

# INGENIERIA DE SISTEMAS COMPUTACIONALES

**CURSO:**

TÉCNICAS DE PROGRAM. ORIE. OBJ.

**DOCENTE:**

MARTIN EDUARDO TORRES RODRIGUEZ

**TEMA:**

PRACTICA DE CAMPO 2

**ALUMNO:**

JONATHAN ALFREDO QUEZADA LOPEZ N00405527



2024

1. Crear repositorio local y clonar uno desde GitHub.

- CREACION DE REPOSITORIO LOCAL

```
jonat@Jonathan MINGW64 ~/Documentos (master)
$ git init Proyectos-2025
Initialized empty Git repository in C:/Users/jonat/Documentos/Proyectos-2025/.git/
```

- CLONAR REPOSITORIO DESDE GITHHUB

```
jonat@Jonathan MINGW64 ~/Documentos (master)
$ git clone https://github.com/jonathanlopez647-source/P00-Git-Casos
Cloning into 'P00-Git-Casos'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

2. Navegación básica por el repositorio ('status', 'log').

- STATUS

```
jonat@Jonathan MINGW64 ~/Documentos/P00-Git-Casos (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

- LOG

```
jonat@Jonathan MINGW64 ~/Documentos/P00-Git-Casos (main)
$ git log
commit 774a873a4666ca8d5f80b9f4efe27be00f87955b (HEAD -> main, origin/main, origin/HEAD)
Author: jonathanlopez647-source <jonathanlopez647@gmail.com>
Date:   Wed Nov 5 20:25:50 2025 -0500

    Initial commit

jonat@Jonathan MINGW64 ~/Documentos/P00-Git-Casos (main)
$ git log --oneline
774a873 (HEAD -> main, origin/main, origin/HEAD) Initial commit
```

3. Primer commit, realizar cambios y guardar con `git add` y `git commit`.

- GIT ADD

```
jonat@Jonathan MINGW64 ~/Documents/P00-Git-Casos (main)
$ git status
On branch main

No commits yet

Changes to be committed:
(use "git rm --cached <file>..." to unstage)
    new file:   Caso1/UsuarioSimple.java
    new file:   Caso2/EstudianteInteractivo.java
    new file:   Caso3/CuentaBancaria.java
```

- GIT COMMIT

```
jonat@Jonathan MINGW64 ~/Documents/POO-Git-Casos (main)
$ git commit -m "Primer commit: agregar archivo guia_git.md"
[main (root-commit) 6abf6a] Primer commit: agregar archivo guia_git.md
  3 files changed, 10 insertions(+)
  create mode 100644 Caso1/UsuarioSimple.java
  create mode 100644 Caso2/EstudianteInteractivo.java
  create mode 100644 Caso3/CuentaBancaria.java
```

4. Visualización del historial de commits.

```
jonat@Jonathan MINGW64 ~/Documents/POO-Git-Casos (main)
$ git log
commit 6abf6ad0fdcb27e576a18440ffe04ff72fda813 (HEAD -> main)
Author: Jonathan <jonathanlopez64@gmail.com>
Date:   Thu Nov 6 21:27:07 2025 -0500

  Primer commit: agregar archivo guia_git.md
```

5. Crear ramas (`git branch`) y cambiar entre ellas (`git checkout`).

- GIT BRANCH

```
jonat@Jonathan MINGW64 ~/Documents/POO-Git-Casos (main)
$ git branch nueva-rama

jonat@Jonathan MINGW64 ~/Documents/POO-Git-Casos (main)
$ git branch nueva-rama2
```

- GIT CHECHOUT

```
jonat@Jonathan MINGW64 ~/Documents/POO-Git-Casos (main)
$ git checkout nueva-rama
Switched to branch 'nueva-rama'

jonat@Jonathan MINGW64 ~/Documents/POO-Git-Casos (nueva-rama)
$
```

6. Fusionar ramas (`git merge`).

```
jonat@Jonathan MINGW64 ~/Documents/POO-Git-Casos (main)
$ git merge nueva-rama
Already up to date.
```

7. Publicar en un repositorio compartido.

- Nos conectamos a donde están alojados nuestros proyectos e inicializamos

```
jonat@Jonathan MINGW64 ~ (master)
$ cd Casos

jonat@Jonathan MINGW64 ~/Casos (master)
$ git init
Reinitialized existing Git repository in C:/Users/jonat/Casos/.git/

jonat@Jonathan MINGW64 ~/Casos (master)
$ git add .
```

- Creamos un commit para identificar la subida de los archivos al repositorio

```
jonat@Jonathan MINGW64 ~/Casos (master)
$ git commit -m "Subiendo desarrollo de casos"
On branch master
nothing to commit, working tree clean
```

- Nos conectamos al repositorio remoto y enviamos los cambios

```

jonat@Jonathan MINGW64 ~/Casos (master)
$ git remote add origin https://github.com/jonathanlopez647-source/POO-Git-Casos

jonat@Jonathan MINGW64 ~/Casos (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 6 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1.65 KiB | 1.65 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/jonathanlopez647-source/POO-Git-Casos/pull/new/m
aster
remote:
To https://github.com/jonathanlopez647-source/POO-Git-Casos
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

```

## 8. Desarrollo de casos

- CASO1

```

Caso1 > J UsuarioSimple.java > UsarioSimple
  5  public class UsarioSimple {
  7    public static void main(String[] args) {
14      System.out.print(s: "Ingrese su edad: ");
15      int edad = scanner.nextInt();
16
17      scanner.nextLine();
18
19      System.out.print(s: "Ingrese su ciudad de residencia: ");
20      String ciudad = scanner.nextLine();
21
22      System.out.println(x: "\nDatos ingresados:");
23      System.out.println("Nombre: " + nombre);
24      System.out.println("Edad: " + edad);
25      System.out.println("Ciudad: " + ciudad);
26
27      scanner.close();
28  }
29

```

PROBLEMAS	SALIDA	CONSOLA DE DEPURACIÓN	<b>TERMINAL</b>	PUERTOS
			PS D:\TÉCNICAS DE PROGRAM.ORIE. OBJ\POO-Git-Casos> & 'C:\Program Files\Java\jdk-25\bin\java.exe review' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\jonat\AppData\Roaming\Code\U\bin' 'Caso1.UsuarioSimple'	
			Ingrese su nombre: JONATHAN	
			Ingrese su edad: 30	
			Ingrese su ciudad de residencia: TRUJILLO	
			Datos ingresados:	
			Nombre: JONATHAN	
			Edad: 30	
			Ciudad: TRUJILLO	
			PS D:\TÉCNICAS DE PROGRAM.ORIE. OBJ\POO-Git-Casos>	

- CASO 2

```
1 package Caso2;
2
3 import java.util.Scanner;
4
5 public class EstudianteInteractivo {
6     private String nombre;
7     private int edad;
8     private String carrera;
9
10    public EstudianteInteractivo(String Nombre, int Edad, String Carrera) {
11        this.nombre = Nombre;
12        this.edad = Edad;
13        this.carrera = Carrera;
14    }
15
16    public String getNombre() {
17        return nombre;
18    }
19
20    public void setNombre(String nombre) {
21        this.nombre = nombre;
22    }
23
24    public int getEdad() {
25        return edad;
26    }
27
28    public void setEdad(int edad) {
29        this.edad = edad;
30    }
31
32    public String getCarrera() {
33        return carrera;
34    }
35
36    public void setCarrera(String carrera) {
37        this.carrera = carrera;
38    }
39
```

```

40     public void mostrarDatos() {
41         System.out.println("Nombre: " + nombre);
42         System.out.println("Edad: " + edad);
43         System.out.println("Carrera: " + carrera);
44     }
45
46     Run | Debug
47     public static void main(String[] args) {
48         Scanner scanner = new Scanner(System.in);
49
50         System.out.print(s: "Ingrese el nombre del estudiante: ");
51         String nombre = scanner.nextLine();
52
53         System.out.print(s: "Ingrese la edad del estudiante: ");
54         int edad = scanner.nextInt();
55         scanner.nextLine(); // Limpiar el buffer
56
57         System.out.print(s: "Ingrese la carrera del estudiante: ");
58         String carrera = scanner.nextLine();
59
60         EstudianteInteractivo estudiante = new EstudianteInteractivo(nombre, edad, carrera);
61
62         System.out.println(x: "\nDatos del Estudiante:");
63         estudiante.mostrarDatos();
64
65         // Cerrar el scanner
66     }
67 }
```

PROBLEMAS	SALIDA	CONSOLA DE DEPURACIÓN	TERMINAL	PUERTOS
			PS D:\TÉCNICAS DE PROGRAM.ORIE. OBJ\POO-Git-Casos> & 'C:\Program Files\Java\jdk-25\bin\java.exe' '-Djava.awt.headless=true' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\jonat\AppData\Roaming\jdt_ws\POO-Git-Casos_f5dfae4a\bin' 'Caso2.EstudanteInteractivo'	
			Ingrese el nombre del estudiante: JONATHAN	
			Ingrese la edad del estudiante: 30	
			Ingrese la carrera del estudiante: INGENIERIA DE SISTEMAS COMPUTACIONALES	
			Datos del Estudiante:	
			Nombre: JONATHAN	
			Edad: 30	
			Carrera: INGENIERIA DE SISTEMAS COMPUTACIONALES	
			PS D:\TÉCNICAS DE PROGRAM.ORIE. OBJ\POO-Git-Casos>	

- CASO 3

```
1 package Caso3;
2
3 import java.util.Scanner;
4
5 public class CuentaBancaria {
6     private String titular;
7     private double saldo;
8
9     public CuentaBancaria(String titular, double saldoInicial) {
10         this.titular = titular;
11         this.saldo = saldoInicial;
12     }
13
14     public void depositar(double cantidad) {
15         if (cantidad > 0) {
16             saldo += cantidad;
17             System.out.println("Depósito exitoso. Saldo actual: " + saldo);
18         } else {
19             System.out.println("La cantidad a depositar debe ser mayor que 0.");
20         }
21     }
22
23     public void retirar(double cantidad) {
24         if (cantidad > 0 && cantidad <= saldo) {
25             saldo -= cantidad;
26             System.out.println("Retiro exitoso. Saldo actual: " + saldo);
27         } else if (cantidad > saldo) {
28             System.out.println("No hay suficiente saldo para realizar el retiro.");
29         } else {
30             System.out.println("La cantidad a retirar debe ser mayor que 0.");
31         }
32     }
33
34     public void mostrarSaldo() {
35         System.out.println("Saldo actual: " + saldo);
36     }
37
38     public void mostrarDatos() {
39         System.out.println("Titular: " + titular);
40         mostrarSaldo();
41     }
}
```

```
Run | Debug
43 public static void main(String[] args) {
44     Scanner scanner = new Scanner(System.in);
45
46     System.out.print(s: "Ingrese el nombre del titular de la cuenta: ");
47     String titular = scanner.nextLine();
48     System.out.print(s: "Ingrese el saldo inicial: ");
49     double saldoInicial = scanner.nextDouble();
50
51     CuentaBancaria cuenta = new CuentaBancaria(titular, saldoInicial);
52
53     boolean salir = false;
54     while (!salir) {
55         System.out.println(x: "\n--- Menú ---");
56         System.out.println(x: "1. Depositar dinero");
57         System.out.println(x: "2. Retirar dinero");
58         System.out.println(x: "3. Ver saldo");
59         System.out.println(x: "4. Ver datos de la cuenta");
59         System.out.println(x: "5. Salir");
59         System.out.print(s: "Seleccione una opción: ");
59         int opcion = scanner.nextInt();
59
59         switch (opcion) {
59             case 1:
59                 System.out.print(s: "Ingrese la cantidad a depositar: ");
59                 double cantidadDepositar = scanner.nextDouble();
59                 cuenta.depositar(cantidadDepositar);
59                 break;
59             case 2:
59                 System.out.print(s: "Ingrese la cantidad a retirar: ");
59                 double cantidadRetirar = scanner.nextDouble();
59                 cuenta.retirar(cantidadRetirar);
59                 break;
59             case 3:
59                 cuenta.mostrarSaldo();
59                 break;
59             case 4:
59                 cuenta.mostrarDatos();
59                 break;
59             case 5:
59                 salir = true;
59                 System.out.println(x: "¡Hasta luego!");
59                 break;
59
59             default:
59                 System.out.println(x: "Opción no válida. Intente de nuevo.");
59                 break;
59         }
59     }
59     scanner.close();
59 }
```

```
& 'C:\Program Files\Java\jdk-25\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=onExceptionMessages' '-cp' 'C:\Users\jonat\AppData\Roaming\Code\User\workspaceStorage\5921cd9372922c8taBancaria'
Ingrese el nombre del titular de la cuenta: JONATHAN
Ingrese el saldo inicial: 1000

--- Menú ---
1. Depositar dinero
2. Retirar dinero
3. Ver saldo
4. Ver datos de la cuenta
5. Salir
Seleccione una opción: 1
Ingrese la cantidad a depositar: 2000
Depósito exitoso. Saldo actual: 3000.0

--- Menú ---
1. Depositar dinero
2. Retirar dinero
3. Ver saldo
4. Ver datos de la cuenta
5. Salir
Seleccione una opción: 3
Saldo actual: 3000.0

--- Menú ---
1. Depositar dinero
2. Retirar dinero
3. Ver saldo
4. Ver datos de la cuenta
5. Salir
Seleccione una opción: 1
```