1. ¿Qué entiendes por el término pruebas funcionales?

R/: Las pruebas funcionales son aquellas que buscan garantizar que el producto software funcione como lo espera el cliente o los stakeholders (Interesados)

1. ¿Cuáles son los pasos importantes que se deben cubrir en las pruebas funcionales?

R/: Las pruebas funcionales se debe tener en cuenta todo el ciclo de vida de un proyecto de software desde la planificación (se priorizan los requisitos de la prueba, se realiza el plan de pruebas y se realiza el debido contexto de todo lo que tiene que ver en el proyecto) en la preparación y diseño del plan de pruebas donde se consigna todo lo relacionado con la descripción de la necesidad que se va a probar, el alcance que vamos a tener en las pruebas, la estrategia que vamos a abordar para realizar las pruebas, los pre requisitos o entradas que vamos a tener en cuenta para certificar la funcionalidad y los supuestos que vamos a tener en la certificación; luego en la fase de ejecución se lleva a cabo la ejecución de los casos de prueba que diseñamos anteriormente, teniendo en cuenta que los errores los debemos reportar al equipo de desarrollo y los impedimentos que se presenten en ésta etapa deben ser reportados al líder del producto para que se realice el debido trámite (si no puede ser solucionado por el), todo lo que se lleve a cabo en esta etapa debe ser documentado y reportado para tener claro el panorama completo de la ejecución.

1. ¿Cuál es la diferencia entre pruebas funcionales y no funcionales?

R/: La gran diferencia entre estos dos tipos de pruebas es que las pruebas funcionales buscan comprobar las funciones de la aplicación y validan que funcionen como se espera, mientras que las no funcionales son pruebas que buscan comprobar el riesgo que corre el producto y nos arroja resultados como el desempeño o un bajo rendimiento que puede afectar al producto en el ambiente de producción.

1. Define en tus propias palabras qué es un release.

R/: Un release es una línea de tiempo donde se lleva a cabo el ciclo de vida de desarrollo de un producto software, en el cual pueden venir incluidos varios Sprint, por ello es importante tener en cuenta que un release tiene un periodo de tiempo mas largo que un Sprint (uno o cinco meses).

1. Escribe una lista de los posibles estados que puede tener un defecto (escribe una pequeña definición de cada uno).

R/: Un defecto puede tener los estados:

Abierto: Es donde se realiza la localización del defecto y se documenta para que la persona que va a darle solución replique el caso nuevamente.

En revisión: La persona que va a brindarle solución al defecto se encuentra revisandolo y replicandolo.

Resuelto: El defecto se ha solucionado y se re asigna al QA para que realice la validación.

Rechazado: No es posible replicar el error y no se vuelve a presentar.

Reabierto: Después de la solución de la persona encargada, la solución no es correcta y el defecto se sigue presentando.

Cerrado: La persona encargada de probar el defecto, vuelve a realizar la prueba y verifica que el defecto fue solucionado

1. ¿Cuáles puntos consideras importantes a la hora de escribir un caso de prueba?

R/: Considero que es importante escribirlos en un lenguaje amigable como Gerkhin para que las personas que lo leen sepan de que se trata el caso de prueba, ademas de eso es importante documentarlo de la mejor manera y con la mayor cantidad de información posible; se debe agregar un título que sea muy disiente y que nos indique de que se va a tratar el caso de prueba, además se debe asignar a la persona que va a llevar a cabo la prueba, el estado actual del caso de prueba y estar asignado a la historia de usuario que tiene relación.

1. ¿Cuáles puntos consideras importantes a la hora de documentar un defecto?

R:/ Para documentar un defecto se debe tener en cuenta varios aspectos tales como: un título igualmente disiente que nos indique que tipo de defecto se está presentando, una descripción completa del defecto presentado, la manera que se inyectó, de que manera se detectó, que funcionalidad afectó, su prioridad y la severidad que tiene el defecto, además de un paso a paso con sus respectivas evidencias, donde nos muestra como se replica el error.

1. ¿Qué diferencias hay entre probar la respuesta de un API y el esquema de respuesta de un API?

R/: La respuesta de una API se refiere al status en el cual se encuentra la API, en el momento de llevar a cabo el envío la API respondería el stado en el cual se encuentra tal como un 200 o de error, si es posible acceder o no a la misma; el esquema de respuesta de la API depende de lo que le solicite por ejemplo si le solicito una cierta cantidad de datos se me entrega una serie de campos y responde con un Json con los diferentes campos solicitados, siempre y cuando el status de la API se encuentre en estado 200.