

Guía Práctica 01: Uso Básico de Gitflow en Proyectos con Git y GitHub Classroom (Desde Cero)

Objetivo General

Aplicar el flujo de trabajo Gitflow desde la clonación de un repositorio asignado mediante GitHub Classroom, hasta la gestión de nuevas funcionalidades, versiones y correcciones de errores usando Git y GitHub.

Requisitos Previos

- Tener Git instalado en el equipo
- Tener una cuenta activa en GitHub
- Haber recibido el enlace de asignación a través de GitHub Classroom

Parte 0: Obtener el Repositorio Asignado

Paso 1: Aceptar la tarea en GitHub Classroom

1. Hacer clic en el enlace que te compartió el docente (GitHub Classroom)
2. Esperar a que se genere tu repositorio personal (automático)
3. Copiar la URL de tu repositorio generado (ej. <https://github.com/org-clase/gitflow-tarea-nombreusuario.git>)

Paso 2: Clonar repositorio asignado y configurar localmente

```
# Clonar el repositorio entregado por el docente
git clone https://github.com/org-clase/gitflow-tarea-nombreusuario.git
cd gitflow-tarea-nombreusuario
```

Parte 1: Configuración Inicial de Gitflow

```
# Crear rama develop desde main
git checkout -b develop
# Subirla a GitHub
git push -u origin develop
```

Parte 2: Crear una Nueva Funcionalidad

```
# Crear rama de funcionalidad desde develop
git checkout -b feature/formulario-contacto develop

# Simular una funcionalidad
echo "Formulario de contacto HTML" > contacto.html

# Registrar cambios
git add contacto.html
git commit -m "Agregar formulario de contacto"

# Volver a develop y fusionar
git checkout develop
git merge feature/formulario-contacto

# Subir cambios a GitHub
git push
```

Parte 3: Preparar una Nueva Versión (Release)

```
# Crear rama de release desde develop
git checkout -b release/v1.0 develop

# Simular preparación de versión
echo "Versión 1.0" > version.txt
git add version.txt
git commit -m "Preparar versión 1.0"

# Fusionar en main
git checkout main
git merge release/v1.0

# Etiquetar versión
git tag -a v1.0 -m "Versión 1.0"

# Subir a GitHub
git push && git push --tags

# Fusionar en develop
```

```
git checkout develop
git merge release/v1.0
```

Parte 4: Arreglar un Error Crítico (Hotfix)

```
# Crear rama hotfix desde main
git checkout -b hotfix/arreglo-titulo main

# Simular arreglo
echo "Arreglo de título en página principal" > fix.txt
git add fix.txt
git commit -m "Corregir título principal"

# Fusionar en main
git checkout main
git merge hotfix/arreglo-titulo

git tag -a v1.0.1 -m "Hotfix 1.0.1"
git push && git push --tags

# Fusionar en develop
git checkout develop
git merge hotfix/arreglo-titulo
```

Resultado Esperado

- main contiene la versión estable del proyecto
- develop está actualizado con todos los cambios
- Las ramas temporales (feature/, release/, hotfix/) pueden eliminarse si se desea