Texto

El contenido generado por IA puede ser incorrecto.GUIA DOCUMENTADA  
INFORME PROYECTO

**Curso:** **TÉCNICAS DE PROGRAM.ORIE. OBJ**

**Grupo: Grupo 2**

Autores: Azaña Tejada Marcial

Castillo Tineo Santiago Junior

Ramos Zevallos Benjamin Franklin

Tarazona Mallqui Anibal

Docente: Esther Tarmeño Juscamaita

Jorge Chicana Aspajo

LIMA-PERU

2025-1

**Mejores prácticas y comandos avanzados en Git**

Git es una herramienta potente de control de versiones que permite **gestionar proyectos de forma eficiente**, especialmente cuando se trabaja en equipo. Para aprovechar todo su potencial, es recomendable seguir ciertas **mejores prácticas** y conocer comandos avanzados.

**Convenciones de nombres y mensajes de commit**

**Ramas**: usar nombres claros y consistentes facilita la comprensión del proyecto.

Ejemplo: main, desarrollo, feature/nueva-funcionalidad, bugfix/correccion-errores.

**mensajes de commit**: deben ser concisos, claros y descriptivos.

Buena práctica:

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

**Estrategias de ramificación**

* **Git Flow**: es un modelo de ramificación popular que organiza el desarrollo en varias ramas:
  + main → versión estable del proyecto.
  + develop → rama de integración donde se prueban nuevas funcionalidades.
  + feature/\* → ramas para desarrollar nuevas funcionalidades.
  + release/\* → ramas para preparar nuevas versiones.
  + hotfix/\* → ramas para correcciones urgentes en producción.
* Este modelo permite mantener un **flujo de trabajo organizado**, donde cada tipo de rama tiene un propósito claro y controlado.

**Comandos avanzados:**

**git stash**

Permite **guardar temporalmente cambios no confirmados** y limpiar el área de trabajo para cambiar de rama sin perder lo que estás haciendo.