

## PRUEBA TÉCNICA MUNCHER

1. Describe con tus propias palabras o mediante ejemplos el proceso de ingeniería de software ideal, desde la captura del requerimiento hasta la salida a producción, de acuerdo a su experiencia.

- Para iniciar el proceso de desarrollo se debe partir de los requerimientos que es donde especificamos el comportamiento que esperamos del sistema y para el cual se está trabajando . Estos los define en un inicio el equipo de producto, quienes deben estar acompañados de diseño para la creación de toda la parte gráfica que se requiere. Al tener los requerimientos y el diseño listo se procede a tener una sesión de pre planning con integrantes del equipo de tecnología para la definición de las tareas, especificar que es posible, de qué manera se puede hacer, dejar creadas las tareas, dejar claros los requerimientos de cada una de las tareas y criterios de aceptación.
- Al tener las tareas creadas se debe proceder a tener la sesión de planning con los involucrados tanto de tecnología como de producto y diseño, en esta sesión se comparten las tareas, se asignan las tareas a quien corresponda y se procede a realizar la correspondiente estimación de cada tarea, estas se deben estimar por tiempo también, en esta sesión ya estando todos los involucrados en el sprint también se suelen aclarar dudas sobre las tareas.  
Dentro de cada una de las tareas existe una sub tarea de qa, que es donde qa estima y hace el seguimiento del estado de su trabajo sobre la tarea principal.
- Luego de la sesión de planning , basado en lo tratado en las anteriores sesiones y sobre todo en los criterios de aceptación, qa debe crear la suite de casos de prueba que se ejecutarán dentro de ese sprint, sin importar la herramienta cada caso debe estar conectado a la tarea que corresponda para la visualización del equipo así poder recibir feedback, si es necesario agregar, cambiar o quitar casos .
- Viene el tiempo de desarrollo, donde se realiza la implementación de todo lo anteriormente estipulado, se desarrollan las tareas y se realiza integración de ser necesario.
- Al estimar por tiempos es claro cuando van a salir las tareas a pruebas, y una vez éstas empiezan a salir a pruebas, QA inicia la ejecución de los casos de prueba en ambiente de pruebas. Durante el proceso de pruebas se realiza el proceso de verificación de los criterios, se realiza el reporte de bugs, estos deben ser reportados dentro del sprint y se deben asignar según corresponda la tarea, también podría realizarse el reporte por medio de la herramienta de gestión de casos de prueba ya que esta genera un reporte más completo y crea el bug conectado al sprint también, todos los bugs deben quedar documentados para el reporte de la ejecución.  
Al reportar el bug, según su prioridad se debe esperar la solución de este y volver revisar para confirmar que este se haya solucionado.

- Luego de que las tareas pasan el proceso de pruebas se despliegan las tareas a ambiente de preproducción o stage, dependiendo lo que requiera la empresa puede existir un proceso de uat o no, si este llega a existir, en este se valida que todo haya quedado como se estipulo.
- Tras finalizar el proceso de QA de todas las tareas, se debe realizar un día de regresión en ambiente de pre producción o stage para la verificación de todas las tareas y el funcionamiento del sistema en general, si en este proceso se encuentran bugs estos también deben quedar documentados.
- Luego de la regresión se realiza el paso a producción, se hace un flujo en producción para ver que el despliegue haya sido exitoso.
- Luego de la salida a producción, cada determinado tiempo se corre el set automatizado para validar el correcto funcionamiento del sistema.

**Nota:**Proceso basado en scrum

2. Conforme a la respuesta anterior identifique las principales actividades en las cuales se puede intervenir como analista de calidad.

-El analista de calidad puede estar involucrado en actividades tales como:

- La preplanning y la planning: en estas se definen las tareas y es de vital importancia que esté al tanto del proceso, la especificación de todo para poder crear correctamente los casos de prueba.
- La estimación de las tareas: Las tareas se deben estimar teniendo en cuenta el tiempo de qa ya que se puede tener más claro el scope del sprint.
- La creación de casos de prueba: Este proceso es responsabilidad del QA.
- Ejecución de casos de prueba: La ejecución de los casos es responsabilidad del QA.
- Reporte de bugs: El reporte, creación y réplica de los bugs, es responsabilidad del QA.
- Uat: En caso de que haya un uat QA se prueba involucrar en la creación de los bugs (si se encuentran) y la gestión de control de cambios en caso de que sea necesario.
- Regresión: El proceso de regresión está a cargo de qa ya que se ejecutan los casos de regresión y si existe la suite automatizada de regresión.
- Prueba del flujo: Tras el despliegue QA realiza una prueba de un flujo principal para la validación de que todo se desplegó bien.
- La automatización: El QA esta involucrado en la creación de los scripts automatizados y la ejecución de estos.

3. ¿Qué es lo más importante a la hora de reportar un defecto (bug) y cómo maneja la prioridad?

-A la hora de reportar un bug se tienen que tener varios factores en cuenta.

- Principalmente se debe replicar para saber qué está pasando realmente y para saber cómo se replica ese bug.
- Reportar y documentar el bug, tras saber cómo replicarlo se debe reportar y para esto se debe documentar correctamente el bug, se escribe que está pasando, cómo se replica y que debería estar pasando, se deja evidencia de foto o video si es necesario.

Ejemplo:

-¿Qué sucede?

Al iniciar sesión arroja error de contraseña así esta sea correcta

-¿Cómo replicarlo?

Ir a la pagina abc.com

Escribir correo en el campo que lo indica

Escribir contraseña en el campo que lo indique

Dar click en iniciar sesión

-¿Qué debería estar sucediendo?

Al poner correo y contraseña correcta debería poder iniciar sesión sin problema.

- Luego de escribir correctamente el bug se debe priorizar, como podemos hacer esto, usualmente hay toda una documentación sobre cómo hacerlo, pero podemos basarnos en que un bug de **prioridad baja** puede ser uno que sea estético y no sea de cara al cliente, ejemplo: en un crm la fuente es de una tonalidad distinta a la que debería, podemos vivir con él por un rato.

Un bug de **prioridad media** pueden ser estos errores que no son bloqueantes pero son de cara al cliente, ejemplo: hay una palabra mal escrita, no bloquea el sistema pero el cliente lo ve, se debe solucionar pronto.

Un bug de prioridad alta es aquel que bloquea una funcionalidad de importancia, ejemplo: el cliente no puede iniciar sesión y sus credenciales están bien, se debe solucionar de inmediato.

Estas priorizaciones son también teniendo en cuenta que tan costoso es tener estos bugs en el sistema, tomando en cuenta los mismos ejemplos, en prioridad baja no nos está costando mucho, en prioridad media nos cuesta pero no es demasiado, prioridad alta, es costoso, podemos perder clientes.

4.¿Qué diferencia hay entre las técnicas de transición de estados y el análisis de valor de borde? De dos ejemplos donde lo haya aplicado en su experiencia laboral explicando el problema y cómo aplicó estas técnicas.

-La técnica de transición de estado está basada en crear los casos para ejecutar transiciones basadas en requerimientos del sistema, por ejemplo:

- Tengo un sistema donde se realiza una consulta en datacrédito al ingresar la cédula, esto para la aprobación de un crédito, una vez esta cédula esté consultada se deben esperar 30 días para volver a consultar, y para que pase la solicitud exitosa la cédula debe estar sin consultar.

Tomamos una cédula sin consultar y hacemos la prueba, en primera instancia la solicitud pasa y es aprobado el crédito. Luego volvemos a realizar la prueba con la misma cédula y este crédito será rechazado por consultas.

En este ejemplo la transición que generó el cambio de estado fue el consultar la cédula con otro crédito, entonces por las reglas del sistema esta solicitud fue rechazada aunque se repitieron los mismos pasos y ambos casos deben ser validados siempre.

-A diferencia de la transición de estado, el análisis de valor border que se basa en crear los casos con los datos límites de cada un requerimiento, un valor de entrada o de salida que podemos ver en el siguiente ejemplo:

- Tenemos una calculadora para ver el monto que pueden aprobarse en un crédito, con esta calculadora se pueden poner valores desde 5.000.000 hasta 50.000.000.

Creamos el caso para validar que el mínimo monto que se pueda poner sea el de 5.000.000 y el máximo monto sea 50.000.000 .

Serían los casos límite, el monto máximo de la calculadora, y el monto mínimo de la calculadora, se validan ambos casos confirmando que si sean los valores correctos como minimo y maximo.

-Teniendo esto claro, la diferencia entre estas dos técnicas es que con la transición de estado nos enfocamos en los casos que pueden tener diferentes estados dependiendo las reglas del sistema, mientras que el análisis de valor border se enfoca en los casos límite de una funcionalidad, como en el ejemplo el máximo o el mínimo valor de la calculadora.

**Nota:** Ejemplos basados en mi experiencia probando el sistema de car finance en grupo r5 , link: <https://creditos.grupor5.com/car-finance/>

5. Se requiere probar una calculadora de tasas de interés para un depósito a término fijo. Para un valor dado de depósito y una edad del cliente, la calculadora muestra la tasa de interés anual.

Aplican las siguientes reglas: ✓ El valor mínimo de depósito es \$6.000.000. ✓ El valor máximo de depósito es de \$60.000.000. ✓ La tasa de interés anual depende del valor de depósito de acuerdo a las siguientes reglas: - \$6.000.000 - \$10.000.000 : 1.0% - \$10.000.001 - \$20.000.000 : 1.5% - \$20.000.001 - \$60.000.000 : 2.1% ✓ Solo adultos (mayores de 18 años) son elegibles para tomar un depósito a término fijo. ✓ Adultos mayores a 62 años tienen una tasa fija del 1.8%. Defina los casos de borde de prueba para este sistema.

#### **Casos de prueba:**

**Escenario:** Validar los montos máximos y mínimos del depósito

Dado como cliente mayor de 18 años

Cuando ingreso a la calculadora

Y pongo como monto 6 millones

Entonces me mostrara la tasa de interés

Dado como cliente mayor de 18 años

Cuando ingreso a la calculadora

Y pongo como monto menos de 6 millones

Entonces no me mostrara la tasa de interés

Dado como cliente mayor de 18 años  
Cuando ingreso a la calculadora  
Y pongo como monto 60 millones  
Entonces me mostrara la tasa de interés

Dado como cliente mayor de 18 años  
Cuando ingreso a la calculadora  
Y pongo como monto más de 60 millones  
Entonces no me mostrara la tasa de interés

**Escenario:** Validación de edad del cliente  
Dado como cliente que ingresó a la calculadora  
Cuando mi edad es 18 años  
Y pongo un monto dentro del rango  
Entonces me mostrara la tasa de interés

Dado como cliente que ingresó a la calculadora  
Cuando mi edad es 17 años  
Y pongo un monto dentro del rango  
Entonces no me mostrara la tasa de interés

Dado como cliente que ingresó a la calculadora  
Cuando mi edad es 63 años  
Y pongo un monto dentro del rango  
Entonces me mostrara la tasa de interés

**Escenario:** Validación de la tasa de interés  
Dado como cliente que ingresó a la calculadora  
Cuando pongo un monto entre \$6.000.000 - \$10.000.000  
Entonces la tasa de interés que arroja será de 1.0 %

Dado como cliente que ingresó a la calculadora  
Cuando pongo un monto entre \$10.000.001 - \$20.000.000  
Entonces la tasa de interés que arroja será de 1.5%

Dado como cliente que ingresó a la calculadora  
Cuando pongo un monto entre \$20.000.001 - \$60.000.000  
Entonces la tasa de interés que arroja será de 2.1%

Dado como cliente que ingresó a la calculadora  
Cuando pongo un monto entre \$6.000.000 - \$60.000.000  
y soy mayor a 62 años  
Entonces la tasa de interés que arroja será de 1.8%

**Nota:** Casos escritos con Bdd, número de casos **11**.

6. ¿Qué herramientas utiliza para revisar los apis y explique el por qué la utiliza?

- Para probar apis he probado herramientas como postman, esta herramienta a la hora de realizar la prueba es muy amigable en cuanto interfaz, es de fácil uso y los reportes son claros, por varios meses me sirvió sin embargo no me parece la herramienta adecuada para tener toda una suite automatizada, la uso frecuentemente para algo rápido ya que es bastante fácil de usar.
- Para automatizar un proyecto más grande he usado rest assured maven, con cucumber porque este me permite hacer peticiones mucho más claras para el lenguaje que manejaba en ese proyecto (java), rest assured ofrece muchas posibilidades de pruebas con respuestas en json, es muy interesante y más la combinación con bdd.
- También automaticen algunas apis con insomnia para graphql, la interfaz es amigable y se pueden mapear múltiples respuestas.

7. ¿Qué herramienta utiliza para realizar pruebas de carga y de estrés?

- Durante un tiempo estuve muy involucrada con pruebas de carga por lo que realice pruebas de concepto de varias herramientas tales como, load ninja, test studio, loadui, loadview y jmeter. teniendo un claro panorama mi inclinación fue hacia jmeter porque los reportes eran claros, la interfaz es amigable, es rápida, pude realizar pruebas de rendimiento y analizar diferentes cargas sobre un servidor, tuve la oportunidad de usarla sobre mongodb y me fue muy bien revisando cantidades de solicitudes extremas para ver cuánto soportaba la base de datos en poco tiempo.

8. ¿Qué metodología ha utilizado para la creación de casos de prueba?

- He escrito casos de prueba step by step, pero me ha ido mejor escribiendo estos casos en bdd, así nos centramos en que los casos van enfocados a la necesidad del cliente o quien requiera el software, estos casos se crean según funcionalidad y usabilidad, depende el producto y lo que requiera en ese momento ya que se realizan casos para el happy path, para los casos de error, para los casos de seguridad etc...

La estructura de estos casos es el escenario donde podemos poner el id de la tarea o la descripción general, Given (dado), en este punto nos ponemos en el estado correcto para ejecutar una acción, When (cuando), en este paso escribimos la acción a realizar, Then(entonces) que es la salida, allí describimos lo que sucede tras la acción, también podemos ver un and(y) que son condiciones que podemos introducir luego o después de las acciones, sobre esto hay ejemplos en la escritura de los casos de prueba en el punto 5.

9. Si has automatizado ¿Qué herramientas has utilizado para realizar pruebas automatizadas, con qué frameworks y por qué?

- Par la automatización de pruebas llevo 3 años trabajando con selenium, es una herramienta espectacular, muy versátil, he automatizado múltiples sistemas con selenium con lenguajes como java y python, en proyectos maven y gradle, para la ejecución junit, pytest y testng y en cuanto patrones para la arquitectura del proyecto

suelo usar pom(Page object model), porque siento que la estructura del proyecto es clara para cualquier persona que vea el código, pero también conozco sobre screenplay.

En cuanto frameworks, conozco sobre serenity, katalon pero este no me gusta mucho ya que es una grabación y mantener el código se vuelve tedioso y también conozco sobre cypress que es una buena herramienta pero por la naturaleza de los lenguajes que manejo me va mejor con selenium.