

HERRAMIENTAS DEL DESARROLLO

Laboratorio: Solicitudes Pull (Pull Requests), Flujos de Trabajo: Merge Conflict y Release

Objetivo:

El objetivo de este laboratorio es aprender cómo trabajar con **Pull Requests (PR)** en un flujo de trabajo colaborativo en Git, cómo resolver conflictos de fusión (merge conflicts), y cómo gestionar un **release** en un repositorio de Git.

Procedimientos Básicos:

1. Crear un Repositorio Remoto:

1. Inicia sesión en GitHub (o GitLab).
2. Crea un nuevo repositorio y configura los detalles, asegurándote de que sea público o privado según tu preferencia.
3. Clona el repositorio a tu máquina local:

```
git clone https://github.com/<usuario>/<nombre_repositorio>.git
```

2. Crear una Rama de Característica (Feature Branch):

Las ramas son útiles para trabajar en nuevas características sin afectar la rama principal.

1. Crea una nueva rama:

```
git checkout -b feature/nueva-caracteristica
```

3. Realizar Cambios y Commit:

Realiza cambios en el código y luego haz un commit.

1. Realiza modificaciones en el archivo de tu elección.
2. Añade los archivos modificados:

```
git add <archivo_modificado>
```

3. Haz un commit con un mensaje claro:

```
git commit -m "Agregado nueva funcionalidad para la característica"
```

4. Subir los Cambios a GitHub (Push):

Envía los cambios al repositorio remoto:

```
git push origin feature/nueva-caracteristica
```

5. Crear una Pull Request (PR):

1. Dirígete al repositorio en GitHub.
2. En la página principal, haz clic en "Compare & pull request" para abrir una nueva PR.
3. Rellena el título y la descripción de la PR, explicando qué cambios realizaste.

6. Revisar y Aprobar la Pull Request:

El propietario del repositorio o un colaborador puede revisar la PR y aprobarla. Si los cambios son correctos, se puede proceder a la fusión (merge).

7. Fusionar la Pull Request:

Una vez que la PR ha sido aprobada, se fusiona con la rama principal (main):

1. Haz clic en "Merge pull request".
2. Confirma la fusión con "Confirm merge".

8. Resolver Conflictos de Fusión (Merge Conflicts):

Si hay cambios conflictivos entre las ramas, Git no puede fusionarlas automáticamente. En este caso, se deberá resolver el conflicto de manera manual.

1. Git marcará los archivos con conflictos. Abre esos archivos para ver las secciones en conflicto.
2. Resuelve el conflicto, eligiendo qué cambios deseas mantener o combinando ambos.
3. Una vez resuelto, marca los archivos como resueltos:

```
git add <archivo_resuelto>
git commit -m "Conflictos resueltos en <archivo>"
```

9. Crear una Etiqueta (Tag) para el Release:

Las etiquetas permiten marcar puntos específicos en la historia del repositorio como versiones de lanzamiento.

1. Crea una etiqueta en el commit más reciente:

```
git tag -a v1.0 -m "Primera versión estable"
```

2. Sube la etiqueta al repositorio remoto:

```
git push origin v1.0
```

10. Actualizar tu Repositorio Local:

Es importante asegurarse de que tu repositorio local esté actualizado con los últimos cambios del repositorio remoto.

```
git pull origin main
```

Preguntas:

1. **¿Qué es una Pull Request (PR) y por qué es importante en el flujo de trabajo colaborativo?**
 - **Respuesta:** Una PR es una solicitud para fusionar una rama con otra, típicamente para integrar cambios de una rama de características a la rama principal. Es importante porque permite revisar, discutir y probar los cambios antes de que se fusionen.
2. **¿Qué son los conflictos de fusión en Git?**
 - **Respuesta:** Los conflictos de fusión ocurren cuando dos ramas modifican las mismas líneas de un archivo, lo que impide que Git fusione automáticamente las ramas. Requiere intervención manual para resolver los conflictos.
3. **¿Qué es un flujo de trabajo de Git?**
 - **Respuesta:** Un flujo de trabajo de Git es un conjunto de reglas y procedimientos que los desarrolladores siguen al utilizar Git en un proyecto. El flujo de trabajo puede incluir prácticas como la creación de ramas, la creación de Pull Requests y la gestión de releases.
4. **¿Por qué es importante utilizar ramas en Git?**
 - **Respuesta:**
5. **¿Qué es un conflicto de fusión y cómo se resuelve en Git?**
 - **Respuesta:**
6. **¿Qué es un release en Git?**
 - **Respuesta:**
7. **¿Cómo se puede revertir una fusión en Git?**
 - **Respuesta:**
8. **Clona un repositorio remoto desde GitHub en tu máquina local.**
 - **Respuesta:**
 - `git clone https://github.com/<usuario>/<nombre_repositorio>.git`
9. **Crea una nueva rama llamada `feature/nueva-caracteristica`.**
 - **Respuesta:**
 - `git checkout -b feature/nueva-caracteristica`
10. **Haz algunos cambios en el archivo `index.html`, añádelos al área de preparación y realiza un commit.**
 - **Respuesta:**
 - `git add index.html`
 - `git commit -m "Agregado nuevo diseño a la página de inicio"`
11. **Sube tus cambios a la rama remota `feature/nueva-caracteristica`.**
 - **Respuesta:**
12. **Crea una Pull Request (PR) para fusionar la rama `feature/nueva-caracteristica` a la rama `main`.**
 - **Respuesta:** (Pasos dentro de GitHub)

13. **Fusión de la Pull Request realizada con la rama `main`.**
 - **Respuesta:** (Pasos dentro de GitHub)
14. **Elimina la rama `feature/nueva-caracteristica` después de fusionarla.**
 - **Respuesta:**
15. **Resuelve un conflicto de fusión en el archivo `app.js`.**
 - **Respuesta:** (Pasos dentro de Git)
16. **Crea una etiqueta (tag) llamada `v1.0` para marcar el primer release del proyecto.**
 - **Respuesta:**
17. **Actualiza tu repositorio local con los últimos cambios de la rama `main`.**
 - **Respuesta:**