TP 2: Git y Git Hub

Alumno: Ferreyra Matias

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

   • *¿Qué es GitHub?*

   GitHub es una plataforma en la nube que nos permite almacenar y compartir código, y trabajar junto con otros usuarios.

   Nos permite hacer un seguimiento de los cambios en el código, colaborar en un proyecto ajeno sin afectar al mismo.

   También nos permite dar acceso a otras personas a nuestro código para que puedan hacer sugerencias, etc.

   • *¿Cómo crear un repositorio en GitHub?*

   Una vez creado el repositorio en Git, debemos abrir la cuenta en GitHub. Allí vamos a "Nuevo repositorio". Luego le damos un nombre al repositorio y damos al botón "Crear repositorio".

   Hasta acá el repositorio estará creado, pero vacío.

   Luego seguimos las instrucciones de GitHub para hacer la conexión con nuestro repositorio local.

   • *¿Cómo crear una rama en Git?*

   Para crear una nueva rama en Git debemos escribir en la consola **git branch + nombre_rama** . Por ejemplo: **git branch rama2**

   • *¿Cómo cambiar a una rama en Git?*

   Para cambiar de rama podemos utilizar la instrucción **git checkout + nombre_rama**. Por ejemplo: **git checkout rama2**

   • *¿Cómo fusionar ramas en Git?*

   Las ramas se pueden fusionar a través del comando **git merge**. Debemos tener en cuenta la rama sobre la que estoy posicionado. Por ejemplo si estoy posicionado en la rama master y ejecuto un **git merge rama2**, el contenido de la rama2 se volcará sobre la rama master.

   • *¿Cómo crear un commit en Git?*

   Para crear un commit debemos ejecutar la instrucción **git commit -m "Descripcion"**.

   "**-m**" nos da la posibilidad de añadir una descripción al commit.

• *¿Cómo enviar un commit a GitHub?*

Para enviar un commit a GitHub lo podemos hacer con la instrucción **git push -u origin main** (o master, según corresponda). Esto si es la primera vez, sino basta con **git push**.

• *¿Qué es un repositorio remoto?*

Es un repositorio que se encuentra alojado en la nube. Por lo tanto podemos acceder a el desde cualquier ubicación siempre que tengamos una conexión a Internet.

• *¿Cómo agregar un repositorio remoto a Git?*

Podemos agregarlo a través de la instrucción **git remote add origin + url del repositorio**. Por ej: **git remote add origin https://github.com/matiacade55/proyecto-inicial-2.git**

• *¿Cómo empujar cambios a un repositorio remoto?*

Si es la primera vez con el comando **git push -u origin master**. Luego de la primera vez se hace con **git push**.

• *¿Cómo tirar de cambios de un repositorio remoto?*

Se hace a través del comando **git pull origin master** o **git pull** si usamos -u en el push.

Tambien se puede hacer con **git fetch** si deseamos obtener los cambios del repositorio remoto pero no aplicarlos.

• *¿Qué es un fork de repositorio?*

Un Fork es una bifurcación. Nos permite crear una copia exacta del código de otro desarrollador, en nuestro repositorio en la nube, para trabajarlo en nuestra cuenta de manera independiente, por lo que los cambios que hagamos solo se verán reflejados en nuestra copia.

• *¿Cómo crear un fork de un repositorio?*

Para crear un Fork, se hace desde el repositorio donde se encuentra el proyecto a copiar, presionando el botor con el icono de bifurcación.

Podemos ponerle una descripción propia y configurar algunas opciones más.

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

Para hacer un pull request primero debo hacer un Fork del proyecto al que quiero contribuir.

Luego debo clonar ese Fork en mi computadora para trabajarlo de forma local.

Luego debo hacer un **add .** y un **commit** para guardar los cambios.

Por último un **push** para enviar esos cambios a mi fork en el repositorio en la nube.

Ahora que tengo los cambios en mi fork, debo solicitar la incorporación de los cambios al “dueño” del código.

Para esto estando en la pagina de mi repositorio bifurcado presiono el botón **pull request** o "**new pull request**".

Luego selecciono el repositorio que quiero modificar (base o head).

Antes de confirmar revisar que los cambios sugeridos sean los que deseo hacer.

Luego presionar el botón "**Crear pull request**". Le ponemos un título y una descripción.

• *¿Cómo aceptar una solicitud de extracción?*

Luego de revisar los cambios sugeridos podemos sugerir cambios o confirmar el merge (la fusión) de los cambios en nuestro repositorio. Esto ultimo se hace presionando el botón verde "**Merge pull request**".

• *¿Qué es una etiqueta en Git?*

Las etiquetas son referencias que apuntan a puntos concretos en el historial de Git.

Existen etiquetas ligeras y anotadas.

• *¿Cómo crear una etiqueta en Git?*

Una etiqueta en Git se crea con el comando **git tag + nombre_etiqueta**. **Por ej: git tag v1.0**

• *¿Cómo enviar una etiqueta a GitHub?*

Para enviar una etiqueta a GitHub se debe hacer de forma explícita, es decir indicando su nombre, a través del comando **git push origin nombre_etiqueta**. Por ej: **git push origin v1.0**.

• *¿Qué es un historial de Git?*

El historial de Git es el registro de todos los cambios que se hicieron organizado por cada commit que se hizo. Nos permite ver que cambios se realizaron, quien los hizo y cuando.

• *¿Cómo ver el historial de Git?*

El historial de Git se puede ver con el comando git log.

También se puede usar **git log -p** para verlo de forma más detallada,

o **git log –oneline** para una vista más compacta, entre otras opciones.

• *¿Cómo buscar en el historial de Git?*

Se pueden realizar búsquedas con distintos comandos dependiendo lo que estemos buscando:

git log -2 : muestra los últimos dos commits.

git log –author : para buscar por autor.

Git log –grep = "palabra clave": Para buscar por palabra contenida en el mensaje del commit.

Tambien se puede buscar por fechas, por palabra especifica dentro del código modificado, etc.

• *¿Cómo borrar el historial de Git?*

Para deshacer cosas se utiliza el comando **git reset**. El mismo mueve el puntero de HEAD y la rama actual a un commit anterior.

Puede afectar el historial de commits, el área de preparación (staging área) y el directorio de trabajo.

git reset --soft HEAD~1 : deshace el ultimo commit pero mantiene los archivos en el área de staging

git reset --mixed HEAD~1 : elimina el último commit y también deshace los cambios del área de staging (git add), pero los archivos siguen intactos en tu directorio de trabajo**.**

Git reset --hard HEAD~1 : borra el ultimo commit y también elimina los cambios en los archivos.

• *¿Qué es un repositorio privado en GitHub?*

Es un repositorio al que solo puede acceder el creador y las personas que el mismo autorice.

• *¿Cómo crear un repositorio privado en GitHub?*

Para crear un repositorio privado se debe elegir la opción “Privado” al seleccionar la visibilidad del repositorio.

• *¿Cómo invitar a alguien a un repositorio privado en GitHub?*

Para invitar a alguien a un repositorio privado se hace desde la configuración del mismo. Ingresando a la opción “Colaboradores”. Luego seleccionamos “Agregar Personas”. En el cuadro de búsqueda escribimos el nombre del usuario a invitar y damos clic en “Agregar Nombre al repositorio”.

A esta persona le llegará un mail y cuando confirme la invitación ya tendrá acceso a nuestro repositorio.

• *¿Qué es un repositorio público en GitHub?*

Un repositorio público es el que es accesible para cualquier persona.

• *¿Cómo crear un repositorio público en GitHub?*

Para crear un repositorio público se debe elegir la opción “Público” al seleccionar la visibilidad del repositorio.

• *¿Cómo compartir un repositorio público en GitHub?*

Para compartir un repositorio publico basta con compartir el link o dirección url del repositorio.

2. Link del repositorio del ejercicio 2: https://github.com/matiacade55/tp-2-git-y-github.git
3. Link del repositorio del ejercicio 3: https://github.com/matiacade55/conflict-exercise.git