

Actividad de Repaso: Introducción a POO

Tema 6: Clases, Objetos y Métodos

Curso: Python de 0 a Interfaces Gráficas

1. Objetivo

Consolidar los conceptos fundamentales de la Programación Orientada a Objetos (POO) mediante la creación de un programa que simule un sistema bancario simple.

- Definir una **Clase** como molde.
 - Usar el constructor `__init__` para inicializar **Atributos**.
 - Crear **Métodos** que modifiquen el estado del objeto (uso de `self`).
 - Instanciar múltiples **Objetos** independientes.
-

2. Descripción del Problema

Deberás crear una clase llamada `CuentaBancaria` que represente la cuenta de un usuario. Esta cuenta debe tener un titular y un saldo, y debe permitir realizar operaciones básicas como depositar y retirar dinero, controlando que no se pueda retirar más de lo que se tiene.

3. Instrucciones Paso a Paso

3.1. 1. Definición de la Clase y Constructor

Crea una clase llamada `CuentaBancaria`.

- Define el método constructor `__init__`.
- Debe recibir dos parámetros: `titular` (nombre de la persona) y `saldo_inicial` (cantidad de dinero con la que abre la cuenta).
- Asigna estos valores a los atributos del objeto usando `self.titular` y `self.saldo`.

3.2. 2. Métodos de Acción

Define los siguientes métodos dentro de la clase:

depositar(self, cantidad): ■ Recibe una cantidad flotante.

- Suma la cantidad al atributo `self.saldo`.
- Imprime un mensaje confirmando el depósito y el nuevo saldo.

retirar(self, cantidad): ■ Recibe una cantidad a retirar.

- **Validación:** Usa un `if` para verificar si `self.saldo` es mayor o igual a la cantidad.
- **Si hay fondos:** Resta la cantidad al saldo e imprime el saldo restante.
- **Si NO hay fondos:** Imprime un mensaje de error ("Fondos insuficientes").

3.3. 3. Método de Representación (`__str__`)

Define el método mágico `__str__(self)`. Debe devolver (return) una cadena de texto que resuma el estado de la cuenta.

- Ejemplo de retorno: `Cuenta de [Titular] | Saldo: $[Saldo]`

3.4. 4. Prueba del Programa (Main)

Fuera de la clase, escribe un bloque de código para probar tu creación:

1. Crea el **Objeto 1**: Una cuenta para Ana con \$1000.
2. Crea el **Objeto 2**: Una cuenta para Roberto con \$500.
3. Realiza operaciones:
 - Ana deposita \$200.
 - Roberto intenta retirar \$1000 (debe fallar).
 - Ana retira \$500.
4. Imprime ambos objetos al final para ver sus estados definitivos.

4. Código Base (Esqueleto)

Utiliza este esqueleto para comenzar tu práctica:

```
1 class CuentaBancaria:
2     def __init__(self, titular, saldo_inicial):
3         # TODO: Inicializar self.titular y self.saldo
4         pass
5
6     def depositar(self, cantidad):
7         # TODO: Sumar cantidad al saldo
8         pass
9
10    def retirar(self, cantidad):
11        # TODO: Verificar fondos y restar si es posible
12        pass
13
14    def __str__(self):
15        # TODO: Devolver string con info de la cuenta
16        return ""
17
18 # --- ZONA DE PRUEBAS ---
19 # Crea tus objetos aqui y llama a sus metodos
```