INTEGRANTE:

-ANGULO RUIZ MATIAS - N00381394

PRACTICA DE CAMPO 1:

Desarrollo de Casos Prácticos - POO y Gi

Introducción

Este proyecto tiene como objetivo aplicar conceptos de Programación Orientada a Objetos (POO) en Java y gestionar el desarrollo mediante el control de versiones con Git y GitHub.

Se desarrollaron tres casos prácticos que refuerzan el uso de clases, objetos, encapsulación y validaciones en Java. Además, se documentó el uso de Git como herramienta fundamental para el trabajo colaborativo y la gestión de versiones.

Desarrollo

Caso 1 – Lectura de datos simples con Scanner

Se implementó un programa que solicita al usuario su **nombre**, **edad** y **altura** mediante la clase Scanner.
Posteriormente, se muestran los datos ingresados en consola.

Archivo: caso1/UsuarioSimple.java

Caso 2 – Clase Estudiante con atributos privados

Se creó una clase Estudiante con atributos privados (nombre, edad, carrera), un **constructor** y métodos **getter/setter** para el acceso controlado a los datos.

En el método main(), se solicita la información de un estudiante con Scanner y luego se muestra en pantalla.

Archivo: caso2/EstudianteInteractivo.java

Caso 3 – Clase CuentaBancaria con validación

Se desarrolló una clase CuentaBancaria que simula operaciones bancarias básicas: **depositar** y **retirar** dinero.

Añadí validación para que el usuario no pueda retirar más del saldo disponible. Los datos son solicitados por teclado utilizando Scanner.

Archivo: caso3/CuentaBancaria.java

Uso de Git

Durante el proyecto se aplicaron los siguientes comandos de Git:

- **git init: inicializar un repositorio local.**
- **git clone: clonar el repositorio desde GitHub.**
- **git add: preparar archivos para el commit.**
- **git commit -m "mensaje": guardar cambios con un mensaje explicativo.**
- **git status: mostrar el estado del repositorio.**
- **git log: ver el historial de commits.**
- **git branch: crear nuevas ramas.**
- **git checkout: cambiar entre ramas.**
- **git merge: fusionar ramas.**
- **git push: enviar cambios al repositorio remoto en GitHub.**
- **git pull: actualizar el repositorio local con los cambios remotos.**

Toda la documentación del uso de Git se encuentra también en el archivo guia_git.md dentro del repositorio.