

TP2 ORMACHEA CHRISTIAN

1-Contestar las siguientes preguntas utilizando las guías y documentación proporcionada (Desarrollar las respuestas):

- ¿Qué es GitHub?

GitHub es una pagina de internet que aloja repositorios para que las personas trabajen en conjunto o para que muestren sus proyectos personales dejándolos como un portafolio.

- ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en GitHub lo que se tiene que hacer es primero tener una cuenta, una vez creada la misma lo único que haces es crear un repositorio nuevo con el botoncito verde y una vez clickeado ahí seleccionar si quieres que sea público o privado y poner si quieres agregar un archivo readme o no.

- ¿Cómo crear una rama en Git?

Para crear una rama tenes que hacer `git branch <nombreRama>`

- ¿Cómo cambiar a una rama en Git?

Para cambiar de rama tenes que hacer `git checkout <nombreRama>`

- ¿Cómo fusionar ramas en Git?

Para fusionar ramas tenes que pararte en la rama desde la que va a haber una fusión y poner `git merge <nombreRamaAFusionar>`

- ¿Cómo crear un commit en Git?

Para crear un commit debes hacer `git commit -m "nombreDelCommit"`

- ¿Cómo enviar un commit a GitHub?

Para enviar un commit lo que debes hacer es pushear al repositorio remoto con el comando `git push`, también antes debes seleccionar el repositorio remoto usando `git remote add <linkDelRepoDeGitHub>`

- ¿Qué es un repositorio remoto?

Un repositorio remoto es el nombre que tienen los repositorios que se crean en GitHub

- ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto a Git lo que puedes hacer es hacer un git clone <linkDelRepoDeGitHub> para clonar el repositorio del link de github o directamente descargarlo.

- ¿Cómo empujar cambios a un repositorio remoto?

Para empujar los cambios lo que tienes que hacer es un git push origin main que sería el push a la rama main que es la original que usa github.

- ¿Cómo tirar de cambios de un repositorio remoto?

Para tirar cambios de un repositorio remoto en GitHub tienes que hacer un git pull. Este comando trae los cambios desde el repositorio remoto intentándolos fusionar con el repositorio actual en el que estoy trabajando

- ¿Qué es un fork de repositorio?

Un fork en un repositorio es una bifurcación, sería como clonar el repositorio solo que el repositorio remoto sigue atado a vos, porque creas una rama nueva que sería una bifurcación de la rama main que está en github, una vez tengas los cambios realizados puedes mandarlos a la rama remota en github.

- ¿Cómo crear un fork de un repositorio?

Para crear un fork lo que puedes hacer es sobre el mismo repositorio que estás parado en github puedes hacer click en el botón que dice fork, ahí se abriría una pestaña que es igual a la pestaña que se abre cuando quieres crear un repositorio nuevo, llenas los datos y listo.

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

Para enviar un pull request a un repositorio puedes hacer click en el repositorio remoto donde dice "create pull request" y ahí seleccionarías la rama que querrías hacer un merge con la que vos ya tienes y en la misma página se enviará la petición para hacer los cambios en la rama principal de github.

- ¿Cómo aceptar una solicitud de extracción?

Para aceptar una solicitud de extracción tienes que ir a la sección que se llama pull request, una vez ahí puedes revisar los cambios y hacer clic en el botón "Aprobar"

- ¿Qué es una etiqueta en Git?

Una etiqueta es una referencia que señala un punto específico en el historial de un repositorio.

- ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta tienes que usar el comando `git tag <nombreEtiqueta>` de esta manera dejas sentada la etiqueta en el repositorio.

- ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta a GitHub puedes usar el comando `git push --tags` esto lo que haces es pushear los cambios al repositorio remoto y directamente dejas sentada la etiqueta.

- ¿Qué es un historial de Git?

El historial de git es un registro de las confirmaciones realizadas en un repositorio de Git.

- ¿Cómo ver el historial de Git?

El historial se puede ver con el comando `git log`

- ¿Cómo buscar en el historial de Git?

Para buscar en el historial puedes usar el comando `git log -L` esto lo que hace es mostrar un historial por línea de código.

- ¿Cómo borrar el historial de Git?

Para borrar el historial de git lo que puedes hacer es directamente eliminar la carpeta `.git` que esta como carpeta oculta adentro del repositorio local que creaste, de esta manera se borran todos los registros de historial.

- ¿Qué es un repositorio privado en GitHub?

Un repositorio privado es un repositorio que solo puede ver el usuario creador y las personas que tienen el link al repositorio directo.

- ¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado en github lo único que se tiene que hacer es crear un repositorio nuevo pero seleccionar el apartado de privado en vez del que es público, esto solo hace que las personas que tengan acceso al link puedan verlo.

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

Para agregar a alguien a un repositorio privado lo que se puede hacer es ir a la pestaña de colaboradores en github y añadir el nombre de usuario de github de la persona a agregar o poner directamente el mail, de esa manera ya tiene acceso al repositorio.

- ¿Qué es un repositorio público en GitHub?

Un repositorio publico de GitHub es un repositorio al que todos los que tengan el link tienen acceso.

- ¿Cómo crear un repositorio público en GitHub?

Para crear un repositorio público lo único que se tiene que hacer es crear un repositorio nuevo en la página y no seleccionar nada, la plantilla pública es la default.

- ¿Cómo compartir un repositorio público en GitHub?

Para compartir un repositorio publico de GitHub solamente se tiene que tener acceso al link, o podes invitarlos como colaboradores pero en realidad ya podrían hacer pull request asi que es lo mismo capaz.

2- Realize estas actividades:

<https://github.com/christian-ormachea/actividadProgramacion1>

3- Realizar la siguiente actividad:

<https://github.com/christian-ormachea/conflict-exercise>