ARQUITECTURA Y SISTEMAS OPERATIVOS

Trabajo Práctico N°1: Introducción a la arquitectura de computadoras

Alumno: Alegría Angel Gabriel

Consignas:

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

• ¿Qué es GitHub?

GitHub es una plataforma de alojamiento de código. Que permite a los desarrolladores almacenar, compartir y colaborar en proyectos. Utiliza Git como sistema de control de versiones y permite trabajar en equipo, gestionar cambios y realizar revisiones de código.

• ¿Cómo crear un repositorio en GitHub?

Para crear un repositorio en repositorio en GitHub se deben seguir los siguientes pasos:

1-Inicia sesión en tu cuenta de GitHub.

2-Haz clic en el botón "New" en la parte superior de la página de inicio.

3-Completa los detalles del repositorio: nombre, descripción, visibilidad (privado o público).

4-Hacer clic en "Create repository" para crear el repositorio.

• ¿Cómo crear una rama en Git?

Para crear una nueva rama en git, se usa el comando: git branch nombre-rama

• ¿Cómo cambiar a una rama en Git?

Para cambiarte a una rama específica en git, usa el comando: git checkout nombre- rama

• ¿Cómo fusionar ramas en Git?

Para fusionar una rama a la rama actual, usa: git merge nombre- rama

Por ej: si estamos en la rama main y queremos fusionarnos con la rama bugfix. Usariamos el comando: git merge bugfix

• ¿Cómo crear un commit en Git?

Para hacer un commit, primero agrega los cambios al archivo con 'git add . Luego, realiza el commit: git commit -m "Mensaje del commit"

• ¿Cómo enviar un commit a GitHub?

Para enviar los commits locales a GitHub se utiliza el comando: git push origin <nombre-de-la-rama>

• ¿Qué es un repositorio remoto?

Un repositorio remoto es una versión de tu repositorio local, que está alojada en un servidor en línea, como GitHub.

• ¿Cómo agregar un repositorio remoto a Git?

Para agregar un repositorio remoto se utiliza el comando: git remote add origin <URL-del-repositorio>

• ¿Cómo empujar cambios a un repositorio remoto?

Para empujar los cambios locales a un repositorio remoto se utiliza el comando: git push origin <nombre-de-la-rama>

• ¿Cómo tirar de cambios de un repositorio remoto?

Para traer cambios de un repositorio remoto y fusionarlos con tu rama local se utiliza el comando: git pull origin <nombre-de-la-rama>

• ¿Qué es un fork de repositorio?

Un "fork" es una copia de un repositorio que puedes modificar sin afectar al repositorio original. Es útil cuando quieres contribuir a un proyecto de otra persona o de código abierto.

• ¿Cómo crear un fork de un repositorio?

Para hacer un fork de un repositorio en GitHub:

Ve a la página del repositorio que deseas bifurcar.

Haz clic en el botón "Fork" en la esquina superior derecha.

Puedes cambiar la descripción (opcional), no afecta al original

Luego haz click en crear fork

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

Haz cambios en tu fork.

Dirígete a la página de tu repositorio en GitHub.

Haz clic en "Pull Request".

Pon un título y agrega una descripcion

Haz clic en "Create Pull Request".

• ¿Cómo aceptar una solicitud de extracción?

Ve a la sección de **"Pull Requests"** de tu repositorio.

Revisa los cambios y haz clic en **"Merge"** para fusionar la solicitud de extracción con tu rama principal.

• ¿Qué es una etiqueta en Git?

Una etiqueta (tag) en Git es un marcador en un commit específico, generalmente usado para señalar versiones o puntos importantes en el historial del repositorio.

• ¿Cómo crear una etiqueta en Git?

Para crear una etiqueta, usa el siguiente comando: git tag <nombre-de-la-etiqueta>

• ¿Cómo enviar una etiqueta a GitHub?

Para enviar una etiqueta al repositorio remoto en GitHub, usa: git push origin <nombre-de-la-etiqueta>

• ¿Qué es un historial de Git?

El historial de Git es un registro de todos los cambios realizados en el repositorio, incluyendo commits, fusiones, etiquetas, etc.

• ¿Cómo ver el historial de Git?

Puedes ver el historial de Git con el siguiente comando: git log

• ¿Cómo buscar en el historial de Git?

Para buscar en el historial de Git, puedes usar: git log --grep="<palabra-clave>"

• ¿Cómo borrar el historial de Git?

Para borrar el historial de git podemos utilizar el comando: git clean –fd que borra los archivos y carpetas sin seguimiento. O borrar la carpeta (.git) en la raíz de la carpeta donde se encuentran los archivos

• ¿Qué es un repositorio privado en GitHub?

Un repositorio privado en GitHub es aquel cuyo código solo es accesible para el creador y los usuarios a los que se les haya dado acceso.

• ¿Cómo crear un repositorio privado en GitHub?

Cuando creas un nuevo repositorio en GitHub, puedes elegir entre hacerlo público o privado. Solo selecciona **"Private"** en las opciones de visibilidad.

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

Ve a la página del repositorio privado

Haz clic en **"Settings"**.

Selecciona **"Manage access"** y luego **"Invite a collaborator"**.

Introduce el nombre de usuario de GitHub de la persona a la que deseas invitar y haz clic en **"Add"**.

• ¿Qué es un repositorio público en GitHub?

Un repositorio público es accesible para cualquier persona. Cualquiera puede ver y clonar el código.

• ¿Cómo crear un repositorio público en GitHub?

Cuando creas un nuevo repositorio, selecciona **"Public"** en las opciones de visibilidad.

• ¿Cómo compartir un repositorio público en GitHub?

Puedes compartir un repositorio público copiando la URL del repositorio en la barra de direcciones de tu navegador y compartiéndola, o usando el botón **"Share"** en GitHub.

2) Realizar la siguiente actividad: https://github.com/Gabriel-Alegria/Gabriel-Alegria-TP2-Programacion1.git

• 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

3) Realizar la siguiente actividad: https://github.com/Gabriel-Alegria/conflict-exercise.git

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 estés 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

• 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.