

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 él. 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 última instantánea de la rama. Esto supone que los cambios que hagas desde este momento en adelante, divergieron de la antigua versión del proyecto, rebobinando el trabajo que habías hecho en la rama, para que puedas avanzar en otra dirección.

7) ¿Cómo fusionar ramas en git?

El proceso de fusión 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"`

8) ¿Cómo crear un commit en Git?

Para realizar un `commit`, es decir, confirmar los cambios que se encuentran en el `stage` área para que pasen al repositorio local, existen varios comandos, el más utilizado es `"git commit -m "mensaje" "`, donde se manda un mensaje para identificar fácilmente qué se hizo en ese `commit`. También podemos utilizar `"git commit"` y en consecuencia se abre un archivo de texto (.txt) en nuestro editor de código preseleccionado donde escribiremos un título del `commit` y podemos detallar o describir un poco sobre el mismo líneas más abajo; al cerrar el archivo se manda lo escrito y la consola de git vuelve a funcionar.

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.

10) ¿Qué es un repositorio remoto?

Los repositorios remotos son versiones de tu proyecto que están hospedadas en internet o en cualquier otra red. Puede tener varios de ellos, y en cada uno tendrás generalmente permisos de lectura o de lectura y escritura. Colaborar con otras personas implica gestionar estos repositorios remotos enviando y trayendo datos de ellos cada vez que necesites compartir tu trabajo.

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`.

12) ¿Cómo empujar cambios a un repositorio remoto?

Para enviar los cambios del repositorio local al repositorio remoto utilizamos el comando push, de la siguiente forma: "git push <nombre del repositorio> <nombre de la rama>".

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

Git pull es un comando de Git utilizado para actualizar la versión local de un repositorio desde otro remoto.

Es uno de los cuatro comandos que solicita interacción de red por Git. Por default, git pull hace dos cosas.

- Actualiza la rama de trabajo actual (la rama a la que se ha cambiado actualmente)
- Actualiza las referencias de rama remota para todas las demás ramas.
- git pull recupera (git fetch) las nuevas confirmaciones y las fusiona (git merge) en tu rama local.

14) ¿Qué es un fork de repositorio?

La función fork se utiliza cuando quieres contribuir a un proyecto existente al que no tienen permiso de edición. Entonces, con este comando "bifurcas" el proyecto y siendo una copia exacta del original puedes trabajar en tu espacio.

15) ¿Cómo crear un fork de un repositorio?

Buscamos algún repositorio que nos parezca interesante y yo quisiera tener ese proyecto en mi repositorio debería de hacer un fork dando click en Fork situado en la parte superior derecha de github. Una vez hecho comenzará el forking y esto sirve porque podremos trabajar con los archivos de ese proyecto y los cambios que hagamos no irán al repositorio original del autor de ese proyecto, si no a nuestra copia local.

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

Como git detecta que hiciste cambios en un proyecto ya existentes o a partir de un fork, plantea la opción de comprar y hacer pull request a través de un botón verde en tu repositorio de github o también puedes hacerlo desde la página de github en la sección de "Branches" e iniciar un pull request desde ahí.

17) ¿Cómo aceptar una solicitud de extracción?

Para hacer el pull request nos dirigiremos a la solapa de **Pull requests** allí daremos click en **new pull request**, veremos una ventana a modo de resumen en donde se reflejarán los cambios que hemos hecho nosotros en comparación al repositorio original. Daremos click en **Create pull request** donde veremos el asunto y más abajo tenemos suficiente lugar para poder explicar en mencionar el porqué ese cambio que hemos realizado nosotros, sería considerado como algo que a el repositorio original le vendrían bien agregarlo

18) ¿Qué es una etiqueta en Git?

Como muchos sistemas de control de versiones, Git te permite etiquetar puntos específicos del historial de un repositorio como importantes. Normalmente, la gente utiliza esta funcionalidad para marcar puntos de liberación (como v1.0, v2.0).

19) ¿Cómo crear una etiqueta en Git?

Crear una etiqueta anotada en Git es sencillo. La forma más fácil de hacerlo es especificar la opción **-a** cuando ejecutas el comando **git tag**

La opción **-m** especifica el mensaje de la etiqueta, el cual es guardado junto con ella. Si no especificas el mensaje de una etiqueta anotada, Git abrirá el editor de texto para que lo escribas

20) ¿Cómo enviar una etiqueta a GitHub?

De forma predeterminada, el git push no transfiere etiquetas a servidores remotos. Tendrá que enviar etiquetas explícitamente a un servidor compartido después de haberlas creado. Este proceso es como compartir ramas remotas: "git push origin <tagname>"

21) ¿Qué es un historial de Git?

A veces necesitamos examinar la secuencia de commits (confirmaciones) que hemos realizado en la historia de un proyecto, para saber dónde estamos y qué estados hemos tenido a lo largo de la vida del repositorio.

22) ¿Cómo ver el historial de Git?

La herramienta más utilizada para esto es "git log", la cual tiene variantes que te permiten ver el historial de distintas formas, facilitando su lectura o resumiendo la información.

23) ¿Cómo buscar en el historial de Git?

Si tu proyecto ya tiene muchos commits, decenas o cientos de ellos, quizás no quieras mostrarlos todos, ya que generalmente no querrás ver cosas tan antiguas como el origen del repositorio. Para ver un número de logs determinado introducimos ese número como opción, con el signo "-".

24) ¿Cómo borrar el historial de Git?

No es recomendable borrar el historial de Git pero a medida que vas ejecutando comandos te aparecen comandos útiles para deshacer lo que has hecho, en cierta medida controlada. Por ejemplo, puedes borrar fácilmente el último commit pero borrar todo el historial de cambios es más complicado.

Primero tienes que crear una rama huérfana, la idea es una rama separada que comienza con una confirmación root diferente. El siguiente paso es agregar todos los archivos nuevos y luego confirmar los cambios. Después eliminas la rama main y renombas a la huérfana como main.

25) ¿Qué es un repositorio privado en GitHub?

Solo tú, las personas con las que compartes el acceso explícitamente y, para los repositorios de organizaciones, algunos miembros de la organización, pueden acceder a los repositorios privados.

26) ¿Cómo crear un repositorio privado en GitHub?

A la hora de crear un nuevo repositorio desde la página de gitHub puedes seleccionar con qué modificador de acceso la quieres, público o privado, es este caso seleccionaremos privado. Por otro lado, podemos privatizar un repositorio ya creado ingresando a la configuración del mismo y si bajamos hasta la última sección encontraremos la Zona peligrosa y nos permite cambiar la visibilidad del repositorio.

27) ¿Cómo invitar a alguien a un repositorio privado en GitHub?

- Solicite el nombre de usuario de la persona que está invitando como colaboradora.
- En GitHub.com, navega a la página principal del repositorio.
- En el nombre del repositorio, haz clic en Configuración.
- En la sección "Acceso" de la barra lateral, haz clic en Colaboradores.
- Haz clic en Agregar personas.
- Comienza a teclear el nombre de la persona que deseas invitar dentro del campo de búsqueda. Posteriormente, da clic en algún nombre de la lista de coincidencias.
- Haga clic en Agregar NOMBRE al REPOSITORIO.
- El usuario recibirá un correo electrónico invitándolo al repositorio. Una vez que acepte la invitación, tendrá acceso de colaborador a tu repositorio.

28) ¿Qué es un repositorio público en GitHub?

Un repositorio público en GitHub significa que cualquier persona que entre a tu perfil podrá verlo.

29) ¿Cómo crear un repositorio público en GitHub?

- En la esquina superior derecha de cualquier página, utiliza el menú desplegable y selecciona New repository (Nuevo repositorio).
- Escriba un nombre corto y fácil de recordar para el repositorio. Por ejemplo: "hola-mundo".
- Opcionalmente, puede agregar una descripción del repositorio. Por ejemplo, "Mi primer repositorio en GitHub".
- Elige la visibilidad del repositorio.
- Seleccione Initialize this repository with a README (Inicializar este repositorio con un archivo Léame).
- Haga clic en Create repository (Crear repositorio).

30) ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio público en GitHub es tan fácil como compartir el link del mismo.

31) ¿Qué es un repositorio de código en GitHub?

El repositorio es el lugar en el que se almacenan las fuentes, configuraciones y archivos varios actualizados e históricos de cambios.

32) ¿Cómo crear un repositorio de código en GitHub?

Primero debemos crear o abrir una cuenta en GitHub, luego clickeamos en nuestra foto de perfil y se desplegará un menú de opciones, seleccionamos nuestros repositorios y le damos click a crear un nuevo repositorio. Ahí mismo podremos configurarlo, darle nombre, descripción, qué visibilidad tendrá, etc.

33) ¿Cómo compartir un repositorio de código en GitHub?

Si quieres compartir un repositorio creado de forma local, puedes añadirle un remoto. Para ello se emplea el código ***git remote add [alias] [url]***, esto añade una URL donde se podrá subir el repositorio con el alias escogido.

34) ¿Qué es un repositorio de documentación en GitHub?

Un repositorio de documentación es un archivo en la que se guarda documentación del código fuente, lugar al que se podrá recurrir si ocurre algún problema, darle refacción o mantenimiento al código.

35) ¿Cómo crear un repositorio de documentación en GitHub?

Crear repositorios de documentación, conlleva al mismo proceso que repositorios comunes de github, solo que el contenido que el repositorio contendrá, será de archivos de texto y documentos

36) ¿Cómo compartir un repositorio de documentación en GitHub?

De la misma forma que compartimos cualquier repositorio, a través del link.

37) ¿Qué es un repositorio de proyecto en GitHub?

Un repositorio de proyecto en GitHub es un espacio digital donde se almacenan y gestionan todos los aspectos de un proyecto, incluyendo el código fuente, la documentación, los recursos multimedia y más.

38) ¿Cómo usar Git para trabajar con otros?

Para trabajar con otras personas primero debemos invitarlos como colaboradores, una vez aceptada la invitación y configurados git, debemos clonar y trabajar como normalmente lo hacemos pero también tenemos que tener en mente realizar más seguido push y pull para mantener los trabajos actualizados.