

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

- **¿Qué es GitHub?**

GitHub es una plataforma en línea que permite guardar proyectos con control de versiones usando Git. Sirve para colaborar con otras personas, compartir código y gestionar cambios.

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

En tu perfil de GitHub, hacé clic en el botón con el signo "+" en la parte superior derecha y seleccioná "New repository". Luego completás el nombre y configuración, y hacés clic en "Create repository".

- **¿Cómo crear una rama en Git?**

Una rama en Git se crea con el comando `git branch nombre-rama`. Esto te permite trabajar en cambios nuevos sin afectar la rama principal (generalmente llamada main o master).

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

Usá el comando `git checkout nombre-rama` o en versiones más nuevas `git switch nombre-rama`. Esto cambia tu trabajo a esa rama específica.

- **¿Cómo fusionar ramas en Git?**

Primero cambiás a la rama en la que querés aplicar los cambios (`git checkout main`) y luego usás `git merge nombre-rama` para combinar los cambios de otra rama.

- **¿Cómo crear un commit en Git?**

Un commit se crea luego de agregar archivos con `git add`, usando el comando `git commit -m "Mensaje del cambio"`. Sirve para guardar un conjunto de cambios.

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

Después de hacer un commit, usá `git push` para enviar los cambios al repositorio remoto (por ejemplo, en GitHub).

- **¿Qué es un repositorio remoto?**

Es una copia del repositorio que está alojada en un servidor como GitHub, GitLab o Bitbucket. Sirve para compartir el código con otros usuarios.

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

Usás el comando `git remote add origin url-del-repositorio`. Esto enlaza tu repositorio local con el repositorio en línea.

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

Usá el comando `git push origin nombre-rama` para enviar los cambios locales a la rama correspondiente en el repositorio remoto.

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

Se usa `git pull` para descargar y combinar los cambios del repositorio remoto en tu copia local.

- **¿Qué es un fork de repositorio?**

Es una copia de un repositorio en tu cuenta de GitHub. Te permite hacer cambios sin afectar el original.

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

En la página del repositorio original en GitHub, hacé clic en el botón “Fork” (arriba a la derecha). Esto lo copia a tu cuenta.

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

Desde tu fork, luego de hacer cambios y subirlos, hacé clic en “Pull Request” y seguí los pasos para proponer que tus cambios se integren al repositorio original.

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

El dueño del repositorio debe revisar la solicitud y hacer clic en “Merge pull request” para aceptar los cambios y combinarlos con el repositorio original.

- **¿Qué es una etiqueta en Git?**

Es una forma de marcar puntos específicos en la historia del proyecto, como versiones o lanzamientos importantes.

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

Usá el comando `git tag nombre-etiqueta` o `git tag -a nombre -m "mensaje"` para crear una etiqueta.

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

Usá el comando `git push origin nombre-etiqueta` para subir esa etiqueta al repositorio remoto.

- **¿Qué es un historial de Git?**

Es el registro de todos los commits hechos en un proyecto. Permite ver los cambios y quién los hizo.

- **¿Cómo ver el historial de Git?**

Con el comando `git log`, podés ver una lista de commits con detalles como autor, fecha y mensaje.

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

Podés usar `git log --grep="palabra clave"` para buscar mensajes específicos, o `git log -p` para ver los cambios realizados.

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

Git no permite borrar el historial directamente, pero podés reescribirlo con comandos como `git rebase` o `git filter-branch`. Esto es avanzado y se debe usar con cuidado.

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

Es un repositorio que solo puede ser visto por vos y por las personas que invites. No es visible públicamente.

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

Al crear un repositorio nuevo, marcá la opción “Private” antes de hacer clic en “Create repository”.

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

Desde la página del repositorio, andá a la pestaña “Settings” > “Collaborators” y agregá el nombre de usuario de la persona que querés invitar.

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

Es un repositorio visible para cualquier persona en internet. Todos pueden verlo, pero solo vos podés modificarlo.

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

Al crear un repositorio nuevo, marcá la opción “Public” y luego hacé clic en “Create repository”.

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

Solo tenés que copiar la URL del repositorio (que aparece en la barra de direcciones del navegador) y compartirla con quien quieras.