

Instituto Tecnológico de las Américas



Nombre:

Osia Miguel Clara Reynoso

Matricula:

2020-10214

Asignatura:

Programación 3

Profesor:

Kelyn Tejada Belliard

Tarea 3:

Desarrolla el siguiente Cuestionario:

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

Desarrollo:

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. Fue creado por Linus Torvalds en 2005 y se ha convertido en una herramienta fundamental para el desarrollo de software colaborativo. Git permite a los desarrolladores mantener un historial completo de cambios en el código, realizar ramificaciones y fusiones de manera eficiente, y colaborar de forma efectiva en proyectos de cualquier tamaño.

¿Qué es Github?

GitHub, por otro lado, es una plataforma de alojamiento de código basada en la web que utiliza Git para el control de versiones. Permite a los desarrolladores alojar, compartir y colaborar en proyectos de software de manera remota. Además de proporcionar características de control de versiones, GitHub ofrece herramientas adicionales como seguimiento de problemas, solicitudes de extracción, wikis, y funciones de gestión de proyectos, lo que lo convierte en una herramienta integral para el desarrollo de software colaborativo y de código abierto.

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

El comando `git init` se utiliza en Git para inicializar un nuevo repositorio en un directorio local. Cuando ejecutas `git init` en un directorio, Git crea un nuevo repositorio vacío en ese directorio, configurando la estructura necesaria para comenzar a rastrear los cambios en los archivos del proyecto.

El propósito principal del comando `git init` es establecer un nuevo repositorio Git en un directorio específico. Una vez que se ha ejecutado `git init`, el directorio se convierte en un repositorio Git local donde se pueden realizar operaciones de

control de versiones, como realizar confirmaciones de cambios, crear ramas, fusionar cambios, entre otras.

Es importante tener en cuenta que `git init` solo se ejecuta una vez en un directorio para establecer el repositorio. Una vez que se ha inicializado el repositorio, puedes comenzar a trabajar en tu proyecto y utilizar otros comandos de Git para gestionar los cambios y colaborar en el desarrollo del software.

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

Una rama en Git es una línea de desarrollo independiente que se bifurca del flujo principal del proyecto, permitiendo el desarrollo paralelo de nuevas características, correcciones de errores o experimentos sin afectar el código en la rama principal.

Se utiliza creando una nueva rama con el comando `git branch <nombre-de-la-rama>`, cambiando a ella con `git checkout <nombre-de-la-rama>`, realizando cambios y confirmaciones específicos en esa rama, y finalmente fusionándola de nuevo en la rama principal cuando el trabajo está completo utilizando `git merge`.

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

Se puede determinar en qué rama uno está actualmente en Git utilizando el comando `git branch`. Este comando mostrará todas las ramas presentes en tu repositorio local y resaltará con un asterisco (*) la rama en la que te encuentras. Simplemente abre tu terminal o línea de comandos y escribe:

```
git Branch
```

El resultado será similar al siguiente:

```
* master
```

```
nueva-funcionalidad
```

En este ejemplo, el asterisco (*) indica que estás actualmente en la rama llamada "master". Si estuvieras en otra rama, esa rama estaría resaltada con el asterisco en lugar de "master".

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

Git fue creado por **Linus Torvalds**, el mismo individuo responsable del desarrollo del kernel de Linux. **Linus desarrolló Git en 2005 como una solución para gestionar el desarrollo del kernel de Linux de manera más eficiente y escalable.** Desde entonces, Git se ha convertido en uno de los sistemas de control de versiones más populares y ampliamente utilizados en el mundo del desarrollo de software.

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

Los comandos mas esenciales son los siguientes:

- **git init:** Este comando se utiliza para inicializar un nuevo repositorio Git en un directorio local.
- **git clone:** Se utiliza para clonar un repositorio Git existente desde una ubicación remota (como GitHub) hacia tu máquina local.
- **git add:** Este comando se utiliza para agregar cambios en archivos al área de preparación (staging) para que puedan ser confirmados.
- **git commit:** Se utiliza para confirmar los cambios en los archivos que han sido agregados al área de preparación. Cada confirmación crea un punto en la historia del proyecto.
- **git status:** Muestra el estado actual del repositorio, incluyendo los cambios que han sido agregados al área de preparación, los cambios que no han sido rastreados, y más.

- **git branch:** Se utiliza para crear, listar y gestionar ramas en el repositorio. Puedes crear nuevas ramas, cambiar entre ramas, renombrarlas, eliminarlas, entre otras operaciones.
- **git checkout:** Se utiliza para cambiar entre ramas o para restaurar archivos desde el repositorio.
- **git merge:** Se utiliza para fusionar una rama en otra. Por ejemplo, puedes fusionar una rama de desarrollo en la rama principal (como master o main).
- **git pull:** Se utiliza para recuperar los cambios desde un repositorio remoto y fusionarlos en la rama local actual.
- **git push:** Se utiliza para enviar tus cambios confirmados a un repositorio remoto, como GitHub.

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

Aquí algunos de los repositorios mas populares y utilizados en la actualidad:

- **Linux Kernel** (<https://github.com/torvalds/linux>): El repositorio oficial del kernel de Linux, mantenido por Linus Torvalds y otros colaboradores.
- **React** (<https://github.com/facebook/react>): Un popular framework de JavaScript para construir interfaces de usuario, mantenido por Facebook.
- **TensorFlow** (<https://github.com/tensorflow/tensorflow>): Una biblioteca de código abierto para aprendizaje automático y redes neuronales, desarrollada por Google.
- **VS Code** (<https://github.com/microsoft/vscode>): El repositorio de Visual Studio Code, un popular editor de código fuente desarrollado por Microsoft.

- Django (<https://github.com/django/django>): Un framework web de alto nivel basado en Python, utilizado para desarrollar aplicaciones web de manera rápida y eficiente.
- Bootstrap (<https://github.com/twbs/bootstrap>): Un framework de código abierto para desarrollo web front-end, mantenido por Twitter.
- Angular (<https://github.com/angular/angular>): Un framework de aplicación web desarrollado por Google, utilizado para construir aplicaciones de una sola página (SPA) y aplicaciones web progresivas (PWA).
- Node.js (<https://github.com/nodejs/node>): Un entorno de tiempo de ejecución de JavaScript de código abierto que permite a los desarrolladores ejecutar JavaScript en el lado del servidor, mantenido por la comunidad Node.js.