

Actividades

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

- ¿Qué es GitHub? **GitHub, es la plataforma web, que se encarga de alojar versionados de repositorios creados con Git**
- ¿Cómo crear un repositorio en GitHub? **Ya habiéndonos logeado, se hace click en el botón '+', situado en la esquina superior derecha, y luego se selecciona la opción 'New Repository'. Finalmente se llenan los datos requeridos y el repositorio quedará creado.**
- ¿Cómo crear una rama en Git? **Utilizando el comando 'git branch x', donde x será el nombre de la rama. Ej: 'git branch tp2-feature'**
- ¿Cómo cambiar a una rama en Git? **Si estamos posicionados en la rama a la que queremos cambiar el nombre, utilizamos el comando 'git branch -m y' donde y será el nuevo nombre. Ahora si estamos posicionados, por ejemplo, en la rama 'main' o 'master' y queremos cambiar el nombre a otra rama, utilizamos 'git branch -m x y' donde x es el nombre anterior, e y el nombre nuevo.**
- ¿Cómo fusionar ramas en Git? **Primero, debemos posicionarnos a la rama que queremos incluir utilizando el comando 'git checkout x' donde x será el nombre de la rama, y luego utilizamos 'git merge y' donde y será la rama destino a donde fusionaremos la rama donde estamos parados.**
- ¿Cómo crear un commit en Git? **Utilizando el comando 'git commit -m x', donde x será el mensaje que representará a nuestro commit. (El commit lo realizamos luego de incluir los archivos requeridos utilizando 'git add .' o bien, 'git add x' donde x es el nombre del archivo a incluir.**
- ¿Cómo enviar un commit a GitHub? **Utilizando el comando 'git push origin x', donde x es el nombre de la branch. Usualmente, haríamos push de la rama 'main' (aunque hay casos en donde también hacemos push de otras ramas)**
- ¿Qué es un repositorio remoto? **Un repositorio remoto, es una copia remota de un repositorio local, pero alojada en un servidor online.**
- ¿Cómo agregar un repositorio remoto a Git? **Utilizamos el comando 'git clone x', donde x corresponde a la url proporcionada por GitHub de nuestro repositorio de interés.**
- ¿Cómo empujar cambios a un repositorio remoto? **Primero utilizando 'git add' los archivos que queremos incluir en nuestro commit, luego procedemos a hacer un commit con 'git commit -m' y un mensaje al final de este, detallando el motivo del commit, y finalmente utilizamos 'git push'**
- ¿Cómo tirar de cambios de un repositorio remoto? **Utilizando el comando 'git pull', que se encarga de traernos el repositorio remoto a nuestro repositorio local y fusionarlo automáticamente, sino utilizamos 'git fetch' que se encargará de traer el repositorio sin fusionarlo, para que podamos analizarlo y definir si queremos o no aplicar un merge.**
- ¿Qué es un fork de repositorio? **Un 'fork', corresponde a una copia de un repositorio remoto, a otro repositorio remoto.**
- ¿Cómo crear un fork de un repositorio? **Primero debemos abrir el repositorio al que queremos crear un 'fork', para luego hacerle click en el botón que se encuentra en la esquina superior derecha, que dice 'fork', y automáticamente nos dirigirá a la vista de**

creación de 'fork'. Llenamos los datos requeridos y quedará creado como un nuevo repositorio.

- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio? Se realizan los siguientes pasos: se crea un fork, hacemos las modificaciones pertinentes, hacemos un commit de nuestros cambios, en GitHub, tocamos el apartado de 'pull request', y por último esperamos la aprobación del propietario del repositorio original.

- ¿Cómo aceptar una solicitud de extracción? En el apartado de pull requests, seleccionamos la pull request que queremos incluir, analizaremos los cambios pertinentes a la solicitud y finalmente se incluyen a la rama principal del repositorio.

- ¿Qué es una etiqueta en Git? Una etiqueta en Git sirve para poder hacer una diferenciación de commits. De forma, que podemos tener una separación más ordenada de las actualizaciones que subimos a nuestros repositorios

- ¿Cómo crear una etiqueta en Git? Utilizando el comando 'git tag x', donde x será el valor del tag (ej: v1.0.0). También podemos utilizar 'git tag -a x -m y', donde x será también el valor del tag, e y, será el valor del mensaje que queremos dejar en ese tag a modo de complemento informativo.

- ¿Cómo enviar una etiqueta a GitHub? Utilizando el comando 'git push --tags'

- ¿Qué es un historial de Git? El historial de Git, también se conoce como 'Log'. En el 'Log' tenemos toda la información pertinente a los commits realizados, desde quien lo hizo, la hora, el tag, el mensaje y el valor del código asignado a ese commit.

- ¿Cómo ver el historial de Git? Utilizando el comando 'git log'. En la documentación tenemos otros ejemplos de este comando, en donde nosotros podemos seleccionar que queremos ver del log. Por ejemplo, si solo queremos ver los últimos 5 cambios, podemos utilizar 'git log -n 5'.

- ¿Cómo buscar en el historial de Git? Se puede realizar búsquedas por los atributos de los commits, que pueden ser los mencionados anteriormente. Ahora, nosotros podemos seleccionarlos utilizando palabras específicas en el comando del log. Por ejemplo: 'git log --prep="x"', donde x será la palabra específica que buscamos. Lo mismo aplica para '--author', en donde filtraremos por el autor especificado.

- ¿Cómo borrar el historial de Git? Si queremos eliminar los commits recientes, podemos utilizar el comando 'git reset --hard/soft HEAD~x' donde x es la cantidad de commits a eliminar, y podemos elegir entre 'hard' y 'soft'. En caso de elegir 'hard' volvemos 'al pasado' 3 commits, perdiendo todos los avances hasta ese commit. En cambio, con 'soft', 'volvemos al pasado', pero mantenemos los cambios y podemos volver a aplicarlos.

- ¿Qué es un repositorio privado en GitHub? Un repositorio privado, es un repositorio al que solo las personas a las que el propietario asigne permiso, podrán ver el contenido de este. El resto de interacciones con este tipo de repositorio están bloqueados y reservados como se mencionó, a las personas habilitadas.

- ¿Cómo crear un repositorio privado en GitHub? Para establecer un repositorio como privado, basta con establecerlo como 'Private' al momento de crear el repositorio, o bien, si tenemos un repositorio creado y queremos que sea privado, ingresamos al repositorio, luego a sus 'settings' y finalmente, en el apartado de 'Danger Zone' damos a 'Change visibility', estableciéndolo así como privado.

- ¿Cómo invitar a alguien a un repositorio privado en GitHub? En el apartado de 'Settings' de nuestro repositorio privado, vamos al apartado de 'Colaboradores' y ahí, ingresamos el username de GitHub de la persona a quien queremos dar acceso.

- ¿Qué es un repositorio público en GitHub? Un repositorio público, es un repositorio a quien cualquier persona puede ver, interactuar y manipular el código de forma local, sin embargo, para modificar el código a nivel repositorio, es necesario pasar por un 'fork' y su respectivo 'pull request'.

- ¿Cómo crear un repositorio público en GitHub? Para crear un repositorio público, basta con crear un repositorio como ya explicamos previamente, y dejamos por defecto 'Public' como tipo de visibilidad del repositorio

- ¿Cómo compartir un repositorio público en GitHub? Cuando nosotros abrimos nuestro repositorio, hay un botón verde, que se llama '<> Code', cuando le hacemos click, se nos despliega un subcuadro que nos muestra la URL del repositorio. De todas formas, podemos simplemente copiar y compartir la URL que tenemos en la barra de búsqueda del explorador al momento de ingresar al repositorio.

2) Crear un repositorio, agregar un archivo, crear y hacer push a una rama a todo el conjunto, dentro del repositorio en GitHub.

<https://github.com/BrunoMorales23/TPN2-UTN>

3) Crear un repositorio, clonarlo desde GitHub, crear una nueva rama, modificar archivo README.md, cambiar a la rama principal, modificar una vez más el archivo README.md, hacer merge de la rama principal con la creada anteriormente, resolver el problema y hacer push de todo el conjunto dentro del repositorio en GitHub.

<https://github.com/BrunoMorales23/TPN2-UTN-pt2>