

Francisco López (39.327.419)

TP2: Git & GitHub

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

1. ¿Qué es GitHub?

Es una plataforma que ofrece alojamiento de repositorios. Permite almacenar, gestionar y compartir proyectos de software utilizando Git (un sistema de control de versiones). Esta herramienta es gratuita y de código abierto, lo que permite el trabajo en equipo en proyectos, el intercambio de código y el trabajo conjunto de forma eficiente.

2. ¿Cómo crear un repositorio en GitHub?

Primero debemos tener una cuenta en la página web de **GitHub**. Una vez que la tenemos, podemos configurar nuestro perfil y empezar a crear repositorios. Podemos hacerlo también desde nuestro **Git** clickeando en New, una vez allí colocar el nombre, alguna que otra descripción de lo que vamos a subir, si queremos que sea público o privado, etc.

3. ¿Cómo crear una rama en Git?

Para crear una nueva rama primero debemos verificar en qué rama estamos (utilizando el comando "**git branch**"), luego creamos la rama (utilizando "**git branch nuevaRama**"). Cambiamos a la rama creada ("**git checkout nuevaRama**") y confirmamos que estamos ahí usando nuevamente "**git branch**". Finalmente subimos la rama al repositorio utilizando el comando "**git push -u origin nuevaRama**".

4. ¿Cómo cambiar a una rama en Git?

Para saltar de una rama a otra, tienes que utilizar el comando git checkout: "**\$ git checkout nuevaRama**". Esto mueve el apuntador HEAD a la rama nuevaRama.

Para crear una nueva rama y saltar a ella puedes utilizar el comando git checkout con la opción -b: "**\$ git checkout -b nuevaRama**"

5. ¿Cómo fusionar ramas en Git?

Primero, debes estar en la rama a la que quieres fusionar los cambios. Una vez en la rama de destino, usamos el comando "**git merge**" para fusionar la rama de origen en la rama actual: "**\$ git merge nuevaRama**" Este comando incorpora los cambios de nuevaRama en la principal (master/main).

6. ¿Cómo crear un commit en Git?

Para crear un **commit** en Git, primero debemos verificar el estado de los archivos utilizando el comando **git status** para ver qué archivos han sido agregados, modificados o eliminados. Luego agregamos archivos a través del comando **git add nombreArchivo**.

Creamos el commit utilizando **git commit -m "mensaje descriptivo"**.

Por último subimos el commit al repositorio usando el comando **git push origin nombreArchivo**.

7. ¿Cómo enviar un commit a GitHub?

Lo enviamos a través del comando **git push origin nombreArchivo**. Si estamos trabajando colaborativamente, podemos usar la opción de crear un Pull Request.

8. ¿Qué es un repositorio remoto?

Es una versión del proyecto almacenada en un servidor en línea o en otra computadora. Permite que varias personas colaboren en el mismo código desde diferentes lugares.

9. ¿Cómo agregar un repositorio remoto a Git?

Iniciamos un repositorio local (**git init**), agregamos el repositorio remoto (**git remote add origin linkRepositorio**), verificamos que el repositorio se agregó correctamente (**git remote -v**) y subimos el código (**git add . / git commit -m "primer commit" / git push -u origin main**).

10. ¿Cómo empujar cambios a un repositorio remoto?

Los cambios los vamos empujando usando el comando **git push**

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

Para obtener los últimos cambios del repositorio utilizamos **git pull**

12. ¿Qué es un fork de repositorio?

Es una copia de un repositorio que se crea en tu cuenta personal sin afectar el original. Así podemos experimentar, hacer cambios y colaborar en proyectos sin modificar el repositorio original.

13. ¿Cómo crear un fork de un repositorio?

En la parte superior derecha de la página del repositorio, haz clic en el botón **"Fork"**. Esperamos y se creará una copia de ese repositorio en nuestra cuenta GitHub.

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

Primero nos aseguramos de que se hicieron los cambios en nuestro fork/rama. Vamos al repositorio original en GitHub donde deseamos enviar la Pull Request. Clickeamos "Pull Requests" en la parte superior del repositorio. Y luego apretamos "New Pull Request". Seleccionamos las ramas, revisamos cambios y clickeamos "Create pull request"

15. ¿Cómo aceptar una solicitud de extracción?

Haz clic en el botón verde **"Merge pull request"**. Después de hacer clic en el botón, GitHub te pedirá confirmar la fusión. Podemos dejar el mensaje predeterminado o escribir uno nuevo. Clic en **"Confirm merge"** para fusionar los cambios.

16. ¿Qué es un etiqueta en Git?

Una **etiqueta** en Git es un marcador que se usa para señalar puntos específicos en la historia del repositorio. No cambian y son inmutables. Son útiles para señalar versiones de producción, lanzamientos o cualquier otro evento importante de nuestro proyecto.

17. ¿Cómo crear una etiqueta en Git?

Creamos etiquetas utilizando **git tag nombreEtiqueta**. Puede variar un poco dependiendo el tipo de etiqueta que creemos.

18. ¿Cómo enviar una etiqueta a GitHub?

Una vez creada, utilizamos **git push origin nombreEtiqueta** para enviarla

19. ¿Qué es un historial de Git?

Es el registro de todos los cambios que se han realizado en un repositorio, incluyendo los commits, las ramas y las etiquetas. Git almacena la historia completa de tu proyecto, lo que te permite revisar y gestionar todas las versiones anteriores de nuestro código.

20. ¿Cómo ver el historial de Git?

Vemos el historial utilizando comandos como **git log**.

21. ¿Cómo buscar en el historial de Git?

Para buscar en el historial podemos utilizar comandos de git log para hacer búsquedas detalladas dentro del repositorio (**git log --grep/--author/<archivo>**)

22. ¿Cómo borrar el historial de Git?

Podemos utilizar **git reset**.

23. ¿Qué es un repositorio privado en GitHub?

Es un repositorio con acceso restringido, sólo las personas que autoricemos pueden ver, clonar, contribuir o modificar el contenido de ese repositorio.

24. ¿Cómo crear un repositorio privado en GitHub?

De la misma forma que para crear uno público, seguimos los pasos y marcamos que queremos que sea privado. Entramos a GitHub, clickeamos New y seleccionamos Private.

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

Para invitar a alguien a colaborar con nuestro repositorio privado vamos a las Configuraciones del repositorio, Gestionar acceso, Invitar colaborador/Buscar colaborador, Seleccionamos y enviamos la invitación. Luego podemos modificar los permisos para cada usuario que invitemos.

26. ¿Qué es un repositorio público en GitHub?

Es un repositorio cuyo contenido es accesible para todo el mundo. Pueden verlo, clonarlo, bifurcarlo (fork) y contribuir al repositorio dependiendo los permisos

27. ¿Cómo crear un repositorio público en GitHub?

Iniciamos sesión en GitHub / Hacemos clic en el botón "+" y seleccionamos "New repository" / Completamos el formulario con el nombre y descripción del repositorio / Seleccionamos la opción "Public" para hacerlo público. Clic en "Create repository".

28. ¿Cómo compartir un repositorio público en GitHub?

Al ser público, cualquier persona puede acceder. Tenemos que copiar el link (en Code) y enviamos esa URL del repositorio a través de cualquier medio.

2) Repositorio Act 2

<https://github.com/franciscolodev/ProgramacionIUTN>

3) Repositorio Act 3

<https://github.com/franciscolodev/conflict-exercise>