

Componente Teórico

Definir los siguientes conceptos desde su punto de vista y experiencia:

- ¿Cómo es el proceso de desarrollo de software utilizando metodologías tradicionales y ágiles?

El proceso de desarrollo de software define la forma como se va a desarrollar el software; consiste en dividir el trabajo que se va a realizar en fases o etapas. Es también conocido como ciclo de vida del software.

El Proceso de desarrollo de software con metodologías tradicionales. Se caracteriza por usar modelos en cascada, incremental, espiral, etc. que sigue un enfoque lineal y secuencial donde cada etapa (planificación, análisis, diseño, implementación, pruebas, paso a producción.) no se puede pasar a la siguiente sin terminar la anterior.

Por otro lado, **El Proceso de desarrollo de software con metodologías ágiles.** Se caracteriza por usar modelos como scrum, Kanban, XP, etc., que se hace de forma ligera, ya que dichos modelos son iterativos e incrementales que permiten entregar poco a poco el valor al cliente (producto mínimo viable) en sus diferentes etapas (Planificación, Requerimientos priorizados, Iteración, paso de producción)

- ¿Cuál es el rol de analista de calidad de software en el proceso de desarrollo de software en proyectos donde se utilice metodologías ágiles?

El rol de un analista de calidad en metodologías ágiles es estar presente durante todo el proceso, para asegurar la calidad del producto que se está desarrollando. Así mismo, debe trabajar de la mano con el cliente para ayudar a definir los requisitos y objetivos de las pruebas; y con el equipo de desarrollo para validar el producto antes de que este llegue al usuario final.

- ¿Cuál es la principal diferencia entre:

A. Verificar y Validar

Verificar: Se asegura de que el software se esté construyendo correctamente, según las especificaciones y estándares técnicos establecidos.

Validar: Se asegura de que el software cumpla con las necesidades y expectativas reales del usuario final o cliente.

B. Issue y Bug.

Issue: Es cualquier tipo de problema, tarea o solicitud en un proyecto de software. Puede incluir mejoras, nuevas características, problemas de rendimiento, entre otros.

Bug: Es un problema específico en el software que causa un comportamiento incorrecto. Ocurre debido a errores en el código o malentendidos en los requisitos, causando errores en la funcionalidad del programa.

C. Novedad y Hallazgo

Novedad: Es algo nuevo y positivo que se añade al software, como una nueva característica o mejora planeada.

Hallazgo: Es un problema o error identificado en el software que necesita ser corregido, como un bug o un fallo de rendimiento.

Explique e indique cuáles son los entregables de las siguientes etapas:

- A. **Planeación:** En esta etapa, se planifica cómo se llevarán a cabo las actividades de aseguramiento de calidad a lo largo del proyecto.

Entregables: Plan específico de pruebas, Cronograma.

- B. **Diseño de Escenarios y Casos de Prueba:** En esta etapa se definen los casos de prueba específicos que se utilizarán para validar el comportamiento del software según las necesidades establecidas previamente.

Entregables: Documento de Especificaciones de Casos de Prueba, Matriz de Rastreo de Requisitos, Escenarios de Prueba

- C. **Ejecución:** Durante esta etapa se ejecutan los casos de prueba diseñados para validar el comportamiento del software.

Entregables: Registro de Ejecución de Pruebas, Informes de Ejecución de Pruebas

- D. **Reporte de Hallazgos:** Después de ejecutar las pruebas, se documentan y comunican los problemas encontrados en el software.

Entregables: Informe de Hallazgos.

- E. **Feedback:** En esta etapa se recibe la retroalimentación del proceso de prueba.

Entregables: Informe Final.

- ¿Indique con qué herramientas de gestión de pruebas ha trabajado y cuál es la de su preferencia y porque?

He usados las herramientas Azure y Mantis, de las 2 prefiero Azure porque es mas completo, permite realizar Testplan (casos de pruebas) y llevar todos los procesos de manera mas detallada.

- ¿Qué se realiza en cada uno de los eventos Scrum?
- **Planning :** Aquí se establece el trabajo que se va a realizar. Tiene una duración de máximo 4 horas para Sprints de 15 días y participa todo el Equipo Scrum.

- **Daily:** Es una reunión diaria que se da a la misma hora y en el mismo lugar. Tiene una duración de máximo 15 minutos y participa únicamente el Equipo Técnico.

Cada persona responde 3 preguntas:

- ¿Qué hice ayer?
- ¿Qué voy a hacer hoy?
- ¿Qué impedimento tengo?

- **Review:** Se realiza al finalizar el Sprint. Tiene una duración de máximo 1 hora para Sprints de 15 días y participa todo el Equipo Scrum.

Se revisa el trabajo realizado durante el sprint, se identifica el trabajo que se hizo bien para seguirlo realizando y se revisa que debe mejorar el equipo.

- **Retrospectiva:** Se realiza al final de cada Sprint, después de la Review y antes del siguiente sprint planning. Tiene una duración de máximo 2 horas para Sprints de 15 días y participa todo el Equipo Scrum.

Se revisa el trabajo realizado durante el sprint, se identifica el trabajo que se hizo bien para seguirlo realizando y se revisa que debe mejorar el equipo.

- Sobre Automatización y POO:

- ¿Mencione cuáles son los pilares de Programación Orientada a Objetos que se deben considerar al automatizar pruebas?

Abstracción. Captura solo las características esenciales del objeto, así como su comportamiento.

Encapsulamiento. Limita el acceso de los atributos internos de la clase. Solo se puede acceder a estos mediante la instancia de la clase.

Herencia. Permite que la clase hija tenga comportamientos de la clase padre.

Polimorfismo. Permite seleccionar los métodos que se utilizan durante tiempo de ejecución haciendo el código más flexible.

- ¿Qué papel juegan los principios SOLID en la automatización de pruebas?

Los principios SOLID son una serie (5) de principios (Principio de Responsabilidad Única, Principio de Abierto/Cerrado, Principio de Sustitución de Liskov, Principio de Segregación de Interfaces, Principio de Inversión de Dependencias) o buenas prácticas que se deben tener como base para crear código legible y mantenible. Para la automatización de pruebas estos principios juegan un papel muy importante, ya que permiten hacer el código más fácil de leer y mejoran su mantenibilidad

- ¿Cómo se definen los escenarios críticos a automatizar?

Primero se deben identificar las funcionalidades, se deben priorizar dichas funcionalidades, y se deben contemplar todos los escenarios posibles que se puedan presentar.

- Defina la pirámide de automatización de Cohn

La pirámide de automatización muestra los diferentes niveles como se deben organizar las pruebas,

dichos niveles están conformados por tres capas principales que son: las pruebas unitarias que es donde está la base amplia y sólida, las pruebas de servicio o integración, estas pruebas son las encargadas de verificar el software sin necesidad de pasar por la interfaz de usuario, y por último las Pruebas de Interfaz de Usuario o End-to-End que buscan simular el comportamiento del usuario final interactuando con el sistema en un entorno similar al de producción.

- Sobre servicios:

A. ¿Qué es un servicio?

Un servicio es una funcionalidad bien definida y estructurada que realiza una tarea específica o un conjunto de tareas relacionadas.

B. Diferencia entre SOAP y REST

SOAP	REST
-Usa protocolos basados en XML para intercambiar mensajes entre aplicaciones. - Define una interfaz formal utilizando documentos WSDL - Crea un contrato cliente/servidor	-Estilo arquitectónico que utiliza HTTP y URLs para acceder a recursos. -Utiliza métodos estándar de HTTP. - No genera por defecto un contrato entre cliente/servidor

C. Defina métodos GET, POST, PUT y DELETE

GET: Obtiene datos.

POST: Crea un nuevo recurso.

PUT: Actualiza datos.

DELETE: Elimina Recursos.

D. Herramientas de ejecución de servicios trabajadas

He probado servicios en herramientas como Postman y SoapUI.

E. ¿Cómo automatizar un servicio?

Para automatizar un servicio primero se debe identificar la funcionalidad, definir los casos de prueba, escoger una herramienta (como Postman o SoapUI), escribir el script para enviar la petición y recibir la respuesta.

F. Validaciones que ha utilizado al momento de automatizar servicios

Algunas de las validaciones que he utilizado al momento de automatizar servicios son las siguientes:

- El código del estado HTTP, es decir, que el servicio retorne 200,400, etc. según el código correspondiente a la prueba.
- He validado el contenido de la respuesta comprobando que el contenido que me devuelve el servicio cumple con el formato y los datos que se esperan
- He realizado validaciones sobre los campos verificando que contengan los valores correctos en el formato adecuado
- He realizado validaciones sobre los tiempos de respuesta que devuelven los servicios.
- He realizado validaciones sobre la respuesta de excepciones que correspondan al error que se espera.
- He verificado que se implementen las medidas de seguridad como la autenticación o autorización según corresponda el caso.

- **Pruebas No Funcionales:**

A. Explique las más utilizadas:

Seguridad. Estas pruebas están relacionadas con la detección de vulnerabilidades y la auditoria de buenas prácticas en el desarrollo.

Performance. También conocidas como pruebas de rendimiento, estas pruebas permiten simular la carga esperada del sistema en un entorno similar al de producción.

B. Herramientas utilizadas para ejecutar dichas pruebas

Seguridad: Hasta el momento no he trabajado ninguna.

Performance: Apache Jmeter.

C. Como realizar pruebas de carga con herramientas como JMETER, WAPT o BlazeMeter.

Para realizar pruebas con Jmeter se crea el plan de pruebas, se agrega el grupo de subprocesos (número de hilos, recuento de ciclo y periodo de despliegue), se agregan elementos de Jmeter (solicitud HTTP, nombre del servidor y IP) y por último se agrega el gráfico de los resultados.

- **Automatización de Móviles:**

Explique los siguientes conceptos:

App Nativas: Son aplicaciones desarrolladas para un sistema operativo específico.

App Híbridas: son aplicaciones que combinan tecnologías web con nativas.

Tipos de sistemas operativos: Android, IOS, Windows Phone.

- ¿En qué difiere la automatización de pruebas web con Selenium y móviles con Appium?

Con Appium se puede automatizar las pruebas para aplicaciones móviles y de escritorio. Con Selenium solo se puede automatizar las pruebas de aplicaciones web en navegadores.

- ¿Qué ha escuchado o trabajado sobre Granjas de dispositivos móviles? Nombre las granjas que ha escuchado y/o trabajado.

No he escuchado ni trabajado con granjas de dispositivos móviles.

- **Seguridad**

A. Conceptos de seguridad en pruebas web y móvil:

La seguridad en aplicaciones web y móviles se refiere al proceso de desarrollar y probar características de seguridad dentro de las aplicaciones para evitar vulnerabilidades.

B. Herramientas utilizadas para pruebas de seguridad

No he escuchado de herramientas utilizadas para pruebas de seguridad.

- **Tecnología Cloud:**

A. ¿Ha trabajado con proyectos Cloud? ¿Herramientas utilizadas?

No he trabajado en proyectos Cloud.

- **Integración Continua:**

A. Defina las tareas de Integración y Despliegue continuo en la cual interviene un automatizador de pruebas

- Ejecución de pruebas automatizadas.
- Gestión de hallazgos
- Monitorear procesos de pruebas
- Resultados obtenidos

B. GIT: ¿Qué es, mencionar los comandos más utilizados, Ha trabajado con GitLab o GitHub?

Git. Git es un software de control de versiones distribuido, esta herramienta permite mantener el historial completo de las versiones que se generan a lo largo del desarrollo.

Comandos más utilizados de Git:

- Git init
- Git clone
- Git add
- Git commit
- Git pull
- Git push
- Git Merge
- Git Status
- Git brach
- Git checkout
- Git reset
- Git log

No he trabajado con GitLab o GitHub.

C. ¿Explique cómo es el proceso de integración continua con Jenkins?

He escuchado de Jenkins, pero no he trabajado con este

D. ¿Explique cómo es el proceso de integración continua con Azure DevOps?

He escuchado de Microsoft Azure, pero no he trabajado con este.