**TRABAJO PRÁCTICO N°2: Git & GitHub**

**Martín Cicala**

**1-**

* ***¿Qué es GitHub?***

Es una plataforma en línea que se utiliza principalmente para la gestión de versiones y colaboración en proyectos de desarrollo de software. Es un entorno que permite almacenar, gestionar y compartir código fuente de manera eficiente.

* ***¿Cuáles son los beneficios de usar GitHub?***

Algunos ejemplos de sus beneficios son:

* Control de versiones
* Colaboración eficiente
* Seguimiento de problemas y tareas
* Facilitar la revisión de código
* Incluir documentación
* Acceso remoto
* Portafolio personal
* ***¿Cómo crear un repositorio en GitHub?***

1. Iniciar sesión
2. Ir al perfil
3. En la pestaña *“Your repositories”*  clickear en *“New”*
4. Configurar el nuevo repositorio:

* Owner (Si el repositorio será de propiedad personal o de una organización)
* Nombre del repositorio
* Descripción
* Definir si el repositorio será de dominio público o privado
* Definir si el repositorio iniciara con un README
* Puede agregarse un archivo .gitignore
* Puede agregarse una licencia

1. Crear repositorio

* ***¿Cuáles son los tipos de ramas en Git?***

1. Rama principal
2. Ramas de características
3. Ramas de corrección de errores
4. Ramas de publicación
5. Ramas de desarrollo
6. Ramas de prueba
7. Ramas personales

* ***¿Cómo crear una rama en Git?***

1. Colocarse en la rama principal (puede usarse el comando ***‘git checkout main’***
2. Crear una nueva rama usando el comando ***‘git branch’*** seguido del nombre de la nueva rama

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

Usando el comando ***‘git checkout’*** o ***‘git switch’***, ambos seguidos del nombre de la rama a la que se desea cambiar

* ***¿Cómo fusionar ramas en Git?***

1. Colocarse en la rama a la que deseas fusionarle otra
2. Usamos el comando ***‘git merge’*** seguido del nombre de la rama que se desa fusionar

* ***¿Cómo crear un commit en Git?***

1. Agregar los cambio al área de preparación con el comando ***‘git add’***
2. Crear el commit usando el comando ***‘git commit’***, esto abrirá un editor de texto donde se puede ingresar un mensaje descriptivo de los cambios realizados

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

1. Crear un repositorio si no se tiene uno
2. Copiar su URL
3. Vincula en repositorio local con el remoto usando el comando ***‘git remote add origin’*** seguido de la URL del repositorio remoto
4. Enviar los Commits usndo el comando ***‘git push’*** seguido del nombre de la rama que se desea enviar y el nombre del repositorio remoto

* ***¿Qué es un repositorio remoto?***

Es una versión del proyecto que se encuentra en un servidor en línea (como GitHub, por ejemplo)

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

Utilizando el comando ***‘git remote add origin’*** seguido de la URL del repositorio remoto

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

Utilizando el comando ***‘git push’***  seguido del nombre del repositorio remoto y el nombre de la rama local que se desea empujar

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

Utilizando el comando ***‘git pull’*** seguido del nombre del repositorio remoto y el nombre de la rama remota de la que deseas traer los cambios

* ***¿Qué es un fork de repositorio?***

Es la creación de una copia independiente de un repositorio ajeno en una cuenta personal. Esto permite trabajar en versiones propias de un proyecto, realizar cambios y experimentar sin afectar el repositorio original

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

1. Ir al repositorio al que se desea hacer un fork
2. Clickear en el botón que dice *“Fork”*
3. Elegir la cuenta de destino

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

1. Clonar el Fork a la Máquina Local utilizando el comando ***‘git clone’*** seguido del URL del Fork
2. Realizar los cambios que deseas proponer en el Fork de la Máquina Local
3. Empujar los cambios al Fork utilizando el comando ***‘git push’***
4. Ir a la página del repositorio original en GitHub y clickear donde dice *“Compare & pull request”*
5. Comparar cambios y proporcionar detalles sobre los cambios realizados
6. Clickear en *“Create pull request”*

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

1. Abrir la pestaña *“Pull requests”* en GitHub
2. Hacer click en *“Merge pull request”*

* ***¿Qué es una etiqueta en Git?***

Es una forma de marcar un punto específico. Se utilizan para identificar versiones significativas de un proyecto.

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

Utilizando el comando ***‘git tag’*** seguido del nombre que quiera darle a la etiqueta

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

Utilizando el comando ***‘git push’*** seguido del nombre del repositorio remoto y el nombre de la etiqueta

* ***¿Qué es un historial de Git?***

Es una secuencia ordenada commits que representan los cambios realizados en un repositorio a lo largo del tiempo

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

Utilizando el comando ***‘git log’***

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

Utilizando el comando ***‘git log’*** seguido de las opciones de filtrado (como ***‘- -author’, ‘- -grep’, ‘- -since’, ‘- -until’***)

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

Clickeando en la opción de Eliminar disponible en la plataforma donde tengamos el repositorio

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

Es un espacio de almacenamiento de GitHub que permite guardar u administrar código de manera confidencial y protegida

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

1. En la página *“your repositories”* clickear en el botón *“new”*
2. Configurar el repositorio, nombrarlo, opcionalmente puede ponerse una descripción y en la siguiente opción colocar *“Private”*

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

1. Acceder al repositorio
2. Ir a la pestaña “*Settings”*
3. Seleccionar *“Manage Access”* y acceder a la sección para invitar colaboradores
4. Ingresar el nombre de usuario o la dirección de correo electrónico de la persona que se desea invitar

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

Es un espacio de almacenamiento donde se puede guardar y compartir código de manera abierta y accesible para cualquier persona en Internet

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

1. En la página *“your repositories”* clickear en el botón *“new”*
2. Configurar el repositorio, nombrarlo, opcionalmente puede colocarse una descripción y en la siguiente opción colocar *“Public”*

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

Compartiendo la URL del repositorio

* ***¿Qué es un repositorio de código en GitHub?***

Es un espacio virtual donde se puede almacenar, gestiona y colaborar en el desarrollo de un código fuente

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

1. Iniciar sesión en GitHub
2. Seleccionar la opción *Your repositories”*
3. Crear un nuevo repositorio clickeando en la opción *“New”*

* ***¿Cómo compartir un repositorio de código en GitHub?***

En la pestaña *“Settings”* acceder a *“Manage Access”* y luego a *“Invite collaborators”*, ingresar el nombre de usuario o el correo electrónico de aquellos a los que se quiere invitar yy luego elegir el nivel de acceso que tendrán (lectura, escritura, administración)

* ***¿Qué es un repositorio de documentación en GitHub?***

Es un espacio donde se puede almacenar y gestionar la documentación relacionada con un proyecto

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

1. Acceder a GitHub
2. Acceder a *“Your repositories”*
3. Clickear en *“New”*

* ***¿Cómo compartir un repositorio de documentación en GitHub?***

1. Acceder a “*Settings”*
2. Luego a *“Manage Access”*
3. Acceder a *“Invite collaborators”, ingresar el nombre o dirección de correo electrónico de aquellos con quien se quiera compartir*
4. Elegir los niveles de acceso

* ***¿Qué es un repositorio de proyecto en GitHub?***

Es un espacio virtual donde puedes almacenar y gestionar todos los aspectos relacionados con un proyecto específico, no solo el código fuente

* ***¿Cómo usar Git para trabajar con otros?***

1. Crear un repositorio compartido

**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

https://github.com/MEC1998/Laboratorio\_1