

## Taller de Git y Github

### ¿Diferencia entre Git y GitHub?

- **Git:** Es el software, la herramienta que se instala en la computadora para rastrear los cambios en los archivos de un proyecto. Es como tener un registro detallado de cada modificación.
- **GitHub:** Es una plataforma en línea (una página web) que aloja repositorios de Git. Facilita la colaboración, el respaldo en la nube y la gestión de proyectos.

### Instalación de Git

- **Windows:** Descargar el instalador desde la página oficial: [Git - Downloads](#) y ejecútalo.
- **MacOs:** debes de instalar primero homebrew: [Homebrew — The Missing Package Manager for macOS \(or Linux\)](#), ejecutando en la terminal el comando que está en esa página.
- Luego abriendo la terminal, y colocando el comando: `$ brew install git`, y listo.
- **Linux:** Usar el comando: `sudo apt-get install git`.

### Creación de cuenta en Github

- Accede a este link: [GitHub](#), crea una cuenta con tu correo principal, y elige un nombre de usuario que te sea fácil recordar.

### Configuración inicial de Git.

- Abre la terminal, colocando en tu buscador "cmd", y por si acaso ejecútalo como administrador. Ejecuta los siguientes comandos, sustituyéndolos por tu nombre de usuario y correo de Github.
- `git config --global user.name "Tu Nombre"`
- `git config --global user.email "tu_correo@ejemplo.com"`

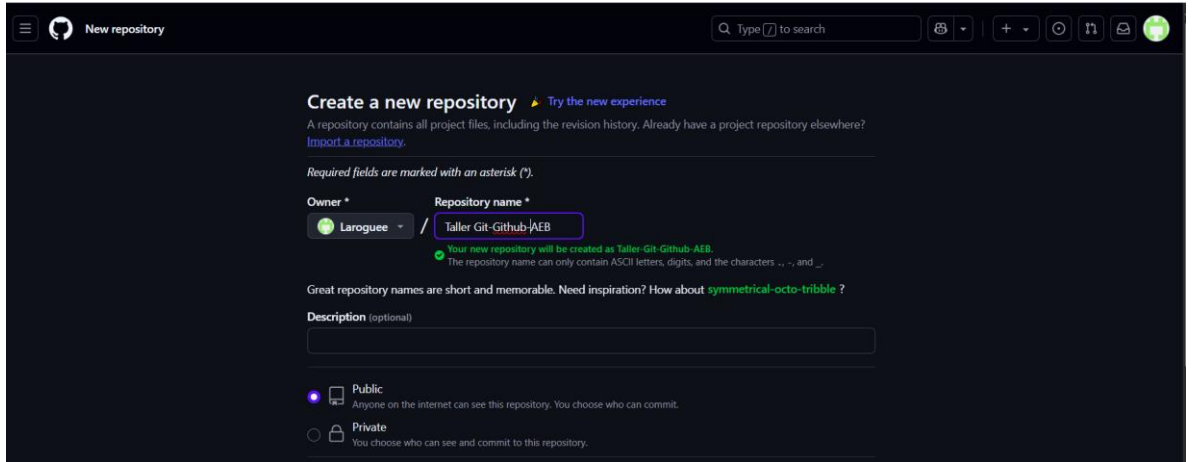
### Ejercicio individual: creando el primer repositorio

1. Creen en el escritorio una carpeta llamada mi-primer-repo
2. Dentro de esa carpeta den click derecho y elijan Open Git Bash Here, eso abrirá una terminal
3. Ejecuten el comando: **git init**
4. Creen un bloc de notas dentro de la carpeta, puedes colocar su top 5 de canciones favoritas o series. No olviden guardar dentro de la carpeta el bloc.
5. Ahora ejecuta el siguiente comando: **Git status** en la misma terminal que se abrió al dar click a Open Git Bash. Verán su bloc de notas como "untracked files" en rojo.
6. Para agregar el archivo al área de preparación debes ejecutar el comando: **git add <nombre\_del\_archivo>**, o **git add .**, este ultimo es para agregar todos los archivos que existen en una carpeta.
7. Para guardar los cambios en el repositorio debes ejecutar el comando: **git commit -m "Mi primer commit"**, lo que esta en comillas lo puedes cambiar.
8. Si quieres ver el historial de commits que se han hecho puedes ejecutar: **git log**

9. Ahora a subir a github el repositorio, entraras a tu cuenta y crearas un nuevo repositorio:



10. Le colocas un nombre, puede ser “mi primer repo”



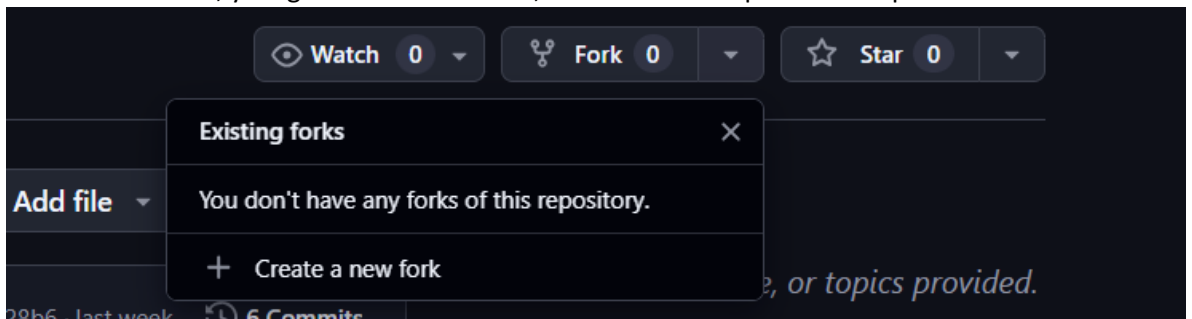
11. Ahora ejecutaras los siguientes comandos en la terminal:
12. `git branch -M main` (se crea la rama principal en la que se subirá el repo)
  13. `git remote add origin` (aquí ira el link del repo de github, sin parentesis)
  14. `git push -u origin main` (este comando es para subir el repo).

Si haces cambios en la carpeta o documento que subiste al repositorio, puedes seguir los siguientes pasos para actualizarlo en github:

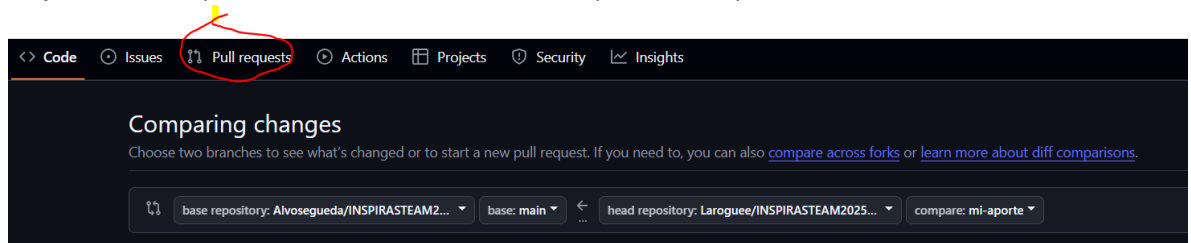
1. `git status`
2. `git add .`
3. `git commit -m "actualización de repo"`
4. `git push origin main`

### Trabajo colaborativo-opción 1

1. accede al siguiente repositorio: <https://github.com/Larogee/Taller-ejercicio-colaborativo-AEB.git>
2. le darás click a for, y luego a **create new fork**, esto hará una copia en tus repositorios



- ahora en tu escritorio darás click derecho y elegirás de nuevo la opción **Open git bash Here**
- ejecutaras el comando **git clone** (aquí el link del repo a clonar, sin paréntesis)
- esto creara una carpeta en tu escritorio con el nombre del repositorio
- Ahora abrirás una terminal de nuevo con Open git bash here, en la carpeta que se genero
- Y ejecutaras los siguientes comandos:
- git branch mi-aporte** (puedes colocar tu nombre de usuario en vez de mi-aporte)
- git checkout mi-aporte**
- ya puedes realizar cambios al documento de texto que esta en el repositorio
- una vez finalices los cambios ejecuta los comandos:
- git push origin mi-aporte**
- para enviar los cambios al repositorio del que hiciste fork, darás click a la **opción crear pull request**, esto lo puedes hacer desde la versión tuya de ese repo



- luego de haber generado el pull request, el dueño del repositorio original fusionara los cambios.

## Trabajo colaborativo -opción 2 agregando colaboradores

### Añadir Colaboradores:

- Ve a la pestaña Settings de tu repositorio.
- En el menú de la izquierda, haz clic en Collaborators and teams.
- Presiona el botón Add people.
- Busca el nombre de usuario de GitHub de cada participante, selecciónalo y asígnale un rol (generalmente Write)
- Cada participante recibirá una invitación por correo electrónico y una notificación en GitHub que deberán aceptar.

### Clonar el repositorio

- Una vez que han aceptado la invitación, cada colaborador debe clonar el repositorio principal directamente a su máquina local. No necesitan hacer un *fork*.
- Usando el comando en una terminal abierta con open git bash en el escritorio:
- git clone <https://github.com/tu-usuario/proyecto-colaborativo-equipo.git>

### Crea una rama personal

- Esta es la regla de oro: **nunca trabajar directamente en la rama main**. Cada colaborador debe crear su propia rama para trabajar de forma aislada.
- Usarás el comando:
- git checkout -b nombre-del-colaborador/su-aporte

### Desarrollar y guardar cambios

1. hacer el trabajo en su propia rama, cada colaborador creará su archivo o modificará los existentes.
2. Guardar los cambios localmente usando los comandos básicos que son
3. Git status
4. Git add .
5. Git commit -m "cambio que se hizo"

### Sincronizar y compartir el trabajo

1. **Actualizar con la Rama Principal:** Antes de subir sus cambios, es una buena práctica asegurarse de que su rama personal está actualizada con los últimos cambios de la rama main.
2. **Usando los siguiente comandos:**
3. Git checkout main
4. Git pull origin main
5. Git checkout nombre-del-colaborador/su-aporte
6. Git merge main

### Subir la rama a Github

1. Usando los comandos
2. Git push origin HEAD
3. Git push origin nombre-del-colaborador/su-aporte

### Revisión y fusión

Crear el Pull Request (Rol: Cada Colaborador):

1. Cada colaborador debe ir a la página del repositorio en GitHub.
2. Aparecerá una notificación amarilla sugiriendo crear un Pull Request para la rama que acaban de subir.
3. Hacen clic y se abrirá un formulario donde deben asegurarse de que la fusión sea de su rama (nombre-del-colaborador/su-aporte) hacia la rama main.
4. Añaden un título y una descripción clara, y crean el Pull Request.

Revisar y Fusionar el Pull Request (Rol: Dueño):

5. En la pestaña Pull Requests del repositorio, verás todas las solicitudes.
6. Puedes abrir cada una, revisar los cambios (Files changed), dejar comentarios o solicitar modificaciones.
7. Si todo está correcto, presionas Merge pull request.
8. **Tip:** Después de fusionar, GitHub te dará la opción de eliminar la rama remota. Es una buena práctica hacerlo para mantener limpio el repositorio.

Actualización Final (Rol: Todos):

9. Una vez que varias ramas han sido fusionadas a main, todos los colaboradores deben actualizar su copia local de main para tener la versión más reciente del proyecto.

10. Generated bash
11. git checkout main
12. git pull origin main