Pruebas unitarias

consisten en una tecnica esencial para probar cada parte del codigo, como funciones o metodos que cada unidad funcione por separado

La importancia: Detecta errores en etapas tempranas y aumenta estabilidad del producto. Mejora la calidad el software desde el principio

Beneficios: Tiene 4

Mejora la calidad del cod

aumentna la confianza

ayudan a detectar errores a tiempo

favorecen un codigo reutilizable

PRACTICAS.

Mantner las lapias simple, documentarlas bien para que todo el equipo pueda usarlas.

un correcto enfoque sistematico a la hora del desarrollo

que las pruebas unitarias sean efectivas y que aporten el maximo valor al proceso de desarrollo Ayuda a mantener la calidad el código a lo largo del tiempo y facilita la detección temprana de problemas

SYSTEM TESTING

pRUEBAS DE SISTEMA

Realizar las pruebas y validar todo lo que se ingresa. Validar datos en excel y que las base de datos estén bien ordenadas que los datos entren y retornen al codigo

Despues de las pruebas unitarias o las de integración es importante el system testing

Pruebas funcionales y o funcionales; Cumpla con lo que se solicita.

Funcionales: El software cumpla con los rendimientos adecuados y los componentes trabajen adecuadamente, se ajusta la escalabilidad entre otras cosas.

DiFERENTES ENFONQUES

CAJA NEGRA: LA funcionalidad externa sin conocer la infraestructura de nuestro software.

END TO END: Evalua la situación de acuerdo al trafico de usuarios por ejemplo

Planificación de casos de prueba: MEtodos para identificar, documentales

HERRAMIENTAS

JIRA, textrail

Selenium: Automatización

Caos de prueba. Mediante un formulario un sistema de gestión de tareas resaltando sus caracterisitcas mas importantes.

ingresar datos validos en el formulario

pulsar el boton guardar

verificar la persistencia de la base de datos

**pruebas de carga**

**prueba de rendimiento que buscan entender como se comporta un sistema**

**evaluar lal capdidad de respuesta, la estabilidad, que tan facil se mantiene el sistema trabajando adecuadamente: Un flujo mas alto de usuarios. QUe limite de carga soporta:**

**Objetivos: Determinar la capacidad maxima antes de la degradacion del servicio**

**Garantizar estabilidad en escenario de alto flujo de trafico**

**diseño de una prueba de carga:**

**4 fases\_ DEFINIR LOS OBJETIVOS. QUE SISTEMA QUEREMOS EVALUAR: correos masivos**

**2. Seleccionar las metricas claves: TIEMPOS DE RESPUESTA, Enviar o recibir datos, CPU MEMORIA EL USO FISICO**

**3. Determinar un escenario de rol o comunicación, una carga progresiva, información constante y que el servidor soporte esa cantidad. Flujo muy grande de datos que ese servidor soporte esa cantidad que no es común de la esperada.**

**4.Uso prolongado: Que el servidor este constantemtne estable en funcionamiento**

**4. Mejora continua. Retroalimentación**

**HERRAMIENTAS: APache Meter, tiene interfaz visual, cuantas solicitudes, Gatling; basado en framework java, Locust(python)**

**EJEMPLOS: Amazon, netflix, MiBancolombia. TENER EN CUENTA UN AMBIENTE CONTROLADO y verificando que la pagina no se caiga en tantos segundos. ANtes del despliegue de información se hacen las pruebas de carga.**

**Plataformas de streaming es un evento masivo que recoge muchas personas como el SUPERBOLW**

**EL BENEFICIO: Detectamos fallos antes del lanzamiento y ahorramos costos y evitamos perdidas de usuarios**

**Antes de salir a producción es la prueba de carga**

**Se usan herramientas para simular el flujo de carga como LOCUST, METER, GATLING**

**es facil de usar la app**

**usabilidad son una tecnica para evaluar que tan facil y agradable resulta para el usuario interactuar con un producto digital, identificando ls problemas, permitiendo ser intuitivo, eficiente y accesible.**

**objetivos**

**detectar dificultades**

**validar si el diseño es intuitivo**

**mejorar la experiencia del usuario**

**tipos de pruebas**

**Moderadas y no moderadas, presenciales, remotas, cualitativas, cuantitativas= tiempos de respuesta**

**Se hacen con un grupo pequeño**

**pasos**

**definir objetivos de la prueba**

**preparar herramientas y entorno**

**Ejectar la prueba**

**ANalizar los resultados**

**Implementar mejoras al diseño**

**Metricas,**

**Tiempo para completar tareas**

**Numero de errores cometidos**

**Tasa de abandono (**

**Grabaciòn de las sesiones: Mapas de calor online, datos biometricos, encuestas para sitios web**

**Beneficios**

**Mejora la experiencia del usuario**

**Ahorra tiempo y costos a largo plazo**

**Aumetna la eficiencia de uso**

**ofrece retroalimentación valiosa del usuario real**

**mejora la toma de decisiones de diseño**

**aumenta la satisfacción y fidelizaciòn**

**EJEMPLO: Una app de pedidos de comida: EL usuario busca un restaurante, selecciona un menu y realiza el pedido.**

**Cnclusiòn:**

**Son esenciaels para mejorar la experiencia del usuario**

**No se necesita un gran muestra:**

**Se deben realizar desde etapas de desarrollo hasta la ejecuciòn.**