

SISTEMAS LEGADOS

Práctica 1

Cajero Automático UnizarBank

Martín Gascón Laste (764429)

Daniel Huici Meseguer (758635)

Eduardo Ruiz (764539)

Resumen

En esta primera práctica se ha obtenido un código fuente en lenguaje COBOL, el cuál fue parcialmente recuperado de una copia de seguridad. La práctica se ha dividido en tres fases: una primera fase de compilación, donde se consigue ejecutar el programa, una segunda fase de solución de errores, donde se corrigen todos los bugs que pueda tener, y finalmente, una tercera fase donde se añaden ciertas funcionalidades sobre la aplicación.

En este documento se muestra cómo se ha procedido con cada una de las fases, cuál ha sido su evolución, problemas que se han presentado, así como la organización entre los miembros que conformamos el grupo.

Compilación

Al obtener el código, lo único que conocemos es que está programado en COBOL, pero no conocemos la variante, y por tanto, tendremos que encontrar un compilador compatible, ya que no cualquiera nos vale.

Se comenzó a trabajar sobre OpenCobol IDE, el cuál traía de serie el compilador GnuCobol 2.0.0. Al tratar de compilar con éste, aparecían una gran cantidad de errores de notación. Con una pequeña investigación descubrimos que éste compilación no soporta notación de Microsoft Cobol (sobre el que está basado el código), por tanto, debemos actualizar a una versión más reciente del compilador, la versión 3.1, que sí que soporta este tipo de notación.

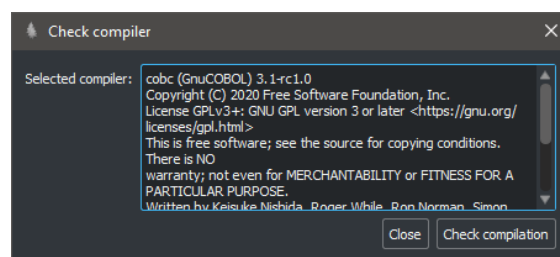


figura 1

Tras actualizar, desaparecieron gran parte de estos errores, pero todavía existían ciertos errores de sintaxis que deberíamos resolver manualmente, como son:

- Falta de comas entre las coordenadas de DISPLAY. Simplemente con añadir una coma se soluciona.

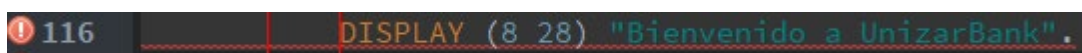


figura 2

- Sobran coordenadas en ACCEPT. En el registro al que hace referencias ya se encuentran implícitas las coordenadas de entrada de texto dónde recoger información, por tanto con quitarlas ya es suficiente.



figura 3

- Fallos menores: como puede ser alguna cláusula mal puesta en algún registro puntual, el cuál podemos solucionar si conocemos bien cómo COBOL maneja los datos y las cláusulas.

Tras haber revisado todo lo mencionado, tendremos la compilación correcta, sin embargo, durante la ejecución del programa nos encontraremos con un error de la propia aplicación, que trataremos en fase de ejecución y solución de errores.

Solución de errores

Como ya se ha comentado, la primera vez que ejecutemos nos vamos a encontrar con una pantalla como esta:



figura 4

Tras una investigación, llegamos a la conclusión de que las lecturas de los ficheros indexados no se están realizando correctamente. Observando los códigos de estados al abrir el fichero, obteníamos el código 35, que atendiendo a las guías que encontramos por internet, corresponde a que no se han encontrado los ficheros correspondiente.

En los ficheros donde se almacena la información (en formato ubd), podemos ver los datos que había guardados de las últimas ejecuciones del programa. Pero podemos comprobar, que en nuestro programa, cuando abrimos un fichero indexado con COBOL, se nos genera un fichero .dat y otro fichero .idx, los cuáles no están presentes. Por ello, debemos generar estos ficheros desde 0, atendiendo a la estructura dada en el código, y así poder trabajar con los ficheros sin tener errores.

Se realizó un pequeño programa en COBOL para generar nuevos ficheros, con la información pre-existente. Una vez tenemos los ficheros formateados como es debido, ya podemos trabajar con ellos, leer y escribir desde el programa.

Un error recurrente a lo largo de todo el programa, nos encontramos que para verificar que el fichero se había abierto correctamente, debíamos comprobar si el código de estado del fichero era distinto de 30, cuando tenía que ser 00. Como podemos ver en la tabla de códigos de error en escobol.com, un error 30 implica un error grave de entrada/salida, normalmente ajeno a Cobol. Nosotros queremos un estado 00, que implica que la operación se ha realizado satisfactoriamente.

Al probar las funcionalidades del programa nos dimos cuenta que al intentar realizar un cambio de clave daba error debido a que faltaba el fichero BANK8.cbl (el cual era el encargado de ejecutar esta funcionalidad). Para solucionar este error se creo un nuevo BANK8.cbl teniendo una interfaz igual que la que se muestra en el enunciado de la práctica. Para asegurarse de que el usuario que quiere cambiar la clave se ha usado el mismo método que al iniciar sesión dando un máximo de 3 intentos para cambiar la clave si el pin actual introducido es incorrecto.

Los colores de la consola eran inexistentes, todo estaba en blanco y negro. Esto ocurría debido a la incompatibilidad de declarar colores junto a una notación `DISPLAY(X,Y)`, por lo que hubo que convertir todos aquellos textos que requerían color, a un formato del tipo `DISPLAY "TEXT" LINE X COLUMN Y`. De esta forma, comenzamos a mostrar los colores. También se han reemplazado los colores de texto negros por colores de texto blancos en los fondos rojos.

También existía un pequeño glitch, en el que aparecían ceros en la esquina superior izquierda que no tenían ninguna validez. Se pueden eliminar introduciendo una clausura `BLANK WHEN ZERO` en las variables `Pressed-Key`, que son las que están provocando la aparición de estos ceros.

Nuevas funcionalidades.

Una vez que se han solucionado todos los fallos en la aplicación, se implementaron ciertas funcionalidades.

- Mejorar la interfaz de la opción de ingresar efectivo. Ahora se muestra una pantalla donde se le permite al usuario introducir cuántos billetes de 10, 20 y 50 desea ingresar, y mostrar el saldo disponible tras realizar la operación.

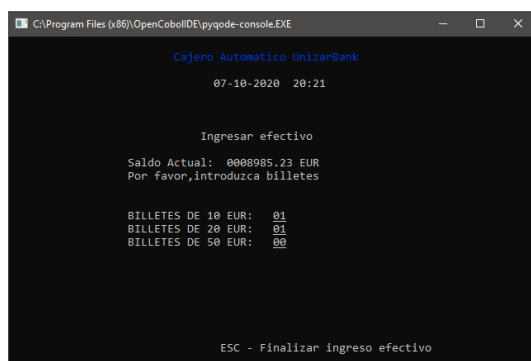


figura 5

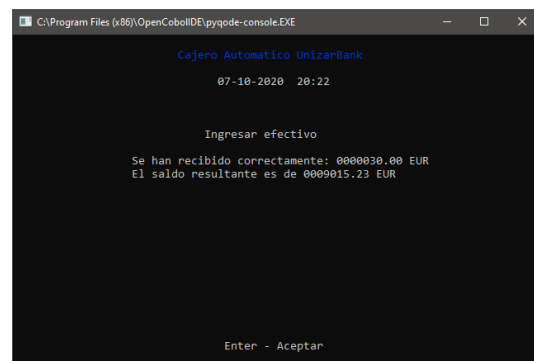


figura 6

- Además de las transferencias realizadas se mostrarán las transferencias mensuales que están planificadas para el periodo escogido en el filtro.

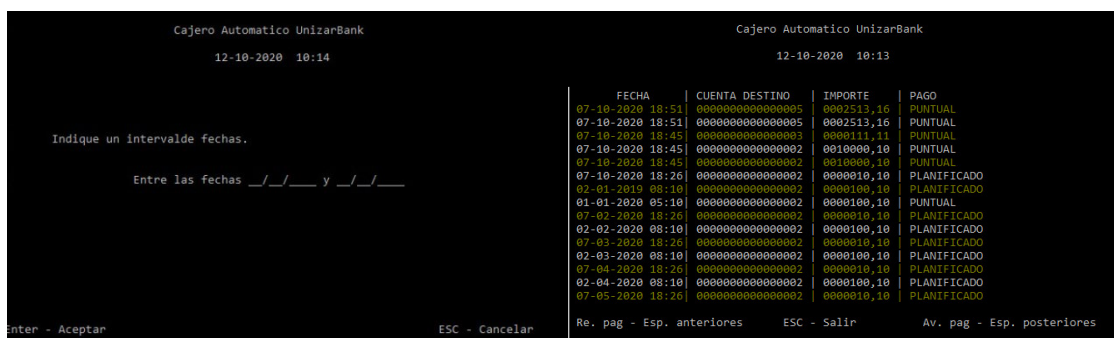


figura 8

Como se puede observar en la figura 8, se muestran en primer lugar las transferencias almacenadas en fichero y a continuación, se muestran todas las transferencias mensuales planificadas en el intervalo de fechas (posteriores a la última transferencia registrada en el fichero transferencias.udb).

Distribución del trabajo.

Tarea	Miembro del equipo	Tiempo invertido/persona
Investigación a cerca del compilador	Eduardo, Daniel y Martín	3h
Resolución problemas con (ausencias de) ficheros	Eduardo, Daniel y Martín	5h
Mejorar de la interfaz de ingreso de efectivo	Daniel	3h
Corrección de problemas relativos al coloreado de la interfaz.	Daniel	2h
Eliminación de etiquetas inicesarias (ceros en pantalla)	Daniel	1h
Desarrollo de la funcionalidad “cambiar clave”	Eduardo	3h
Corrección de fallos leves en funcionalidades ya existentes.	Eduardo y Daniel	2h
Solución de problemas en la visualización de movimientos.	Martín	3h
Desarrollo de la nueva funcionalidad listar transferencias.	Martín	8h
Desarrollo de la nueva funcionalidad realizar transferencias manuales y puntuales.	Eduardo	3h