



VNIVERSITAT ID VALÈNCIA

**Laboratorio de ESTRUCTURAS DE
DATOS Y ALGORITMOS**
Grado en Ciencia de Datos (1º)
Curso 2023-24**Práctica Nº 4: Clases y Secuencias**

Periodo de realización: Semana del 15 a 19/03/2024

Metodología de trabajo

Fases	Tipo	Dedicación
1. Resolver las tareas planteadas en este guion <u>antes</u> de iniciar la sesión presencial en el laboratorio.	No presencial	Máx. 4,5 h.
2. Resolver en el laboratorio un <u>nuevo ejercicio</u> basado en la resolución de las tareas previas.	Presencial	Máx. 3 h.

Introducción

El ámbito bancario es un campo de trabajo muy importante para los científicos de datos y es interesante empezar a familiarizarse con algunos conceptos básicos. Por ejemplo, el concepto de cuenta bancaria.

Las cuentas bancarias de los/las clientes de un banco son entidades que contienen (entre otros) los datos identificativos de la persona titular de la cuenta, un código de cuenta (IBAN), un saldo (dinero) disponible y un conjunto de operaciones realizadas en esa cuenta (ingreso o reintegros).

El código que identifica una cuenta tiene una estructura compleja que requiere cierta explicación. Las cuentas se identifican mediante un código que cumple con lo establecido en el Código Internacional de Cuenta Bancaria (*International Bank Account Number*), conocido por las siglas IBAN. Este código es un formato internacionalmente acordado para identificar cuentas bancarias transfronterizas con un riesgo reducido de errores de transcripción. Originalmente fue adoptado por el Comité Europeo de Estándares Bancarios (ECBS), y después como un estándar internacional bajo norma ISO 13616.

El IBAN consta de un máximo de 34 caracteres alfanuméricos, en función del país. En España el IBAN tiene 24 dígitos y está formado por 2 dígitos de país (ES), 2 dígitos de control y 20 dígitos del código completo de la cuenta bancaria.

Ejemplo IBAN: ES9820385778983000760236

Código del país ISO 3166	2 Dígitos de control IBAN	BBAN (Código Básico de Cuenta Bancaria)			
		Código de la entidad	Oficina de la cuenta	2 Dígitos de control Banco	Número de cuenta
ES	98	2038	5778	98	3000760236

El objetivo de la primera parte de la práctica es construir una clase que represente las cuentas bancarias de los clientes de un banco. Como siempre, antes de realizar la sesión de laboratorio se deberán realizar las tareas que se indican en los siguientes ejercicios.

Ejercicios (Fase 1)

Tarea 1

Se debe definir una clase *CuentaBancaria* que represente este tipo de concepto. Una cuenta bancaria debe tener como atributos el identificador de cuenta (un objeto de la clase IBAN proporcionada), el nombre del titular de la cuenta y el saldo/dinero disponible en la cuenta. Los valores de estos atributos se establecerán en el constructor (se pasarán como argumentos). La clase permitirá también realizar las siguientes operaciones con sus objetos:

- *GetIBAN*: Devolver el identificador IBAN de la cuenta (objeto de la clase IBAN).
- *GetTitular*: Devolver el nombre del titular de la cuenta.
- *GetSaldo*: Devolver el saldo actualmente disponible en la cuenta (en euros).
- *Ingresar*: Ingresar una cierta cantidad de dinero en la cuenta (en euros). Incrementa el saldo. El argumento debe de ser un número positivo. En caso contrario, el operador debe lanzar una excepción.
- *Retirar*: Retirar una cierta cantidad de dinero en la cuenta (en euros). Reduce el saldo. El argumento debe de ser un número positivo. En caso contrario, el operador debe lanzar una excepción.
- *str*: Que convierte a formato string el contenido del objeto.

Ejemplo de uso de una cuenta:

```
c = CuentaBancaria("ES9820385778983000760236", "P. Pérez", 1000)
# Ingresar 500 € en la cuenta, saldo = 1.500€
c.Ingresar(500)
# Retirar 300 € de la cuenta, saldo = 1.200€
c.Retirar(300)
```

En Aula Virtual se puede descargar la clase IBAN (ya implementada) que representa objetos con el formato estándar y que incluye operaciones para obtener un formato estructurado de este código (en grupos de 4 caracteres separados por blancos) y para conocer cada uno de los elementos estructurales del código por si fuera necesario para alguna operación.

Tarea 2

Añadir a la clase *CuentaBancaria* un mecanismo para almacenar las últimas operaciones realizadas por el cliente. Como se ha visto antes, las operaciones pueden implicar ingresar o retirar un cierto importe de dinero en la cuenta. Además del importe, cada operación tiene asociado un concepto, que no es más que un texto que describe el motivo de la operación.

En Aula Virtual se puede descargar la clase *OperacionBancaria* (ya implementada) que representa objetos de este tipo.

Lo que se debe hacer en la clase *CuentaBancaria* es:

- Añadir una estructura **pila** para guardar las operaciones, de manera que se pueda almacenar/consultar la información de las operaciones de la cuenta desde más reciente a la más antigua.
- Modificar las operaciones *Ingresar* y *Retirar* para que cada una de estas funciones genere una nueva *OperacionBancaria* con el importe ingresado o retirado, según corresponda, y su concepto asociado. Esa operación se debe almacenar en la pila de operaciones de la cuenta. Como la operación bancaria tiene asociado un concepto, tanto a *Ingresar* como *Retirar* se les debe añadir un nuevo argumento que sea el concepto que motiva el ingreso o la retirada de dinero. **Importante:** la operación *Retirar* debe generar una *OperacionBancaria* de importe negativo.
- Añadir una nueva operación a la clase, *MostrarOperaciones*, que mostrará por pantalla todas las operaciones almacenadas en la pila (primero la más reciente).

Tarea 3

Verificar con el programa de test proporcionado ("test_cuentas.py") el correcto funcionamiento de la clase que has creado. Revisa que todos los resultados obtenidos con el programa se corresponden con lo esperado.

Fase 2: Tarea final

Resolución de un nuevo ejercicio planteado durante la sesión presencial en el laboratorio.