Mini Curso de Git y GitHub

Configuración Inicial de Git

Antes de usar Git, necesitamos configurarlo con tu nombre y correo electrónico. Esto es importante porque cada commit que realices estará asociado con esta información.
Los comandos son:

- `git config --global user.name "TuNombre"`: Configura tu nombre de usuario globalmente (en tu máquina).
- `git config --global user.email "TuCorreo@example.com"`: Configura tu correo electrónico.

Puedes verificar la configuración con:
- `git config --list`: Lista todas las configuraciones de Git en tu máquina.

Estos pasos son esenciales para que Git registre tus cambios correctamente.

Inicialización de un Repositorio Local

Un repositorio local es una carpeta en tu máquina donde Git realiza el seguimiento de tus archivos y sus cambios. Para inicializarlo:

- Navega hasta la carpeta del proyecto con `cd /ruta/del/proyecto`.
- Ejecuta `git init`: Este comando crea una carpeta oculta llamada `.git`, que contiene toda la información necesaria para que Git funcione.

Para verificar el estado del repositorio:

- `git status`: Te muestra los archivos que no están bajo seguimiento y los que han sido modificados.

Este paso convierte tu carpeta en un repositorio Git.

Añadir Archivos al Repositorio

Para que Git comience a rastrear archivos, primero debes agregarlos al área de preparación (staging area). Esto se hace con:

- `git add archivo.txt`: Agrega un archivo específico al área de preparación.
- `git add .`: Agrega todos los archivos en la carpeta actual.

Una vez agregados, puedes verificar el estado con `git status`. Verás los archivos listos para el commit.

Crear un Commit

Un commit es un punto de control en la historia de tu proyecto. Representa un estado del proyecto en un momento dado. Para crear un commit:

- `git commit -m "Mensaje descriptivo"`: Crea un commit con un mensaje que describa los cambios realizados.

El commit almacena el estado actual de los archivos en el repositorio, permitiéndote regresar a este

punto en el futuro.

Crear un Repositorio Remoto

Un repositorio remoto, como en GitHub, te permite compartir tu proyecto y colaborar con otros. Para conectarlo:

1. Crea un repositorio en GitHub (vacío, sin README ni archivos adicionales).
2. Conecta el repositorio local con el remoto usando:
   - `git remote add origin https://github.com/TuUsuario/TuRepositorio.git`.

Verifica la conexión con:
- `git remote -v`: Muestra las URLs asociadas al repositorio remoto.

Subir Cambios al Remoto (Push)

El comando `git push` sube tus commits locales al repositorio remoto. Los pasos son:

1. Verifica que tu rama esté configurada correctamente:
   - `git branch -M main`: Configura la rama principal como `main`.

2. Sube los cambios:
   - `git push -u origin main`: Sube los commits locales a la rama `main` del remoto.

El flag `-u` configura la rama local para rastrear automáticamente la remota, facilitando futuros `push`.

Sincronizar Cambios Remotos (Pull)

Para sincronizar tu repositorio local con los cambios remotos, usa el comando `git pull`:

- `git pull origin main`: Descarga los cambios de la rama `main` y los fusiona con tu rama local.

Si hay conflictos, Git te pedirá que los resuelvas manualmente editando los archivos en conflicto.

Resolver Conflictos

Un conflicto ocurre cuando los cambios en el remoto y los locales afectan la misma línea de código.

Para resolverlos:

1. Abre los archivos afectados y busca secciones como:

   ```
   <<<<<<< HEAD
   (tus cambios locales)
   =======
   (cambios remotos)
   >>>>>>> nombre-de-la-rama
   ```

2. Edita el archivo para conservar el código correcto.
3. Marca el conflicto como resuelto con `git add archivo_con_conflicto`.
4. Completa el merge con `git commit -m "Conflictos resueltos"`.