

Instituto Tecnológico de Costa Rica

Escuela de Mecatrónica

Automatización y Redes industriales

Tarea 1

Preguntas Teóricas

Andres Matamoros

Ismael Piedra Quirós

II Semestre 2024

- 1) ¿Explique la principal utilidad de git como herramienta de desarrollo de código?

La principal utilidad de Git es su sistema de control de versiones distribuido. Esto permite gestionar y rastrear los cambios en un código de manera eficaz y colaborativa. Esto es posible gracias a que el software mantiene un registro completo de los cambios realizados en el código a lo largo del tiempo, y es posible revertir estos cambios a versiones anteriores con el fin de evitar y corregir errores cuando el código no funciona como se esperaba. El software también permite una colaboración eficiente entre varios desarrolladores para trabajar en equipo,

- 2) Explique la diferencia entre git y github .

La diferencia entre git y github es que git es el sistema de control de versiones de código. Por otro lado github es un programa para alojar repositorios y servicios basados en git. Git almacena repositorios en máquinas locales, y github almacena en la nube, permitiendo acceso remoto y colaboración.

- 3) ¿Qué es un branch?

Son ramas que permiten desarrollar nuevas funcionalidades sin afectar el código estable, son ramas de desarrollo independientes, que pueden integrarse al código estable cuando puedan estar terminadas.

- 4) En el contexto de github. ¿Qué es un Pull Request?

Es una solicitud que se puede hacer para que los cambios hechos en una rama sean revisados e integrados en otro código. Se hacen pruebas y verificaciones para luego de ser aprobado, se unan las ramas en cuestión. Esto facilita las colaboraciones en el código y mantiene su calidad.

- 5) ¿Qué es un commit?

Es básicamente guardar cambios realizados en el código en el historial de un repositorio en github por ejemplo. Registra permanentemente los cambios desde el último commit.

- 6) Describa lo que sucede al ejecutar la siguiente operación: “git rebase main”.

Se utiliza para mover o rebasear la base de la rama actual sobre la punta de la rama main. Este comando se ejecuta cuando estás en una rama diferente y quieres incorporar los cambios más recientes de la rama main en la rama actual, pero de una manera que produzca un historial de commits más lineal y limpio, en lugar de crear un commit de fusión.

7) Explique que es un “merge conflict” y como lo resolvería.

Ocurre cuando Git no puede automáticamente reconciliar diferencias en los archivos al intentar fusionar (merge) dos ramas. Esto sucede cuando los cambios en las mismas partes del mismo archivo difieren entre las dos ramas. Para resolverlo se puede intentar una fusión con `git merge`, , si no sucede, el programa marca que hay un conflicto y en donde, para poder resolverlo manualmente y localizar las diferencias en las diferentes ramas. Luego se marca el conflicto como resuelto y se completa la fusión y se aplica un commit.

8) ¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?

Es un tipo de prueba de software que se enfoca en verificar el correcto funcionamiento de las unidades más pequeñas de código, típicamente funciones o métodos individuales en un programa. Estas pruebas son esenciales para asegurar que cada parte del código funcione correctamente en aislamiento antes de integrarse con otras partes del sistema.

9) Bajo el contexto de pytest. ¿Cuál es la utilidad de un “assert”?

Se utiliza para verificar que una condición específica sea verdadera durante la ejecución de una prueba. Si la condición resulta ser falsa, pytest registra una falla en la prueba, proporcionando información detallada sobre la comparación que falló, lo que facilita la identificación y corrección de errores.

10) Mencione y explique 3 errores de formato detectables con Flake8

1. E501: Línea Demasiado Larga

El error E501 se produce cuando una línea de código supera los 79 caracteres, que es el límite recomendado por PEP 8. Se puede corregir usando líneas mas cortas usando un salto de línea.

2. E302: Falta de Espacios en Blanco antes de Definiciones de Función o Clase

El error E302 se produce cuando no hay dos líneas en blanco antes de una definición de función o clase a nivel de módulo, según lo recomendado.

3. E265: Falta de Espacio Después del Carácter # en Comentarios

El error E265 se produce cuando no hay un espacio después del carácter # en un comentario.