PARTE 3 Preguntas abiertas

Pregunta 1: ¿Cuál es la diferencia entre pruebas de carga y estrés?

* Pruebas de carga: Se enfocan en evaluar el rendimiento del sistema bajo una carga esperada para determinar si puede manejar la carga prevista durante un período de tiempo prolongado.
* Pruebas de estrés: Se centran en llevar al sistema más allá de sus límites normales para evaluar su estabilidad y ver cómo se comporta bajo condiciones extremas o de sobrecarga.

Pregunta 2: ¿Qué probarías si tuvieras que probar una nueva versión que solventa un error crítico en producción?

Probaría específicamente la funcionalidad relacionada con el error crítico solucionado para garantizar que se haya corregido correctamente y que no se hayan introducido nuevos problemas. Además, se deberían realizar pruebas de regresión para asegurarse de que la corrección no haya afectado otras áreas del sistema.

Pregunta 3: ¿Cómo reportarías los errores encontrados que hacían los desarrolladores?

* Proporcionaría una descripción clara y detallada del error, incluyendo pasos para reproducirlo y cualquier información relevante.
* Asignaría una severidad y prioridad adecuadas al error.
* Proporcionaría capturas de pantalla o registros si es necesario para ayudar en la reproducción o comprensión del problema.

Pregunta 4: ¿Cómo le darías seguimiento a errores reportados a Desarrollo?

Utilizaría un sistema de seguimiento de problemas como Jira o Asana para registrar y rastrear el estado de los errores. Esto incluiría asignar el error a un desarrollador, establecer una fecha objetivo para su resolución y realizar un seguimiento del progreso hasta que se haya corregido y verificado.

Pregunta 5: ¿Cuáles serían los niveles de severidad que manejarían para reportar errores y con base en qué?

Generalmente se usan los siguientes:

* Crítico: Impide el uso fundamental del sistema o una función importante.
* Mayor: No impide el uso fundamental del sistema, pero afecta significativamente la funcionalidad.
* Moderado: Problemas que no afectan críticamente la funcionalidad pero que deben solucionarse.
* Mínimo: Problemas estéticos o de presentación que no afectan la funcionalidad.

Pregunta 6: ¿Puedes explicar qué es la metodología de pruebas Ágil y cuáles son sus principios fundamentales?

La metodología de pruebas Ágil se centra en pruebas continuas e iterativas dentro del marco de desarrollo ágil. Sus principios fundamentales incluyen:

Adaptabilidad y respuesta al cambio.

Colaboración entre equipos de desarrollo y de prueba.

Enfoque en la entrega temprana y frecuente de software funcional.

Pruebas continuas durante todo el ciclo de vida del desarrollo.

Pregunta 7: ¿Cuál es la diferencia entre pruebas funcionales y pruebas no funcionales? Proporciona ejemplos de cada una.

* Pruebas funcionales: Verifican que el software cumpla con los requisitos funcionales, es decir, que realiza lo que se espera que haga. Ejemplo: Pruebas de registro de usuario en un sitio web.
* Pruebas no funcionales: Evalúan aspectos no relacionados con la funcionalidad, como el rendimiento, la usabilidad y la seguridad del software. Ejemplo: Pruebas de carga para evaluar el rendimiento del sistema bajo una carga determinada.

Pregunta 8: ¿Qué es la automatización de pruebas y cuáles serán algunos casos donde sería más beneficioso automatizar pruebas en lugar de realizar pruebas manuales?

La automatización de pruebas es el proceso de utilizar herramientas y scripts para ejecutar pruebas de forma automatizada.

Es beneficioso en casos donde: Se requieren pruebas repetitivas, se necesita realizar pruebas en múltiples configuraciones, se desea realizar pruebas de regresión de manera eficiente y se necesita ejecutar pruebas complejas que serían difíciles de realizar manualmente.

Pregunta 9: ¿Puedes mencionar algunas herramientas populares de automatización de pruebas y describir en qué escenarios podrían ser más útiles?

Algunas herramientas populares de automatización de pruebas incluyen Selenium WebDriver para pruebas web, Appium para pruebas móviles, JUnit y TestNG para pruebas unitarias en Java, y Cucumber para pruebas de comportamiento basadas en BDD

Pregunta 10: ¿Qué casos no automatizarías?

No automatizaría casos de prueba que son altamente volátiles y propensos a cambios frecuentes, como pruebas exploratorias o pruebas de usabilidad. También evitaría automatizar pruebas que requieren una evaluación subjetiva o humana, como pruebas de diseño o pruebas de accesibilidad. Además, no automatizaría pruebas que son de bajo valor o que solo se ejecutan una vez.