

1)

- **¿Qué es GitHub?**

GitHub es una plataforma de desarrollo colaborativo basada en Git. Permite a los desarrolladores alojar repositorios, colaborar en proyectos de software, realizar seguimiento de cambios y gestionar versiones de código de manera eficiente.

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

1. inicia sesión en GitHub.
2. Haz clic en el botón + en la esquina superior derecha y selecciona Nuevo repositorio.
3. Ingresa un nombre para tu repositorio, una descripción opcional y configura la visibilidad (público o privado).
4. Haz clic en Crear repositorio.

- **¿Cómo crear una rama en Git?**

Ejecuta el comando en la terminal:

```
git branch nombre_rama
```

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

Para cambiar a una rama usa:

```
git checkout nombre_rama
```

- **¿Cómo fusionar ramas en Git?**

ejecuta:

```
git merge nombre_rama
```

- **¿Cómo crear un commit en Git?**

Agrega los cambios con:

1. git add . Crea un commit con un mensaje descriptivo:

```
git commit -m "Descripción del cambio"
```

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

Sube los commits locales a GitHub con:

```
git push origin nombre_rama
```

- **¿Qué es un repositorio remoto?**

Un repositorio remoto es una copia alojada en un servidor externo, como GitHub, que permite compartir código entre desarrolladores.

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

Ejecuta el siguiente comando para agregar un repositorio remoto:

```
git remote add origin URL_del_repositorio
```

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

```
git push origin nombre_rama
```

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

```
git pull origin nombre_rama
```

- **¿Qué es un fork de repositorio?**

Un **fork** es una copia de un repositorio que te permite modificarlo sin afectar el original.

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

1. Ve al repositorio en GitHub.
2. Haz clic en el botón "Fork" en la parte superior derecha.
3. Selecciona la cuenta donde deseas alojar el fork.

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

1. Realiza cambios en tu fork.
2. Ve al repositorio original y haz clic en "Compare & pull request".
3. Añade una descripción y envía la solicitud.

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

1. Abre el repositorio en GitHub.

2. Ve a la pestaña "Pull Requests".
3. Revisa los cambios y haz clic en "Merge Pull Request".

¿Qué es una etiqueta en Git?

Una etiqueta (**tag**) es una marca que se usa para identificar versiones específicas del código, como versiones de lanzamiento.

¿Cómo crear una etiqueta en Git?

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

¿Cómo enviar una etiqueta a GitHub?

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

¿Qué es un historial de Git?

El historial de Git es el registro de todos los commits realizados en el repositorio.

¿Cómo ver el historial de Git?

```
git log
```

¿Cómo buscar en el historial de Git?

```
git log --grep "texto_a_buscar"
```

¿Cómo borrar el historial de Git?

El historial no se borra fácilmente. Puedes reescribir la historia con:

```
git reset --hard commit_id
```

Pero esto puede afectar el estado del proyecto y es recomendable hacerlo con precaución.

¿Qué es un repositorio privado en GitHub?

Un repositorio privado es aquel que solo es accesible para sus colaboradores y propietarios.

¿Cómo crear un repositorio privado en GitHub?

1. Ve a "Nuevo repositorio".
2. Marca la opción "Privado".
3. Configura los detalles y haz clic en "Crear repositorio".

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

1. Ve al repositorio y abre la pestaña "Configuración".
2. Selecciona "Colaboradores".
3. Agrega el usuario con su nombre de GitHub y envía la invitación.

¿Qué es un repositorio público en GitHub?

Un repositorio público es accesible para cualquier persona y permite la colaboración abierta.

¿Cómo crear un repositorio público en GitHub?

Sigue el mismo proceso que con un repositorio privado, pero selecciona "Público" en la configuración.

¿Cómo compartir un repositorio público en GitHub?

Comparte la URL del repositorio o usa la opción de "Clonar" para facilitar el acceso.

2)

1 Creación del repositorio: Se realizó un **fork** del repositorio indicado y se configuró como **público**, incluyendo un archivo **README.md**.

2 Agregar un archivo: Se creó **mi-archivo.txt** desde GitHub con contenido de prueba y se guardó con un **commit** descriptivo.

Comandos en Git Bash:

```
echo "Este es mi archivo" > mi-archivo.txt
git add .
git commit -m "Agregando mi-archivo.txt"
git push origin main
```

3 Creación de una rama: Se creó la rama **nueva-rama** para separar cambios y se pasó a trabajar en ella.

Comandos en Git Bash:

```
git branch nueva-rama
```

4 Subida de cambios: Se agregó **archivo-nuevo.txt** en la nueva rama y se subió a GitHub.

Comandos en Git Bash:

```
echo "Cambio en la nueva rama" > archivo-nuevo.txt
git add .
git commit -m "Agregando archivo-nuevo.txt en nueva-rama"
git push origin nueva-rama
```

3)

<https://github.com/AgustinValentinAlbornoz/conflict-exercise>