

Guía Práctica: GitHub Flow con GitHub Classroom

Objetivo General

Aplicar la estrategia GitHub Flow para desarrollar proyectos de forma colaborativa usando Git y GitHub Classroom desde cero.

♦ Requisitos Previos

- Tener Git instalado en el equipo
- Tener una cuenta de GitHub activa
- Estar inscrito en una asignación de GitHub Classroom proporcionada por el docente

Parte 0: Aceptar la asignación de GitHub Classroom

1. El docente comparte un enlace como:
<https://classroom.github.com/a/abc123>
2. Haz clic en el enlace y acepta la tarea.
3. GitHub creará automáticamente un repositorio con tu nombre: nombre-alumno-nombre-tarea
4. Copia la URL de tu repositorio.

Parte 1: Clonar el repositorio en tu equipo

```
# Cambia a la carpeta donde deseas guardar el proyecto
cd Documentos/proyectos
```

```
# Clona tu repositorio
git clone https://github.com/usuario/nombre-alumno-tarea.git
cd nombre-alumno-tarea
```

Parte 2: GitHub Flow - Flujo de trabajo paso a paso

GitHub Flow se basa en trabajar con ramas y hacer Pull Requests contra la rama principal (main).

1. Crear una rama para una nueva funcionalidad o cambio

```
git checkout -b agregar-formulario
```

2. Realizar cambios en los archivos del proyecto

```
echo "Formulario de contacto" > contacto.html  
git add contacto.html  
git commit -m "Agregar formulario de contacto"
```

3. Subir la rama al repositorio remoto

```
git push origin agregar-formulario
```

4. Crear un Pull Request en GitHub

1. Ir al repositorio en GitHub
2. Verás un aviso para crear un Pull Request desde tu rama
3. Haz clic en "Compare & pull request"
4. Escribe un título y descripción clara
5. Haz clic en "Create pull request"

5. Revisar y fusionar (mergear) la rama

1. Haz clic en "Merge pull request"
1. Luego en "Confirm merge"
2. Finalmente, elimina la rama si ya no se usará

Parte 3: Repetir el flujo para cada cambio importante

1. Crea una nueva rama para cada cambio
2. Realiza los cambios y haz commit
3. Sube la rama y haz Pull Request
4. Revisa y mergea

Comandos útiles

```
# Ver en qué rama estoy  
git branch
```

```
# Ver historial de cambios  
git log --oneline
```

```
# Cambiar de rama  
git checkout nombre-rama
```

```
# Eliminar una rama local  
git branch -d nombre-rama
```

Resultado Esperado

- El repositorio en GitHub refleja los cambios en main a través de Pull Requests
- Cada funcionalidad nueva fue trabajada en su propia rama
- La historia del proyecto está ordenada, clara y rastreable

Buenas Prácticas

- Hacer commits pequeños y descriptivos
- Nombrar las ramas con claridad (feature/, fix/, test/, etc.)
- Crear un Pull Request por cada tarea
- Revisar el código antes de hacer merge