

Microprocesadores y microcontroladores (MT7003)

Tarea 1

Preguntas teóricas

Estudiantes:

- José Rodolfo Castrillo Cubero (2020154039)
- Carlos David Argüello Hidalgo (2021045923)

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

Git es una herramienta de Control de Versiones, el cual “es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante” (Git, s.f., párr. 2). Su principal utilidad reside en su modelo de “ramificación” (branching), lo cual permite el crear nuevas “branches” para explorar ideas nuevas, con la posibilidad de devolverse en cualquier momento a versiones anteriores, aplicar remiendos y mezclar ramas entre sí (Git, s.f.). De esta manera, es fácil llevar una estructura organizada de producción, mantenimiento e innovación.

2) ¿Qué es un branch?

Es una rama o línea de diseño que toma la base de la línea principal de desarrollo y en la que se pueden seguir desarrollando nuevas versiones sin afectar la línea principal o main

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

Es una solicitud de que una branch pueda ser mezclada (merge) en otra (usualmente la master/main). Permite que miembros del repositorio puedan realizar observaciones sobre los cambios que se pretenden realizar hasta que se obtenga el visto bueno y el administrador realice el merge, luego del cual se suele eliminar la branch secundaria (freeCodeCamp.org, 2020, 43:24).

4) ¿Qué es un commit?

Se trata de guardar en Git los cambios hechos en los archivos seleccionados anteriormente con el comando “git add” (freeCodeCamp.org, 2020, 6:03). Tal commit debe estar acompañado de una buena explicación sobre los cambios realizados; puede ser revertido en caso de que sea necesario.

5) Describa lo que sucede al ejecutar la siguiente operación: “git cherry-pick ”.

Al ejecutar tal instrucción, se tomará cierto commit de una branch y se aplicará a otra (Atlassian, s.f.). Suele ser utilizado cuando se desea incorporar cierta funcionalidad específica hecha en una branch distinta pero sin tener que hacer un merge con todos los commits de la otra branch.

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

Un “merge conflict” es un tipo de error que puede ocurrir a la hora de realizar un merge en una Branch. Las principales causas de dicho error son: que dos o más desarrolladores estén editando la misma línea de código al mismo tiempo, o que un desarrollador elimine un archivo mientras otro está trabajando en este (Atlassian, s.f.). La forma de resolver un merge conflict consiste en editar manualmente el archivo que está en conflicto.

7) ¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?
Una prueba unitaria, como su nombre lo indica, es una prueba que permite evaluar el funcionamiento de una unidad de código, la cual puede ser desde algo tan pequeño como una línea de código, como una función o método completo (Smartbear, s.f.).

8) Bajo el contexto de pytest. ¿Cuál es la utilidad de un “assert”?
Un assert se puede utilizar bajo el contexto de pytest para verificar el resultado esperado a la hora de realizar un test (Pytest, s.f.).

- 9) ¿Mencione y explique 3 errores de formato detectables con Flake8?
- E101: indentation contains mixed spaces and tabs: Este error indica que la cantidad de espacios en la tabulación de una línea es distinta a la cantidad por defecto configurada.
 - E999 IndentationError: unexpected indent: Este error indica que hay una tabulación inesperada en una línea de código.
 - E501: line too long (maximum 79 characters): Este error indica que la cantidad de caracteres en una línea supera el máximo establecido.

10) Explique la funcionalidad de parametrización de pytest.
La funcionalidad de parametrización en pytest permite al desarrollador crear múltiples argumentos parametrizados para la realización de pruebas de manera más autónoma y con mayor eficacia (Pytest, s.f.).

Referencias

Git. (s.f.). *1.1 Inicio - Sobre el Control de Versiones - Acerca del Control de Versiones*. <https://git-scm.com/book/es/v2/Inicio---Sobre-el-Control-de-Versiones-Acerca-del-Control-de-Versiones>

Git. (s.f.). *About*. <https://git-scm.com/about>

Git. (s.f.). *3.1 Ramificaciones en Git - ¿Qué es una rama?*. <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-%C2%BFQu%C3%A9-es-una-rama%3F>

freeCodeCamp.org. (2020, May 28). *Git and GitHub for Beginners - Crash Course* [Video]. YouTube. <https://www.youtube.com/watch?v=RG0j5yH7evk>

Atlassian. (s.f.). *Git cherry pick*. <https://www.atlassian.com/git/tutorials/cherry-pick>

Atlassian. (s.f.). *Git merge conflicts*. <https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts#:~:text=Git%20can%20handle%20most%20merges,working%20in%20a%20team%20environment>.

Smartbear. (s.f.). *What is unit testing*. <https://smartbear.com/learn/automated-testing/what-is-unit-testing/>

Pytest. (s.f.). *asserting whit the assert statement*. <https://docs.pytest.org/en/stable/how-to/assert.html#asserting-with-the-assert-statement>

Pytest. (s.f.). *How to parametrize fixtures and test functions*. <https://docs.pytest.org/en/stable/how-to/parametrize.html>