

- **¿Qué es GitHub?**

GitHub es una plataforma online que se usa para guardar y gestionar proyectos. Es ideal para trabajar en equipo, ya que permite que varias personas colaboren en el mismo código, haciendo un seguimiento de los cambios que se realizan.

- **¿Cómo crear un repositorio en GitHub?**

En la pagina principal de **GitHub** se hace click en “**Create New**” y ahí se selecciona **New Repository**

En la próxima pantalla se puede seleccionar una template  
Hay que elegir el nombre del repositorio y escribirlo  
Se debe seleccionar si es publico o privado

Y para finalizar Create Repository

- **¿Cómo crear una rama en Git?**

En la consola de comandos se escribe

```
git branch <nombre_del_branch>
```

- **¿Cómo cambiar a una rama en Git?**

```
git checkout <nombre_del_branch>
```

- **¿Cómo fusionar ramas en Git?**

```
git merge <nombre_del_branch_a_fusionar>
```

- **¿Cómo crear un commit en Git?**

```
git commit -m “Mensaje o descripción de lo que se hizo”
```

- **¿Cómo enviar un commit a GitHub?**

```
git push origin <nombre_del_branch>
```

- **¿Qué es un repositorio remoto?**

Un repositorio remoto es una versión del repositorio que se encuentra alojada en un servidor o en una plataforma como GitHub.

- **¿Cómo agregar un repositorio remoto a Git?**

```
git remote add origin <URL_del_repositorio >
```

- **¿Cómo empujar cambios a un repositorio remoto?**

```
git push origin <nombre_de_la_rama>
```

- **¿Cómo tirar de cambios de un repositorio remoto?**

```
git pull origin <nombre_de_la_rama>
```

- **¿Qué es un fork de repositorio?**

Un fork es una copia de un repositorio que se crea en la cuenta propia de GitHub

- **¿Cómo crear un fork de un repositorio?**

En el repositorio que se quiere hacer el fork, se hace click en el botón Fork, se selecciona la cuenta o la organización y se hace click en Create Fork

- **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?**

En el proyecto clonado realiza los cambios y súbelos a una nueva rama. Luego ir al fork en **GitHub**, se hace clic en la pestaña **Pull Requests**, luego en **New Pull Requests**.

Agrega un título y descripción explicando tus cambios. Selecciona la rama destino (generalmente main o master) del repositorio original y finalmente clic en **Create pull request**.

- **¿Cómo aceptar una solicitud de extracción?**

En GitHub se hace clic en **Pull Requests**, se selecciona el **Pull Request** que se quiere revisar. Los cambios se verán en la pestaña **Files Changed**. Revisa los cambios y si se aceptan los cambios se hace clic en **Merge Pull Request**. Confirma el merge y elimina la rama si es necesario.

- **¿Qué es un etiqueta en Git?**

Una **etiqueta (tag)** es un marcador que se usa para señalar un punto específico en el historial de Git, generalmente para indicar versiones importantes como lanzamientos

- **¿Cómo crear una etiqueta en Git?**

Para agregar una etiqueta

```
git tag v1.0
```

Para agregar etiqueta con mensaje

```
git tag -a v1.0 -m "Versión 1.0"
```

- **¿Cómo enviar una etiqueta a GitHub?**

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

Y si hay mas de una etiqueta

```
git push origin --tags
```

- **¿Qué es un historial de Git?**

El **historial de Git** es un registro de todos los commits realizados en un repositorio, mostrando cómo ha cambiado el proyecto con el tiempo.

- **¿Cómo ver el historial de Git?**

```
git log
```

Para un resumen compacto.

```
git log --oneline
```

Para un historial gráfico

```
git log --graph --oneline --all.
```

- **¿Cómo buscar en el historial de Git?**

```
git log --grep="término_a_buscar"
```

O para buscar por un autor:

```
git log --author="Nombre"
```

- **¿Cómo borrar el historial de Git?**

Existen varias opciones pero siempre giran alrededor del comando `git reset`.

`git reset` quita todos los archivos y carpetas del proyecto

`git reset nombre-archivo` quita el archivo indicado

`git reset nombre-carpeta` quita todos los archivos de esa carpeta

`git reset nombre-carpeta/nombre-archivo` quita ese archivo que está a su vez dentro de esa carpeta

`git reset nombre-carpeta/*.extensión` quita todos los archivos que cumplen con la condición indicada previamente dentro de esa carpeta

Siempre hay que tener presente que estos cambios se ejecutan en el repositorio local y que si se trabaja en equipo y hay que pushearlos es conveniente hablarlo previamente.

- **¿Qué es un repositorio privado en GitHub?**

Es un repositorio cuyo acceso está restringido. Solo los usuarios con permiso, como el propietario o colaboradores invitados, pueden acceder.

- **¿Cómo crear un repositorio privado en GitHub?**

Son los mismos pasos que la creación de un repositorio, detallado con anterioridad, pero con la diferencia de que en el apartado de tipo de visibilidad hay que seleccionar Private.

- **¿Cómo invitar a alguien a un repositorio privado en GitHub?**

Ve al repositorio y entra en "Settings".

Selecciona "Collaborators".

Agrega el usuario y envía la invitación

- **¿Qué es un repositorio público en GitHub?**

Es un repositorio que puede ser visto y clonado por cualquier persona

- **¿Cómo crear un repositorio público en GitHub?**

Son los mismos pasos detallados con antelación, pero con la diferencia de que en el apartado de tipo de visibilidad hay que seleccionar Public.

- **¿Cómo compartir un repositorio público en GitHub?**

Como es de acceso ilimitado basta con copiar la URL y compartirla en el medio elegido