Práctico 2: Git y GitHub

Actividades 1

¿Qué es GitHub?

Es una plataforma online de desarrollo colaborativo donde podemos almacenar, compartir y trabajar junto con otros usuarios.
Podemos guardar nuestro trabajo, código y demás en un repositorio remoto, en forma pública o privada. Seguir y administrar los cambios en el código a lo largo del tiempo manteniendo un registro detallado. También permite que otros usuarios revisen y sugieran cambios. Seguir a otros usuarios y viceversa.

¿Cómo crear un repositorio en GitHub?

Debemos crear primero una cuenta. Al hacerlo disponemos de un perfil, símil red social.
En la esquina superior derecha hacemos click en el ícono '+' y seleccionamos nuevo repositorio. Le damos un nombre. Añadimos una descripción (opcional) que describe de qué se trata el proyecto.
Elegimos la visibilidad del repositorio, que puede ser pública o privada.
Podemos también inicializar el repositorio con Readme (opcional), esto creará un archivo con información básica.

¿Cómo crear una rama en Git?

Para crear una rama lo hacemos a través del **comando git branch** "nombre de la rama". También con este comando podemos verificar en qué rama nos encontramos.

¿Cómo cambiar a una rama en Git?

Para cambiar de rama utilizamos el **comando git checkout** "nombre de la rama". También podemos crear y cambiar de rama en un solo paso con el **comando git checkout -b** "nombre de la rama"

¿Cómo fusionar ramas en Git?

Para fusionar ramas lo realizamos a través del **comando git merge**.

¿Cómo crear un commit en Git?

Para crear un commit, primero chequeamos que estemos en el repositorio correcto con el comando cd / ruta /repositorio. Luego revisamos el estado del repositorio con el comando git status. Continuamos con el comando git add . para agregar todos los archivos y carpetas modificados. Para agregar archivos específicos utilizamos el comando git add archivo (el que queremos).
Ahora para que Git registre las modificaciones y tener un punto para ver como fue cambiando nuestro proyecto en cada instante utilizamos el **comando git commit -m** "mensaje descriptivo del commit"
Si queremos verificar que el commit se haya realizado en forma correcta lo realizamos con el comando git log (muestra historial de commits en el repositorio).

¿Cómo enviar un commit a GitHub?

Para hacerlo, una vez dentro de nuestro perfil de GitHub, hacemos click en el ícono '+' y luego click en new repository. Le colocamos un nombre (podemos hacer que coincida con el del repositorio local). También una descripción (si queremos). Podemos dejarlo público o privado.
Ya tenemos creado un repositorio, copiamos la instrucción git remote add origin https … y lo pegamos en la consola de Git (lo que estamos haciendo es agregar la dirección de nuestro repositorio remoto).
Ahora para subir los cambios de nuestro repositorio local al repositorio remoto utilizamos el **comando git push -u origin rama** (rama es el nombre de la rama donde estamos trabajando, ej: main o master).
Para ver el cambio, en la consola ponemos git add . (Para agregar todos los archivos)

¿Qué es un repositorio remoto?

Es un almacenamiento de código, que se encuentra fuera de nuestra máquina local, desde el cual podemos colaborar con otros usuarios. Podemos además de almacenar y gestionar nuestros proyectos de manera centralizada, compartir los cambios, sincronizar el código y que múltiples usuarios trabajen en el mismo código.
Un repositorio remoto está alojado en un servidor (GitHub), donde guardamos las versiones de nuestros proyectos.

¿Cómo agregar en repositorio remoto a Git?

Para hacerlo, si no tenemos el repositorio local inicializado utilizamos el comando git init. Luego agregamos el repositorio remoto a través del **comando git remote add origin** <URL del repositorio remoto>
Para chequear si se agregó correctamente utilizamos el comando git remote -v

¿Cómo empujar cambios a un repositorio remoto?

Una vez que tenemos configurado nuestro repositorio remoto a través de git remote add origin <URL del repositorio> , verificado que este configurado con comando git remote -v , hecho los commit y agregado los cambios, empujamos (push) los cambios al repositorio remoto por medio del **comando git push -u origin main** (o también master).

¿Cómo tirar cambios de un repositorio remoto?

Para tirar (pull) los cambios de un repositorio remoto a Git, utilizamos el **comando git pull**
Este comando obtiene (fetch) los cambios del repositorio remoto y fusiona (merge) con la rama local.
Para especificar una rama remota y una local, usamos el comando git pull origin <nombre de rama>. (Origin, nombre del repositorio por defecto. Nombre de rama, nombre de la rama en el repositorio remoto de la cual quiero los cambios). Si queremos descargar los cambios desde el remoto sin fusionarlo a la rama local utilizamos el comando git fetch
Si queremos fusionar estos cambios lo hacemos con el comando git merge origin/nombre de rama.

¿Qué es un fork de repositorio?

Es una copia de un repositorio que se crea en nuestra cuenta de GitHub (sistema de control de versiones). Esto permite trabajar con nuestro proyecto sin riesgo de afectar el proyecto original. Al hacer un fork se obtienen los archivos, ramas e historial del repositorio local. A partir de este momento, podemos modificarlo, agregar cosas, corregir errores, etc. Si queremos agregar nuestros cambios al repositorio original, hacemos un pull request, donde el propietario del repositorio original decide si acepta o no estos cambios.

¿Cómo crear un fork de un repositorio?

Para crearlo, vamos al repositorio de GitHub que queremos hacer un fork. Click en botón fork, en la parte superior derecha. Seleccionamos la cuenta u organización que queremos hacer el fork. Una vez completado el proceso, podemos trabajar en nuestra copia del repositorio sin afectar el original. Podemos cambiar la descripción del repositorio. Este cambio quedara en nuestro repositorio, y no en el original. Click en create fork.
A diferencia del clonado, que descarga el repositorio localmente, el fork se realizada a través de una copia generada en nuestra cuenta de usuario.

Si deseamos trabajar de manera local, lo podemos clonar usando Git. Utilizamos para ello el comando git clone https://github.com/usuario/nombre repositorio.git

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

Para hacerlo, primero hacemos un fork. Lo clonamos si es necesario. Creamos una nueva rama, esto evita conflictos con la rama principal. Realizamos los cambios, agregamos los archivos modificados, hacemos un commit y subimos los cambios al repositorio remoto
git add .
git commit -m “descripción de los cambios realizados”
git push origin <nombre rama>
Dentro del repositorio de GItHub (fork clonado). Nos pregunta si deseamos crear un pull request para la rama que subimos. Luego click en compare & pull request. Si no vemos el mensaje, en la página de nuestro repositorio hacemos click en pull request. Luego click en new pull request. Chequeamos que la rama sea la misma rama del repositorio original (o rama en la que queremos hacer el pull request) y comparamos los cambios con la rama de destino. Le ponemos un título y una descripción. Hacemos click en create pull request.

¿Cómo aceptar una solicitud de extracción?

Accedemos al repositorio en GitHub donde recibimos solicitud de extracción. Vamos a la pestaña pull request (solicitudes de extracción) y seleccionamos la solicitud que deseamos aceptar. Chequeamos los cambios propuestos, los archivos modificados. Luego podemos dejar comentarios y aprobar la solicitud. Podemos solicitar otros cambios.
Para aceptar la solicitud click en merge pull request (fusionar solicitud de extracción). Luego seleccionamos create a merge commit (crear commit de fusión). Finalmente click en confirm merge (confirmar fusión).

¿Qué es una etiqueta en Git?

Una etiqueta o tag es una referencia que se usa para marcar puntos específicos en la historia del proyecto. A diferencia de las ramas, que continuan evolucionando, las etiquetas son fijas. Util para señalar versiones estables o hitos importantes del proyecto. Existen dos tipos de etiquetas:
_Etiquetas ligeras: es un puntero a un commit específico. No tienen fecha o autor.
_Etiquetas anotadas: más completas y almacenan información adicional, como nombre del autor, fecha y un mensaje.

¿Cómo crear una etiqueta en Git?

Para crear una etiqueta ligera utilizamos el comando git tag <nombre de etiqueta>.
Para una etiqueta anotada el comando git tag nombre de etiqueta -m “mensaje”
Para ver todas las etiquetas el comando git tag
Para ver detalles de una etiqueta el comando git show <nombre de etiqueta>
Para subir una etiqueta al repositorio remoto con el comando git push origin <nombre de etiqueta>
Para subir todas las etiquetas con el comando git push - -tags

¿Cómo enviar una etiqueta a GitHub?

Una vez creada una etiqueta en nuestro repositorio local, para subirla a GitHub lo hacemos de la siguiente manera:
_Para enviar una sola etiqueta utilizamos el **comando git push origin** <nombre de etiqueta>
_Para enviar todas las etiquetas usamos el **comando git push origin - -tags**
Podemos verificar la etiqueta subida en la sección ‘releases’ o ‘tags’

¿Qué es un historial de Git?

Es un registro de todos los cambios (commits) realizados en un repositorio a lo largo del tiempo. Este historial es fundamental para el control de versiones, permite revisar el progreso, deshacer cambios, comparar versiones o restaurar código a estados anteriores. El historial incluye commits, ramas, etiquetas, y fusión de ramas

¿Cómo ver el historial de Git?

Para verlo usamos el **comando git log** Este comando muestra una lista de todos los commits realizados, comenzando por el más reciente. También el hash del commit, autor, fecha y mensaje.

¿Cómo buscar en el historial de Git?

Tenemos varios comandos, depende de lo que busquemos
_Para buscar una palabra o frase en los mensajes del commit usamos el comando git log - -grep=“palabra”
_Para buscar los commits por autor con el comando git log- -author=“nombre del autor”
_Para buscar por fechas usamos el comando git log - -since=“fecha” - -until=“fecha”

_Para buscar una cambio específico en los archivos del repositorio con el comando git log -S“texto a buscar”
_Para buscar un commit específico podemos usar el comando git show <commit-hash>

¿Cómo borrar el historial de Git?

Para borrar todo el historial de Git y comenzar desde cero (manteniendo archivos actuales),
_Creamos un nuevo repositorio vacío:
comando rm -rf .git (borra la carpeta .git (el historial y configuración del repositorio)
comando git init (vuelve a inicializar el repositorio)
_Agregar y confirmar archivos actuales:
comando git add .
comando git commit -m “commit sin historial”
_Volver a agregar el repositorio remoto (si es necesario):
comando git remote add origin <URL del repositorio>
comando push -u origin master - -force (sobreescribe esto el historial en el remoto)

Para borrar commits específicos del historial (con git rebase o filter-branch). Cuando no queremos borrar todo el historial, sino ciertos commits o cambiar los últimos:
comando git rebase -i Head~n (n numero de commits a modificar).
Cambiar ‘pick’ por ‘drop’ en los commits que deseamos eliminar o realizar cambios como ‘edit’ para modificar un commit específico.

Para eliminar un archivo o historia específica utilizando el comando git filter-branch:
Git filter-branch - -force - -index-filter \ “git rm - -cached - -ignore-unmatch <archivo>” \ - -prune-empty - -tag-name-filter cat - - - -all (elimina todo el historial de todos los commits)
Forzar el push (sobreescribiendo el historial remoto):
git push origin - -force - -all
git push origin - -force tags

Para borrar el historial de un archivo específico (con git filter-repo
Instalamos git filter-repo:
pip install git-filter-repo
Eliminamos el archivo del historial:
git filter-repo - -path <archivo> - -invert-paths
Hacemos un push forzado:
git push origin - -force - -all
git push origin - -force - -tags

¿Qué es un repositorio privado en GitHub?

Tipo de repositorio que solo las personas con acceso explícito pueden ver, clonar o contribuir en el código y archivos almacenados.

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

Dentro de GitHub, en el menú de la izquierda seleccionamos ‘manage access’ (gestionar acceso). Luego click en ‘invite a collaborator’ (invitar a un colaborador). Luego escribimos el nombre de usuario a la que queremos invitar. Después seleccionar usuario con click ‘add’ o ‘invite’

¿Qué es un repositorio público en GitHub?

Es un espacio donde se almacena proyectos, documentación u otros archivos. Es accesible para cualquier persona. Al ser públicos los usuarios pueden ver, clonar, bifurcar (fork) y contribuir al repositorio.

¿Cómo crear un repositorio público en GitHub?

Dentro de la cuenta hacemos click en el ícono ‘+’ en la esquina superior derecha. Seleccionamos ‘new repository’
Ahora lo configuramos:
- Repository name (nombre repositorio)
- Description (descripción) Opcional. Para añadir descripción del proyecto.
- **Public**, para que el repositorio sea público.

Luego click en ‘create repository’

¿Cómo compartir un repositorio público en GitHub?

Abrimos el repositorio en GitHub
Copiamos la URL de nuestro repositorio
Compartimos el enlace