Guía de preguntas sobre Git y GitHub

1. ¿Qué es GitHub?

GitHub es una plataforma en línea que permite almacenar proyectos utilizando el sistema de control de versiones Git. Permite colaborar en proyectos de software, realizar revisiones, manejar versiones, y compartir código desde cualquier lugar.

2. ¿Cómo crear un repositorio en GitHub?

1. Ingresar a https://github.com

2. Hacer clic en "New" o "Crear repositorio nuevo"

3. Ingresar nombre, descripción y seleccionar visibilidad (público o privado)

4. Opcionalmente añadir README, .gitignore y licencia

5. Hacer clic en "Create repository"

3. ¿Cómo crear una rama en Git?

Usar el comando:

git branch nombre-de-la-rama

4. ¿Cómo cambiar a una rama en Git?

Usar:

git checkout nombre-de-la-rama

O en versiones modernas:

git switch nombre-de-la-rama

5. ¿Cómo fusionar ramas en Git?

Estando en la rama destino (por ejemplo main), ejecutar:

git merge nombre-de-la-rama

6. ¿Cómo crear un commit en Git?

1. Preparar archivos con:

git add archivo.ext

2. Crear el commit:

git commit -m "Mensaje descriptivo"

7. ¿Cómo enviar un commit a GitHub?

Primero asegurarse de tener el repositorio remoto configurado. Luego:

git push origin nombre-de-la-rama

8. ¿Qué es un repositorio remoto?

Es una versión del repositorio almacenada en la nube (como GitHub), que permite compartir y sincronizar cambios entre distintos desarrolladores.

9. ¿Cómo agregar un repositorio remoto a Git?

git remote add origin https://github.com/usuario/repositorio.git

10. ¿Cómo empujar cambios a un repositorio remoto?

git push origin nombre-de-la-rama

11. ¿Cómo tirar de cambios de un repositorio remoto?

git pull origin nombre-de-la-rama

12. ¿Qué es un fork de repositorio?

Es una copia de un repositorio de otro usuario en tu cuenta, que te permite experimentar y proponer cambios sin afectar el original.

13. ¿Cómo crear un fork de un repositorio?

1. Ingresar al repositorio original en GitHub

2. Hacer clic en el botón "Fork"

3. Seleccionar tu cuenta de destino

14. ¿Cómo enviar una solicitud de extracción (pull request)?

1. Subí tus cambios a tu fork

2. Ingresá al repositorio original

3. Hacé clic en "Compare & pull request"

4. Escribí un mensaje y confirmá el envío

15. ¿Cómo aceptar una solicitud de extracción?

Desde el repositorio original en GitHub, revisar la solicitud, comentar si es necesario, y hacer clic en "Merge pull request".

16. ¿Qué es una etiqueta en Git?

Es una referencia que apunta a un commit específico, generalmente usada para marcar versiones o hitos del proyecto.

17. ¿Cómo crear una etiqueta en Git?

git tag -a v1.0 -m "Versión 1.0"

18. ¿Cómo enviar una etiqueta a GitHub?

git push origin v1.0

19. ¿Qué es un historial de Git?

Es el registro de todos los commits realizados en un repositorio.

20. ¿Cómo ver el historial de Git?

git log

21. ¿Cómo buscar en el historial de Git?

git log --grep="texto a buscar"

22. ¿Cómo borrar el historial de Git?

No se recomienda borrar el historial, pero podrías reiniciar el repo:

git checkout --orphan nueva-rama

(Uso avanzado, con precaución)

23. ¿Qué es un repositorio privado en GitHub?

Es un repositorio al que solo acceden los usuarios autorizados, ideal para proyectos cerrados o personales.

24. ¿Cómo crear un repositorio privado en GitHub?

Al crearlo, seleccionar la opción "Private" en lugar de "Public".

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

1. Ir a "Settings" del repositorio

2. Elegir "Manage access"

3. Hacer clic en "Invite a collaborator"

4. Ingresar el nombre de usuario y enviar la invitación

26. ¿Qué es un repositorio público en GitHub?

Es un repositorio visible para cualquier persona en Internet.

27. ¿Cómo crear un repositorio público en GitHub?

Al crearlo, seleccionar la opción "Public"

28. ¿Cómo compartir un repositorio público en GitHub?

Compartiendo la URL del repositorio, por ejemplo:

https://github.com/usuario/repositorio

Segunda Parte - Práctica con Git y GitHub

1. Crear un repositorio en GitHub:

- Ingresar a https://github.com

- Hacer clic en "New"

- Nombrar el repositorio como mi-primer-repo

- Elegir público y agregar un README

- Click en "Create repository"

2. Clonar el repositorio localmente:

git clone https://github.com/tu_usuario/mi-primer-repo.git

cd mi-primer-repo

3. Agregar archivo:

echo "Este es mi archivo de prueba" > mi-archivo.txt

git add .

git commit -m "Agregando mi-archivo.txt"

git push origin main

4. Crear nueva rama:

git checkout -b develop

echo "Archivo desde la rama develop" > archivo-develop.txt

git add .

git commit -m "Archivo agregado en la rama develop"

git push origin develop

Simulación de conflicto:

1. Crear repositorio llamado conflict-exercise en GitHub con README

2. Clonar:

git clone https://github.com/tu_usuario/conflict-exercise.git

cd conflict-exercise

3. Crear y editar rama:

git checkout -b feature-branch

echo "Este es un cambio en la feature branch." >> README.md

git add README.md

git commit -m "Added a line in feature-branch"

4. Volver a main y editar:

git checkout main

echo "Este es un cambio en la main branch." >> README.md

git add README.md

git commit -m "Added a line in main branch"

5. Merge:

git merge feature-branch

6. Resolver conflicto:

Editar README.md, borrar las marcas de conflicto (<<<<<<<, =======, >>>>>>>)

Guardar, luego:

git add README.md

git commit -m "Resolved merge conflict"

7. Push final:

git push origin main

git push origin feature-branch