

## UD6 – Caso Práctico 2: Banco

### OBJETIVO

Implementar mediante Programación Orientada a Objetos (POO) una aplicación que sirva de gestión de cuentas bancarias de un banco.

### 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 saber el nombre del titular y el saldo (euros).
- El saldo puede ser positivo y también negativo pero como mínimo -100 €. Es decir, no se permiten cuentas con un saldo menor a -100 € ni realizar operaciones que dejen una cuenta con un saldo inferior a dicha cantidad.
- 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 Transferencia.
  - 5 Agregar cuenta.
  - 6 Eliminar cuenta.
  - 7 Buscar cuenta.
  - 8 Mostrar morosos.
  - 9 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). En el caso de una retirada, también deberá comprobarse que la operación está permitida (una cuenta no puede tener un saldo inferior a -100 €). Se deberá actualizar el saldo y mostrar la información de la cuenta.
- **Opción 4. Transferencia.** Permitirá transferir dinero de una cuenta de origen a otra de destino, es decir, retirar dinero de una cuenta e ingresarlo en otra. Se mostrará por pantalla la lista de cuentas y toda su información, y se le pedirá al usuario que elija una cuenta de origen y otra de destino (deben ser cuentas diferentes). Al igual que al ingresar y retirar dinero, deberán realizarse las comprobaciones pertinentes para

evitar que se transfieran valores negativos así como dejar una cuenta con saldo inferior a -100 €.

- **Opción 5. 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 6. Eliminar cuenta:** Permitirá eliminar una cuenta bancaria. Se mostrará por pantalla la lista de cuentas y se le pedirá al usuario que elija una. Se comprobará que la cuenta elegida existe y en tal caso se borrará de la lista.
- **Opción 7. 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 8. Mostrar morosos:** Mostrará la información de todas las cuentas con un saldo negativo.
- **Opción 9. Salir:** Terminará el programa.

## PASOS

1. **Lee detenidamente los requisitos** de la aplicación planteada. Varias veces si es necesario. Subraya o anota los aspectos más relevantes.
2. Realiza el **diagrama de clases** del programa planteado. Deberá incluir las clases necesarias (con sus atributos y métodos) así como las relaciones entre ellas. Utiliza algún software de diseño de diagramas como por ejemplo **draw.io**.
3. **Programa la aplicación** implementando las clases que has diseñado en el diagrama de clases.