



# TECNICAS DE PROGRAMA.ORIE. OBJ.

**UNIVERSIDAD PRIVADA DEL NORTE**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS COMPUTACIONALES**

**TEMA:**

PRACTICA DE CAMPO - 2

**CURSO:**

TECNICAS DE PROGRAMACION ORIENTADO A OBJETOS

**INTEGRANTES:**

GRUPO 03			
Estudiante 01	LOURDES SHARON HUAMAN MAMANI	ID	N00518813
Estudiante 02	MIGUEL JOHNNY HUARCAYA PINEDO	ID	N00244468
Estudiante 03	MANUEL ALBERTO MIRANDA ARONES	ID	N00443714
Estudiante 04	JOSE CARLOS SOSA BELLODAS	ID	

**PROFESOR:**

MARTIN EDUARDO TORRES RODRIGUEZ

## **INTRODUCCION**

El presente informe muestra el desarrollo de casos prácticos aplicando Programación Orientada a Objetos (POO) en Java, mediante el programa Eclipse y el uso de Git/GitHub como sistema de control y subida en el repositorio

El trabajo integra conceptos básicos de Java (clases, objetos, atributos, métodos) junto con las principales operaciones de Git (commits, ramas, fusiones y colaboración en repositorios compartidos).

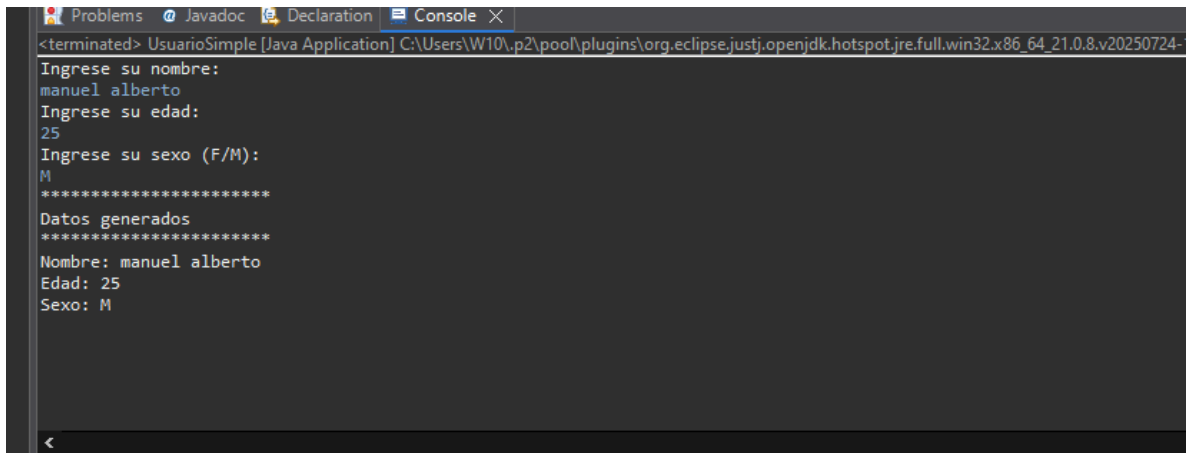
## **ACTIVIDADES REALIZADAS MEDIANTE GIT/GITHUB**

1. Desarrollar una guía de primeros pasos con git (guia\_git.md).
2. Crear un repositorio local y clonar uno desde github.
3. Navegación básica (git log como git status)
4. Primer commit y control de versiones como git add y git commit
5. Visualización del historial de ramas
6. Fusión de ramas
7. Publicación del proyecto en un repositorio
8. Documentación del uso de Git
9. Desarrollo de los casos de prácticas en Java.
10. Subida del código, guía y reporte final al repositorio.

## CASO 01

Desarrolla un programa en Java que solicite datos por teclado usando la clase Scanner. El programa debe ser guardado en la carpeta 'caso1/' y subido al repositorio con commits explicativos.

```
1 package caso1;
2 import java.util.Scanner; //se importa la biblioteca;
3
4 public class UsuarioSimple {
5
6     public static void main(String[] args) {
7         // TODO Auto-generated method stub
8         //creo el objeto scanner
9         Scanner sc = new Scanner(System.in);
10
11         //Pido los datos y antes de eso printeo un texto
12         System.out.println("Ingrese su nombre: ");
13         String nombre = sc.nextLine();
14
15         System.out.println("Ingrese su edad: ");
16         int edad = sc.nextInt();
17
18         System.out.println("Ingrese su sexo (F/M): ");
19         String sexo = sc.next();
20
21         sc.close();
22
23         //Muestro los datos capturados
24         System.out.println("*****");
25         System.out.println("Datos generados");
26         System.out.println("*****");
27         System.out.println("Nombre: "+ nombre);
28         System.out.println("Edad: "+ edad);
29         System.out.println("Sexo: "+ sexo);
30
31     }
32
33 }
34
```

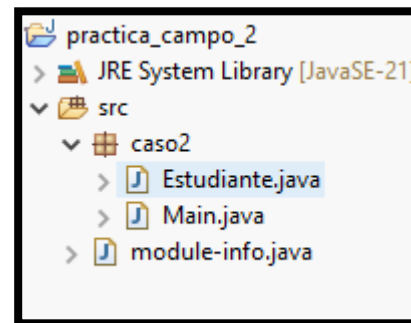


```
<terminated> UsuarioSimple [Java Application] C:\Users\W10\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-  
Ingrese su nombre:  
manuel alberto  
Ingrese su edad:  
25  
Ingrese su sexo (F/M):  
M  
*****  
Datos generados  
*****  
Nombre: manuel alberto  
Edad: 25  
Sexo: M  
<
```

## CASO 02

Implementar una clase Java llamada 'Estudiante' con atributos privados, constructor y métodos 'get/set'. El estudiante debe ser creado desde el método 'main()' y sus datos deben ser mostrados con uso de Scanner. Publica tu código en 'caso2/'.

- En este caso mi Package se llama caso2 con mis dos clases java que son Estudiante.java y Main.java.



- La clase Estudiante contiene sus atributos privados con sus métodos.

```
public class Estudiante {  
  
    private String nombre;  
    private int edad;  
    private String carrera;  
    private String telefono;  
    private String direccion;  
  
    public Estudiante(String nombre, int edad, String carrera,  
        this.nombre = nombre;  
        this.edad = edad;  
        this.carrera = carrera;  
        this.direccion=direccion;  
        this.telefono=telefono;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
}
```

- Los datos están mostrados con el uso de Scanner. Donde podemos ingresar los datos estudiantes.

```
public static void main(String[] args) {  
  
    Scanner obj= new Scanner(System.in);  
  
    System.out.print("Ingrese el nombre del estudiante: ");  
    String nombre = obj.nextLine();  
  
    System.out.print("Ingrese la edad del estudiante: ");  
    int edad = obj.nextInt();  
    obj.nextLine();  
  
    System.out.print("Ingrese la carrera del estudiante: ");  
    String carrera = obj.nextLine();  
  
    System.out.print("Ingrese el teléfono del estudiante: ");  
    String telefono = obj.nextLine();  
  
    System.out.print("Ingrese la dirección del estudiante: ");  
    String direccion = obj.nextLine();  
}
```

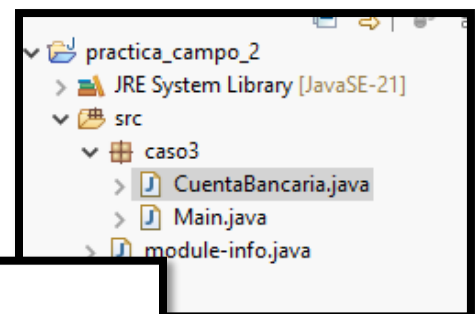
RESULTADO:

```
<terminated> Main [Java Application] C:\Users\LOURDES\Desktop\ECLIPSE\eci  
Ingrese el nombre del estudiante: Maria  
Ingrese la edad del estudiante: 20  
Ingrese la carrera del estudiante: Arte  
Ingrese el teléfono del estudiante: 013516459  
Ingrese la dirección del estudiante: Av. La molina 1278  
*** Datos del Estudiante ***  
Nombre: Maria  
Edad: 20  
Carrera: Arte  
Direccion: 013516459  
Telefono: Av. La molina 1278
```

## CASO 03

Consigna: Crea una clase Java llamada 'CuentaBancaria' que permita operaciones básicas como depositar y retirar, validando que no se retire más del saldo. Usa Scanner para el ingreso de datos. El código debe estar en la carpeta 'caso3/' con commits bien documentados.

- Caso 3 con dos clases java que son CuentaBancaria.java y Main.java.
- Clase Cuenta Bancaria.



```
package caso3;

public class CuentaBancaria {

    private String titular;
    private double saldo;

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

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

    public void retirar(double monto) {
        if (monto <= 0) {
            System.out.println("El monto a retirar debe ser positivo.");
        } else if (monto > saldo) {
            System.out.println("Fondos insuficientes. Saldo actual: " + saldo);
        } else {
            saldo -= monto;
            System.out.println("Retiro exitoso. Nuevo saldo: " + saldo);
        }
    }

    public double getSaldo() {
        return saldo;
    }

    public String getTitular() {
        return titular;
    }
}
```

## TECNICAS DE PROGRAMA.ORIE. OBJ.

- Realizado el uso de Scanner, donde se ingresa con el nombre del titular y poder realizar la consulta, deposito o retiro a su cuenta.

```
package caso3;

import java.util.Scanner;
public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

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

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

        CuentaBancaria cuenta = new CuentaBancaria(titular, saldoInicial);

        int opcion;
        do {
            System.out.println("\n--- Menú ---");
            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 = sc.nextInt();

            switch (opcion) {
                case 1:
                    System.out.println("Saldo actual: " + cuenta.getSaldo());
                    break;
                case 2:
                    System.out.print("Monto a depositar: ");
                    double montoDeposito = sc.nextDouble();
                    cuenta.depositar(montoDeposito);
                    break;
                case 3:
                    System.out.print("Monto a retirar: ");
                    double montoRetiro = sc.nextDouble();
                    cuenta.retirar(montoRetiro);
                    break;
                case 4:
                    System.out.println("Gracias por usar el sistema.");
                    break;
                default:
                    System.out.println("Opción no válida.");
            }
        } while (opcion != 4);
    }
}
```



- Resultado.

```
<terminated> New_configuration [Java Application] C:\Users\INTEL\p2\pool\plugins\org.eclipse.justi.openjdk  
Ingrese el nombre del titular: Miguel  
Ingrese el saldo inicial: 50  
  
--- Menú ---  
1. Consultar saldo  
2. Depositar  
3. Retirar  
4. Salir  
Seleccione una opción: 2  
Monto a depositar: 30  
Depósito exitoso. Nuevo saldo: 80.0  
  
--- Menú ---  
1. Consultar saldo  
2. Depositar  
3. Retirar  
4. Salir  
Seleccione una opción: 4  
Gracias por usar el sistema.
```

## CONCLUSIÓN

El uso de Git y GitHub permiti ó organizar mejor el desarrollo del proyecto, llevar un historial de cambios y facilitar la colaboraci ó n.

## LINK GITHUB:

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