



Tecnicatura universitaria en programación a distancia

Materia: Programación 1

Alumno: Matías Facundo Herrera

Trabajo práctico N°2, Trabajo colaborativo: Git y GitHub

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

- ¿Qué es GitHub?

Es una página web que permite subir tus repositorios locales utilizando el software para control de versiones git. En esta plataforma podemos compartir nuestros proyectos con el mundo y contribuir en proyectos de otras personas.

- ¿Cómo crear un repositorio en GitHub?

Primero se debe tener un cuenta en GitHub. Si no la tienes, la plataforma te pedirá crear una más otras verificaciones necesarias. Una vez creada debes ir a la sección de repositorios y clickear en la opción “New”. Se te pedirá que nombres el repositorio y una vez creado, la plataforma te enseñara en una mini guía como subir tus archivos local a tu nuevo repositorio remoto.

- ¿Cómo crear una rama en Git?

En cualquier rama que estés ubicado. Ingresa el comando “git branch” más el nombre de la nueva rama en la consola dentro de tu repositorio local iniciado anteriormente.

- ¿Cómo cambiar a una rama en Git?

Para cambiar de una rama a otra simplemente ingresa el comando “git checkout” más el nombre de la rama en la cual te quieres ubicar.

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas en git se utiliza el comando “git merge” más el nombre de la rama con el cual quieres fusionar la rama donde actualmente estás ubicado. Ejemplo: Estoy en la rama main y al hacer un merge con “branch example” estoy haciendo la fusión desde main, trayendo los cambios de “branch example” a mi rama main o rama principal.

- ¿Cómo crear un commit en Git?

Antes de crear una “foto” de nuestros archivos realizando un commit, debemos tener los archivos preparados en el stage. Ingresamos el comando “git add” seguido de los archivos que queremos agregar al stage. Si queremos agregar todos nuestros cambios de nuestro repositorio local al stage, simplemente colocamos un punto (.)

Una vez subido lo que queremos al “stage”, usamos el comando “git commit” para tener todo actualizado en nuestro repositorio local. Luego podremos subirlo a nuestro repositorio remoto.

- ¿Cómo enviar un commit a GitHub?

Para enviar un commit realizamos un push hacia nuestro repositorio remoto. Ingresamos el comando “git push origin [rama a enviar]”. La variable “rama a enviar”, es la rama que queremos enviar con nuestros cambios a nuestro repositorio remoto. No siempre necesariamente queremos subir nuestra rama principal o main.

- ¿Qué es un repositorio remoto?

Es un espacio de almacenamiento que GitHub y otras plataformas nos ofrece para tener un seguimiento de nuestros archivos de manera remota. En donde enviamos nuestros cambios de manera local desde nuestro dispositivo.

- ¿Cómo agregar un repositorio remoto a Git?

Debemos ingresar el comando “git remote add origin [link repositorio remoto]”. El link del repositorio es brindado por GitHub cuando creamos un nuevo repositorio vacío.

- ¿Cómo empujar cambios a un repositorio remoto?

Como se explico anteriormente, se debe utilizar el comando “git push”

- ¿Cómo tirar de cambios de un repositorio remoto?

- ¿Qué es un fork de repositorio?

Es una copia de otro repositorio en cual realizamos directamente de forma remota. Es decir, clonamos un repositorio de otro usuario a nuestra cuenta de GitHub.

- ¿Cómo crear un fork de un repositorio?

Debemos ubicarnos en un repositorio ajeno. Por ejemplo: Estamos en el repositorio de Jaimito porque queremos agregar mejoras a su código. Pinchamos en el botón “fork”, GitHub nos pedirá una configuración rápida y luego la clonara a nuestra cuenta. Ya contamos con el repositorio de Jaimito en nuestra cuenta.

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

Un pull request se realiza a un repositorio al cual le hemos realizado un fork con anterioridad. Una vez que tengamos nuestros cambios realizados en nuestra copia de GitHub, clickeamos en el botón que nos permite realizar un pull request. Luego debemos esperar si nuestros cambios son extraídos o rechazados por el autor original del repositorio.

- ¿Cómo aceptar una solicitud de extracción?

Del lado del creador original del repositorio, tendremos una sección en la cual podremos revisar los pull request. Desde esa sección de la plataforma podremos aceptar, rechazar y agregar sugerencias de cambios a las modificaciones propuestas por otros usuarios.

- ¿Qué es un etiqueta en Git?

Son una serie de números que siguen las reglas del versionado semántico. Se utilizan para indicar nuevas versiones con cambios mayores, menores y parches de arreglo en un software.

- ¿Cómo crear una etiqueta en Git?

Se utiliza el comando git tag en la carpeta de nuestro repositorio local. Podremos colocar un mensaje e indicar que números queremos colocar según nuestras necesidades y versión de nuestro software.

- ¿Cómo enviar una etiqueta a GitHub?

Los enviamos con el comando `git push --tags`.

- ¿Qué es un historial de Git?

Es el historial de los commits realizados en un repositorio.

- ¿Cómo ver el historial de Git?

Para ver todos los commits realizados en el historial del repositorio, ejecutamos el comando `git log` en el mismo. Se nos mostrará el orden cronológico de commits en orden inverso

- ¿Cómo buscar en el historial de Git?

Puedes buscar los commits que necesites por ejemplo mediante la búsqueda de palabra clave en los comentarios de los commit. Esto se hace con el comando `git log --grep="palabra"`.

Nótese que debe agregarse la "palabra clave" o flag `--grep`.

- ¿Cómo borrar el historial de Git?

Esto podemos hacerlo utilizando el comando `git rebase`.

Un ejemplo sería escribir `git rebase -i HEAD~5`. Esto nos mostrará los últimos 5 commits de nuestro repositorio en un menú interactivo. Debemos anotar con la etiqueta "drop" los commits que querramos eliminar. Una vez salimos del menú ingresamos el comando `git push --force` para guardar los cambios.

- ¿Qué es un repositorio privado en GitHub?

Es un repositorio el cual sólo puede ser visto por su creador y por los invitados. No es visible para el público ni usuarios no autorizados.

- ¿Cómo crear un repositorio privado en GitHub?

Al crear un nuevo repositorio en GitHub debemos marcarlo como privado.

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

Debemos clicar en el botón de agregar repositorios simplemente.

- ¿Qué es un repositorio público en GitHub?

Es un repositorio que es abierto al público y no se necesita ningún tipo de permiso para verlo. Cualquier persona en internet puede clonarlo, realizar un fork y pull request.

- ¿Cómo crear un repositorio público en GitHub?

Al crear un nuevo repositorio, GitHub nos muestra la opción de hacerlo público o privado.

- ¿Cómo compartir un repositorio público en GitHub?

Podemos compartirlo simplemente copiando el URL de nuestro repositorio o invitando colaboradores. Nótese que esto podemos realizarlo sin que necesariamente sea un repositorio de carácter privado.

2)

Link del ejercicio 2 finalizado en GitHub: <https://github.com/mati7122/git-github-tp-utn/tree/main>

3)

Link del ejercicio 3 finalizado en GitHub: <https://github.com/mati7122/conflict-exercise>