

Tarea 1

GitHub, Pytest y Flake 8

Preguntas Teóricas (20 pts, 2pts c/u)

1) ¿Diferencie la herramienta Git de Github?

1. Git es un Sistema de Control de Versiones Distribuido (DVCS) utilizado para guardar diferentes versiones de un archivo (o conjunto de archivos) para que cualquier versión sea recuperable cuando lo desee. Generalmente, es mejor para uso individual.
2. GitHub es una plataforma basada en la web que incorpora las características de control de versiones de git para que puedan ser utilizadas de forma colaborativa. También incluye características de gestión de proyectos y equipos, así como oportunidades para la creación de redes y la codificación social.

<https://www.freecodecamp.org/espanol/news/git-vs-github-what-is-version-control-and-how-does-it-work/>

<https://kinsta.com/es/base-de-conocimiento/git-vs-github/>

2) ¿Qué es un branch?

En Git las Ramas son espacios o entornos independientes para que un Desarrollador sea Back-end, Front-end, Tester, etc. pueda usar y así trabajar sobre un mismo Proyecto sin chancar o borrar el conjunto de archivos originales del proyecto, dándonos flexibilidad para desarrollar nuestro proyecto de manera más organizada.

<https://blog.nubecolectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/>

3) ¿Cómo se crea un nuevo Branch?

Para generar un Branch se hace clic en la pestaña de código del repositorio y en el top de la lista se hace clic en la flecha que indica hacia abajo, luego se ingresa el nombre del branch en la caja de texto y finalmente hacer clic en el botón de crear Branch.

<https://docs.github.com/en/get-started/quickstart/hello-world>

4) ¿Qué es un commit?

Los commits están en el corazón del uso de Git. Puedes pensar en un commit como una captura de tu proyecto, donde se crea una nueva versión de ese proyecto en el repositorio actual. Dos características importantes de los commits son:

- puedes recordar los cambios a los que se les hizo commits en una fecha posterior, o revertir el proyecto a esa versión

- Si varios commits editan diferentes partes del proyecto, no se sobrescribirá entre sí aunque los autores de los commits no se conozcan entre sí. Este es uno de los beneficios de usar Git sobre una herramienta como Dropbox o Google Drive.

<https://www.freecodecamp.org/espanol/news/el-comando-git-commit-explicado/>

5) ¿Qué es la operación “git stash”?

El comando git stash almacena temporalmente (o guarda en un *stash*) los cambios que hayas efectuado en el código en el que estás trabajando para que puedas trabajar en otra cosa y, más tarde, regresar y volver a aplicar los cambios más tarde. Guardar los cambios en stashes resulta práctico si tienes que cambiar rápidamente de contexto y ponerte con otra cosa, pero estás en medio de un cambio en el código y no lo tienes todo listo para confirmar los cambios.

<https://www.atlassian.com/es/git/tutorials/saving-changes/git-stash>

6) Explique de forma gráfica: ¿Qué sucede en mi Branch local cuando hago “git rebase master”?

To rebase, make sure you have all the commits you want in the rebase in your master branch. Check out the branch you want to rebase and type `git rebase master` (where master is the branch you want to rebase on).

It is also possible to rebase on a different branch, so that for example a branch that was based on another branch (let's call it feature) is rebased on master:

```

      /---o-o branch
    /---o-o-o-o---o-o----- feature
----o--o-o-A----o---o-o-o-o--o--o- master

```

After `git rebase master branch` or `git rebase master` when you have checked out the branch, you'll get:

```

      /---o-o-o-o---o-o----- feature
----o--o-o-A----o---o-o-o-o--o--o- master
                        \---o-o branch

```

<https://www.freecodecamp.org/news/the-ultimate-guide-to-git-merge-and-git-rebase/>

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

Con la ayuda de unit test, es posible comprobar los componentes individuales de los programas informáticos. Estos test permiten examinar el correcto funcionamiento de cada uno de los elementos antes de que ocupen su lugar en el concepto general de un programa.

Además, ayudan a verificar de forma relativamente rápida y fácil si el componente funciona según lo previsto por el desarrollador. Los test unitarios son una de las formas más eficaces para descubrir la mayor cantidad posible de errores del código en las fases tempranas de desarrollo del software.

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-papel-del-unit-test-en-el-desarrollo-de-software/>

8) Bajo el contexto de pytest. ¿Qué es un “assert”?

assert() en testing

La función `assert()` puede ser también muy útil para realizar *testing* de nuestro código, especialmente para test unitarios o *unit tests*. Imaginate que tenemos una función `calcula_media()` que como su nombre indica calcula la media de un conjunto de números.

```
def calcula_media(lista):  
    return sum(lista)/len(lista)
```

En el mundo de la programación es muy importante probar o *testear* el software, para asegurarse de que está libre de errores. Gracias al uso de `assert()` podemos realizar estas comprobaciones de manera automática.

```
assert(calcula_media([5, 10, 7.5]) == 7.5)  
assert(calcula_media([4, 8]) == 6)
```

Por lo que si hacemos que estas comprobaciones sean parte de nuestro código, podríamos proteger nuestra función, asegurándonos de que nadie la “rompa”.

Sirve para comprobar las funciones

<https://ellibrodepython.com/assert-python>

9) ¿Qué es Flake 8?

“Linting” means running a basic quality tool against your code. The tool will check your code syntax and provide instructions on how to clean it.

There are a couple of good python code linter tools you can use. The one I’ve recently discovered is a **Flake8**. Which is “the wrapper which verifies pep8, pyflakes and circular complexity “. It has a low rate of false positives.

<https://medium.com/python-pandemonium/what-is-flake8-and-why-we-should-use-it-b89bd78073f2>
<https://flake8.pycqa.org/en/latest/>

10) Explique la diferencia entre un “log de error” y un “valor de error de retorno”

Un log de error es un registro donde se almacenan todos los errores posibles que se pueden presentar en el programa. Un valor de error de retorno identifica un error específico, ubicado en el log de error.