



Estudiante

Esteilor Paniagua Mateo 2022-1966

Tema:

Tarea 3

Maestro:

Kelyn Tejeda Belliard

Asignatura:

Git repo

Fecha:

1/04/2024

Tarea 3	1
Punto1:	3
1. ¿Qué es Git?	3
2. ¿Cuál es el propósito del comando git init en Git?	3
3. ¿Qué representa una rama en Git y cómo se utiliza?	3
4. ¿Cómo puedo determinar en qué rama estoy actualmente en Git?	4
5. ¿Quién es la persona responsable de la creación de Git y cuándo fue desarrollado?	4
6. ¿Cuáles son algunos de los comandos esenciales de Git y para qué se utilizan?	4
7. ¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?	5
Punto 2:	6
Enlace GitHub :	6

Punto1:

Desarrolla el siguiente Cuestionario

1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido, diseñado para rastrear cambios en archivos y coordinar el trabajo entre múltiples personas en proyectos de desarrollo de software. Permite a los desarrolladores trabajar de manera colaborativa en un mismo proyecto, manteniendo un historial completo de todas las modificaciones realizadas en los archivos.

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 existente o para crear uno nuevo. Este comando crea un nuevo subdirectorio `.git` que contiene todos los archivos necesarios para el repositorio Git, incluyendo la base de datos de objetos, la configuración y el registro de referencias.

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

Una rama representa una línea de desarrollo independiente, que incluye un conjunto específico de commits. Las ramas permiten a los desarrolladores trabajar en diferentes características o versiones del proyecto de forma paralela sin interferir entre sí. Se utilizan para realizar cambios experimentales, trabajar en nuevas características o realizar correcciones de errores sin afectar la rama principal.

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

Se puede determinar en qué rama se está utilizando el comando `git branch`. La rama actual estará resaltada con un asterisco (*) al lado de su nombre en la lista que muestra el comando `git branch`.

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

Git fue creado por Linus Torvalds, el mismo creador del kernel de Linux. Comenzó su desarrollo en 2005 como una alternativa al sistema de control de versiones existente en ese momento (BitKeeper), que se había utilizado para el desarrollo del kernel de Linux, pero se volvió restringido para su uso gratuito.

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

Algunos comandos esenciales de Git incluyen:

- `git add`: Agrega cambios en archivos al área de preparación (staging area) para ser confirmados.
- `git commit`: Confirma los cambios en el repositorio.
- `git push`: Envía cambios locales confirmados al repositorio remoto.
- `git pull`: Obtiene cambios desde el repositorio remoto y los fusiona con la rama actual.
- `git clone`: Clona un repositorio remoto en un nuevo directorio local.
- `git merge`: Fusiona una rama específica en la rama actual.
- `git checkout`: Cambia entre ramas o restaura archivos desde un commit específico.
- `git status`: Muestra el estado actual del repositorio, incluyendo archivos modificados, agregados y sin seguimiento.

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

1. GitHub: Es la plataforma de alojamiento de código más popular del mundo. Cuenta con millones de repositorios públicos y privados, y una comunidad activa de desarrolladores.

2. GitLab: Es una plataforma similar a GitHub, que ofrece funcionalidades adicionales como CI/CD, gestión de tickets y wikis.

3. Bitbucket: Es la plataforma de alojamiento de código de Atlassian. Se integra con otras herramientas de Atlassian como Jira y Confluence.

4. GitLab: Es una plataforma de alojamiento de código autohospedada. Es una buena opción para empresas que necesitan tener un mayor control sobre sus datos.

5. SourceForge: Es una de las plataformas de alojamiento de código más antiguas. Ofrece espacio de almacenamiento gratuito para proyectos de código abierto.

6. CodePlex: Es una plataforma de alojamiento de código de Microsoft. Se utiliza principalmente para proyectos de código abierto .NET.

7. Gitea: Es una plataforma de alojamiento de código autohospedada. Es una buena opción para empresas que necesitan tener un mayor control sobre sus datos.

8. Docker Hub: Es un registro de imágenes de Docker. Se utiliza para almacenar y compartir imágenes de Docker.

9. npm: Es un gestor de paquetes para JavaScript. Se utiliza para instalar y gestionar paquetes de JavaScript.

10. RubyGems: Es un gestor de paquetes para Ruby. Se utiliza para instalar y gestionar paquetes de Ruby.