Presentación.

Nombre: Edison Agüero.

Matricula: 2023-1130

Carrera: Desarrollo de software.

Materia: Programación 3.

Docente: Kelyn Tejada.

Tema: Tarea 3.

Fecha: 21/11/2024

1 ¿Qué es Git?

Git es un sistema de control de versiones distribuido ampliamente utilizado para el seguimiento y administración de cambios en proyectos de software. Permite a los desarrolladores trabajar de manera colaborativa y controlar el historial de modificaciones en un repositorio local.

2 ¿Para qué funciona el comando git init?

El comando **git init** se utiliza para inicializar un nuevo repositorio de Git en un directorio existente. Esto crea un subdirectorio oculto llamado **.git**, donde se almacenan todos los archivos y metadatos necesarios para el seguimiento del control de versiones en el proyecto.

3 ¿Qué es una rama en Git?

Una rama en Git es una línea separada de desarrollo dentro de un repositorio. Permite trabajar en características, correcciones o experimentos sin afectar la rama principal (main o master). Las ramas facilitan el trabajo en paralelo y la integración de cambios cuando están listos.

4 ¿Cómo saber en cuál rama estoy?

Puedes saber en qué rama estás utilizando el comando "**git branch**".

El nombre de la rama actual estará marcado con un asterisco (*). También puedes ver esta información en el prompt de la terminal si está configurada para mostrar el estado de Git.

5 ¿Quién creó Git?

Git fue creado en 2005 por **Linus Torvalds**, el creador del núcleo Linux. Fue desarrollado como una herramienta para gestionar el desarrollo del proyecto Linux, buscando ser rápida, eficiente y distribuida.

6 ¿Cuáles son los comandos más esenciales de Git?

Algunos de los comandos más esenciales de Git son:

git init: Inicializa un nuevo repositorio.

git clone: Clona un repositorio existente.

git add: Añade archivos al área de preparación (staging area).

git commit: Guarda los cambios en el repositorio con un mensaje descriptivo.

git status: Muestra el estado actual del repositorio.

git push: Envía los cambios al repositorio remoto.

git pull: Descarga y fusiona los cambios del repositorio remoto en el repositorio local.

git branch: Muestra, crea o elimina ramas.

git merge: Es el proceso de combinar los cambios realizados en una rama con otra rama, generalmente fusionando una rama secundaria con la rama principal.

7 ¿Qué es Git Flow?

Git Flow es una **metodología de ramificación y flujo de trabajo** para el control de versiones con **Git** diseñada para facilitar la **colaboración en equipos y la administración de características**, lanzamientos y correcciones de errores de manera estructurada. Aunque **no es parte oficial de Git**, es una convención popular y existe un conjunto de herramientas que ayudan a implementar esta metodología.

main: Contiene la versión estable y lista para producción.

develop: Para integrar el trabajo de desarrollo continuo.

Ramas de características (feature): La **rama Feature** se utiliza para **añadir nuevas funcionalidades a un proyecto**, desarrollándolas por separado antes de unirlas a la rama principal.

Ramas de corrección (hotfix): La rama Hotfix sirve para **manejar problemas y errores críticos en producción** sin que esto afecte al trabajo desarrollado en la rama Develop.

Ramas de liberación (release): Si la rama Develop se considera lo suficientemente estable, se puede **lanzar como una nueva versión del software** a través de una **rama Release**.

8 ¿Qué es Trunk-Based Development?

Es una estrategia de Git donde existe un trunk(un branch principal, usualmente llamado master/main) en el cual todo el equipo colabora he integra directamente (hace push)

Bibliografía.

¿Qué es GitFlow y para qué sirve? - APIHack

¿Por qué Trunk-Based Development? - DEV Community

Cómo obtener la rama actual en Git | TRSPOS

Comandos de GIT Básicos - Guía Completa

2024-C-3-1615-2880-TDS-007-SDN: Conceptos básico