



# PRACTICA DE CAMPO 2

TÉCNICAS DE PROGRAM. ORIE. OBJ

Profesor:

Martin Eduardo Torres Rodríguez

Alumno:

Brandon Smith Aranda Tacza

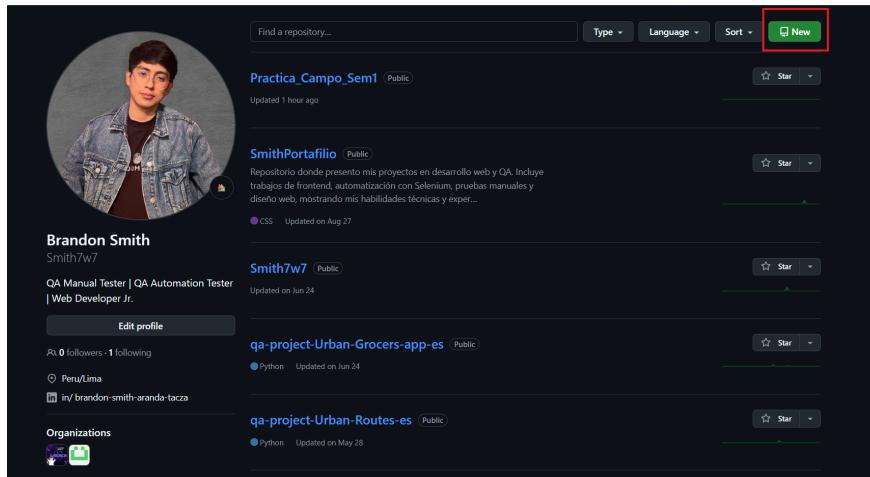
2025

## Practica de Campo Semana 2

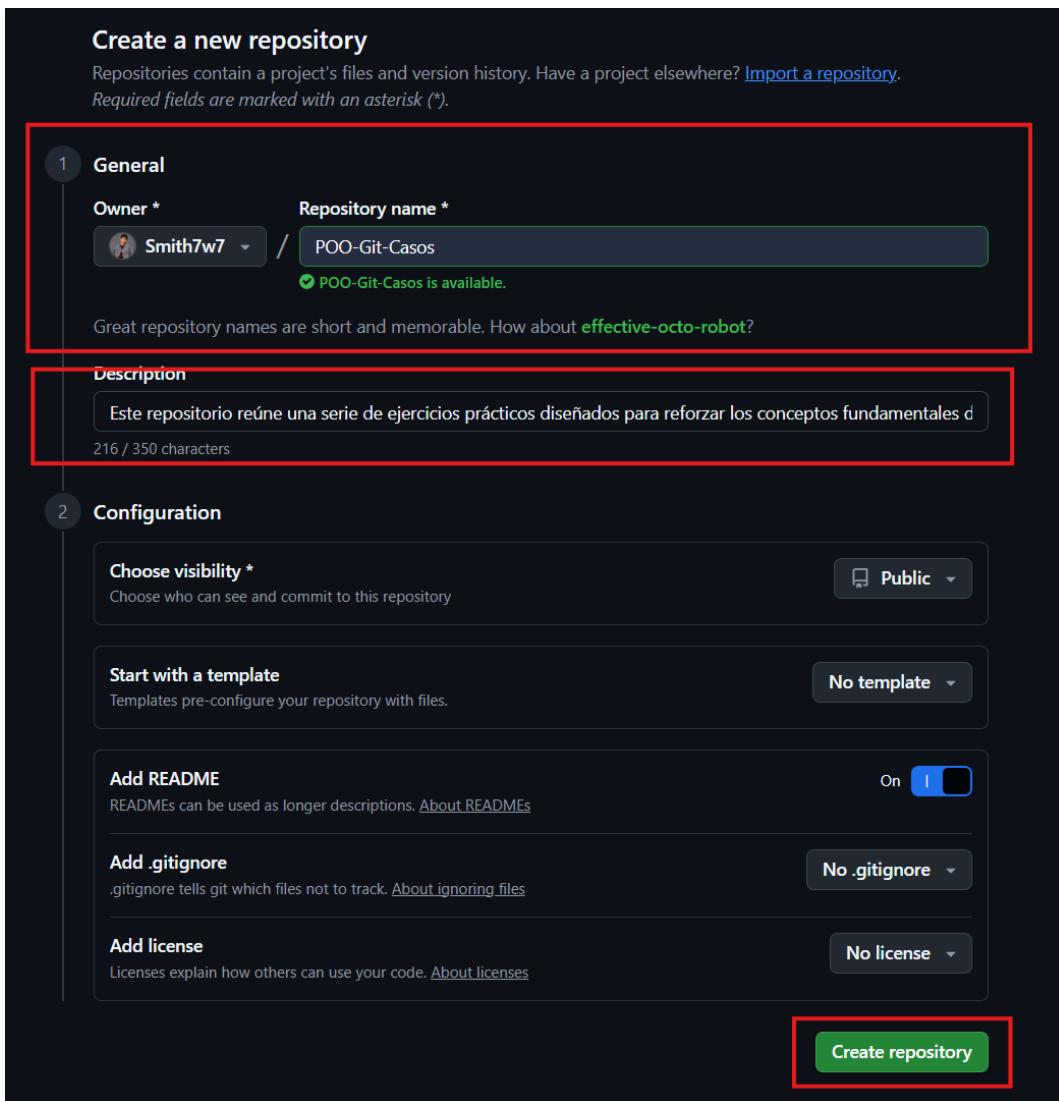
Repository GitHub: <https://github.com/Smith7w7/POO-Git-Casos>

### 1. Creación repositorio en GitHub

Seleccionamos la opción “NEW” para iniciar el proceso de creación de un nuevo repositorio en GitHub.



Se asigna un nombre al repositorio, una descripción opcional y posteriormente se hace clic en el botón “Create Repository” para completar su creación.



**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 \*** Smith7w7 / **Repository name \*** POO-Git-Casos  POO-Git-Casos is available.

Great repository names are short and memorable. How about [effective-octo-robot?](#)

**Description**

Este repositorio reúne una serie de ejercicios prácticos diseñados para reforzar los conceptos fundamentales d  
216 / 350 characters

**2 Configuration**

**Choose visibility \*** Public

**Start with a template** No template

**Add README** On

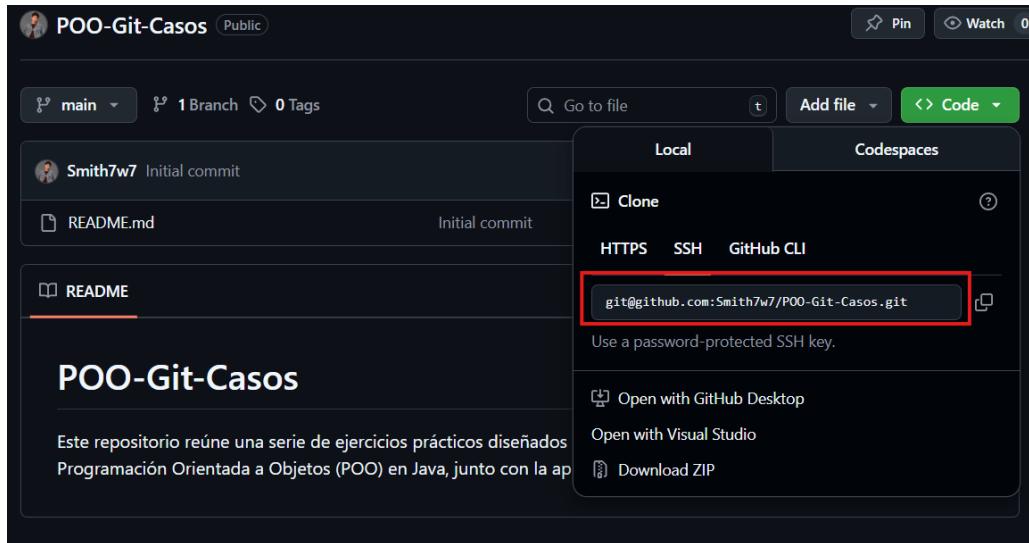
**Add .gitignore** No .gitignore

**Add license** No license

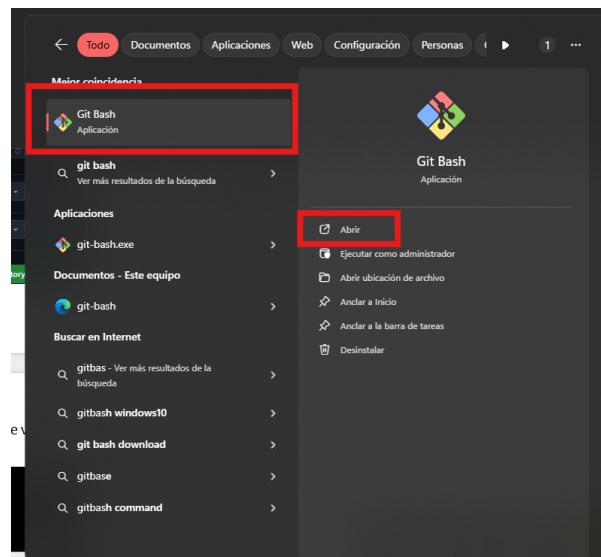
**Create repository**

## 2. Clonar repositorio GitHub en maquina local

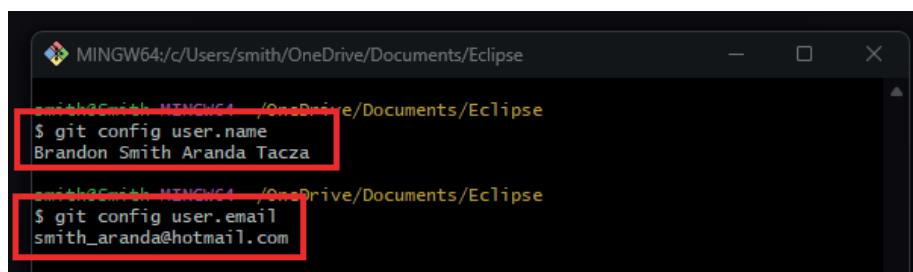
Se copia la URL del repositorio generada en GitHub para su posterior utilización.



Se inicia el programa Git Bash.



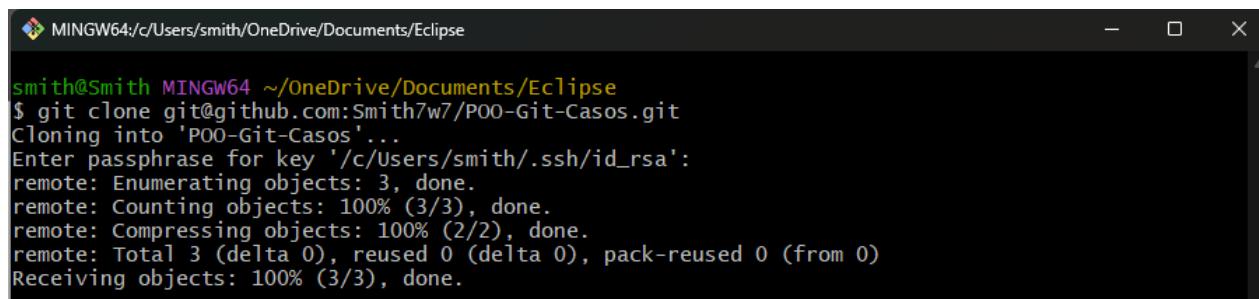
Configurar tu sesión de GitHub previamente para que Git sepa de donde viene el repositorio



```
MINGW64:/c/Users/smith/OneDrive/Documents/Eclipse
$ git config user.name
Brandon Smith Aranda Tacza
MINGW64:/c/Users/smith/OneDrive/Documents/Eclipse
$ git config user.email
smith_aranda@hotmail.com
```

The screenshot shows a terminal window with a dark background and light-colored text. It displays two commands being run: '\$ git config user.name' followed by 'Brandon Smith Aranda Tacza', and '\$ git config user.email' followed by 'smith\_aranda@hotmail.com'. Both of these command lines are highlighted with red boxes.

Se ejecuta el comando git clone "URL\_del\_repositorio\_GitHub" para clonar el repositorio remoto en el entorno local.



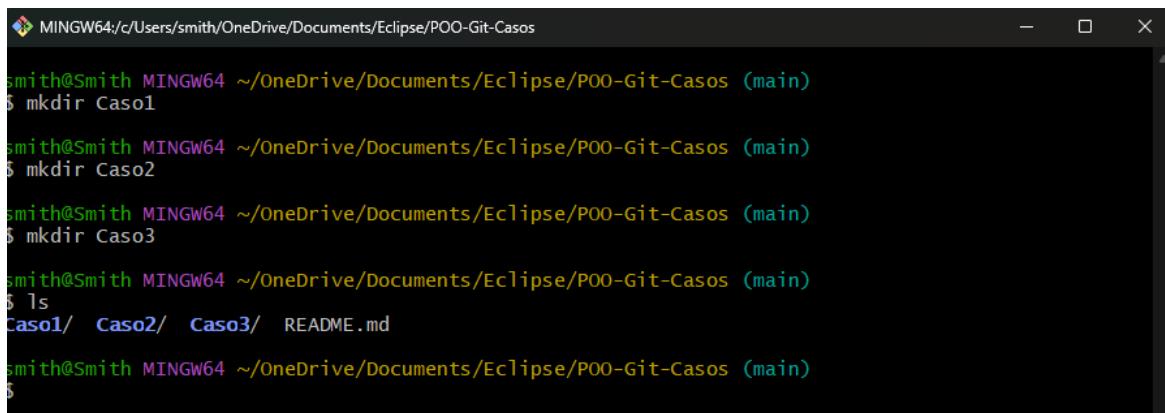
```
MINGW64:/c/Users/smith/OneDrive/Documents/Eclipse
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse
$ git clone git@github.com:Smith7w7/POO-Git-Casos.git
Cloning into 'POO-Git-Casos'...
Enter passphrase for key '/c/Users/smith/.ssh/id_rsa':
remote: Enumerating objects: 3, done.
remote: Counting objects: 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.
```

Se verifica que el repositorio haya sido clonado correctamente en el entorno local.

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
POO-Git-Casos	✓	26/10/2025 20:35	Carpeta de archivos	

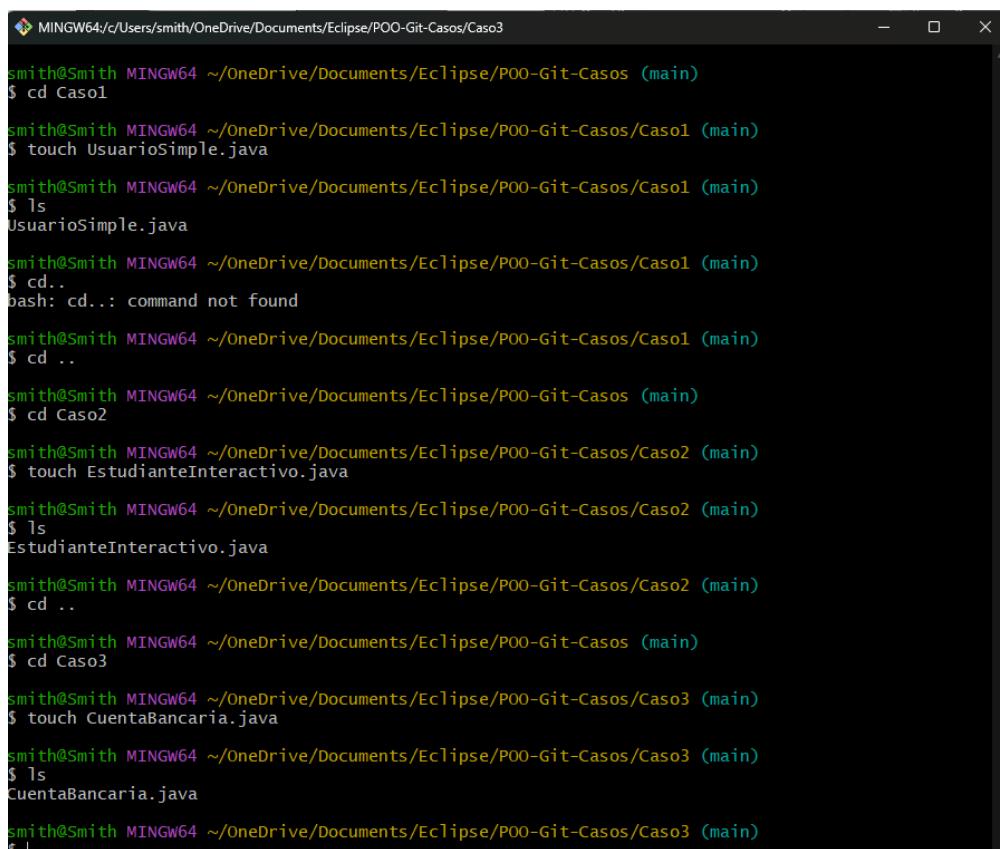
### 3. Primer Commit

Se crean las carpetas Caso1, Caso2 y Caso3, conforme a la estructura requerida en el proyecto.



```
MINGW64:/c/Users/smith/OneDrive/Documents/Eclipse/POO-Git-Casos
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ mkdir Caso1
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ mkdir Caso2
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ mkdir Caso3
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ ls
Caso1/ Caso2/ Caso3/ README.md
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$
```

Se crean los archivos .java segun la estructura establecida



```
MINGW64:/c/Users/smith/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso1
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ cd Caso1
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso1 (main)
$ touch UsuarioSimple.java
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso1 (main)
$ ls
UsuarioSimple.java
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso1 (main)
$ cd..
bash: cd..: command not found
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso1 (main)
$ cd ..
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ cd Caso2
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso2 (main)
$ touch EstudianteInteractivo.java
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso2 (main)
$ ls
EstudianteInteractivo.java
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso2 (main)
$ cd ..
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ cd Caso3
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso3 (main)
$ touch CuentaBancaria.java
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso3 (main)
$ ls
CuentaBancaria.java
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos/Caso3 (main)
$ |
```

Se efectúa la navegación dentro del repositorio local y se ejecuta el primer commit, registrando los cambios iniciales.

```
MINGW64:/c/Users smith/OneDrive/Documents/Eclipse/POO-Git-Casos
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ ls
Caso1/ Caso2/ Caso3/ README.md

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Caso1/
    Caso2/
    Caso3/

nothing added to commit but untracked files present (use "git add" to track)

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git add .

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git commit -m "Primer commit"
[main 74b615c] Primer commit
 3 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Caso1/UsuarioSimple.java
 create mode 100644 Caso2/EstudianteInteractivo.java
 create mode 100644 Caso3/CuentaBancaria.java

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git push
Enter passphrase for key '/c/Users/smith/.ssh/id_rsa':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 517 bytes | 258.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:Smith7w7/POO-Git-Casos.git
  b737689..74b615c main -> main

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git log
commit 74b615cd8619cc8b4e92646d4c58ef86b97bff8a (HEAD -> main, origin/main, origin/HEAD)
Author: Brandon Smith Aranda Tacza <smith_aranda@hotmail.com>
Date:   Sun Oct 26 20:49:19 2025 -0500

    Primer commit

commit b737689af7501869673511767b6c0f6329845e40
Author: Brandon Smith <72725197+Smith7w7@users.noreply.github.com>
Date:   Sun Oct 26 20:30:10 2025 -0500

    Initial commit

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
```

Se genera una nueva rama utilizando el comando branch y se cambia entre ellas con checkout, lo que permite desarrollar funcionalidades de manera independiente. Posteriormente, los cambios se combinan mediante la operación merge, integrando el trabajo en el repositorio principal.

```
MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git branch Rama2

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git checkout
Your branch is up to date with 'origin/main'.

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git checkout Rama2
Switched to branch 'Rama2'

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (Rama2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git pull
Enter passphrase for key '/c/Users.smith/.ssh/id_rsa':
Already up to date.

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git merge Rama2
Already up to date.

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git push
Enter passphrase for key '/c/Users.smith/.ssh/id_rsa':
Everything up-to-date

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ |
```

Se comprueba que los cambios estén correctamente sincronizados y actualizados en el repositorio remoto de GitHub.

The screenshot shows a GitHub repository page for 'POO-Git-Casos'. At the top, there are navigation buttons for 'Pin', 'Watch 0', 'Fork 0', and 'Star 0'. Below that, a dropdown menu shows 'main' (selected), '1 Branch', and '0 Tags'. A search bar contains 'Go to file' and a code switcher. On the right, there's an 'About' section with a detailed description of the repository's purpose: 'Este repositorio reúne una serie de ejercicios prácticos diseñados para reforzar los conceptos fundamentales de Programación Orientada a Objetos (POO) en Java, junto con la aplicación de control de versiones con Git.' Below this are sections for 'Readme', 'Activity', 'Stars 0', 'Watching 0', and 'Forks 0'. The 'Releases' section indicates 'No releases published' and 'Create a new release'. The 'Packages' section shows 'No packages published' and 'Publish your first package'. The main content area displays the repository's history with three commits from 'Smith7w7': 'Primer commit' (74b615c, 5 minutes ago), 'Caso1' (Primer commit, 5 minutes ago), 'Caso2' (Primer commit, 5 minutes ago), 'Caso3' (Primer commit, 5 minutes ago), and 'README.md' (Initial commit, 24 minutes ago). A large 'README' section contains the same descriptive text as the 'About' section.

## 4. Desarrollo de casos

### CASO 1: Lectura de datos simples con Scanner

The screenshot shows the Eclipse IDE interface with the Java code for reading simple data using the Scanner class. The code prompts the user for their name and age, then prints a greeting message. The terminal window below shows the execution of the program, where the user inputs 'Brandon Smith' and '24', and the program outputs 'Hola Brandon Smith, tienes 24 años.'

```
1 package caso1;
2
3 import java.util.Scanner;
4
5 public class UsuarioSimple {
6
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        System.out.print("Ingrese su nombre: ");
11        String nombre = sc.nextLine();
12
13        System.out.print("Ingrese su edad: ");
14        int edad = sc.nextInt();
15
16        System.out.println("Hola " + nombre + ", tienes " + edad + " años.");
17        sc.close();
18    }
19
20 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: UsuarioSimple + ×

```
PS C:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1> ^C
PS C:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1>
PS C:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1> c:; cd 'c:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1' & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smith\AppData\Roaming\Code\User\workspaceStorage\5e1c4c61750881cea7104c4d2e67815f\redhat.java\jdt_ws\Practica_Campo_Sem1_85919330\bin' 'caso1.UsuarioSimple'
Ingrese su nombre: Brandon Smith
Ingrese su edad: 24
Hola Brandon Smith, tienes 24 años.
PS C:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1>
```

### CASO 2: Clase Estudiante con atributos privados

The screenshot shows the Eclipse IDE interface with the Java code for a student class (EstudianteInteractivo) that uses private attributes and Getters/Setters. The code includes a nested static class Estudiante and a method to print student information. The terminal window below shows the execution of the program, which prints the student's details.

```
1 package caso2;
2
3 import java.util.Scanner;
4
5 public class EstudianteInteractivo {
6
7     // Clase interna Estudiante con atributos privados
8     static class Estudiante {
9         private String nombre;
10        private int edad;
11        private String carrera;
12
13        // Constructor
14        public Estudiante(String nombre, int edad, String carrera) {
15            this.nombre = nombre;
16            this.edad = edad;
17            this.carrera = carrera;
18        }
19
20        // Métodos Getters y Setters
21        public String getNombre() {
22            return nombre;
23        }
24
25        public void setNombre(String nombre) {
26            this.nombre = nombre;
27        }
28
29        public int getEdad() {
30            return edad;
31        }
32
33        public void setEdad(int edad) {
34            this.edad = edad;
35        }
36
37        public String getCarrera() {
38            return carrera;
39        }
40
41        public void setCarrera(String carrera) {
42            this.carrera = carrera;
43        }
44
45        // Método para mostrar los datos del estudiante
46        public void mostrarInformacion() {
47            System.out.println("\n Información del Estudiante:");
48            System.out.println("Nombre: " + nombre);
49            System.out.println("Edad: " + edad);
50            System.out.println("Carrera: " + carrera);
51        }
52    }
53 }
```

```

53  // Método principal (main)
54  Run | Debug
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
76      scanner.close();
77  }
78
79 
```

PROBLEMS ① OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

.EstudanteInteractive'
--- Registro de Estudiante ---
Ingrese el nombre del estudiante: Brandon
Ingrese la edad: 24
Ingrese la carrera: Ingenieria de Sistema

  Información del Estudiante:
Nombre: Brandon
Edad: 24
Carrera: Ingenieria de Sistema
PS C:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1> 
```

## CASO 3: Clase CuentaBancaria con validación

```

caso3 > J MainJava > ...
1 package caso3;
2
3 import java.util.Scanner;
4
5 class CuentaBancaria {
6     private double saldo;
7     // Método para depositar dinero
8     public void depositar(double monto) {
9         saldo += monto;
10    }
11    // Método para retirar dinero
12    public void retirar(double monto) {
13        if (monto <= saldo) {
14            saldo -= monto;
15        } else {
16            System.out.println("Fondos insuficientes."); // Mensaje si no hay saldo suficiente
17        }
18    }
19    // Método para consultar el saldo
20    public double getSaldo() {
21        return saldo;
22    }
23 }
24
25 public class Main {
26     Run | Debug
27     public static void main(String[] args) {
28         Scanner sc = new Scanner(System.in);
29         CuentaBancaria cuenta = new CuentaBancaria();
30         System.out.print("Monto a depositar: ");
31         cuenta.depositar(sc.nextDouble()); // Guarda y deposita el monto
32         System.out.print("Monto a retirar: ");
33         cuenta.retirar(sc.nextDouble()); // Guarda y retira el monto
34         System.out.println("Saldo final: " + cuenta.getSaldo());
35         sc.close();
36     }
37 }
```

PROBLEMS ① OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1> & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\smith\AppData\Roaming\Code\User\workspaceStorage\5e1c4c61750881cea7104c4d2e67815f\redhat.java\jdt_ws\Practica_Campo_Sem1_85919330\bin' 'caso3.Main'
Monto a depositar: 200
Monto a retirar: 100
Saldo final: 100.0
PS C:\Users\smith\OneDrive\Documents\Eclipse\Practica_Campo_Sem1> 
```

Tras completar la codificación y colocar los archivos en la ruta adecuada del entorno local, se actualiza el repositorio remoto en GitHub utilizando los comandos correspondientes del control de versiones.

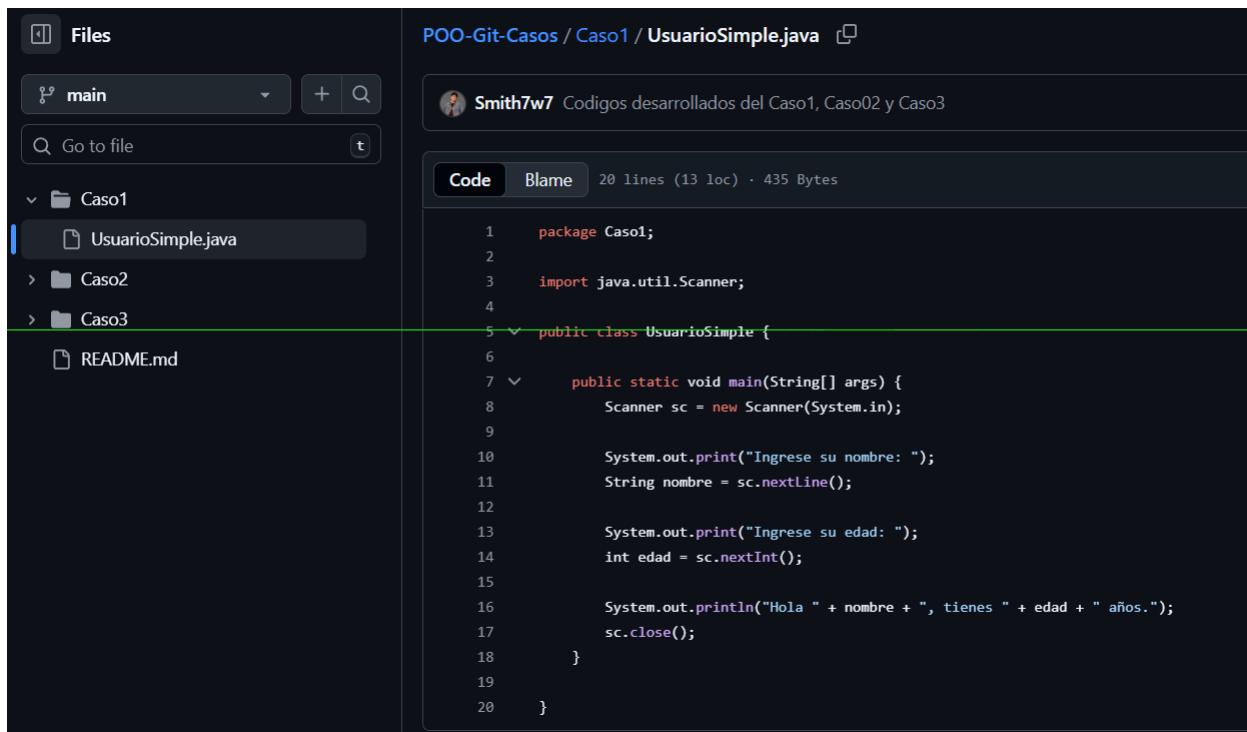
```
MINGW64:/c/Users smith/OneDrive/Documents/Eclipse/POO-Git-Casos
smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git add .

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git commit -m "Códigos desarrollados del Caso1, Caso02 y Caso3"
[main a4e351d] Códigos desarrollados del Caso1, Caso02 y Caso3
 4 files changed, 134 insertions(+)
 delete mode 100644 Caso3/CuentaBancaria.java
 create mode 100644 Caso3/Main.java

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git push
Enter passphrase for key '/c/Users/smith/.ssh/id_rsa':
Enumerating objects: 14, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (8/8), 1.72 KiB | 879.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:Smith7w7/POO-Git-Casos.git
    74b615c..a4e351d main -> main

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$
```

Verificar la correcta sincronización y actualización de los cambios



Se edita el archivo README al repositorio y se registra una descripción

The screenshot shows a GitHub repository interface. At the top, there are buttons for 'main' branch, '1 Branch', '0 Tags', 'Go to file', 'Add file', and 'Code'. Below this is a list of commits:

Author	Commit Message	Time Ago	
Smith7w7	Revise README structure and add case descriptions	d6ed527 · 1 minute ago	
	Caso1	Codigos desarrollados del Caso1, Caso02 y Caso3	6 minutes ago
	Caso2	Codigos desarrollados del Caso1, Caso02 y Caso3	6 minutes ago
	Caso3	Codigos desarrollados del Caso1, Caso02 y Caso3	6 minutes ago
	README.md	Revise README structure and add case descriptions	1 minute ago

Below the commits is the 'README' file content:

```
POO-Git-Casos

Descripción del Proyecto

Este repositorio contiene casos prácticos de Programación Orientada a Objetos (POO) en Java, desarrollados como parte de un trabajo académico. El proyecto incluye tres casos de estudio que demuestran conceptos fundamentales de POO y el uso de Git para control de versiones.
```

Under 'Estructura del Repositorio', the directory tree is shown:

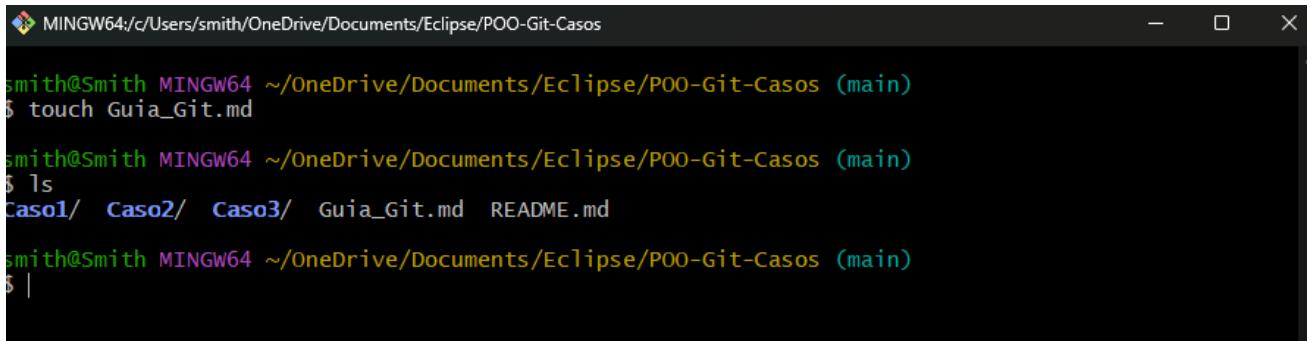
```
POO-Git-Casos/
├── caso1/
│   └── UsuarioSimple.java
├── caso2/
│   └── EstudianteInteractivo.java
└── caso3/
    └── CuentaBancaria.java
├── guia_git.md
└── README.md
└── informe_final.pdf
```

Sincronizar el repositorio local con los cambios realizados en GitHub, asegurando la actualización de los archivos en el equipo.

```
MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git pull
Enter passphrase for key '/c/Users/smith/.ssh/id_rsa':
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 6 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (6/6), 3.99 KiB | 255.00 KiB/s, done.
From github.com:Smith7w7/POO-Git-Casos
  a4e351d..d6ed527 main      -> origin/main
Updating a4e351d..d6ed527
Fast-forward
 README.md | 148 ++++++-----+
 1 file changed, 147 insertions(+), 1 deletion(-)

smith@Smith MINGW64 ~/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$
```

Crear el archivo Guia\_Git.md en el entorno local y posteriormente se documentan los comandos utilizados durante el desarrollo del proyecto.

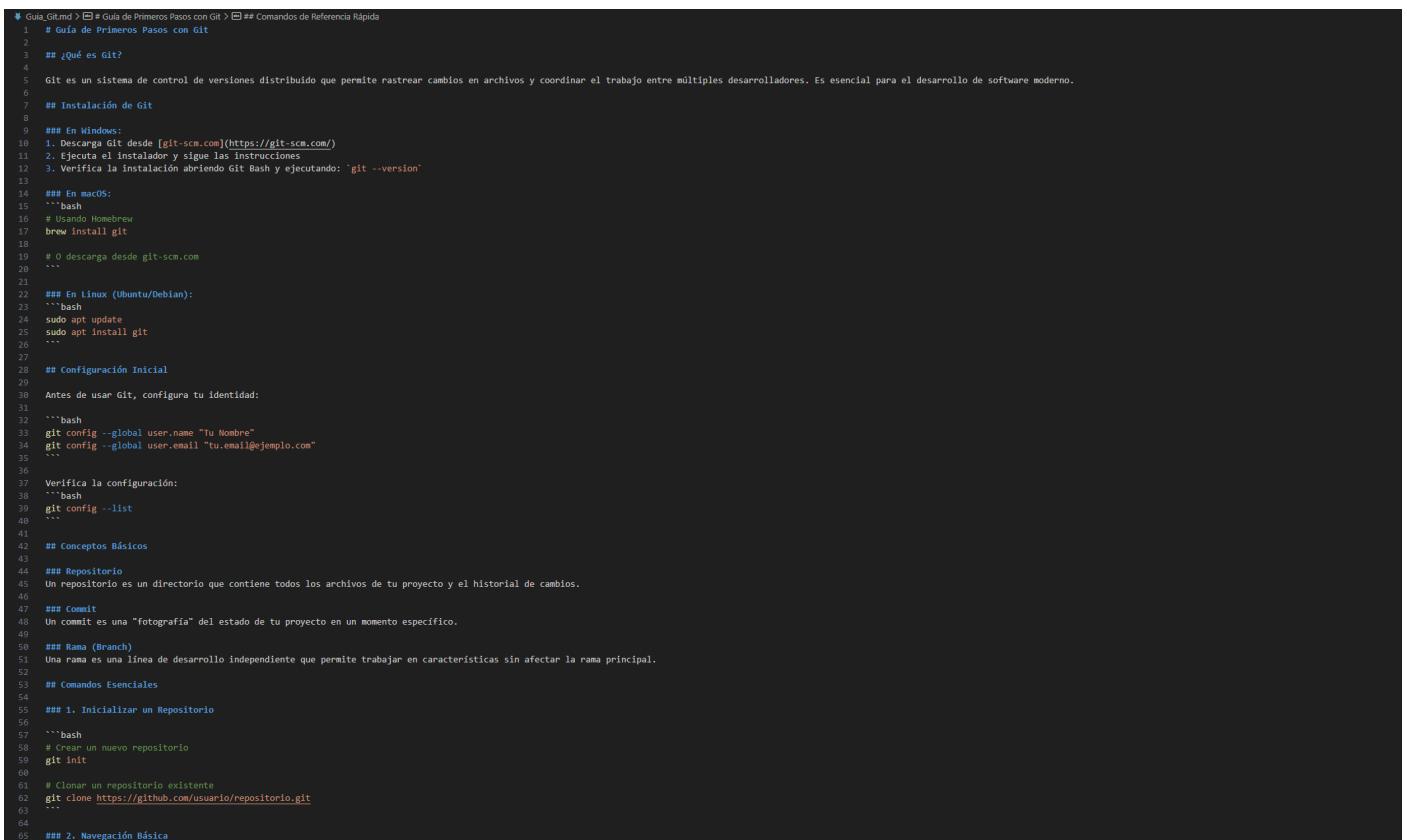


```
MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ touch Guia_Git.md

MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ ls
Caso1/ Caso2/ Caso3/ Guia_Git.md README.md

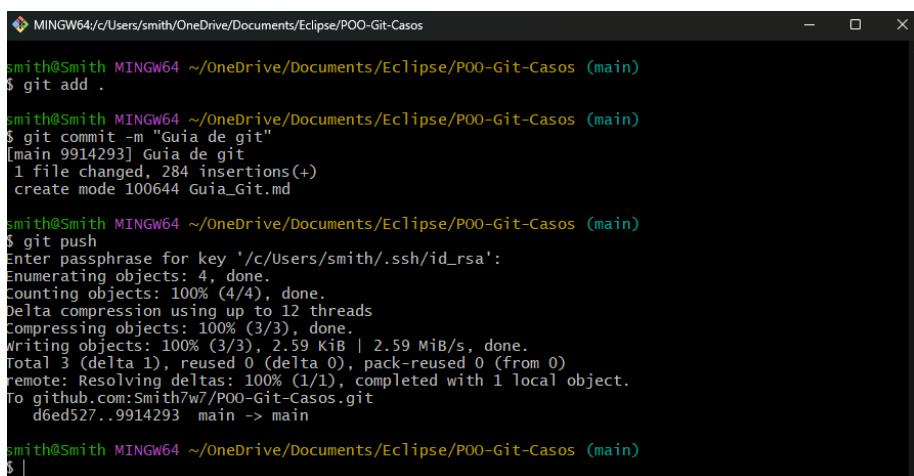
MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ |
```

Crear el archivo Guia\_Git.md en el entorno local y posteriormente se documentan los comandos utilizados durante el desarrollo del proyecto.



```
# Guia_Git.md > # Guía de Primeros Pasos con Git > # Comandos de Referencia Rápida
1 # Guía de Primeros Pasos con Git
2
3 ## ¿Qué es Git?
4
5 Git es un sistema de control de versiones distribuido que permite rastrear cambios en archivos y coordinar el trabajo entre múltiples desarrolladores. Es esencial para el desarrollo de software moderno.
6
7 ## Instalación de Git
8
9 ### En Windows:
10 1. Descarga Git desde [git-scm.com](https://git-scm.com/)
11 2. Ejecuta el instalador y sigue las instrucciones
12 3. Verifica la instalación abriendo Git Bash y ejecutando: 'git --version'
13
14 ### En macOS:
15 1. bash
16 # Usando Homebrew
17 brew install git
18
19 # O descarga desde git-scm.com
20 ...
21
22 ### En Linux (Ubuntu/Debian):
23 1. bash
24 sudo apt update
25 sudo apt install git
26 ...
27
28 ## Configuración Inicial
29
30 Antes de usar Git, configura tu identidad:
31
32 1. bash
33 git config --global user.name "Tu Nombre"
34 git config --global user.email "tu.email@ejemplo.com"
35 ...
36
37 Verifica la configuración:
38 1. bash
39 git config --list
40 ...
41
42 ## Conceptos Básicos
43
44 ### Repositorio
45 Un repositorio es un directorio que contiene todos los archivos de tu proyecto y el historial de cambios.
46
47 ### Commit
48 Un commit es una "fotografía" del estado de tu proyecto en un momento específico.
49
50 ### Rama (Branch)
51 Una rama es una línea de desarrollo independiente que permite trabajar en características sin afectar la rama principal.
52
53 ## Comandos Esenciales
54
55 ### 1. Inicializar un Repositorio
56
57 1. bash
58 # Crear un nuevo repositorio
59 git init
60
61 # Clonar un repositorio existente
62 git clone https://github.com/usuario/repositorio.git
63 ...
64
65 ### 2. Navegación Básica
```

Registrar los cambios realizados en el entorno local y sincronizarlos con el repositorio remoto en GitHub.



```
MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git add .

MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git commit -m "Guía de git"
[main 9914293] Guía de git
 1 file changed, 284 insertions(+)
 create mode 100644 Guia_Git.md

MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ git push
Enter passphrase for key '/c/users/smith/.ssh/id_rsa':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 2.59 KiB | 2.59 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:Smith7w7/POO-Git-Casos.git
   d6ed527..9914293 main -> main

MINGW64:/c/Users.smith/OneDrive/Documents/Eclipse/POO-Git-Casos (main)
$ |
```