

## Sesión 7: Git Flow (CLI/GUI)

Organiza el desarrollo con ramas con roles claros: main, develop, feature, release, hotfix.

### Qué vas a lograr

- Entender el rol de cada rama.
- Usar Git Flow por CLI o emularlo con convenciones.
- Integrar con PR/MR en GitHub/GitLab/Bitbucket.

### Requisitos

- Conocimiento de ramas/merges (Sesión 6).

### Instalar soporte Git Flow (opcional)

- Linux

```
sudo apt install -y git-flow
```

- macOS

```
brew install git-flow-avh
```

- Windows

```
# Usa scoop/chocolatey para instalar git-flow-avh o sigue el flujo manual sin la he
```

### Inicializar flujo

```
git flow init # acepta nombres por defecto: main/develop
```

### Operaciones típicas

- Feature

```
git flow feature start login  
# desarrolla y commitea  
git flow feature finish login
```

- Release

```
git flow release start 1.2.0  
# actualiza CHANGELOG/version  
git flow release finish 1.2.0
```

- Hotfix

```
git flow hotfix start 1.2.1  
# arregla bug crítico  
git flow hotfix finish 1.2.1
```

## Integración con plataformas

- Usa PR/MR para integrar features y releases.
- Protege main/develop con revisiones y checks.
- Estrategias: squash para features; merge commit para releases.

## Alternativas

- Trunk-Based Development: ramas cortas, feature flags, releases continuas.