Allison Scarlet Paula Beato
2022-2075

Tarea de GIT

1. ¿Qué es Git?

Es un software de control de versiones, lo que significa que ayuda a registrar los cambios realizados en un proyecto a lo largo del tiempo.

2. ¿Cuál es el propósito del comando git init en Git?

Se utiliza para crear un nuevo repositorio Git. Un repositorio es una carpeta que contiene todos los archivos de tu proyecto, así como el historial de versiones de esos archivos.

Al ejecutar git init en una carpeta, se crea un nuevo subdirectorio llamado .git. Este subdirectorio contiene todos los datos necesarios para realizar un seguimiento de los cambios en tu proyecto.

3. ¿Qué representa una rama en Git y cómo se utiliza?

Representa una línea de desarrollo independiente. Es como una copia del código fuente de tu proyecto, donde puedes trabajar en nuevas características o correcciones de errores sin afectar la rama principal (master).

Utilización:

Las ramas se utilizan para:

- Desarrollar nuevas características: Puedes crear una nueva rama para trabajar en una nueva característica sin afectar la rama principal. Esto te permite probar la nueva característica y asegurarte de que funciona correctamente antes de integrarla en la rama principal.
- Corregir errores: Puedes crear una nueva rama para corregir un error específico. Esto te permite aislar el error y solucionarlo sin afectar el resto del código.
- Colaborar con otros: Puedes compartir una rama con otros desarrolladores para que puedan trabajar en ella y contribuir al proyecto.

Comandos básicos de ramas:

git branch <nombre_rama>: Crea una nueva rama.

git checkout <nombre_rama>: Cambia a la rama especificada.

git merge <nombre_rama>: Integra los cambios de la rama especificada en la rama actual.

git branch -d <nombre_rama>: Elimina la rama especificada.

Allison Scarlet Paula Beato
2022-2075

4. **¿Cómo puedo determinar en qué rama estoy actualmente en Git?**

Para saber en qué rama estás trabajando actualmente en Git, puedes usar el comando git branch: *git branch*

Este comando mostrará una lista de todas las ramas existentes en tu repositorio, y la rama actual estará marcada con un asterisco (*) al lado.

5. **¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado?**

Git fue creado por Linus Torvalds, el famoso creador del kernel de Linux. El desarrollo de Git comenzó en abril de 2005, después de que BitKeeper, el sistema de control de versiones que se usaba para el desarrollo del kernel de Linux, dejara de ser gratuito para el desarrollo de Linux.

Torvalds diseñó Git para que fuera eficiente, confiable y compatible con el mantenimiento de versiones de aplicaciones con un gran número de archivos de código fuente. Git se ha convertido en el sistema de control de versiones más utilizado del mundo, y es utilizado por millones de desarrolladores de software para realizar un seguimiento de los cambios en sus proyectos.

6. **¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan?**

Inicialización:

git init: Crea un nuevo repositorio Git en la carpeta actual.

Seguimiento de cambios:

git add <archivo>: Agrega archivos al área de preparación para el siguiente commit.

git commit -m "Mensaje": Registra los cambios en el área de preparación como un commit con un mensaje descriptivo.

git status: Muestra el estado actual del repositorio, incluyendo los archivos modificados y sin confirmar.

Colaboración:

git clone <url>: Clona un repositorio remoto en tu máquina local.

git push: Envía tus cambios locales al repositorio remoto.

git pull: Obtiene los cambios del repositorio remoto y los fusiona con tu rama local.

Allison Scarlet Paula Beato
2022-2075

Navegación:

git branch: Muestra una lista de las ramas existentes.

git checkout <nombre_rama>: Cambia a la rama especificada.

git log: Muestra el historial de commits del repositorio.

Resolución de conflictos:

git merge <nombre_rama>: Fusiona los cambios de la rama especificada con tu rama actual.

git reset: Deshace los cambios locales.

Otros comandos útiles:

git diff: Muestra las diferencias entre dos versiones de un archivo.

git stash: Guarda los cambios sin confirmar para trabajar en otra rama.

git clean: Elimina archivos sin seguimiento del directorio de trabajo.

7. **¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?**

Gitlab

Microsoft/Windows

Nodejs/node

Atlassian/bitbucket