

CERTIFICACIÓN DE APLICACIÓN WEB “LIQUOR LOGIX”

Definición del proceso de pruebas

Contenido

- Propósito
- Contexto actual
- Objetivo
- Estrategia de pruebas
- Alcance
- Criterios de suspensión y reanudación de pruebas
- Requerimientos de prueba
- Diseño de casos de prueba
 - Pruebas funcionales
 - Pruebas en la etapa de sistema
 - Técnica de pruebas caja negra
 - Partición equivalente
 - Casos de uso
- Resumen de ejecución
- Defectos detectados
- Evaluación de riesgos
- Evaluación y cierre
- Liberación a producción
 - Conclusiones

Propósito

Esta sección del documento engloba los siguientes objetivos:

1. Definir el alcance y enfoque de las pruebas a realizar
2. Identificar los componentes de software que deben ser probados y las características o escenarios específicos a ser probados
3. Delimitar las tareas de pruebas y el esfuerzo requerido para desarrollar cada una de ellas
4. Identificar los recursos necesarios (humanos, software, hardware, etcétera) para cubrir y desarrollar las tareas de pruebas
5. Describir los riesgos que pueden afectar el plan de pruebas
6. Establecer los artefactos de pruebas que se producirán durante el proceso de pruebas

Contexto actual

En el local host de una máquina cualquiera se tiene acceso al modulo categoria vamos a “creacion categoria” la cual permite ingresar datos, solicitando el diligenciamiento de los siguientes campos:

- Nombre (alfanumerico de la A a la Z, numeros del 1 hasta 9 sin caracteres especiales)
- Ubicación (alfabético de hasta 15 caracteres de la A a la Z, numeros del 1 hasta 9 sin caracteres especiales)

Objetivo

Realizar el proceso de pruebas al módulo web “creacion categoria” con la finalidad de contar con los elementos mínimos necesarios para poder determinar si el módulo cumple con la definición de calidad.

Calidad de software:

“La totalidad de la funcionalidad y prestaciones de un producto de software que están relacionadas con su capacidad de satisfacer las necesidades explícitas o implícitas”.

Estrategia de pruebas

Según la necesidad comercial y del usuario es básico para el correcto funcionamiento del módulo validar:

- Manuales
- Dinamicas
- De caja negra
- Funcionales.

A continuación se muestra la estrategia de pruebas definida dentro del alcance.

Alcance

En este apartado se mostrará cómo llevar a cabo el proceso de pruebas básico para el módulo web “Creacion Categoria”.

- **Pruebas Modulares.** Pruebas que validan la correcta funcionalidad del Módulo o Subsistema

Las funcionalidades a probar son las siguientes:

- Agregar categoria

- Actuliza
- Eliminar categoria

Las pruebas se enfocarán a validar la funcionalidad para el aseguramiento de la calidad del software según.

Las pruebas se realizarán a nivel de sistema, además de que la funcionalidad a validar deberá estar completa.

Las técnicas existentes para la validación de un sistema se componen de técnicas estáticas y dinámicas, siendo estas últimas las más utilizadas debido su fácil aplicación y sus óptimos resultados; las técnicas dinámicas están divididas en técnicas de caja negra y caja blanca, éstas últimas se enfocan a validar la complejidad y cobertura del código; para este caso de estudio sólo aplicaremos las técnicas dinámicas de caja negra:

- Partición equivalente
- Valores límite
- Casos de uso

El tipo de pruebas no funcionales, pruebas estructurales y pruebas asociadas al cambio, serán objeto de estudio en un planteamiento futuro quedando fuera de alcance para este proyecto.

Criterios de suspensión y reanudación de pruebas

La ejecución de pruebas será suspendida en los siguientes casos:

- No se cuenta con un ambiente de pruebas
- La versión entregada no supera la validación general, pues presenta muchas fallas
- El incremento de defectos es muy alto
- No se cuenta con datos de prueba

Se podrán reanudar las pruebas una vez corregidos o solucionados los puntos anteriores.

Requerimientos de prueba

ID	Requerimiento de prueba	Tipo	Complejidad
1	Nombre Categoría	Funcionalidad	Media
2	Ubicación “ciudad”	Funcionalidad	Media

Diseño de casos de prueba

Pruebas funcionales

Las pruebas funcionales se dividen en estáticas y dinámicas, la revisión está enfocada a las pruebas funcionales-dinámicas de caja negra, entendiendo que nuestro objeto de estudio serán únicamente las entradas y salidas en el sistema sin importar cómo opera internamente.

Pruebas en la etapa de sistema

De acuerdo a la estrategia de pruebas, los casos son ejecutados a nivel sistema, es decir una versión completa, una versión lo más semejante al ambiente productivo.

Se aplicarán pruebas dinámicas de caja negra, las cuales comprenderán de las siguientes técnicas:

Técnica de caja negra

- Partición equivalente
- Valores límite
- Casos de uso

Técnica de pruebas caja negra

a) Partición equivalente

La partición en clases de equivalencia se realiza dividiendo los posibles valores de entrada y los posibles valores de salida. El rango de valores definido se agrupa en clases de equivalencia.

Para este caso tenemos 2 campos con las siguientes características

- Nombre (alfanumerico de la A a la Z, numeros del 1 hasta 9 sin caracteres especiales)
- Ubicación (alfanumerico de 1 hasta 15 caracteresde la A a la Z, numeros del 1 hasta 9 sin caracteres especiales)

La partición equivalente será aplicada de la siguiente manera:

Nombre (alfabético de 4 hasta 15 caracteres)

Toda cadena alfanumerico de la A a la Z, numeros del 1 hasta 9 sin caracteres especiales res es válida, ejemplo:

- Clase de equivalencia válida: Gaseosas, Cerveza, Dulces...
- Clase de equivalencia inválida: Gaseos·", Cer@zaa, Nadia_360
- Clase de equivalencia inválida valores numéricos: 1256487865, 13157654.488, -469841521

Se considera que para cada clase de equivalencia definida sólo es necesario un representante, es decir no es práctico probar con todos los valores ya que el resultado siempre será el mismo.

Ubicación "Ciudad" (alfabético de 4 hasta 15 caracteres)

Toda cadena alfanumerico de 4 hasta 15 caracteresde la A a la Z, numeros del 1 hasta 9 sin caracteres especiales es válida, ejemplo:

- Clase de equivalencia válida: Bogota, Medellin, Cali...
- Clase de equivalencia inválida caracteres especiales: Cal_&, Bogot@, castañeda#
- Clase de equivalencia inválida valores numéricos: 12564,878, 131548_8, -469841521

Se considera que para cada clase de equivalencia definida sólo es necesario un representante, es decir no es práctico probar con todos los valores ya que el resultado siempre será el mismo.

c) Casos de Uso

Identificador	Nombre
CU007	Crear Categoría
Requerimiento de prueba	
REQ003	
Precondiciones	
<input type="checkbox"/> NA	
Objetivo	
<input type="checkbox"/> Permitir al usuario crear una Categoría para su inventario	
Flujo Principal	
<ol style="list-style-type: none">1. El caso de uso inicia cuando el usuario ejecuta la opción categoría en su sistema.2. El módulo presenta formulario para crear nueva categoría.3. El usuario ingresa los campos: nombre Ubicación y da clic en “Guardar”.4. Si la información es correcta el módulo manda el mensaje alerta.5. La categoría se muestra en la lista de categorías.6. Fin de caso de uso.	
Flujo Alternativo CU001-1	
<ol style="list-style-type: none">1. El flujo alternativo inicia cuando el usuario ejecuta la opción categoría en su sistema.2. El módulo presenta formulario para crear nueva categoría.3. El usuario ingresa el campo nombre Ubicación y da clic en guardar (al ingresar el nombre o ubicación deja campos en blanco o en un formato incorrecto).4. Si la información es incorrecta el módulo manda el mensaje alerta.5. Fin del flujo alternativo.	
Anexos: https://youtu.be/G-bABmyvsL0?si=vxKqfYtkQLsevn2W	

Resumen de ejecución

ID	Requerimiento	ID	Tipo	Caso de prueba	Resultado
3	Agregar categoria	TTF		Clase Inválida números negativos. Nombre	Pass
		TTF		Campos Requeridos. Nombre	Pass
		TTF		Clase inválida caracteres especiales y numericos. Nombre	Pass
		TTF		Clase inválida Caracteres no numéricos. Nombre	Pass
		TTF		ClaseInválida números negativos Ubicacion	Pass
		TTF		CamposRequeridos. Ubicacion	Pass
		TTF		Clase inválida caracteres especiales y numericos. Ubicacion	Pass
		TTF		Clase inválida Caracteres no numéricos. Ubicacion	Pass
		TTF		ClaseInválida.X<9Dígitos. Nombre	Pass
		TTF		ClaseInválida.X>9Dígitos Ubicacion	Pass
		TTF		ClaseInválida.X>15Caracteres Nombre	Pass
		TTF		ClaseInválida.X>15Caracteres Ubicacion	Pass
		TTF		Categoria duplicada	Pass



Evaluación y cierre

Evaluación. El módulo sometido a pruebas presentó un comportamiento continuo sin riesgos de operación elevados, el equipo de aseguramiento de calidad recomienda instalar el aplicativo en producción con un período de estabilización y mejora continua.

Cierre. Se ejecutaron 2 casos de prueba en un ciclo completo. En el cual se detectaron 0 defectos no funcionales..

Liberación a producción

El aplicativo será liberado a producción con la valoración del área Aseguramiento de la Calidad, adicional a esto, es requerida la autorización por escrito del:

- Gerente de sistemas
- Líder de Pruebas
- Gerente del área de negocio

Comentarios finales

La operación del sistema es estable, los riesgos probables son menores ya que no interfieren ni comprometen la funcionalidad principal del sistema, adicional a esto no hay comprometidos bienes materiales ni daño físico.

El sistema entrará en un proceso de mejora continua.

Conclusiones

Las pruebas de software son todo un proceso, dentro del proceso de desarrollo, que debe llevarse a cabo para asegurar la calidad del software. Algo que debemos reconocer es la naturaleza compleja de las mismas. Las pruebas pueden iniciarse en cualquier momento dependiendo del tipo de pruebas que se requiera.

El proceso de pruebas es de vital importancia dentro del ciclo de vida del software, de esta manera se puede verificar la calidad del producto antes de su puesta en producción.

El proceso de pruebas ayudó a la recolección de los requerimientos, así como a evitar inconsistencias, omisiones y confusiones. El análisis permitió que las correcciones se realizaran de manera más fácil y en etapas tempranas.

Como todo aspecto de la vida, la calidad es sinónimo de satisfacción y seguridad. Las pruebas de software ganan cada día más terreno ya que lo que siempre buscamos es mantener a nuestro cliente contento. En este aspecto existe un tema de complicidad con el usuario ya que es el único que puede dar el visto bueno del trabajo realizado por todo un equipo, desde los analistas hasta el ingeniero de pruebas.

El ingeniero de pruebas debe aprender a desarrollar una buena comunicación con todo el equipo para que éste acepte las críticas u observaciones de manera positiva, ésta no es una tarea fácil incluso es un punto relevante para el éxito del proyecto.

Tener una estrategia de pruebas en la planeación, desarrollo y entrega del producto puede reducir costos y/o evitar gastos en corrección de incidentes que son detectados en etapas tempranas y de esa forma asegurar la calidad y reducir la probabilidad de riesgos que se puedan presentar en el desarrollo del software y su puesta en producción.

Al final, las pruebas terminan siendo una inversión y pueden ahorrar tiempo y dinero.