

PROYECTO III

Se solicita diseñar un sistema bancario de prueba utilizando el paradigma orientado a objetos, dividiendo el sistema en clases que contengan métodos y atributos según su naturaleza.

El arquitecto de software dividió el sistema en las siguientes clases:

1. Clase Persona

Atributos:

- Nombre
- Apellido
- Cédula

2. Clase Usuario (Hereda de Persona)

Atributos:

- Nombre de usuario
- Contraseña
- Cuenta (Será una clase)

3. Clase Cuenta

Atributos:

- Número de cuenta (valor numérico entero de 12 dígitos)
- Saldo (dinero en USD)

Métodos:

- Depositar (agregar saldo a la cuenta)
- Retirar (retirar saldo de la cuenta)

4. Clase SistemaBancario**Atributos:**

- Usuarios (lista de clases **Usuario**)
- Sesión (aquí guardarás la cédula del usuario que inició sesión, la cédula es string)

Métodos:

- Menú de iniciar sesión:
 - ◆ Pedir: nombre de usuario y contraseña
- Menú de usuario:
 - ◆ 1. Crear cuenta bancaria (esta opción solo debe estar disponible si el usuario aún no tiene una cuenta bancaria)
 - ◆ 2. Depositar
 - ◆ 3. Retirar
 - ◆ 4. Transferir
 - ◆ 6. Cerrar sesión (volver al menú de inicio de sesión)

Requerimientos:

- Al crear una cuenta bancaria esta debe ser un valor string con doce (12) dígitos, también se debe pedir el saldo inicial. Por ejemplo:

Cuenta bancaria: **523658745214**

Saldo inicial: **400.00**

- Los usuarios solo pueden crear una cuenta bancaria, cuando creen una la opción de **Crear cuenta bancaria** se deshabilita (**esto no quiere decir que se quita de la lista, solo que cuando el usuario la seleccione se le mencione que ya posee una cuenta bancaria**)
- **Al depositar** dinero se debe validar que **solo se ingresen valores numéricos y no negativos**
- **Al retirar** dinero se debe validar que solo se ingresen valores numéricos y no negativos
- **Al transferir** se debe pedir la cédula de la persona a quien se va a transferir, si no existe se debe pedir otra cédula, si existe entonces se debe escribir el monto que deseamos transferir siempre y cuando corresponda a lo que tengamos disponible en nuestra cuenta, automáticamente debemos actualizar la cuenta de la persona a quien transferimos
- **Depositar, retirar y transferir debe tener en cuenta algo sumamente importante**, y es que si depositaremos, retiraremos o transferiremos una cantidad que sobrepasa el saldo que tiene el usuario en la cuenta debemos mostrar un mensaje de fondos insuficientes

- Al cerrar sesión el sistema debe llevarnos al menu de login por lo tanto el atributo de sesión debe estar vacío de nuevo (puede ser None o string vacío)

Consideraciones:

- Es sumamente importante validar cada entrada de datos, nada puede estar vacío o tener un tipo de datos erróneo, al ser un sistema bancario la seguridad es importante
- Se debe tener en cuenta que cuando se realicen depósitos, retiros o transferencias se deben validar valores numéricos con un máximo de dos (2) decimales (Investigar).
- La clase de **SistemaBancario** debería tener un método main() que ejecute toda la app

IMPORTANTE:

El sistema debe tener dos usuarios de prueba registrados por defecto, y son los siguientes:

Primer usuario:

- Nombre: John
- Apellido: Doe
- Cédula: 26275576
- Nombre de usuario: johndoe
- Contraseña: 123456

Segundo usuario:

- Nombre: Ryan
- Apellido: Smith
- Cédula: 25645888
- Nombre de usuario: ryansmith
- Contraseña: 123456

Fecha de entrega:

12PM 08-09-2024