

Parte 1: Configuración y trabajo con Git y GitHub – Actividades realizadas

1. Creación de repositorio

- Se creó un repositorio local usando `git init`.
- Se creó un repositorio remoto en GitHub y se vinculó con el local usando:

```
git remote add origin https://github.com/wilito48/PROYECTO_POO_CAMPO.git
```

2. Clonado del repositorio

- Cada miembro del equipo clonó el repositorio remoto en su máquina local usando:

```
Git clone https://github.com/wilito48/PROYECTO_POO_CAMPO.git
```

3. Gestión de ramas

- Se crearon nuevas ramas de desarrollo con:

```
git branch nombre_rama
```

- Se cambió entre ramas con:

```
git checkout nombre_rama
```

4. Cambios en el código y confirmaciones

- Se realizaron modificaciones al código.
- Se añadieron archivos al área de staging y se confirmaron con:

```
Git add .
```

```
git commit -m "Mensaje descriptivo del cambio"
```

5. Visualización del historial de cambios

- Se revisó el historial de commits con:

```
git log
```

6. Fusión de ramas

- Se fusionaron ramas a la rama principal con:

```
git merge nombre_rama
```

7. Resolución de conflictos de fusión

- Se detectaron y resolvieron conflictos manualmente en los archivos involucrados.
- Se completó la fusión luego de editar y hacer commit.

8. Colaboración con Pull Requests (PRs)

- Se crearon ramas de funcionalidad y se subieron con:

```
git push origin nombre_rama
```
- Se abrió un Pull Request desde GitHub para revisión y discusión.
- Se documentó la revisión de código mediante comentarios y aprobaciones en el PR.

9. Uso de comandos avanzados

- ```
git stash
```

: se guardaron cambios temporales sin hacer commit.
- ```
git revert
```

: se revirtió un commit específico para deshacer errores sin borrar historial.
- ```
git cherry-pick
```

: se aplicó un commit específico de otra rama.