**Resumen de Git y GitHub**

**Git** es un sistema de control de versiones distribuido, lo que significa que permite llevar un seguimiento de los cambios realizados de los archivos a lo largo del tiempo. Ayuda a gestionar proyectos de software, llevar un historial de cambios, y colaborar con otros sin perder el control sobre las versiones del proyecto.

**GitHub** es una plataforma en línea basada en Git que permite almacenar los proyectos de código fuente en repositorios, y colaborar con otros desarrolladores de forma remota.

**Pasos para Usar Git y GitHub:**

**1. Crear un Repositorio en GitHub:**

* Ve a [**GitHub**](https://github.com/) y crea una cuenta si no tienes una.
* Una vez que hayas iniciado sesión, haz clic en el botón verde **"New"** (nuevo) en la parte superior para crear un nuevo repositorio.
* Asigna un **nombre** al repositorio y selecciona si lo quieres hacer **público** o **privado**.
* **No** selecciones la opción de inicializar el repositorio con un archivo **README.md**, ya que en este caso lo vamos a hacer manualmente.

**2. Configurar Git Localmente:**

1. **Instalar Git:** Si aún no tienes **Git** instalado en tu computadora, descarga e instala desde [aquí](https://git-scm.com/downloads).
2. **Configurar Git (si es la primera vez que lo usas):** Abre Git Bash y configura tu nombre y correo electrónico:

bash

Copiar

git config --global user.name "Tu Nombre"

git config --global user.email "tu-email@dominio.com"

**3. Usar Git Desde Git Bash:**

1. **Abrir Git Bash:**
   * Abre la carpeta en la que tienes tu proyecto o archivos, haz clic derecho en cualquier lugar de la carpeta y selecciona **"Git Bash Here"**. Esto abrirá la terminal de **Git Bash** directamente en esa carpeta.
2. **Inicializar el Repositorio Git:** En la consola de Git Bash, inicializa tu proyecto como un repositorio de Git:

bash

git init

Esto creará un directorio .git dentro de tu carpeta, lo que indica que es un repositorio Git.

1. **Enlazar tu Repositorio Local con GitHub (repositorio remoto):** Después de crear el repositorio en GitHub, obtén la URL del repositorio (por ejemplo, https://github.com/tu-usuario/tu-repo.git). Ahora, enlaza tu repositorio local con el remoto en GitHub usando el siguiente comando:

bash

git remote add origin https://github.com/tu-usuario/tu-repo.git

1. **Agregar Archivos al Repositorio Local:** Si ya tienes archivos en la carpeta del proyecto, agrégalos al área de preparación de Git (staging area):

bash

git add .

Este comando agrega **todos** los archivos de la carpeta al área de preparación, listos para ser confirmados en el repositorio local.

1. **Realizar el Primer Commit:** Después de agregar los archivos, realiza tu primer commit para registrar los cambios. Asegúrate de poner un mensaje descriptivo entre comillas:

bash

git commit -m "Primer commit"

1. **Subir los Cambios a GitHub (Push):** Finalmente, sube los cambios al repositorio remoto en GitHub. Usa el siguiente comando para hacer el "push" a la rama **master** (o **main** si es el caso):

bash

git push -u origin master

**Nota**: Si es la primera vez que haces "push", Git te pedirá autenticarte con tus credenciales de GitHub (usuario y contraseña).

**4. Verificación en GitHub:**

* Una vez que hayas subido el código, ve a tu repositorio en GitHub.
* Verás que los archivos se han cargado y están visibles en la página de tu repositorio.

**Flujo de Trabajo Básico en Git y GitHub:**

1. **git init**: Inicializa un nuevo repositorio.
2. **git remote add origin URL**: Enlaza tu repositorio local con el remoto de GitHub.
3. **git add .**: Agrega todos los archivos al área de preparación para ser confirmados.
4. **git commit -m "mensaje"**: Realiza un commit con los cambios añadidos.
5. **git push origin master**: Sube los cambios al repositorio remoto en GitHub.

**Resumen de Comandos Usados:**

* **git init**: Crea un repositorio Git en la carpeta actual.
* **git remote add origin <URL>**: Asocia el repositorio local con uno remoto (en GitHub).
* **git add .**: Agrega todos los archivos del proyecto al área de preparación.
* **git commit -m "mensaje"**: Guarda los cambios realizados en el repositorio local.
* **git push origin master**: Sube los cambios al repositorio remoto en GitHub.

**Consejos Adicionales:**

* Si haces más cambios en tus archivos, solo repite los pasos de **git add .**, **git commit**, y **git push** para seguir subiendo los cambios.
* Puedes verificar el estado de tu repositorio en cualquier momento con **git status**.
* Es recomendable hacer commits frecuentes para mantener un buen historial de cambios.