

Caso de estudio

CERTIFICACIÓN DE APLICACIÓN WEB LIQUORLOGIS MODULO “PRODUCTOS”

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
 - Valores límite
 - Casos de uso
- Resumen de ejecución
- Defectos detectados
- Evaluación de riesgos
- Evaluación y cierre
- Liberación a producción
- Conclusiones

1. 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

2. Contexto actual

En el modulo producto nos permite almacenar e ingresar datos de los ingresos de productos , solicitando el ingreso de los siguientes campos:

- **Codigo:** letras de la A la Z minúsculas o mayúsculas, números de 1 hasta 70 y sin caracteres especiales.
- **Nombre:** letras de la A la Z minúsculas o mayúsculas, números de 1 hasta 70 y sin caracteres especiales.
- **Precio:** solo ingresa números positivos enteros de 1 hasta 25 caracteres.
- **Stock:** solo ingresa números positivos enteros de 1 hasta 25 caracteres.
- **Categoría:** obligatorio preestablecido.
- **Foto o imagen del producto:** no obligatorio 1 imagen en formato jpg, jpeg o png

3. Objetivo

Realizar el proceso de pruebas al módulo “producto” 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”.

4. Estrategia de pruebas

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

- Adecuación
- Exactitud
- Interoperabilidad
- Seguridad
- Cumplimiento de funcionalidad

Se contempla que la aplicación local soporta hasta 1000 productos, por tal motivo no es necesario ni será caso de estudio para este ejemplo realizar pruebas de volumen o estrés, nos enfocaremos solamente a validar los atributos funcionales de la calidad y algunos atributos no funcionales.

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

5. Alcance

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

Las pruebas comenzarán con la etapa de las pruebas Modulares, en las cuales mediante pruebas dinámicas, casos de prueba diseñados y set de datos válidos e inválidos, se verificara la funcionalidad del módulo a probar, durante esta etapa se identificarán los casos de prueba adecuados para ejecutar la siguiente etapa que son las pruebas integrales las cuales tienen como objetivo validar la integración de la aplicación de un módulo.

- **Pruebas Modulares.** Pruebas que validan la correcta funcionalidad del Módulo o Subsistema
- **Pruebas de Integración.** Pruebas que aseguran el correcto funcionamiento del flujo de trabajo

Las funcionalidades a probar son las siguientes:

- Agregar Productos
- Consultar Productos
- Eliminar Producto

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

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

6. 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.

7. Requerimientos de prueba

ID	Requerimiento de prueba	Tipo	Complejidad
1	Agregar productos	Funcionalidad	Alto
2	Consultar productos	Funcionalidad	Medio
3	Eliminar productos	Funcionalidad	Medio

7. Diseño de casos de prueba

7.1 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.

7.2 Pruebas en la etapa de sistema

De acuerdo a la estrategia de pruebas, los casos son ejecutados a nivel sistema, 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 5 campos con las siguientes características

- Nombre (alfabético de hasta 15 caracteres)
- Apellido (alfabético de hasta 15 caracteres)
- Correo electrónico (Alfanumérico de hasta 15 caracteres con validación de @ y dominio)
- Número de celular a 10 dígitos (numérico de 10 dígitos)
- Edad (numérico de 2 dígitos entero positivo)

La partición equivalente será aplicada de la siguiente manera:

Código (letras de la A la Z minúsculas o mayúsculas, números de 1 hasta 70 y sin caracteres especiales)

Toda cadena alfanumérica con un máximo de hasta 70 caracteres es válida, ejemplo:

- Clase de equivalencia válida: A01,A02,B07,C65
- Clase de equivalencia inválida: Klauss12, K@rla, Rumpelstiltskinn, Nadia_360

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.

Nombre (letras de la A la Z minúsculas o mayúsculas, números de 1 hasta 70 y sin caracteres especiales).

Toda cadena alfanumérica con un máximo de hasta 70 caracteres es válida, ejemplo:

- Clase de equivalencia válida: Villegas, Zambrano, Rio, Castillo
- Clase de equivalencia inválida caracteres especiales: corona_&, c@rranza, castañeda#
- Clase de equivalencia inválida valores numéricos: 12564878, 1315488, -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.

Precio (solo ingresa números positivos enteros de 1 hasta 25 caracteres).

Toda cadena numérica con un máximo de hasta 25 caracteres es válida Clase de equivalencia válida:

- Clase de equivalencia inválida: 2500, 7800, 9500, 135000...

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.

Stock: solo ingresa numeros positivos enteros de 1 hasta 25 caracteres.

Toda cadena numérica de 10 dígitos enteros positivos

Ejemplo:

- Clase de equivalencia válida: 5555555555, 2345678952, 5544228877, 01234567890
- Clase de equivalencia inválida cadena mayor a 25 dígitos: 55415637890865442288771245, 01234567890124509876356793668
- Clase de equivalencia inválida CADENA MENOR A 15 DIGITOS: 552266, 457896, 254546446
- Clase de equivalencia inválida valores no numéricos : roberto23, “#\$%%12, 56****Clau
- Clase inválida números negativos: -5555555555, -1234567890
- Clase inválida números decimales: 5.098783125, 1234.567891

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.

Categoría (obligatorio preestablecido)

Selección predeterminada del sistema

- CERVEZAS
- DULCES
- GALLETAS

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.

Caso de Uso

Identificador	Nombre
CU003	INGRESAR PRODUCTO
Requerimiento de prueba	
REQ005	
Precondiciones	
<input type="checkbox"/> Debe ser un producto valido	
Objetivo	
<input type="checkbox"/> Permitir al usuario modificar producto	
Flujo Principal	
Flujo Alternativo CU003-1	
<ol style="list-style-type: none"> 1. El flujo alternativo inicia cuando el usuario ejecuta la aplicación agenda en su pc. 2. El módulo presenta formulario para crear nuevo producto, campo para realizar búsqueda y opción para consulta de todos los productos. 3. El usuario da clic en el botón “productos”. 4. El módulo muestra la información de los ingreso de productos “Editar” y “Eliminar”. 5. El usuario da clic en el botón “Editar” del producto que desee modificar. 6. El módulo presenta la información del producto habilitada para su edición. 7. El usuario realiza cambios en alguno(s) de los campos: nombre, stock, precio, categoria, el usuario da clic en “cancelar” 8. El módulo no guarda los cambios. 9. El módulo muestra pantalla todos los contactos. 10. Fin del flujo alternativo. 	
Postcondiciones	
<ul style="list-style-type: none"> - El producto se actualiza 	
Anexos : https://youtu.be/c51HIs1JQD8	

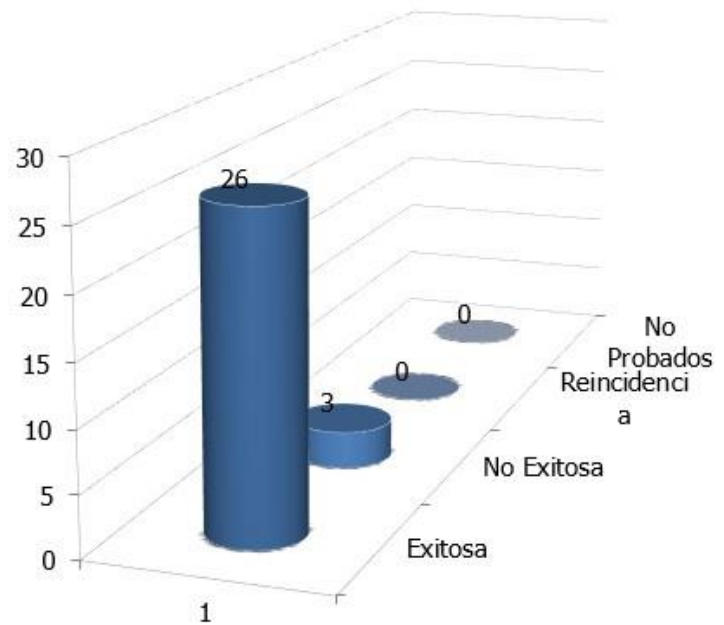
Resumen de ejecución

Estatus de prueba

Ejecución

Estatus	Ciclo 1
Exitosa	26
No Exitosa	3
Reincidencia	0
No Probados	0

General	
Requerimientos de prueba	6
Casos de Prueba	29
Happy Path	1
Automatizados	0



9 Defectos detectados

ID	Defecto	Riesgo	Consecuencia	Impacto	Estatus
45	Homogenizar etiquetas	Ninguno	NA	NA	Cerrado
46	Cabeceras	Ninguno	NA	NA	Cerrado
47	Etiqueta buscar contacto	Ninguno	NA	NA	Cerrado

10 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 casos de prueba en un ciclo completo. En el cual no se detectaron defectos no funcionales. Tras la corrección de los defectos se ejecutó el re-test en el cual los defectos reportados fueron corregidos correctamente. Completando la valoración del módulo se aplicaron pruebas de regresión del flujo principal, en el cual no se detectaron defectos.

11 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

12 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.