**Trabajo Práctico: Git y GitHub**

**Renzo Di Laudo – Comisión 3**

**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 para el control de versiones y la colaboración en proyectos de desarrollo de software. Utiliza Git como su sistema de control de versiones y permite a los desarrolladores almacenar, compartir, y colaborar en proyectos de programación, ya sea de forma pública o privada.

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

Para crear un repositorio en GitHub se deben seguir los siguientes pasos:

1. Inicia sesión en tu cuenta de GitHub.
2. Haz clic en el botón “+” en la esquina superior derecha y selecciona “Nuevo repositorio”.
3. Introduce el nombre del repositorio, una descripción opcional y selecciona si será público o privado.
4. Puedes inicializarlo con un archivo README y elegir una licencia.
5. Haz clic en “Crear repositorio”.

* **¿Cómo crear una rama en Git?**

Para crear una rama en GitHub se deben seguir los siguientes pasos:

1. Abre tu terminal o CLI.
2. Navega al directorio de tu proyecto local.
3. Usa el comando: git branch nombre\_de\_la\_rama. Esto crea una nueva rama.
4. Para cambiar a esta rama, usa: git checkout nombre\_de\_la\_rama.

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

Utilizando el siguiete comando: git checkout nombre\_de\_la\_rama

* **¿Cómo fusionar ramas en Git?**

Primero se cambia la rama que recibirá los cambios con el siguiente comando:

git checkout nombre\_de\_la\_rama\_principal

Luego se fusiona la otra rama con el comando:

git merge nombre\_de\_la\_rama\_a\_fusionar

* **¿Cómo crear un commit en Git?**

Primero se agregan los cambios en el archivo:

git add nombre\_del\_archivo

Luego se ejecuta el comando commit:

git commit -m "Mensaje descriptivo del cambio"

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

Antes de enviar el commit es necesario haber configurado un repositorio remoto:

git remote add origin URL\_del\_repositorio

Luego se envía el commit con:

git push origin nombre\_de\_la\_rama

* **¿Qué es un repositorio remoto?**

Es una copia de un repositorio que está almacenada en un servidor en línea (como GitHub), lo que permite compartir y colaborar en proyectos.

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

Se agrega inicializando un repositorio con: git init

Y luego se agrega con:

git remote add origin URL\_del\_repositorio

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

Utilizando el comando:

git push origin nombre\_de\_la\_rama

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

Utilizando el siguiente comando:

git pull origin nombre\_de\_la\_rama

* **¿Qué es un historial de Git?**

Es el registro de todos los cambios y commits realizados en el repositorio. Muestra quién hizo cada cambio y cuándo.

* **¿Cómo ver el historial de Git?**

Utilizando el comando git log

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

Se pueden buscar términos específicos con el siguiente comando:

git log --grep="texto a buscar"

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

No es posible borrar el historial completamente desde Git, si es posible reiniciar el repositorio eliminando el archivo .git y creando uno nuevo aunque esto no se recomienda porque se pierden todos los registros.

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

Es un repositorio accesible solo por quienes tengan permiso. Es ideal para proyectos que no quieres compartir públicamente.

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

Al crear un repositorio, selecciona la opción “Privado” en el formulario de configuración.

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

**Para invitar a alguien a un repositorio privado hay que:**

1. Ir a la configuración del repositorio.
2. Hacer clic en “Colaboradores”.
3. Ingresar el nombre de usuario del colaborador y envía la invitación.

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

Es un repositorio que cualquiera puede ver y clonar, aunque solo los usuarios autorizados pueden realizar cambios.

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

Seleccionando la opción “Público” al crear el repositorio.

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

Compartiendo la URL del repositorio que aparece en la barra de dirección del navegador.

2) Realizar la siguiente actividad:

* Crear un repositorio.
  + Dale un nombre al repositorio.
  + Elije el repositorio sea público.
  + Inicializa el repositorio con un archivo.
* Agregando un Archivo
  + Crea un archivo simple, por ejemplo, "mi-archivo.txt".
  + Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.
  + Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).
* Creando Branchs
  + Crear una Branch
  + Realizar cambios o agregar un archivo
  + Subir la Branch