



# Manual Básico de Git

## 1. Configuración inicial

- `git config --global user.name "email.contractor"`
- `git config --global user.email "email.contractor@bbva.com"`
- → Se hace una sola vez. Define tu identidad en los commits.

## 2. Crear o clonar repositorios

- `git init` # Inicializar un repo nuevo (No se usa en el banco)
- `git clone https://bitbucket.globaldevtools.bbva.com/bitbucket/scm/gl_glcj/proyecto.git`  
Clonar un repositorio.

## 3. Ciclo básico (trabajo diario)

- `git status` # Ver estado de los archivos
- `git add archivo.js` # Agregar un archivo específico
- `git add .` # Agregar todos los cambios
- `git commit -m "feat(js): Mensaje claro del cambio en ingles [ticket-jira]"`  
# Confirmar cambios, hay diferentes tipos feat, fix, chore.
- `git push origin main` # Enviar cambios al remoto
- `git pull origin main` # Traer cambios del remoto

## 4. Ramas (branches)

- `git branch` # Ver ramas existentes
- `git branch feature/nueva-funcionalidad` # Crear nueva rama
- `git checkout feature/nueva-funcionalidad` # Cambiar de rama
- `git checkout -b fix/bug-123` # Crear y cambiar de rama
- `git push -u origin feature/nueva-funcionalidad` # Subir rama nueva

Nota: Las ramas en el banco se crean desde el jira.

## 5. Revisar historial

- `git log --oneline` # Ver historial simple
- `git diff` # Ver diferencias antes de hacer commit

## 6. Merge y rebase

- `git merge main` # Traer cambios de otra rama a la actual
- `git rebase main` # Reorganizar commits sobre otra rama

## 8. Reset y revert (deshacer seguro)

- `git checkout -- archivo.js` # Deshacer cambios sin commitear
- `git reset archivo.js` # Quitar archivo del stage
- `git revert <id_commit>` # Revertir un commit específico
- `git reset --soft HEAD~1` # Quitar último commit pero mantener cambios

## 9. Buenas prácticas

- Siempre trabajar en ramas (no en main)  
En el banco se usa **master** para componentes y **develop** para Soluciones Funcionales.
- Mensajes de commit claros (ej: "fix(js): error in the warning modal").
- Hacer **git pull** antes de empezar a trabajar.
- Revisar cambios con `git status` y `git diff` antes de hacer un commit.
- Nunca subir archivos grandes o credenciales → usar `.gitignore`.