

## UD5 – Caso Práctico 2: App de Cuentas Bancarias

### OBJETIVO

Implementar un programa modular que sirva de gestión de cuentas bancarias.

### REQUISITOS

- Se deben poder almacenar hasta un máximo de 100 cuentas bancarias. Al iniciar la aplicación esta no tendrá ninguna cuenta registrada.
- De cada cuenta bancaria necesitamos poder registrar el nombre del cliente y el saldo (cantidad en euros). El saldo puede ser tanto positivo como negativo.
- El usuario interactuará con la aplicación mediante un menú de texto que le permita elegir entre distintas opciones numeradas. Tras cada operación volverá a mostrarse el menú, a no ser que el usuario decida salir de la aplicación. Las opciones son:
  - 1 Ver cuentas.
  - 2 Ingresar dinero.
  - 3 Retirar dinero.
  - 4 Agregar cuenta.
  - 5 Eliminar cuenta.
  - 6 Buscar cuenta.
  - 7 Mostrar morosos.
  - 8 Salir.
- **Opción 1. Ver cuentas:** Mostrará las cuentas registradas y toda su información, una por línea, y debe estar numerada (ver ejemplo abajo). Si no hay cuentas mostrará el mensaje "No hay cuentas".
  - 0. María Saldo: 1.500,25 €
  - 1. Jose Saldo: 512 €
  - ...
- **Opciones 2 y 3. Ingresar y retirar dinero:** Permitirá ingresar o retirar dinero en una cuenta. Se mostrará por pantalla la lista de cuentas y toda su información, y se le pedirá que elija una, además de la cantidad a ingresar o retirar. Se comprobará que la cuenta elegida existe y que la cantidad a ingresar o retirar es un valor mayor que cero (no se aceptan negativos). De lo contrario se volverá a pedir la información una y otra vez hasta que introduzca información válida. Se deberá actualizar el saldo.
- **Opción 4. Agregar cuenta:** Permitirá añadir una cuenta bancaria siempre y cuando la lista no esté llena ni exista otra cuenta al mismo nombre (no se permite que una misma persona tenga dos cuentas). Pedirá el nombre y el saldo inicial y se registrará en la lista de cuentas.
- **Opción 5. Eliminar cuenta:** Permitirá eliminar una cuenta bancaria. Se mostrará por pantalla la lista de cuentas y se le pedirá que elija una. Se comprobará que la cuenta elegida existe. De lo contrario se volverá a pedir la información hasta que elija una cuenta existente. La cuenta se borrará de la lista.

- **Opción 6. Buscar cuenta:** Permitirá buscar cuentas. Pedirá al usuario un campo de texto T y luego mostrará, en forma de lista, la información de todas las cuentas que en su nombre contengan T (sin diferenciar entre mayúsculas y minúsculas). Si no se encontrara ninguna cuenta se mostrará el texto “No se han encontrado cuentas”.
- **Opción 7. Mostrar morosos:** Mostrará la información de todas las cuentas con un saldo negativo.
- **Opción 8. Salir:** Terminará el programa.

## PASOS

1. **Realiza el diseño descendente (top-down)** del programa, dividiendo el problema en subproblemas más pequeños, sucesivamente hasta que sean simples y no sea necesario dividirlos más.
2. **Piensa de qué forma puedes almacenar la información más importante.** ¿Qué variables o estructuras de datos puedes necesitar? ¿De qué tipo serían?
3. **Identifica qué subproblemas convendría programar como una función.** Identifica también si hay subproblemas parecidos que puedan ser resueltos por la misma función mediante parámetros, o si conviene crear funciones sobrecargadas.
4. **Haz una lista con las funciones** que necesitarás programar (no todos los subproblemas necesitan ser funciones). Incluye la cabecera completa (tipo devuelto, nombre y parámetros) y una comentario explicando qué hace cada una de ellas.
5. **Implementa el programa:** primero todas las funciones y por último el main.