



Nombre: Juan Elias Hernandez Jimenez

Matricula: 2023-0613

Carrera: Desarrollo de software

Materia: Metodología De Investigación

Tema: Octava actividad: La fenomenología

Maestro: Ramon Leonardo Diaz Diaz

1.¿Qué es Git?

Git es un sistema de control de versiones distribuido que permite a los desarrolladores gestionar y rastrear los cambios en el código fuente durante el desarrollo de software. Git permite trabajar de manera colaborativa en proyectos, facilitando la integración de cambios de múltiples desarrolladores y la recuperación de versiones anteriores del código.

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

El comando `git init` se utiliza para inicializar un nuevo repositorio Git en un directorio. Al ejecutar `git init`, se crea un nuevo subdirectorio llamado `.git` en el directorio actual, que contiene todos los archivos necesarios para el control de versiones. Este comando establece el directorio como un repositorio Git vacío listo para comenzar a rastrear archivos.

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

En Git, una rama (o branch) es una línea de desarrollo independiente del repositorio principal. Permite trabajar en diferentes características o correcciones sin afectar el código principal. Las ramas se utilizan para experimentar y desarrollar funcionalidades de forma aislada. Una vez que el trabajo en una rama está completo y probado, se puede fusionar (merge) con la rama principal (usualmente `main` o `master`).

Para crear una nueva rama, se usa el comando ``git branch nombre-de-la-rama``. Para cambiar a una rama, se usa ``git checkout nombre-de-la-rama``. Desde Git 2.23, también se puede usar ``git switch nombre-de-la-rama`` para cambiar de rama.

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

Puedes determinar en qué rama estás actualmente utilizando el comando ``git branch`` o ``git status``. El comando ``git branch`` lista todas las ramas en el repositorio y marca la rama activa con un asterisco (``*``). El comando ``git status`` también muestra la rama actual en la primera línea de su salida.

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

Git fue creado por Linus Torvalds en 2005. Linus Torvalds, el creador del sistema operativo Linux, desarrolló Git para gestionar el código fuente del kernel de Linux después de que el sistema de control de versiones anterior, BitKeeper, dejara de ser de código abierto.

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

- ``git init``: Inicializa un nuevo repositorio Git.

- ``git clone <url>``: Clona un repositorio remoto en el directorio local.
- ``git add <archivo>``: Agrega archivos al área de preparación (staging area) para el próximo commit.
- ``git commit -m "mensaje"```: Crea un nuevo commit con los archivos en el área de preparación.
- ``git status``: Muestra el estado de los archivos en el repositorio y la rama actual.
- ``git pull``: Descarga y fusiona cambios desde un repositorio remoto a la rama actual.
- ``git push``: Envía los commits locales a un repositorio remoto.
- ``git branch``: Lista, crea o elimina ramas.
- ``git merge <rama>``: Fusiona la rama especificada con la rama actual.
- ``git log``: Muestra el historial de commits.

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

Algunos de los servicios de repositorios Git más reconocidos y utilizados en la actualidad incluyen:

- GitHub: Una plataforma popular para alojamiento de repositorios Git y colaboración en proyectos. Ofrece características como pull requests, revisiones de código y un amplio ecosistema de integraciones.

- GitLab: Una plataforma que proporciona tanto servicios de alojamiento de repositorios Git como herramientas de CI/CD (integración continua/despliegue continuo). Ofrece características similares a GitHub, pero también incluye funcionalidades de gestión de proyectos y seguimiento de problemas.
- Bitbucket: Un servicio de alojamiento de repositorios Git (y Mercurial) que proporciona características de colaboración y CI/CD. Es conocido por su integración con otras herramientas de Atlassian, como Jira y Confluence.
- SourceForge: Un servicio de alojamiento de proyectos de código abierto que ofrece repositorios Git (y otros sistemas de control de versiones), así como herramientas de gestión de proyectos y descargas.