

TECNOLÓGICO DE COSTA RICA

Licenciatura Ingeniería en Mecatrónica



MT-7003 Microprocesadores y microcontroladores

Tarea #1

Ing. Rodolfo Piedra Camacho

Estudiantes:

**Diana Cerdas
2020023718**

**Jeaustin Calderón
2020027413**

I Semestre 2023

Preguntas Teóricas

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

Git es una herramienta sofisticada para el manejo de versiones. Permite administrar el código fuente de los proyectos de una manera versátil y ágil. Tiene una gran cantidad de funciones, dentro de ellas permite gestionar las versiones del proyecto desde una consola, permitiendo un uso más eficiente. Asimismo, permite colaborar en un ambiente remoto/local que facilita un entorno colaborativo, haciendo posible que varios desarrolladores con distintas especialidades trabajen en funciones distintas del proyecto al mismo tiempo y sin tener problemas sobre la futura compatibilidad [1].

2) ¿Qué es un branch?

Es una herramienta que permite administrar las versiones del proyecto de manera paralela. Se podría decir que son líneas de tiempo distintas, pero paralelas. Al crear un proyecto se tiene como default la Branch denominada como master. Por lo general se crean branches para trabajar en nuevos features, una vez que se termina el feature, se unen las versiones y se borra la branch [2].

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

Pull request es una propuesta de cambio que se puede realizar a un proyecto. Se presenta una versión del proyecto realizada con cambio a la última versión subida y se espera por su aprobación o rechazo [1].

4) ¿Qué es un commit?

Es lo que sostiene el estado actual en el repositorio. Se puede considerar como un nodo en una lista enlazada. Todos los commit están relacionados con el commit previo. Desde cualquier commit es posible retroceder a mirar el historial de los commits [3].

5) Describa lo que sucede al efectuar las siguientes operaciones: “git fetch” “git rebase origin/master”

- Git fetch: permite descargar los nuevos commits desde el servidor remoto al servidor local, de manera que se tiene la misma versión en ambos [1].
- Git rebase origin/master: permite realizar un rebase de una rama al master [4], esto permite reestructurar la manera en que están definidos los punteros en los nodos (commits) esto permite tener un historial más limpio y entendible [5].

6) Explique que es un “merge conflict” o “rebase conflict” en el contexto de tratar de hacer merge a un Pull Request o de completar una operación git rebase.

Un “merge conflict” o un “rebase conflicto” suceden cuando la herramienta Github no es capaz de realizar las operaciones de merge o git rebase de forma automática. En este caso, existen dos versiones diferentes de una misma línea de código, por lo que se genera una contradicción en el programa. Otra forma en la que se pueden presentar los dos conflictos mencionados es que una persona realice un cambio a un archivo y que otra persona borre este mismo archivo. Como consecuencia, la herramienta va a entrar en confusión, pues cuando se dan las instrucciones de las operaciones merge y rebase, no es posible encontrar el archivo en la carpeta de trabajo [6] [7].

Para solucionar el problema solamente se le debe indicar a Git cuál es el código que se desea conservar, o bien, se puede cambiar el código para unir ambos comandos o líneas [8].

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

Son una serie de pruebas que tienen como objetivo verificar individualmente que cada uno de los elementos de un código funcionen correctamente. Es importante destacar que estas son pruebas fáciles y rápidas, que por lo general son ejecutadas automáticamente, por medio de funciones o librerías según el lenguaje de programación. Generalmente, las pruebas unitarias son utilizadas para el control de calidad de un programa, de modo que se obtengan la mayor cantidad de errores posibles y brindar una mejor experiencia al usuario [9][10].

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

Es una función estándar de Python, la cual se utiliza también en pytest para realizar pruebas en el código y poder verificar si los resultados cumplen con lo esperado. Esta función es bastante útil ya que fácilmente se pueden probar funciones de Python y observar los resultados que se obtienen. A continuación, se muestra un código de ejemplo obtenido de [11]:

```
# content of test_assert1.py
def f():
    return 3

def test_function():
    assert f() == 4
```

Figura 1. Ejemplo de aplicación de la función assert en Pytest [11].

```

$ pytest test_assert1.py
===== test session starts =====
platform linux -- Python 3.x.y, pytest-7.x.y, pluggy-1.x.y
rootdir: /home/sweet/project
collected 1 item

test_assert1.py F [100%]

===== FAILURES =====
_____ test_function _____

    def test_function():
>         assert f() == 4
E         assert 3 == 4
E         + where 3 = f()

test_assert1.py:6: AssertionError
===== short test summary info =====
FAILED test_assert1.py::test_function - assert 3 == 4
===== 1 failed in 0.12s =====

```

Figura 2. Resultado obtenido del ejemplo de función assert en Pytest [11].

9) ¿Qué es Flake 8?

Esta es una librería de Python que contiene PyFlakes, pycodestyle y Ned Batchelder's McCabe script. La principal función de Flake8 es comparar el código de un programa con el PEP 8, el cual es un documento que contiene una serie de guías y recomendaciones sobre cómo utilizar el lenguaje de Python correctamente. Por lo tanto, se puede decir que esta librería ayuda a mantener el código ordenado, evita errores de sintaxis, revisa la complejidad de la lógica utilizada y se asegura de cumplir con los estándares del lenguaje [12].

10) Explique la funcionalidad de parametrización de pytest.

La parametrización en pytest es una herramienta que se utiliza cuando se desea probar el código con múltiples valores de entrada. El principal objetivo de esta funcionalidad es permitir verificar diferentes casos en una sola operación y así evitar generar líneas de código innecesarias y que se logre agilizar el proceso de prueba de errores y control de calidad de los softwares [13].

Referencias bibliográficas

- [1] J. Weirich, s.f, "Git Immersion". [En línea]. Disponible en: <https://gitimmersion.com/index.html>. [Consultado: 14 febrero 2023].
- [2] Tutorialspoint, s.f, "Git Fast Version Control". [En línea]. Disponible en: https://www.tutorialspoint.com/git/git_tutorial.pdf. [Consultado: 15 febrero 2023].
- [3] A. Baire, «GIT for Beginners,» 2 junio 2022. [En línea]. Disponible en: <https://people.irisa.fr/Anthony.Baire/git/git-for-beginners-handout.pdf>. [Consultado: 15 febrero 2023].
- [4] W3docs, s.f, "Git rebase". [En línea]. Disponible en: <https://www.w3docs.com/learn-git/git-rebase.html#definition-1>. [Consultado: 15 febrero 2023].

- [5] J. Wachira, 16 agosto 2022, "Git Rebase Origin/Branch vs. Git Rebase Origin Branch". [En línea]. Disponible en: <https://www.delftstack.com/howto/git/git-rebase-origin-master/#difference-between-the-git-rebase-originbranch-vs-git-rebase-origin-master-commands>. [Consultado 15 febrero 2023].
- [6] GitHub Docs, sf, "Resolving a merge conflict using the command line". [En línea]. Disponible en: <https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-using-the-command-line> [Consultado el 13 de febrero de 2023]
- [7] Knoldus Inc, 30 de noviembre de 2021, "How To Resolve Merge Conflicts In Git Pull Request (PR)?". [En línea]. Disponible en: <https://medium.com/@knoldus/how-to-resolve-merge-conflicts-in-git-pull-request-pr-e0b4901c9b79> [Consultado el 13 de febrero de 2023].
- [8] GitHub Docs, s.f, "Resolving merge conflicts after a Git rebase". [En línea]. Disponible en: <https://docs.github.com/en/get-started/using-git/resolving-merge-conflicts-after-a-git-rebase> [Consultado el 13 de febrero de 2023].
- [9] Python, 10 de febrero de 2023, "Infraestructura de tests unitarios". [En línea]. Disponible en: <https://docs.python.org/es/3.9/library/unittest.html> [Consultado el 13 de febrero de 2023].
- [10] Digital Guide IONOS, 14 de marzo de 2023, "Unit testing: la prueba de calidad para software". [En línea]. Disponible en: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-papel-del-unit-test-en-el-desarrollo-de-software/> [Consultado el 13 de febrero de 2023].
- [11] Pytest, s.f, "How to write and report assertions in tests". [En línea]. Disponible en: <https://docs.pytest.org/en/7.1.x/how-to/assert.html> [Consultado el 13 de febrero de 2023].
- [12] V. Freitas, 5 de agosto de 2016, "How to Use Flake8". [En línea]. Disponible en: <https://simpleisbetterthancomplex.com/packages/2016/08/05/flake8.html> [Consultado el 14 de febrero de 2023].
- [13] Pytest, s.f, "Parametrizing tests". [En línea]. Disponible en: <https://docs.pytest.org/en/7.1.x/example/parametrize.html> [Consultado el 13 de febrero de 2023].