

Trabajo Practico N2

PROGRAMACION

García Cintia

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

- ¿Qué es GitHub?

GitHub es una comunidad donde se puede compartir nuestros repositorios de manera pública o privada. Podemos ver repositorios de otras personas, se pueden clonar, guardar en favoritos.

- ¿Cómo crear un repositorio en GitHub?

Lo primero es abrir GitHub, ingresar la cuenta que debemos tener, luego vamos del lado derecho donde hay un + y seleccionamos nuevo repositorio, luego seleccionamos público o privado y crear repositorio.

- ¿Cómo crear una rama en Git?

Debemos estar en la terminal y debemos poner git Branch y el nombre de la nueva rama que queremos crear.

- ¿Cómo cambiar a una rama en Git?

Para cambiar de rama debemos escribir en la terminal git checkout y "el nombre de la rama".

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas debemos posicionarnos en la rama donde queremos que se guarden los cambios, ponemos git checkout "nombre de la rama "

Luego ponemos git merge "el nombre de la otra rama que queremos fusionar"

Todo lo que tenga en la rama de abajo se va a volcar en la rama de arriba.

- ¿Cómo crear un commit en Git?

Luego de inicializar git init y desp de haberlo guardado con git add .
Abajo ponemos git commit -m ""(entre comillas el cambio que estamos generando)

- ¿Cómo enviar un commit a GitHub?

Para poder enviar un commit a GitHub, primero debemos clonarlo, luego podemos hacer cambios en el código, los agregamos mediante el comando git add . , creamos un commit -m (ingresando el mensaje con el cambio que realizamos) y con el comando git push origin (main o master o con la rama que estamos trabajando) los enviamos a GitHub.

- ¿Qué es un repositorio remoto?

Un repositorio remoto es una copia de un proyecto que se encuentra en internet.

Que puedo trabajarlo desde distintos lugares, o en grupo, o bien en proyectos de código abierto (con otros programadores de cualquier lugar)

- ¿Cómo agregar un repositorio remoto a Git?

Lo primero es vincular el repositorio local al remoto y asignarle nombre mediante el comando git remote add (nombre) y la url
Luego para verificar que se haya agregado correctamente lo verificamos mediante git remote -v

- ¿Cómo empujar cambios a un repositorio remoto?

Para empujar cambios en un repositorio se realiza mediante git push origin (master o main).

- ¿Cómo tirar de cambios de un repositorio remoto?

Se puede realizar mediante el comando `git pull origin (nombre de la rama)`

- ¿Qué es un fork de repositorio?

En git un fork es una copia de un repositorio creado en una cuenta diferente permitiendo desarrollar cambios sin afectar el origen original. El fork genera una copia en la cuenta del usuario.

- ¿Cómo crear un fork de un repositorio?

Lo primero que debemos hacer es iniciar sesión con nuestra cuenta (GitHub), luego vamos a la carpeta y arriba aparece FORK, podemos cambiar la descripción del repositorio, (la cual queda modificado en el nuestro y no en el original), hacemos click en crear fork y esperamos se copie en nuestra cuenta de GitHub.

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

“Es una solicitud de extracción o petición”

Lo primero es hacer un fork de repositorio, luego clonamos el fork utilizando `git clone (url)`.

Podemos crear una nueva rama (`git checkout -b “nombre de la rama”`), realizamos los cambios que necesitamos (`git add .` y `git commit -m “con la descripción de los cambios”`) y subimos los cambios a fork mediante `git push origin (nombre de la rama)`

Por ultimo en el repositorio hay un botón que dice Compare pull request, le ponemos una descripción de los cambios y click en create pull request.

- ¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud debemos estar en el repositorio, en la pestaña pull request (podemos ver las solicitudes pendientes) Seleccionamos la solicitud de extracción, la verificamos y en caso de estar de acuerdo hacemos click en merge pull request, luego en confirm merge.

- ¿Qué es un etiqueta en Git?

Una etiqueta en Git es una marca que nos señala puntos específicos en la historia del repositorio. Se utiliza para señalar partes importantes en el desarrollo.

Estas son fijas y no cambian.

.

- ¿Cómo crear una etiqueta en Git?

Git utiliza 2 tipos de etiquetas: ligeras y anotadas.

Una etiqueta ligera es un puntero a un commit específico, en cambio las anotadas se guardan en la base de datos de Git como objetos enteros.

Para crear una etiqueta ligera `git tag` (nombre de la etiqueta), esta se creara sobre el último commit de la rama que estemos situados.

Las etiquetas anotadas contienen mayor información, además de apuntar un commit, guarda el nombre del autor, la fecha y un mensaje se debe poner `git tag -a(nombre)`, también se puede especificar el commit al que se desea aplicar la etiqueta.

Cabe aclarar que las etiquetas creadas en el repositorio local no se suben al remoto, para que se envíen debemos realizarlo mediante `git push origin (etiqueta)`

- ¿Cómo enviar una etiqueta a GitHub?

Como dije anteriormente las etiquetas se deben enviar a GitHub, luego de haberlas creado, se debe realizar mediante `git push origin(etiqueta)`

Si se desea enviar varias etiquetas a la vez se debe utilizar el comando `-tags` del comando `git push`

`Git push origin -tags.`

- ¿Qué es un historial de Git?

El historial en Git es un registro de las confirmaciones de un repositorio, donde se muestran los cambios realizados en el código base.

- ¿Cómo ver el historial de Git?

Para ver el historial debemos usar el comando git log, este las lista de manera inversa, se muestra desde lo mas reciente.

- ¿Cómo buscar en el historial de Git?

Se puede buscar de varias formas, dependiendo de lo que estamos buscando.

Si buscamos por un mensaje de commit especifico, se puede usar git log -crep="mensaje de commit"

Si se busca por autor: git log -author="nombre del autor"

Si se busca por fecha: git log -since o -until

Si se busca en un archivo puntual git log --<ruta/del/archivo

Si buscamos por un término dentro de los cambios git log -S"termino que buscamos"

- ¿Cómo borrar el historial de Git?

Se puede realizar de varias maneras, dependiendo de lo que deseamos borrar.

Si deseamos borrar algunos commits específicos se debe usar git rebase-i HEAD~n (si se desea el ultimo commit HEAD~1)

Borrar todo y comenzar nuevamente rm -rf .git, se debe inicializar git init y comenzamos nuevamente.

Si es solo el historial de un archivo especifico debemos usar git filter-branch -force -index-filter

Si queremos borrar commit recientes git reset -soft HEAD~n

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub, es un espacio donde se pueden guardar archivos, códigos y revisiones de forma segura y solo es visible para los que poseen permiso.

- ¿Cómo crear un repositorio privado en GitHub?

Crear un repositorio privado se realiza creando un repositorio, donde está el + a la derecha en la pantalla, seleccionamos nuevo repositorio, escribimos el nombre del repositorio y luego seleccionamos donde dice privado y crear repositorio.

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

Para invitar a alguien para que participe en un repositorio privado, le solicitamos el nombre del usuario. Luego en configuración, vamos a donde dice colaboradores y hacemos click en agregar personas, tecleamos el nombre de la persona que deseamos agregar y agregándolo, le llegara un mail al usuario que deseamos agregar, una vez que este acepte tendrá acceso al repositorio.

• ¿Qué es un repositorio público en GitHub?

Es un espacio de almacenamiento de código, de archivos y revisiones de archivos que se encuentra disponible para todas las personas que utilizan internet.

• ¿Cómo crear un repositorio público en GitHub?

Iniciamos sesión en GitHub, vamos donde está el +, nuevo repositorio, completamos con el nombre que le vamos a dar y seleccionamos la opción de público, y luego crear repositorio.

• ¿Cómo compartir un repositorio público en GitHub?

Una vez subido un repositorio, se debe compartir el enlace del mismo (url)

Por ejemplo: https://github.com/tu_usuario/tu_repositorio

