

Trabajo Practico N° 2

ROBINO Raul Alberto

1)

- ¿Qué es GitHub?

Github es una plataforma en línea que permite alojar y compartir proyectos de software, utilizando el sistema de control de versiones Git.

- ¿Cómo crear un repositorio en GitHub?

- 1) Ingresar al sitio github.com.
- 2) Loguearse como usuario del sitio.
- 3) Ir a la esquina superior derecha en cualquier pagina y seleccionar “Nuevo Repositorio”.
- 4) Escribir el nombre del nuevo repositorio.
- 5) Añadir una descripción.
- 6) Elegir visibilidad del repositorio.
- 7) Hacer clic en “Crear Repositorio”.

- ¿Cómo crear una rama en Git?

Con el comando “**git branch nueva_rama**” en consola.

- ¿Cómo cambiar a una rama en Git?

Con el comando “**git checkout nueva_rama**” en consola.

- ¿Cómo fusionar ramas en Git?

Con el comando “**git merge**”.

- ¿Cómo crear un commit en Git?

Con el comando “**git commit -m “Mensaje descriptivo”**”. Siendo -m la opción para escribir el mensaje continuando el comando git commit.

- ¿Cómo enviar un commit a GitHub?

Con el comando “**git push origin main**” se pushea el repositorio a Github siendo en este caso main la rama que se esta trabajando.

- ¿Qué es un repositorio remoto?

Un "repositorio externo" en el contexto de desarrollo de software y gestión de código, es un almacén de código remoto, accesible a través de internet, donde se guarda y se gestiona el código fuente de un proyecto, permitiendo a múltiples desarrolladores colaborar y versionar el código de forma eficiente.

- ¿Cómo agregar un repositorio remoto a Git?

Se utiliza el comando “**git remote add origin https://github.com/OWNER/REPOSITORY.git**”

- ¿Cómo empujar cambios a un repositorio remoto?

Con el comando “**git push origin main**”

- ¿Cómo tirar de cambios de un repositorio remoto?

Con el comando “**git pull origin nombre_rama**”

- ¿Qué es un fork de repositorio?

Es la creación de una copia de un repositorio en nuestro usuario de Github de manera que puede ser editado y trabajado sin afectación al repositorio original.

- ¿Cómo crear un fork de un repositorio?

Cuando ingresamos a repositorio que queremos hacer fork en Github, hacemos clic en el botón “Fork” en la esquina superior derecha. Le asignamos un nombre al repositorio, descripción y luego “Create fork”.

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

- 1) En Github, se ingresa en la pagina del repositorio.

- 2) En el menú “Branch”, elige la rama que contiene tus confirmaciones.
- 3) Luego, encima de la lista de archivos hacer clic sobre “Comparar y solicitud de incorporación de cambios”
- 4) En el menú desplegable seleccionar la rama que se quiera combinar los cambios y después en el menú desplegable de la rama de comparación para elegir la rama de tema en la que se realizó los cambios.
- 5) Escribir un título y una descripción de mi solicitud de extracción.
- 6) Hacer clic en “Crear solicitud de incorporación de cambios”

- ¿Cómo aceptar una solicitud de extracción?

- 1) En el nombre del repositorio , hacer clic en “Pull requests”
- 2) Seleccionar la solicitud de la lista desplegable.
- 3) En la solicitud de incorporación de cambios, hacer clic en “Archivos cambiados”
- 4) Revisar los cambios propuestos.
- 5) Hacer clic en “Review changes”
- 6) Ingresa comentarios.
- 7) Seleccionar “Aprobar” y finalmente en “Enviar revisión”

- ¿Qué es un etiqueta en Git?

Es una referencia que apunta a un punto específico en el historial del repositorio, como un commit, y se utiliza para marcar momentos importantes como versiones de lanzamiento o características claves.

- ¿Cómo crear una etiqueta en Git?

Utilizando el comando “**git tag -a versión -m “Mensaje de Version”**”

- ¿Cómo enviar una etiqueta a GitHub?

Con la opción de **-tags** en el comando **git push**

- ¿Qué es un historial de Git?

Es la lista de confirmaciones hechas sobre un repositorio en orden cronológico inverso.

- ¿Cómo ver el historial de Git?

Se utiliza el comando “**git log**”

- ¿Cómo buscar en el historial de Git?

Utilizando las siguientes opciones del comando “**git log**”

- **Ver el historial completo:** git log
 - **Ver el historial de una rama específica:** git log <nombre_de_la_rama>
 - **Filtrar por autor:** git log --author=<nombre_de_autor>
 - **Filtrar por fecha:** git log --since=<fecha> o git log --until=<fecha>
 - **Buscar por mensaje de commit:** git log --grep=<palabra_clave>
 - **Buscar por contenido de código:** git log -S "<cadena_de_texto>"
 - **Ver los cambios en un archivo específico:** git log <nombre_del_archivo>
 - **Ver el historial de una función o línea de código:** git log -L
:<nombre_de_funcion>:<nombre_de_archivo>
- ¿Cómo borrar el historial de Git?
 - git rm <archivo>: Elimina el archivo del índice y del directorio de trabajo.
 - git filter-repo: Como se mencionó antes, puede ser útil para eliminar archivos o datos específicos del historial.

- ¿Qué es un repositorio privado en GitHub?

Es un espacio para almacenar código, archivos y revisiones que solo yo y otras personas autorizadas que tienen acceso pueden verlo.

- ¿Cómo crear un repositorio privado en GitHub?

En el momento de crear un repositorio nuevo, seleccionar la opción de Privado.

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

- 1) Dentro de mi repositorio seleccionar la opción de “Settings”

- 2) Seleccionar la sección de “Access” y luego “Colaboradores”
- 3) Hacer clic en “Agregar personas”
- 4) Escribir el nombre de usuario a invitar.
- 5) Hacer clic en Agregar NOMBRE al REPOSITORIO
- 6) El usuario recibirá un correo electrónico con la invitación. Cuando acepte tendrá acceso al repositorio.

- ¿Qué es un repositorio público en GitHub?

Un repositorio publico es un lugar donde puede almacenar y compartir código, archivos y el historial de cambiso de un proyecto, accesible a cualquier persona en internet.

- ¿Cómo crear un repositorio público en GitHub?

A la hora de la creación del repositorio seleccionar la opción de Publico.

- ¿Cómo compartir un repositorio público en GitHub?

Compartiendo el enlace del repositorio en el sitio de Github.

2) https://github.com/RalbertR/Ejercicio2_TP2

3) <https://github.com/RalbertR/conflict-exercise>