UTN
Programación 1

TP-02

(Programación 1)

Actividad 1

1) ¿Qué es GitHub?

GitHub es una plataforma basada en la nube donde puedes almacenar, compartir y trabajar junto con otros usuarios para escribir código.

Almacenar tu código en un "repositorio" en GitHub te permite lo siguiente:

- **Presentar o compartir** el trabajo.
- **Seguir y administrar** los cambios en el código a lo largo del tiempo.
- Dejar que otros usuarios **revisen** el código y realicen sugerencias para mejorarlo.
- **Colaborar** en un proyecto compartido, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores antes de que esté listo para integrarlos.

El trabajo colaborativo, una de las características fundamentales de GitHub, es posible gracias al software de código abierto Git, en el que se basa GitHub.

2) ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub necesitamos tener un cliente de GIT (controlador de versiones) instalado en nuestra computadora.

Desde Github debemos crear el repositorio desde la sección “Repositories” con el botón “New” (new repository), allí indicar un nombre para el mismo, descripción, su tipo de accesibilidad (privado o público) y algunas otras configuraciones predeterminadas que ofrece la plataforma respecto del repositorio y ya tendríamos nuestro repositorio remoto creado . Una vez hecho esto podemos, desde la consola bash incluida en la instalación de GIT o de la terminal que tengamos en nuestro equipo (cmd, powershell, etc.), ejecutar el comando (estando ubicados en el directorio donde queremos clonar el repositorio remoto) “`git clone`

`URL_DE_TU_REPOSITORIO`” donde debemos colocar el url que designa GitHub al repositorio anteriormente creado. También podríamos por ejemplo inicializar un repositorio local y luego sincronizarlo con el creado en GitHub.

3) ¿Cómo crear una rama en Git?

Para crear una rama en GIT debemos ubicarnos con la consola en el directorio de nuestro repositorio y ejecutar el comando “git branch NUEVA_RAMA”.

4) ¿Cómo cambiar a una rama en Git?

Para cambiar de rama en GIT utilizamos el comando “git checkout NOMBRE_RAMA”. Para ver a qué ramas podemos cambiar podemos ejecutar simplemente “git branch” comando que nos retorna una lista con las ramas del repositorio sobre el cual estamos parados.

5) ¿Cómo fusionar ramas en Git?

Para fusionar ramas en GIT nos ubicamos con el comando anterior (git checkout) en una de las dos ramas que queramos fusionar y ejecutamos “git merge RAMA_2_A_FUSIONAR”.

6) ¿Cómo crear un commit en Git?

Para crear un commit en Git debemos “stagear” los cambios realizados primero con “git add ARCHIVO_A_STAGEAR” para stagear un archivo en particular o “git add .” para stagear todo archivo modificado y luego ejecutamos “git commit” para realizar el commit en sí. (A este último comando podemos agregar opcionalmente “-m NOMBRE_COMMIT” para dejar una descripción o nombre del commit.)

7) ¿Cómo enviar un commit a GitHub?

Para enviar un commit a nuestro repositorio remoto debemos ejecutar el comando “git push” con el cual se hará una copia exacta de nuestros cambios/ediciones de nuestro repositorio local al “repo” remoto.

8) ¿Qué es un repositorio remoto?

Un repositorio remoto es una copia exacta de un repositorio local pero en la nube. Básicamente una copia de algún directorio de nuestra pc en internet.

9) ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git ejecutamos el comando “git clone URL_REPO_REMOTO”.

10) ¿Cómo empujar cambios a un repositorio remoto?

Para “empujar” cambios a un repositorio remoto debemos:

- git add .
- git commit -m “Nuevo commit”
- git push

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

Tiramos de los cambios de un repositorio remoto con el comando “git pull”

En este comando podemos especificar tanto el repositorio remoto como la rama desde los cuales queremos hacer el “pull” por ejemplo: “git pull origin main” donde “origin” es el nombre del repositorio remoto y main es la rama de la cual vamos a tirar.

12) ¿Qué es un fork de repositorio?

Con el fork lo que hacemos es “copiar” un repositorio ajeno a nuestro usuario de GitHub. Básicamente es crear un repositorio propio usando de base un repositorio ajeno.

13) ¿Cómo crear un fork de un repositorio?

Para crear un fork de un repositorio debemos ir a la página principal del repositorio del cual queremos hacer el fork y clickear el botón homónimo “Fork”. Una vez clickeado nos llevará a otra página donde podemos elegir que nombre queremos darle al repositorio, una descripción y si queremos “forkear” solo la rama “master”.

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

Desde la pestaña “Pull requests” del repositorio que queramos hacer un pull request donde comparamos la rama en la que estuvimos trabajando y la rama main del repositorio original, explicamos nuestros cambios etc.

15) ¿Qué es un etiqueta en Git?

Una **etiqueta (tag)** en Git es un marcador estático que se utiliza para señalar un punto específico en el historial de commits, generalmente asociado a versiones estables de un proyecto (como `v1.0.0`).

16) ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta, ejecuta `git tag -a v1.0.0 -m "Versión 1.0 estable"` desde la rama deseada, donde `-a` indica una etiqueta anotada (con mensaje) y `-m` agrega una descripción.

17) ¿Cómo enviar una etiqueta a GitHub?

Para enviar la etiqueta a github usamos `git push origin v1.0.0`. Las etiquetas son útiles para gestionar releases, facilitando el acceso a versiones específicas sin depender de hashes de commits.

18) ¿Qué es un historial de Git?

Es un registro de los cambios que se hicieron en un repositorio. Este registro no es mas que una lista con cada commit realizado sobre el repositorio.

19) ¿Cómo ver el historial de Git?

Podemos ver el historial de Git con el comando “git log”.

20) ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, usamos `git log` con filtros como `--grep="texto"` para buscar en mensajes de commit, `-S"palabra"` para encontrar cambios de código específicos, o `--author="nombre"` para filtrar por autor. Si necesitas ver quién modificó una línea en particular, `git blame archivo.txt` te muestra el commit y autor de cada cambio. Para buscar en un rango de fechas, añade `--since="2024-01-01"`. Por ejemplo, `git log --grep="fix login" -p` buscará commits con "fix login" en su mensaje y mostrará los cambios realizados.

21) ¿Cómo borrar el historial de Git?

Para borrar el historial de Git debemos ejecutar el comando `git checkout --orphan nueva_rama` para crear una rama sin historial. Luego añadimos los archivos con `git add -A` y hacemos un commit inicial usando `git commit -m "Primer commit"`. Después eliminamos la rama principal con `git branch -D main`, renombramos nuestra nueva rama a main con `git branch -m main` y finalmente forzamos el push al repositorio remoto usando `git push -f origin main`. Esto dejará solo el último commit, eliminando todo el historial anterior del repositorio.

22) ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es un repositorio al cual tendrán acceso (clonar, modificar, etc.) solo aquellos usuarios que especifiquemos.

23) ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado debemos seguir los pasos especificados en la pregunta número 2 y en la configuración de accesibilidad seleccionar “Privado”.

24) ¿Cómo invitar a alguien a un repositorio privado en GitHub?

Para invitar a alguien a un repositorio privado debemos entrar en la configuración del repositorio y seleccionarlos como “colaboradores” del mismo.

25) ¿Qué es un repositorio público en GitHub?

Un repositorio público es un repositorio de, valga la redundancia, acceso público. Esto implica que cualquier usuario puede ver y clonar el repositorio, sin embargo no todos pueden modificar el mismo.

26) ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público debemos seguir los pasos especificados en la pregunta número 2 y en la configuración de accesibilidad seleccionar “Publico”.

27) ¿Cómo compartir un repositorio público en GitHub?
Para esto podemos compartir directamente el enlace del repositorio, por ejemplo:

https://github.com/tu-usuario/tu-repositorio-publico

Actividad 2

https://github.com/rober-mndz/repositorio-TP-02-ProgI

Actividad 3

https://github.com/rober-mndz/conflict-exercise

Repositorio general de TP’s:
https://github.com/rober-mndz/UTN-TUPaD-P1