

## Actividades 1)

Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):

### ¿Qué es GitHub?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

### ¿Cómo crear un repositorio en GitHub?

- 1- Accede a GitHub e inicia sesión.
- 2- Haz clic en el botón "+" en la parte superior derecha y selecciona **"New repository"**.
- 3- Ingresa un **nombre** para el repositorio.
- 4- (Opcional) Agrega una descripción.
- 5- Selecciona si quieres que el repositorio sea **público o privado**.
- 6- (Opcional) Marca la opción para inicializar con un `README.md`, agregar `.gitignore` o una licencia.
- 7- Haz clic en **"Create repository"**.

### ¿Cómo crear una rama en Git?

`git branch NOMBRE-DE-LA-RAMA` (crear nueva rama)

### ¿Cómo cambiar a una rama en Git?

`git checkout NOMBRE-DE-LA-RAMA` (se cambio a la rama NOMBRE-DE-LA-RAMA)

### ¿Cómo fusionar ramas en Git?

- 1\_ Antes de fusionar, asegúrate de estar en la rama donde quieres aplicar los cambios (por lo general, `main`)
- 2\_ Para fusionar la rama `mi-rama` en `main`, usa: **`git merge mi-rama`**

### ¿Cómo crear un commit en Git?

Con el comando `git commit -m "ESCRIBIR COMMIT"`

### ¿Cómo enviar un commit a GitHub?

Con el comando `git push u-`

### ¿Qué es un repositorio remoto?

Un **repositorio remoto** es una versión de tu repositorio almacenada en un servidor en línea. Permite que varias personas trabajen en el mismo proyecto desde diferentes lugares, manteniendo el código sincronizado.

### ¿Cómo agregar un repositorio remoto a Git?

- 1\_ Ve a GitHub e inicia sesión.
- 2\_ Crea un nuevo repositorio (`New repository`).
- 3\_ Copia la URL del repositorio remoto
- 4\_ Vincular el repositorio remoto con tu proyecto local  
`git remote add origin *URL*`
- 5\_ Subir el código al repositorio remoto  
`git push -u`

### ¿Cómo empujar cambios a un repositorio remoto?

Para empujar los cambios al repositorio remoto se usa el comando `git push -u`

### ¿Qué es un fork de repositorio?

Un **fork** es una copia de un repositorio que se crea en tu propia cuenta de GitHub. Te permite hacer cambios sin afectar el repositorio original.

### ¿Cómo crear un fork de un repositorio? ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?

#### Hacer un fork del repositorio

- 1\_ Ir al repositorio original en GitHub que quieres forkar.
- 2\_ En la esquina superior derecha, haz clic en el botón **"Fork"**.
- 3\_ GitHub creará una copia del repositorio en tu cuenta.
- 4\_ Ahora tienes tu propia versión del repositorio en

#### Crear Un Pull request

- 1\_ Ve a tu repositorio en GitHub
- 2\_ GitHub te sugerirá crear un **pull request** cuando detecte cambios en una rama.
- 3\_ Haz clic en **"Compare & pull request"**.
- 4\_ Asegúrate de que el PR se dirige a la rama correcta (por lo general, `main` o `develop`).
- 5\_ Escribe un título y una descripción explicando los cambios.
- 6\_ Haz clic en **"Create pull request"**.

### ¿Cómo aceptar una solicitud de extracción?

- 1\_ Ve a tu repositorio en GitHub.
- 2\_ Haz clic en la pestaña **"Pull Requests"**.
- 3\_ Busca la solicitud de extracción que quieres revisar y ábrela.

### ¿Qué es un etiqueta en Git?

Una **etiqueta (tag)** en Git es un marcador que se usa para señalar un punto específico en la historia del repositorio. Las etiquetas son similares a las ramas, pero no cambian con el tiempo. Son útiles para identificar versiones estables del código.

**¿Cómo crear una etiqueta en Git?**

Con el comando `git tag -a`

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

Con el comando `git push --tags`

**¿Qué es un historial de Git?**

El historial de Git es el registro de todos los cambios realizados en un repositorio a lo largo del tiempo. Contiene información sobre cada commit, como: Quién hizo el cambio (autor), Cuándo se hizo, Qué cambios se realizaron, Mensaje del commit describiendo el cambio, Identificador único (hash) de cada commit.

**¿Cómo ver el historial de Git?**

Con el comando `git log` o el comando `git log --oneline` para el historial más completo.

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

Si recuerdas parte del mensaje de un commit, usa:

`git log --grep="palabra clave"`

Si quieres ver los commits hechos por una persona específica:

`git log --author="nombre o email"`

Si quieres saber quién cambió un archivo y cuándo:

`git log -- nombre-del-archivo`

Para buscar commits entre dos fechas:

`git log --since="2024-03-01" --until="2024-03-31"`

Si quieres encontrar commits donde se haya añadido o modificado una palabra clave en el código:

`git log -S "funcion_suma"`

Si solo quieres ver los identificadores de los commits sin detalles:

`git log --pretty=format:"%h - %s" --grep="palabra clave"`

**¿Cómo borrar el historial de Git?**

1\_ `git checkout --orphan nueva-rama`

2\_ `git add .`

3\_ `git commit -m "Reinicio del historial"`

4\_ `git branch -D main` # Elimina la rama principal

5\_ `git branch -m main` # Renombra la nueva rama a 'main'

6\_ `git push --force origin main` # Sobrescribe el repositorio remoto

Esto elimina todo el historial anterior del repositorio remoto y lo reemplaza con un único commit.

### ¿Qué es un repositorio privado en GitHub?

Un **repositorio privado** en GitHub es un repositorio al que solo tienen acceso su creador y las personas a las que este les haya otorgado permisos.

### ¿Cómo crear un repositorio privado en GitHub?

- 1- Accede a GitHub e inicia sesión.
- 2- Haz clic en el botón "+" en la parte superior derecha y selecciona **"New repository"**.
- 3- Ingresa un **nombre** para el repositorio.
- 4- (Opcional) Agrega una descripción.
- 5- Selecciona sea **privado**.
- 6- (Opcional) Marca la opción para inicializar con un `README.md`, agregar `.gitignore` o una licencia.
- 7- Haz clic en **"Create repository"**.

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

- 1\_ Entra al repositorio y ve a "Settings".
- 2\_ En la barra lateral, selecciona "Collaborators".
- 3\_ Haz clic en "Add people" y escribe el nombre de usuario o correo de la persona.
- 4\_ Selecciona el nivel de permiso (solo lectura, edición, administración).
- 5\_ La persona recibirá una invitación y podrá acceder al repositorio.

### ¿Qué es un repositorio público en GitHub?

Un **repositorio público** en GitHub es un repositorio de código abierto al que **cualquier persona puede acceder, ver y clonar**. A diferencia de los repositorios privados, los repos públicos permiten compartir código con la comunidad, colaborar con otros desarrolladores y contribuir a proyectos de código abierto.

### ¿Cómo crear un repositorio público en GitHub?

- 1- Accede a GitHub e inicia sesión.
- 2- Haz clic en el botón "+" en la parte superior derecha y selecciona **"New repository"**.
- 3- Ingresa un **nombre** para el repositorio.
- 4- (Opcional) Agrega una descripción.
- 5- Selecciona sea **público**.
- 6- (Opcional) Marca la opción para inicializar con un `README.md`, agregar `.gitignore` o una licencia.
- 7- Haz clic en **"Create repository"**.

### ¿Cómo compartir un repositorio público en GitHub?

Compartir el link y dejar que hagan un fork o clonen el repositorio.