Trabajo Práctico N°2 Mariano Videla. 1 PROG 2

* **¿Que es GitHub?**

GitHub es un alojamiento de repositorios de control de versiones, en el que se puede almacenar y gestionar proyectos.

* **¿Cuáles son los beneficios de usar GitHub?**

Puede satisfacer las necesidades de proyectos de cualquier tamaño, desde proyectos personales hasta proyectos de grandes empresas.

Ofrece seguridad para proteger los proyectos de los usuarios.

Cuenta con una gran comunidad de desarrolladores para ayudar y colaborar.

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

En nuestra cuenta de GitHub debemos hacer click en new, una vez allí colocar el nombre, alguna descripción de lo que vamos a subir, si queremos que sea público o privado,etc. Luego damos click en create repository.

Nos saldrán varias líneas de comandos como:

…or create a new repository on the comand line..

Nos indica lo que debemos hacer para poder mandar desde nuestro

repositorio local al repositorio que hemos creado en github y enlazarlos.

…or push an existing repository from the command line..

Este caso es para cuando ya tenemos un repositorio local y solamente deseamos mandarlo a la plataforma.

Para ello se deben tipear los siguientes comandos en el bash:

git remote add origin URL del repositorio.

git push -u origin master

Luego refrescamos la página de github para ver como se ha subido el repositorio.

* **¿Cuáles son los tipos de ramas en Git?**

Ramas de Largo Recorrido.

Ramas Puntuales.

Ramas Remotas.

* **¿Cómo crear una rama en Git?**

Se utiliza el comando git branch:

git branch nombre\_de\_la\_rama

Esto creará un nueva rama.

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

Para cambiar de una rama a otra, se utiliza el comando git checkout nombre\_rama. Esto cambia a la rama que deseamos ir.

* **¿Cómo fusionar ramas en Git?**

Primero nos posicionamos en la rama en la que deseamos fusionar los cambios: git checkout nombre\_rama\_destino

Luego se utiliza git merge nombre\_rama\_a\_fusionar para fusionar la rama deseada.

* **¿Cómo crear un commit en Git?**

Primero se agregan los cambios al area de stage con: git add .

Despues se realiza el commit acompañado de un msj, con : git commit -m "Mensaje"

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

Estando en la rama que deseamos commitear se utiliza: git push origin nombre\_rama

* **¿Qué es un repositorio remoto?**

Es una versión del proyecto que está alojada en un servidor o servicio externo (GitHub, GitLab, Bitbucket).

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

Se utiliza: git remote add origin URL . Donde URL es la que corresponde al repositorio remoto.

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

Nos posicionamos en la rama deseada y se utiliza: git push origin nombre\_rama

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

Estando en la rama deseada usamos: git pull origin nombre-de-la-rama

Lo que actualiza la rama local con los cambios del repositorio remoto.

* **¿Qué es un fork de repositorio?**

Un fork de repositorio es una copia que se crea de un repositorio existente de otro usuario en nuestra cuenta de GitHub.

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

Vamos a la página del repositorio original en GitHub, hacemos click en el boton fork y seleccionamos nuestra cuenta donde queremos crear en el fork.

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

Si los cambios están en una rama de nuestro fork empujamos los cambios a nuestro repositorio remoto: git push origin nombre\_rama. Luego vamos al repositorio donde queremos hacer los cambios y hacemos click en pull requests y luego en new pull request. Seleccionamos la rama y el repositorio original para hacer la comparación y luego hacemos click en create pull request.

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

Vamos al repositorio donde se ha enviado la solicitud de extracción, seleccionamos pull requests, damos click en la solicitud de extracción que queremos aceptar y hacemos click en el boton merge pull request. Luego, confirmamos haciendo click en confirm merge**.**

* **¿Qué es un etiqueta en Git?**

Git tiene la posibilidad de etiquetar puntos específicos del historial como

importantes. Esta funcionalidad se usa típicamente para marcar versiones de lanzamiento.

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

La forma más fácil de crear una etiqueta anotada es especificar la

opción -a cuando ejecutamos el comando git tag:

git tag -a v1.4 -m 'mi version 1.4'

git tag

v0.1

v1.3

v1.4

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

Habiendo creado la etiqueta en nuestro repositorio local,para enviarla podemos empujar la etiqueta al respositorio remoto usando: git push origin nombre\_etiqueta

* **¿Qué es un historial de Git?**

Es un registro de todos los cambios realizados en un repositorio. Incluye información sobre cada commit, como el mensaje del commit, el autor, la fecha y los cambios específicos realizados. Permite rastrear y revertir cambios.

* **¿Cómo ver el historial de Git?**

Para esto se ultiliza: git log

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

Existen distintas opciones, por ej. Para buscar por el mensaje de un commit se utiliza: git log --grep="texto a buscar"; por autor se utiliza : git log --author="nombre del autor"; o por fecha: git log --since="fecha inicio" --until="fecha fin"

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

Para borrar el historial se utiliza: rm -rf .git

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

Es un repositorio cuyo contenido solo es accesible para los colaboradores que nosotros elijamos.

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

Repetir los pasos de la tercera pregunta de este TP y en la sección de visibility, seleccionamos la opción private.

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

Desde la página del repositorio en GitHub hacemos click en settings y seleccionamos collaborators & teams. En el campo search by username, full name, or email address, escribimos el nombre de usuario de GitHub de la persona que deseamos invitar y seleccionamos al colaborador deseado y hacemos click en add collaborator.

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

Es un repositorio cuyo contenido esta disponible para cualquier persona.

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

Se crea siguiendo los mismos pasos que el repositorio privado, pero en la seccion visibility, seleccionamos la opcion public.

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

Se puede enviar la URL del repositorio deseado a la persona que deseamos compartir, atravez de correo, mensajes, redes sociales, etc.

Tambien se puede hacer desde el boton share o iconos de redes sociales que estan en la pagina de nuestro repositorio.

* **¿Qué es un repositorio de código en GitHub?**

Es un repositorio donde se almacena y gestiona el código fuente de un proyecto de software.

* **¿Cómo crear un repositorio de código en GitHub?**

Se crea igual que un repositorio comun, dandole un nombre y asignando opciones de visibilidad (publico o privado), crear un archivo Readme, gitignore, etc.

* **¿Cómo compartir un repositorio de código en GitHub?**

Al igual que un repositorio comun, puede compartirse atravez de su URL.

* **¿Qué es un repositorio de documentación en GitHub?**

Es un repositorio diseñado específicamente para almacenar y gestionar documentos relacionados con un proyecto en particular. Puede contener información textual, guías, tutoriales, manuales y otros documentos de soporte

* **¿Cómo crear un repositorio de documentación en GitHub?**

Siguiendo los pasos anteriores de repositorio y repositorio de codigo y elijiendo opciones como crear archivos readme, gitignore y tambien elijiendo su visibilidad.

* **¿Cómo compartir un repositorio de documentación en GitHub?**

Atraves de su URL.

* **¿Qué es un repositorio de proyecto en GitHub?**

Es un espacio para almacenar, gestionar y colaborar en el desarrollo de un proyecto. Puede contener archivos de codigo fuente, historial de versiones, archivos readme y contributing, configuracion, etc.

* **¿Cómo usar Git para trabajar con otros?**

Realizando acciones como clonar un repositorio remoto o subir uno de forma publica o en caso de ser privado compartiendo la URL. Creando ramas que permitan trabajar de forma simultanea en un mismo proyecto. Realizando cambios y actualizando al repositorio remoto atraves de add y commit teniendo una sincronizacion entre nuestro repositorio local y el remoto. Son algunas formas de usar git para trabajar y colaborar con otras personas.