

Desarrollo de actividades.

1) GitHub es una plataforma basada en la nube que permite a los desarrolladores almacenar, gestionar y colaborar en proyectos de código utilizando Git, un sistema de control de versiones.

2) Para crear un repositorio en GitHub tenemos que seguir los siguientes pasos:

1. Ve a GitHub e inicia sesión.
2. Haz clic en el botón **New** (o ve a <https://github.com/new>).
3. Escribe un nombre para el repositorio.
4. Elige la visibilidad:
 - **Public**: Cualquiera puede verlo.
 - **Private**: Solo tú y las personas con acceso pueden verlo.
5. (Opcional) Marca la casilla Add a README file para incluir un archivo de descripción.
6. Haz clic en **Create repository**.

3) El comando para crear una nueva rama en Git es:

git branch nombre-rama

4) El comando para cambiar de rama en Git es:

git checkout nombre-rama

5) El comando para fusionar una rama con otra en Git es:

git merge nombre-rama

6) El comando para crear un commit en Git es:

git commit -m "mensaje"

7) El comando para enviar un commit a GitHub es:

git push -u origin master

8) Un **repositorio remoto** es una versión de tu código almacenada en un servidor en la nube, como GitHub, GitLab, etc. A diferencia de un **repositorio local**, que está en tu computadora, el remoto permite colaborar con otros desarrolladores y acceder a tu código desde cualquier lugar.

9) El comando para agregar un repositorio remoto a Git es:

git remote add origin url

9) El comando para empujar los cambios a un repositorio remoto es:

git push origin main

10) Un **fork** es una copia de un repositorio en tu cuenta de GitHub. Sirve para hacer cambios sin afectar el proyecto original y, si es necesario, proponer mejoras al repositorio original mediante un **pull request**.

11) Para crear un fork de un repositorio sigue los siguientes pasos:

1. Ve al repositorio que quieres copiar en GitHub.
2. Haz clic en el botón **Fork** (arriba a la derecha).
3. GitHub creará una copia del repositorio en tu cuenta.

12) Para enviar una **solicitud de extracción (Pull Request)** a un repositorio en GitHub sigue los siguientes pasos:

1. **Haz un fork** del repositorio en GitHub.
2. **Clona tu fork** en tu PC:
3. `git clone https://github.com/tu_usuario/repositorio_forkeado.git`
4. `cd repositorio_forkeado`
5. **Crea una nueva rama:**
6. `git checkout -b mi-nueva-rama`
7. **Realiza cambios y confírmalos:**
8. `git add .`
9. `git commit -m "Descripción del cambio"`
10. **Sube los cambios a tu fork:**
11. `git push origin mi-nueva-rama`
12. **En GitHub, ve a tu repositorio forkeado** y haz clic en **"Compare & pull request"**.
13. **Describe tus cambios** y presiona **"Create pull request"**.

13) Para aceptar una **solicitud de extracción (Pull Request)** en GitHub, sigue estos pasos:

1. **Ve al repositorio en GitHub** donde se creó el Pull Request.
2. **Haz clic en la pestaña "Pull requests"**.
3. **Selecciona la solicitud de extracción** que desees aceptar.
4. **Revisa los cambios** en la pestaña "Files changed".
5. **(Opcional) Comenta o solicita cambios** si es necesario.
6. Si todo está bien, haz clic en **"Merge pull request"**.
7. Confirma la fusión presionando **"Confirm merge"**.
8. **(Opcional) Elimina la rama** si ya no es necesaria.

14) Una **etiqueta (tag)** en Git marca un punto específico en la historia del proyecto, generalmente para versiones importantes.

15) El comando para crear una etiqueta en Git es:

git tag -a v1.0 -m "Versión 1.0"

16) El comando para enviar una etiqueta a GitHub es:

git push origin v1.0

Para enviar todas las etiquetas:

git push origin --tags

17) El historial en git es el registro de todos los commits realizados en un repositorio.

18) El comando para ver el historial en Git es:

git log

19) El comando para buscar en el historial en Git es:

git log --grep="palabra clave"

20) El comando para borrar el historial de Git es:

Git no permite borrar el historial completamente, pero puedes **resetear el repositorio**:

rm -rf .git

git init

Advertencia! Esto borrará todo el historial.

21) Un repositorio privado en GitHub es un repositorio accesible solo para los colaboradores autorizados.

22) Para crear un repositorio privado en GitHub, sigue los siguientes pasos:

1. En GitHub, haz clic en "**New repository**".
2. Elige un nombre y marca la opción "**Private**".

3. Haz clic en "**Create repository**".

23) Para invitar a alguien a un repositorio privado en GitHub, sigue los siguientes pasos:

1. Ve al repositorio en GitHub.
2. Haz clic en "**Settings**" → "**Collaborators**".
3. Agrega el usuario y envía la invitación.

24) Un repositorio público en GitHub es un repositorio visible para cualquier persona en GitHub.

25) Para compartir un repositorio público en GitHub comparte la URL del repositorio, por ejemplo:

<https://github.com/usuario/repositorio>