

Tarea 1

Preguntas Teóricas

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

Git es un sistema de control de versiones distribuido lo cual lo caracteriza como uno de los sistemas con mayor uso en el trabajo de proyectos de programación personales o grupales dado su flexibilidad de manejar diversos cambios en el proyecto principal. Para un proyecto grande que necesite mucha atención en los detalles y existan varios desarrolladores trabajando en el mismo proyecto de debe cuidar una versión funcional o aquella que sea la base del trabajo, lo que se llama “master” como la rama principal de trabajo, posteriores modificaciones lo que implementan son copias con los nuevos cambios para evitar errores en el código funcional, en caso de que los cambios mejoren la implementación del proyecto completo entonces se fusiona con la rama principal.

2. Explique la diferencia entre git y github

Git es el sistema de control de versiones central el cual se maneja localmente en la PC del desarrollador y lleva cuenta del historial de cambios realizados a un proyecto en concreto, en el mismo el único que puede acceder es el desarrollador. En cambio, para GitHub es la herramienta que se basa en el uso de Git pero le añade funcionalidades remotas para el trabajo conjunto de un equipo de programación o guardar el proyecto en la nube.

3. ¿Qué es un branch?

Como se mencionó en Git se crea una rama principal llamada master el cual es el código base o funcional que el desarrollador elige y define como tal. Para mantener el orden y control en los cambios que se añaden al código Git hace una replica exacta del repositorio en una nueva rama que sale del master, ello con la finalidad de que si los

cambios implementados no resultan como lo esperado y crean error en el código, siempre exista la posibilidad de volver atrás y mantener un punto de inicio viable del proyecto. Si el cambio fue exitoso entonces Git vincula el código en conjunto con el master para unificar la mejora.

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

Es un medio por el cual uno de los desarrolladores al realizar cambios o mejoras de su rama pide a los demás integrantes del proyecto que revisen los cambios y verifiquen la correcta funcionalidad del código antes de unificarla con el master.

5. ¿Qué es un commit?

Un commit es la forma de conseguir guardar los pequeños cambios a través del código para ir teniendo un historial de las cosas que se han ido haciendo y para dar la posibilidad de devolverse en algún punto si se llegara a necesitar. Todas las acciones para confirmar el cambio deben ser realizadas por el desarrollador, primero debe añadir con el comando `git add` los cambios realizados al código, luego realizar el commit en el código desde su PC con el comando `git commit -m` para agregar un texto descriptivo de los cambios o cosas agregadas y finalmente vincular con el servidor remoto con el comando `git push`. Todo lo que se haya realizado aparecerá en la cuenta del desarrollador.

6. Describa lo que sucede al ejecutar la siguiente operación: “git rebase main”.

Se trata de un medio por el cual se reorganiza el origen de donde se creó una rama en particular, llevándola del punto de creación original al principio del master si así se desea, bajo diferentes comandos se tiene flexibilidad en hacia adonde se mueve la rama. Al final de la ejecución del comando se visualiza la rama en un origen diferente de creación haciendo que mejore la visualización del historial y orden del proyecto.

7. Explique que es un “merge conflict” y como lo resolvería.

Son las ocasiones en las que, al tratar de unificar ramas del código, Git detecta errores que no puede solucionar sin antes ordenar el espacio de trabajo como por ejemplo cambio en las líneas de código de un mismo archivo o bien que no encuentre un archivo. Es recomendable que primero se resuelva el conflicto de forma local antes de llevarlo a GitHub.

Para solucionarlo se debe buscar el/los archivos en conflicto esto por medio de comandos como `git status` para observar el error y donde corregirlo. Luego buscar el archivo solucionar el error manteniendo el código o cambiando por lo que hay en la nueva rama y finalmente agregar un commit para conservar los cambios. Con ello se puede resolver el merge conflict y finalizar la unificación entre ramas.

8. ¿Qué es una Prueba Unitaria o Unittest en el contexto de desarrollo de software?

Es una forma de mantener el código en constante revisión puesto que se realiza de forma automática, esto se logra al mantener pequeñas partes de este con una prueba o test para verificar su correcta ejecución. Si al escribir el código por cada pequeña parte realizada se crea su unidad de prueba se facilita saber dónde ocurrió un error y deja espacio para mejorar y ampliar la programación y evitar el tiempo en búsqueda de fallos.

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

Es una palabra reservada para indicar que una condición debe cumplirse al realizar un test de un código, si estas condiciones son verdaderas el test se ejecuta exitosamente y si no, da como resultado un reporte de error.

10. Mencione y explique 3 errores de formato detectables con Flake8

- E501: Line too long

Este error de formato se genera cuando una línea de código tiene una extensión mayor a 79 caracteres.

- E302: Expected 2 blank lines, found 1

Este aparece cuando hay dos líneas en blanco entre las definiciones de clases o funciones.

- E225: Missing whitespace around operator

Se produce cuando no hay un espacio en blanco alrededor de un operador.