

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

- ¿Qué es GitHub?

Una plataforma en línea que permite almacenar proyectos que usan el sistema de control de versiones Git. Almacena los repositorios en la nube, lo que facilita el trabajo colaborativo.

- ¿Cómo crear un repositorio en GitHub?

Ingresamos a la página principal: github.com → generamos usuario y contraseña válidos → nuevo repositorio (asignamos nombre válido), definimos si es público o privado y agregamos una descripción opcional → Creamos el repositorio

- ¿Cómo crear una rama en Git?

Con el comando `git branch -m <nombre de la rama nueva>`

- ¿Cómo cambiar a una rama en Git?

Con el comando `git checkout <nombre de la rama>`

- ¿Cómo fusionar ramas en Git?

Con el comando `git merge <nombre de la rama a sumar>`

Con esto se fusionará la rama nombrada con la rama sobre la que estamos posicionados.

- ¿Cómo crear un commit en Git?

Con el comando `git commit -m "mensaje que describe la modificación"`

- ¿Cómo enviar un commit a GitHub?

Con el comando `git push -u <nombre del repositorio remoto + nombre de la rama a subir>` Ej: `git push -u origin main`

Si es la primera vez y debemos vincular el repositorio local con el remoto: Vamos a nuestra cuenta en GitHub y entramos a la opción de crear un nuevo repositorio. Debemos nombrarlo. Una vez creado copiamos el comando `git remote add <nombre del repositorio> <url del repositorio remoto>` y lo pegamos en la terminal local. De ahí en más sólo hay que repetir el primer comando.

- ¿Qué es un repositorio remoto?

Es un espacio en la nube (alojado en un servidor) que sirve para almacenar contenido al que se puede acceder desde distintos dispositivos. Según la función y los permisos dados para el repositorio, éste puede ser público o privado. Y pueden trabajar en él varias personas o una sola.

- ¿Cómo agregar un repositorio remoto a Git?

Si es la primera vez y debemos vincular el repositorio local con el remoto: Vamos a nuestra cuenta en GitHub y entramos a la opción de crear un nuevo repositorio. Debemos nombrarlo. Una vez creado copiamos el comando `git remote add <nombre del repositorio + url del repositorio remoto>` y lo pegamos en la terminal local. Ej: `git remote add origin https://github.com/usuario/mi-proyecto.git`

- ¿Cómo empujar cambios a un repositorio remoto?

Con el comando `git push -u <nombre del repositorio remoto + nombre de la rama a subir>` Ej: `git push -u origin main`

- ¿Cómo tirar de cambios de un repositorio remoto?

Si entendemos “tirar de cambios” como deshacernos de modificaciones en el repositorio remoto, la forma es con el comando: `git push origin --delete <nombre de la rama>` y con eso nos deshacemos de toda la rama (o branch) que esté alojada en nuestro repositorio remoto.

- ¿Qué es un fork de repositorio?

Un fork es una copia completa de un repositorio de GitHub que se crea dentro de tu cuenta

- ¿Cómo crear un fork de un repositorio?

Entramos en nuestra cuenta de github. Ingresamos al link del repositorio que queremos copiar en nuestra cuenta. Hacemos clic en el botón fork y modificamos la descripción si eso ayuda a nuestra organización. Con eso ya podemos modificar los archivos sin alterar el repositorio original.

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

Vamos al repositorio original en GitHub. Apretamos el botón “Fork” de la esquina superior derecha. Luego hacemos git clone en el área de trabajo para modificar el código.

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

Una vez que hacemos el fork y clonamos el repositorio (`git clone`). Creamos una nueva rama (`git branch -m nuevarama`). Nos aseguramos de estar posicionados en la nueva rama, hacemos los cambios que necesitamos, los guardamos (`git add .`), hacemos un `git commit -m “ descripción”`, donde damos cuenta de los cambios que queríamos hacer. Subimos los cambios a nuestro repositorio (`git push origin nuevarama`). Abrimos nuestro github, vamos al lugar del fork donde encontraremos el botón que dice **“Compare & pull request”**, al presionarlo podemos añadir un título y una descripción de los cambios. Una vez completado, presionamos **“Create pull request”** y nuestras modificaciones se envían al dueño original del proyecto que decidirá qué hacer con nuestro trabajo.

- ¿Cómo aceptar una solicitud de extracción?

Estando en nuestro repositorio de github vamos al apartado de Pull requests encima de la lista de archivos del proyecto donde encontraremos las pull request recibidas. Podemos revisarlas y comentarlas. Si queremos aceptar cambios presionamos **“Merge pull request”**. Que tiene distintas opciones para el merge:

- Create a merge commit:** (por defecto): fusiona la PR con un commit adicional.

- Squash and merge:** combina todos los commits de la PR en uno solo.

- Rebase and merge:** reescribe el historial como si los commits fueran hechos sobre la rama base directamente.

Elegimos la opción que queremos y presionamos **“Confirm merge”**.

- ¿Qué es una etiqueta en Git?

Es una referencia a un punto específico en el historial de commit. Puede ser una etiqueta ligera: (`git tag <nombre>`) solo para remarcar el commit. O puede ser una etiqueta anotada (`git tag -a <nombre> -m "mensaje"`) donde además de poder incluir un mensaje, podemos obtener metadatos de la etiqueta (quién la creó y cuándo)

- ¿Cómo crear una etiqueta en Git?

Para etiquetas ligeras: `git tag <nombre>`

Para etiquetas anotadas: `git tag -a <nombre> -m "mensaje"`.

- ¿Cómo enviar una etiqueta a GitHub?

Para subir una etiqueta: `git push origin <nombre de la etiqueta>`

Para subir todas las etiquetas: `git push origin --tags`

- ¿Qué es un historial de Git?

Es el registro de todos los cambios realizados en un repositorio, organizados cronológicamente como una serie de commits. Sirve para ver qué cambios se hicieron, quién los hizo y cuándo. El historial nos ayuda a ver la evolución del proyecto y recuperar versiones anteriores si llegamos a necesitarlas.

- ¿Cómo ver el historial de Git?

Se utiliza el comando `git log` para ver el historial completo de commits de la rama en que estamos posicionados al momento de ejecutar el comando. Cada entrada incluye: hash, autor, fecha y mensaje del commit.

- ¿Cómo buscar en el historial de Git?

Hay distintas formas de buscar en el historial.

- Búsqueda por autor: `git log --author="nombre autor"`. También se puede buscar al autor y ver qué cambios hizo: `git log -p --author="nombre autor"`
- Búsqueda por palabra clave: `git log --grep="palabra clave"`. Esta forma es combinable: `git log --grep="p clave 1" --grep="p clave 2" --all-match`
- Búsqueda por archivo específico: `git log index.html`
- Búsqueda por fecha: `git log --since="aaaa-mm-dd"`
- Búsqueda simplificada: `git log --oneline --graph --all`. El historial se ve compacto, en forma de ramas

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio al que solo puede ver y acceder la persona que lo crea y las personas a las que éste autorice. Con esta opción se puede controlar quién puede leer, clonar o contribuir al código.

- ¿Cómo crear un repositorio privado en GitHub?

Seguimos todos los pasos para la creación de un repositorio y tildamos la opción "Private" antes de generarlo.

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

Siendo el administrador del repositorio ingresamos a éste, nos vamos a la opción "Settings" del menú de arriba a la derecha. Seleccionamos "Manage access" y "Invite a collaborator". Escribimos el nombre del usuario de Github o su correo electrónico, lo seleccionamos y presionamos "Send Invitation".

- ¿Qué es un repositorio público en GitHub?

Significa que el repositorio es visible para todos y cualquier usuario puede ver el código, descargarlo, clonar el repositorio o hacer un fork. Se puede gestionar quién tiene acceso a la edición de la versión propia del código. Por lo demás, el acceso es libre.

- ¿Cómo crear un repositorio público en GitHub?

Seguimos todos los pasos para la creación de un repositorio y tildamos la opción "Public" antes de generarlo.

- ¿Cómo compartir un repositorio público en GitHub?

Basta con compartir el enlace