

## NATALIA DIAZ - COMISION 7

### Práctico 2: Git y GitHub

#### Objetivo:

El estudiante desarrollará competencias para trabajar con Git y GitHub, aplicando conceptos fundamentales de control de versiones, colaboración en proyectos y resolución de conflictos, en un entorno simulado y guiado.

#### Resultados de aprendizaje:

1. Comprender los conceptos básicos de Git y GitHub: Identificar y explicar los principales términos y procesos asociados con Git y GitHub, como repositorios, ramas, commits, forks, etiquetas y repositorios remotos.
2. Manejar comandos esenciales de Git: Ejecutar comandos básicos para crear, modificar, fusionar y gestionar ramas, commits y repositorios, tanto en local como en remoto.
3. Aplicar técnicas de colaboración en GitHub: Configurar y utilizar repositorios remotos, realizar forks, y gestionar pull requests para facilitar el trabajo colaborativo.
4. Resolver conflictos en un entorno de control de versiones: Identificar, analizar y solucionar conflictos de merge generados en un flujo de trabajo con múltiples ramas.

#### Actividades

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

- **¿Qué es GitHub?**

GitHub es una plataforma virtual que nos permite compartir repositorios remotos de manera publica y/o privada con otras personas.

Permite almacenar y gestionar proyectos, así como también, trabajar en conjunto sobre esos proyectos u otros con demás personas.

- **¿Cómo crear un repositorio en GitHub?**

- 1- Crear una cuenta en GitHub
- 2- Click en Repositories -> New
- 3- Colocarle nombre, descripción y elegir si será publico o privado
- 4- Click en Create Repository
- 5- Nos aparecerán distintas opciones de comandos, para poder asociar el repositorio local que ya hemos creado a la plataforma de GitHub.
- 6- Copiamos y pegamos el comando en el bash:  
\$ git remote add origin https://github.com/tucanal/nombre-repo.git  
\$ git push -u origin master

- **¿Cómo crear una rama en Git?**

Para crear una nueva rama en Git, usaremos el comando git Branch:

>git Branch nuevarama.

Para visualizar las ramas que tenemos creadas, usaremos el comando >git Branch.

Arecera un listado de las reamas creadas y un \* en la rama en la cual estamos posicionados.

- **¿Cómo cambiar a una rama en Git?**

Para cambiar a una nueva rama, usaremos en comando >git checkout nuevarama

- **¿Cómo fusionar ramas en Git?**

Para fusionar una nueva rama dentro de la rama principal ramas, debemos estar ubicados en la rama principal y usaremos el comando >git merge nuevarama

- **¿Cómo crear un commit en Git?**

Un commit se utiliza para guardar los cambios del proyecto en el repositorio.

Para hacer un commior debemos , primero usar el comando >git add . para cargar los archivos modificados.

Luego usar el comando >git commit -m "Especificación de los cambios realizados"

Así, los cambios quedan guardados en el repisitorio.

- **¿Cómo enviar un commit a GitHub?**

Una vez creado el commit en nuestro repositorio locar, debemos enviar los commits al repositorio en GitHub con el comando >git push origin nombreproyecto

- **¿Qué es un repositorio remoto?**

Es una versión de tu proyecto que está en línea, al cual puedes acceder desde cualquier computador. También puede ser publico y visible por demás personas.

- **¿Cómo agregar un repositorio remoto a Git?**

Para agregar un repositorio repoto a git debemos usar el comando >git remote add nombreelegido url

Con el comando >git fetch nombredelrepositorioremoto, traemos toda la información del repositorio remoto a tu repositorio local.

- **¿Cómo empujar cambios a un repositorio remoto?**

Para empujar los cambios a un repositorio remoto usamos el comando >git push origin nombredelproyecto

- **¿Cómo tirar de cambios de un repositorio remoto?**

Tirar de cambios significa fusionar los cambios de un repositorio remoto con mi repositorio local.

Para realizarlo debo asegurarme de que este posicionado en la rama de mi repositorio usando el comando >git checkout nombredelarama y luego, usaremos el comando >git pull origin nombredelarama en el repositorio remoto que queremos mover al local.

- **¿Qué es un fork de repositorio?**

Un fork es una copia de un repositorio ya existente, que nos permite realizarle modificaciones sin afectar el original. No se descarga en el repositorio local como un clonado, sino que genera una copia en la cuenta del usuario.

- **¿Cómo crear un fork de un repositorio?**

Debemos ir a GitHub, al repositorio al cual deseamos hacerle un fork, haremos click en el botón “Fork”. Podemos cambiar el nombre y la descripción y luego hacer click en “Create Fork”. Podemos editarlo en nuestro repositorio local con algún cambio que nos pareciese necesario, y luego con el comando `>git push origin master` mandamos los cambios al fork que hicimos.

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

Iremos a la solapa de pull request y haremos click en new pull request. Podemos explicar cuales son los cambios que realizamos y luego clicar en Create pull request.

- **¿Cómo aceptar una solicitud de extracción?**

Nos llegara un mensaje con los cambios que a realizado y si consideramos que es una buena modificación podemos usar el comando `>merge pull request` y fusionar los cambios del repositorio de la otra persona en el repositorio original.

- **¿Qué es una etiqueta en Git?**

Una etiqueta en Git es un marcador que se utiliza para señalar puntos específicos en los commits, generalmente usado para identificar versiones importantes del proyecto.

- **¿Cómo crear una etiqueta en Git?**

Podemos crear etiquetas ligeras o lightweight, las cuales solo son referencias sin información adicional, usando el comando `>git tag etiqueta`  
Otra opción es crear etiquetas anotadas o Annotated, las cuales incluyen nombre del autor, fecha y un mensaje de anotación, usando el comando `>git tag -a etiqueta -m “descripcion”`

- **¿Cómo enviar una etiqueta a GitHub?**

Una vez creada la etiqueta en tu repositorio local, la empujas a github con el comando `>git push origin etiqueta` o para enviar todas las etiquetas juntas, usamos el comando `>git push origin --tags`

- **¿Qué es un historial de Git?**

Es un registro de todos los cambios realizados en un repositorio, organizados en una secuencia de commits, muestra los cambios realizados, quien los realizo y la fecha en que se realizaron.

- **¿Cómo ver el historial de Git?**

Usando el comando `>git log`

- **¿Cómo buscar en el historial de Git?**

Según lo que se necesite encontrar, podemos usar distintos comandos:

- Para buscar commits que tengan una palabra específica, usamos el comando `>git log --grep="palabraespecifica"`
- Para buscar commits por autor usamos el comando `>git log --author="Nombredelautor"`
- Para buscar commits en un archivo específico usamos el comando `>git log --nombearchivo`
- Para buscar commits en un rango de fechas específico, usamos los comandos `>git log --since="000-00-00" --until="0000-00-00"`

- **¿Cómo borrar el historial de Git?**

Para borrar el historial se utiliza el comando `git reset` y hay varias formas de utilizarlo:

Para borrar todos los archivos y carpetas del proyecto usamos el comando `>git reset`

Para borrar un archivo en particular usamos el comando `>git reset nombredelarchivo`

Para borrar todos los archivos de una carpeta usamos el comando `<git reset nombrecarpeta/`

- **¿Qué es un repositorio privado en GitHub?**

Un repositorio privado es un repositorio al cual solo puede acceder el propietario y las personas a las cuales el propietario les da acceso.

- **¿Cómo crear un repositorio privado en GitHub?**

Al crear un repositorio en GitHub, la página te da la opción de elegir si queremos que el repositorio sea público o privado, debemos seleccionar en la opción "Private".

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

Una vez dentro del repositorio, iremos a la opción "Settings" y luego a "Collaborators".

Clickeamos en "Add people" e ingresaremos el usuario de las personas a las cuales queremos invitar.

Selecciona el nivel de acceso que deseas otorgar: Read, Triage, Write, Maintain, o Admin. Haz clic en el botón "Add" para enviar la invitación.

- **¿Qué es un repositorio público en GitHub?**

Un repositorio público en GitHub es un repositorio visible y accesible para cualquier persona de internet.

- **¿Cómo crear un repositorio público en GitHub?**

Al crear un repositorio en GitHub, la página te da la opción de elegir si queremos que el repositorio sea público o privado, debemos seleccionar en la opción "Public".

- **¿Cómo compartir un repositorio público en GitHub?**

Para compartir un repositorio, el mismo te proporcionará un enlace directo que puedes copiar desde la opción "<>Code".