

Trabajo Práctico N.º 2

Git y GitHub

Alumno: Bonanno Nicolas

Ejercicios:

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

- ¿Qué es GitHub?

Es una plataforma de desarrollo colaborativo , basada en la nube que permite a desarrolladores gestionar proyectos de software utilizando el sistema de control Git. Pueden almacenar y compartir código en repositorios, contribuir en proyectos de código abierto y ayudar a otros programadores.

- ¿Cómo crear un repositorio en GitHub?

Debemos ingresar a GitHub e iniciar sesión, una vez dentro en la esquina superior derecha encontramos el icono “+”, le damos click y elegimos la opción de “New Repository”. Una vez dentro rellenamos los campos:

- Repository name

- Description (opcional)

- Elegimos la opción de un repositorio visible o privado

Luego le damos a “Create repository”.

- ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama de git, en nuestra terminal antes que nada ejecutamos (git branch) para poder ver todas las ramas y saber en cual rama estamos ubicados. Ahora que sabemos, ejecutamos “git checkout (nombre de la rama)” para desplazarnos a dicha rama.

- ¿Cómo fusionar ramas en Git?

A la hora de fusionar debemos estar situados en la rama que queremos que tome el rol principal en dicha fusión. Nos ubicamos utilizando los comandos “git branch y git checkout”. Acto seguido fusionamos las ramas mediante “git merge (nombre de la rama a fusionar)”

- ¿Cómo crear un commit en Git?

Primero que nada nos aseguramos estar dentro del repositorio indicado y creamos el commit con el comando “git commit -m “ , donde creamos y guardamos el commit con un mensaje descriptivo. Si hemos realizado modificaciones en el código, previo al commit debemos añadir los archivos modificados mediante “git add .”

- ¿Cómo enviar un commit a GitHub?

Nos aseguramos de seguir los pasos de la pregunta anterior, y luego mediante el comando "git push origin main" enviaremos el commit a GitHub, si deseamos enviar otra rama que no sea la principal, en vez de main escribiremos la rama que queremos enviar.

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu repositorio que está alojada en un servidor en la nube o en otra máquina como por ej. GitHub. Las características principales son que permite colaborar con otros desarrolladores, funciona como copia de seguridad y también se sincroniza con el repositorio local mediante comando como git push (para subir cambios) y git pull (para descargar cambios).

- ¿Cómo agregar un repositorio remoto a Git?

Mediante el comando "git remote add origin url". Por ejemplo teniendo un repositorio remoto de GitHub, vamos a nuestra terminal y nos dirigimos a nuestro repositorio local, nos fijamos que estemos situado en la carpeta del mismo y mediante "git remote add origin <https://github.com/tu-usuario/tu-repositorio.git>" donde sustituiremos el nombre de usuario y el repositorio de github. Para verificar si este se agregó correctamente, utilizamos el comando "git remote -v"

- ¿Cómo empujar cambios a un repositorio remoto?

Utilizando el comando "git push origin main", si estas en otra rama cambia el main por la rama que deseas utilizar.

- ¿Cómo tirar de cambios de un repositorio remoto?

Para traer los cambios de la rama principal utilizamos el comando "git pull main", en caso de utilizar otra rama como principal cambiamos el nombre de main. Luego revisamos si hay posibles diferencias entre el archivo local y el remoto, en caso de haber conflictos los resolvemos y acto seguido los volvemos a agregar mediante "git add ." y "git commit -m", por últimos los cargamos al remoto con "git push origin main".

- ¿Qué es un fork de repositorio?

Es una copia de otro repositorio en tu cuenta de GitHub, puedes modificar proyectos sin afectar el original y también contribuir con mejoras.

- ¿Cómo crear un fork de un repositorio?

Te diriges al repositorio que quieres "forkear", arriba a la derecha esta la palabra "fork" en donde haciendo click GitHub creará una copia del repositorio en tu cuenta.

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

Dentro del repositorio hay una opción para contribuir y dentro la solicitud de "pull request", le damos click y nos abre una ventana donde colocaremos un título y una descripción explicando los cambios que deseamos realizar sobre el repositorio original, luego le damos a "Create pull request". Solo queda esperar a que el dueño del repositorio revise la solicitud y ver si la aprueba.

- ¿Cómo aceptar una solicitud de extracción?

Abrimos GitHub, nos dirigimos a nuestro repositorio y luego en la pestaña de "Pull request", nos aparecerán las solicitudes que deseamos revisar. Si está todo bien aceptamos los cambios mediante "Review changes" → "Approve" → "Submit review".

- ¿Qué es un etiqueta en Git?

Una etiqueta (tag) en Git es una referencia que marca un punto específico en el historial de commits

- ¿Cómo crear una etiqueta en Git?

Mediante el comando “git tag (nombre de la etiqueta)”, también podemos crear una etiqueta con mensaje mediante “git tag (etiqueta) -m (mensaje)”

- ¿Cómo enviar una etiqueta a GitHub?

Subiremos una etiqueta mediante el comando “git push origin (etiqueta)”, de la misma manera que las subimos también podemos eliminarlas mediante “git push origin --delete (etiqueta)”

- ¿Qué es un historial de Git?

El historial de Git es el registro de todos los cambios realizados en un repositorio, organizados en una línea de tiempo de commits. Es útil para rastrear cambios, deshacer errores y trabajar en equipo.

- ¿Cómo ver el historial de Git?

Mediante “git log” veremos el historial de commits con su ID(hash), autor, fecha y mensaje. También mediante “git --online” podemos ver un historial más resumido solo con ID y el mensaje de commit.

- ¿Cómo buscar en el historial de Git?

En Git podemos buscar commits específicos en el historial utilizando filtros.

git log --grep

git log --author

git log --since

git log --oneline --grep

- ¿Cómo borrar el historial de Git?

Si deseamos eliminar el historial de Git lo podemos hacer mediante “rm -rf .git” esto borrará completamente el historial de Git.

- ¿Qué es un repositorio privado en GitHub?

Esto va a ser un repositorio en donde tendrás privacidad de tus proyectos y solo tú puedas dar acceso a otra gente para poder verlo.

- ¿Cómo crear un repositorio privado en GitHub?

Nos dirigimos a crear un nuevo repositorio normal y corriente, luego de colocarle nombre y alguna descripción debajo nos da la opción de poder hacerlo privado. Con esto podemos ser selectivos a la hora de quienes queremos que vea nuestro repositorio.

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

En GitHub nos dirigimos a la parte superior derecha de la pantalla donde está nuestro perfil y vamos al apartado de ajuste, luego a repositorios y estando allí elegimos el repositorio

privado y clickeamos en donde dice “colaboradores”, al entrar tendremos un lugar en donde nos dejara agregar personas, buscamos al usuario para enviarle la solicitud.

- ¿Qué es un repositorio público en GitHub?

Es un repositorio publico accesible para todos en internet. En donde la gente podra contribuir al proyecto y enviar solicitudes de extracción que juegos deberan ser aceptadas por el dueño del repositorio.

- ¿Cómo crear un repositorio público en GitHub?

Al igual que hicimos antes con el repositorio privado, este de inicio se crea igual como cualquier repositorio, con el cambio de que tenemos que darle a la opción de “publico” , para que este mismo sea accesible para todos en internet.

- ¿Cómo compartir un repositorio público en GitHub?

Podemos compartir el repositorio mediante una URL o tambien compartiendo a colaboradores como con la opción del privado, aqui da lo mismo de si es repositorio sea publico porque el usuario tendrá acceso igual.

2) Realizar la siguiente actividad:

URL : <https://github.com/niicobona/primer-repositorio.git>

3) Realizar la siguiente actividad:

URL : <https://github.com/niicobona/ejercicio-conflictos.git>