

Preguntas Teóricas (20 pts, 2pts c/u)

1) Diferencie la herramienta Git de GitHub

Git es un sistema de control de versiones que permite al usuario administrar y dar seguimiento al historial del código fuente de un proyecto, el cual se instala localmente en el dispositivo del usuario. Por su parte, GitHub es un servicio de alojamiento en la nube el cual le permite al usuario administrar repositorios Git con mayor facilidad junto a sus compañeros de trabajo con edición a tiempo real. Entonces, la principal diferencia radica en que Git es una herramienta que los desarrolladores instalan localmente para manejar código fuente, mientras que GitHub es un servicio en línea donde los desarrolladores que utilizan Git pueden conectarse y subir o descargar recursos.

2) ¿Qué es un branch?

Un Branch en Git representa una línea independiente de desarrollo. Los Branches sirven como abstracción para el proceso de edit/stage/commit, o en español, edición/ensayo/confirmación. Un Branch es una forma de solicitar un nuevo directorio de trabajo, área de ensayo e historial de proyecto. Básicamente entonces, un Branch es un puntero movable el cual apunta a algún commit; el nombre por defecto de un Branch base o principal es master.

3) ¿Qué es un commit?

En Git, commit es el término utilizado para guardar cambios. En este entorno se necesita correr el comando commit para mover los cambios del directorio del área de ensayo al repositorio local.

4) ¿Qué es la operación cherry-pick?

Cherry-pick es un comando en Git el cual permite seleccionar commits arbitrarios de Git por referencia y agregarlos al HEAD, es decir, al Branch activo o actual. En otras palabras, Cherry-pick será el comando que permite extraer un commit de un Branch y aplicarlo a otra.

Este comando se utiliza por ejemplo cuando se aplicó un commit al Branch incorrecto, en este caso se escoge el Branch correcto para luego aplicar Cherry-pick al commit y enviarlo donde pertenece.

5) ¿Qué hace el comando git stash?

El comando git stash se utiliza cuando se desea registrar el estado actual del directorio de trabajo y del índice, pero se desea volver a un directorio de trabajo limpio. Git stash se encarga de guardar las modificaciones locales y revierte el directorio de trabajo para que calce con el commit del HEAD.

6) ¿Compare las operaciones git fetch y git pull?

Ambos comandos se utilizan para descargar información de un repositorio remoto. El comando git fetch aquel que descarga, pero nunca integra estos datos descargados en los archivos de trabajo. Por su parte, git pull realiza la misma función de descargar, pero además copia esos cambios del repositorio remoto en el local. En términos más simples, git pull realiza un git fetch seguido de un git merge.

7) Asumiendo que usted está en un Branch llamado “secundario” y su Branch principal se llama “master”, ¿qué resultado espera de hacer git rebase master? ¿Qué resultado espera de hacer git rebase origin/master?

El commando git rebase hace referencia al proceso de mover o combinar secuencias de commits a un nuevo commit base. En este caso, al aplicar este comando se está cambiando la base del Branch de un commit a otro para hacer parecer como si se hubiera creado ese Branch de otro commit.

Cuando se aplica git rebase master, se mueve la base del Branch secundario hacia el final del Branch master tal y como se muestra a continuación:

```
A---B---C secundario
/
D---E---F---G master

git rebase master
          A'--B'--C' secundario
          /
D---E---F---G master
```

Acá se observa cómo el Branch secundario se mueve al final del Branch master.

Por su parte, al aplicar git rebase origin/master, se mezcla en el Branch solicitado (origin/master), y se aplican los commits que se han hecho localmente al final del historial sin crear un commit de mezcla. El Branch origin/master es una copia local del Branch master en el repositorio remoto llamado origin.

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

Es un tipo de testeo de software donde partes o componentes individuales de un software se ponen a prueba. Esto se realiza con el propósito de validar que cada parte del desarrollo del software funcione como se espera, donde se prueba tanto el correcto funcionamiento de este como el salto de errores en su debido momento.

9) Bajo el contexto de pytest. ¿Qué es un "assert"?

Las afirmaciones o "assertions", en inglés, son pruebas las cuales devuelven un resultado booleano, es decir, un estado verdadero o falso (True o False). Si una prueba de assert falla, la ejecución del programa se detiene y el resto del código no se corre. Para pruebas unitarias, o Unit tests, cada método o función debe tener un solo assert en su algoritmo, y se utilizan para confirmar que los resultados obtenidos son los esperados.

10) ¿Qué es Flake 8?

Es una librería de Python la cual encierra ciertas herramientas capaces de detectar errores de programación. Flake8 utiliza un método de comparación con un estilo de código conocido como PEP8, además de detectar errores de programación como lo podría ser una biblioteca importada sin utilizar, un nombre no definido, entre otros. También, se encarga de evaluar la complejidad ciclomática, la cual como su nombre lo indica, resalta la complejidad del programa. La complejidad ciclomática, es una métrica de software utilizada para medir el número de caminos independientes en el código fuente.

Referencias bibliográficas

- Alarcón, J. (2019). Git: los conceptos de "master", "origin" y "HEAD". En *Campus MVP*. Recuperado de: <https://www.campusmvp.es/recursos/post/git-los-conceptos-de-master-origin-y-head.aspx>
- Chacon, S., Straub, B. (2021). *Pro Git* (2ª). Recuperado de: <https://git-scm.com/book/en/v2>
- Chand, S. (2019). Git vs Github – Demystifying The Differences. En *Eureka*. Recuperado de: <https://www.edureka.co/blog/git-vs-github/>
- Gienow, M. (2018). Working with Branches in Git and GitHub. En *The New Stack*. Recuperado de: <https://thenewstack.io/dont-mess-with-the-master-working-with-branches-in-git-and-github/>
- Git Branch*. (n.d.) Recuperado de: <https://www.atlassian.com/git/tutorials/using-branches>
- Git Branches: List, Create, Switch to, Merge, Push, & Delete. (2020). En *noble desktop*. Recuperado de: <https://www.nobledesktop.com/learn/git/git-branches>
- Git Cherry Pick*. (n.d.) Recuperado de: <https://www.atlassian.com/git/tutorials/cherry-pick#:~:text=git%20cherry%2Dpick%20is%20a,be%20useful%20for%20undoing%20changes>.
- Git Fetch vs Pull: What's the Difference Between the Git Fetch and Git Pull Commands? (2019). En *FreeCodeCamp*. Recuperado de: <https://www.freecodecamp.org/news/git-fetch-vs-pull/#:~:text=git%20fetch%20is%20the%20command,changes%20from%20the%20remote%20repository>.

Git rebase. (n.d.) Recuperado de: <https://www.atlassian.com/git/tutorials/rewriting-history/git-rebase#:~:text=From%20a%20content%20perspective%2C%20rebasing,them%20to%20the%20specified%20base.>

Git vs Github: What's the Difference and How to Get Started with Both. (2021). En *Kinsta*. Recuperado de: <https://kinsta.com/knowledgebase/git-vs-github/>

Hossain, S. (2019). Git Fetch vs Git Pull. En *Medium*. Recuperado de: https://medium.com/@sabbirhossain_70520/git-fetch-vs-git-pull-691823ed4239

McKenzie, C. (2018). Git vs. GitHub: What is the difference between them? En *The Server Side*. Recuperado de: <https://www.theserverside.com/video/Git-vs-GitHub-What-is-the-difference-between-them>

Merging vs. Rebasing. (n.d.) Recuperado de: <https://www.atlassian.com/git/tutorials/merging-vs-rebasing>

Null, D. (2017). What is Flake8 and why we should use it? En *Medium: Python Pandemonium*. Recuperado de: <https://medium.com/python-pandemonium/what-is-flake8-and-why-we-should-use-it-b89bd78073f2>

PyTest Tutorial: What is, How to Install, Framework, Assertions. (n.d.). Recuperado de: <https://www.guru99.com/pytest-tutorial.html>

Senyuzluler, F. (2018). git – Rebase vs Merge. En *Medium*. Recuperado de: <https://medium.datadriveninvestor.com/git-rebase-vs-merge-cc5199edd77c>

Simplilearn. (2020). Git vs GitHub: What are the Major Differences. En *Simplilearn*. Recuperado de: <https://www.simplilearn.com/tutorials/git-tutorial/git-vs-github>

Steps of Saving Changes: Git Commit Command, Staging and Pushing. (2018). En *BitDegree*. Recuperado de: <https://www.bitdegree.org/learn/git-commit-command#:~:text=In%20Git%2C%20commit%20is%20the,the%20local%20repository%20of%20Git.>

Unit Testing Tutorial: What is, Types, Tools & Test EXAMPLE. (n.d.) Recuperado de: <https://www.guru99.com/unit-testing-guide.html>