



Programación III

Tarea Git

Parte 1: Teoría

1. ¿Qué es Git?

Git es un sistema de control de versiones, estos sistemas permiten tener un control de cada cambio realizado en un proyecto de software. Git es el sistema más utilizado para el control de versiones en la industria por el hecho de ser de código abierto y su gran soporte por su comunidad y las empresas interesadas.

Git se caracteriza por haber iniciado este movimiento de los sistemas de control de versiones y a ver sentados las bases de sus futuros competidores. En git como núcleo principal los repositorios, estos son “almacenes” donde puedes alojar los archivos de tu proyecto de manera local y donde se registran cada cambios que subas decidas guardar en el repositorio. Otras de las virtudes de git son la creación de las ramas y la capacidad subir tu repositorio a un servicio de repositorios en la nube (discutiremos estos aspectos más adelante).

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

El propósito del comando “git init” es de crear un nuevo repositorio vacío o en el caso de que ya exista un repositorio en la ruta especificada este lo reiniciara desde cero como si fuera un repositorio vacío.

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

Una rama en git es una “versión” alternativa del almacenamiento principal de nuestro repositorio (este se designa por defecto como “main” o “master”) con el propósito de generar cambios independientes entre los desarrolladores sin necesidad de alterar el almacenamiento principal o rama principal, o alterar el trabajo realizado por otros desarrolladores. Por ejemplo, utilizando la metodología Scrum, si un miembro del equipo necesita trabajar en una historia de usuario este puede crear otra rama que deribe de la rama principal la cual podrá realizar sus tareas sin necesidad de afectar las tareas de otros miembros del equipo.

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

Existen varios comandos para poder saber en que rama estamos utilizando en este momento, hablemos de cuales son:

- Git status: Este comando es utilizado para brindarnos información del directorio donde estamos trabajando actualmente principalmente si hemos realizado un cambio en nuestro código que aun no se ha enviado al repositorio local nos lo dira. En la primera línea de información nos dirá en que rama estamos trabajando.

- Git brach: Este comando se utiliza para mostrar, modificar y eliminar ramas existentes. Al ejecutar el comando nos listara todas la ramas de nuestro repositorio y la rama donde estemos trabajando actualmente estará señalada con un asterico (*).
- Git rev-parse --abbrev-ref HEAD: git rev-parse se utiliza para varios propósitos, pero al agregar los parámetros --abbrev-ref HEAD este nos brindara el la rama con la cual estamos trabajando sin ninguna información adicional.

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

En el año 1992 un desarrollador llamado Linus Benedict Torvalds o coloquialmente llamado Linus Torvalds había iniciado un proyecto llamado Linux por estar disgustado con el funcionamiento de otro sistema operativo llamado Minix. Linux fue creado con la filosofía de código abierto, que dicta que todos los que quisieran podía ver el código fuente de su proyecto, aun según Wikipedia este lo lanzo al publico porque habían ciertas partes necesarias para un SO que el no tenia mucha ganas de trabajar.

Mucho años después, más concretamente en 2002, la base de desarrolladores que estaban trabajando en este proyecto superaban los cientos, así que Linus decidió utiliza un sistema de control de versiones distribuido (DVCS, por sus siglas en inglés) llamado BitKeeper. BitKeeper permitía que todos los usuarios que estaban trabajando en su proyecto tenga una copia entera del proyecto que podía ser considerado el “repositorio principal” a demás de tener un mejor control de las aportaciones de cada desarrollador. Pero para 2005 la compañía de BitKeeper decidió cancelar la versión gratuita del sistema lo que provocó el descontento de la comunidad de Linux y al propio Linus.

Durante el 2005, Linus busco alternativa DVCS a BitKeeper pero ninguna lo convencía, así que decidió que el iba a crear su propio DVCS. Linus se propuso 5 objetivos que debía cumplir su DVCS: Muy alta velocidad para subir los cambios, un diseño muy simplificado y fácil de entender, soporte para desarrollo no línea o sea que puedan trabajar con múltiples ramas en paralelo, completamente distribuido y muy eficiente. Así el 7 de abril de 2005 (según Wikipedia) nació Git, cumpliendo por mucho todos los objetivos.

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

- Git init: el anteriormente mencionado comando que sirve para crear un nuevo repositorio o “reiniciar” uno ya existente.
- Git clone: este comando se utiliza para clonar un repositorio remoto y crear un repositorio local con todo el proyecto clonado.
- Git add: Este comando añade los cambios realizados o seleccionados en la parte conocida como área de preparación (es un área donde se almacenan todos los cambios antes de ser guardados en el repositorio).
- Git commit: Este comando guarda en el repositorio local los cambios almacenados en el área de preparación con un mensaje el cual el desarrollador redactara los cambios que realizo en ese commit.
- Git log: Este comando nos muestra todos los commits realizados.

- Git Branch: como si había hablado anteriormente, este comando se utiliza para crear, alterar y eliminar una rama existen y también lista todas la ramas que tiene el proyecto.
 - Git checkout: Este comando se utiliza para cambiar entre una rama a otra especificada.
 - Git merge: Este comando fusiona la rama actual con una rama especificada.
 - Git remote: Este comando añade un repositorio remoto al proyecto para almacenar el proyecto en dicho repositorio.
 - Git fetch: Este comando se utiliza para ver los cambios que se han realizados por otros compañeros en un repositorio remoto sin descargarlos en el repositorio local.
 - Git pull: Este comando nos muestras los cambios realizados por otros compañeros en un repositorio remoto y los descarga en nuestro repositorio local.
 - Git push: Este comando sube todos los commits guardados en nuestro repositorio local y los sube a un repositorio remoto.
7. ¿Puedes mencionar algunos de los repositorios de Git más reconocidos y utilizados en la actualidad?
- Github: Es el servicio de repositorios remotos mas utilizados en el mundo. Se caracteriza por tener una interfaz muy user-frendly, tener varias herramientas como Github pages para crear una pagina web desde los servidores de github, github para crear tarea automatizadas y Github codespace que te permite utilizar una maquina virtual para probar código.
 - Gitlab: Es una plataforma DevOps para almacenar repositorios git. Se caracteriza por tener herramientas CI/CD para poder almacenar y probar tus proyectos, un servicio de hosting donde puede almacenar tus productos y distribuirlos a través de sus propios servidores y herramientas de ayuda para detectar vulnerabilidades de seguridad de tu proyecto.
 - Bitbucket: Es una plataforma de repositorio remoto de git creado por Atlassian, lo mismo que distribuyen la plataforma Jira. Este se caracteriza por la integración de otros productos de Atlassian como Jira y Confluence.

Parte 2: Proyecto practico

Repositorio del proyecto: <https://github.com/EdRRojas/Tarea-3---Publishers>

Innecesario ya que este documento estará alojado en el propio repositorio.