



Presentación.

Nombre(s):

Thammy Franceily

Apellidos:

Vargas Acosta

Matricula:

2021-1809

Materia:

Programación III

Maestro:

Kelyn Tejada Belliard

Tema:

Módulo V – Asignación #3

I. Desarrolla el siguiente cuestionario.

1. ¿Qué es Git?

***Git** es un sistema de control de versiones distribuidos en el cual permite que un equipo de desarrolladores pueda trabajar juntos en un mismo proyecto y que se pueda tener un registro de todos los cambios realizados al código a lo largo del tiempo establecido.*

2. ¿Para qué funciona el comando **Git init**?

***Git init** se utiliza para poder crear un nuevo repositorio de Git en una dirección de trabajo que ya existe, es decir, en una carpeta ya existente. Cuando se ejecuta el comando en esa carpeta, Git creará un nuevo repositorio en esa carpeta y comenzará a hacer un seguimiento de los cambios que se hagan en los archivos en esa dirección.*

3. ¿Qué es una rama?

***Una rama** es una línea en la cuál se puede desarrollar el código. Cuando se crea un repositorio, siempre se crea una rama principal, llamada MAIN. Se puede trabajar en esta rama, pero lo recomendable es que para editar o crear nuevas funciones se creen otras ramas y cuando se termine el propósito con el cual se creó, se fusione con la rama principal.*

4. ¿Cómo sabes en que rama estás?

Para saber en qué rama se está trabajando se debe abrir la CMD o la terminal del repositorio de Git y escribir:

git branch

Este comando va a mostrar todas las ramas del repositorio y marcará un () en la rama que se está trabajando. También, se puede utilizar:*

git status

Este comando muestra la rama actual y más especificaciones en comparación al git branch.

5. ¿Quién creo Git?

Git fue creado por el desarrollador de software Linus Torvalds en 2005.

6. ¿Cuáles son los comandos más esenciales de Git?

Git init. *Se utiliza para inicializar un nuevo repositorio en una carpeta seleccionada.*

Git add. *Con este comando se mueven los cambios del directorio del trabajo al área del entorno de ensayo.*

Git commit. *Se utiliza para guardar los cambios realizados en el código. Para poder subir un commit debe añadirse un comentario o algún mensaje.*

Git status. *Con este comando podemos visualizar el estado del repositorio, donde se aprecia los archivos modificados, la rama actual y demás especificaciones.*

Git log. *Se utiliza para ver el historial de los commits que se han realizado en el repositorio, mostrando quién los realizó, la fecha y descripción de cada commit.*

Git branch. *Con este comando se visualizan todas las ramas del repositorio y se pueden crear nuevas.*

Git checkout. *Se utiliza para cambiarse de rama.*

Git merge. *Con este comando se utiliza para fusionar cambios de una rama a otra.*

Git clone. Se utiliza para clonar un repositorio ya existente en una nueva carpeta.

Git push. Este comando se utiliza para enviar los cambios locales al repositorio remoto.

7. ¿Qué es Git Flow?

Git Flow es una metodología sobre el control de versiones que trabaja en utilizar ramas específicas y un flujo de trabajo estructurado para el desarrollo de software de un equipo.

Las ramas principales de esta metodología es **la rama MAIN**, que debe contener el código con la aplicación ya funcional, y **la rama DEVELOP**, donde se desarrolla todas las funcionalidades del proyecto.

De la rama DEVELOP, nacen otras ramas:

- **Feature:** Rama para crear nuevas funcionalidades o características.
- **Release:** Rama que se utiliza para corregir errores y realizar pruebas antes de que se publique la nueva versión.

También existe la rama de **Hotfix**. Esta rama nace de la **MAIN** y se crea para corregir errores críticos en esta rama, al finalizar se fusiona con la rama MAIN y DEVELOP.

8. ¿Qué es Trunk Based Development?

Trunk Based Development es una estrategia de desarrollo de software en la que se trabaja de forma directa en la rama principal del repositorio en lugar de utilizar ramas separadas para cada tarea o funcionalidad en comparación a git Flow.

En esta estrategia los desarrolladores realizan cambios

pequeños y frecuentes en la rama principal, y utilizan pruebas automáticas y code reviews para garantizar que los cambios no rompan la funcionalidad existente del software.

II. Desarrolla un ejercicio práctico en Azure DevOps o GitHub con las siguientes características.

- Link de GitHub donde se utiliza la técnica Git Flow y que contenga un proyecto funcional.

<https://github.com/mimird06/Asignacion3.git>

Bibliografía

<https://www.atlassian.com/es/git/tutorials/what-is-git#:~:text=Git%20es%20un%20proyecto%20de,sistema%20operativo%20Linux%2C%20en%202005>

https://www.atlassian.com/es/git/tutorials/using-branches/git-checkout#:~:text=Para%20saber%20qué%20ramas%20están,rama%20actual%2C%20ejecuta%20git%20branch%20.&text=En%20el%20ejemplo%20anterior%2C%20se,este%20caso%2C%20la%20rama%20feature_inprogress_branch%20

<https://www.atlassian.com/es/git/glossary>

<https://www.atlassian.com/es/git/tutorials/setting-up-a-repository/git-clone#:~:text=git%20clone%20se%20utiliza%20principalmente,puede%20acceder%20por%20máquina%20remota>

<https://www.atlassian.com/es/git/tutorials/syncing/git-push#:~:text=El%20comando%20git%20push%20se,local%20a%20un%20repositorio%20remoto>

<https://www.atlassian.com/es/continuous-delivery/continuous-integration/trunk-based-development>