Edixón González - Ingeniería informática

Como estudiantes, es importante seguir buenas prácticas para el manejo de ramas en GitHub.

Aquí hay algunas recomendaciones que considero importantes:

Usar ramas dedicadas: Cada vez que trabajas en una nueva función o vas a corregir errores, crea una rama dedicada para eso en lugar de trabajar directamente en la rama principal. De esta forma evitas dañar todo el código si te equivocas o si la nueva función que quieres agregar produce errores con lo que ya tienes creado, y solo después de terminar de crear esa función y confirmar que funciona o de arreglar los errores unificas esa rama a la rama principal.

Usar nombres descriptivos para las ramas: Debemos darle a cada rama un nombre descriptivo que indique claramente su propósito. Esto nos ayudará a mantener nuestro repositorio organizado y facilitará la colaboración con otros miembros de equipo en caso de que trabajes con compañeros, también esto puede prevenir mezclar algo que no estaba listo por error.

Usar commits frecuentemente: Debemos realizar commits frecuentes y con mensajes descriptivos que expliquen los cambios que realizamos. Esto nos permitirá realizar un seguimiento de los cambios y facilitará la colaboración con otros miembros del equipo.

Usar git status: Antes de realizar cualquier cambio, debemos asegurarnos de ejecutar el comando git status para verificar el estado actual del repositorio. Esto nos permitirá saber en qué rama nos encontramos actualmente y qué cambios se han realizado desde la última vez que sincronizamos nuestro repositorio local con el repositorio remoto.

Usar git add de manera selectiva: En lugar de usar git add para agregar todos los archivos modificados al área de preparación, es mejor agregar solo los archivos en los que estuvimos trabajando. Esto evitará que se agreguen archivos innecesarios al repositorio y mantendrá el repositorio más limpio.

Usar .gitignore: El archivo .gitignore nos permite especificar patrones de nombres de archivos o directorios que Git debe ignorar al realizar operaciones como add o commit. Esto nos permitirá evitar agregar archivos innecesarios al repositorio y mantenerlo más limpio

Siguiendo estas buenas prácticas, podemos mantener nuestro repositorio organizado y facilitar la colaboración con otros miembros del equipo.

Parte práctica:

