

# Configuración de Hooks de Git

Usuario

September 4, 2024

## 1 Introducción

Este informe documenta los pasos realizados para configurar los hooks de Git en el repositorio de desarrollo2. Se han creado y configurado tres hooks: **pre-commit**, **pre-push**, y **post-merge**.

## 2 Inicialización del Repositorio Git

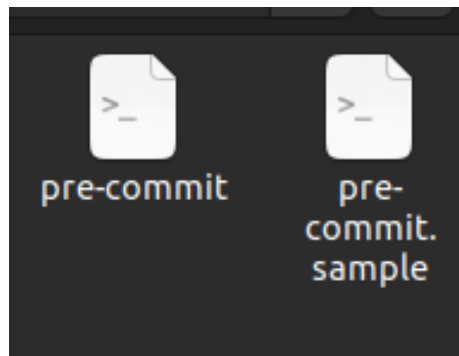
Para comenzar, se inicializó un repositorio Git en la carpeta **desarrollo2**:

```
cd desarrollo2
git init
```

## 3 Configuración del Hook pre-commit

El hook **pre-commit** se activa antes de realizar un commit. Se creó el archivo y se configuró para que sea ejecutable:

```
cd .git/hooks
touch pre-commit
```



Después, se editó el archivo usando `nano`:

```
nano pre-commit
```



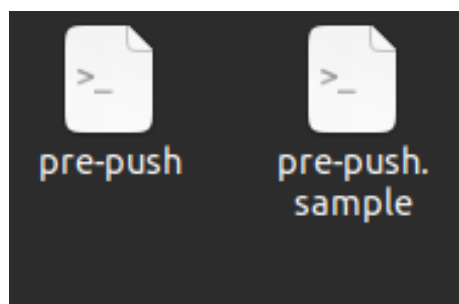
The screenshot shows a terminal window with the title 'usuario@FR15: ~/desarrollo2/git/hooks'. The nano editor is open, showing the content of the 'pre-commit' file. The text in the editor is as follows:

```
GNU nano 6.2 pre-commit
#!/bin/bash
files=$(git diff --cached --name-only --diff-filter=ACM | grep '\.py$')
if [ "$files" != "" ]; then
  unformatted=$(black --check $files)
  if [ $? -ne 0 ]; then
    echo "Hay archivos sin formatear. Formatea el código antes de hacer commit."
    exit 1
  fi
fi
```

## 4 Configuración del Hook pre-push

El hook `pre-push` se activa antes de empujar (push) al repositorio remoto. Se creó el archivo y se configuró para que sea ejecutable:

```
touch pre-push
```



Después, se editó el archivo usando `nano`:

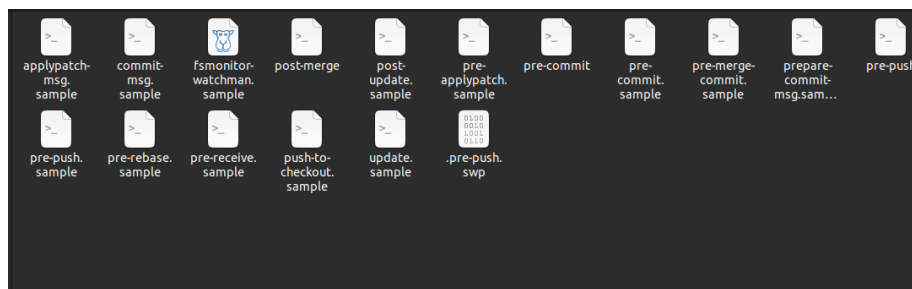
```
nano pre-push
```

```
GNU nano 6.2 pre-push
#!/bin/bash
echo "Ejecutando pruebas antes del push..."
if ! make test; then
    echo "Las pruebas fallaron. Push abortado."
    exit 1
fi
```

## 5 Configuración del Hook post-merge

El hook `post-merge` se activa después de realizar un merge. Se creó el archivo y se configuró para que sea ejecutable:

```
touch post-merge
```



Después, se editó el archivo usando `nano`:

```
nano post-merge
```

```
GNU nano 6.2 post-merge
#!/bin/bash
if git diff-tree -r --name-only --no-commit-id ORIG_HEAD HEAD | grep -q 'package.json'; then
    echo "package.json ha cambiado. Instalando dependencias..."
    npm install
fi
```

## 6 Conclusión

La configuración de los hooks de Git permite automatizar tareas esenciales en el flujo de trabajo, garantizando la consistencia y calidad del código antes de cada commit, push, o merge.