

1. Desarrolla el siguiente Cuestionario

1- ¿Que es Git?

Git es un sistema de control de versiones distribuido, que permite rastrear cambios en archivos y colaborar con otros desarrolladores en un proyecto.

2- ¿Para que funciona el comando Git init?

El comando **git init** se utiliza para inicializar un nuevo repositorio de Git en un directorio específico. Esto crea un nuevo repositorio vacío donde se pueden comenzar a agregar archivos y rastrear cambios.

3- ¿Que es una rama?

Una rama en Git es una línea de desarrollo independiente que se crea a partir de la rama principal (o rama "master") del repositorio. Las ramas se utilizan para trabajar en nuevas características o arreglos sin afectar la rama principal, lo que permite a los desarrolladores trabajar en paralelo en diferentes funciones del proyecto.

4- ¿Como saber es que rama estoy?

Para saber en qué rama está actualmente, se puede utilizar el comando "git branch" en la línea de comandos de Git. La rama actual se mostrará con un asterisco (*) junto a su nombre.

5- ¿Quien creo git?

Git fue creado por Linus Torvalds, el creador del kernel de Linux, en 2005.

6- ¿Cuales son los comandos más esenciales de Git?

- **git add:** para agregar archivos al área de preparación.
- **git commit:** para confirmar los cambios y registrarlos en el historial del repositorio.
- **git push:** para enviar los cambios locales al repositorio remoto.
- **git pull:** para obtener los cambios más recientes del repositorio remoto.
- **git merge:** para fusionar cambios de una rama a otra.

7- ¿Que es git Flow?

Git Flow es un modelo de flujo de trabajo para Git que se utiliza para organizar y estructurar el proceso de desarrollo de software en un proyecto. Este modelo utiliza dos ramas principales: "develop" y "master", y establece reglas para crear y fusionar ramas de características y arreglos.

8- Que es trunk based development?

Trunk Based Development es un enfoque de desarrollo de software en el que todos los cambios se realizan en la rama principal del repositorio (también conocida como tronco o "trunk"), en lugar de utilizar ramas de características separadas. Este enfoque

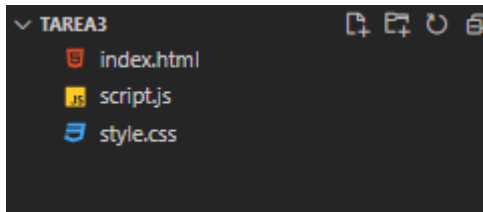
se enfoca en la entrega continua y en mantener el código siempre listo para producción.

2. Desarrolle un ejercicio práctico en Azure DevOps o GitHub con las siguientes características

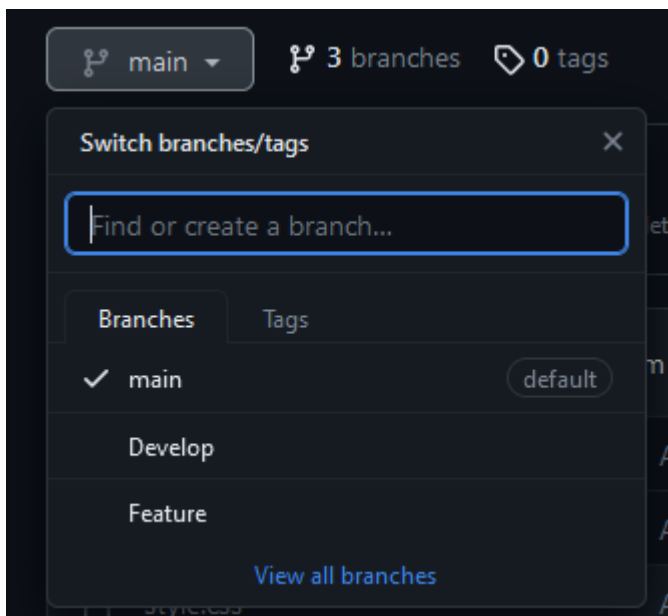
Entregas:

Link de GitHub donde se desarrollen las siguientes actividades:

-Crear un proyecto.



-Utilizar la técnica Git Flow en su proyecto.



-Proyecto funcional.

Bienvenido a Mate Calc+

