

## Práctico 2: Git y GitHub

Héctor Romero

### Actividades

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

- ¿Qué es GitHub?: Es una plataforma utilizada como repositorio de código, que permite el desarrollo colaborativo de proyectos utilizando el control de versiones Git.
- ¿Cómo crear un repositorio en GitHub? Primero: crear una cuenta en GitHub; Segundo: Crear un nuevo repositorio en la plataforma desde el menú desplegable (Nuevo repositorio); Tercero: Configurar, nombre, descripción, visibilidad (privado o público) e inicializarlo con un archivo. Cuarto: Configurar el repositorio LOCALMENTE clonando un repositorio o agregando archivos; y por último Quinto: Sincronizar el repositorio local con GitHub.
- ¿Cómo crear una rama en Git? Para crear una rama en Git debemos utilizar el comando “**git branch** (nombre de la rama)” verificando en que rama nos encontramos.
- ¿Cómo cambiar a una rama en Git? Para cambiar de rama utilizamos el comando “**git checkout** nombre\_de\_la\_rama>”
- ¿Cómo fusionar ramas en Git? Se utiliza el comando “**git merge** <nombre\_de\_la\_rama>”
  - ¿Cómo crear un commit en Git? El commit es fundamental ya que es el comando que registra los cambios en el repositorio local y crea los puntos de referencia para rastrear cambios o revertirlos. El formato es **git commit** <descripción\_del\_cambio>
- ¿Cómo enviar un commit a GitHub? Para que los commit se envíen a GitHub debemos utilizar el comando **git push**
- ¿Qué es un repositorio remoto? Es un servidor remoto donde podemos alojar nuestro repositorio local. Lo que permite compartir y colaborar en los proyectos.
- ¿Cómo agregar un repositorio remoto a Git? Se utiliza el comando **git remote add** especificando la URL del repositorio remoto.
- ¿Cómo empujar cambios a un repositorio remoto? Se debe utilizar el comando **git push**
- ¿Cómo tirar de cambios de un repositorio remoto? Se debe utilizar el comando **git pull** <nombre\_del\_remoto> <nombre\_de\_la\_rama>
- ¿Qué es un fork de repositorio? Es una copia de un repositorio existente que se crea en la cuenta de un usuario. Permite trabajar en una versión independiente del proyecto original.
- ¿Cómo crear un fork de un repositorio? Primero, iniciar sesión en GitHub; Segundo, buscar el repositorio que deseas copiar (forkear); Tercero, crear el “fork” utilizando el botón Fork y seleccionar la cuenta en la que deseas crearlo.
- ¿Cómo enviar una solicitud de extracción (pull request) a un repositorio? Primero realizar los cambios en el Fork; Segundo, ir al repositorio original y en GitHub y utilizar “New pull request”; Tercero, seleccionar la rama y utilizar el botón Create pull request”
- ¿Cómo aceptar una solicitud de extracción? Utilizando el botón “Merge pull request”
- ¿Qué es una etiqueta en Git? Son puntos de referencia para marcar cambios específicos en la historia de un proyecto.
- ¿Cómo crear una etiqueta en Git? Con el comando **git tag** <nombre\_de\_la\_etiqueta>

- ¿Cómo enviar una etiqueta a GitHub? Con el comando **git push --tags** envía todas las etiquetas; con el comando **git push origin <nombre\_de\_la\_etiqueta>** envía una etiqueta específica.
- ¿Qué es un historial de Git? El historial de git es un registro de los cambios realizados en un proyecto, incluyendo commits, ramas, fusiones etiquetas. Permite entender la evolución de un proyecto, la colaboración entre los desarrolladores y la resolución de problemas.
- ¿Cómo ver el historial de Git? Para ver el historial de git podemos utilizar los comandos **git log** (muestra el historial de un proyecto); **git log --graph** (Muestra el historial de commites en gráficos); **git log --oneline** (muestra el historial como una sola línea por commit)
- ¿Cómo buscar en el historial de Git?:
 

Buscar commits por mensaje: <b>git log --grep</b>	Ejemplo: <b>git log --grep="fixbug"</b>
Buscar commits por autor: <b>git log --author</b>	Ejemplo: <b>git log --author="Hector Romero"</b>
Buscar commits por fecha: <b>git log --until</b>	Ejemplo: <b>git log --until="2025-04-18"</b>
Buscar commits por fecha: <b>git log --since</b>	Ejemplo; <b>git log --since="1 week"</b>
Buscar archivos: <b>git log --archivo.txt</b>	
Buscar cambios en un archivo: <b>git log -p archivo.txt</b>	
Buscar commits que hayan introducido o eliminado una línea: <b>git log --"función específica"</b>	
- ¿Cómo borrar el historial de Git?:
  - git reset --hard** (reestablece el repositorio a un estado anterior eliminando todos los commits posteriores.
  - git filter -branch** (reescribe el historial de git, permitiendo eliminar o modificar commits específicos.
  - .git** en la raíz el proyecto eliminara todo el historial
- ¿Qué es un repositorio privado en GitHub? Es un tipo de repositorio que solo es accesible para usuarios que han sido invitados o tienen permiso para ver y contribuir al proyecto. Son de acceso restringido, tienen visibilidad limitada y tiene control de acceso.
- ¿Cómo crear un repositorio privado en GitHub? Dentro de la sesión de la cuenta de Gitub, crear nuevo repositorio (nombre del repositorio, descripción) y visibilidad privada. Configurar el acceso invitando colaboradores y configurando permisos.
- ¿Cómo invitar a alguien a un repositorio privado en GitHub? En el repositorio al cual deseamos invitar, ir a *settings* y luego *collaborators*. Agregar colaborador con *add collaborators* ingresando el nombre de usuarios o su dirección de mail. Seleccionar nivel de colaboración y enviar imitación con *add*.
- ¿Qué es un repositorio público en GitHub? Los repositorios visible y accesible para cualquier persona en internet.
- ¿Cómo crear un repositorio público en GitHub? Dentro de la sesión de la cuenta de Gitub, crear nuevo repositorio (nombre del repositorio, descripción) y visibilidad pública.
- ¿Cómo compartir un repositorio público en GitHub? Compartiendo el enlace directo o la URL.

Actividad 2: <https://github.com/Romero-Hector/tp1-repo>

Actividad 3: <https://github.com/Romero-Hector/tp-repo-1.git>