

Plan Formativo Full Stack Python	
Módulo	Programación Avanzada en Python
Tema	Clases y Objetos
Nivel de Dificultad	Media
Ejecución	Grupal
Duración	40 Min
Código Ejercicio	E.3.1.G1
Intención del aprendizaje o aprendizaje esperado:	
<ul style="list-style-type: none"> • Creación de Clases. • Creación de atributos y métodos. • Creación de objetos de una clase. Aplicación de métodos y uso de atributos. 	
Planteamiento del Problema:	
<p>Banco</p> <p>Una financiera ofrece cuentas similares a las de un banco a sus clientes.</p> <p>Se solicitan crear las clases Cliente y Financiera.</p> <p>La clase Cliente debe tener atributos:</p> <ul style="list-style-type: none"> • nombre • id (Único). • saldo. <p>La clase Cliente debe contar con los métodos:</p> <ul style="list-style-type: none"> • init. • girar. • abonar. • mostrar_saldo. <p>La clase Financiera debe tener atributos:</p> <ul style="list-style-type: none"> • nombre. • Id (Único). • saldo_institucional. • clientes (Lista de objetos de clase Cliente). <p>La clase Financiera debe tener los métodos:</p>	

- init.
- agregar_cliente.
- eliminar_cliente.
- transferir.
- giros_totales.
- abonos_totales.
- mostrar_saldo_institucional.

Además usted debe implementar la mecánica para operación de líneas de crédito. Se sugiere considerar que línea de crédito es un saldo negativo que es cubierto por el saldo_institucional y es de \$1Millón por cliente. El límite máximo que puede estar asignado a líneas de crédito no puede superar el 10% del saldo_institucional. Esto limitará la cantidad máxima de clientes o así como también el monto disponible para transferencias desde el saldo_institucional. El método agregar_cliente debe verificar que se cumple la condición del 10%, de lo contrario debe rechazar la creación del cliente. El método transferir debe verificar que el saldo_institucional satisfaga la condición del 10% según los clientes actuales, de lo contrario debe rechazar la transferencia.

El método transferir debe permitir transferencias entre clientes; y entre clientes y financiera.

Los id de clientes y financieras deberán ser creados con el módulo uuid y utilizando uuid4().

Las clases mencionadas deberán estar implementadas en un módulo llamado finanzas.

Deberá crear un script principal que importe el módulo finanzas. Y realice las siguientes operaciones:

1. Crear 2 financieras de la clase Financiera con nombre y saldo inicial de \$100Millones.
2. Crear 4 clientes por cada financiera.
3. Realizar 3 operaciones por cada cliente de distinto tipo (giro, abono).
4. Realizar giros en dos clientes que demuestren que el saldo no puede ser menor a -1000000.
5. Realizar 4 transferencias entre clientes.
6. Realizar 2 transferencias entre clientes y financiera.
7. Realizar dos operaciones en cada financiera, que demuestren que no pueden dejar a clientes con saldo menos a -1000000.
8. Realizar dos operaciones en cada financiera que demuestren que se rechazan transferencias por violar el límite del 10% del límite institucional.
9. Demostrar que su código no puede agregar clientes en casos en que el límite de 10% del saldo institucional se viole.
10. Imprima el estado de cuenta de cada cliente y la financiera.

11. Muestre sus resultados en una tabla.

Recursos Bibliográficos :

[1] Módulo 3 - Contenido 1: "Describe los conceptos fundamentales del paradigma de orientación a objetos haciendo la distinción entre el concepto de Clase y de Objeto"