Guía Práctica de Git para Principiantes

¡Bienvenido! Esta guía contiene una serie de ejercicios diseñados para enseñarte los fundamentos de Git. Realiza cada paso en tu terminal para familiarizarte con el flujo de trabajo esencial.

Prerrequisitos

- **Git instalado:** Si no lo tienes, descárgalo desde git-scm.com.
- **Una terminal o línea de comandos:** (Terminal en Mac/Linux, Git Bash en Windows).

Ejercicio 1: Configuración Inicial

Antes de empezar a usar Git, es importante que le digas quién eres. Esta información se registrará en cada commit (confirmación) que hagas.

Acción: Abre tu terminal y ejecuta los siguientes comandos, reemplazando los datos de ejemplo con los tuyos.

```
# Configura tu nombre de usuario
git config --global user.name "Tu Nombre Completo"

# Configura tu correo electrónico
git config --global user.email "tu.correo@ejemplo.com"
```

Explicación:
El comando git config se usa para establecer variables de configuración de Git. La bandera --global asegura que esta configuración se aplique a todos los proyectos (repositorios) que manejes en tu sistema. Git usará estos datos para firmar tus contribuciones.

Ejercicio 2: Creando tu Primer Repositorio

Un repositorio (o "repo") es simplemente una carpeta de proyecto cuyo historial de cambios está siendo rastreado por Git.

Acción:

1. Crea una nueva carpeta para tu proyecto y navega hacia ella.
   ```
   mkdir mi-primer-proyecto
   cd mi-primer-proyecto
   ```

2. Inicializa Git dentro de esa carpeta.
   ```
   git init
   ```

Explicación:
El comando git init crea una subcarpeta oculta llamada .git dentro de tu proyecto. En esta carpeta, Git almacena toda la información del historial, los commits y la configuración del repositorio. Con este simple paso, tu carpeta se ha convertido en un repositorio de Git.

Ejercicio 3: El Flujo de Trabajo Básico (Añadir y Confirmar)

Este es el ciclo más común que usarás en Git: modificar archivos, añadirlos al "área de preparación" (Staging Area) y luego confirmar esos cambios en el historial.

Acción:

1. **Revisa el estado del repositorio.**
   git status

   Verás un mensaje que dice que estás en la rama master o main y que no hay nada que confirmar.
2. **Crea un nuevo archivo.**
   # Puedes usar 'touch' en Mac/Linux o 'echo.' en Windows
   touch README.md

3. **Vuelve a revisar el estado.**
   git status

   Ahora Git te mostrará el archivo README.md bajo la sección "Untracked files" (archivos no rastreados).
4. **Añade el archivo al área de preparación (Staging Area).**
   git add README.md

5. **Revisa el estado una vez más.**
   git status

   El archivo ahora aparece en "Changes to be committed" (cambios a ser confirmados).
6. **Confirma los cambios con un mensaje descriptivo.**
   git commit -m "Creación inicial del archivo README"

7. **Visualiza el historial de commits.**
   git log

   Verás tu primer commit, con tu nombre, correo, la fecha y el mensaje que

escribiste.

Explicación:

- git status: Es tu mejor amigo. Te dice exactamente qué está pasando en tu repositorio.
- git add: Prepara los cambios para el próximo "snapshot" (la confirmación). No los guarda permanentemente, solo los pone en una zona de espera. Esto te permite agrupar varios cambios en un solo commit.
- git commit: Toma todo lo que está en el área de preparación y lo guarda como un punto permanente en el historial del proyecto. El mensaje (-m) es crucial para entender qué se hizo en ese cambio.
- git log: Te permite ver la historia completa de los commits del proyecto.

Ejercicio 4: Trabajando con Ramas (Branches)

Las ramas te permiten trabajar en nuevas funcionalidades o arreglos sin afectar la línea principal de desarrollo (que usualmente es main o master).

Acción:

1. **Crea una nueva rama.**
   git branch anadir-descripcion

2. **Cambia a la nueva rama.**
   # El comando 'switch' es más moderno y recomendado
   git switch anadir-descripcion

3. **Modifica el archivo README.md**. Ábrelo con cualquier editor de texto y añade la línea: Este es mi primer proyecto con Git.
4. **Añade y confirma el cambio en esta rama.**
   git add README.md
   git commit -m "Añade descripción al README"

5. **Regresa a la rama principal.**
   git switch main

   Si ahora abres README.md, verás que está vacío. ¡No te asustes! El cambio solo existe en la otra rama.
6. **Fusiona los cambios de la otra rama en main.**
   git merge anadir-descripcion

Ahora, si revisas README.md, verás que el texto que añadiste ha sido incorporado a la rama main.

Explicación:

- git branch <nombre-rama>: Crea una nueva línea de desarrollo.
- git switch <nombre-rama>: Te permite moverte entre las diferentes ramas.
- git merge <nombre-rama>: Trae los cambios de la rama especificada y los fusiona en la rama en la que te encuentras actualmente.

Ejercicio 5: Conectando con un Repositorio Remoto (GitHub)

Un repositorio remoto te permite guardar una copia de tu proyecto en un servidor (como GitHub, GitLab o Bitbucket) para colaborar con otros o simplemente tener una copia de seguridad.

Acción:

1. **Ve a GitHub.com** y crea un nuevo repositorio (haz clic en "New repository"). No lo inicialices con un README u otros archivos; déjalo vacío.
2. **Copia la URL del repositorio** que te proporciona GitHub. Se verá algo como https://github.com/tu-usuario/mi-primer-proyecto.git.
3. **Conecta tu repositorio local con el remoto.**
   git remote add origin https://github.com/tu-usuario/mi-primer-proyecto.git

4. **Sube (push) tus cambios a GitHub.**
   # El flag -u establece la conexión para futuros pushes
   git push -u origin main

Explicación:

- git remote add origin <URL>: Le dice a tu repositorio local que existe un "remoto" llamado origin (el nombre estándar) en la URL especificada.
- git push: Envía tus commits confirmados desde tu repositorio local al repositorio remoto. La primera vez, -u origin main le dice a Git que la rama main local debe estar conectada con la rama main en origin.

Glosario Rápido

- **Repositorio (Repo):** La base de datos que almacena el historial de cambios de un proyecto.
- **Commit:** Una "fotografía" o instantánea guardada del estado de tus archivos en un momento dado.
- **Rama (Branch):** Una línea de desarrollo independiente.

- **Fusionar (Merge):** El acto de combinar los cambios de una rama con otra.
- **Remoto (Remote):** Una versión de tu repositorio alojada en un servidor externo.
- **Push:** Enviar tus commits locales a un repositorio remoto.
- **Pull:** Traer los cambios desde un repositorio remoto a tu repositorio local.