Git Básico

Arturo Silvelo

Try New Roads

Ejercicios Prácticos

Git Hooks y CI/CD

Objetivos

Al finalizar estos ejercicios serás capaz de:

- Entender el problema de los hooks nativos de Git
- Configurar hooks nativos en `.git/hooks/`
- Implementar hooks con Husky
- Comparar ambas soluciones
- Configurar GitHub Actions para CI/CD

Ejercicio 1: Hooks Nativos

Explorando el problema

Archivos necesarios: `hooks.zip`

1. **Extraer y explorar:**

```
unzip hooks.zip
cd hooks-ejemplo
ls -la .git/hooks/
```

2. **Probar los hooks:**

```
# Restaurar permisos (se pierden al comprimir)
chmod +x .git/hooks/*
```

Ejercicio 1: Resultados esperados

Pre-commit hook:

- Debe fallar si hay archivos muy grandes

A. Probar Pre-commit (archivos grandes):

```
# Crear archivo muy grande para probar
dd if=/dev/zero of=archivo-grande.txt bs=2M count=1
git add archivo-grande.txt
git commit -m "add large file"
# ebe fallar por tamaño
```

Commit-msg hook:

- Debe aceptar un formato especifico

```
# Crear cambio pequeño
echo "cambio menor" >> README.md
git add README.md

# Intentar con formato incorrecto
git commit -m "mensaje mal formateado"
# Debe fallar por formato
```

Pre-push hook:

- Debe rechazar push directo a main/master
- Obtener la rama actual: `git rev-parse --abbrev-ref HEAD`

```
# Asegurar que estamos en main
git switch main

# Crear cambio y commit
echo "cambio en main" >> README.md
git add README.md
git commit -m "feat: change in main"

# Intentar push directo a main
git push origin main
```

Ejercicio 2: Hooks con Husky

Archivos necesarios: `hooks-husky.zip`

1. **Extraer y configurar:**

```
unzip hooks-husky.zip
cd hooks-husky-ejemplo
npm install  # Instala Husky automáticamente
```

2. **Explorar configuración:**

```
# Ver configuración en package.json
cat package.json

# Ver hooks instalados
ls -la .husky/
```