LABORATORIO 1

TP N2

Mauro Garzia

Comision B

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

• ¿Qué es GitHub?

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el control de versiones de Git.

• ¿Cuáles son los beneficios de usar GitHub?

Los beneficios de usar GitHub son: poder intercambiar archivos desde la misma plataforma, tener varias ramas de un solo proyecto, volver a un punto de guardado especifico del proyecto en el que estemos trabajando

• ¿Cómo crear un repositorio en GitHub?

Primero hay que loguearse en la pagina de GitHub, una vez hecho esto se debe ir a la parte de repositorios y tocar en “Nuevo”. Despues existe la la opción para crearlo con un ReadMe, o sin el. En caso de ser sin el, lo que se debe hacer es ir a la terminal de GitBash y poner el comando “git init” para inicializar el repositorio, luego se coloca el comando “git remote add orgin” y la url del repositorio. Con esto ya se pueden subir archivos usando los comando “git add” y luego agregando un commit con “git commit -u ‘Nombre del commit’ ”, y asi ya aparecerían los archivos en el repositorio.

• ¿Cuáles son los tipos de ramas en Git?

Hay ramas padre y ramas hijo, además de una rama principal referida a un proyecto y las demás sub-ramas en las que se han hecho modificaciones de problemas surgidos en los que se debe trabajar sin comprometer lo que seguimos haciendo, luego estas ramas deberían ser fusionadas.

• ¿Cómo crear una rama en Git?

Para crear una rama en Git se usa el comando git branch (nombre de la rama) o git checkout -b (nombre de la rama) para crearla y saltar automáticamente a ella.

• ¿Cómo cambiar a una rama en Git?

Para cambiar de una rama a otra se utiliza el comando git checkout (nombre de la rama)

• ¿Cómo fusionar ramas en Git?

Para fusionar ramas se utiliza el comando git merge (nombre de la rama), para hacer esto es necesario encontrarse en la rama principal, para asi traer la informacion de la rama secundaria y combinarla con la principal.

• ¿Cómo crear un commit en Git?

Para crear un commit en Git se usa el comando git commit -u “Nombre del commit”, esto luego de haber agregado el los cambios del archivo al stage con git push.

• ¿Cómo enviar un commit a GitHub?

Un commit se envia usando el comando git commit -u “Nombre del commit” luego de haber adherido, eliminado o modificado algún archivo que se encuentra en el repositorio, seguido de esto se usa el comando git push para mandar el commit.

• ¿Qué es un repositorio remoto?

Un repositorio remoto es un repositorio donde guardamos los archivos o proyectos que tenemos en nuestro disco local, pero estos están guardados en la red, a su vez podemos mandar los archivos o extraerlos con ciertos comandos, para poder ser modificados. El repositorio puede ser privado o publico.

• ¿Cómo agregar un repositorio remoto a Git?

Con el comando de git init para inicializar el repositorio y luego con el comando git remote add origin y la url del repositorio que queremos agregar.

• ¿Cómo empujar cambios a un repositorio remoto?

Se pueden empujar cambios utilizando el comando git push, esto despues de haberlos agregado con git add y haber hecho un commit.

• ¿Cómo tirar de cambios de un repositorio remoto?

Los cambios desde un repositorio remoto se tiran con el comando git pull

• ¿Qué es un fork de repositorio?

Es una copia de un repositorio que no ha parecido interesante y que queremos hacerle algunas modificaciones, esto sirve para poder trabajar con los archivos de ese proyecto y los cambios que hagamos no iran al repositorio original del autor de ese proyecto, sino a nuestra copia local.

• ¿Cómo crear un fork de un repositorio?

Para crear un fork buscamos algún repositorio que nos parezca interesante y damos click en Fork situado en la parte superior derecha de github. Luego comenzara el forking.

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

Para enviar una solicitud de pull request hay que dirigirse a la solapa de Pull request, allí daremos click en new pull request, veremos una ventana a modo de resumen en donde se reflejarán los cambios que hemos hecho nosotros en comparación al repositorio original y mas abajo tenemos lugar para explayarnos en los cambios que hemos hecho.

• ¿Cómo aceptar una solicitud de extracción?

El autor del repositorio verá en sus pull requests el mensaje que le hemos enviado, para que lo pueda observar y si lo considera realizar el cambio pertinente (además de poder responderle al usuario que le ha propuesto ese cambio).

• ¿Qué es un etiqueta en Git?

Una etiqueta en Git es una funcionalidad que permite al usuario marcar puntos específicos del historial como importantes, esta funcionalidad se usa típicamente para marcar versiones de lanzamiento. Pueden ser ligeras (un puntero a un commit en especifico) y anotadas (se guardan en la base de datos de Git como objetos enteros).

• ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta en Git se debe especificar la opción -a cuando ejecutas el comando git tag, Ej: git tag -a v1.4 -m ‘my versión 1.4’. La opción -m especifica el mensaje de la etiqueta. Todo esto es para crear una etiqueta anotada.

Para crear una etiqueta ligera no se pasan las opciones -a,-s, ni -m. Simplemente se pone : git tag v1.4-lw

Para etiquetar un commit que no habias etiquetado antes se debe especificar el cheksum del commit al final del comando: git tag -a v1.2 9fceb02

• ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub se gace con git psuh origin (nombre de la etiqueta), para enviar varias etiquetas a la vez se puede usar la opción de git push --tags

• ¿Qué es un historial de Git?

Un historial de Git es la secuencia de los commits que se han hecho durante el tiempo en el que hemos estado trabajando en el proyecto, se utiliza para ver donde estamos y que hemos estado haciendo.

• ¿Cómo ver el historial de Git?

Para ver el historial de Git se debe ingresar el comando git log, y el historial de los commits aparecerán desde los últimos que aparecen arriba hasta los primeros que salen mas abajo.

• ¿Cómo buscar en el historial de Git?

Podemos usar el comando git log --oneline para ver mas cantidad de commits en la pantalla, también podemos usar git log (numero) para ver una cantidad de commits en especifico, por si no queremos ver todo el historial de commits. Para ver información extendida de un commit usamos el comando git log –(numero) -p , esto mostraria los cambios en el código de los últimos dos commits. Para ver mas cambios en concreto de un commit usamos git show.

• ¿Cómo borrar el historial de Git?

Primero eliminamos el repositorio de forma local utilizando el comando rm -ef .git, luego creamos un nuevo repositorio desde cero en el mismo derictorio.

• ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es un repositorio donde no todos pueden ver su contenido, solo las personas a las que el dueño del repositorio les de acceso pueden ver su contenido.

• ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en GitHub lo que se debe hacer es que en el momento de la creación del repositorio hay una opción “Private”, donde si la marcamos nuestro repositorio solo será visible para nosotros y un grupo selecto de personas a las que les mandemos el repositorio.

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

Para invitar a alguien a un repositorio privado se debe solicitar el nombre de la persona a la que se esta queriendo invitar, luego en Configuraciones se selecciona “Acceso”, luego en “Agregar personas”, se pone el nombre de la persona que se desea invitar y después hay que hacer click en “Agregar ‘Nombre de la persona’ al Repositorio”.

• ¿Qué es un repositorio público en GitHub?

Un repositorio publico en GitHub es un repositorio donde cualquier persona podrá ver su contenido.

• ¿Cómo crear un repositorio público en GitHub?

Para crear u repositorio publico en GitHub lo que se debe hacer es que en momento de crearlo debemos elegir la opción “Public”, la cual hará que nuestro repositorio sea publico para cualquier persona.

• ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio publico en GitHub lo que se debe hacer es compartir el link de la pagina del repositorio y mandárselo a la persona que queramos que tenga acceso a este.

• ¿Qué es un repositorio de código en GitHub?

Un repositorio de código en GitHub es un repositorio es subido contenido de código de programación.

• ¿Cómo crear un repositorio de código en GitHub?

Simplemente vamos a nuestro perfil de GitHub, buscamos la opción de crear un repositorio nuevo y lo hacemos, el repositorio se puede configurar antes de terminar de crearlo.

• ¿Cómo compartir un repositorio de código en GitHub?

Para compartir un repositorio se puede simplemente compartir el link que lleve al mismo, en caso de querer agregar personas que contribuyan al repositorio hay que agregarlos como colaboradores desde la configuración del repositorio

• ¿Qué es un repositorio de documentación en GitHub?

Un repositorio de documentación es un tipo de repositorio que se usa para almacenar y gestionar documentos relacionados a un proyecto, los documentos en este repositorio pueden contener información como guías, manuales, instrucciones de instalación y cualquier documentación relevante para el proyecto.

• ¿Cómo crear un repositorio de documentación en GitHub?

Crear un repositorio de documentación es igual que crear cualquier repositorio solo que en este caso se guardan documento y es necesario que tenga una estructura bien organizada de acuerdo a nuestras necesidades.

• ¿Cómo compartir un repositorio de documentación en GitHub?

Se puede compartir el link del repositorio para que las personas lo visiten o podemos agregar colaboradores al repositorio.

• ¿Qué es un repositorio de proyecto en GitHub?

Un repositorio de proyecto es un espacio donde se almacena, gestiona y organiza el código fuente, recursos y archivos relacionados con un proyecto de software. De esta forma se puede gestionar el desarrollo y mantener un historial de cambios.

• ¿Cómo usar Git para trabajar con otros?

Para empezar cada colaborador tiene que clonar el repositorio en sus computadoras, crear nuevas ramas para soluciones de problemas o nuevas características, los cambios realizados sincronizarlos con el repositorio remoto y enviarlos.

También podemos revisar el trabajo de nuestros compañeros y dejarles comentarios en caso de ser necesario.