

**FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES**

GUÍA DOCUMENTADA - INFORME DE PROYECTO

Autores:

- José luiz Garcia Moreno
- Sebastian Enrique Rodríguez Leiva
- José Carlos Jauregui Zavaleta
- Milagros Yeraldin Polo Niquin
- Frank Junior Alberto Flores Castillo
- Fernando Giuseppen Muñoz Asuncion

Curso: Técnicas de Programación Orientada a Objetos
Docente del Curso: Ing. Jorge Ricardo Perez Vigil

Trujillo – Perú
2025

TRABAJO DE CAMPO - S04: BUENAS PRÁCTICAS Y COMANDOS AVANZADOS

◆ Instrucciones Resueltas

1. Buenas prácticas en Git

- Nombres de ramas descriptivos: `feature/login`, `bugfix/email`
- Mensajes claros: `git commit -m "Corrige validación de DNI"`

2. Estrategias de ramificación - Git Flow

- Ramas: `main`, `develop`, `feature/`, `release/`, `hotfix/`
- Se usa `develop` como rama base para desarrollo y se integran cambios vía PR.

3. Comandos avanzados

- Guardar cambios sin hacer commit:
`git stash`
- Revertir un commit sin eliminarlo:
`git revert <hash>`
- Aplicar commit específico en otra rama:
`git cherry-pick <hash>`

✂ Aplicación en Proyecto

- Se adoptó Git Flow con ramas claras para cada etapa del desarrollo.
- Se utilizó `git stash` antes de un cambio urgente de rama.
- Se usó `git revert` tras detectar un bug en producción.
- Se aplicó `git cherry-pick` para portar una mejora puntual desde `develop` a `hotfix`.