Trabajo Práctico Nº 2: GIT y GIT HUB

Alumna: Díaz de Quintana, Melisa

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

- ¿Qué es GitHub?

  GitHub es una plataforma que ofrece alojamiento de repositorios de control de versiones. Es un sistema de control de versiones distribuido. Permite a los desarrolladores llevar un registro de los cambios realizados en archivos a lo largo del tiempo, generalmente en proyectos de software.

- ¿Cómo crear un repositorio en GitHub?

  Iniciar sesión.

  Luego clickear en Create New Repository. Una vez ahí colocamos el nombre, una descripción si se desea, agregar configuraciones deseadas.

  Dos opciones:

  ...or create a new repository on the comand line: Indica los pasos para enviar desde el repositorio local al repositorio creado enGitHub.

  ...or push an existing repository from the command line: Si ya se tiene un repositorio local y enviarlo al existente.

  Para conectar el repositorio local con el remoto tipear comandos en el bash:

  git remote add origin https://github.com/tu_usuario/mi-repositorio.git

  git push -u origin master

- ¿Cómo crear una rama en Git?

  Una vez inicializado el repositorio de git, crea por defecto la rama o branch principal (main o master). Es una línea de tiempo o de cronología en la que se puede plasmar los distintos puntos en los que hubo cambios en el proyecto.

  Para crear una nueva rama, se usará el comando git branch:

  git branch nuevaRama

- ¿Cómo cambiar a una rama en Git?

Se utiliza el comando checkout nombre-rama:

git checkout nuevaRama

- ¿Cómo fusionar ramas en Git?

  Se debe estar en la rama a la se quiere hacer los cambios. Suponiendo que los cambios queremos aplicarlos en la rama master:

  git checkout master

  Luego para aplicar los cambios de ramanueva a master utilizamos el comando merge.

  git merge ramanueva

  Se incorporan los cambios de ramanueva a la rama master.

- ¿Cómo crear un commit en Git?

  Se guardan los cambios realizados en el repositorio. Se utiliza el comando commit.

  Primero agregamos los cambios al área de preparación:

  git add archivo (para agregar el archive “archivo”)

  o

  git add . (para agregar todos los archivos)

  Una vez realizados los cambios se procede al commit:

  git commit –m “descripción cambios”

- ¿Cómo enviar un commit a GitHub?

  Primero debemos clonar el repositorio de GitHub a nuestra máquina local:

  git clone https://github.com/tu_usuario/tu_repositorio.git

  cd tu_repositorio

  Haz cambios en los archivos del repositorio y agrégalos:

  git add nombre_de_archivo o git add .

  Crea un commit de tus cambios:

  git commit -m "Descripción de los cambios"

  Para enviar los commits al repositorio en GitHub, debemos empujarlos con: git push origin nombre_de_la_rama

- ¿Qué es un repositorio remoto?

  Un repositorio remoto es una versión de un repositorio que está almacenada en un servidor o en una plataforma en línea, como **GitHub** o GitLab. A diferencia de un repositorio local, que está en tu computadora, un repositorio remoto te permite almacenar tu código y otros archivos en un servidor accesible desde cualquier lugar, lo que facilita la colaboración, el respaldo y el acceso a tu proyecto desde diferentes dispositivos.

- ¿Cómo agregar un repositorio remoto a Git?

  Ya con el repositorio local en tu computadora, se puede vincular con un repositorio remoto usando el comando remote add. Esto permite que puedas enviar y recibir cambios entre ambos.

  ```
  git remote add
  ```

- ¿Cómo empujar cambios a un repositorio remoto?

  Después de hacer cambios y confirmarlos (mediante un commit) en tu repositorio local, puedes subirlos al repositorio remoto usando el comando git push.

  ```
  git push -u origin main
  ```

- ¿Cómo tirar de cambios de un repositorio remoto?

  Si otras personas han hecho cambios al repositorio remoto, se pueden importar esos cambios al repositorio local con el comando git pull.

  ```
  git pull origin main
  ```

- ¿Qué es un fork de repositorio?

  Un fork es una copia de un repositorio de código fuente que se crea en tu propia cuenta. Al hacer un fork, obtienes una versión completa del repositorio original, lo que te permite experimentar y hacer cambios sin afectar al proyecto original.

  Es una herramienta comúnmente utilizada en proyectos de código abierto para que los desarrolladores puedan colaborar sin necesidad de permisos directos en el repositorio principal.

- ¿Cómo crear un fork de un repositorio?

Busca el repositorio en el que estás interesado y ve a su página. Hacer click en el botón Fork en la esquina superior derecha. Esperar a que se cree el fork, GitHub va a hacer una copia del repositorio en tu cuenta personal. Una vez completado el proceso, serás redirigido a tu versión del repositorio, que ahora estará bajo tu cuenta de GitHub.

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

  Para hacer el pull request nos dirigiremos a la solapa de Pull requests 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 (el código original). Daremos click en Create pull request donde veremos el asunto (colocamos algún mensaje global) y más abajo tenemos suficiente lugar para poder explayarnos en mencionar el por qué ese cambio que hemos realizado nosotros, sería considerado como algo que al repositorio original le vendrían bien agregarlo.

- ¿Cómo aceptar una solicitud de extracción?

  El autor del repositorio verá en sus pull requests el mensaje del pull request, para que lo pueda observar y si lo considera realizar el cambio pertinente. Opcionalmente el usuario original podrá hacer un merge a la rama principal del proyecto en cuestión.

- ¿Qué es un etiqueta en Git?

  Es un marcador o referencia que se utiliza para señalar un punto específico en el historial de un repositorio, generalmente para marcar una versión importante o un hito en el desarrollo del proyecto.

- ¿Cómo crear una etiqueta en Git?

  Hay dos tipos principales de etiquetas: ligeras y anotadas.

  Etiquetas ligeras (lightweight tags): Son simplemente un puntero a un commit específico. Es como un marcador que no contiene más información que la referencia al commit.

  Se usa cuando solo deseas "marcar" un punto específico en el tiempo sin mucha información adicional.

  Etiquetas anotadas (annotated tags): Son más completas y se almacenan como objetos completos en el repositorio. Una etiqueta anotada contiene información adicional como el nombre del autor, la fecha y un mensaje. Además, puedes firmar una etiqueta con GPG para mayor seguridad.

Las etiquetas anotadas son recomendadas cuando deseas un registro más detallado y oficial de la etiqueta.

- ¿Cómo enviar una etiqueta a GitHub?

  Utilizar el comando git tag.

  git tag v1.0 (etiqueta ligera)

  git tag -a v1.0 -m "Versión 1.0" (etiqueta anotada)

- ¿Qué es un historial de Git?

  El historial de Git es una secuencia de commits que representan los cambios realizados en un proyecto a lo largo del tiempo. Se puede visualizar con comandos como git log, y ofrece información detallada sobre cada cambio, quién lo hizo y cuándo. Esto es fundamental para gestionar versiones, colaborar entre equipos y rastrear el progreso de un proyecto.

- ¿Cómo ver el historial de Git?

  El comando principal para ver el historial de commits en Git es 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 buscar commits que han modificado un archivo específico, usa git log seguido del nombre del archivo:

  git log -- nombre_del_archivo

  Para buscar commits en un rango de fechas específico, usa las opciones --since y --until:

  git log --since="2024-01-01" --until="2024-01-31"

  Para encontrar commits hechos por un autor específico, usa --author:

  git log --author="Nombre del Autor"

- ¿Cómo borrar el historial de Git?

  El comando git reset se utiliza sobre todo para deshacer las cosas. Se mueve alrededor del puntero HEAD y opcionalmente cambia el index o área de

ensayo y también puede cambiar opcionalmente el directorio de trabajo si se utiliza - hard. Esta última opción hace posible que este comando pueda perder tu trabajo si se usa incorrectamente, por lo que asegúrese de entenderlo antes de usarlo.

Existen distintas formas de utilizarlo:

- 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?

  Un repositorio privado en GitHub es un repositorio cuyo contenido está restringido a las personas que tienen permisos específicos. Es ideal para proyectos que contienen información sensible o privada y permite un control completo sobre quién puede ver o colaborar en el código. A diferencia de los repositorios públicos, los privados son invisibles para el público general y solo los usuarios autorizados pueden acceder a ellos.

- ¿Cómo crear un repositorio privado en GitHub?

  Iniciar sesión en tu cuenta de GitHub. Al crear el nuevo repositorio te da opciones de configuración del mismo. En la opción "Visibility", selecciona "Private". Una vez configurado “Create repository”

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

  Desde el repositorio al que quiero invitar a alguien accedo a la configuración del repositorio (“settings”) y buscar la opción “Collaborators & teams”; luego busco a quien quiero darle acceso por nombre de usuario, nombre completo o mail, “Search by username, full name or email address”. Luego selecciono el tipo de acceso que quiero ofrecerle a la persona, Read (solo lectura), Write (puede modificar), Admin (full access). Clickear “add” para enviar la invitación.

  La persona invitada deberá aceptar la invitación.

- ¿Qué es un repositorio público en GitHub?

  Es un repositorio cuyo código y contenido son accesibles por cualquier persona en la web. Los repositorios públicos son ideales para proyectos de código abierto y permiten a los desarrolladores colaborar entre sí a través de pull requests, issues y otras funcionalidades de GitHub. Son gratuitos y accesibles para todos, y pueden ayudar a mejorar la visibilidad de tu trabajo y a recibir contribuciones de la comunidad global.

- ¿Cómo crear un repositorio público en GitHub?

  Inicializada la sesión en GitHub hacer click en el botón New para crear un nuevo repositorio. Completar la información básica del repositorio, nombre, descripción, etc. En la sección Visibility seleccionar Public. Una vez configurado, crear el repositorio “Create repository”.

- ¿Cómo compartir un repositorio público en GitHub?

  En la parte superior de la página de tu repositorio, encontrarás la barra de direcciones del navegador con la URL de tu repositorio. GitHub también ofrece un pequeño icono de "compartir" en la página de tu repositorio, que te permite compartir directamente en redes sociales o generar un enlace de forma rápida.

  EJERCICIO 2: https://github.com/melisaddq/Primer-repo-GIT.git

  EJERCICIO 3: https://github.com/melisaddq/conflict-exercise.git