Práctico 2: Git y GitHub

•¿Qué es GitHub?

La plataforma que ofrece alojamiento para repositorios de control de versiones, que permite a los desarrolladores almacenar y gestionar sus proyectos de software. Gratis y de código abierto, permite el trabajo en equipo en proyectos, el intercambio de código y el trabajo conjunto de forma eficiente.

• ¿Cómo crear un repositorio en GitHub?

primero crear una cuenta en la web de GitHub, configurar el perfil y empezar a crear repositorios. Hacer click en New, colocar un nombre, alguna que otra descripción de lo que vamos a subir, si queremos que sea público o privado y demás y click en Create Repository.

• ¿Cómo crear una rama en Git?

Para crear una nueva rama, se usará el comando git branch: + nombre de la rama nueva

• ¿Cómo cambiar a una rama en Git?

Para cambiar a una rama nueva, utilizar el comando git checkout: + nombre de la rama nueva

Para crear una nueva rama y colocarse en ella, en un solo paso, utilizar el comando git checkout con la opción -b: $ git checkout -b + nombre de la rama nueva

• ¿Cómo fusionar ramas en Git?

Primero, debes estar en la rama a la que quieres fusionar los cambios, ej: $ git checkout master ó main y $ git merge nuevaRama

• ¿Cómo crear un commit en Git?

Primero realiza los cambios en los archivos de tu proyecto que deseas guardar en el commit, Luego agregar los cambios al área de preparación. con git add + nombre del archivo o para todos los archivos $ git add . Luego describir los cambios con $ git commit –m “Mensaje con el nombre de los cambios”

• ¿Cómo enviar un commit a GitHub?

Primero clonar el repositorio de GitHub a nuestra máquina local con el comando: git clone https://github.com/tu\_usuario/tu\_repositorio.git y cd tu\_repositorio, realizar los cambios en los archivos del repositorio y agrégalos con el comando: git add nombre\_del\_archivo o git add .; crear el comnit con git commit -m "Descripción de los cambios" y enviarlos al repositorio con el comando git push origin nombre\_de\_la\_rama

• ¿Qué es un repositorio remoto?

Son versiones de tu proyecto que están hospedadas en Internet o en cualquier otra red.

• ¿Cómo agregar un repositorio remoto a Git?

Con el comando git remote add para vincular tu repositorio local con un repositorio remoto y asignar un nombre a ese remoto

• ¿Cómo empujar cambios a un repositorio remoto?

Con los comandos: git pull origin nombre\_de\_la\_rama lo obtenemos y lo Empujamos al repositorio remoto: git push origin nombre\_de\_la\_rama

• ¿Cómo tirar de cambios de un repositorio remoto?

Con el comando git pull origin nombre\_de\_la\_rama

• ¿Qué es un fork de repositorio?

Es un botón en la cuenta de otro usuario que al tocarlo creará una copia del repositorio en tu propia cuenta de GitHub, al clonarlo se podrá utilizar de forma independiente y realizar cambios en el de forma local.

• ¿Cómo crear un fork de un repositorio?

1. Ir a la página principal del repositorio que quieres forkear
2. Presionar el botón Fork
3. Elegir un nombre y una descripción para el fork (opcional)
4. Presionar Create fork

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

Ir a la solapa de Pull requests allí daremos click en new pull request – colocar un mensaje, e indicar porque deberían de aceptar el cambio y aplicarlo en el repositorio

• ¿Cómo aceptar una solicitud de extracción?

Luego de ver el mensaje en sus pull requests hará click en Merge pull request y de esta manera podrá agregar a su repositorio los cambios que hizo un usuario

• ¿Qué es una etiqueta en Git?

Es una etiqueta que marca puntos o detalles específicos del historial como importantes

• ¿Cómo crear una etiqueta en Git?

usar el comando git tag o GitHub Desktop.

• ¿Cómo enviar una etiqueta a GitHub?

Crear una etiqueta en tu repositorio local, empujarla al repositorio remoto con git push origin + nombre de la etiqueta y para empujar todas las etiquetas nuevas con el comando: git push origin --tags

• ¿Qué es un historial de Git?

Es la secuencia de todos los cambios realizados en un repositorio de Git

• ¿Cómo ver el historial de Git?

Con el comando: git log

• ¿Cómo buscar en el historial de Git?

Para buscar commits que contengan una palabra o frase específica en el mensaje de commit, usa git log con la opción –grep: git log --grep="palabra clave"

Para encontrar commits hechos por un autor específico, usa --author: git log --author="Nombre del Autor"

• ¿Cómo borrar el historial de Git?

Con los comandos:

git reset: Quita del stage todos los archivos y carpetas del proyecto.

git reset + nombreArchivo: Quita del stage el archivo indicado.

git reset + nombreCarpeta/: Quita del stage todos los archivos de esa carpeta.

git reset + nombreCarpeta/nombreArchivo: Quita ese archivo del stage (que a la vez está dentro de una carpeta).

git reset + nombreCarpeta/\*.extensión: Quita todos los archivos que cumplan con la condición indicada previamente dentro de esa carpeta del stage.

• ¿Qué es un repositorio privado en GitHub?

Es un tipo de repositorio donde el contenido es accesible solo para usuarios que han sido autorizados.

• ¿Cómo crear un repositorio privado en GitHub?

Ir a la página de principal de creación de repositorio, hacer clic en el botón “+” y seleccionar “New Repository”, Seleccionar la configuración de privacidad en privado, completar demás campos y crear

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

Acceder al repositorio, y tocar dentro en la pestaña "Settings", Seleccionar "Collaborators" y tocar el botn Add people, ingresar el nombre del usurio y permitir accesos según preferencia, y tocar en el botón add para enviar invitacion

• ¿Qué es un repositorio público en GitHub?

Es un repositorio cuyo contenido es accesible a cualquier persona en Internet.

• ¿Cómo crear un repositorio público en GitHub?

Ingresar a la página de creación de repositorios, s hacer clic en el botón “+” y seleccionar “New Repository” o hacer clic en “New” completar los campos deseados y el botón de créate repository

• ¿Cómo compartir un repositorio público en GitHub?

Dentro del repositorio, copia la URL del cuadro de texto que dice "<> Code” y dentro de un repositorio propio