

Programacion I
Trabajo Práctico N.º 2: Git y GitHub

Actividad 1:

- **¿Qué es GitHub?**

Es una plataforma online que nos permite alojar repositorios.

- **¿Cómo crear un repositorio en GitHub?**

Teniendo una cuenta de GitHub, basta con ir al símbolo más en la barra superior y seleccionar nuevo repositorio. Luego vamos a asignarle un nombre y podemos seleccionar si queremos que sea público o privado. Por último darle al botón de crear repositorio.

- **¿Cómo crear una rama en Git?**

Para crear una rama en Git podemos usar el comando: "git branch nombre_rama".

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

Una vez que hemos creado esta rama nueva, si quisiéramos cambiar a esta, usamos el comando: "git checkout nombre_rama".

- **¿Cómo fusionar ramas en Git?**

Nos posicionamos en la rama hacia la cual vamos a dejar los cambios conjuntos guardados, utilizando el comando git checkout para llegar a esa rama. Desde aquí usamos el comando "git merge " + el nombre de la rama que queremos fusionar.

- **¿Cómo crear un commit en Git?**

Debemos hacer 2 pasos, el primero en el cual preparamos los cambios para ser guardados mediante el comando "git add .". Y luego hacemos el commit propiamente dicho con el comando "git commit -m "comentarios" ". Donde podemos escribir entre las comillas un comentario que describa el commit que estamos haciendo para tener una referencia mas adelante.

- **¿Cómo enviar un commit a GitHub?**

Para enviar un commit a GitHub hacemos algo conocido como un push. Para esto necesitamos tener nuestro repositorio ya creado en GitHub, y asociado a nuestro repositorio local. Una vez hecho esto podemos desde nuestro repositorio local usar el comando: "git push -u origin main", donde "main" debe ser reemplazado por el nombre de la rama donde esta el commit que deseamos enviar a GitHub.

- **¿Qué es un repositorio remoto?**

Es el repositorio que hemos guardado en una plataforma online, como ser GitHub. Este se encuentra guardado en los servidores de este servicio a diferencia de un repositorio local que guardamos en nuestra computadora.

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

Lo que vamos a hacer en este caso se conoce como un “clon”, que es una copia exacta de un repositorio desde GitHub a nuestro ambito local. Para esto vamos a ir a GitHub y buscar el repositorio deseado, alli podremos encontrar la url que vamos a copiar, y desde nuestra terminal usaremos el comando “git clon” + la url del repositorio deseado para crear una copia en nuestro ambito local.

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

Desde la terminal debemos primero preparar los cambios con “git add .”, luego hacemos el commit con “git commit -m “comentario que describe los cambios”, y por ultimo hacemos el envio con el comando “git push -u origin main”, siendo main el nombre de la rama que deseamos enviar.

Si es la primera vez que lo hacemos quizás debamos enlazar el repositorio remoto al local, copiando el url del repositorio remoto, y pegandolo en nuestra terminal mediante el comando “git remote add origin URL...”.

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

Es un proceso parecido al push, donde nos posicionamos sobre la rama local que queremos que reciba los cambios, y desde allí usamos la línea “git pull”, alternatively podemos agregarle el nombre de la rama que queremos traer (pull) desde el repositorio remoto. El comando quedaria asi: “git pull origin nombre_de_la_rama”

- **¿Qué es un fork de repositorio?**

Un fork de repositorio es una copia exacta de un repositorio que tomamos de otro usuario de GitHub. Esta copia se aloja en nuestra cuenta de GitHub y los cambios que hagamos sobre el se mantienen independientes, no afectan al repositorio local del cual lo hemos copiado.

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

Basta con tener una cuenta en GitHub, buscar el repositorio que nos interesa, y hacer click en el botón fork. La copia de este repositorio se cargara en nuestro perfil.

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

Primero debemos haber hecho el fork de un repositorio al cual queramos contribuir. Una vez tengamos una copia nuestra, hacemos una copia del url de nuestra copia, para poder hacer un “clone”, y obtener así una copia local para trabajar con este repositorio de manera local.

Una vez tengamos la copia local (el clon), y hagamos los cambios deseados, los guardamos, y hacemos un push, que actualiza la copia del repositorio que tenemos en nuestro GitHub.

(todos estos pasos ya fueron explicados en puntos anteriores)

Cuando se ha hecho el push de esta nueva rama, podremos verla en GitHub. Y ahora si podemos presionar en el boton “pull requests”, que nos lleva a una ventana nueva, donde

veremos el boton “new pull request”. Seleccionamos la rama que hemos creado desde nuestro repositorio con las modificaciones, y a la izquierda podemos seleccionar a que rama del repositorio original se va a fusionar (merge).

Ahora podemos presionar “create pull request”, punto en el cual podemos cambiar el nombre de la solicitud y agregar comentarios. La solicitud queda a la espera de la aprobacion del dueño del repositorio original.

- **¿Cómo aceptar una solicitud de extracción?**

Acto seguido del punto anterior, si somos quienes recibimos el pull request, vamos a tener la posibilidad de revisar los cambios propuestos, y revisar cualquier conflicto que pueda existir. Si todo esta correcto, dentro de la solicitud veremos un boton “merge pull request” para que se fusionen los cambios propuestos en nuestra rama del repositorio original.

- **¿Qué es una etiqueta en Git?**

Una etiqueta es una marca de un momento específico en el desarrollo de nuestro repositorio. Nos permite identificar un momento preciso, como una cierta versión, algun cambio importante dentro del historial, por ejemplo la solución a un error crítico.

- **¿Cómo crear una etiqueta en Git?**

Utilizamos el comando “git tag” y el nombre que queramos asignarle, si estamos marcando una version podríamos usar algo como lo siguiente: `git tag v1.0.0`

- **¿Cómo enviar una etiqueta a GitHub?**

Para enviar una etiqueta al repositorio de GitHub usamos un push, con el nombre de la etiqueta: `git push origin v1.0.0`

- **¿Qué es un historial de Git?**

Es como la linea historica de todos los cambios que hemos hecho, compuesto por todos los commits, que son instantes dentro de esa linea de tiempo.

- **¿Cómo ver el historial de Git?**

Con “git log” podemos ver un historial de todos los commits dentro de la rama sobre la cual estamos posicionados.

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

“git log” es el comando que utilizamos para ver un listado de los commits. El comando tiene varias opciones para ajustar la busqueda por nombre de archivo, por fecha, por autor, o por palabras filtrando a traves de un “string” que proporcionamos nosotros.

- **¿Cómo borrar el historial de Git?**

Podemos usar el comando `git reset` que ofrece distintas opciones de borrado de commits.

`Git reset --soft HEAD~n`: HEAD hace referencia al último commit, y “n” al número de commits que queremos borrar desde este último hacia atrás. La palabra `soft` hace referencia a que se

borraran los commits seleccionados de la sección de trabajo de git, pero los mueve al “staging”. Para borrarlo de manera definitiva usamos:
Git reset –hard HEAD~n

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

Es un repositorio que solo está disponible para ser visto o para hacer fork por el creador y por los usuarios que tengan permiso para colaborar.

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

Seguimos el mismo proceso que hemos explicado en la segunda pregunta, pero seleccionamos la opción de privado en lugar de público.

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

Nos dirigimos a la pagina de nuestro repositorio privado, y desde la seccion de configuracion podemos agregar un colaborador, indicando su nombre de usuario o email.

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

Es un repositorio que esta disponible para cualquier usuario, esta a la vista, y cualquiera puede hacer un fork del mismo, o push requests.

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

Siguiendo los pasos del punto número 2, podremos seleccionar si el repositorio que estamos creando en GitHub va a ser público al momento de crearlo.

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

Vamos a la página del repositorio en GitHub que queremos compartir, y aquí vamos a encontrar la URL del mismo. Esta Url puede ser compartida con cualquier persona que tendra acceso a esta página de GitHub por medio del url.