

## TP 2 Programación (GIT y GITHUB) – Brian Zapata Marin

- **¿Qué es GitHub?**
  - Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones<sup>1</sup> Git. Permite a múltiples personas trabajar juntas en el mismo código.
- **¿Cómo crear un repositorio en GitHub?**
  - Se crea desde la interfaz web de GitHub, haciendo clic en el botón "New repository" en tu perfil o en la página de repositorios. Luego se configura el nombre y la visibilidad.
- **¿Cómo crear una rama en Git?**
  - Se utiliza el comando `git branch <nombre_de_la_rama>` para crear una nueva rama localmente. Esto crea una copia del estado actual del código.
- **¿Cómo cambiar a una rama en Git?**
  - Se utiliza el comando `git checkout <nombre_de_la_rama>` para mover el directorio de trabajo y el HEAD a la rama especificada. Esto permite trabajar en esa rama.
- **¿Cómo fusionar ramas en Git?**
  - Se utiliza el comando `git merge <nombre_de_la_rama_a_fusionar>` mientras estás en la rama donde quieres integrar los cambios. Esto combina el historial de ambas ramas.
- **¿Cómo crear un commit en Git?**
  - Primero se deben agregar los cambios al área de staging con `git add .` o archivos específicos, y luego se ejecuta `git commit -m "Mensaje descriptivo"`. Esto guarda los cambios.
- **¿Cómo enviar un commit a GitHub?**
  - Se utiliza el comando `git push origin <nombre_de_la_rama>` para subir los commits de tu rama local al repositorio remoto en GitHub.
- **¿Qué es un repositorio remoto?**
  - Es una versión del repositorio que está alojada en un servidor, como GitHub, y que permite la colaboración entre diferentes desarrolladores.
- **¿Cómo agregar un repositorio remoto a Git?**
  - Se utiliza el comando `git remote add origin <URL_del_repositorio_remoto>` para establecer una conexión con el repositorio en GitHub. 'origin' es un alias común para el repositorio principal.
- **¿Cómo empujar cambios a un repositorio remoto?**
  - Se utiliza el comando `git push <nombre_del_remoto> <nombre_de_la_rama>`, comúnmente `git push origin main`. Esto envía tus commits locales a la rama especificada del repositorio remoto.
- **¿Cómo tirar de cambios de un repositorio remoto?**
  - Se utiliza el comando `git pull <nombre_del_remoto> <nombre_de_la_rama>`, comúnmente `git pull origin main`. Esto descarga los cambios del repositorio remoto y los fusiona con tu rama local.
- **¿Qué es un fork de repositorio?**
  - Es una copia personal de un repositorio que reside en tu propia cuenta de GitHub. Permite realizar cambios sin afectar el repositorio original.
- **¿Cómo crear un fork de un repositorio?**
  - Se realiza desde la interfaz web de GitHub, haciendo clic en el botón "Fork" en la página del repositorio que deseas copiar. Esto crea una copia en tu cuenta.
- **¿Cómo enviar una solicitud de extracción (pull request) a un repositorio?**
  - Después de hacer un fork y realizar cambios, se crea una "Pull Request" desde la interfaz de GitHub para proponer tus cambios al repositorio original.
- **¿Cómo aceptar una solicitud de extracción?**
  - Los mantenedores del repositorio original revisan la solicitud y, si están de acuerdo, pueden fusionar los cambios a la rama principal a través de la interfaz de GitHub.
- **¿Qué es una etiqueta en Git?**
  - Es un punto de referencia estático en el historial de Git, utilizado típicamente para marcar lanzamientos de versiones (por ejemplo v1.0 o v1.0.1).
- **¿Cómo crear una etiqueta en Git?**

- Se utiliza el comando `git tag <nombre_de_la_etiqueta>` para crear una etiqueta ligera en el commit actual, o `git tag -a <nombre_de_la_etiqueta> -m "Mensaje"` para una etiqueta anotada.
- **¿Cómo enviar una etiqueta a GitHub?**
  - Se utiliza el comando `git push origin <nombre_de_la_etiqueta>` para enviar una etiqueta específica, o `git push origin --tags` para enviar todas las etiquetas locales al repositorio remoto.
- **¿Qué es un historial de Git?**
  - Es un registro de todos los commits realizados en un repositorio, mostrando cuándo se hicieron los cambios, quién los hizo y qué cambios se realizaron.
- **¿Cómo ver el historial de Git?**
  - Se utiliza el comando `git log` para mostrar una lista de los commits, con información como el autor, la fecha y el mensaje. Existen varias opciones para personalizar la salida.
- **¿Cómo buscar en el historial de Git?**
  - Se pueden usar opciones con `git log`, como `--grep="<patrón>"` para buscar commits por su mensaje, o `--author="<autor>"` para buscar por autor.
- **¿Cómo borrar el historial de Git?**
  - Con el comando `git reset` (quita del stage todos los archivos y carpetas del proyecto)
- **¿Qué es un repositorio privado en GitHub?**
  - Es un repositorio cuyo contenido solo es visible para el propietario y las personas a las que explícitamente se les ha concedido acceso.
- **¿Cómo crear un repositorio privado en GitHub?**
  - Al crear un nuevo repositorio en GitHub, se selecciona la opción "Private" en la configuración de visibilidad. Esto asegura que solo los colaboradores invitados puedan verlo.
- **¿Cómo invitar a alguien a un repositorio privado en GitHub?**
  - Desde la página del repositorio privado, se va a la sección "Settings" (Configuración) y luego a "Collaborators" (Colaboradores), donde se pueden invitar usuarios de GitHub por su nombre de usuario o correo electrónico.
- **¿Qué es un repositorio público en GitHub?**
  - Es un repositorio cuyo contenido es visible para cualquier persona en GitHub. Permite la colaboración abierta y la transparencia del código.
- **¿Cómo crear un repositorio público en GitHub?**
  - Al crear un nuevo repositorio en GitHub, se selecciona la opción "Public" en la configuración de visibilidad. Esto hace que el repositorio sea accesible para todos.
- **¿Cómo compartir un repositorio público en GitHub?**
  - Se comparte la URL del repositorio. Cualquier persona puede ver el código, clonarlo y crear forks.

**EN MI REPO PODRA VER QUE CUMPLO CON EL RESTO DE ACTIVIDADES DE ESTE TP-2 DE PROGRAMACION**

**<https://github.com/BrianEZM>**