espera de mensajes del sistema serán tareas paralelas (threads)

Presentación de archivos de texto con paginado

Tendrá las siguientes funcionalidades :

Abrir fichero y cargar el texto en memoria.

Pintar n líneas de texto previo borrado del fondo.

Gestionar texto adelante o atrás.

Liberar la memoria ocupada.

El Estado de espera

El estado de espera es único para toda la aplicación. Su función es estar pendiente de la llegada de algún evento. Cuando llega, éste evento se envía al estado actual para su análisis y posterior tratamiento si es necesario.

Utilizaremos las funciones del KBD API para programar el estado de espera. Los siguientes eventos son tratados directamente por el estado de espera :

EV_SALIR	Botón de salir
EV_TIMER_USR1	Temporizador de menús, mensajes NOK, etc.
EV_TIMER_USR2	Temporizador de menús 2 (Duración de beeps)
EV_TIMER_ON	Temporizador de host.
EV_TIMER_FIN_APL	Temporizador de terminación de aplicación.

Mensajes de Sistema

Estos mensajes son enviados por el sistema central para requerir una acción de la aplicación, Generalmente abandonar.

Se presentará un mensaje NOK al cliente y se abandonará.

Gestión de pantallas

La gestión de pantalla se basa en el API del CrtHandle. Es decir, se usan los mecanismos de acceso y manejo de pantallas definidos para él. Por tanto, se usará los "Templates" para generar zonas activas con o sin iconos.

Se usará, también, el api del kbd para acceder tanto a los eventos generados por el punteo de la pantalla como al teclado.

Estas funciones se recubrirán, generándose una librería de funciones de acceso que tenga en cuenta lo siguiente :

- Carga de templates
- Carga de screens
- Carga de iconos
- Carga de textos
- Cambiar fuente de letra
- Cambiar atributos de escritura
- Posicionar el cursor.
- Modificar un color de la paleta
- Borrar la pantalla
- Establecer color de fondo
- Establecer color de primer plano

Activar y desactivar zonas activas

Esperar una cadena desde el teclado.

3.2.1.1.3.3.3 Estados de Interacción

A continuación se describen los diferentes estados que forman la aplicación de localidades. Alguno de estos estados son procesos en los que la interacción es con algún dispositivo diferente de la pantalla (host, impresora, etc.)

Estos estados se dividen en tres grupos. Cada grupo posee su propio espacio funcional.

1- De usuario

Está definido por :

- Estado gráfico
- Título
- Opciones selectivas
- Barra de navegación (salir, retroceder ,mas opciones, información, ayuda, continuar, índice).
- Espera de selección por el usuario.

2 - De comunicaciones

Está definido por :

- Procesos de comunicación con el host
- Procesos de desglose de la información recibida
- Tratamiento de la información recibida (Gestión de asientos)

3 - De impresión de información, comprobantes y tickets

Está definido por :

- Estado gráfico
- Título
- Si es impresora Indicador de proceso
- Si es pantalla icono contenedor del texto
- Barra de navegación

- Espera de terminación de impresora si lo es
- Espera de la selección del usuario

Inicialización

La aplicación se inicia en segundo plano. Construye el menú de agrupaciones y espera la orden de activación por parte del Sistema.

Agrupaciones

El primer menú se construye en la inicialización.

Hay dos versiones :

- o Mas opciones (Mismo nivel)
- o Subagrupación (Siguiete nivel)
- TABLAS :
 - o función OpcAgruMis()
 - o función OpcAgruSub();
 - o Función PopcAgru() desapila una opción
 - o Apila hasta ocho registros en la pila AGR de la tabla de agrupaciones.
- Titulo : "Venta de entradas"
 - o Puede tener más de una página.
 - o Cuando pase al siguiente nivel debe conservar la página actual.
- Opciones :
 - o Generales (8)
 - o Salir
 - o Reimpresión (inicialmente) / Retro (en posteriores pantallas)
 - o Índice espectáculos (si existe) o info. (si existe. PRIORIDAD)
 - o Mas opciones (si hay)

Si hay un fichero llamado "TXT"+<CODAGR>+".TXT" es que hay info.

Proceso

Se crearán tantas pilas como subagrupaciones se pidan. Cuando se retroceda, estas pilas, al volver al anterior se destruirán.

Espectáculo por código. Se usa para espectáculos de test.

Se accede tocando en la esquina superior izquierda de la pantalla.. Este estado se utiliza para pedir espectáculos de pruebas. Estos espectáculos empiezan por 'ZZZ' y tienen el código de agrupación a blancos.

- Opciones :
 - o Campo de entrada alfanumérico de 5 caracteres (cod. espectáculo)
 - o Salir
 - o Continuar (Validación Espectáculo)

Índice de espectáculos

Se presenta una lista de espectáculos disponibles.

Reimpresión

Inicia el flujo de reimpresión de tickets.

Cuando se obtiene de host los datos de espectáculo, sesion, zona, etc., procederemos a imprimir los tickets.

- Opciones
 - o Campo de entrada de datos para número de referencia (10 caracteres)
 - o Continuar
 - o Salir
- TABLAS :
 - o Funciones OpcEspe(), OpcSese(), OpcZona()
 - o El host el espectáculo, sesión, zona a la que corresponde el número de referencia.

Espectáculos

Genera el menú de espectáculos en función de un código de agrupación.

- Titulo
 - o Nombre de la agrupación escogida
- TABLAS :
 - o función OpcEspe();
 - o función POpcEspe()

- Opciones
 - o hasta 8 opciones Generales
 - o Salir
 - o Retroceder
 - o Mas Opciones
 - o Info (si hay)
- Varios
 - o Si un espectáculo está agotado, se sustituirá el literal 3 por : "****
Localidades agotadas *****"

Todos los tipos de rep. excepto abonos (IMAX)

- TABLAS :
 - o función XOpcEspeT();
- Opciones
 - o hasta 8 Generales de tres líneas de código
 - o Salir
 - o Retroceder
 - o Mas Opciones
 - o Info (si hay)
- Proceso
 - o Se toman todos los códigos de agrupación de todos los tipos de representación excepto abonos ('>'A') y se actualiza CodAgr[n].
 - o Se buscan los espectáculos con sesiones a la venta de los códigos de agrupación antes mencionados.

Espectáculos de un tipo de Representación

Es igual que (6) Pero teniendo en cuenta que se escoge por tipo de representación. Esto implica que en la validación de sesiones interviene el tipo de representación.

Calendario

Los espectáculos cuyo campo periodicidad sea distinto de blancos acceden, después de la selección del espectáculo, a éste estado.

Todas las pantallas de IMAX convergen en esta.

Se muestra el icono correspondiente al primer mes permitido por la tabla TABTICIX.DAT Siempre que tenga sesiones a la venta. Cada DIA con sesiones a la venta esta activo. Los días sin sesiones a la venta están inactivos.

- TABLAS :
 - o Se accede a la tabla TABTICIX.DAT para retornar un array con la disponibilidad de los días del mes, el mes a mostrar y el máximo número de meses a mostrar.
- Titulo
 - o "Venta de entradas" ó <Título del espectáculo>
 - o <Título del mes presentado>
- Opciones :
 - o Días con sesiones a la venta.
 - o Salir
 - o Mes anterior
 - o Mes siguiente
 - o Texto explicativo de las celdas del mes.

Cines. Selección de fecha

- Opciones
 - o Calendario de las cuatro semanas siguientes al día actual. Si hay cambio de mes se pondrán los dos títulos.
 - o Opción Cines
 - o Opción Películas

Validación de espectáculo

Tipo PROCESO

- TABLAS :
 - o ValidaEspe()
 - o Valida un espectáculo y carga los datos asociados al espectáculo.
 - o Va a presentar la pantalla de información si tiene.

Sesiones

Si TipoSesion=1 saltamos a Zonas

- TABLAS :
 - o OpcSesi();
 - o POpSesi();
- Titulo
 - o <Literal corto> de la TABTICGE.
- Opciones
 - o Hasta 8 Generales de dos líneas de texto
 - o Salir
 - o Retroceder
 - o Mas Opciones
 - o Info (si hay, si no muestra la del espectáculo)
- Varios
 - o Si una sesión está agotada, se sustituirá el literal 3 por : "*****
Sesión agotada *****"

Validación sesión

- TABLAS :
 - o ValSesi();
 - o Valida una sesión
- Varios
 - o Va a presentar la pantalla de información de la sesión si la tiene.

Gestión Zonas

Si el campo Grafico del recinto es 0 Va a entrada del número de localidades

- TABLAS :
 - o TABSU_01
 - o Acceso a tabla de zonas, graftic.dat y tabticsu
- Titulo
 - o <Literal corto> de la TABTICGE
- Opciones :

- Salir
 - Sesiones
 - Espectáculos (solo 1 nivel)
 - Retroceder (+1 nivel)
 - Ayuda (si existe info de esp. o sesión)
 - Opciones de zonas (hasta 8)
 - Tienen colores distintos. Cada color representa la zona del mismo color en el gráfico. En las tablas, las opciones se codifican según su número de color, (índice de la paleta).
 - Si la zona se subdivide en zonas la opción contendrá como primer literal :
"VARIAS ZONAS"
Si no el literal de la TABTICSU.DAT
 - El segundo literal podrá tener :
 - De Z.ZZ9 a Z.ZZ9 (rango de precios si se subdivide en zonas)
 - De Z.ZZ9 Ptas. Si es una zona final se presenta el precio
 - AGOTADAS. Si las localidades de ésta zona están agotadas.
 - Descripción de la sesión (los tres literales de descripción) debajo del gráfico.
 - Gráficos de zonas
 - El primer gráfico será el del icono ZZZZ.PCC siendo ZZZZ el campo <Id gráfico 1 nivel> de la tabla GRAFTIC.DAT.
- Varios :
- Se continua cuando se selecciona una zona final.

Entrada del número de localidades

- Titulo
- <Literal corto> de la TABTICGE.
Literal con el número máximo y mínimo de localidades (total) a pedir.

Si TipoEspectador>0 se presentan tantos campos de entrada de datos (2 dígitos) como tipos de espectador estén definidos.

- Opciones
 - o Campos de entrada de datos.
 - o Retroceder
 - o Salir
 - o Ayuda (si existe info esp. o info ses.)
 - o Continuar

Si hay reserva, petición de la reserva.

- PROCESO
 - o Se ejecuta cuando <Indicación de reserva> en la TABTICSE es mayor que 0
 - o Prestación 0090

Recepción de la reserva.

- Si Resultado de la Ejecución en el host es :
 - o 01 (OK) los posibles bloques a recibir son :
 - 0330 (detalle de las localidades reservadas)
 - 1322 (Bloque de matriz de asientos)
 - 1323 (Bloque de disponibilidad de zonas alternativas)
 - Disp 0 Existen reservas pendientes de confirmación.
 - Disp 01 a 98 : número de asientos disponibles.
 - Disp 99 : Zona agotada.
 - o 02 (NOK)
 - Puede recibirse un bloque 1323
 - o Bloques 0400 o 0300 de NOK
 - o 10 (NOK) Localidades agotadas para el espectáculo
 - o 11 (NOK) Localidades agotadas en la sesión
 - o 12 (NOK) Localidades agotadas en la zona
 - Puede recibirse un bloque 1323

- 13 (NOK) No hay suficientes localidades en la zona
 - Puede recibirse un bloque 132
 - bloque 0400 o 0300 de NOK
 - Si el aforo tiene más de una zona, volver a zonas. En caso contrario volver a sesiones.

- Selección de las butacas.

- Opciones

- descripción de los tipos de iconos
- Mensaje indicando que la reserva es la mejor.
- Salir (se anula la reserva en curso)
- Cambio de selección si :
- No se ha pulsado tres veces desde la última reserva aceptada por el host
- El bloque 1322 indica que la reserva es modificable.

- Aceptar selección

Envío de la modificación de la reserva

- Proceso

- Se podrá seleccionar las zonas libres, entre las cuales quiere que el sistema le asigne el número de localidades solicitadas anteriormente. El número de zonas que podrá indicar será el siguiente :
- Hasta tres zonas de asientos
- Si ha solicitado menos de 9 localidades, 4 localidades más de las que ha indicado.
- Si ha seleccionado más de 8 localidades, un 50 % más de localidades más de las que ha indicado (redondeando a la unidad superior)

- Opciones

- Instrucciones para modificar la selección
- Salir (anulación de la reserva)
- Selección inicial (vuelve al paso anterior)
- Aceptar selección

Envío al host de la petición de cambio de reserva :

Prestación 0158

La aplicación de TIK al host establece unos máximos de modificación de una misma petición :

- Dos cuando TTS las pueda realizar
- Máximo de tres intentos.

La respuesta será :

- OK 01 Contendrá una nueva matriz de asientos
- NOK 02 Se anula la reserva actual y se finaliza la prestación
- NOK 14 se ha producido un cambio de disponibilidades y se envía una nueva matriz de asientos.

presentación por pantalla de los datos de las localidades

Se muestran los datos de las localidades reservadas para que el usuario las acepte o no.

- Opciones
 - o Icono de presentación textual.
 - o Opciones de SI/NO
- Varios
 - o Si el usuario no acepta las localidades se enviará una anulación de la reserva y se volverá a zonas /sesiones / espectáculos.

Petición de crédito / débito si la tarjeta es propia / crédito

Si el activador es tarjeta propia a crédito, debe pedirse al cliente que seleccione en que modo quiere operar.

- Titulo
 - o <Literal corto> de la TABTICGE
- Opciones
 - o Opciones de SI /NO (crédito / débito)

Envío de la confirmación de la reserva

Envío de la prestación 0091 / 0096 / 0094 / 0101 (propia deb, propia cred., libreta, ajena)

Envío de la compra.

Envío de la prestación 0092 / 0097 / 0095 / 0102 (propia deb, propia cred., libreta, ajena)

Si el resultado de la ejecución es :

- 01 (OK) deberá recibirse un bloque 0330 con el detalle de las localidades y mostrarlas para su aceptación.
- 10 (NOK) Localidades agotadas para el espectáculo
- 11 (NOK) Localidades agotadas para la sesión
- 12 (NOK) Localidades agotadas para la zona

Confección de las entradas

Se imprimen las entradas solicitadas por el cliente.

Se imprimen en el idioma del campo Idioma de la TABTICSE

Si se imprimen todas las entradas :

- Se indicará que se retiren las localidades y se preguntará si se desea comprobante de la operación.

Si se imprimen menos entradas de las pedidas :

- Mensaje NOK0322
- Generar mensaje informativo de entrega parcial Prestación 0093
- Comprobante donde se especifican las localidades solicitadas y entregadas y donde debe ir a recogerlas.

Si no se entrega ninguna entrada :

- NOK0321
- Envío de una anulación total

3.2.1.1.4 Recursos de OS2

Recursos propios

Hebras y semáforos

Las hebras y los semáforos permiten la ejecución paralela y sincronizada de partes del código.

La unidad de ejecución en multitarea es la hebra, lo que corresponde a una función de 'C'.

La Hebra se lanza con la función :

```
_beginthread()
```

del compilador c-6.12 o la función

```
DosCreateThread()
```

del Api de Os2.

Cuando finaliza la función, automáticamente termina la hebra.

El uso de semáforos permite sincronizar el flujo dentro de cada tarea con las demás.

Memoria compartida

Un segmento de memoria compartida se crea con las funciones :

```
DosAllocShrSeg(tamaño, nombre, selector);
```

El proceso hijo toma el selector a éste segmento con :

```
DosGetShrSeg(nombre, selector)
```

- Procesos

Un proceso hijo se ejecutará asincrónicamente. Si el proceso padre debe esperar a que termine el hijo para continuar, utilizará la función

```
DosCWait();
```

El proceso padre puede eliminar los procesos hijos llamando a

```
DosKillProcess();
```

Se utilizará el área de DG para pasar datos al proceso hijo. En general crearemos variables con nombres tales como : PARAML1,PARAML2, PARAMC1,PARAMC2..

El tipo de información devuelta por el proceso hijo depende del proceso específico lanzado por el proceso padre. De aquí que no es necesaria una estructura genérica de devolución de parámetros. Cada proceso sabe que datos debe buscar. En general, excepto en casos como los de comunicaciones, será suficiente con un código de retorno y un parámetro adicional.

Los procesos hijos se ejecutan con la función :

```
DosExecPgm(char far * error,  
            unsigned tamaño,
```

```
unsigned modo,  
char far * args,  
char far *envs,  
RESULTCODES Result,  
char far * nombre) ;
```

Donde mode, en general, será de ejecución asíncrona y salvando el código de terminación del hijo.

Ya que un proceso hijo puede finalizar a instancias del proceso padre, el proceso hijo instalará una función de salida a la lista de terminaciones donde se cerrarán los recursos que haya utilizado el proceso hijo. Esto se hace con la función:

DosExitList(); Pueden ejecutarse procesos síncronos o asíncronos. En éste último caso debe utilizarse semáforos para controlar el flujo entre los procesos.

Compilación

Parámetros de compilación para un módulo EXE con c run time en DLL

Opciones compilación: /c /MD /Aw /Gs /j /Oi /W4 /Zp /Ot /OI /FPa

Parámetros de compilación para un módulo DLL con c run time en DLL

Opciones compilación: /c /MD /Alfu /Gs /j /Oi /W4 /Zp /Ot /OI /FPa

Parámetros de compilación para un módulo EXE con c run time estática

Opciones compilación: /c /MT /Aw /Gs /j /Oi /W4 /Zp /Ot /OI /FPa

Parámetros de compilación para un módulo DLL con c run time estática

Opciones compilación: /c /ML /Alfw /G2 /Gs /Oi /W4 /Zp /Ot /OI /FPa

Opciones de enlazado: /NOI /STACK:8192 /BATCH /FAR /PMTYPE:VIO
/NOD:LLIBCA

Si se desea compilar para CodeView se añadirá: /Zi /Od /Or y eliminar /Oi

Librerías :

Exe con C run time en DLL: crtexe.obj y crtlib.lib y os2.lib

Exe con C estático: llibcmt.lib y os2.lib

Dll con C run time en DLL: crtdll.obj y crtlib.lib y os2.lib

Dll con C estático: llibcdll.lib y os2.lib

Librerías externas:

crtapi.lib

kbdapi.lib

Recursos Ajenos: Templates, screens, iconos, fuentes

ATM.INI

Define datos para el sistema CRT.

Se definen de donde se toman los recursos como templates, iconos, fuentes.

Templates

Los templates son definiciones de áreas activas con iconos de activo/seleccionado/inactivo.

o sin ellos.

Se encuentran en el directorio \4CX\OSCR\TEMPLATE

Para cambiarlo basta con compilarlo:

CTEM TEMPLATE.MK

Y dejar el TEMPLATE.BIN en?

Para cargar un template basta con llamar a:

TchSetTemplate(Numero)

Para activar una zona activa basta con llamar a:

activte_key(Codigo)

Donde código corresponde a un código de scan del teclado. (Puede tomar valores superiores para evitar interferencias).

Estos códigos se definen en el template.

Screens

Las screens son ficheros que empaquetan diferentes funciones graficas. Se cargan con la función display_screen del CRT_API.

Se crean con la utilidad MPGSC.EXE.

Iconos

Los iconos deben compilarse como recurso para crear la librería

ICONSLIB.DLL

La librería de definición está en \4CX\OSCR\ICON

Para compilar, ejecutar:

MAKE ICOSNSLIB.MK

Y poner la DLL en c:\4CX\ICSTM\DLL

3.2.1.1.5 Organización de la aplicación y sus datos.

Módulos funcionales :

- Gestión de pantallas (Modulo principal)
- Impresora de recibos
- Impresora de billetes de localidades y transporte
- Gestión de tablas de localidades, imax y cines
- Gestión de tablas de billetes de transporte
- Gestión del calendario
- Gestión de butacas
- Dispensación de tickets
- Tarjeta chip
- Comunicaciones
- Recibos
- Mensajería

Estos módulos se implementarán como sigue :

Modulo	tipo
Gestor pantallas	EXE
Impresora de billetes y loc.	DLL
Gestión de tablas	DLL
Calendario	función
butacas	función

Chip	función
comunicaciones	función
recibos	función
Mensajería	función

Asociado a la gestión de tablas, se creará un procedimiento para gestionar una pila con los datos de cada pantalla de menú cuando sea necesario avanzar y retroceder entre varias pantallas. El gestor de pilas debe poder gestionar varias pilas a la vez.

Modulo principal y subordinados

GLOC.C	Contiene el bucle de decisión y el main (MODULO PRINCIPAL)
ILOC.C	Funciones de inicialización
CRT.C	Funciones de pantallas
UTIL.C	Utilidades varias
RECIBO.C	construye los recibos.
COMUNI.C	Gestiona buffers de comunicaciones
TACHIP.C	Gestiona tarjeta chip.
CALEND.C	Gestiona el calendario
BUTACA.C	Gestiona las butacas
MENSAJ.C	Gestiona mensajes NOK y de PROCESO
MENSOS.C	Gestiona Mensajes desde y al sistema

Estos módulos se entienden como módulos principales de su respectivo espacio funcional. Pueden tener, naturalmente, módulos subordinados. En este caso deberán tener el mismo nombre pero añadiendo dos cifras :

CALEND01.C

CALEND02.C

Los includes seguirán el mismo criterio.

DLLs

Las siguientes son las DLL que apoyarán el modulo principal. Lo son porque se prevé su uso en diferentes procesos (LOCALIDADES y BILLETES) o por necesidad de estructuración (TABLAS)

GesLoc.DLL	Gestión de las tablas de localidades
GesTPU.DLL	Gestión del tiquetrón

3.2.1.1.6 Descripción del proceso Gestión de pantallas

Proceso principal : Gestión de pantallas

Constituye el proceso principal de la aplicación.

Funcionalidades	<p>Inicializaciones y comprobaciones diversas. (memoria compartida, dispositivos, etc.)</p> <p>Bucle de decisión.</p> <p>Lectura y presentación de datos específicos de cada pantalla.</p> <p>proceso inherente a cada pantalla.</p> <p>Procesamiento de la selección del usuario y demás eventos externos (comunicaciones)</p> <p>Interacción con el usuario.</p>
Descripción	<p>Procesos iniciales en BACK</p> <p>Inicializaciones globales de datos. (memoria compartida)</p> <p>Inicializaciones referentes a datos recibidos del llmador.</p> <p>Inicializaciones de tablas. (si son necesarias)</p> <p>Construir el menú de agrupaciones.</p> <p>Colocar en EP (Estructura de pantalla) mensaje de llamada a PROC0001</p> <p>Esperar a Activación para entrar en el menú de agrupaciones.</p> <p>Función principal : bucle de decisión :</p>

Una estructura EP contiene el evento y datos asociados actual. Un bucle de decisión gestionará este evento.

En esta estructura la recogida del evento está dentro de las funciones hijas de los bucles esto es, cada pantalla gestionará su propio bucle de espera. Si se genera un evento que implica un cambio de decisión (por ej. el usuario presiona la tecla de salir) se vuelve al bucle para que sea procesado.

```
WHILE ( EP->evento )
{
    switch ( EP->evento )
    {
        case M_PANTALLA :
            switch( EP->param1 )
            {
                case PROC0001 :
                    leeGraf(EP->param1);
                    ExecProc0001();
                case .....
            }
        }
    }
```

Tareas principales :

Presentación de gráficos.

Se crearán tablas (en disco o en código) para describir los aspectos genéricos de la pantalla :

Screens para cargar aquellos gráficos y textos fijos de la pantalla.

gráficos y textos independientes.

Timers de usuario

	<p style="text-align: center;">Tipo de pantalla : MENU o PROCESO</p> <p>Se crearán las siguientes funciones gráficas basándose en el display_data().</p> <p style="padding-left: 40px;">SetIco(Icono,x,y)</p> <p style="padding-left: 40px;">SetTxt(Texto,x,y,fuente,color)</p> <p style="padding-left: 40px;">Set2Txt(Texto1,Texto2,x,y,fuente,color,interl);</p> <p style="padding-left: 40px;">Set3Txt(Texto1,Texto2,Texto3,x,y,fuente,color,interl);</p> <p style="padding-left: 40px;">GotoXY(X,Y);</p> <p>Presentación de menús.</p> <p>A partir de las estructuras de DG que habrán sido actualizadas por el proceso hijo correspondiente, se muestran los menús en pantalla.</p>
--	--

3.2.1.1.7 Módulo de gestión de pantalla

Las funciones para realizar la gestión de pantalla (a excepción de PutTemplate) que se detallan a continuación, se basan en la utilización del comando display_data de CRT API. La definición de estas funciones se encuentra en el fichero CRT.H

El modo de utilización es el siguiente: la ejecución no se realiza en cada llamada a una función de pantalla sino que la secuencia de llamadas se va almacenando en un bbuffer que se vvisualiza con la llamada a la función OutText.

Todas las funciones consideran la posición 0,0 en la esquina superior izquierda de la pantalla, a excepción de PutTemplate que lo considera en la esquina inferior izquierda.

Display_data	
Realiza el display de datos en pantalla.	
Prototipo: void APIENTRY	Parámetros:

display_data(data_type,crt_device,display_data)	UINT data_type TRANS_DATA o MAINT_DATA UINT crt_device FRONT_CRT o REAR_CRT PCHAR display_data datos a displayar (cadena ASCII terminada en null)Retorno: Las funciones para la gestión de pantalla utilizan todos los valores TRANS_DATA y FRONT_CRT. En el tercer parámetro se envía el código ASCII o el código hexadecimal de la función a realizar.
<p>SetCur</p> <p>Posiciona el cursor en las coordenadas de pantalla indicadas. Si las coordenadas quedan fuera de rango para el tamaño actual de pantalla, la posición del cursor no variará.</p> <p>Las dimensiones de la pantalla son 16x32 para el modo de transacción y 25x80 para el modo de mantenimiento.</p>	
Prototipo: void SetCur(int x, int y)	Parámetros: int x coordenada x int y coordenada y
<p>MoveTo:</p> <p>Posiciona el cursor en las coordenadas de pantalla indicadas en píxeles. Si las coordenadas quedan fuera de rango para el tamaño actual de pantalla, la posición del cursor no variará.</p>	
Prototipo: void MoveTo(int x, int y)	Parámetros: int x coordenada x int y coordenada y
<p>AvanzaLin</p> <p>Posiciona el cursor en la siguiente fila y en la columna 0 de pantalla. Si el cursor se encuentra en la última fila, éste se posiciona en la primera.</p>	
Prototipo: void AvanzaLin(void)	
<p>ClsGr</p> <p>Borra la pantalla devolviendo los valores por defecto a los parámetros de pantalla. La posición del cursor se inicializará a 0,0.</p>	
Prototipo: void ClsGr(void)	

SetBackColor	
Asigna el color de fondo para la pantalla devolviendo el resto de parámetros a sus valores por defecto.	
Prototipo: void SetBackColor(int color)	Parámetros: int color color de fondo
SetCharColor	
Asigna el color de carácter. Todos los caracteres que se displayen después de este comando tendrán el color seleccionado.	
Prototipo: void SetCharColor(int color)	Parámetros: int color color de carácter
SetCharBackColor	
Asigna el color de fondo para los caracteres. Todos los caracteres que se displayen después de este comando tendrán el color seleccionado.	
Prototipo: void SetCharBackColor(int color)	Parámetros: int color color de fondo del carácter
SetCharSize	
Asigna el tamaño para los caracteres. Esta función sólo tendrá efecto si la fuente de letra actual está considerada como escalable en el fichero de configuración de ATM. El tamaño por defecto es 1. El tamaño 2 representa el doble de altura y el doble de anchura que el 1, y el 3 el triple.	
Prototipo: void SetCharSize(int mida)	Parámetros: int mida tamaño del carácter. El rango de valores es ['1'-'3'].
SetCharFont	
Cambia la fuente de letra. Si la fuente seleccionada no está disponible se asignará la fuente por defecto.	
Prototipo: void SetCharFont(int font)	Parámetros: int font fuente de letra
SetCharAtr	
Cambia los atributos color, medida y fuente de letra.	
Prototipo: void SetCharAtr(int backColor, int color, int size, int font)	Parámetros: int backColor color de fondo para carácter

	int color	color de carácter
	int size	medida del carácter
	int font	fuentes de letra
SetColorPaleta		
Cambia la paleta de color.		
Prototipo: void SetColorPaleta(int cn, int cv)	Parámetros: int cn índice de la paleta actual int cv color que será transformado aproximadamente a un color RGB	
SetIcon		
Visualiza un icono en la posición actual del cursor, desplegándose hacia abajo y hacia la derecha. Si el icono a partir de esa posición sobrepasa los límites de la pantalla queda recortado.		
Prototipo: void SetIcon(int ico)	Parámetros: int ico número de icono. El rango aceptado es [0-999]	
SetIconP		
Visualiza un icono en la posición indicada en píxeles dentro del rango de una resolución EGA, desplegándose hacia abajo y hacia la derecha. Si el icono a partir de esa posición sobrepasa los límites de la pantalla queda recortado.		
Prototipo: void SetIconP(int x, int y, int ico)	Parámetros: int x coordenada x. El rango es [0..639] int y coordenada y. El rango es [0..349] int ico número de icono. El rango aceptado es [0-999]	
PutTemplate		
Carga el template indicado en la pantalla. Para los templates debe considerarse la posición 0,0 en la esquina inferior izquierda de la pantalla.		
Prototipo: void PutTemplate(USHORT num)	Parámetros: USHORT num número identificador del template	

<p>CentraText</p> <p>Visualiza un texto en la pantalla, centrado en la posición actual del cursor.</p>	
<p>Prototipo:</p> <p>void CentraText(char *text)</p>	<p>Parámetros:</p> <p>char *text Texto a visualizar</p>
<p>DretaText</p> <p>Visualiza un texto en la pantalla, alineado a la derecha.</p>	
<p>Prototipo:</p> <p>void DretaText(char *text)</p>	<p>Parámetros:</p> <p>char *text Texto a visualizar</p>
<p>VideoOn</p> <p>Invierte los colores de fondo y carácter para todas las salidas de texto comprendidas entre la activación (ON) y la desactivación (OFF) del modo inverso.</p> <p>Si se activa el modo inverso y este debe permanecer activo hasta el final de la ejecución, es necesario antes de finalizar desactivar el modo inverso.</p>	
<p>Prototipo:</p> <p>void VideoOn(char *onoff)</p>	<p>Parámetros:</p> <p>char *onoff onoff puede valer "ON" o "OFF".</p>
<p>PutText</p> <p>Añade un texto al buffer de visualización.</p>	
<p>Prototipo:</p> <p>void PutText(char *text)</p>	<p>Parámetros:</p> <p>char *text Texto a visualizar</p>
<p>OutText</p> <p>Visualiza el buffer, es decir, todas las ordenes dadas con las instrucciones precedentes a excepción de PutTemplate.</p>	
<p>Prototipo:</p> <p>void OutText(void)</p>	

3.2.1.1.8 Parametrización de Estados Gráficos

En este apartado se presentan las instrucciones de alguna de las partes de la aplicación, concretamente la que se refiere a...

3.2.1.1.8.1 Iconos genéricos

ICONO	X	Y	X2	Y2	Al.	Xt	Yt	Fnt	C	Int	texto
MISSHOR	112	58	529	120	C	209	19	UNIV24	7	19	
PANTA0	224	121	421	395							
SORTIR	8	413	150	473							
TITOL	0	0	639	52	C	0	45	UNIV37	0	0	
MISSPET	16	58	421	119	C	106	19	UNIV27	7	19	

3.2.1.1.8.2 Parámetros de las pantallas de agrupaciones, espectáculos, y sesiones

Cada opción tiene tres líneas de texto. La segunda y la tercera tienen el mismo formato, por lo que sólo se indica una posición y un interlineado.

ICONO	X	Y	Alineado	Xtexto	Ytexto	Fuente de letra	Color	Interlineado
OPGENE	8	53	I	69	75	BIT8X16	0	8
			I	69	94	BIT8X8	2	14
OPGEND	433	53	D	571	75	BIT8X16	0	8
			D	571	94	BIT8X8	2	14
OPGENE	8	42	I	69	164	BIT8X16	0	8
			I	69	183	BIT8X8	2	14
OPGEND	433	142	D	571	164	BIT8X16	0	8
			D	571	183	BIT8X8	2	14
OPGENE	8	231	I	69	253	BIT8X16	0	8
			I	69	272	BIT8X8	2	14
OPGEND	433	231	D	571	253	BIT8X16	0	8
			D	571	272	BIT8X8	2	14
OPGENE	8	320	I	69	342	BIT8X16	0	8
			I	69	361	BIT8X8	2	14

OPGEND	433	320	D	571	342	BIT8X16	0	8
			D	571	361	BIT8X8	2	14

3.2.1.1.8.3 Pantalla de Crédito débito

ICONO	X	Y	X2	Y2	Al.	Xt	Yt	Fnt	C	Int	texto
OPGENE	8	231	209	300	I	60	24	UNIV24	0	19	
OPGEND	33	231	632	300	D	142	24	UNIV24	0	19	
MISSHOR											¿Desitja efectuar l'operació amb càrrec al seu compte corrent o amb càrrec al crèdit de la targeta?
PANTA0											
SORTIR											

3.2.1.1.9 Pruebas alfa

El documento que se reproduce a continuación corresponde a las plantillas utilizadas directamente por las personas que realizan las pruebas alfa de una aplicación.

Este documento puede ser utilizado directamente para la realización de las pruebas, o bien puede ser la base a partir de la cual se genera una hoja de ruta en la que se especifican datos tales como quien es el probador o cuantas veces y con que variaciones se deben realizar las pruebas en cada caso.

Proyecto: Servicaixa. Módulo de Localidades⁶²

Fecha: Marzo de 1996

Condiciones generales para las pruebas

⁶² El sistema Servicaixa al que ya nos hemos referido antes constituye una aplicación compleja y las plantillas representadas en este documento correspondientes al Módulo de Localidades corresponden a un 10% aproximadamente de las plantillas en el documento original.

Se comprueba para cada idioma, se observan las inversiones de interacción, que todos los gráficos y textos sean correctos, así como las opciones de retroceso, y que el teclado no interacciona con "templates".

En todos los retornos al Menú Principal verificar comunicaciones con sistema.

Tabla para la implementación de la hoja de ruta

Espectáculos en general:

Agrupaciones:	
Literal 3 "Localidades agotadas"	
Sin espectáculos (NOK 0312)	
Con un espectáculo	
Con varios espectáculos	
Acceso por índice	
Acceso test (zona sup. izq.)	
Probar códigos ZZZ.	
Probar otros códigos	
Probar códigos erróneos	
Operativas Especiales-Entradas Pendientes (después de Entrega Parcial)	
Activadores propios (Cuenta asociada)	
Tarjetas ajenas (PIN encifrado)	
Salir al Menú Principal	
Espectáculos	
Sin info espectáculo	
Con info espectáculo	
Calendario	
Sin calendario	
Con calendario:	
Comprobar paginados (periodicidad)	
- Meses	
- Semanas	
- Días	

Probar día inicial .	
Probar día intermedio .	
Probar día final .	
Probar año bisiesto .	
Sesiones	
Con una sesión	
Con varias sesiones	
Literal 3 "Sesión agotada" .	
Sin ayuda (sin info espectáculo) .	
Con ayuda (con info espectáculo) .	
Sin info sesión	
Con info sesión	
Zonas	
Sin zonas	
Con zonas	
Sin ayuda (sin info espec. ni sesión) .	
Con ayuda (sin info sesión, con info espec)	
Con ayuda (con info sesión) .	
Comprobar literales:	
- Varias zonas	
- Zona final	
- Rango de precio	
- Precio final	
- Agotadas	
Tipos Espectador	
Nº de localidades	
Sin ayuda (sin info espec. ni sesión) .	
Con ayuda (sin info sesión, con info	

espec)	
Con ayuda (con info sesión)	
Tipos de espectador:	
- Un Tipo	
- Dos Tipos	
- Tres Tipos	
- Cuatro Tipos	
- Cinco Tipos	
- Seis Tipos	
- Siete Tipos	
- Ocho Tipos	

Menú Imax

Información general (opción ayuda)	
Abonos	
Por películas	
Por cualquiera	
Calendario:	
Comprobar paginados (periodicidad)	
- Meses	
- Semanas	
- Días	
Probar DIA inicial	
Probar DIA intermedio	
Probar DIA final	
Probar año bisiesto	
Menú Sesiones de un DIA:	
Sin opción ayuda (Sin info espec)	
Con opción ayuda (con info espec)	
Selección:	

- Sin info sesión	
- Con info sesión	
- Sin zonas	
- Con zonas	

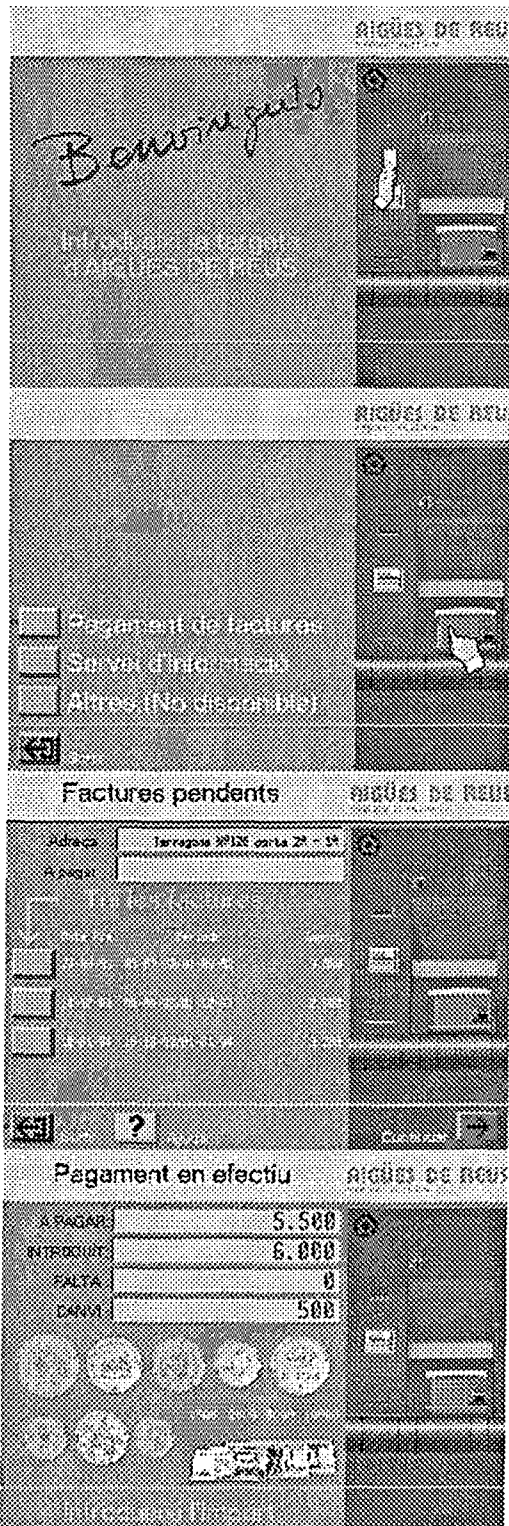
Menú Cines:

Calendario:	
Probar DIA inicial .	
Probar DIA intermedio .	
Probar DIA final .	
Probar año bisiesto .	
Acceso por cines	
Menú sesiones:	
- Sin info. Cine	
- Con info. Cine	
- Sin info. Peli.	
- Con info. Peli.	
Acceso por películas	
Menú sesiones:	
- Sin info. Peli.	
- Con info. Peli.	
- Sin zonas	
- Con zonas	
Nº localidades:	
Sin info. Cine .	
Con info. Cine .	
Sin info. Peli..	
Con info. Peli.	

Después de finalizar con las pruebas de la selección inicial, el documento continua con las pruebas en las comunicaciones con el "host", y las pruebas de impresión.

3.2.1.2 Proyecto TAI Reus

Fecha: octubre de 1997



Los sistemas transaccionales requieren la gestión simultanea de un cierto número de dispositivos. Un sistema para cobrar recibos, como por ejemplo el de la compañía de Aguas de Reus, ha de disponer de dispensadores y aceptadores de monedas y billetes además de lectores de tarjeta y ha de ser capaz de facilitar la realización de procesos interactivos sencillos y rápidos por parte del usuario. (1993)

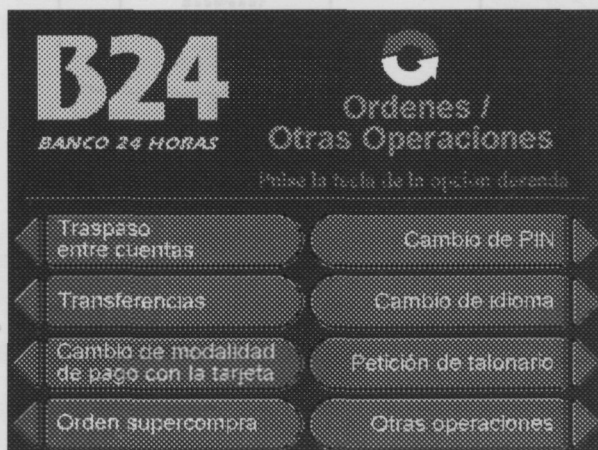
3.2.1.2.1 Valoración Gestión de Stocks. Requisitos planteados

Documento de referencia "Requerimientos a cumplir por el Software de Gestión de la Caixa".

Este documento forma parte de un estudio técnico sobre máquinas de autoservicio para Aigües de Reus, por lo tanto es un componente técnico de la Oferta Abierta.

- Entrada de existencias iniciales en cada hopper.
- Entrada de existencias iniciales en el monedero.
- Entrada de existencias iniciales en la unidad receptora de billetes.
- Entrada de existencias iniciales en el monedero.
- Entrada del punto de ruptura para cada hopper.
- Entrada del punto de ruptura para el billeteo.
- Entrada del número máximo que admite la unidad receptora de billetes.
- Entrada del número máximo de billetes que admite el monedero.
- Puesta a cero de los contadores de cada hopper y del billeteo.
- Puesta a cero de los contadores del monedero y de la unidad receptora de billetes.
- Listado de las existencias de cada hopper y del billeteo.
- Listado de las existencias del monedero y de la unidad receptora de billetes.
- Nº de errores en las devoluciones de cada hopper y del billeteo.
- Nº de errores en las entradas de monedas y de billetes.

3.2.1.3 Proyecto BBV TAI



Los terminales de autoservicio se han extendido de forma sistemática por todas las instituciones financieras. El terminal BBV, pocos años después del sistema Servicaixa, integra en un único sistema las funcionalidades de los cajeros y con las nuevas transacciones.

Fecha: abril de 1995

3.2.1.3.1 Flujo de control describiendo los estados de venta de localidades

Lógica general para el suministro de entradas a Espectáculos

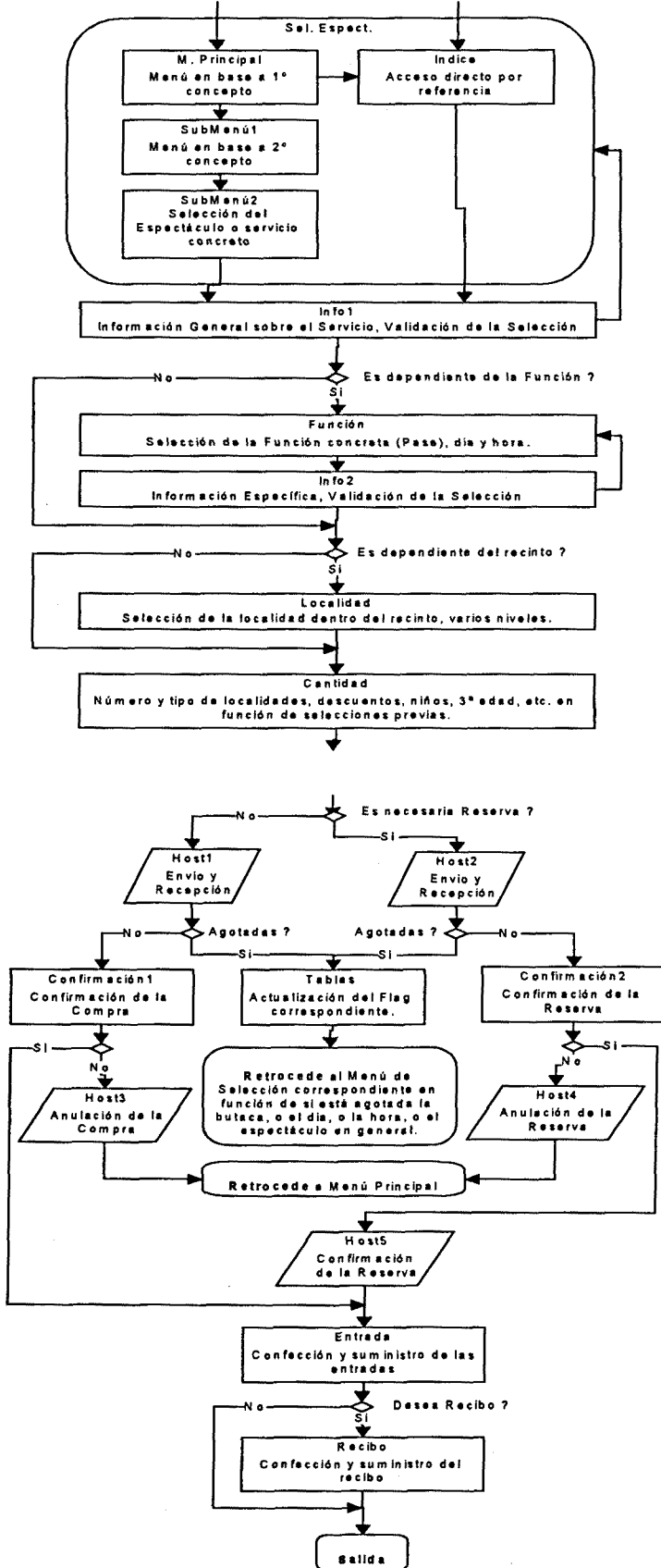


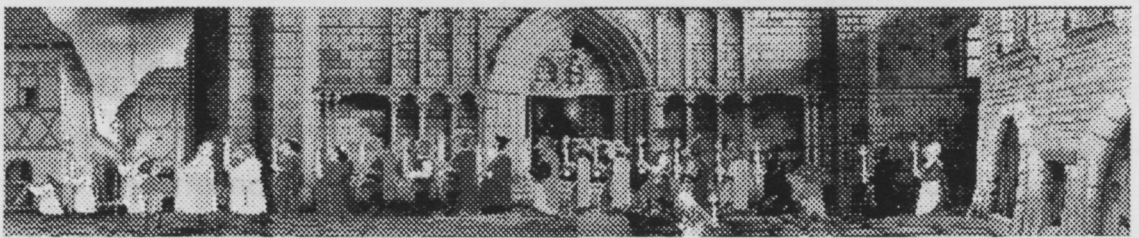
Ilustración 13. Flujo de control describiendo los estados de venta de localidades, BBV TAI

3.2.2 1.1 Juegos interactivos un presupuesto en base a un juego de referencia

3.2.2.1 Proyecto Juego de Aventura

Fecha: febrero de 1996

3.2.2.1.1 Características técnicas del juego.

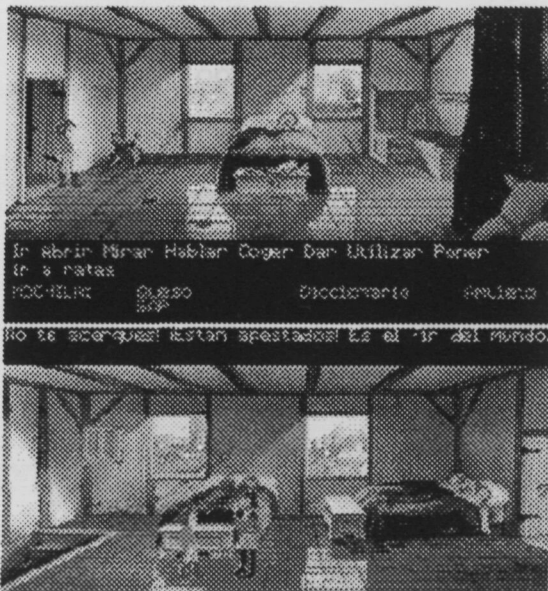


Tarjeta de Audio Sound Blaster o compatibles.

Las imágenes corresponden a capturas de pantalla del juego de aventura producido para la Fundación "la Caixa" que fue distribuido entre los escolares de toda la región junto a otros elementos como soporte informativo en la campaña contra el SIDA.

El juego tiene un carácter instructivo y se basa en el aprovechamiento del paradigma de la aventura gráfica para vehicular un conjunto de interesantes contenidos históricos sobre las epidemias.

Escenas.



aciones a 256 colores con la paleta
co
áficos de cualquier tamaño.
e las escalas, que pueden variar de
áreas por las que pueden caminar los
lica del camino más corto.
conjunto de verbos de acción para cada
maño en función de la posición dentro de

3.2.2.1.1 Generación de un presupuesto en base a un juego de referencia

Componente técnico para la Oferta Abierta

3.2.2.1.1.1 Características técnicas del juego.

El juego de aventura basado en la historia de las epidemias:

- Trabajo realizado bajo de encargo de la Fundación de la Caixa.
- Desarrollado para DOS 3.3 o superiores, Modo Real gestionando memoria extendida y/o expandida.
- Tarjeta de Audio Sound-Blaster o compatibles.
- Previstas versiones en Modo Protegido para nuevos juegos. Previsto mayor soporte para tarjetas de audio.
- Programado en Borland C++ 3.1.
- Librerías gráficas de Genus Microprogramming.
- Librerías B-Tree Filer de Turbo Power Software.
- Escenas:
 - o Los Fondos son digitalizaciones a 256 colores con la paleta optimizada para cada gráfico.
 - o Capacidad de scroll de gráficos de cualquier tamaño.
 - o Capacidad de definición de las escalas, que pueden variar de forma continua.
 - o Capacidad para definir las áreas por las que pueden caminar los objetos, búsqueda automática del camino más corto.
 - o Capacidad de definir el conjunto de verbos de acción para cada escena.
- Objetos:
 - o Cambio automático de tamaño en función de la posición dentro de la escena, escala continua.

- Pueden tener indeterminados conjuntos de actividades.
- Pueden responder o no la interacción de usuario.
- Pueden tener movimientos en bucle, temporizado, seguir un camino, o buscar el camino más corto a un punto.

3.2.2.1.1.2 **Calculo del costes de la producción de gráficos:**

Se ha tomado como referencia el juego "Indiana Jones en el templo maldito" de la empresa Lucas Arts para la realización de un cálculo de los recursos necesarios para su desarrollo de un juego que emule as funcionalidades y las reglas del mismo.

Según el siguiente desglose:

Diseño de los escenarios y personajes:

Partiendo del guión suministrado por el cliente, la Fundació de la Caixa, es necesario generar las matrices de dibujo de todas las escenas, objetos y personajes de la historia.

Estas matrices van acompañadas de la adecuada documentación gráfica y textual sobre los ambientes personajes u objetos que se han de dibujar.

Elementos de coste que se deben diferenciar

Tabla resumen de los costes de producción gráfica			
Tareas	Personal	Días de trabajo	Hipótesis de volumen
Diseño y dibujo de escenarios	Dibujante	2	150
Grafismo 2D de los escenarios	Grafista 2D	3	225
Animación de desplazamiento de personaje	Grafista 2D	0,2	48
Animación de la manipulación de objetos	Grafista 2D	0,3	31
Animaciones repetitivas:	Grafista 2D	0,3	112
Animaciones de interacción	Grafista 2D	0,2	75

Supuestos aplicados sobre el juego de Indiana Jones:

Escenarios: 75

- 75×2 días = 150 días de dibujante
- 75×3 días = 225 días de grafista 2D

Personajes: Uno principal más otro añadido cada cinco escenas

- 16 personajes $\times 3$ días = 48 días de grafista 2D

Manipulación de objetos: Dos manipulaciones por esc. pers. principal. + Una cada tres pers.

- $((75 \times 2) + 5) \times 0,2 = 31$ días de grafista 2D

Cinco animaciones repetitivas por escena:

- $5 \times 75 \times 0,3 = 112$ días de grafista 2D

Cinco animaciones de interacción por escena:

- $5 \times 75 \times 0,2 = 75$ días de grafista 2D

3.2.2.1.1.3 Cálculo del costes de la producción de Audio:

Efectos sonoros con presencia de tarjeta digitalizadora:

Elementos de coste que se deben diferenciar

Tabla resumen de los costes de producción gráfica			
Tareas	Personal	Días de trabajo	Hipótesis de volumen
Puertas que se abren o cierran, botellas que se rompen, golpes varios, etc. (días por efecto) (mínimo de 20 efectos)	Técnico de sonido	0,2	30
Efectos sonoros para altavoz PC (días por efecto)	Técnico de sonido	0,1	15
Música de fondo con presencia de tarjeta MIDI: Duración media de 10seg a partir de los cuales puede repetirse. (días por fondo)	Músico	0,2	5
Música de fondo para altavoz PC (días por fondo)	Músico	0,1	2,5

Supuestos aplicados sobre Indiana Jones:

Dos efectos por escena:

$$2 \times 75 (0,2+0,1) = 45 \text{ días de técnico de sonido}$$

Un fondo diferente cada 3 escenas:

$$75 / 3 (0,2+0,1) = 7,5 \text{ días de Músico}$$

3.2.2.1.2 Especificaciones de los elementos gráficos

3.2.2.1.2.1 Fondo

Es un gráfico formato pcx, se carga al entrar en una nueva Escena, el resto de objetos de una Escena (personajes, secuencias, etc.) se sobre imponen en modo transparente sobre el fondo.

Los fondos pueden tener varias Escenas, las transiciones entre Escenas de un mismo fondo se realizan por scroll. En principio el ancho de una Escena es igual a la Xmax del modo gráfico. Además, y también en principio, como máximo puede haber 3 Escenas en un fondo.

Las Escenas de un fondo se enumeran de izquierda a derecha, por lo tanto un fondo puede contener Escena 1; o Escena 1 y 2; o Escena 1, 2, y 3.

Reconocer el número de Escenas que posee un fondo puede hacerlo automáticamente el programa en función del tamaño del gráfico.

3.2.2.1.2.2 Descripción de Elementos Gráficos

Son las animaciones gráficas elementales, un "Objeto" que se mueve en una Escena puede usar (poseer o controlar) varios Elementos.

Composición de los Elementos:

Un conjunto "Cuadros" (frames), son dibujos bitmap, todos ellos de la misma dimensión.

El valor de Ancho (X) de un Cuadro.

El valor de Alto (Y) de un Cuadro.

El Nº total de Cuadros, valor mínimo = 1.

El "Paso":

Es un único valor en píxeles, tanto para X como para Y, de desplazamiento entre Cuadros. Este valor puede ser cero, en este caso tendremos una animación que no se desliza por pantalla.

Este valor será utilizado como base para calcular el desplazamiento real en X e Y dentro de la Escena, y así poder realizar un movimiento en cualquier dirección.

Como referencia para el Operario encargado de la edición de Elementos, este valor coincide con el desplazamiento real en horizontal o vertical.

La "Cadencia":

Todos los objetos animados de una Escena funcionan (dibujan un nuevo Cuadro) sincrónicamente con una base de tiempos.

Esta base de tiempos puede ser función del N° de Cuadros mínimo por segundo para obtener una animación adecuada, alrededor de 10C/s. Interesa un valor bajo para poder animar muchos elementos a la vez sin variar su velocidad. Interesa un valor alto para aumentar la calidad de las animaciones.

El valor Cadencia indica cuantas unidades de tiempo tienen que pasar para cambiar de Cuadro.

En el caso de N° de Cuadros = 1 el valor Cadencia no tiene significado.

La mayor velocidad se consigue con el valor de Cadencia = 1.

La "Profundidad":

Este valor se mide en "decímetros".

La función del valor Profundidad estriba en conseguir que los objetos de una Escena tengan un volumen virtual, en lo que se refiere a colisiones entre objetos.

Dado que se mide en decímetros y no en píxeles, permite cambiar la profundidad de un objeto automáticamente al cambiar de Escena. Esto quiere decir que una Escena, entre otros, debe poseer el valor de Profundidad Total (también en decímetros). Para una Escena concreta la Profundidad Total equivale a un n° de píxeles, por lo tanto será conocida la profundidad en píxeles de cada Elemento de esta Escena.

Dibujo de los Cuadros de los Elementos:

En función de la coordenada Z:

La coordenada Z (Profundidad de Escena), ES UN VALOR COMPLETAMENTE INDEPENDIENTE DEL ELEMENTO. Este valor se calcula en función de la coordenada Y que actualmente posee el Objeto y el valor Profundidad Total que posee la Escena.

Este valor se utiliza para saber que Cuadro de cada Elemento que está ejecutando un Objeto queda tapado por otro con una Z menor.

Por lo tanto el valor Z de la siguiente posición que se va a dibujar, define el orden de dibujo de cada Cuadro de cada Elemento. Mayor Z da mayor prioridad de dibujo, de esta forma los Cuadros con mayor profundidad quedarán "machacados" por los Cuadros con menor profundidad.

En función de la Base de Tiempos:

Ya se ha definido el valor Cadencia de un Elemento, sin embargo un elemento con Cadencia mayor de uno, debe dibujar algún Cuadro EN TODOS LOS TIEMPOS, de lo contrario no sería posible tratar la coordenada Z.

Supongamos tres Elementos A, B y C.

A tiene 6 Cuadros y una Cadencia de 1

B 3 Cuadros y Cadencia de 4

C 2 Cuadros y Cadencia de 2.

En un diagrama de Tiempos podría representarse así:

A1	A2	A3	A4	A5	A6
B1	B1	B1	B1	B2	B2
C1	C1	C2	C2	C1	C1
T1	T2	T3	T4	T5	T6

Clasificación de los Elementos:

En función de los parámetros Paso y N° de Cuadros es posible definir 4 tipos de Elementos:

	Paso	0,00	N
N° Cuadro			
1,00		Estático	Móvil
N		Secuencia Estática	Secuencia Móvil

3.2.2.1.3 Normas de edición de contenidos

Este archivo tiene las pruebas y los ejemplos de la edición de textos con Amipro para el proyecto SIDA.

Explicación del Funcionamiento:

Para generar los archivos debe existir el archivo SIDA.STY en el subdir. ESTILOS de Amipro. Al abrir un archivo nuevo se le asocia esta hoja de estilos.

Existen tres estilos de párrafo para gestionar los textos y códigos:

Código

Para escribir los códigos, máx. de 16 car. por Intro se accede a párrafos tipo Castellano, tecla de acceso rápido F2.

Convierte minúsculas en mayúsculas.

Tipo código

Castellano

Para escribir los textos en castellano, en principio pueden ser de hasta dos líneas

Tipo Castellano

Catalá

Para escribir los textos en catalán, en principio pueden ser de hasta dos líneas

Tipo Cátala

Configuración de Amipro:

Impresora especificada para el archivo

EPSON ESC/P2

En "Vista-Preferencias":

Con tabuladores y retornos

Con formato final.

Forma de generar el archivo ASCII:

Opción Guardar Como:

Archivo ASCII, con las siguientes opciones:

Con finales de línea (Amipro 3.01) y **SIN nombres de estilo**

Sin finales de línea (Amipro 3.00) y **SIN nombres de estilo**

ASCII PC de 8 bits.

Es decir que al salvar a ASCII cada idioma debe ser una única línea, **del tamaño que sea pero solo una línea.**

Ejemplo:

LJF34-343

No puedo coger este objeto.

No puc agafa aquet objete.

LJDI34343334LJGF

Oh! gracias.

Oh! merçes.

lkjdfij

Lo siguiente es la prueba para probar los tamaños de las frases, **NO TIENE INTERES PARA EL USUARIO-EDITOR**

0123456789123456

Con estos conjuntos monumentales gigantescos forman contraste algunas capillas aisladas que se levantan durante la dinastía XVIII, como la de Elefantina (Fig. 54), que parece presentir el sentido de las proporciones del templo griego.

Además de estos templos construidos sobre la superficie de la tierra, los egipcios labran otros excavados. Se encuentran en Ipsambul, en nubia, la región más meridional del valle, varias veces perdida y otras tantas reconquistada, y expuesta siempre a la invasión y al saqueo de los etíopes. Conservan aún el nombre griego de speos, es decir, cueva. Tanto el Speo mayor (Fig. 46) como el menor de Ipsambul tienen gigantescas

3.2.2.2 Proyecto Juego de Ajedrez

Fecha: Marzo de 1996

Este estudio se enmarca en el entorno de una Oferta Abierta, realizada para la empresa Salvat para la realización de un juego de ajedrez; incluye una descripción de los módulos software del API propuesto. También incluye un estudio de otros programas, que se puede considerar parte de Requisitos de Usuario.

3.2.2.2.1 Selección de las librerías

Se ha escogido el programa de Ajedrez suministrado, como ejemplo, con el compilador Borland Pascal 7.0

Este programa puede compilarse para:

- Windows
- DOS Modo Protegido
- DOS Modo Real (ejemplo en Modo Texto)

El programa se compone del tratamiento de usuario, y de un módulo de cálculo que controla las operaciones lógicas vinculadas directamente con el juego. Esta estructura permite modificar el entorno gráfico, las opciones, etc. En el caso de otros programas de ajedrez, y sin llegar a un acuerdo con los autores, no se puede garantizar este nivel de control sobre el producto final.

Se tomarán como referencia las funciones de este módulo:

- Control del identificador de juego:
 - function NewGame(var GameHandle: HChess): TChessError;
Asigna un identificador al juego.
 - function DisposeGame(CH: HChess): TChessError;
Libera un identificador de juego
- Control de movimientos:
 - function ParseMove(Move: PChar; var Change: TChange): TChessError;
Guarda un cambio, una sentencia de movimiento.
 - function RetractMove(CH: HChess; const Move: TMove): TChessError;

- Retracta el último movimiento. No se puede ejecutar durante un periodo de búsqueda.

function SubmitMove(CH: HChess; const Change: TChange): TChessError;

Asume un movimiento para el jugador actual, solo se puede invocar en periodos de espera, o búsqueda adelantada.

function VerifyMove(CH: HChess; const Change: TChange): TChessError;

Verifica un cambio pero no lo guarda, es una llamada legal solo en periodos de búsqueda adelantada.

- Control de Búsqueda:

function ComputerMove(CH: HChess; TimeLimit: LongInt): TChessError;

Arranca la búsqueda de un movimiento, retornará inmediatamente. Es necesario llamar a la función Think para realizar la búsqueda actual. TimeLimit es 1/18 seg.

function ForceMove(CH: HChess): TChessError;

Fuerza al ordenador a realizar un movimiento con la información que posee en este momento, el movimiento se completará en la siguiente llamada a Think, solo se puede ejecutar durante un periodo de búsqueda.

function ThinkAhead(CH: HChess): TChessError;

Arranca un periodo de búsqueda, suponiendo que el oponente escogerá la línea principal de juego, si es así, la nueva búsqueda continua donde lo dejó ThinkAhead, si no, arrancará una búsqueda nueva. Solo se puede llamar durante estado de espera.

function AbortSearch(CH: HChess): TChessError;

Aborta el periodo de búsqueda que se está realizando.

- Control del Proceso de Ajedrez:

function Think(CH: HChess; TimeLimit: LongInt; var Status: TSearchStatus): TChessError;

Realiza una búsqueda de movimiento, debería ser invocado cuando el programa de ajedrez está a la espera.

- Edición del Tablero:

No son válidas durante las búsquedas.

function SetBoard(CH: HChess; const ABoard: TBoard): TChessError;

Sustituye el tablero actual con el especificado.

```
function SetPlayer(CH: HChess; APlayer: TColor): TChessError;
```

Asigna el jugador actual.

```
function MakeChange(CH: HChess; Color: TColor; const Change:  
TChange): TChessError;
```

Realiza las siguientes modificaciones en el tablero:

Si la localización "fuente" está en blanco, la pieza es nueva, si el destino está en blanco la pieza se toma del tablero, si ninguno es blanco la pieza será movida, si el tipo de pieza no coincide con la de la posición fuente, el nuevo tipo será asignado.

— Control de estado:

```
function GetSearchStatus(CH: HChess): TSearchStatus;
```

Retorna el estado de la búsqueda de movimiento.

```
function GetChessStatus(CH: HChess; var Count: Integer): TChessStatus;
```

Retorna el estado actual del juego.

```
function MoveToStr(const Move: TMove; var Str: array of Char):  
TChessError;
```

Formatea el movimiento como una cadena de caracteres.

— Sigüientes funciones no pueden invocarse durante una búsqueda

```
function GetLastMove(CH: HChess; var Move: TMove): TChessError;
```

Retorna el último movimiento.

```
function GetHintMove(CH: HChess; var Move: TMove): TChessError;
```

Sugiere un movimiento.

```
function GetBoard(CH: HChess; var ABoard: TBoard): TChessError;
```

Retorna el estado actual del tablero, si se está realizando una búsqueda será el estado inicial.

```
function GetPlayer(CH: HChess): TColor;
```

Retorna el jugador que posee el turno.

```
function GetValidMoves(CH: HChess; Change: TChange; var Moves: array  
of TMove): TChessError;
```

Retorna la lista de movimientos válidos.

— Las sigüientes funciones se usan solo durante la búsqueda:

function GetCurrentMove(CH: HChess; var Move: TMove): TChessError;

Retorna el movimiento por el que comenzó la búsqueda.

function GetMainLine(CH: HChess; var Value: Integer; var Line: array of TMove): TChessError;

Retorna la línea de juego principal utilizada por el ordenador.

function GetNodes(CH: HChess): LongInt;

Retorna el nº de nodos procesados durante la última o actual búsqueda.

3.2.2.2 Comparación de varios programas de ajedrez comercializados actualmente para PC

La siguiente relación ha sido calificada a partir del análisis realizado con un jugador de ajedrez. Las calificaciones están realizadas sobre 10.

— Interacción

Define la calidad y versatilidad del Interface con el Usuario.

— Estrategia

Define la calidad de juego, el nº de piezas que tiene en cuenta la línea de juego del programa, las aperturas (despliegue de fichas inicial), la avidez por los gambitos (posibilidad de despistar al programa sobre las verdaderas intenciones del oponente, ofreciendo fichas fáciles de eliminar).

Programa	Autor	Interacción	Estrategia
Gambit (Kasparov)	Electronics Arts	9	9
ChessMaster 3000	The Software Toolworks	7	8
OwlChess	Borland	4	6
Chess Chanpion 2175 ELO	Oxford Softworks	5	6
Chess Windows	Free Software Foundation	3	5

Kchess	Ark Angles	4	5
Colosus Chess	Proge In	2	3
WinChess	David Norris	6	3

Comentarios sobre el programa Gambit (Kasparov)

Este programa ha obtenido calificaciones muy altas, tanto en lo relativo a estrategia, reconocimiento de aperturas, etc., como en la interacción, y en este último caso por haber resultado la "Mejor Línea de Juego" con un segundo tablero, mayor número de interacciones en primer plano, y video digital (voz digitalizada sincronizada con animaciones).

Sin embargo puede no ser la mejor solución para el caso tratado. Esto es debido a los importantísimos requisitos Hardware y Software, que dudamos sean accesibles para la mayoría de los interesados.

El programa necesita una gran cantidad de memoria expandida (2048 KB) y PRACTICAMENTE TODA la memoria convencional (590.000 bytes), esta particular distribución (para los sistemas operativos actuales), en el caso de poseer un ordenador que la soporte, requiere una configuración de arranque especial, amén de los problemas de configuración que conlleva, normalmente no será la adecuada para el resto de programas, sobre todo en entornos Windows que se pierden literalmente 2MB de memoria.

Descripción de la interacción de OwlChess

El programa está realizado en Inglés.

En primer plano:

- Un tablero de 2D con numeración de filas y columnas, correctamente resuelto.
- Un cuadro de estado con la siguiente información.
- Turno y color
- Tiempo total, y el parcial de cada jugador.
- Nº de nodos procesados y velocidad de proceso.
- Listado de todos los movimientos.
- Mejor línea de juego, listado de los mejores movimientos a realizar, según el programa.

— Valor de la línea de juego.

Accesibles a través de menús

— De juego:

— Salvar y cargar partidas.

— Modo demostración, solo juega el programa.

— Pausa / continuar

— Salida

— De movimiento:

— Ir atrás.

— Ir adelante.

— Que mueva el programa por el usuario.

— Entrada Algebraica para el siguiente movimiento.

— De Configuración:

— Modo de Juego:

— Tiempo máximo para un juego.

— Tiempo máximo para un movimiento.

— Mismo tiempo de juego que el usuario.

— Sin tiempo límite.

— Duración del tiempo del programa dedicado a juego, (relación Interacción / juego).

— Uno o Dos jugadores.

— SI/NO permite pensar al programa en tiempo de usuario.

— SI/NO muestra tablero durante una pausa del programa.

3.2.2.2.3 Mejoras al sistema de referencia

Cargar Partidas Famosas

No solo consiste en cargar una partida como ya hace el programa, si no:

— Generar los archivos, con los comentarios. (no existen actualmente)

- Tratamiento en pantalla de las partidas y sus comentarios.
- Cargar Ejercicios

Este es un caso similar a las partidas en lo referente a los comentarios, pero normalmente los ejercicios implican varios posibles movimientos, y algunos erróneos también podrían estar comentados. Por lo tanto en este caso se ofrece la posibilidad de un árbol lógico de movimientos y comentarios.

Mejor línea de juego mostrada en segundo tablero

Consiste en visualizar gráficamente sobre un segundo tablero el "pensamiento" del programa, de forma similar a como lo hace Gambit.

- Opción de tablero 3D
- Mejora de la interacción
- Mejora del interfaz usuario, gráficos, selección por iconos, mayor número de selecciones en primer plano.

3.2.2.2.4 Ampliaciones

Estas mejoras no se presupuestan puesto que es necesario un análisis de requisitos previo.

- EAO
- Comentarios sobre aperturas y otras jugadas tipo.
- Tratamiento Multimedia
- Tarjeta de sonido-voz, animaciones, etc.

3.2.2.2.5 Tareas y Plazos (Costes)

Tarea	Personal	Días de trabajo
Oferta y análisis funcional	Analista	12
Generar análisis orgánico del programa	Programador	6

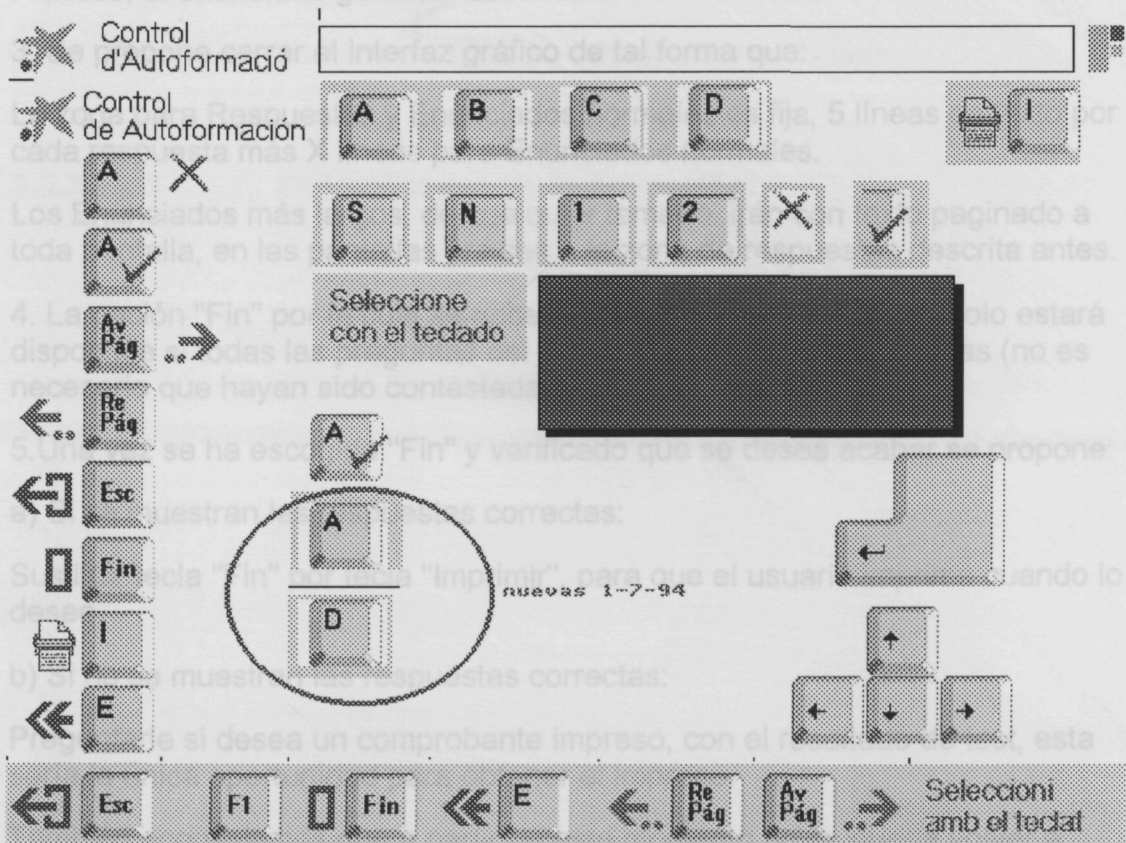
Traducción	Analista	2
Primer Idioma	Programador	8
Traducción	Analista	2
Por cada idioma añadido	Programador	5
Manuales De instalación	Analista	3
Manuales De usuario	Analista	5
	Editor	8
Tratamiento de archivos de partidas y ejercicios	Analista	5
	Grafista	3
	Programador	15
Mejor línea de juego mostrada en segundo tablero	Analista	1
	Grafista	2
	Programador	3
Opción de tablero 3D	Analista	3
	Grafista	5
	Programador	7
Mejora de la interacción	Analista	5
	Grafista	10
	Programador	12

3.2.3 Aplicaciones de autoaprendizaje

3.2.3.1 Proyecto CAVA Sistema de evaluación

Fecha: marzo de 1996

Como ejemplo en la clase de aplicación de autoformación se ha seleccionado un programa de autoevaluación, realizado para la evaluación periódica de los trabajadores de una entidad financiera. La imagen recoge todos los gráficos utilizados en la aplicación del proyecto CAVA.



Se diseñó para poder ser distribuido en disquetes que también recogían los resultados.

Este documento es el informe del Diseñador de Interfaz, aconsejado por un Experto en contenidos pedagogo y un Guionista.

3.2.3.1.2.1 Documentos previos

3.2.3.1.1 Justificación de propuestas de diseño para el cliente:

Se consideran leídos y comprendidos los documentos previos guardados en la

1. Se propone no informar sobre las respuestas correctas, motivos:

Una persona una vez ha observado la nota obtenida, puede decidir no entregar el disquete por perdida o por destrucción.

Es seguro que aquellas personas que deseen conocer los resultados correctos antes de contestar el test, tendrán medios de conseguirlo.

funcionamiento de la aplicación.

2. Se añade una nueva tecla en la zona de opciones de navegación que aparece cuando se ha entrado en un "Caso Práctico".

Mediante esta tecla es posible acceder, desde cualquier pregunta del Caso Práctico, al enunciado genérico del mismo.

3. Se propone cerrar el interfaz gráfico de tal forma que:

La zona para Respuestas y Enunciados normales es fija, 5 líneas de texto por cada respuesta más X líneas para Enunciados normales.

Los Enunciados más largos, de cualquier tamaño, irán con texto paginado a toda pantalla, en las pantallas previas a la zona de respuestas descrita antes.

4. La opción "Fin" por la cual se accede a la finalización del Test, solo estará disponible si todas las preguntas del programa han sido visualizadas (no es necesario que hayan sido contestadas).

5. Una vez se ha escogido "Fin" y verificado que se desea acabar se propone:

a) Si se muestran las respuestas correctas:

Sustituir tecla "Fin" por tecla "Imprimir", para que el usuario imprima cuando lo desee.

b) Si no se muestran las respuestas correctas:

Preguntarle si desea un comprobante impreso, con el resultado de test, esta sería la única oportunidad para obtener el impreso.

También aconsejamos que a partir de este momento no se pueda volver a entrar en el programa, y únicamente avise que se debe devolver el disquete al Departamento de Formación.

3.2.3.1.2 Documento de entrada en producción

3.2.3.1.2.1 Documentos previos

Se consideran leídos y comprendidos los documentos previos guardados en la carpeta.

Sin embargo las especificaciones de los Documentos Previos contradictorias con las reflejadas en el presente documento no son válidas.

Los documentos previos son:

- Análisis Funcional, y Análisis de Requisitos: Descripción genérica del funcionamiento de la aplicación.

- Borrador Programa de lectura de disquetes, y Análisis Cliente1: Detalles a tener en cuenta en el análisis final del programa de lectura de disquetes.
- Listado de preguntas con las respuestas correctas.
- Listado de Módulos (códigos de referencia a materias)

3.2.3.1.2.2 Flujo genérico de la Aplicación:

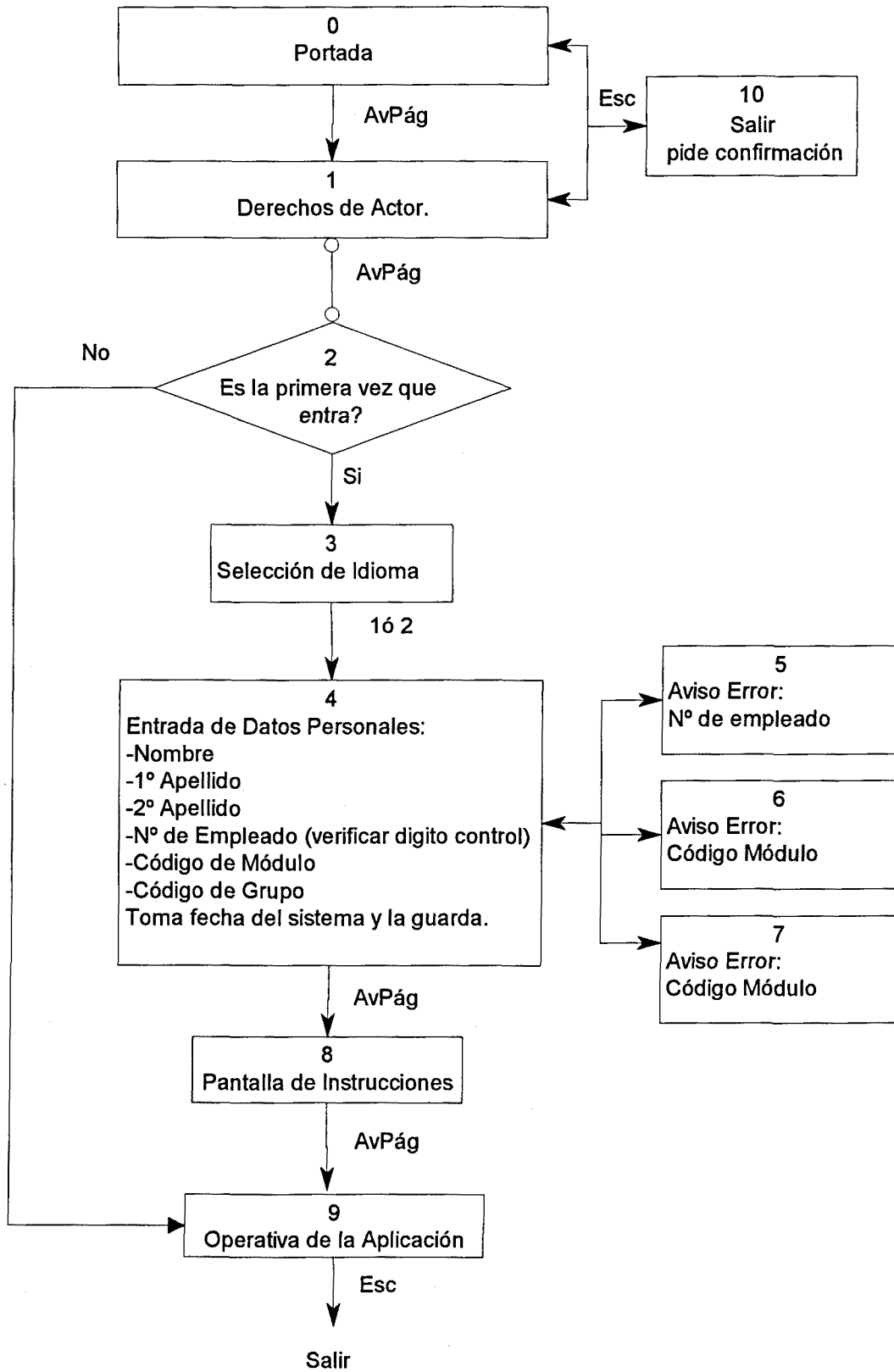


Ilustración 14. Flujo general. CAVA

Operativa de la Aplicación:

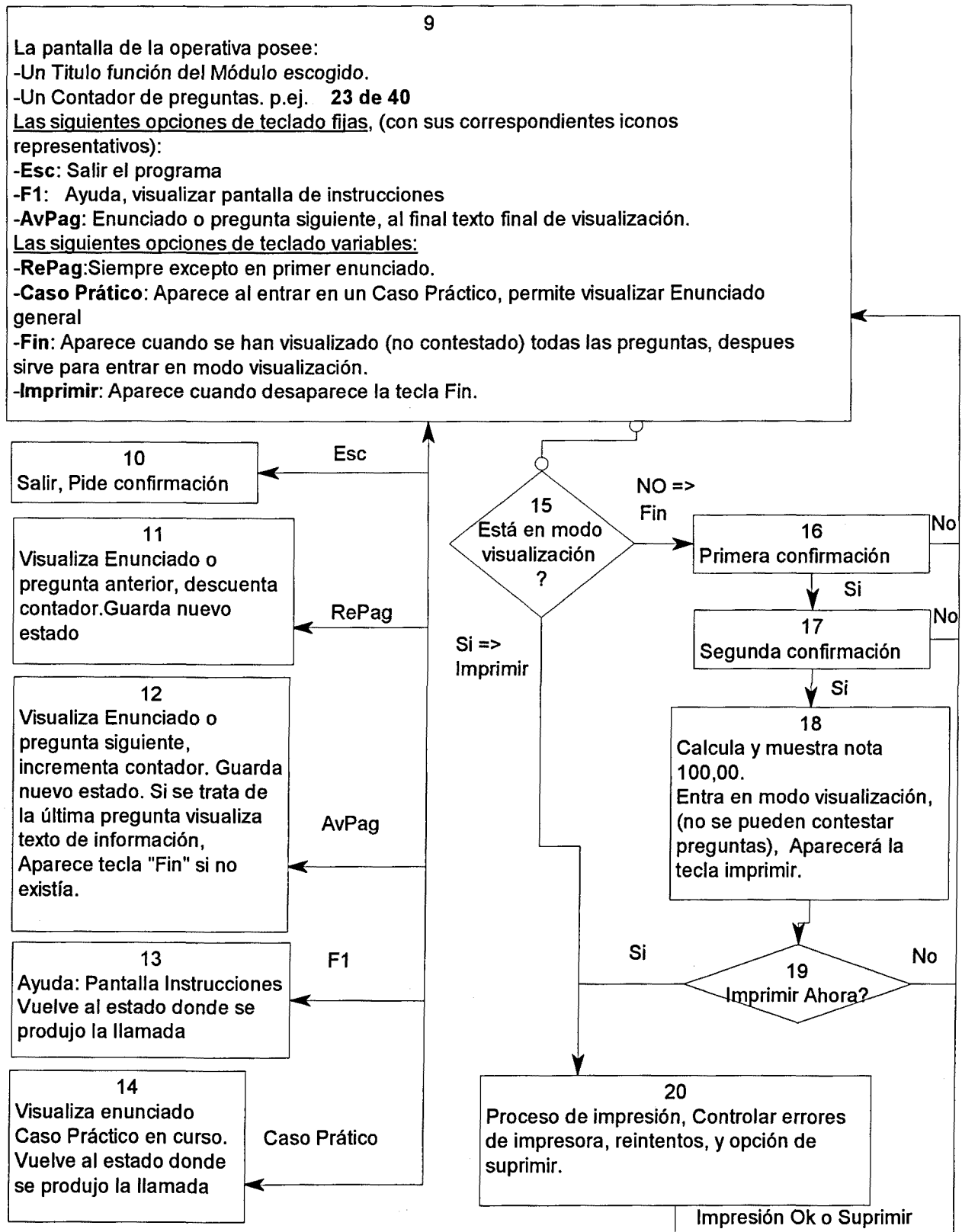


Ilustración 15. Operativa de la aplicación. CAVA

3.2.3.1.2.3 Descripción detallada de estados

Recordamos que la aplicación rueda sobre disquete, debe ocupar un mínimo de memoria convencional, y aprovechar al máximo la memoria instalada, del tipo que sea, para datos de la aplicación, gráficos, textos, índices, etc.

En el disquete, en principio, deben residir los textos de 10 Módulos formativos en dos idiomas.

Un usuario, alumno, solo puede escoger uno de los Módulos, y una vez finalizado devolver el mismo disquete, que ya contendrá el resultado, y sus datos personales.

La aplicación puede funcionar en "Modo Respuesta" o en "Modo Visualización", en el primero pueden responder preguntas, que quedan marcadas y modificar las respuestas. En el segundo se visualizan las respuestas del usuario y las respuestas correctas, pero no se pueden modificar las respuestas.

En todos los casos siempre existe la opción de ir hacia adelante y hacia atrás, dentro del estado 9 "Operativa de la Aplicación".

- **Nota:** Los gráficos están nombrados en función del idioma, de tal forma que los que empiezan por 0 son comunes a los dos idiomas, los que empiezan por 1 son para Castellano, y por 2 para Catalán.

Los parciales de borrado empiezan por el carácter "_".

A partir de aquí, en este documento, los gráficos se mencionarán omitiendo este dígito.

- **Nota:** Interacción válida Beep bueno, interacción no válida Beep malo.

Portada, entrada en el programa	
Reflejado en flujo por el estado 0	
Proceso de Carga: 1.Carga pantalla de fondo FONDO.PCX 2. Carga parciales de datos en el siguiente orden:	Interacción: »Si aprietan ESC va a "Salir". »Si aprietan AvPág va a "Derechos de Autor".

<p>POR1, POR2, POR3, POR4, POR6, POR5.PCC</p> <p>3.Carga parciales de navegación en el siguiente orden:</p> <p>ESC, AVPAG, MSELEC.PCC</p>	
<p>Derechos de Autor</p> <p>Reflejado en flujo por los estados 1 y 2.</p>	
<p>Proceso de Carga</p> <p>1.Carga parcial de datos POR7.PCC.</p>	<p>Interacción:</p> <p>»Si aprietan ESC va a "Salir".</p> <p>»Si aprietan AvPág :</p> <p>Se debe reconocer si es la primera vez que entra el usuario, o lo que es lo mismo, si es la primera vez que se utiliza el disquete (Marca de Utilizado).</p> <p>a)Es la primera vez que utiliza el disquete: Accede a "Selección de Idioma" estado 3.</p> <p>b)No es la primera vez:</p> <p>»Comprueba en que página se quedó la última vez:</p> <p>»Si pertenece a un Caso Práctico o no, y la posición del texto del enunciado.</p> <p>»Si está en modo respuesta o en modo visualización.</p> <p>»Con estos datos invoca al proceso "Operativa de la Aplicación" estado 9.</p>
<p>Selección del Idioma</p> <p>Reflejado en flujo por el estado 3.</p>	
<p>Proceso de Carga:</p> <p>1. Carga parciales de datos en el siguiente orden:</p> <p>IDIOMA.PCC</p>	<p>Interacción:</p> <p>»Si aprietan ESC va a "Salir".</p> <p>»Si aprietan tecla:</p>

<p>2. Carga parciales de navegación en el siguiente orden:</p> <p>Borra parcial AvPág. con _AVPAG.PCC</p>	<p>1 ==> .Castellano</p> <p>2 ==> Catalán</p> <p>En los dos casos van a "Entrada Datos". A partir de este momento los documentos que se muestran pueden estar sujetos al idioma. Sin embargo no se puede grabar en disquete nada relativo a este hecho, puesto que puede abandonar antes de empezar el curso</p>
<p>Entrada de datos</p> <p>Reflejado en flujo por el estado 4.</p>	
<p>Proceso de Carga:</p> <p>1. Carga FONDO.PCX</p> <p>2. Carga parciales de datos en el siguiente orden:</p> <p>LOGO,PERSON,</p> <p>3. Escritura de Cabeceras de campo en el idioma escogido.</p> <p>4. Carga del parcial IPERSON.PCC</p> <p>5. Carga parciales de navegación en el siguiente orden:</p> <p>ESC, AVPAG, MSELEC.PCC</p>	<p>Interacción:</p> <p>» Si aprietan ESC va a "Salir".</p> <p>» Introducción de texto según características del campo de entrada</p> <p>» Nombre</p> <p>10 car, alfabéticos, en mayúsculas.</p> <p>» 1º Apellido</p> <p>10 car, alfabéticos, en mayúsculas.</p> <p>» 2º Apellido</p> <p>10 car, alfabéticos, en mayúsculas.</p> <p>» Nº de Empleado</p> <p>7 car, numéricos.</p> <p>Se rellena a ceros por la izquierda.</p> <p>Posee validación de dígito de control, si falla va a "Error Número de Empleado".</p> <p>» Código de Módulo:</p> <p>4 car, alfanumérico.</p> <p>Tiene validación, debe ser uno de los disponibles.</p> <p>Si no cumple validación va "Error Código de</p>

	<p>Módulo".</p> <p>»Código de Grupo: 5 car, numérico.</p> <p>Tiene validación de que se introduzcan 5 dígitos y no menos, si no es así va "Error Código de Grupo".</p> <p>»Si aprietan AvPág:</p> <ol style="list-style-type: none"> 1. Almacena los datos introducidos por el alumno en disquete. 2. Procesa los archivos implicados en la selección de Idioma y Módulo. 3. Va a "Instrucciones de entrada".
<p>Instrucciones de Entrada</p> <p>Reflejado en flujo por estado 8</p> <p>Muestra la misma información que "Instrucciones" pero procesa diferente.</p>	
<p>Proceso de Carga</p> <ol style="list-style-type: none"> 1. Carga _PREGUN.PCC. 2. Carga parcial de instrucciones, INST.PCC. 3. Escribe texto correspondiente en el idioma activo. 	<p>Interacción</p> <p>»Si aprietan ESC va a "Salir".</p> <p>»Si aprietan AvPág :</p> <ol style="list-style-type: none"> 1. Prepara datos para ir a "Operativa de la Aplicación": <ul style="list-style-type: none"> »Página en que se quedó la última vez=0 »Caso Práctico = NO »Modo Respuesta 2. Va a "Operativa de la Aplicación".

3.2.3.1.2.4 Operativa de la Aplicación

Reflejado en flujo por los estados 9 y 15.

Proceso de Carga:	Interacción:
-------------------	--------------

<p>1.Carga texto a mostrar en pantalla en la memoria, comprueba si se trata de una pantalla de respuesta (primer estilo de párrafo <EP>). En este caso también comprueba el estado de las respuestas, si están seleccionadas, y cual es la correcta, si está en modo visualización.</p> <p>2.Carga FONDO.PCX</p> <p>3.Carga parciales LOGO, TITOL.PCC</p> <p>4.Escribe el texto del Título del Módulo seleccionado en el idioma correspondiente.</p> <p>5.Si es <EP>:</p> <ul style="list-style-type: none">1.Carga EP.PCC y escribe texto Enunciado Pequeño, <EP> y <EPC>.2.Carga A.PCC y escribe texto.3.Carga B.PCC y escribe texto.4.Carga C.PCC y escribe texto.5.Carga D.PCC y escribe texto.6.Marca las respuestas que estén seleccionadas con MARCA.PCC.7.Si está en Modo Visualización, marca respuestas correctas con CORRECT.PCC. <p>6.Si no es <EP>:</p> <ul style="list-style-type: none">1.Carga EG.PCC y escribe texto Enunciado Grande	<p>Durante este estado, el alumno puede leer, contestar / rectificar preguntas/respuestas, acabar el examen, imprimir comprobante y textos, visualizar respuestas contestadas y correctas.</p> <p>El idioma de trabajo ha sido seleccionado al inicio del programa, en la primera ejecución, y el archivo de textos implicado es único de entre los 20 que pueden residir en el disquete.</p> <p>En todo momento el alumno puede avanzar y retroceder a lo largo del texto del examen, sin embargo, dependiendo del tipo de enunciado que se esté visualizando, será diferente el proceso.</p> <p>También está influenciado por el Modo Respuesta/Visualización, en el primero se puede contestar y rectificar, en el segundo solo se puede visualizar respuestas correctas y seleccionadas.</p> <p>Otro detalle es el funcionamiento de los "Casos Prácticos", que poseen un Enunciado común a varias preguntas, este enunciado debe poder visualizarse desde cualquier pregunta del "Caso Práctico" del que se trate. Los "Casos Prácticos" siempre van contiguos al final del texto.</p> <p>Cada vez que se produzca un cambio de página, debe grabarse en el disquete el nuevo estado, para poder acceder, en una nueva ejecución, y sin perder datos, aunque por ejemplo, el ordenador se haya apagado de súbito. Sin embargo, esto se realizará después de mostrar el siguiente estado y mientras el alumno está leyendo el nuevo texto.</p>
--	--

<p><EG> y <EGC>.</p> <p>7. Escribe contador de preguntas p. Ej.: 12 de 40.</p> <p>8. Carga parciales ESC, F1.PCC.</p> <p>9. Si está en modo Visualización carga I.PCC.</p> <p>10. Si está en modo Respuesta y Visto carga FIN.PCC.</p> <p>11. Si está en las respuestas de un Caso Práctico carga E.PCC.</p> <p>12. Si no es la primera página carga REPAG.PCC</p> <p>13. Carga AVPAG, y MSELEC.PCC.</p>	
<p>A, B, C, y D (Selección de respuesta)</p>	
	<ul style="list-style-type: none"> - Solo están activas si estamos en <EP> y Modo Respuesta. - Son conmutadores que seleccionan o deseleccionan las respuestas. - Nota: Al inicio todas las respuestas están deseleccionadas. <ol style="list-style-type: none"> 1. Se anota en memoria el nuevo estado. 2. Si está deseleccionada carga MARCA.PCC en correspondiente. 3. Si está seleccionada carga _MARCA.PCC en correspondiente.
<p>E (Enunciado Caso Práctico)</p> <p>Solo está activa si estamos dentro de las preguntas de un Caso Práctico.</p> <p>Salva estado invocante.</p>	
<p>Proceso de Carga:</p>	<p>Interacción:</p>

<p>1. Carga EG.PCC y texto del offset guardado para el Caso Práctico.</p> <p>2. Carga _E.PCC y desactiva.</p> <p>3. Carga AVPAG.PCC, puede que ya exista.</p>	<p>Funcionan teclas de ESC, F1, y Fin / imprimir, como en el caso invocante.</p> <p>Teclas de AvPág/RePág, funcionan paginando el texto del Enunciado, excepto si estamos en la última Pág./primera Pág. respectivamente, en este caso se va al estado invocante</p>
<p>AvPág (Página siguiente excepto última página)</p> <p>Reflejado en flujo por el estado 12.</p> <p>Si está en primera página carga REPAG.PCC y activa tecla.</p>	
<p>Si estamos en Enunciado Pequeño (pantalla de respuestas)</p>	<p>1. Si página siguiente empieza por <EP>:</p> <ol style="list-style-type: none"> 1. Si Modo Respuesta anota las respuestas finalmente seleccionas en memoria o disquete. 2. Anota la página a la que nos dirigimos en memoria o disquete. 3. Carga EP.PCC y escribe su texto. 4. Si Modo Visualización carga A.PCC. 5. Carga _RESP.PCC en A) y escribe su texto. 6. Si Modo Visualización carga B.PCC. 7. Carga _RESP.PCC en B) y escribe su texto. 8. Si Modo Visualización carga C.PCC. 9. Carga _RESP.PCC en C) y escribe su texto. 10. Si Modo Visualización carga D.PCC. 11. Carga _RESP.PCC en D) y escribe su texto. 12. Si Modo Visualización comprueba respuestas correctas y carga CORRECT.PCC donde corresponda. 13. Comprueba estado de respuestas seleccionadas en la página que se está

	<p>visualizando y carga MARCA.PCC donde corresponda.</p> <p>14.Incrementa contador (_CONTA.PCC, para borrar), guarda en memoria o disquete.</p> <p>2.Si página siguiente empieza por <EG> o <CP>:</p> <p>1.Si empieza por <CP> anota offset del Enunciado.</p> <p>2.Si Modo Respuesta anota las respuestas finalmente seleccionas en memoria o disquete.</p> <p>3.Anota la página a la que nos dirigimos en memoria o disquete.</p> <p>4.Carga EG.PCC y escribe su texto.</p> <p>5. Si empieza por <EG> incrementa contador. (_CONTA.PCC, para borrar), guarda en memoria o disquete.</p> <p>6.Desactiva teclas A, B, C, y D.</p> <p>3.Si no existen más páginas:</p> <p>1.Va a "Último"</p>
Si estamos en Enunciado Grande	<p>1.Si página siguiente empieza por <EP>:</p> <p>1.Anota la página a la que nos dirigimos en memoria o disquete.</p> <p>2.Carga _PREGUN.PCC.</p> <p>3.Carga EP.PCC y escribe su texto.</p> <p>4.Carga A.PCC y escribe su texto.</p> <p>5.Carga B.PCC y escribe su texto.</p> <p>6.Carga C.PCC y escribe su texto.</p> <p>7.Carga D.PCC y escribe su texto.</p> <p>8.Si Modo Visualización comprueba respuestas correctas y carga CORRECT.PCC donde corresponda.</p> <p>9.Comprueba estado de respuestas seleccionadas en la página que se está visualizando y carga MARCA.PCC donde</p>

	<p>corresponda.</p> <p>10. Si Modo Respuesta activa teclas A, B, C, y D.</p> <p>2. Si página siguiente empieza por <EG>:</p> <p>1. Anota la página a la que nos dirigimos en memoria o disquete.</p> <p>2. Carga EG.PCC y escribe su texto.</p>
<p>Si estamos en Caso Práctico (visualizando el enunciado general)</p>	<p>1. Si página siguiente empieza por <EP>:</p> <p>1. Anota la página a la que nos dirigimos en memoria o disquete.</p> <p>2. Carga _PREGUN.PCC.</p> <p>3. Carga EP.PCC y escribe su texto.</p> <p>4. Carga A.PCC y escribe su texto.</p> <p>5. Carga B.PCC y escribe su texto.</p> <p>6. Carga C.PCC y escribe su texto.</p> <p>7. Carga D.PCC y escribe su texto.</p> <p>8. Si Modo Visualización comprueba respuestas correctas y carga CORRECT.PCC donde corresponda.</p> <p>9. Comprueba estado de respuestas seleccionadas en la página que se está visualizando y carga MARCA.PCC donde corresponda.</p> <p>10. Incrementa contador (_CONTA.PCC, para borrar), guarda en memoria o disquete.</p> <p>11. Carga y activa E.PCC.</p> <p>12. Si Modo Respuesta activa teclas A, B, C, y D.</p> <p>2. Si página siguiente empieza por <EG> o <CP>:</p> <p>1. Anota la página a la que nos dirigimos en memoria o disquete.</p> <p>2. Carga EG.PCC y escribe su texto.</p> <p>3. Si es <EG> incrementa contador</p>

	<p>(_CONTA.PCC, para borrar), guarda en memoria o disquete.</p> <p>4.Si es <EG> carga y activa E.PCC.</p>
<p>RePág (Página anterior)</p> <p>Reflejado en flujo por el estado 11.</p> <p>Si accede a primera página carga _REPAG.PCC, y desactiva tecla.</p>	
<p>Si estamos en Enunciado Pequeño (pantalla de respuestas)</p>	<p>1.Si página anterior empieza por <EP>:</p> <ol style="list-style-type: none"> 1.Si Modo Respuesta anota las respuestas finalmente seleccionas en memoria o disquete. 2.Anota la página a la que nos dirigimos en memoria o disquete. 3.Carga EP.PCC y escribe su texto. 4.Si Modo Visualización carga A.PCC. 5.Carga _RESP.PCC en A) y escribe su texto. 6.Si Modo Visualización carga B.PCC. 7.Carga _RESP.PCC en B) y escribe su texto. 8.Si Modo Visualización carga C.PCC. 9.Carga _RESP.PCC en C) y escribe su texto. 10.Si Modo Visualización carga D.PCC. 11.Carga _RESP.PCC en D) y escribe su texto. 12.Si Modo Visualización comprueba respuestas correctas y carga CORRECT.PCC donde corresponda. 13.Comprueba estado de respuestas seleccionadas en la página que se está visualizando y carga MARCA.PCC donde corresponda. 14.Decrementa contador (_CONTA.PCC, para borrar), guarda en memoria o disquete.

	<p>2. Si página anterior empieza por <EG> o <CP>:</p> <ol style="list-style-type: none">1. Si Modo Respuesta anota las respuestas finalmente seleccionadas en memoria o disquete.2. Anota la página a la que nos dirigimos en memoria o disquete.3. Carga EG.PCC y escribe su texto.4. Si es <CP> decrementa contador (_CONTA.PCC, para borrar), guarda en memoria o disquete.5. Si empieza por <CP> carga _E.PCC y desactiva tecla.6. Desactiva teclas A, B, C, y D.
<p>Si estamos en Enunciado Grande</p>	<p>1. Si página anterior empieza por <EP>:</p> <ol style="list-style-type: none">1. Anota la página a la que nos dirigimos en memoria o disquete.2. Carga _PREGUN.PCC.3. Carga EP.PCC y escribe su texto.4. Carga A.PCC y escribe su texto.5. Carga B.PCC y escribe su texto.6. Carga C.PCC y escribe su texto.7. Carga D.PCC y escribe su texto.8. Si Modo Visualización comprueba respuestas correctas y carga CORRECT.PCC donde corresponda.9. Comprueba estado de respuestas seleccionadas en la página que se está visualizando y carga MARCA.PCC donde corresponda.10. Decrementa contador (_CONTA.PCC, para borrar), guarda en memoria o disquete.11. Si Modo Respuesta activa teclas A, B, C, y D. <p>2. Si página anterior empieza por <EG> o <CP>:</p>

	<ol style="list-style-type: none"> 1. Anota la página a la que nos dirigimos en memoria o disquete. 2. Carga EG.PCC y escribe su texto. 3. Si <CP> decrementa contador (_CONTA.PCC, para borrar), guarda en memoria o disquete. 4. Si empieza por <CP> carga _E.PCC y desactiva tecla.
<p>Si estamos en Caso Práctico (visualizando el enunciado general)</p>	<ol style="list-style-type: none"> 1. Si página anterior empieza por <EP>: <ol style="list-style-type: none"> 1. Anota la página a la que nos dirigimos en memoria o disquete. 2. Carga _PREGUN.PCC. 3. Carga EP.PCC y escribe su texto. 4. Carga A.PCC y escribe su texto. 5. Carga B.PCC y escribe su texto. 6. Carga C.PCC y escribe su texto. 7. Carga D.PCC y escribe su texto. 8. Si Modo Visualización comprueba respuestas correctas y carga CORRECT.PCC donde corresponda. 9. Comprueba estado de respuestas seleccionadas en la página que se está visualizando y carga MARCA.PCC donde corresponda. 10. Decrementa contador (_CONTA.PCC, para borrar), guarda en memoria o disquete. 11. Si accede a la última pregunta de un caso práctico anterior (Si existe algún offset en la pila de casos prácticos), carga y activa E.PCC. 12. Si Modo Respuesta activa teclas A, B, C, y D. 2. Si página anterior empieza por <CP>: <ol style="list-style-type: none"> 1. Anota la página a la que nos dirigimos en

	memoria o disquete. 2.Carga EG.PCC y escribe su texto.
F1 (Ayuda)	Reflejado en flujo por el estado 13. Acceden al aviso "Instrucciones".
AvPág (Página siguiente en última página)	Accede a aviso "Último".
Fin (Finalizar Modo Respuesta)	Esta opción solo está activa si estamos en Modo Respuesta y ya se han visualizado todas las respuestas, es decir, se ha accedido al estado "Ultimo". Va a aviso "Finalizar".
I (Imprimir)	Esta opción solo está activa si estamos en Modo Visualización. Va a aviso "Imprimir".
Esc (Salir)	Va a aviso "Salir".

Avisos

A continuación se describen los procesos de los estados denominados Avisos, que poseen la característica de que salvan el estado anterior para volver al mismo.

Para salvar el estado anterior, en el ámbito de gráficos, se debe guardar:

- 1.La zona de navegación (_NAVEGA.PCC)
- 2.Dependiendo de si el aviso es grande o pequeño:
 - o Si es pequeño el arrea que ocupará el parcial CE.PCC.
 - o Si es grande el área que ocupará el parcial (_PREGUN.PCC o GE.PCC).

<p>Error "Número de Empleado"</p> <p>Reflejado en flujo por el estado 5.</p> <p>Salva el estado invocante para aviso pequeño</p>	
<p>Proceso de Carga:</p> <p>1.Carga CE.PCC y escribe texto correspondiente a este error en idioma activo.</p> <p>2.Carga _NAVEGA.PCC.</p> <p>3.Carga MTOQUE.PCC</p>	<p>Interacción:</p> <p>Si toca cualquier tecla vuelve al estado invocante.</p>
<p>Error "Código de Módulo"</p> <p>Reflejado en flujo por el estado 7.</p> <p>Salva el estado invocante para aviso pequeño</p>	
<p>Proceso de Carga:</p> <p>1.Carga CE.PCC y escribe texto correspondiente a este error en idioma activo.</p> <p>2.Carga _NAVEGA.PCC.</p> <p>3.Carga MTOQUE.PCC</p>	<p>Interacción:</p> <p>»Si toca cualquier tecla vuelve al estado invocante.</p>
<p>Instrucciones</p> <p>Reflejado en flujo por el estado 13.</p> <p>Salva estado invocante para aviso grande.</p>	
<p>Proceso de Carga:</p> <p>1.Carga _PREGUN.PCC</p> <p>2.Carga INST.PCC y escribe texto correspondiente a este error en idioma activo.</p> <p>3.Carga _NAVEGA.PCC.</p> <p>4.Carga MTOQUE.PCC</p>	<p>Interacción:</p> <p>»Si aprietan cualquier tecla vuelve al estado invocante.</p>

<p>Último</p> <p>A este estado se accede al apretar AvPág desde la última pregunta de "Operativa".</p> <p>Salva el estado invocante para aviso grande</p>	
<p>Proceso de Carga:</p> <p>1.Carga GE.PCC y escribe texto correspondiente a este error en idioma activo.</p> <p>2.Carga _NAVEGA.PCC.</p> <p>3.Carga MTOQUE.PCC</p>	<p>Interacción:</p> <p>Si aprietan cualquier tecla vuelve al estado invocante, cargando, después de restaurar los gráficos, FIN.PCC</p> <p>Activa Modo Visto, en el estado invocante, activa tecla FIN.</p>
<p>Finalizar</p> <p>Reflejado en flujo por los estados 16, 17, 18, y 19.</p> <p>Guarda estado invocante para aviso grande.</p>	
<p>Fase 1. Proceso de Carga</p> <p>1.Carga _PREGUN.PCC</p> <p>2.Carga CE.PCC y texto de primera confirmación en idioma activo.</p> <p>3.Carga _NAVEGA.PCC.</p> <p>4.Carga SI, y NO.PCC.</p> <p>5.Carga MSELEC.PCC</p>	<p>Fase 1. Interacción</p> <p>»Si selecciona N vuelve al estado invocante.</p> <p>»Si selecciona S pasa a Fase 2</p>
<p>Fase 2. Proceso de Carga</p> <p>1.Carga CE.PCC y texto de segunda confirmación en idioma activo.</p>	<p>Fase 2. Interacción</p> <p>»Si selecciona N vuelve al estado invocante.</p> <p>»Si selecciona S pasa a Fase 3.</p>
<p>Fase 3. Proceso de Carga</p> <p>1.Calcula Nota, pasa parámetro a proceso invocante para que se ponga en Modo Visualización, y por lo tanto cargar y activar I.PCC después de restaurar gráficos.</p>	<p>Fase 3. Interacción</p> <p>»Si toca cualquier tecla vuelve al estado invocante.</p>

<p>2.Carga NOTA.PCC y texto específico de la nota (100,00).</p> <p>3.Carga INOTA.PCC</p> <p>4.Carga MTOQUE.PCC</p>	
<p>Imprimir</p> <p>Reflejado en flujo por el estado 20.</p> <p>Salva estado invocante para aviso grande.</p>	
<p>Fase 1. Proceso de Carga:</p> <p>1.Carga _PREGUN.PCC</p> <p>2.Carga 1.PCC y escribe texto "Imprimir solo comprobante con nota"</p> <p>3.Carga 2.PCC y escribe texto "Imprimir comprobante y todos los textos del cuestionario, con respuestas correctas, aproximadamente 30 minutos"</p> <p>4.Carga _NAVEGA.PCC.</p> <p>5.Carga ESC.PCC.</p> <p>6.Carga MSELEC.PCC</p>	<p>Fase 1. Interacción</p> <p>»Si aprieta ESC vuelve a estado invocante.</p> <p>»Si aprieta 1 o 2:</p> <p>1.Carga CE.PCC y texto "Impresión en curso"</p> <p>2.Carga _NAVEGA.PCC</p> <p>3.Lanza la impresión solicitada.</p> <p>»Si es Ok retorna a estado invocante.</p> <p>»Si error Impresora va a Fase 2</p>
<p>Fase 2. Proceso de Carga</p> <p>1.Carga CE.PCC y texto error.</p> <p>2.Carga ESC, I.PCC</p>	<p>Fase 2. Interacción</p> <p>»Si aprieta ESC vuelve al estado invocante.</p> <p>»Si aprieta I reintenta impresión.</p> <p>»Si impresión Ok vuelve a estado invocante.</p> <p>»Si error impresión vuelve a Fase 2</p>
<p>Salir</p> <p>Reflejado en flujo por el estado 10.</p> <p>Salva estado invocante para aviso pequeño.</p>	
<p>Proceso de Carga:</p> <p>1.Carga CE.PCC y texto</p>	<p>Interacción</p> <p>»Si aprieta S, salva estado para</p>

correspondiente en idioma activo.	próxima ejecución y sale a DOS.
2.Carga _NAVEGA.PCC	»Si aprieta N, vuelve a estado invocante.
3.Carga SI, y NO.PCC,	
4.Carga MSELEC.PCC	

3.2.3.1.2.5 Descripción de la representación de textos

La representación de los textos consiste en ir paginando preguntas y sus respuestas.

Los Enunciados y las Respuestas poseen zonas de posición fija en pantalla e independiente de la cantidad de texto. El comienzo de escritura dentro de una zona también es fijo.

El texto que debe escribirse en una zona se suministra como una cadena ASCII sin saltos de línea, es decir, en el momento de la representación en pantalla, se debe controlar que el texto no sobrepase la longitud de la zona en cuestión, si es así se generará una línea nueva.

Tipos de pantallas de texto

Existen dos tipos de pantallas de texto:

- 1.Enunciados Pequeños y cuatro Respuestas:

Son **cinco zonas** fijas en una sola pantalla, una para Enunciado y cuatro para Respuestas, cada una puede contener **hasta cuatro líneas**, la fuente de letra es la misma, pero la zona para Enunciado es más larga que las zonas de Respuestas.

- 2.Enunciados Grandes y Casos Prácticos:

En los casos que el enunciado no cabe en la zona de cuatro líneas de la pantalla de respuestas descrita antes.

Estas son pantallas previas (cualquier nº de ellas) a la pantalla de respuestas, por lo tanto en estos casos también existirá texto del

enunciado en la zona de Enunciado Pequeño en la Pantalla de Respuestas (aunque solo diga: "Seleccione las respuestas correctas").

Esta zona de texto admite 19 líneas.

Descripción del fichero de textos

Existen 10 ficheros ASCII, uno por cada código Módulo que se puede seleccionar en la primera entrada en el programa.

Estos ficheros se generan desde la opción de exportar ASCII de Amipro, de tal forma que se utilizan los estilos de párrafo para identificar las diferentes partes del archivo.

En la opción de exportar ASCII se debe escoger:

- Con finales de línea
- Con estilos de párrafo
- ASCII PC de 8 bits

Para editar los archivos con Amipro se utilizará la hoja de estilos CAVA.STY, en esta situación es archivo resultante contendrá un párrafo de cualquier longitud en cada línea, y al principio de la línea un descriptor del estilo de párrafo, p. ej. una Pantalla de Respuestas podría ser:

<EP>¿Texto de la pregunta?

<RB>2050111

<RM>4106115

<RM>4223508

<RB>2806133

Descripción de los estilos de párrafo:

<EG> Enunciado Grande

Debe coincidir con el inicio de una pantalla de un Enunciado a toda pantalla. El inicio de un enunciado a toda pantalla siempre será este estilo de párrafo, pero si existe más de una pantalla completa, todas ellas empezarán con este estilo.

<EGC> Enunciado Grande Continuación

En el caso de que el texto de un enunciado a toda pantalla, lleve finales de línea.

Por lo tanto estos párrafos se escriben en la misma pantalla y en la línea siguiente.

Una pantalla de un Enunciado Grande puede poseer como máximo:

- Un solo estilo <EG>
- Desde 0 hasta 18 estilos <EGC>

<EP> y <EPC> Enunciado Pequeño y Enunciado Pequeño Continuación

Texto para la zona de enunciado de la Pantalla de Respuestas. A continuación del enunciado deben venir los estilos de las preguntas A), B), C), y D), en este orden.

<RM> y <RMC> Respuesta Mala y Respuesta Mala Continuación

Texto para una zona de respuesta indica que la respuesta **no es correcta**.

<RB> y <RBC> Respuestas Buena y Respuesta Buena Continuación

Texto para una zona de respuesta indica que la respuesta **si es correcta**.

<CP> y <CPC> Caso Práctico y Caso Práctico Continuación

Son enunciados a toda pantalla, con las mismas consideraciones que los casos <EG y EGC>, sin embargo es posible que se escriban sobre un fondo diferente.

La característica principal de estos enunciados es que son genéricos para todas las preguntas siguientes (cada una con su correspondiente enunciado <EG,EGC,EP,EPC>), a partir de la existencia de un <CP> debe aparecer la opción de visualizar el enunciado genérico del caso práctico desde cualquiera de sus preguntas.

Después de un <CP> puede aparecer otro si el texto no cabe en una sola pantalla. Por lo tanto el texto del enunciado genérico de un caso práctico empieza en el primer <CP> y acaba en el siguiente <EP> ó <EG> (el enunciado de la primera pregunta de este caso práctico).

Si ya estamos dentro de las preguntas de un caso práctico y aparece otro <CP> implica que entramos en el siguiente caso práctico.

3.2.3.1.2.6 Flujo de visualización

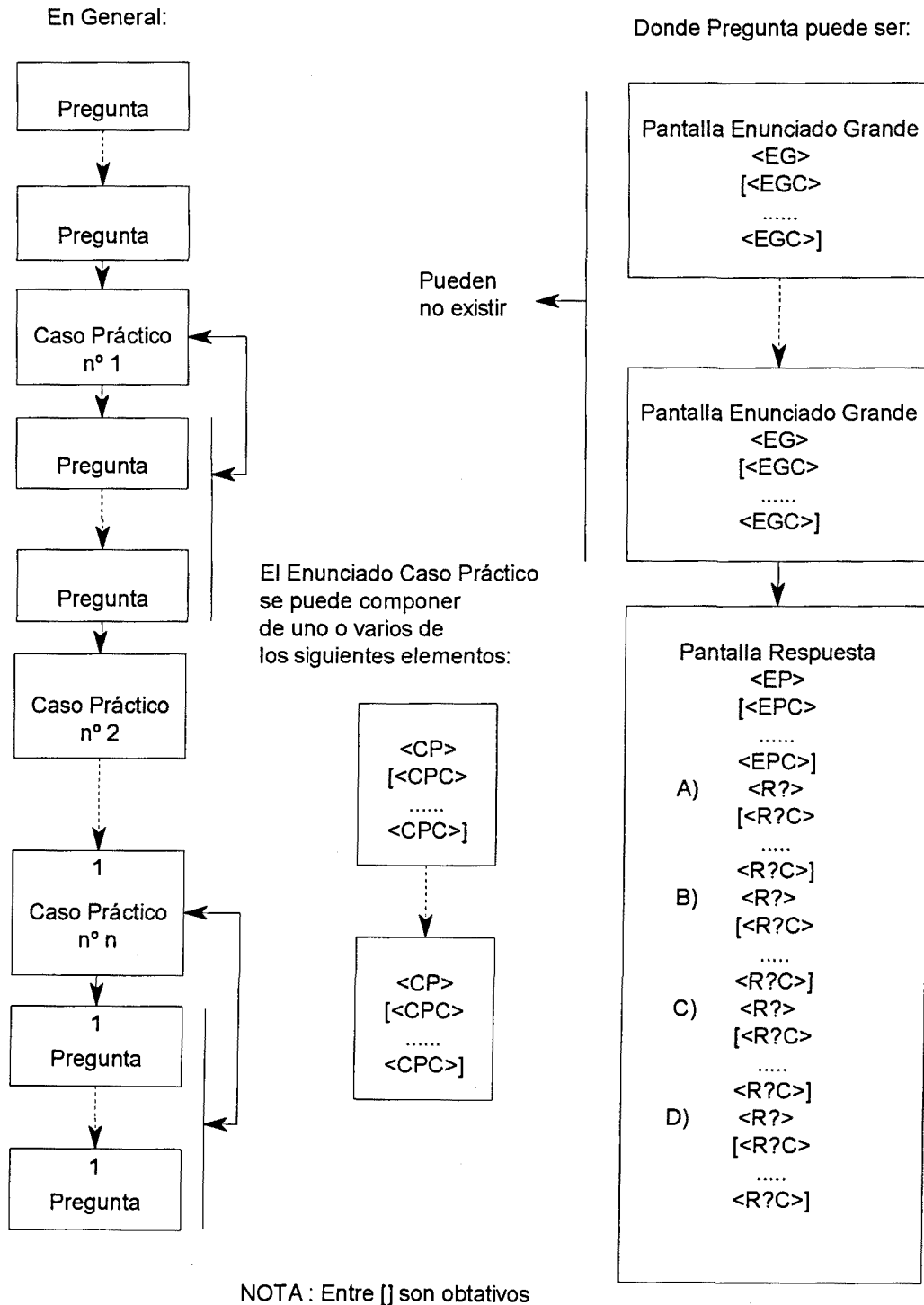


Ilustración 16. Flujo de visualización, edición Amipro, CAVA

3.2.3.2 Proyecto Aidea

Este proyecto desarrollado en entre los años 2000 y 2001, es uno de los mejores ejemplos de la complejidad de la producción multimedia.

También, por ser uno de los últimos desarrollos, refleja mejor el Modelo propuesto.

Las características básicas están descritas en las siguientes imágenes, se recogen también dos documentos muy interesantes. El primero es parte del Plan de producción y el segundo es parte del Guión interactivo.

Por respecto a la privacidad, los nombres de las personas que aparecen son ficticios.

Estos dos documentos también forman parte de las Normas de edición puesto que han sido utilizados para la formación del personal.

Proyecto AIDEA

Características principales del sistema

Producto para formación:

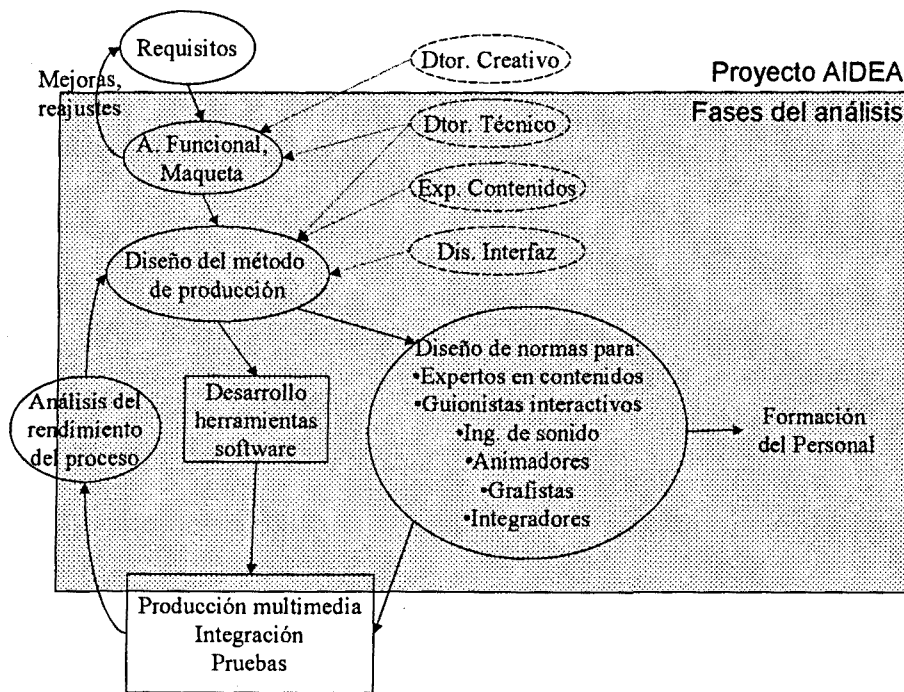
- Gestión compleja de autores, guionistas y contenidos.
- Alta calidad en la comunicación interactiva.
- Muchos usuarios, alto coste por fallos del sistema.

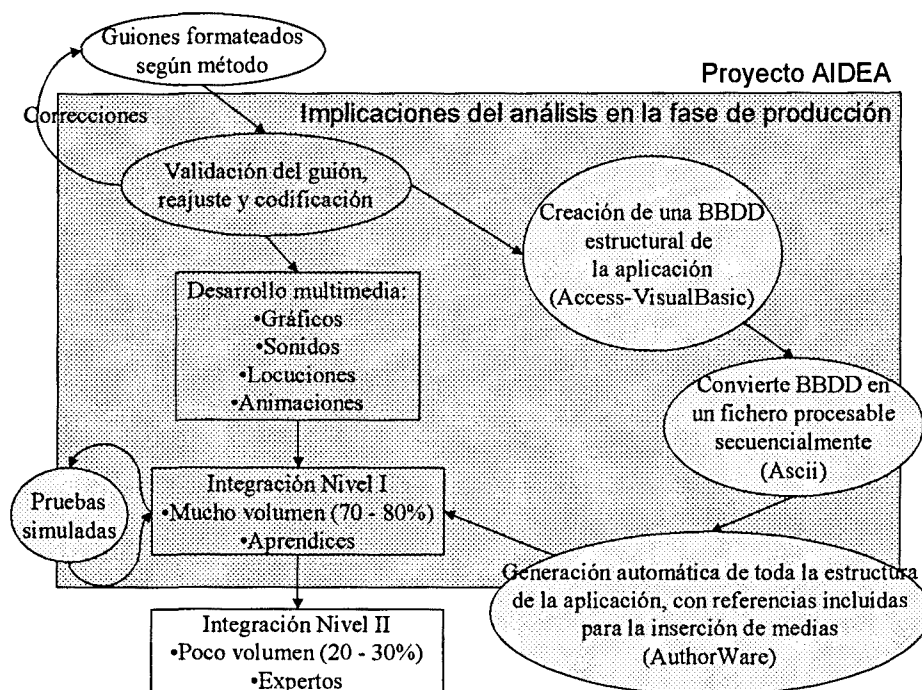
Producción intensiva y extensiva:

- Se trata de producir muchas asignaturas, y a su vez extensas, tanto simultáneamente como durante largos periodos de tiempo.
- Altos niveles de fiabilidad en los procesos productivos.
 - Los costes imputables al proceso productivo deben estar completamente controlados.

Recursos de producción dispares:

- Capacidad de producción en centros diferentes, teletrabajo.
- No coexistencia, en lugar ni tiempo, entre encargados y operadores.
- Gestión compleja de los recursos humanos.
- El flujo de datos entre las diferentes fases de producción debe ser sencillo, eficaz y sin ambigüedades.





3.2.3.2.1 Proceso de producción Aidea

Índice

- Introducción
- Recepción de la documentación
- Codificación de la documentación
- Construcción del DAT (Base de datos para la generación automática de código Authorware)
- Edición
- Locución
- Integración
- Control de calidad (pruebas alfa)
- Backup's
- Entrega al cliente

3.2.3.2.1.1 Introducción

Esquema resumen de procesos, documentación y responsables:

Proceso	Documentación, Productos	Responsables
Recepción de documentación en Word i JPEG	Informe de guionización Impresión de guiones Recogida en carpeta única	José María
Codificación	Guiones impresos Comprobación Informe de incidencias de codificación	Jordi (1/2) María José Montse
Construcción del DAT	Guiones numerados Comprobación	Nuria
Edición: Edición 1	Guiones numerados Ficha de pruebas Subproducto en formato "Modelo"	Joan Sandra Alba
Montador de secuencias	Guiones numerados Ficha de pruebas Subproducto en formato "Modelo"	Juan (1/2) Ricardo
Productor de interacción	Guiones numerados Ficha de pruebas Subproducto en formato "Modelo"	Dolores Lourdes Eva Montse
Glosario, Teoría, Test	Guiones numerados	Julia

autoevaluación	Ficha de pruebas Subproducto en formato "Modelo"	
Locuciones	Guiones de locución Ficha de pruebas Corte	María Alberto
Ajuste de integración 1: sobre locuciones ficticias	Producto casi acabado	Armando Nieves Manuel
Ajuste de integración 2: sobre locuciones reales	Producto acabado	Armando Nieves Manuel
Control de calidad	Informe	César María Eduardo Pilar José
Backup	Copias al servidor CD'S	María Enrique
Entrega al cliente	CD'S	María

Otras responsabilidades:

Coordinación: María

Director de producción: Manuel

Director Técnico: Bráis

Se propone la captación constante de personas para incorporar en el proceso productivo.

Se propone un curso de formación-selección en nuestras herramientas para incorporar nuevo personal. Puede tener dos enfoques:

Curso de editor de tutoriales:

Una sesión de introducción de 3 horas para personas que tengan ya una formación de base (EM, alumnos del Master) con la finalidad de editar AIDEA.

(Responsable: Bráis)

Curso de editor multimedia.

Un curso para el mismo colectivo, 30 horas de formación dirigida a formar personal para integrarlos en el equipo de materiales GMMD.

(Responsable: Armando)

Objetivos

Determinar los costes precisos de las diferentes etapas de producción

Llegar a conseguir el ratio: 1 lección por día.

Responsabilizar a los diferentes elementos de la producción de la calidad de su proceso y asegurar el menor número de errores en cada etapa.

Establecer herramientas de formación para mejorar el proceso e incorporar más personal.

Extrapolar los procesos de producción de AIDEA a todos los proyectos en trámite.

3.2.3.2.1.2 Recepción de la documentación

Es muy importante adecuar los guiones a las expectativas del cliente y a herramientas de producción. Dada la envergadura del proyecto es impensable encajar procesos artesanales en una estructura industrial.

Se propone estudiar el proceso de guionización para establecer las variables que intervienen en la validación de los mismos por parte del cliente y tratar de incidir en el proceso para optimizarlo y adecuarlo a las herramientas disponibles.

Se trata de conseguir copias de guiones antes y después de la validación, analizarlas y generar una documentación (**Informe de guionización**) que incluya tanto sugerencias a los guionistas para adaptarse a los gustos del cliente, como orientaciones de los tratamientos interactivos para adaptarse a las herramientas disponibles.

Se analizarán también los documentos de errores generados por AIDEA para determinar hasta que punto se trata de mal funcionamiento o de demanda de prestaciones no previstas en el guión.

(Responsable: Antonio)

Ritmo de entrega 18 lecciones / 3 semanas, reparto no homogéneo.

Es fundamental asegurar una cadencia constante de entrada de guiones.

3.2.3.2.1.3 Codificación

El codificador escribe, en lápiz sobre los guiones impresos, la numeración de los distintos pasos que constituyen cada uno de los temas.

Los codificadores asignan valores que determinan, sin ambigüedad, el tipo de paso de que se trate:

TIPO1: Locución o 1 gráfico:

Entrada,

Gráfico,

Texto

Locución

Cierra el paso

TIPO2: AniFlash

Entrada

Gráfico 1

Lanza Flash

Locución

Grafico2

Cierra

TIPO3: Cartela lección (es tipo 1)

Entrada

Gráficos finales

Locución

Cerrar

TIPO4: Animación Path

Entrada

Graf1

Los

Graf2

Animación

Cerrar

TIPO5 pasa a tipo 500 a tipo 599

Entrada

Graf1

loc

Cerrar

Interacción

00 única buena

01 Más buenas

02 teclado

03 Desplazamiento

04 ABC

05 Teclado y mouse

06 Textolntro

TIPO6 Secuencia gráfica

Entrada

Graf1

Loc

Graf 2

Espera tiempo

Cerrar

TIPO7 pasa a tipo 700 a tipo799

Entrada

Graf1

loc

Cerrar

Interacción

00 única buena

01 Más buenas

02 teclado

03 Desplazamiento

04 ABC

05 Teclado y mouse

06 TextoIntro

Se debe comprobar una vez que la asignación de código es correcta.

En caso de duda en la asignación, porque el paso no se adapta a ninguno de los previstos, el codificador modificará la interacción para adaptarla a las especificaciones y generará un informe (**Informe de incidencias de codificación**) con la incidencia que entregará a la coordinación del proyecto **María** que ha su vez hará llegar al Director de producción **Manuel**, que si lo considera oportuno evaluará la importancia de la incidencia y la comentará con el cliente

Ritmo de trabajo 1 lección / hora.

3.2.3.2.1.4 Construcción del DAT (Base de datos para la generación automática de código Authorware)

El archivo .DAT es la base para construir la estructura del programa para la edición.

Se debe comprobar una vez que la escritura es correcta, por persona diferente.

Ritmo de 1 lección / 2 horas

3.2.3.2.1.5 Edición

Se obtienen tres estructuras acompañadas de tres copias de la documentación para otras tantas tareas de edición:

Edición 1: Se integran gráficos, animaciones, fondos, resaltes, Juanos, mascotas que no son secuencias, ni interacciones. (Precaución con la utilización de los nombres en el interior de la secuencia. Utilización de iconos configurados que tienen "prevent automatic erase" y un icono de cálculo).

Se puede ejecutar en un entorno simulado

Parece útil realizar primero esta edición i después montar las secuencias "tipo1"

Se debe probar la estructura contestando bien y contestando mal, adelante resolver, y a Demotest avance y retorno una vez presentado el resultado

Montador de secuencias: Discrecionalmente se introducen transiciones, carga secuencial de gráficos y pequeñas animaciones de desplazamiento que pretenden dinamizar los pasos de tipo 1.

Productor de interacciones: Existen unos modelos predefinidos de interacción que deben ser utilizados.

En el caso de la detección de alguna interacción no definida se debe generar un informe: **Informe de interacción** que se hará llegar al coordinador y éste al Director de producción.

Existen unas tareas externas al proceso: Glosario que incorpora texto y gráficos, la teoría y los tests de autoevaluación .

Procurar trabajar sin desplazar archivos entre máquinas, parece un factor de corrupción de los archivos.

3.2.3.2.1.6 Locución

Generar la documentación para las locuciones. Comprobarla.

Hacer las locuciones con una cadencia de 18 lecciones cada 3 semanas.

Para ello se convocan 4 locutores 3h y la mascota 1,5 h por cada bloque.

Cortes: 18 lecciones/semana.

3.2.3.2.1.7 Integración

Ajuste de integración1: Sobre locuciones ficticias generadas automáticamente (esta fase se realiza sólo si no se dispone de las locuciones finales). El resultado debe ser perfectamente coherente.

Ajuste de integración 2 : Sobre locuciones reales. Si se ha realizado el ajuste de interacción 1 debería bastar con sustituir los archivos de locución y ajustar los sincronismos, si es necesario.

El responsable puede necesitar nuevos elementos gráficos, otros medias o interacciones. Se dirigirá a los operarios de procesos anteriores con **PRIORIDAD ABSOLUTA**.

El responsable de esta fase debe dejar la aplicación en situación de ser entregada al cliente.

Pruebas: Se debe probar la estructura contestando bien y contestando mal, adelante resolver, y a Demotest avance y retorno una vez presentado el resultado.

3.2.3.2.1.8 Control de calidad (pruebas alfa)

Consiste en comprobar el producto final y de generar un **Informe de calidad** para la fase 6.

3.2.3.2.1.9 Backup's

Proceso de Backup's a diseñar por M.A.Brigos y a implementar por Enrique

3.2.3.2.1.10 Entrega al cliente

Responsable: María