

Actividad | #1 | Aseguramiento de la calidad

Ingeniería en Desarrollo de Software



TUTOR: Jessica Hernández Romero

ALUMNO: Edgar Enrique Cuamea Ochoa

FECHA: 26 de agosto del 2025

Contenido

Introducción.	3
Descripción.	4
Justificación.	5
Desarrollo.....	6
Plan de pruebas de caja negra.	8
Lista de verificaciones	11
Conclusión.	15

Introducción.

En esta actividad veremos el desarrollo del aseguramiento de la calidad de un software, por lo que se realizarán diferentes pruebas y métodos para asegurar que el software se entregue sin defectos y asegurando su funcionamiento dependiendo de los requisitos que se verán en el desarrollo de la actividad, este software tiene diferentes requisitos técnicos así como algunos requisitos funcionales tales como el desarrollo en algún lenguaje de programación así como la necesidad de realizar diferentes copias de seguridad cada cierto tiempo, para ello realizaremos una lista para asegurar que los atributos de la calidad estén descritos correctamente, además diseñaremos un plan de pruebas de caja negra, este plan nos servirá para evaluar el funcionamiento del software sin conocer su estructura interna por lo que solo se supervisará la entrada y salida de datos esperando no encontrar ningún error, así como la lista de verificación de los requerimientos que se necesitan para el software.

Descripcion.

En el desarrollo de la actividad estableceremos diferentes atributos de calidad, estos son algunos atributos medibles del sistema que nos permitirán saber cómo el sistema desarrollado satisface algunas necesidades tales como usabilidad, flexibilidad, portabilidad, etc. Realizaremos nuestro plan de pruebas de caja negra por lo que se realizarán algunas pruebas del sistema, ya sea algunos registros de clientes o empleados, modificaciones o incluso la eliminación de estos para poder verificar que el módulo desarrollado está realizado correctamente y no presenta errores durante el proceso a analizar, realizaremos una lista de verificación de los requerimientos, estos tendrán como objetivo analizar el software para que cumpla con todos los requisitos que el cliente expresó antes de desarrollar el software por lo que verificaremos que los requisitos estén realizados correctamente antes de entregar el software y asegurando la calidad del software reduciendo las posibilidades de entregar software con defectos y errores al aplicar estos métodos.

Justificación.

Realizaremos los atributos de calidad para asegurar que el software cumpla con los requisitos de usabilidad, portabilidad, adaptabilidad, etc. Además de cumplir con los requisitos de funcionalidad, rendimiento o seguridad que el cliente requiere, de esta forma comprobaremos que cumple con todos los requisitos previos antes de realizar la prueba de la caja negra, esta tendrá como objetivo realizar un análisis de la funcionalidad de cada uno de los módulos construidos, con este método de la caja negra solo realizaremos comprobaciones de entrada y salida ya que solo verificaremos que el software cumpla con los requisitos viendo desde la perspectiva de usuario final, probando diferentes métodos para tratar de encontrar errores y poder solucionarlos antes de entregar el software, además realizaremos la lista de verificación de los requerimientos ya que esta es una herramienta que nos asegura que cumpliremos con todos los requisitos de definidos por el cliente para el proyecto de software, además de poder identificar problemas de requisitos en etapas tempranas al desarrollo, de esta forma estaremos cumpliendo con las revisiones del software para poder entregar un producto de calidad.

Desarrollo.

Realizaremos un análisis de atributos, así como el plan de pruebas de caja negra, para ello contamos con un programa para una compañía que necesita llevar un control de su inventario y clientes en un sistema multiplataforma en línea y debe de poder accederse desde la web en cualquier dispositivo, tenemos los siguientes requisitos

- El sistema será desarrollado con los lenguajes HTML5, CSS, Python y un patrón de arquitectura MVC.
- El modelo encargado de conectarse a la base de datos será postgresSQL.

Se pide que el programa presente las siguientes características:

- Se le solicitará al personal ingresar su número de empleado y contraseña, los cuáles serán consultados en la base de datos. De esta manera, si el usuario no existe o la contraseña no corresponde a este, se le informará mediante un mensaje de error.
- Toda la información presentada por la aplicación deberá mostrarse conforme a los formatos corporativos definidos por la organización.
- El sistema deberá permitir tener dos perfiles: Administrador y Empleado.
- El sistema deberá permitir al administrador dar de alta a los diferentes empleados; en caso de que ya exista un empleado y el administrador quiera volver a agregarlo, el sistema deberá mostrar un error.
- El sistema deberá permitir a los empleados dar de alta los productos del inventario; en caso de que el producto ya exista, deberá mostrar un mensaje de error.
- El sistema proporcionará la opción de generación de reportes de inventarios, clientes y empleados.

Requerimientos funcionales:

- El sistema debe ser veloz.
- El sistema debe ser fácil de entender.
- El sistema debe tener información encriptada.
- La información debe ser respaldada cada 7 días.

Atributos de calidad

Una vez que tenemos los requisitos del sistema, describiremos los atributos de calidad, en este caso tenemos los siguientes:

Usabilidad: el sistema debe ser fácil de usar para los usuarios finales, debe tener un buen diseño y navegación intuitiva

Fiabilidad: el sistema debe funcionar de forma estable durante su uso

Eficiencia: el sistema debe responder rápidamente ante cualquier acción, incluyendo el uso eficiente de los recursos de hardware y software

Portabilidad: el sistema debe de ser portable, poder ser transferido a diferentes entornos de trabajo, así como a un nuevo sistema operativo

Mantenibilidad: el sistema debe poder recibir actualizaciones, poder ser modificado, corregido o adaptado a un entorno nuevo

Estos son algunos requisitos para asegurar la calidad de todo el software creado, ya que tenemos nuestros atributos de calidad, realizaremos nuestro plan de pruebas de caja negra, en este caso tendremos que probar los módulos desarrollados para el software de inventario, para ello

revisaremos los requisitos para saber a qué modulo le realizaremos un test.

Plan de pruebas de caja negra.

Analizamos los diferentes requisitos, tenemos que el sistema debe conectarse a la base de datos utilizada en PostgreSQL, para este caso se tiene que realizar diferentes pruebas de conexión, con la base de datos, ya sea desde consola o diferentes dispositivos, en este caso se deberán realizar diferentes pruebas de conexión con la base de datos ya que es indispensable que esto se realice correctamente antes de poder ingresar clientes, productos, empleados, antes de realizar las pruebas se debe asignar a un encargado para que realice las pruebas en busca de errores así poder encontrar fallos en este.

El sistema debe poder acceder se desde un sistema web, en este caso se debe realizar la conexión con la base de datos desde una computadora, una Tablet y un teléfono para verificar que la conexión se establece desde los diferentes dispositivos, en cada una de las pruebas se debe tener un registro de en qué dispositivo se realizó la conexión así como el método a utilizar además de documentar las fallas en caso de no realizarse la conexión, al ser una página web donde se pueda ingresar, necesitamos verificar la conexión a internet para poder ingresar desde cualquier lado o utilizando una red privada que contará con el acceso al sistema de inventarios,

Cada prueba debe de realizarse anotando un numero de folio para poder tener un seguimiento del proceso a probar y documentar el error en caso de ser encontrado. Una vez que verificamos que la base de datos pueda conectarse correctamente, pasaremos a la pantalla de inicio de sesión ya que se debe ingresar un numero de empleado así como ola contraseña para poder

ingresar al sistema, se espera que en caso de ser erróneos alguno de los datos mostrara una mensaje de error ya que es necesario para saber que ingresamos un dato mal , se ingresaran un total de 5 usuarios , ya sean reales o falsos con contraseñas reales y falsas para verificar su funcionamiento desde los 3 dispositivos, por lo que la base de datos debe verificar el contenido de las dos casillas de usuario y contraseña para poder ingresar al sistema, en caso de mostrar un errores con el ingreso de los usuarios y contraseñas reales, anotar el numero de folio y documentar lo sucedido para después replicarla en los dispositivos restantes para verificar que el problema es de la conexión con la base de datos o algún error durante el desarrollo del módulo.

El sistema deberá tener 2 perfiles, uno de empleado y uno de administrador por lo que es necesario replicar los pasos anteriores para verificar el funcionamiento de inicio de sesión en los perfiles de administrador y de empleado, de esta forma descartaremos errores presentados con las conexiones de las bases de datos, así como comprobar el funcionamiento del módulo de inicio de sesión, en caso de mostrar ingresar mal el usuario o contraseña, se espera que se muestre un mensaje de error.

El sistema debe de permitir que solo el administrador pueda dar de alta a empleados, por lo que se procede a ingresar diferentes empleados para verificar que el sistema los da de alta correctamente, ingresaremos el mismo empleado 2 veces para verificar que el sistema reconoce que es un duplicado y pueda mostrar el mensaje de error, probaremos los mismos métodos en teléfono, computadora y Tablet para verificar que el ingreso de datos se realiza correctamente

El sistema debe permitir que el empleado pueda dar de alta los productos de inventario, en

este caso se deberá ingresar diferentes productos para darlos de alta, se ingresaran productos con el mismo nombre esperando a que muestre el error de producto duplicado, se debe poder ingresar los productos desde los diferentes dispositivos a testear.

El sistema debe poder generar reportes de inventarios, clientes y empleados, por lo que una vez que se hayan ingresado diferentes empleados al sistema así como los diferentes productos de inventarios y clientes, debemos de generar un reporte de los empleados, productos y clientes que se realizaron durante un periodo de tiempo, en este caso debemos de imprimir el reporte para verificar que durante el periodo de pruebas se hayan ingresado correctamente todos los datos de los empleados, productos y clientes que se registraron en la base de datos, en caso de notar que hay errores en el reporte como algunos movimientos de ingreso de empleados, clientes o de productos faltantes, se tendrá un numero de folio con el movimiento que se realizo en la prueba para poderlo replicar, de esta forma si el error es replicable, probablemente sea un error del modulo que se encarga de realizar esa acción en concreto, por lo que de esta forma se enumeran las operaciones de prueba, detección de errores durante las pruebas así como al final ya que es necesario tener un control de lo que se realiza en este plan de pruebas de caja negra, ya que no conocemos como está construido los módulos del software y solo comprobamos que funcione correctamente al realizar las pruebas.

El sistema debe de poder respaldar la información cada 7 días por lo que se realiza un respaldo de las operaciones que se realizaron durante las pruebas además de esperar a que se realice el respaldo automáticamente para después poder verificar si se realizo el respaldo de manera automática para verificar el funcionamiento del modulo

Lista de verificaciones

Tenemos diferentes requisitos que se necesitan para poder entregar el software por lo que realizamos una lista de verificaciones para comprobar que todos los requisitos sean verificados e integrados correctamente por lo que realizamos el tipo de requerimiento, descripción y observación sobre estos requisitos

Requerimientos técnicos	Descripción	Observación
Arquitectura	El sistema debe de ser desarrollado con un patrón MVC	El sistema tiene su interfaz de usuario sin interactuar con el modelo y el controlador
Lenguajes de programación	El sistema debe de ser desarrollado con HTML5, CSS y Python	Se utiliza estos lenguajes de programación para poder presentar el sistema en una página web
Base de datos	El modelo de base de datos debe de ser PostgreSQL	El sistema se conecta a la base de datos para poder realizar los respaldos y poder verificar las credenciales de inicio de sesión
Acceso	El sistema debe de ser multiplataforma en línea con acceso web	Al utilizar lenguaje de programación html5, puede mostrarse en una pagina web accesible desde cualquier

		dispositivo con acceso a internet
Requerimientos funcionales		
Autenticación	El sistema debe de solicitar numero de empleado y contraseña	El sistema solicita el ingreso de un numero de empleado y contraseña para poder iniciar sesión
Validación de credenciales	El sistema debe de validar las credenciales en la base de datos	El sistema se comunica con la base de datos, en caso de no conectarse a la base de datos o verificar que el usuario o contraseña son incorrectos, entonces mostrara un error
Mensajes de error de autenticación	El sistema debe de mostrar un mensaje de error si el usuario o contraseña son incorrectos	El sistema muestra un error al ingresar usuarios y contraseñas erróneos
Perfiles de usuario	El sistema debe de tener dos perfiles distintos, administrador y empleado	El sistema pide seleccionar un usuario de empleado o administración para poder iniciar sesión
Alta de empleados	El sistema debe permitir al administrador dar de alta a	Solo el administrador puede dar de alta a empleados o

	empleados	incluso eliminarlos
Alta de productos	El sistema debe permitir que los empleados puedan dar de alta los productos	Los empleados así como el administrador pueden dar de alta los productos de inventario
Duplicidad	El sistema debe mostrar un mensaje de error al detectar un empleado o un producto duplicado o que ya existe	El sistema muestra los mensajes de error al detectar que el empleado o producto que se quiere dar de alta ya esta registrado en la base de datos
Generación de reportes	El sistema debe generar reportes de inventarios, clientes y empleados	El sistema proporciona una opción para generar reportes de venta, inventario, clientes y productos
Rendimiento	El sistema debe de ser veloz	El sistema responde rápidamente al momento de iniciar sesión y dar de alta diferentes productos y realizar las consultas
Usabilidad	El sistema debe de ser fácil de usar	La interfaz del sistema es fácil de entender además de ser fácil de utilizar sin necesidad

		de tener que explicar paso a paso ya que es intuitivo
Seguridad	El sistema debe de ser seguro	El sistema este encriptado, por lo que la información, así como las contraseñas de los usuarios está protegida
Respaldo	El sistema debe de respaldar la información cada 7 días	El sistema realiza los respaldos automáticamente de cada operación cada domingo y los sube a la base de datos

Conclusión.

Es importante realizar diferentes pruebas para asegurar la calidad del software, en nuestro caso realizamos diferentes ingresos de usuarios para iniciar sesión además de dar de alta a diferentes empleados y productos así como identificar los atributos de calidad que necesitamos verificar en nuestra lista de verificaciones, por lo que podemos concluir que el sistema funciona correctamente, podemos realizar pruebas mas extensas por lo que al no conocer como es la estructura del sistema en nuestras pruebas, es importante documentar los intentos, hora, procedimiento realizado para poder replicar el error, por lo que de esta forma podemos verificar que el error solo se muestra al realizar ese procedimiento por lo que se puede generar un reporte para su solución, es necesario tener un plan de pruebas tipo checklist para saber que se realizara primero o asignar personal a realizar pruebas a diferentes módulos del sistema, podemos realizar pruebas de uso del sistema no contempladas en el desarrollo como tratar de ingresar un producto como alta de empleado o un empleado como producto en busca de errores ya que el usuario final no siempre utiliza el sistema a como el desarrollador lo diseña