

Asignatura

Arquitectura del Software



Profesor

Yago Fontenla Seco

{yago.fontenla1@uie.edu}

Control de versiones (VCS)

El **control de versiones (VCS)** es un sistema que **registra todos los cambios realizados sobre los archivos** de un proyecto a lo largo del tiempo.

Permite:

- Guardar **versiones del código** (snapshots).
- **Comparar y revertir** cambios fácilmente.
- **Colaborar** con otros sin sobrescribir el trabajo.
- Mantener **historial y trazabilidad** del proyecto.



Ejemplo:

Cada commit funciona como una **foto del estado del código** en un momento específico, con autor, fecha y descripción.

Ventajas

Trazabilidad

- Cada cambio queda registrado (quién, cuándo y por qué).
- Facilita auditorías y mantenimiento.

Trabajo colaborativo

- Varios desarrolladores pueden trabajar en paralelo sin pisarse.
- Se integran cambios mediante *merge* o *pull requests*.

Recuperación ante errores

- Permite volver a una versión anterior estable si algo falla.
- Minimiza el riesgo de pérdida de trabajo.

Integración con CI/CD

- Cada commit puede activar pipelines automáticos de build y test.

Git



Creado por **Linus Torvalds** (2005) para el desarrollo del kernel de Linux. Es un **sistema distribuido, rápido y seguro**. Permite gestionar proyectos con **ramas (branches)** y **fusionar cambios (merge)** de forma eficiente.

Conceptos básicos:

- **Repositorio (repo)**: colección del código y su historial.
- **Commit**: registro individual de cambios.
- **Branch**: línea paralela de desarrollo.
- **Merge**: integración de ramas.
- **HEAD**: referencia al commit actual.

Trabajando con Git

Comandos esenciales:

`git init` → inicializa un nuevo repositorio local.

`git add .` → prepara archivos para el commit.

`git commit -m "mensaje"` → guarda un snapshot del código.

`git branch nueva_rama` → crea una rama.

`git checkout nueva_rama` → cambia de rama.

`git merge rama` → fusiona ramas.

Más conceptos:

Repositorio remoto: copia compartida del proyecto en la nube (GitHub, GitLab, Bitbucket).

Push: subir cambios locales al servidor remoto.

Pull: traer cambios del remoto al local.

Clone: descargar un repositorio remoto completo.

Pull Request (PR): solicitud de revisión y fusión de cambios.

Trabajando con Git

Flujo típico local:

1. Crear o clonar un proyecto.
2. Trabajar en una rama propia.
3. Realizar commits descriptivos.
4. Fusionar al branch principal (main/master) tras revisión.

Flujo típico remoto:

1. Clonar el repositorio principal.
2. Crear una rama para la funcionalidad.
3. Hacer commits y “push” al remoto.
4. Crear un **PR** para revisión.
5. Merge tras aprobación.

El PR es clave en DevOps: permite **revisiones de código colaborativas** y activa pipelines automáticos (CI/CD).

Asignatura

Arquitectura del Software



Profesor

Yago Fontenla Seco

{yago.fontenla1@uie.edu}