**Trabajo practico N2 Laboratorio**

**1)¿Qué es GitHub?**

GitHub es un servicio de alojamiento de repositorios de control de versiones, donde los desarrolladores pueden almacenar sus proyectos de software. Es gratuito y de código abierto, permite a los usuarios trabajar en proyectos en equipo, compartir códigos y colaborar de forma eficiente.

**2)¿Qué beneficios trae GitHub?**

Hay muchos beneficios en usar GitHub para control de versiones y colaboración en equipo.

Las cuales son:

* Fácil uso: Es muy fácil de usar, incluso para quienes no conocen el control de versiones.
* Escalabilidad: Puede escalar para cumplir necesidades de proyectos de diferentes índoles, desde personales hasta industriales.
* Seguridad: GitHub ofrece muchas características de seguridad para proteger cada proyecto.
* Comunidad: Cuenta con una gran comunidad de desarrolladores que pueden ayudar y colaborar.

**3)¿Cómo crear repositorios de GitHub?**

Cuando creas un repositorio, creas una copia de tu código en GitHub. Puedes clonar el repositorio en tu computadora para trabajar en el localmente.

Ya clonado el repositorio, puedes trabajar en el. Puedes hacer cambios en el código y guardarlos como “Commits”.

Puedes enviarlos a los servidores con el comando push, y puedes recibirlos con el comando pull

**4)¿Cuáles son los tipos de ramas creados en GitHub?**

* **Rama local**: La rama local es aquella que habita dentro de tu computadora, donde todos los cambios que hagas se irán guardando hasta que sean subidos. Pero no recibe cambios externos
* **Rama remota:** La rama remota está dentro del sistema de Github, donde todo progreso que se decida “Pushear” será subido, allí se irá guardando todo cambio del archivo realizado por nosotros o otros usuarios. Al “Pullear” la rama se reciben los cambios hechos en la rama remota.

**5)¿Cómo crear una rama en Git?**

Simplemente se crea un nuevo apuntador para que lo puedas mover libremente.

Para ello, usaremos el comando git branch

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

Para saltar de una rama a otra, debemos utilizar el código git checkout.

Este comando realiza dos acciones: nos transporta a una rama y revierte los archivos de tu directorio, dejándolos tal y como estaban en la ultima instantánea de la rama. Esto supone que los cambios que hagas desde este momento en adelante, divergiran de la antigua versión del proyecto, rebobinando el trabajo que habias hecho en la rama, para que puedas avanzar en otra dirección.

**7)¿Cómo fusionar ramas en git?**

El proceso de fusionado se conoce como merge y puede llegar a ser muy simple o más complejo si se encuentran cambios que Git no pueda procesar de manera automática. Git para procesar los merge usa un antecesor común y comprueba los cambios que se han introducido al proyecto desde entonces, combinando el código de ambas ramas.

Situándonos en una rama, escribimos git merge “Nombreotrarama”

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

Después de haber confirmado tus cambios y haber vinculado el repositorio de Git local con el repositorio remoto de GitHub, el siguiente paso que quieres dar es enviar tus cambios o archivos al servidor remoto. git push envía tus commits al repositorio remoto.

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

git push origin “RAMA”

Recordemos que usamos el alias origin para el repositorio remoto.

Cuando creamos un repositorio en GitHub, nos crea una rama por defecto llamada main, podemos en la configuración cambiar para que la rama que se cree por defecto se llame master.

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