

Programación I - Trabajo Práctico 2

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

¿Qué es GitHub?

Es una plataforma de desarrollo colaborativo en la nube donde los desarrolladores pueden almacenar, compartir y trabajar en proyectos de código utilizando el sistema de control de versiones GIT.

¿Cómo crear un repositorio en GitHub?

Se puede crear un repositorio desde GitHub en la sección de repositorios de la misma página, donde le colocamos el nombre del repo, la descripción, la privacidad y otras opciones más. Y para hacer un repositorio desde Git, estando en el directorio deseado, en la consola colocamos el comando “git init”, donde aparecerá el archivo oculto “.git”, para que Git rastree el directorio actual.

¿Cómo crear una rama en Git?

Para crear una rama ejecutamos el comando “git branch” seguido del nombre de la rama nueva, por ejemplo: “git branch features”.

¿Cómo cambiar a una rama en Git?

Para cambiar a una rama tenemos que ejecutar el comando “git checkout” y seguido el nombre de la rama al que queremos ir, por ejemplo “git checkout features”.

¿Cómo fusionar ramas en Git?

Para fusionar ramas tenemos que ejecutar el comando “git merge”, seguido del nombre de la rama que queremos traer los cambios, por ejemplo, estoy en la rama “main” y quiero traer los cambios de la rama “features”, entonces ejecuto “git merge features”.

¿Cómo crear un commit en Git?

Para crear un commit en git lo que hacemos es ejecutar el comando “git commit -m “ seguido de un mensaje dentro de dobles comillas, el mensaje describe resumido los cambios que hicimos en ese commit.

¿Cómo enviar un commit a GitHub?

Para enviar un commit o muchos commits desde el repositorio local a GitHub, se usa el comando “git push”.

¿Qué es un repositorio remoto?

Es un repositorio que se encuentra en un servidor como GitHub y se utiliza para almacenar y compartir el código fuente entre diferentes desarrolladores.

¿Cómo agregar un repositorio remoto a Git?

Para agregarlo tenemos que ejecutar el comando “git remote add origin “ y la url del repositorio remoto que queremos agregar.

¿Cómo empujar cambios a un repositorio remoto?

Para empujar (enviar) cambios a un repositorio remoto ejecutamos el comando “git push”.

¿Cómo tirar de cambios de un repositorio remoto?

Para tirar (traer) cambios de un repositorio remoto, ejecutamos el comando “git pull”.

¿Qué es un fork de repositorio?

Un fork es una copia exacta e independiente del repositorio original, pero con diferente url, que nos permite trabajar en él sin afectar al repositorio original.

¿Cómo crear un fork de un repositorio?

Un fork se crea en la interfaz de GitHub, en la página del repositorio original, hay un botón “Fork”. Con eso podemos hacer el fork para crear una copia original del repositorio en nuestra cuenta personal.

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

Desde el repositorio forkeado, se hace click en una sección “Pull requests”, se selecciona las ramas de origen y de destino, se añade un título y descripción de los cambios y luego se envía la solicitud al repositorio original.

¿Cómo aceptar una solicitud de extracción?

En el repositorio original, en la pestaña de “Pull requests” se selecciona la solicitud de extracción, se revisa los cambios para evitar conflictos en el código, luego se hace click en “Merge pull request” y se confirma la fusión.

¿Qué es una etiqueta en Git?

Es un marcador para un punto específico en el historial de Git, usado generalmente para marcar versiones de release o puntos importantes en el desarrollo.

¿Cómo crear una etiqueta en Git?

Para crear una etiqueta se usa el comando `"git tag v1.0"`, así creamos una etiqueta ligera. También podemos crear una etiqueta con un mensaje descriptivo con el comando `"git tag -a v1.0 -m 'Versión 1.0'"`.

¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta usamos el comando `"git push origin v1.0"` para una etiqueta específica por ejemplo, y si queremos enviar todas las etiquetas usamos el comando `"git push origin --tags"`.

¿Qué es un historial de Git?

Es un registro de todos los commits realizados en un repositorio, mostrando los cambios, quién los hizo y cuándo.

¿Cómo ver el historial de Git?

Para ver el historial de Git usamos el comando `"git log"`, podemos también ver el historial con cambios con `"git log -p"` o ver el historial resumido con `"git log --oneline"`.

¿Cómo buscar en el historial de Git?

Podemos buscar por un mensaje de commit con el comando `"git log --grep='texto a buscar'"` o también podemos buscar cambios que incluyan cierto código con el comando `"git log -S 'código a buscar'"`.

¿Cómo borrar el historial de Git?

Para borrar el historial tenemos que crear una rama nueva sin commits previos, eliminar la rama original y forzar la actualización en el repositorio remoto, todo ese proceso es irreversible.

¿Qué es un repositorio privado en GitHub?

Es un proyecto que solo pueden ver y colaborar los usuarios que han sido específicamente invitados en el mismo.

¿Cómo crear un repositorio privado en GitHub?

Para crear un repositorio privado, en el momento de crear un nuevo repositorio en GitHub y seleccionamos la opción de "Private".

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

Para invitar a alguien lo que hacemos es ir a la configuración del repositorio, seleccionamos "manage access", hacemos clic en "Invite collaborator" e ingresamos el nombre de usuario o correo de la persona que queremos invitar.

¿Qué es un repositorio público en GitHub?

Es un proyecto de código abierto visible para cualquier persona en internet.

¿Cómo crear un repositorio público en GitHub?

El mismo proceso que para el privado, vamos a crear el nuevo repositorio y seleccionamos la opción "Public".

¿Cómo compartir un repositorio público en GitHub?

Podemos compartirlo con la URL del mismo repositorio.

2. Realizar la siguiente actividad:

- Crear un repositorio.
- Agregando un archivo.
- Creando Branchs.

Link del repositorio: <https://github.com/Tomu98/TP2-Ejercicio2>

3. Realizar la siguiente actividad:

- Crear un repositorio en GitHub.
- Clonar el repositorio a tu máquina local.
- Crear una nueva rama y editar un archivo.
- Volver a la rama principal y editar el mismo archivo.
- Hacer un merge y generar un conflicto.
- Resolver el conflicto.
- Subir los cambios a GitHub.
- Verificar en GitHub.

Link del repositorio: <https://github.com/Tomu98/TP2-conflict-exercise>