



Instituto Tecnológico de las Américas (ITLA)

Carrera o Técnico:

Tecnólogo en Desarrollo de Software

Asignatura:

Programación III

Profesor:

Kelyn Tejada Belliard

Asunto:

Tarea 3 – Herramientas de administración de fuentes – Punto 1
Cuestionario

Estudiante:

Braulio Peralta

Matricula:

2022-0606

1. ¿Qué es Git?

Git es un sistema de control de versiones distribuido que controla y administra las versiones de un programa, aplicación o proyecto durante la evolución de este. Permite dividir el trabajo en ramas y trabajar en equipos de distintas áreas de manera individual o cooperativa de manera más eficiente de manera local o remota. Git es la herramienta más comúnmente utilizada para lograr este objetivo y permite gestionar todos los cambios que se realizan y registrar cuando se realizan las actividades y el autor de estas.

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

Este comando crea un repositorio Git vacío o reinicializar uno ya existente. Es un directorio .git con subdirectorios y ficheros de plantillas. Se creará una rama inicial sin confirmaciones.

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

Es una línea independiente de desarrollo derivada de la rama principal que permite trabajar, como su nombre lo indica, independientemente de las otras ramas o la principal, se utiliza generalmente para añadir características, configuraciones o modificaciones para luego fusionarlas con otra rama, generalmente la principal. El comando `git branch <nombre de la rama o funcionalidad>` permite crear las ramas mientras para borrar las ramas comando `git branch -d<nombre de la rama o funcionalidad>`.

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

Utilizando el comando `git branch` mostrará todas las ramas y la rama marcada con el asterisco señalará la rama que se está trabajando.

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. Linus comenzó a trabajar en Git en abril de 2005 y anunció la primera versión estable en abril de 2007.

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

- `git init`:

Esto crea un subdirectorio nuevo llamado `.git`, el cual contiene todos los archivos necesarios del repositorio – un esqueleto de un repositorio de Git. Todavía no hay nada en tu proyecto que esté bajo seguimiento.

- `git fetch`:

Descarga los cambios realizados en el repositorio remoto.

- `git merge <nombre_rama>`:

Impacta en la rama en la que te encuentras parado, los cambios realizados en la rama “nombre_rama”.

- `git pull`:

Unifica los comandos *fetch* y *merge* en un único comando.

- `git commit -m "<mensaje>"`:

Confirma los cambios realizados. El “mensaje” generalmente se usa para asociar al *commit* una breve descripción de los cambios realizados.

- `git push origin <nombre_rama>`:

Sube la rama “nombre_rama” al servidor remoto.

- `git status`:

Muestra el estado actual de la rama, como los cambios que hay sin commitear.

- `git add <nombre_archivo>`:

Comienza a trackear el archivo “nombre_archivo”.

- `git checkout -b <nombre_rama_nueva>`:

Crea una rama a partir de la que te encuentres parado con el nombre “nombre_rama_nueva”, y luego salta sobre la rama nueva, por lo que quedas parado en esta última.

- `git checkout -t origin/<nombre_rama>`:

Si existe una rama remota de nombre “nombre_rama”, al ejecutar este comando se crea una rama local con el nombre “nombre_rama” para hacer un seguimiento de la rama remota con el mismo nombre.

- `git branch`:

Lista todas las ramas locales.

- `git branch -a`:

Lista todas las ramas locales y remotas.

- `git branch -d <nombre_rama>`:

Elimina la rama local con el nombre “nombre_rama”.

- `git push origin <nombre_rama>`:

Commitea los cambios desde el branch local origin al branch “nombre_rama”.

- `git remote prune origin`:

Actualiza tu repositorio remoto en caso de que algún otro desarrollador haya eliminado alguna rama remota.

- `git reset --hard HEAD`:

Elimina los cambios realizados que aún no se hayan hecho *commit*.

- `git revert <hash_commit>`:

Revierte el *commit* realizado, identificado por el “hash_commit”.

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

GitHub: GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

GitLab: Gitlab Inc. es una compañía de núcleo abierto y es la principal proveedora del software GitLab, un servicio web de forja, control de versiones y DevOps basado en Git. Además de gestor de repositorios, el servicio ofrece también alojamiento de wikis y un sistema de seguimiento de errores, todo ello publicado bajo una licencia de código abierto, principalmente.

Bitbucket: Bitbucket es un servicio de alojamiento basado en web, para los proyectos que utilizan el sistema de control de versiones Mercurial y Git.

Bibliografía

Recursos de Apoyo – Plataforma Virtual ITLA

[Git - git-init Documentation \(git-scm.com\)](https://git-scm.com/docs/git-init)

[Git - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/Git)

[GitHub - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/GitHub)

[GitLab - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/GitLab)

[Bitbucket - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/Bitbucket)