

INGENIERIA DE SISTEMAS

TEC. DE PROG. ORIENTADA A OBJETOS



Práctica de Campo: Semana 2

INTEGRANTES: “GRUPO 11”

- | | |
|----------------------------------|-------------|
| • Gustavo Adolfo Picon Alvarez | - N00402043 |
| • Julio Ruben Nuevo Llanos | - N00400693 |
| • Kleider Irving Ramos Gutierrez | - N00399349 |
| • Angela Belen Santos Medina | - N00502327 |
| • Angel Samuel Chauca Huaylla | -N00502331 |

DOCENTE:

- Martín Eduardo Torres Rodríguez

ÍNDICE

| | |
|--|-----------|
| ACTIVIDADES ESPERADAS..... | 2 |
| I. CREACION DE REPOSITORIO REMOTO | 2 |
| II. CLONACION DEL REPOSITORIO EN LOCAL Y USO DE COMANDOS GIT STATUS, GIT LOG | 3 |
| III. PRIMER COMMIT Y USO DE GIT ADD, GIT COMMIT, LOG DECORATE | 3 |
| IV. VISUALIZACION DE HISTORIALS DE COMMITS | 4 |
| V. CREACION DE NUEVA RAMA - DEVELOP..... | 5 |
| VI. FUSIONAR RAMAS DEVELOP -> MAIN | 5 |
| VII. PUBLICAR EN UN REPOSITORIO COMPARTIDO | 5 |
| CASO 1: Lectura de datos simples con Scanner | 6 |
| I. CODIGO FUENTE..... | 6 |
| II. EJECUCION DEL PROGRAMA | 7 |
| III. COMANDO GIT USADO | 7 |
| CASO 2: Clase Estudiante con atributos privados | 8 |
| I. CODIGO FUENTE..... | 8 |
| II. EJECUCION DEL PROGRAMA | 10 |
| III. COMANDO GIT USADO | 10 |
| CASO 3: Clase CuentaBancaria con validación | 11 |
| I. CODIGO FUENTE..... | 11 |
| II. EJECUCION DEL PROGRAMA | 14 |
| III. COMANDO GIT USADO | 15 |

ACTIVIDADES ESPERADAS

I. CREACION DE REPOSITORIO REMOTO

Create a new repository


Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository](#).

Required fields are marked with an asterisk ().*

1


General

Owner *

 nae051293-create

Repository name *

POO-Git-Caso

 POO-Git-Caso is available.

Great repository names are short and memorable. How about [expert-memory](#)?

Description

Desarrollo de Casos Prácticos - POO y Git


41 / 350 characters

2

Configuration

Choose visibility *

Choose who can see and commit to this repository

 Public

Add README

READMEs can be used as longer descriptions. [About READMEs](#)

On ☒

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

No license

Create repository

II. CLONACION DEL REPOSITORIO EN LOCAL Y USO DE COMANDOS GIT STATUS, GIT LOG

```
BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace
$ git clone https://github.com/nae051293-create/P00-Git-Casos.git
Cloning into 'P00-Git-Casos'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace
$ ls
ClaseP00_semana_2/  P00-Git-Casos/

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace
$ cd P00-Git-Casos/

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/P00-Git-Casos (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/P00-Git-Casos (main)
$ git log
commit 4adc8efcecdfee5d56fcfa00e2c3f800736638a8 (HEAD -> main, origin/main, origin/HEAD)
Author: nae051293-create <nae051293@gmail.com>
Date:   Fri Oct 24 20:45:42 2025 -0500

    Initial commit
```

III. PRIMER COMMIT Y USO DE GIT ADD, GIT COMMIT, LOG DECORATE

```
BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/P00-Git-Casos (main)
$ git add .

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/P00-Git-Casos (main)
$ git commit -m "Primer commit: se agrega guia_git.md en el proyecto"
[main d3bd7f6] Primer commit: se agrega guia_git.md en el proyecto
1 file changed, 153 insertions(+)
create mode 100644 guia_git.md

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/P00-Git-Casos (main)
$ git log --oneline --graph --decorate
* d3bd7f6 (HEAD -> main) Primer commit: se agrega guia_git.md en el proyecto
* 4adc8ef (origin/main, origin/HEAD) Initial commit
```

```
BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/P00-Git-Casos
$ git push

Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.49 KiB | 1.49 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/nae051293-create/P00-Git-Casos.git
  4adc8ef..d3bd7f6  main -> main
```

IV. VISUALIZACION DE HISTORIALS DE COMMITS

```
USER@DESKTOP-MG9F408 MINGW64 ~/eclipse-workspace/P00_Git_Casos (main)
$ git log
commit 7c26b3a045c5a7ec5424dd804e0b9e45e98270c7 (HEAD -> main, origin/develop, develop)
Author: Angel <angel.19.leo@gmail.com>
Date:   Fri Oct 24 23:55:37 2025 -0500

    Caso 3: se crea programa para cuenta bancaria

commit 3d7b636f4c154a208f7a5cc2841fa5774a88c1fd
Author: Angel <angel.19.leo@gmail.com>
Date:   Fri Oct 24 23:25:31 2025 -0500

    Caso 2: se crea programa para estudiante

commit acd9dc1d6fdd04502f31c86d0ccde80ec3bcf828
Author: nae051293-create <nae051293@gmail.com>
Date:   Fri Oct 24 22:44:58 2025 -0500

    Caso 1: se crea programa que solicita datos con Scanner

feat:estructura base de los casos practicos git y P00

commit ac30718a3d14c2f03969329e9614c26bf5302c49
Author: nae051293-create <nae051293@gmail.com>
Date:   Fri Oct 24 22:05:07 2025 -0500

    docs: update readme and guia git

commit d3bd7f6e6ff7377bd462646503c12974e9707b48 (origin/main, origin/HEAD)
Author: DESKTOP-B239GNH\BELEN <belennae@gmail.com>
Date:   Fri Oct 24 21:02:37 2025 -0500

    Primer commit: se agrega guia_git.md en el proyecto

commit 4adc8efcecdfee5d56fcfa00e2c3f800736638a8
Author: nae051293-create <nae051293@gmail.com>
Date:   Fri Oct 24 20:45:42 2025 -0500

    Initial commit
```


V. CREACION DE NUEVA RAMA - DEVELOP

```
BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/POO-Git-Casos (main)
$ git branch develop

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/POO-Git-Casos (main)
$ git branch
  develop
* main

BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/POO-Git-Casos (main)
$ git checkout develop
Switched to branch 'develop'

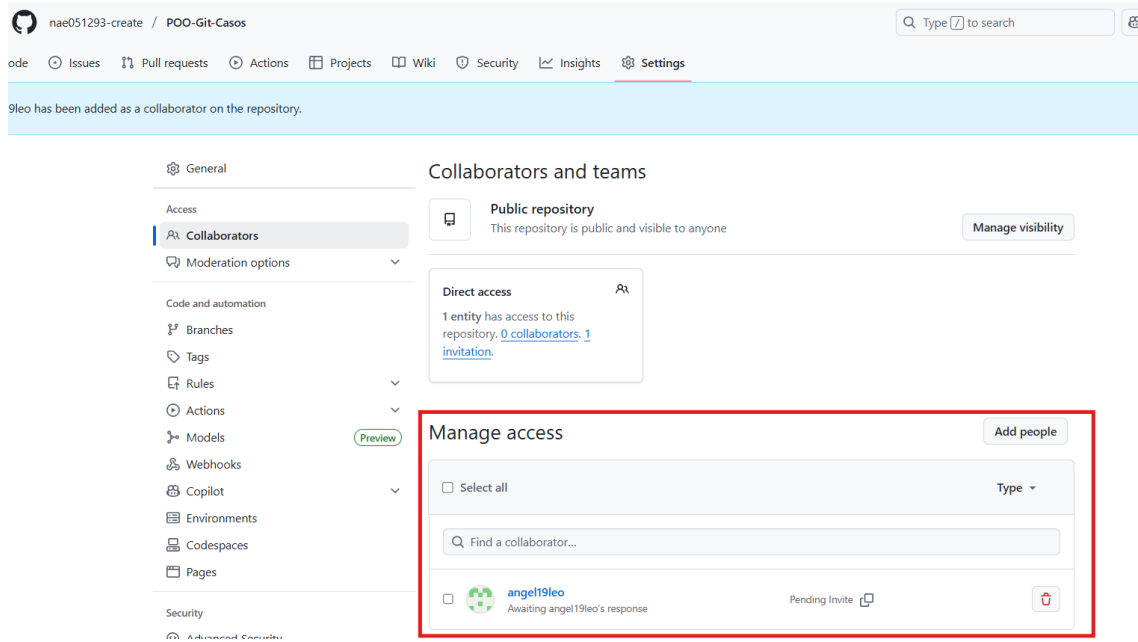
BELEN@DESKTOP-B239GNH MINGW64 ~/eclipse-workspace/POO-Git-Casos (develop)
$ git status
On branch develop
nothing to commit, working tree clean
```

VI. FUSIONAR RAMAS DEVELOP -> MAIN

```
USER@DESKTOP-MG9F408 MINGW64 ~/eclipse-workspace/POO_Git_Casos (main)
$ git merge develop
Updating d3bd7f6..7c26b3a
Fast-forward
 README.md | 87 ++++++
 bin/caso1/UsuarioSimple.class | Bin 0 -> 1428 bytes
 bin/caso2/EstudianteInteractivo.class | Bin 0 -> 418 bytes
 bin/caso3/CuentaBancaria.class | Bin 0 -> 397 bytes
 guia_git.md | 133 ++++++-----
 src/caso1/UsuarioSimple.java | 21 +++++
 src/caso2/EstudianteInteractivo.java | 78 ++++++
 src/caso3/CuentaBancaria.java | 103 ++++++
 8 files changed, 386 insertions(+), 36 deletions(-)
 create mode 100644 bin/caso1/UsuarioSimple.class
 create mode 100644 bin/caso2/EstudianteInteractivo.class
 create mode 100644 bin/caso3/CuentaBancaria.class
 create mode 100644 src/caso1/UsuarioSimple.java
 create mode 100644 src/caso2/EstudianteInteractivo.java
 create mode 100644 src/caso3/CuentaBancaria.java
```

VII. PUBLICAR EN UN REPOSITORIO COMPARTIDO

<https://github.com/nae051293-create/POO-Git-Casos.git>



CASO I: Lectura de datos simples con Scanner

I. CODIGO FUENTE

```
package caso1;

import java.util.Scanner;

public class UsuarioSimple {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Scanner sc = new Scanner(System.in);

        System.out.print("Ingrese su nombre: ");
        String nombre = sc.nextLine();

        System.out.print("Ingrese su edad: ");
        int edad = sc.nextInt();

        System.out.println("Hola " + nombre + ", tienes " + edad + " años.");
        sc.close();
    }
}
```


CASO 2: Clase Estudiante con atributos privados

I. CODIGO FUENTE

```
package caso2;

import java.util.Scanner;

public class EstudianteInteractivo {

    // Clase interna Estudiante con atributos privados
    static class Estudiante {
        private String nombre;
        private int edad;
        private String carrera;

        // Constructor
        public Estudiante(String nombre, int edad, String carrera) {
            this.nombre = nombre;
            this.edad = edad;
            this.carrera = carrera;
        }

        // Métodos Getters y Setters
        public String getNombre() {
            return nombre;
        }

        public void setNombre(String nombre) {
            this.nombre = nombre;
        }

        public int getEdad() {
            return edad;
        }

        public void setEdad(int edad) {
            this.edad = edad;
        }

        public String getCarrera() {
            return carrera;
        }
    }
}
```

```

    }

    public void setCarrera(String carrera) {
        this.carrera = carrera;
    }

    // Método para mostrar los datos del estudiante
    public void mostrarInformacion() {
        System.out.println("\n Información del Estudiante:");
        System.out.println("Nombre: " + nombre);
        System.out.println("Edad: " + edad);
        System.out.println("Carrera: " + carrera);
    }
}

// Método principal (main)
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.println("=== Registro de Estudiante ===");

    System.out.print("Ingrese el nombre del estudiante: ");
    String nombre = scanner.nextLine();

    System.out.print("Ingrese la edad: ");
    int edad = scanner.nextInt();
    scanner.nextLine(); // Limpiar buffer

    System.out.print("Ingrese la carrera: ");
    String carrera = scanner.nextLine();

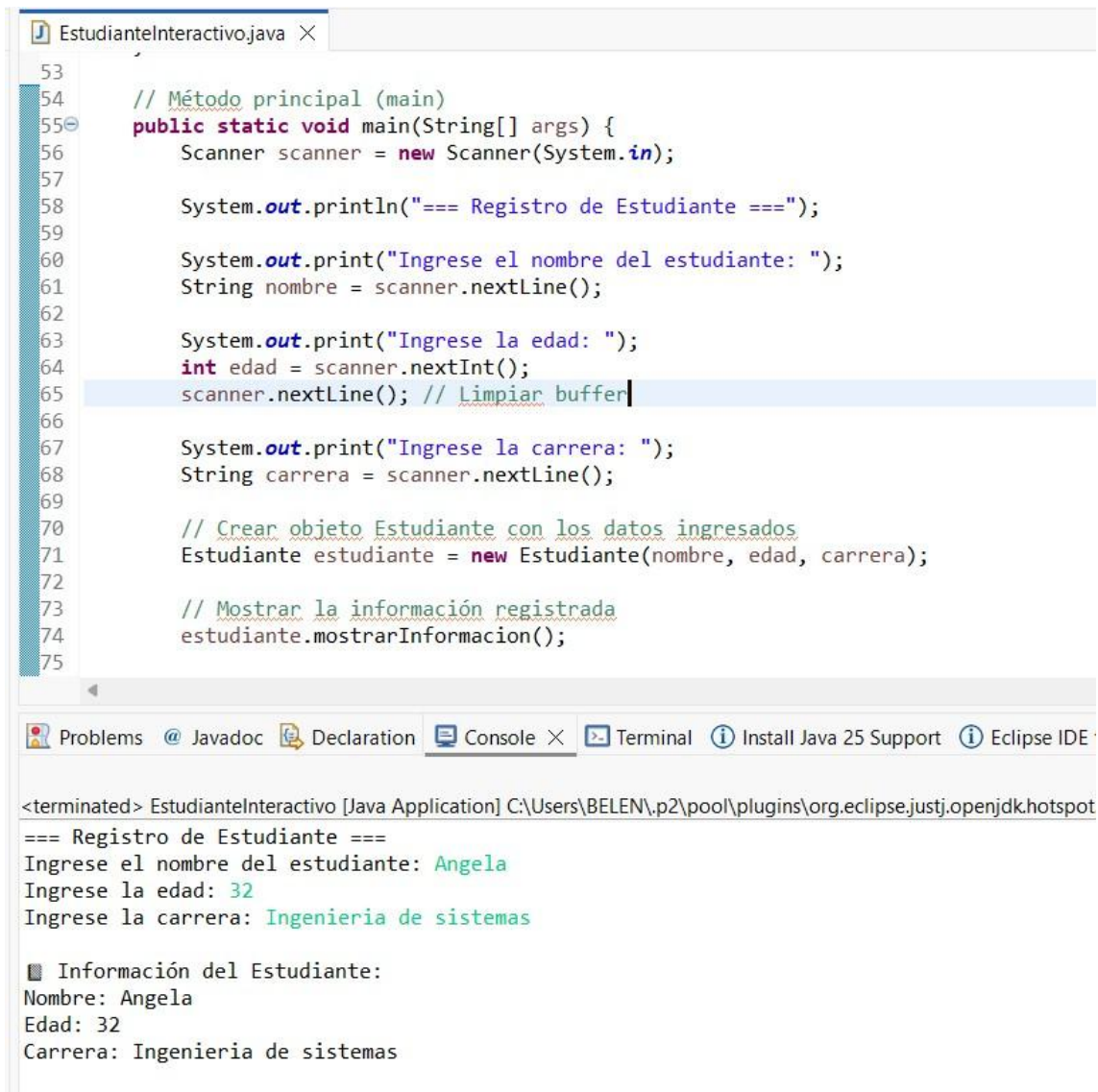
    // Crear objeto Estudiante con los datos ingresados
    Estudiante estudiante = new Estudiante(nombre, edad, carrera);

    // Mostrar la información registrada
    estudiante.mostrarInformacion();

    scanner.close();
}
}

```

II. EJECUCION DEL PROGRAMA



```

53
54 // Método principal (main)
55 public static void main(String[] args) {
56     Scanner scanner = new Scanner(System.in);
57
58     System.out.println("=== Registro de Estudiante ===");
59
60     System.out.print("Ingrese el nombre del estudiante: ");
61     String nombre = scanner.nextLine();
62
63     System.out.print("Ingrese la edad: ");
64     int edad = scanner.nextInt();
65     scanner.nextLine(); // Limpiar buffer
66
67     System.out.print("Ingrese la carrera: ");
68     String carrera = scanner.nextLine();
69
70     // Crear objeto Estudiante con los datos ingresados
71     Estudiante estudiante = new Estudiante(nombre, edad, carrera);
72
73     // Mostrar la información registrada
74     estudiante.mostrarInformacion();
75

```

Problems @ Javadoc Declaration Console × Terminal ⓘ Install Java 25 Support ⓘ Eclipse IDE

```

<terminated> EstudianteInteractivo [Java Application] C:\Users\BELEN\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
=== Registro de Estudiante ===
Ingrese el nombre del estudiante: Angela
Ingrese la edad: 32
Ingrese la carrera: Ingenieria de sistemas

■ Información del Estudiante:
Nombre: Angela
Edad: 32
Carrera: Ingenieria de sistemas

```

III. COMANDO GIT USADO

```

USER@DESKTOP-MG9F408 MINGW64 ~/eclipse-workspace/P00_Git_Casos (develop)
$ git status
On branch develop
Your branch is up to date with 'origin/develop'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   bin/caso2/EstudianteInteractivo.class
        modified:   src/caso2/EstudianteInteractivo.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .classpath
        .project
        .settings/
        bin/caso2/EstudianteInteractivo$Estudiante.class
        bin/module-info.class
        src/module-info.java

no changes added to commit (use "git add" and/or "git commit -a")

USER@DESKTOP-MG9F408 MINGW64 ~/eclipse-workspace/P00_Git_Casos (develop)
$ git add src/caso2/EstudianteInteractivo.java

USER@DESKTOP-MG9F408 MINGW64 ~/eclipse-workspace/P00_Git_Casos (develop)
$ git commit -m "Caso 2: se crea programa para estudiante"
[develop 3d7b636] Caso 2: se crea programa para estudiante
1 file changed, 72 insertions(+), 4 deletions(-)

```

CASO 3: Clase CuentaBancaria con validación

I. CODIGO FUENTE

```

package caso3;

import java.util.Scanner;

public class CuentaBancaria {

    // Atributos privados
    private String titular;
    private double saldo;

    // Constructor
    public CuentaBancaria(String titular, double saldoInicial) {
        this.titular = titular;
        this.saldo = saldoInicial;
    }
}

```

```

// Métodos getter
public String getTitular() {
    return titular;
}

public double getSaldo() {
    return saldo;
}

// Método para depositar dinero
public void depositar(double monto) {
    if (monto > 0) {
        saldo += monto;
        System.out.println("Depósito exitoso. Nuevo saldo: S/ " + saldo);
    } else {
        System.out.println("El monto a depositar debe ser mayor a 0.");
    }
}

// Método para retirar dinero (con validación)
public void retirar(double monto) {
    if (monto > 0) {
        if (monto <= saldo) {
            saldo -= monto;
            System.out.println("Retiro exitoso. Saldo restante: S/ " + saldo);
        } else {
            System.out.println("Fondos insuficientes. No se puede retirar más del saldo disponible.");
        }
    } else {
        System.out.println("El monto a retirar debe ser mayor a 0.");
    }
}

// Método para mostrar el saldo actual
public void mostrarSaldo() {
    System.out.println("Saldo actual de " + titular + ": S/ " + saldo);
}

// Método principal (main)
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

```



```

System.out.println("=== Sistema de Cuenta Bancaria ===");
System.out.print("Ingrese el nombre del titular: ");
String titular = scanner.nextLine();

System.out.print("Ingrese el saldo inicial: ");
double saldoInicial = scanner.nextDouble();

CuentaBancaria cuenta = new CuentaBancaria(titular, saldoInicial);

int opcion;
do {
    System.out.println("\n--- Menú de operaciones ---");
    System.out.println("1. Consultar saldo");
    System.out.println("2. Depositar");
    System.out.println("3. Retirar");
    System.out.println("4. Salir");
    System.out.print("Seleccione una opción: ");
    opcion = scanner.nextInt();

    switch (opcion) {
        case 1:
            cuenta.mostrarSaldo();
            break;
        case 2:
            System.out.print("Ingrese el monto a depositar: ");
            double deposito = scanner.nextDouble();
            cuenta.depositar(deposito);
            break;
        case 3:
            System.out.print("Ingrese el monto a retirar: ");
            double retiro = scanner.nextDouble();
            cuenta.retirar(retiro);
            break;
        case 4:
            System.out.println("Gracias por usar el sistema. ¡Hasta pronto!");
            break;
        default:
            System.out.println("Opción inválida. Intente nuevamente.");
    }
} while (opcion != 4);

scanner.close();

```

```
}
}
```

II. EJECUCION DEL PROGRAMA

```
<terminated> CuentaBancaria [Java Application] C:\Users\BELEN\p2\pool\plugins\org.eclip
```

```
=== 🏠 Sistema de Cuenta Bancaria ===
```

```
Ingrese el nombre del titular: Angel
```

```
Ingrese el saldo inicial: 45600
```

```
--- Menú de operaciones ---
```

```
1. Consultar saldo
```

```
2. Depositar
```

```
3. Retirar
```

```
4. Salir
```

```
Seleccione una opción: 1
```

```
🕒 Saldo actual de Angel: S/ 45600.0
```

```
--- Menú de operaciones ---
```

```
1. Consultar saldo
```

```
2. Depositar
```

```
3. Retirar
```

```
4. Salir
```

```
Seleccione una opción: 2
```

```
Ingrese el monto a depositar: 20
```

```
☒ Depósito exitoso. Nuevo saldo: S/ 45620.0
```

```
--- Menú de operaciones ---
```

```
1. Consultar saldo
```

```
2. Depositar
```

```
3. Retirar
```

```
4. Salir
```

```
Seleccione una opción: 3
```

```
Ingrese el monto a retirar: 10
```

```
☒ Retiro exitoso. Saldo restante: S/ 45610.0
```

```
--- Menú de operaciones ---
```

```
1. Consultar saldo
```

```
2. Depositar
```

```
3. Retirar
```

```
4. Salir
```

```
Seleccione una opción: 4
```

```
🕒 Gracias por usar el sistema. ¡Hasta pronto!
```

III. COMANDO GIT USADO

```
$ git status
On branch develop
Your branch is up to date with 'origin/develop'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   bin/caso2/EstudianteInteractivo.class
    modified:   bin/caso3/CuentaBancaria.class
    modified:   src/caso2/EstudianteInteractivo.java
    modified:   src/caso3/CuentaBancaria.java

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    .classpath
    .project
    .settings/
    bin/caso2/EstudianteInteractivo$Estudiante.class
    bin/module-info.class
    src/module-info.java
```

```
USER@DESKTOP-MG9F408 MINGW64 ~/eclipse-workspace/P00_Git_Casos (develop)
$ git add src/caso3/CuentaBancaria.java

USER@DESKTOP-MG9F408 MINGW64 ~/eclipse-workspace/P00_Git_Casos (develop)
$ git commit -m "Caso 3: se crea programa para cuenta bancaria"
[develop 7c26b3a] Caso 3: se crea programa para cuenta bancaria
 2 files changed, 98 insertions(+), 5 deletions(-)
```