

Práctica para Alumnos: Uso de GitHub con Issues, Pull Requests, Colaboradores, y Comandos Básicos de Git

Esta práctica guiará a los alumnos a través de los conceptos y comandos básicos de Git y GitHub, incluyendo la gestión de issues, la creación de pull requests, la adición de colaboradores, y los comandos esenciales de Git. Al final de la práctica, los alumnos habrán trabajado con clonación de repositorios, control de versiones, ramas, manejo de archivos README.md, gestión de issues, colaboración en equipo, y realizarán commits desde dos colaboradores diferentes.

Requisitos:

- Cuenta en GitHub
- Git instalado en su máquina
- Editor de texto o IDE
- Dos cuentas de GitHub para simular colaboración (puede ser la cuenta personal y otra secundaria o de un compañero)

Paso a Paso

1. Crear un Nuevo Repositorio en GitHub

1. Ve a GitHub y crea un nuevo repositorio.
2. Asigna un nombre al repositorio (ejemplo: practica-github-html-css).
3. Marca la opción para inicializar el repositorio con un README.md.

2. Clonar el Repositorio

```
git clone https://github.com/tu-usuario/practica-github-html-css.git
cd practica-github-html-css
```

3. Crear una Nueva Rama

1. Crear una nueva rama llamada feature-inicial.

```
git branch feature-inicial
```

2. Cambiar a la nueva rama.

```
git checkout feature-inicial
```

4. Crear Archivos HTML y CSS

1. En tu editor de texto, crea un archivo index.html y agrega el siguiente contenido:

```
html

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Práctica GitHub</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Bienvenidos a la Práctica de Git y GitHub</h1>
  <p>Esta es una página simple para aprender los conceptos básicos de Git y GitHub.</p>
</body>
</html>
```

2. Crea un archivo styles.css y agrega el siguiente contenido:

```
css

body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  color: #333;
  text-align: center;
  padding: 50px;
}

h1 {
  color: #0056b3;
}
```

5. Hacer Cambios y Commit

1. Agrega los archivos creados al área de preparación (staging).

```
git add index.html styles.css
```

2. Haz un commit de los cambios.

```
git commit -m "Agregar archivos index.html y styles.css"
```

3. Empuja la nueva rama al repositorio remoto.

```
git push origin feature-inicial
```

6. Agregar Colaboradores

1. En GitHub, ve a la pestaña "Settings" del repositorio.
2. En el menú lateral, selecciona "Manage access".

3. Haz clic en "Invite a collaborator" y agrega el nombre de usuario de tu colaborador. En este caso, añade la cuenta secundaria o la de tu compañero.

7. Crear y Resolver Issues

1. En GitHub, ve a la pestaña "Issues" del repositorio y crea un nuevo issue llamado "Agregar sección de instalación en README".
2. Desde la línea de comandos, crea una nueva rama para resolver el issue.

```
git checkout -b feature-instalacion
```

3. Edita el archivo README.md para agregar una sección de instalación.

markdown

Instalación

Para instalar este proyecto, clona el repositorio y navega al directorio:

4. Agrega y haz commit de los cambios.

```
git add README.md
```

```
git commit -m "Agregar sección de instalación en README.md"
```

5. Empuja la nueva rama al repositorio remoto.

```
git push origin feature-instalacion
```

8. Crear un Pull Request

1. Ve a GitHub y crea un nuevo Pull Request desde la rama feature-instalacion hacia main.
2. Describe los cambios realizados y solicita la revisión del colaborador.

9. Revisar y Fusionar el Pull Request

1. El colaborador revisará el Pull Request y proporcionará comentarios si es necesario.
2. Una vez aprobado, fusiona el Pull Request en la rama main.

10. Realizar Commits desde dos Colaboradores

Colaborador 1

1. El colaborador 1 (tú) crea una nueva rama llamada feature-titulo.

```
git checkout -b feature-titulo
```

2. Edita el archivo index.html y cambia el título de la página.

```
html
```

```
<title>Práctica Git y GitHub - Alumno 1</title>
```

3. Agrega y haz commit de los cambios.

```
git add index.html
```

```
git commit -m "Cambiar título de la página - Alumno 1"
```

4. Empuja la nueva rama al repositorio remoto.

```
git push origin feature-titulo
```

Colaborador 2

1. El colaborador 2 clona el repositorio.

```
git clone https://github.com/tu-usuario/practica-github-html-css.git
```

```
cd practica-github-html-css
```

2. El colaborador 2 crea una nueva rama llamada feature-estilos.

```
git checkout -b feature-estilos
```

3. Edita el archivo styles.css y agrega un nuevo estilo.

```
css
```

```
p {  
    font-size: 18px;  
}
```

4. Agrega y haz commit de los cambios.

```
git add styles.css
```

```
git commit -m "Agregar estilo a los párrafos - Alumno 2"
```

5. Empuja la nueva rama al repositorio remoto.

```
git push origin feature-estilos
```

11. Revisar y Fusionar Pull Requests de Colaboradores

1. Ambos colaboradores crean Pull Requests desde sus respectivas ramas (feature-titulo y feature-estilos) hacia main.
2. Revisan y fusionan los Pull Requests en la rama main.

Resumen de Comandos Usados

- **Clonar repositorio:** `git clone`
- **Crear nueva rama:** `git branch`
- **Cambiar de rama:** `git checkout`
- **Agregar cambios al área de preparación:** `git add`
- **Hacer commit de los cambios:** `git commit`
- **Fusionar ramas:** `git merge`
- **Enviar cambios al repositorio remoto:** `git push`
- **Traer cambios del repositorio remoto:** `git pull`

Conclusión

Al finalizar esta práctica, los alumnos habrán adquirido experiencia práctica con los comandos básicos de Git y GitHub, trabajado con archivos HTML y CSS simples, y colaborado en equipo realizando commits y pull requests desde diferentes colaboradores. Esto les preparará para colaborar eficientemente en proyectos de desarrollo de software en equipo y aplicar prácticas de control de versiones en sus proyectos.