

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

La principal utilidad de git es llevar el registro de los cambios en los códigos de forma distribuida, es decir, es un sistema de control de versiones (Gómez, 2022).

### **2) ¿Qué es un branch?**

Una Rama ("branch") según Gómez (2015), se refiere a que un conjunto de archivos puede ser ramificado o bifurcado en un punto en el tiempo de manera que, a partir de ese momento, dos copias de esos archivos se pueden desarrollar a velocidades diferentes o en formas diferentes de forma independiente el uno del otro.

Las ramas permiten desarrollar diferentes características, corregir errores, o experimentar con seguridad ideas nuevas en un área contenida del repositorio (GitHub, Inc., 2024a).

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

Un Pull Request es cuando uno de los colaboradores le pide a su equipo que revise y apruebe los cambios realizados a una "branch" antes de fusionarlos con la "main branch" (Vega, 2020). Al realizar esto, se permite identificar errores o realizar cambios sin alterar el código original. El Pull Request permite visualizar las diferencias entre ambas ramas (GitHub, Inc., 2024c).

### **4) ¿Qué es un commit?**

Tal y como explica Gómez (2015), un commit o "confirmar" es escribir o mezclar los cambios realizados en la copia de trabajo del repositorio. Los términos 'commit' y 'checkin' también se pueden utilizar como sustantivos para describir la nueva revisión que se crea como resultado de confirmar.

El comando de Git *git commit* es uno de los más importantes y más utilizados. Cuando se realiza un cambio en el proyecto y se ha probado a fondo, se puede utilizar este comando para crear una instantánea de su estado actual. Git considera esta instantánea como una versión segura y solo hará cambios en ella si se le indica explícitamente. Si se está trabajando con un repositorio local, git commit también creará allí la instantánea sin afectar a otros repositorios, pero permitiendo que se modifiquen con git push y git pull (Digital Guide Ionos, 2022b).

### **5) Describa lo que sucede al ejecutar la siguiente operación: "git cherry-pick <SHA>".**

Cuando se ejecuta, Git buscará el commit dado por SHA y aplicará los cambios realizados sobre la rama, creando un nuevo commit en esa rama; es decir, se copia un commit de una rama a otra (Git, 2023).

### **6) Explique que es un “merge conflict” y cómo lo resolvería.**

Los conflictos de fusión ocurren cuando se hacen cambios contrapuestos en la misma línea de un archivo o cuando una persona edita un archivo y otra persona borra el mismo archivo. Estos conflictos suceden cuando se fusionan ramas que tienen confirmaciones de cambios contrapuestas, y Git necesita ayuda para decidir qué cambios incorporar en la fusión final (GitHub, Inc., 2024b).

Para resolver un merge conflict se debe editar de forma manual el archivo conflictivo para seleccionar los cambios que se quieren mantener en la fusión final (GitHub, Inc., 2024b). El proceso se detiene antes de que se realice el git merge y se tiene la opción de resolver el conflicto manualmente, ya que usando el comando git status, se puede ver exactamente qué archivos se deben cambiar. Estos se pueden modificar, añadir con git add y ejecutar un nuevo git commit. Después, se podrá realizar un git merge para las ramas o branches de Git en cuestión, y eliminar las que ya no se necesiten (Digital Guide Ionos, 2022a).

Si el conflicto es ocasionado por cambios de líneas contrapuestos se puede resolver en GitHub usando el editor de conflictos. Mientras que para los otros tipos de conflictos de fusión, se deben resolver en un clon local del repositorio y subir el cambio a la rama en GitHub (GitHub, Inc., 2024b).

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

Las pruebas unitarias son aquellas que se encargan de comprobar que un fragmento de código funcione (KeepCoding, 2023). Estas son pruebas automatizadas y van a realizar una prueba independiente, es decir, ese segmento no depende de otras partes del código (Shaik, 2023).

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

La función del assert es realizar comprobaciones. En pruebas es especialmente utilizada en unittest, y ayuda a asegurarse de que el software está libre de errores (El libro de Python, 2024a).

### **9) ¿Mencione y explique 3 errores de formato detectables con Flake8?**

Según El libro de Python (2024b) algunos de los problemas más frecuentes suelen ser:

- Líneas demasiado largas: Se recomienda limitar el tamaño de cada línea a 79 caracteres, para evitar tener que hacer scroll a la derecha. Este límite también permite tener abiertos múltiples ficheros en la misma pantalla. Por otro lado se limita el uso de docstrings y comentarios a 72 caracteres.

- Uso incorrecto de espacios y líneas en blanco: el uso adecuado de espacios y líneas en blanco mejoran la calidad visual y la legibilidad del código, permitiendo un mejor entendimiento del mismo, sin embargo, un uso excesivo de ellas puede ser molesto y desorganizado. Algunas recomendaciones son: rodear las funciones y clases con dos líneas en blanco, dejar una línea en blanco entre los métodos de una clase, usar líneas en blanco para agrupar pasos similares y separar con una línea diferentes funcionalidades.
- Código mal indentado: Un bloque indentado se representa usando cuatro espacios y aunque el uso del tabulador pueda parecer lo mismo no se recomienda su uso. También se puede indentar el código para evitar tener líneas muy largas, que resultan difíciles de leer.

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

La funcionalidad de parametrizar es probar diferentes entradas y salidas con el mismo código de prueba (Vidal, 2023). Las pruebas parametrizadas son las pruebas basadas en datos, lo que permite hacer más pruebas con menor código (Ramkumar R, 2023).

## **Bibliografía**

- Digital Guide Ionos. (2022a). *Git branch merge: cómo volver a unir ramas*.  
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-merge-branch/>
- Digital Guide Ionos. (2022b). *Git commit: crear instantáneas en Git*.  
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/git-commit/>
- El libro de Python. (2024a). *Assert en Python*.  
<https://ellibrodepython.com/assert-python#assert-en-testing>
- El libro de Python. (2024b). *Python PEP8: Escribiendo Código Fácil de Leer*.  
<https://ellibrodepython.com/python-pep8>
- Git. (2023). *Git-cherry-pick - Apply the changes introduced by some existing commits*.  
Git-scm. <https://git-scm.com/docs/git-cherry-pick>
- GitHub, Inc. (2024a). *Acerca de las ramas*. Documentación de GitHub.  
<https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-branches>
- GitHub, Inc. (2024b). *Acerca de los conflictos de fusión*. Documentación de GitHub.  
<https://docs.github.com/es/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/about-merge-conflicts>
- GitHub, Inc. (2024c). *Acerca de los pull request*. Documentación de GitHub.  
<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/about-pull-requests>
- Gómez, J. (2022). *¿Qué es Git? Descubre la herramienta de control de versiones más popular*. Tokio School. <https://www.tokioschool.com/noticias/que-es-git/>
- Gómez, S. (2015). *INTRODUCCIÓN A GIT Y GITHUB - DÍA 1*. Universidad de Córdoba. pp.6.  
<https://www.uco.es/aulasoftwarelibre/wp-content/uploads/2015/11/git-cosfera-dia-1.pdf>
- KeepCoding. (2023, 3 de mayo). *¿Qué son las pruebas unitarias de software?* KeepCoding Tech School. <https://keepcoding.io/blog/que-son-las-pruebas-unitarias-de-software/>
- Ramkumar, R. (2023). *PyTest : Parameter and Fixtures*. Medium.  
<https://medium.com/@ramanish1992/pytest-parameter-and-fixtures-13f6fdbd48c9>
- Shaik, H. (2023). *Prueba unitaria con el módulo unittest de Python*. GEEKFLARE.  
<https://geekflare.com/es/unit-testing-with-python-unittest/>

- Vega, F. (2020). *Utilizando Pull Requests en GitHub*. Platzi.  
<https://platzi.com/clases/1557-git-github/19957-utilizando-pull-requests-en-github/>
- Vidal, P. (2023). *¡Pytest vs Unittest! La batalla definitiva: Descubre por qué Pytest conquista a los desarrolladores de Python*. El código Perfecto.  
<https://elcodigoperfecto.blog/2023/05/29/pytest-vs-unittest-la-batalla-definitiva-descubre-por-que-pytest-conquista-a-los-desarrolladores-de-python/>