

Principios Básicos de POO

Nombre del Estudiante

15 de marzo de 2021

Docente

Alejandro Rodas Vásquez
Universidad Tecnológica de Pereira

Antes de resolver el taller

Antes de resolver el taller, usted debe consultar:

- Cómo crear una lista
- Cómo almacenar varios objetos en ella
- Recorrer la lista y mostrar todos los objetos (sus atributos)

Ejercicio

Crea una clase Cuenta (bancaria) con atributos para el número de cuenta (un entero largo), el Documento de Identidad del cliente (otro entero largo), Nombre del cliente y Saldo actual. Así mismo se debe definir:

1. Constructor por defecto Número de cuenta, Documento de Identidad, Nombre, Saldo.
2. Para el número de cuenta no habrá mutador. Método depositarDinero(): actualizará el saldo de la cuenta aplicando 19 % de retención.
3. retirarDinero(double): permitirá sacar una cantidad de la cuenta (si hay saldo) Método que nos permita mostrar todos los datos de la cuenta.
4. Almacenar en una lista (cuentaClientes) todas las cuentas creadas.
5. Crear el diagrama de Clase pertinente (Consultar en los libros de la bibliografía)
6. Crear el diagrama Componentes pertinente (Consultar en los libros de la bibliografía)

Requerimientos de la Arquitectura

La aplicación debe tener dos Componentes (Modelo, UI) y que serán invocados desde un main.

Evidencias

Diagrama de Clases

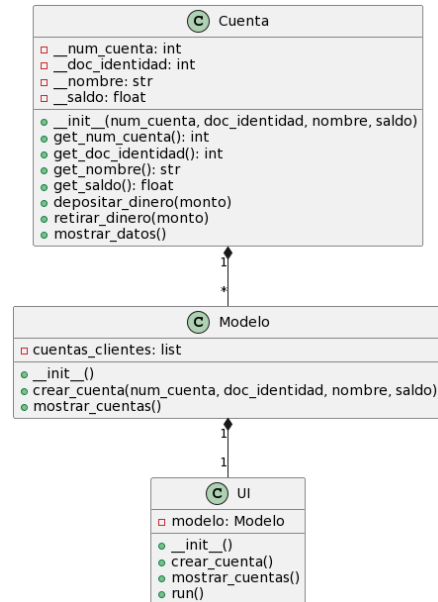
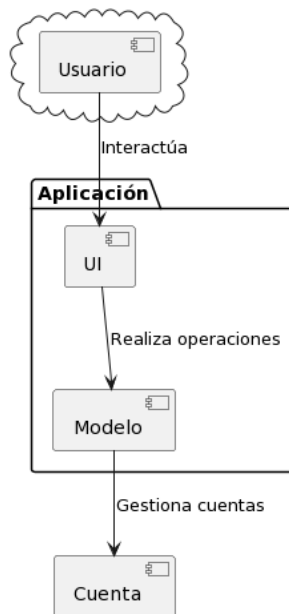


Diagrama de Componentes



Código Fuente del main.py

```
# Clase Cuenta
class Cuenta:
    # Constructor
    def __init__(self, num_cuenta, doc_identidad, nombre, saldo):
        self.__num_cuenta = num_cuenta
        self.__doc_identidad = doc_identidad
        self.__nombre = nombre
        self.__saldo = saldo

    # Getters
    def get_num_cuenta(self):
        return self.__num_cuenta

    def get_doc_identidad(self):
        return self.__doc_identidad

    def get_nombre(self):
        return self.__nombre

    def get_saldo(self):
        return self.__saldo

    # Método para depositar dinero (con retención del 19%)
    def depositar_dinero(self, monto):
        monto_deposito = monto - (monto * 0.19)
        self.__saldo += monto_deposito

    # Método para retirar dinero
    def retirar_dinero(self, monto):
        if self.__saldo >= monto:
            self.__saldo -= monto
        else:
            print("Saldo insuficiente")

    # Método para mostrar datos de la cuenta
    def mostrar_datos(self):
        print(f"Número de cuenta: {self.__num_cuenta}")
        print(f"Documento de identidad: {self.__doc_identidad}")
        print(f"Nombre: {self.__nombre}")
        print(f"Saldo: {self.__saldo}")

# Componente Modelo
class Modelo:
    def __init__(self):
```

```

        self.cuentas_clientes = []

    def crear_cuenta(self, num_cuenta, doc_identidad, nombre, saldo):
        cuenta = Cuenta(num_cuenta, doc_identidad, nombre, saldo)
        self.cuentas_clientes.append(cuenta)

    def mostrar_cuentas(self):
        for cuenta in self.cuentas_clientes:
            cuenta.mostrar_datos()
            print("-" * 30)

# Componente UI
class UI:
    def __init__(self):
        self.modelo = Modelo()

    def crear_cuenta(self):
        num_cuenta = int(input("Ingrese el número de cuenta: "))
        doc_identidad = int(input("Ingrese el documento de identidad: "))
        nombre = input("Ingrese el nombre: ")
        saldo = float(input("Ingrese el saldo inicial: "))
        self.modelo.crear_cuenta(num_cuenta, doc_identidad, nombre, saldo)

    def mostrar_cuentas(self):
        self.modelo.mostrar_cuentas()

    def run(self):
        while True:
            print("\nMenú:")
            print("1. Crear cuenta")
            print("2. Mostrar cuentas")
            print("3. Salir")
            opcion = input("Ingrese una opción: ")

            if opcion == "1":
                self.crear_cuenta()
            elif opcion == "2":
                self.mostrar_cuentas()
            elif opcion == "3":
                break
            else:
                print("Opción inválida")

# Main
if __name__ == "__main__":
    ui = UI()

```

```
ui.run()
```