

Tp 2 Github

Santiago Almiron

Actividad 1)

1. ¿Qué es GitHub?

- GitHub es una plataforma en línea basada en Git que permite a desarrolladores gestionar el control de versiones, colaborar en proyectos y alojar código de manera eficiente. Ofrece herramientas como repositorios públicos y privados, solicitudes de actualización del código (pull requests), seguimiento de problemas (issues) y automatización de tareas.

2. ¿Cómo crear un repositorio en GitHub?

- **Desde la plataforma web:** Inicia sesión en GitHub, haz clic en el ícono "+" > "New repository", ingresa el nombre, selecciona la visibilidad (público o privado) y haz clic en "Create repository".
- **Desde la consola (Git):**
 1. Abre la terminal y navega hasta la carpeta donde deseas crear el repositorio.
 2. Ejecuta `git init` para inicializar un repositorio local.
 3. Agrega archivos con `git add .` y haz un commit con `git commit -m "Primer commit"`.
 4. Conecta el repositorio remoto con `git remote add origin <URL-del-repositorio>`.
 5. Sube los cambios con `git push -u origin main`.

3. ¿Cómo crear una rama en Git?

- Para crear una rama, ejecuta el comando: `git branch <NOMBRE-DE-LA-RAMA>`.

4. ¿Cómo cambiar a una rama en Git?

- Para cambiar de rama, utiliza el comando: `git checkout <NOMBRE-DE-LA-RAMA>`.

5. ¿Cómo fusionar ramas en Git?

- Para fusionar dos ramas, utiliza el comando: `git merge <NOMBRE-DE-LA-RAMA>`.

6. ¿Cómo crear un commit en Git?

- Los commits se crean utilizando el comando: `git commit -m <MENSAJE>`.

7. ¿Cómo enviar un commit a GitHub?

- Después de hacer el commit, empuja los cambios con el comando: `git push origin <NOMBRE-DE-LA-RAMA>`.

8. ¿Qué es un repositorio remoto?

- Un repositorio remoto en Git es una copia de tu proyecto almacenada en un servidor en la nube (por ejemplo, GitHub, GitLab o Bitbucket). A diferencia del repositorio local, que está en tu computadora, el remoto se puede acceder y colaborar en línea.

9. ¿Cómo agregar un repositorio remoto a Git?

- Para agregar un repositorio remoto, utiliza el comando: `git remote add origin <URL-del-repositorio>`.

10. ¿Cómo empujar cambios a un repositorio remoto?

- Para empujar cambios, usa el comando: `git push <URL-O-ALIAS> <NOMBRE-DE-LA-RAMA>`.

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

- Para actualizar tu rama local con los cambios del repositorio remoto, usa el comando: `git pull <URL-O-ALIAS> <NOMBRE-DE-LA-RAMA>`.

12. ¿Qué es un fork de repositorio?

- Un fork es una copia de un repositorio creada en tu propia cuenta de GitHub, lo que te permite hacer cambios sin afectar el proyecto original. Es útil para contribuir a proyectos de código abierto o experimentar con el código de otro sin riesgos. Después de modificar tu fork, puedes enviar un pull request para proponer cambios al repositorio original.

13. ¿Cómo crear un fork de un repositorio?

- Para crear un fork, ingresa al repositorio en GitHub, haz clic en el botón "Fork" en la parte superior derecha, y GitHub creará una copia del repositorio en tu cuenta.

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

- Para enviar una solicitud de extracción, ve a GitHub, selecciona "New pull request", selecciona las ramas a fusionar y describe los cambios que realizaste.

15. ¿Cómo aceptar una solicitud de extracción?

- Aceptar una solicitud de extracción significa integrar los cambios propuestos en la rama destino. Por ejemplo, si tienes `rama1` y `rama2`, y realizas una PR desde `rama1` hacia `rama2`, aceptar la solicitud integrará los cambios de `rama1` en `rama2`.

16. ¿Qué es una etiqueta en Git?

- Una etiqueta (tag) en Git es una referencia que marca puntos específicos en la historia del repositorio, generalmente para indicar versiones importantes o lanzamientos. A diferencia de las ramas, las etiquetas son estáticas y no cambian.

17. ¿Cómo crear una etiqueta en Git?

- Para crear una etiqueta, utiliza el comando: `git tag -a v1.0 -m "Versión 1.0"`.

18. ¿Cómo enviar una etiqueta a GitHub?

- Para enviar una etiqueta específica a GitHub, usa el comando: `git push origin v1.0`. Para enviar todas las etiquetas, usa: `git push origin --tags`.

19. ¿Qué es un historial de Git?

- El historial de Git es el registro de todos los cambios realizados en un repositorio. Permite ver qué modificaciones se hicieron, quién las realizó y en qué momento.

20. ¿Cómo ver el historial de Git?

- Para ver el historial, usa el comando: `git log`.

21. ¿Cómo buscar en el historial de Git?

- **Buscar por palabra clave en los mensajes de commit:** `git log --grep="palabra_clave"`.
- **Buscar cambios en un archivo específico:** `git log -- archivo.txt`.

- **Buscar un texto dentro del contenido de los commits:** `git log -S 'texto_a_buscar'`.

22. ¿Cómo borrar el historial de Git?

- El historial se puede borrar eliminando el archivo `.git` y volviendo a crear el repositorio.

23. ¿Qué es un repositorio privado en GitHub?

- Un repositorio privado es aquel al que solo se puede acceder si el creador o los administradores otorgan permisos. No se puede hacer un fork ni enviar pull requests sin los permisos adecuados.

24. ¿Cómo crear un repositorio privado en GitHub?

- Al crear un nuevo repositorio en GitHub, selecciona la opción "private" para que sea privado.

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

- Desde la configuración del repositorio, puedes agregar colaboradores y otorgarles acceso al repositorio privado.

26. ¿Qué es un repositorio público en GitHub?

- Un repositorio público es accesible para cualquier persona, que puede ver el código, hacer un fork o proponer pull requests sin requerir permisos especiales.

27. ¿Cómo crear un repositorio público en GitHub?

- Al crear un nuevo repositorio en GitHub, selecciona la opción "public" para que sea público.

28. ¿Cómo compartir un repositorio público en GitHub?

- Un repositorio público se puede compartir simplemente compartiendo el enlace de la URL del repositorio.

Actividad 2)

[Actividad 2](#)

Actividad 3)

[Actividad 3](#)