

ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS
PROGRAMACIÓN ORIENTADA A OBJETOS

PROFESOR: Ing. Yadira Franco R

PERÍODO ACADÉMICO: 2025 B

TALLER 7 – Grupal 2 integrantes

TÍTULO:

INVESTIGACIÓN Y PRACTICA

Estudiante

Claudia Coello

Mateo Piedra

2025 B

Sistema Bancario: Uso de Clases

Abstractas y Jerarquías

UML CREAR INICIALMENTE

Aplicar el try y catch

1. Clase Abstracta: ` Persona`

Definición: Una clase abstracta no puede instanciarse directamente. Sirve como modelo base para otras clases y puede contener métodos abstractos (sin cuerpo) que deben ser implementados por sus subclases.

Atributos comunes sugeridos:

Private String nombre;

Private String cedula;

Private String direccion;

Private String telefono;

Métodos comunes analizar su implementación y cuando utilizar

public void actualizarDatos(String direccion, String telefono) { ... }

public abstract void mostrarRol();

Investigación sugerida:

- ¿Qué es una clase abstracta en Java?

- ¿Por qué no se puede instanciar una clase abstracta?

- ¿Debe el constructor de una clase abstracta ser también abstracto?

- Rpta: No. Una clase abstracta puede tener un constructor, pero no puede ser abstracto.

Sirve para inicializar atributos comúnmente heredados.

- ¿Cuándo usar y cuándo no usar clases abstractas? Y ejemplos de escenarios cuando aplicar y cuando no

CREAR LA PRACTICA CON EL MODELADO UML

2. Modificadores de acceso: ` private` vs. ` protected`

- private: Solo accesible dentro de la misma clase.

- protected: Accesible dentro de la misma clase, subclases, y paquete.

- public: Accesible desde cualquier lugar.

Uso sugerido:

- Usa protected en clases abstractas para que los atributos sean accesibles por las subclases.

- Usa private cuando quieras ocultar atributos que solo se modifiquen mediante getters/setters.

3. Subclase: `Cliente extends Persona`

Responsabilidad: Consumidor de productos bancarios.

Métodos específicos:

+ ingresarAlSistema(): boolean

+ registrarCuenta(tipo: String): void

+ solicitarPrestamo(monto: double): void

+ agregarTarjetaCredito(tarjeta: TarjetaCredito): void

+ verResumenFinanciero(): void

Uso: Se pueden crear objetos Cliente en el main() utilizando Scanner para ingresar datos

como nombre, cédula y tipo de cuenta.

4. Subclase: `Empleado extends Persona`

Responsabilidad: Gestionar productos y datos bancarios de clientes.

- Cajero

- BalconServicios

- JefeAgencia (Gerente)

Métodos sugeridos para Empleado:

+ autenticarEmpleado(usuario: String, clave: String): boolean

+ crearCuentaParaCliente(cliente: Cliente, tipo: String): void

+ modificarDatosCliente(cliente: Cliente): void

+ registrarPrestamo(cliente: Cliente, monto: double): void

+ cerrarCuenta(cliente: Cliente, cuenta: Cuenta): void

Métodos adicionales sugeridos por rol:

Para Cajero:

+ procesarRetiro(cliente: Cliente, monto: double): void

+ consultarSaldo(cliente: Cliente): void

+ procesarDeposito(cliente: Cliente, monto: double): void

Para BalconServicios:

+ registrarNuevoCliente(): Cliente

+ actualizarDatosCliente(cliente: Cliente): void

Para JefeAgencia:

+ aprobarPrestamo(cliente: Cliente, monto: double): boolean

+ generarReporteOperaciones(): void

+ evaluarEmpleado(empleado: Empleado): void

5. Flujo sugerido para una implementación con menú y objetos

1. Mostrar menú principal:

- Registrar cliente

- Ingresar como cliente

- Ingresar como empleado (cajero, balcón, jefe agencia)

2. Usar Scanner para capturar datos.

3. Crear objetos de tipo Cliente o Empleado según selección.

4. Permitir navegación entre funciones según el rol:

- Cliente: abrir cuenta, ver saldo, solicitar préstamo

- Cajero: retiro, depósito

- Balcón: registro, modificación de datos

- Jefe de Agencia: aprobación y reportes

5. Aplicar encapsulamiento (private, protected) correctamente para proteger los atributos.

La implementación con Scanner y menús mejora la interacción con el usuario y refuerza el

entendimiento práctico del tema.

Subir al Git hub – código e informe - estar listo para Exponer

```
----Bienvenido al sistema Bancario----  
1. Ingresar como cliente  
2. Ingresar como empleado  
3. Registrar cliente  
4. Registrar empleado  
5. Salir  
Elija una opcion: 1  
Ingrese su cedula: 178092938  
Cliente no registrado. Intentelo nuevamente.  
  
----Bienvenido al sistema Bancario----  
1. Ingresar como cliente  
2. Ingresar como empleado  
3. Registrar cliente  
4. Registrar empleado  
5. Salir  
Elija una opcion: 2  
Ingrese su cedula: 182719983  
Empleado no registrado.
```

```
----Bienvenido al sistema Bancario----
```

1. Ingresar como cliente
2. Ingresar como empleado
3. Registrar cliente
4. Registrar empleado
5. Salir

```
Elija una opcion: 3
```

```
--- Registrar Cliente ---
```

```
Nombre: Claudia
```

```
Cedula: 1726418629
```

```
Direccion: mi casa
```

```
Telefono: 0998569412
```

```
Tipo de cuenta: Ahorros
```

```
Ahorros iniciales: 80
```

```
Cliente registrado exitosamente.
```

```
----Bienvenido al sistema Bancario----
```

1. Ingresar como cliente
2. Ingresar como empleado
3. Registrar cliente
4. Registrar empleado
5. Salir

```
Elija una opcion: 1
```

```
Ingrese su cedula: 1724418629
```

```
Cliente no registrado. Intentelo nuevamente.
```

```
----Bienvenido al sistema Bancario----
```

1. Ingresar como cliente
2. Ingresar como empleado
3. Registrar cliente
4. Registrar empleado
5. Salir

```
Elija una opcion: 1
```

```
Ingrese su cedula: 1726418629
```

```
Bienvenido Claudia
```

```
Tipo de cuenta: Ahorros
```

```
Ahorros: 80.0
```

```
----Bienvenido al sistema Bancario----
```

1. Ingresar como cliente
2. Ingresar como empleado
3. Registrar cliente
4. Registrar empleado
5. Salir

```
Elija una opcion: 2
```

```
Ingrese su cedula: 1726416899
```

```
Empleado no registrado.
```

```
----Bienvenido al sistema Bancario----
```

1. Ingresar como cliente
2. Ingresar como empleado
3. Registrar cliente
4. Registrar empleado
5. Salir

```
Elija una opcion: 4
```

```
--- Registrar Empleado ---
```

```
Nombre: Mateo
```

```
Cedula: 1726476997
```

```
Direccion: su casa
```

```
Telefono: 0999788
```

```
Puesto:
```

1. Cajero
2. Balcon de servicio
3. Gerente

```
Elija una opcion: 3
```

```
Sucursal: quito
```

```
Empleado registrado exitosamente.
```

```

----Bienvenido al sistema Bancario----

1. Ingresar como cliente
2. Ingresar como empleado
3. Registrar cliente
4. Registrar empleado
5. Salir

Elija una opcion: 2

Ingrese su cedula: 1726476997

Bienvenido Mateo

Rol:

----Bienvenido al sistema Bancario----

1. Ingresar como cliente
2. Ingresar como empleado
3. Registrar cliente
4. Registrar empleado
5. Salir

Elija una opcion: 5

Opcion inexistente.

```

Gracias por usar el sistema.

Process finished with exit code 0

<https://github.com/AdrianMP16/Tarea-7-POO.git>

