

TECNOLÓGICO DE COSTA RICA

Licenciatura Ingeniería en Mecatrónica



Microprocesadores y Microcontroladores

Tarea 1

Dr.-Ing. Carlos Adrián Salazar García

Erick Adrián Rodríguez Piñeiro
Taylor Jose Ruiz Segura
Brandon Enrique Segovia Sanchez

II Semestre 2024

Preguntas Teóricas

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

Git es una herramienta que proporciona mucha flexibilidad y confiabilidad al desarrollo de código. Permite a los desarrolladores rastrear cambios en el código y quien los realizó por medio de un historial de modificaciones, por lo que se puede recuperar cualquier versión del proyecto. Además, permite la creación de ramas, que son copias separadas del código donde se pueden realizar pruebas o desarrollar nuevas características sin afectar a la rama principal. Si los cambios son estables, se pueden fusionar a la rama principal. Lo anterior se conoce como "Branching" y "Merging". Además es importante recalcar que Git permite que un proyecto lo trabajen múltiples desarrolladores.

2) Explique la diferencia entre git y github

Git es un software que permite gestionar y controlar los cambios en el código fuente de un proyecto. Mientras que github es una plataforma de hospedaje de repositorios git basada en la nube, que además ofrece servicios adicionales sobre el control de versiones.

Además, git se puede utilizar en cualquier entorno, local o remoto. Mientras que github necesita obligatoriamente tener internet para su funcionamiento.

3) ¿Qué es un branch?

Es una versión del código del proyecto que se está trabajando. Son útiles debido a que se pueden hacer cambios sin afectar al código principal. Para luego, si es aceptado, fusionarse al principal.

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

Es una solicitud que el creador de una rama en un repositorio realiza al dueño del repositorio principal para que este último revise y apruebe los cambios hechos en la rama.

5) ¿Qué es un commit?

Un commit es una acción básica en Git que documenta un conjunto de cambios en un repositorio e incluye una breve descripción de los cambios. Este proceso permite un control de versiones eficaz y agiliza la colaboración, el seguimiento y la gestión de proyectos.

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

El comando ‘git rebase main’ aplica los commits de la rama actual a la rama main, creando una secuencia de commits más limpia y estructurada. Este proceso facilita la incorporación de cambios y ayuda a mantener organizado el historial de commits, aunque puede ser necesario resolver conflictos manualmente.

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

Los merge conflict ocurren cuando Git no puede fusionar automáticamente dos ramas debido a importantes diferencias entre estas, como cambios en las mismas líneas de un archivo o si alguno fue eliminado en una rama mientras que en la otra fue modificado. Para resolver este problema, debe editar manualmente los archivos en conflicto, decidir cómo aplicar los cambios, marcar el conflicto como resuelto (‘git add nombre_del_archivo’) y completar el proceso de fusión (‘git commit’ o ‘git merge –continue’). Herramientas visuales (GitKraken, SourceTree, VSCode) y comandos específicos (‘git mergetool’, ‘git log –merge’) pueden ayudarle a gestionar y resolver estos conflictos de forma más eficiente.

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

Es una prueba para una parte significativamente pequeña y aislada del código principal, se realiza para saber si una función, método, o aplicación está funcionando correctamente y es aplicado de manera en que un problema pueda ser subdividido para su análisis y solución.

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

Un “Assert” es una validación de Pytest que verifica valores esperados para una función, es decir que si una función debe retornar un valor numérico, nos podría confirmar si se cumple o no este valor esperado por medio de la terminal de Python

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

1. F401 - “Module imported but unused”: Se importa un módulo pero este no es utilizado, se puede ocasionar tanto en llamar a varios módulos, o que se haya nombrado equivocadamente al módulo y por ello no se encuentra en uso.
2. F706 - “a return statement outside of a function/method”: La consola devolverá un mensaje de SyntaxError cuando se escriba return fuera de una función, ya que esta no estará vinculada a la función.
3. F524 - “string dot format missing arguments”: Cuando se utiliza la función de str.format, pero se omiten argumentos que están siendo llamados, por lo cual devolverá KeyError ya que parte de la función quedará sin llamar.