FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

PRACTICA DE CAMPO 2 “CASOS 1, 2 Y 3”

Autores:

Palomino Ramos, Renzo Alexander N00408148

Curso:

Técnicas de programación orientada a objetos

Docente:

Torres Rodríguez, Martin Eduardo

Repositorio:

https://github.com/re286estudentupn-svg/mi-primer-project

Lima – Perú

2025-2

1. Objetivo

Aplicar control de versiones con Git y GitHub usando ramas, merges, Pull Requests y resolución de conflictos; además, desarrollar tres casos en Java (Scanner, POO con encapsulamiento y validación de operaciones).

2. Metodología (Acción → Producto)

Acción: Estructuré el repo, desarrollé cada caso en su propia rama, abrí Pull Request y fusioné a main. Simulé y resolví un conflicto.

Producto: Historial con commits por caso y merge registrado; PRs con revisión; repo compartido en GitHub.

3. Desarrollo y Evidencias

3.1. Caso 1 — Lectura con Scanner (carpeta caso1/)

El programa pide nombre, edad y ciudad, y luego los muestra.

Evidencia 1

(C:\Windows\system32>cd "C:\Users\ALEX\Desktop\mi primer project"

C:\Users\ALEX\Desktop\mi primer project>

C:\Users\ALEX\Desktop\mi primer project>cd caso1

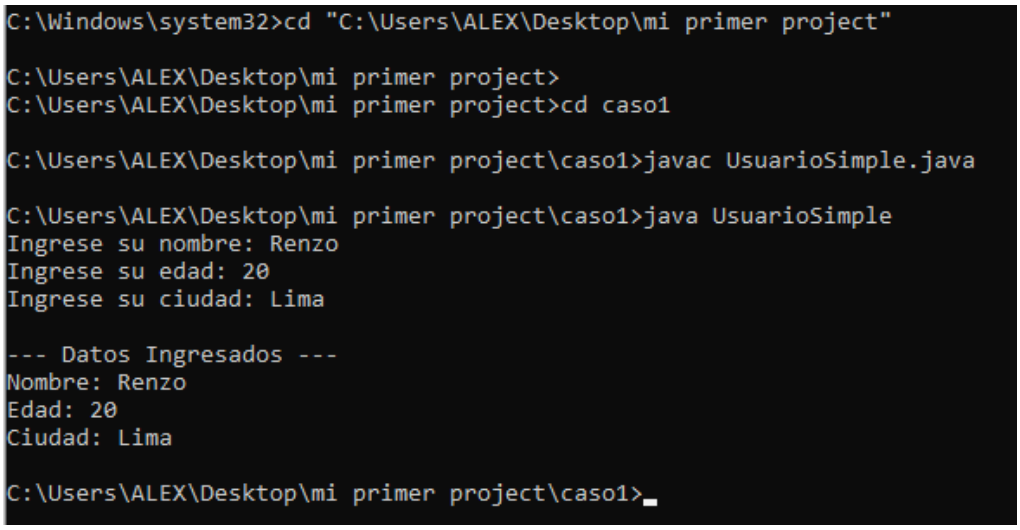
C:\Users\ALEX\Desktop\mi primer project\caso1>javac UsuarioSimple.java

C:\Users\ALEX\Desktop\mi primer project\caso1>java UsuarioSimple

Ingrese su nombre: Renzo

Ingrese su edad: 20

Ingrese su ciudad: Lima



Evidencia 2 (commit) git log --oneline --graph --decorate --all

```
C:\Users\ALEX\Desktop\mi primer project\caso1>git log --oneline --graph --decorate --all
* 1c0d29e (HEAD -> main, origin/main, origin/HEAD) feat: agregar/actualizar casos 1-3 y documentos
* b1aef10 Update guia_git.md
* 8855ede Add files via upload
*   e68583e merge: integrar feature/saludo
|\
| * 76fdedd (origin/feature/saludo, feature/saludo) feat: ampliar saludo en feature
|/
* 09ded67 feat: añadir hola.txt
* e6c8153 chore: init repo con README y .gitignore

C:\Users\ALEX\Desktop\mi primer project\caso1>
```

3.2. Caso 2 — Clase Estudiante con encapsulamiento (carpeta caso2/)

Se implementó Estudiante con privados, constructor y getters/setters; se crea desde main leyendo con Scanner.

Evidencia 3 (ejecución):

C:\Users\ALEX\Desktop\mi primer project>cd caso2

C:\Users\ALEX\Desktop\mi primer project\caso2>javac EstudianteInteractivo.java

C:\Users\ALEX\Desktop\mi primer project\caso2>java EstudianteInteractivo

CÃ³digo: N00408148

Nombre: Renzo

Ciclo (nÃºmero): 5

```
C:\Users\ALEX\Desktop\mi primer project>cd caso2

C:\Users\ALEX\Desktop\mi primer project\caso2>javac EstudianteInteractivo.java

C:\Users\ALEX\Desktop\mi primer project\caso2>java EstudianteInteractivo
CÃ³digo: N00408148
Nombre: Renzo
Ciclo (nÃºmero): 5
Creado: Estudiante{codigo='N00408148', nombre='Renzo', ciclo=5}
```

Evidencia 4

git log --merges -n 3

```
C:\Users\ALEX\Desktop\mi primer project\caso2>git log --merges -n 3
commit e68583edd17de55dc9fd21c44b360d6c7f6c616d
Merge: 09ded67 76fdedd
Author: Renzo <re286estudentupn@gmail.com>
Date:   Sun Oct 26 19:36:03 2025 -0500

    merge: integrar feature/saludo
```

3.3. Caso 3 — CuentaBancaria con validación (carpeta caso3/)

Depósitos y retiros con validación (no permitir retiro > saldo).

Evidencia 5 (ejecución):

C:\Users\ALEX\Desktop\mi primer project>cd caso3

```
C:\Users\ALEX\Desktop\mi primer project\caso3>javac CuentaBancaria.java
error: file not found: CuentaBancaria.java
Usage: javac <options> <source files>
use --help for a list of possible options

C:\Users\ALEX\Desktop\mi primer project\caso3>java CuentaBancariaApp
Titular: Renzo
Saldo inicial: 200

--- Menu ---
1. Depositar
2. Retirar
3. Consultar saldo
0. Salir
Opcion: 1
Monto: 30
Deposito exitoso. Saldo: 230.0

--- Menu ---
1. Depositar
2. Retirar
3. Consultar saldo
0. Salir
Opcion: 2
Monto: 50
Retiro exitoso. Saldo: 180.0

--- Menu ---
1. Depositar
2. Retirar
3. Consultar saldo
0. Salir
Opcion: 3
```

Saldo actual: 180.0

--- Menu ---

1. Depositar

2. Retirar

3. Consultar saldo

0. Salir

--- Menu ---

1. Depositar

2. Retirar

3. Consultar saldo

0. Salir

Opcion: 2

Monto: 500

Fondos insuficientes. Saldo: 180.0

--- Menu ---

1. Depositar

2. Retirar

3. Consultar saldo

0. Salir

Opcion: 0

Adios.

```
--- Menu ---
1. Depositar
2. Retirar
3. Consultar saldo
0. Salir
Opcion: 2
Monto: 500
Fondos insuficientes. Saldo: 180.0

--- Menu ---
1. Depositar
2. Retirar
3. Consultar saldo
0. Salir
Opcion: 0
Adios.
```

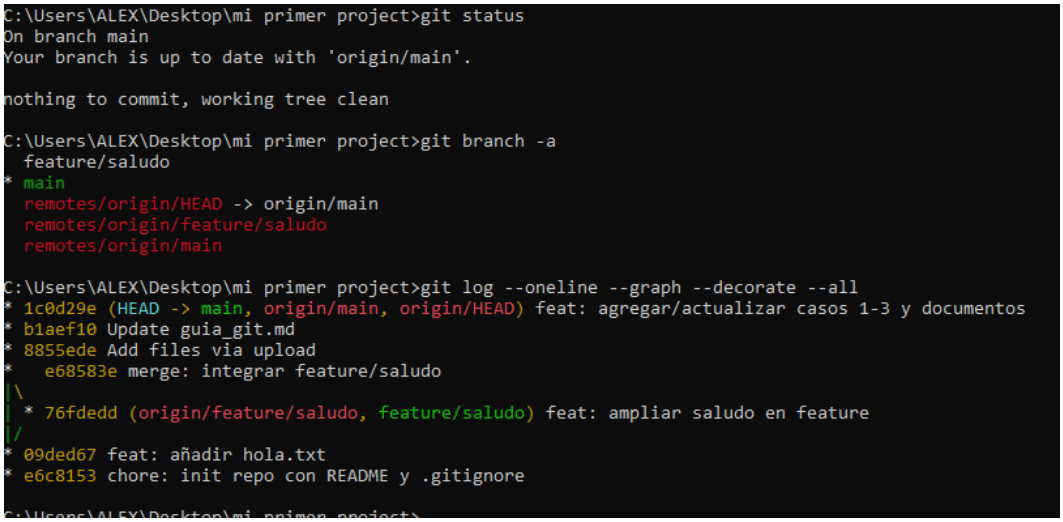
4. Control de versiones y colaboración (Git/GitHub)

Evidencia 6 (ramas e historial):

git status

git branch -a

git log --oneline --graph --decorate –all



5. Resultados

Caso 1: correcta lectura e impresión de datos.

Caso 2: encapsulamiento funcional, creación de objeto y conflicto resuelto.

Caso 3: validaciones de operaciones funcionando (mensajes claros).

Git/GitHub: flujo con branches, merges, PR, revisión y clonado exitosos.

6. Conclusiones

El uso de ramas y PR mejora la organización y revisión del código; la resolución de conflictos refuerza buenas prácticas de trabajo en equipo. GitHub facilita colaboración e integración.