TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

Materia: PROGRAMACION 1

Alumno: Hernan Jorge Urich

Trabajo Práctico N.º 2: Git-GitHub

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

• ¿Qué es GitHub?

Es una plataforma en la nube que permite almacenar, gestionar, crear repositorios, rastrear cambios y compartir código de manera eficiente, utilizando el sistema de control de versiones llamado Git.

• ¿Cómo crear un repositorio en GitHub?

Luego de crear un usuario en GitHub, e iniciar sesión se presiona el boto de color verde NEW repositorio, que está dentro de tu perfil, se desplegará una página para completar el nombre del repositorio, una descripción (opcional), decidir si el repositorio será público o privado, si desea agregar un README.MD (para agregar una descripción del repositorio), un archivo de licencias, y/o un archivo .gitignore.

• ¿Cómo crear una rama en Git?

Abrir la terminar y ubicarnos en el repositorio en el que se esta trabajando, ubicado en la rama principal (main, master) usar el comando *git branch <nombre de la rama nueva>*

• ¿Cómo cambiar a una rama en Git?

Al crear una nueva rama no se cambia en forma inmediata a ese branch, se debe colocar el comando *git checkout –b <nombre de la rama nueva>*

• ¿Cómo fusionar ramas en Git?

Hay que asegurarse de estar ubicado en la rama a la cual se va a fusionar los cambios, (mergear rama_nueva a main) debemos estar en la rama main o master, segun el caso. Se debe ejecutar *git merge <nombre de la rama nueva>*

En aso de producirse un conflicto Git lo notificara y permitira resolverlo. En caso de un conflicto se vera en el editor de codigo usado la siguiente

<<<<<<< HEAD

(Tu código actual)

=======

(Código de la rama que estás fusionando)

>>>>>>>

En este caso se eliminan las marcas y se decide que parte del codigo se va a conservar. Luego se agregan los cambios y commitea el archivo. En el caso de estar trabajando con un archivo del repositorio remoto se debe usar el comando git push origin main para subir los cambios al repositorio de GitHub. Con estos pasos la rama nueva estara integrada a la rama principal.

• ¿Cómo crear un commit en Git?

Luego de realizar el agregado de los cambios en un archivo con el comando git add <nombre del archivo> o todos los archivos con git add . se debe realizar el commit utilizando el comando *git commit –m "mensaje del commit realizado, una breve descripcion."*

• ¿Cómo enviar un commit a GitHub?

Luego de realizar el commit del archivo modificado, se necesita vincularlo con GitHub, se debe copiar la URL de tu repositorio en GitHub. Luego se tipea el comando git remote add origin URL_DEL_REPOSITORIO, luego para enviar los cambios se utiliza el comando *git push origin main* (puede ser que se esté usando otra rama o el master)

• ¿Qué es un repositorio remoto?

Es una versión del repositorio local de mi PC almacenado en un servidor en línea o en la nube, como GitHub, GitLab, sirve para almacenar proyectos.

• ¿Cómo agregar un repositorio remoto a Git?

Al crear un repositor en GitHub se debe copiar la URL de ese repositorio, luego abrir la terminar Bush o la consola de VSCode, y estando en la carpeta del proyecto se agrega el comando git remote add origin URL_DEL_REPOSITORIO (la URL que se copió del repositorio de GitHub), luego para verificar que se vinculó la URL se usa el comando git remote –v, mostrara la URL vinculada.

• ¿Cómo empujar cambios a un repositorio remoto?

Luego de configurar el repositorio remoto detallado en el paso anterior, se debe estar seguro de haber agregado y commitidado todos los archivos, se utiliza el comando *git push origin main* (se pude cambiar el main por el nombre de la rama que se está por pushiar.)

• ¿Cómo tirar de cambios de un repositorio remoto?

En el caso de un conflicto con los cambios remotos, Git notificará y se deberá solucionar antes de empujar. En este caso se usa el comando git pull origin main para traer los últimos cambios y resolver el conflicto.

• ¿Qué es un fork de repositorio?

Es una copia de un repositorio que se crea en tu cuenta de GitHub, sirve para trabajar de manera independiente en un proyecto sin afectar el repositorio original.

• ¿Cómo crear un fork de un repositorio?

Ubicado en el repositorio que se desea forkear se debe presionar el boton Fork en la parte superior derecha de la pantalla, este creara una copia del repositorio en tu cuenta, luego puedes cambiar el nombre del repo, agregar descripciones. Se puede clonar tu fork localmente.

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

Es proponer cambios al repositorio originar desde un fork o una rama diferente. Se debe ingresar al usuario de GitHub, hacer un fork del repositorio original o si estás trabajando dentro del mismo repositorio crear una rama diferente para guardar tus cambios. Luego de realizar los cambios en el proyecto, se debe abrir la pestaña "Pull Requests" en la página principal del repositorio, ubicada en la parte superior. Luego se debe hacer clic en "New Pull Request”, luego se selecciona la rama base, sea la del proyecto original (main, develop), luego se selecciona la rama de tu trabajo como la rama a fusiona. Luego se debe completar el formulario del pull request, con un título, descripción con los cambios realizados, etc. Luego se crea el pull request haciendo clic en "Create Pull Request” para enviarlo.

• ¿Cómo aceptar una solicitud de extracción?

Abrir el repositorio donde se creó el pull request, luego se debe abrir la pestaña “Pull Requests”, se selecciona el pull request que se desea aceptar haciendo clic para abrirlo, revisar los cambios en "Files changed", en caso de haber problemas puede dejar comentarios. Si todo está en orden se hace clic en el botón "Merge Pull Request", se puede escribir un mensaje para documentar la fusión. Luego se hace clic en "Confirm Merge", luego GitHub te ofrecerá eliminar la rama asociada, lo cual es opcional

• ¿Qué es una etiqueta en Git?

Son etiquetas o tag los marcadores utilizados para identificar puntos específicos en la historia de un repositorio.

• ¿Cómo crear una etiqueta en Git?

En la terminal bush de git, ubicado en la carpeta del proyecto se marca un tag ligero con el comando *git tag <nombre de la etiqueta>*. Se puede agregar información adicional o texto usando el –m "texto explicativo"

• ¿Cómo enviar una etiqueta a GitHub?

git push origin <nombre_de_la_etiqueta>

• ¿Qué es un historial de Git?

Es un registro de todos los commit que se efectuaron en el proyecto, permite revisar modificaciones, cuando se hicieron, quien las hizo, y proporciona un seguimiento detallado del progreso del proyecto.

• ¿Cómo ver el historial de Git?

Con el comando *git log*, el cual mostrara el identificado de commit (hash), el autor, la fecha, y el mensaje commit.

• ¿Cómo buscar en el historial de Git?

Para buscar en el historial se puede usar el comando *git log –grep="palabra_clave"*

También se puede usar el comando *git log –s "texto_a_buscar"*, para buscar un cambio especifico en el código. O usar el comando *git grep "texto_a_buscar"*

• ¿Cómo borrar el historial de Git?

Es poco común borrar un historial, pero puede usarse el comando *rm –rf .git*

• ¿Qué es un repositorio privado en GitHub?

Es un almacenamiento de proyectos que solo son visibles por el usuario o las personas que invites.

• ¿Cómo crear un repositorio privado en GitHub?

Luego de iniciar sesión en GitHub se selecciona la opción "New" de repositorios, luego se le asigna un nombre, una descripción (opcional), y se selecciona la opción Private. Luego se puede agregar un README.md, un archivo .gitignore y el tipo de licencia (todo opcional), luego clic en Creat Repository.

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

Ubicado en el repositorio privado de GitHub, hacer clic en la pestaña "Setting", ubicada en la parte superior del repositorio, ir a la sección "Collaborators" dentro de la sección "Access". Luego hacer clic en el botón "Add People" escribiendo el email de la persona que deseas invitar. Luego enviarle la invitación "Invite", lo que generara una invitación al colaborador para que acepte la invitación.

• ¿Qué es un repositorio público en GitHub?

Es un almacenamiento de proyectos que son visibles por todos los usuarios de GitHub.

• ¿Cómo crear un repositorio público en GitHub?

Luego de iniciar sesión en GitHub se selecciona la opción "New" de repositorios, luego se le asigna un nombre, una descripción (opcional), y se selecciona la opción "Public". Luego se puede agregar un README.md, un archivo .gitignore y el tipo de licencia (todo opcional), luego clic en Creat Repository.

• ¿Cómo compartir un repositorio público en GitHub?

Se comparte utilizando la URL del repositorio creado en la página de Github. Se copia la dirección y se comparte con otros usuarios.

2) Realizar la siguiente actividad:

• Crear un repositorio.

o Dale un nombre al repositorio.

o Elije el repositorio sea público.

o Inicializa el repositorio con un archivo.

• Agregando un Archivo

o Crea un archivo simple, por ejemplo, "mi-archivo.txt".

o Realiza los comandos git add . y git commit -m "Agregando mi-archivo.txt" en la línea de comandos.

o Sube los cambios al repositorio en GitHub con git push origin main (o el nombre de la rama correspondiente).

• Creando Branchs

- o Crear una Branch
- o Realizar cambios
- o agregar un archivo
- o Subir la Branch

URL DE RESPUESTA INCISO 2:

https://github.com/h-j-urich/Trabajo_Practico2_inciso2.git

3) Realizar la siguiente actividad:

Paso 1: Crear un repositorio en GitHub

• Ve a GitHub e inicia sesión en tu cuenta.

• Haz clic en el botón "New" o "Create repository" para crear un nuevo repositorio.

• Asigna un nombre al repositorio, por ejemplo, conflict-exercise.

• Opcionalmente, añade una descripción.

• Marca la opción "Initialize this repository with a README".

• Haz clic en "Create repository".

Paso 2: Clonar el repositorio a tu máquina local

• Copia la URL del repositorio (usualmente algo como https://github.com/tuusuario/conflict-exercise.git).

• Abre la terminal o línea de comandos en tu máquina.

• Clona el repositorio usando el comando:

git clone https://github.com/tuusuario/conflict-exercise.git

• Entra en el directorio del repositorio: cd conflict-exercise

Paso 3: Crear una nueva rama y editar un archivo

• Crea una nueva rama llamada feature-branch: git checkout -b feature-branch

• Abre el archivo README.md en un editor de texto y añade una línea nueva, por ejemplo: Este es un cambio en la feature branch.

• Guarda los cambios y haz un commit: git add README.md git commit -m "Added a line in feature-branch"

Paso 4: Volver a la rama principal y editar el mismo archivo

• Cambia de vuelta a la rama principal (main): git checkout main

• Edita el archivo README.md de nuevo, añadiendo una línea diferente: Este es un cambio en la main branch.

• Guarda los cambios y haz un commit: git add README.md git commit -m "Added a line in main branch"

Paso 5: Hacer un merge y generar un conflicto

• Intenta hacer un merge de la feature-branch en la rama main: git merge feature-branch

• Se generará un conflicto porque ambos cambios afectan la misma línea del archivo README.md.

Paso 6: Resolver el conflicto

• Abre el archivo README.md en tu editor de texto. Verás algo similar a esto:

<<<<<<< HEAD

Este es un cambio en la main branch.

=======

Este es un cambio en la feature branch.

>>>>>>> feature-branch

• Decide cómo resolver el conflicto. Puedes mantener ambos cambios, elegir uno de ellos, o fusionar los contenidos de alguna manera.

• Edita el archivo para resolver el conflicto y guarda los cambios (Se debe borrar lo marcado en verde en el archivo donde estes solucionando el conflicto. Y se debe borrar la parte del texto que no se quiera dejar).

• Añade el archivo resuelto y completa el merge:

git add README.md

git commit -m "Resolved merge conflict"

Paso 7: Subir los cambios a GitHub

• Sube los cambios de la rama main al repositorio remoto en GitHub: git push origin main 4

• También sube la feature-branch si deseas:

git push origin feature-branch

Paso 8: Verificar en GitHub

• Ve a tu repositorio en GitHub y revisa el archivo README.md para confirmar que los cambios se han subido correctamente.

• Puedes revisar el historial de commits para ver el conflicto y su resolución.

URL RESPUESTA EJERCICIO INCISO 3:

https://github.com/h-j-urich/conflict-exercise.git