

**Preguntas Teóricas (20 pts, 2pts c/u)****1) ¿Explique la principal utilidad de git como herramienta de desarrollo de código?**

Debido a que es un sistema de control de versiones, en el ámbito de desarrollo de código permite resguardar las versiones generadas de un proyecto y evitar el típico error causado al cambiar partes o funciones del mismo y que deje de funcionar/compilar, lo cual provee una manera de recuperar el programa funcional y partir de nuevo del punto seguro evitando dolores de cabeza durante la escritura de los códigos en desarrollo.

Además, gracias a sus características permite el desarrollo de funcionalidades nuevas a nivel de código para un proyecto de desarrollo, sin correr el riesgo de que los errores afecten una versión estable del mismo, por medio de la utilización de branches.

**2) ¿Qué es un branch?**

Es un espacio o entorno independiente para el desarrollo de código/archivos, con el fin de tener una mayor organización durante la realización de un proyecto.

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

Permiten aportar cambios a un repositorio de manera que los demás colaboradores puedan revisar y debatir los mismos y agregar confirmaciones previo a la incorporación del cambio propuesto en la estructura interna del mismo (branches).

**4) ¿Qué es un commit?**

Un commit representa la base principal del trabajo realizado en Git, ya que permite respaldar el trabajo realizado salvando las versiones generadas durante la construcción del proyecto.

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

Git fetch descarga los commits, archivos y referencias desde un repositorio remoto al repositorio propio.

Git rebase permite cambiar la base de un branch de una confirmación a otra haciendo parecer que fue creada desde ahí. En el caso de **origin/master**, el cambio de base ocurre desde el tracking branch of origin y se da el cambio de base al branch llamado de esta manera (el rebase se hace hacia **origin/master**).

**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.**

Estos conflictos ocurren cuando elementos ya existentes en la rama de archivos a los que se le hará merge o rebase fueron modificados, es decir, que el cambio en el código no es simplemente una adición de contenido, sino también una alteración de lo que ya existía.

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

Consiste en depurar el código en instancias específicas según cada funcionalidad, muy diferente a la práctica común de sencillamente correr un código entero mientras se usan prints para encontrar errores.

Las pruebas unitarias toman fragmentos de códigos (típicamente funciones “aisladas”) y los prueban al ingresar entradas y comprobando salidas, para así conocer el comportamiento práctico del código, optimizando recursos informáticos y simplificando la verificación de errores

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

Para no tener que interpretar “manualmente” si una prueba unitaria fue exitosa o no, en pytest se definen los “assert”, estos funcionan como verificadores de que condiciones específicas se cumplan.

Por ejemplo, si en determinado fragmento de código se debe obtener siempre un valor “True”, se define un assert que lo corrobore, de no cumplirse, el pytest indica que este assert falló, siendo que se esperaba un True pero en realidad hay un False.

**9) ¿Qué es Flake 8?**

Es una herramienta de análisis de código, a diferencia de pytest que analiza el código durante su funcionamiento, Flake 8 lo hace de manera estática, es decir, analiza aspectos como problemas de estilo, declaración de variables innecesarias, errores de indentación, etc.

**10) Explique la funcionalidad de parametrización de pytest.**

Esta permite plantear pruebas que se deben ejecutar con distintos conjuntos de datos de entrada, permite hacer pruebas más robustas al simular un mismo escenario bajo distintas condiciones, ya sea simples cambios en valores variables o incluso cambios en el tipo de variables.

## Referencias

- [1] "Git". Git. <https://git-scm.com/> (accedido el 2 de agosto de 2023).
- [2] Nube Colectiva. "Que son las ramas (branches) en Git, como utilizarlas y otros detalles". Blog de Programación y Desarrollo | Nube Colectiva. [https://blog.nubecollectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/#google\\_vignette](https://blog.nubecollectiva.com/que-son-las-ramas-branches-en-git-como-utilizarlas-y-otros-detalles/#google_vignette) (accedido el 2 de agosto de 2023).
- [3] GitHub Docs. "Acerca de las solicitudes de incorporación de cambios - Documentación de GitHub". GitHub Docs. <https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests> (accedido el 2 de agosto de 2023).
- [4] KeepCoding Bootcamps. "¿Qué es un Commit en Git?" KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-un-commit-en-git/> (accedido el 2 de agosto de 2023).
- [5] Atlassian. "Atlassian Git Tutorial". Atlassian. <https://www.atlassian.com/git/tutorials/syncing/git-fetch#:~:text=In%20review,%20git%20fetch%20is,of%20collaborative%20git%20work%20flows>. (accedido el 2 de agosto de 2023).
- [6] Stack Overflow. "git rebase origin" vs. "git rebase origin/master". Stack Overflow. [https://stackoverflow.com/questions/5963597/git-rebase-origin-vs-git-rebase-origin-master#:~:text=git%20rebase%20origin%20means%20"rebase,of%20origin%20to%20rebase%20with](https://stackoverflow.com/questions/5963597/git-rebase-origin-vs-git-rebase-origin-master#:~:text=git%20rebase%20origin%20means%20) (accedido el 2 de agosto de 2023).
- [7] freeCodeCamp.org. "The Ultimate Guide to Git Merge and Git Rebase". freeCodeCamp.org. <https://www.freecodecamp.org/news/the-ultimate-guide-to-git-merge-and-git-rebase/#:~:text=To%20rebase,%20make%20sure%20you,you%20want%20to%20rebase%20on> (accedido el 2 de agosto de 2023).
- [8] Atlassian. "Git merge conflicts". Atlassian. <https://www.atlassian.com/git/tutorials/using-branches/merge-conflicts> (accedido el 2 de agosto de 2023).
- [9] A. Feregrino. "Probando nuestro código con pytest". Blog Python Pytest. <https://old.tacosdedatos.com/pruebas-unitarias-pytest> (accedido el 2 de agosto de 2023).
- [10] G. Diaz. "Escribiendo código de alta calidad en Python". Medium. <https://medium.com/@gonzaloandres.diaz/escribiendo-codigo-de-alta-calidad-en-python-par-te-2-linters-64ffd8d2df91#:~:text=Flake8%20es%20una%20librer%C3%ADa%20de,para%20Overificar%20la%20complejidad%20ciclom%C3%A1tica>. (accedido el 3 de agosto de 2023).