

## Actividad | 3 | Plan de pruebas y reglas para el versionado e incidencias.

### Aseguramiento de calidad.

Ingeniería en Desarrollo de  
Software.



TUTOR: Jessica Hernández Romero.

ALUMNO: Carlos Ariel Nicolini

FECHA: 07/04/2024

## Índice

<b>Introducción .....</b>	<b>3</b>
<b>Descripción .....</b>	<b>4</b>
<b>Justificación .....</b>	<b>6</b>
<b>Desarrollo.....</b>	<b>7</b>
• <b>Plan de pruebas integral.....</b>	<b>7</b>
• <b>Reglas para el versionado e incidencias .....</b>	<b>10</b>
<b>Conclusión.....</b>	<b>13</b>
<b>Referencias.....</b>	<b>14</b>

# Introducción

En un mercado en constante cambio y competencia creciente, la calidad del software y la creación de una buena experiencia de usuario son cruciales. En plan de pruebas testing aparece así como uno de los pasos indispensables para lograr que un software destaque entre otros.

Un plan de pruebas de software es una guía detallada para asegurar la calidad del software final a través de la realización de pruebas exhaustivas y efectivas antes de su lanzamiento. Es un documento que describe los pasos a llevar a cabo y el enfoque de las pruebas en un proyecto de desarrollo de software.

Su objetivo es garantizar la calidad y el rendimiento del software antes del lanzamiento. UN plan de pruebas establece objetivos y metas específicas, proporciona una estructura para las actividades de testing (cuales se realizaran, cómo y por quien), determina los criterios de aceptación de las pruebas, facilitando la evaluación objetiva para determinar si el software o servicio es apto para su lanzamiento.

# Descripción

## **Contextualización:**

La compañía Cajas y derivados S.A. de C.V. ha crecido tanto que necesita llevar si control de inventario y clientes en un sistema multiplataforma en línea. De la misma manera, se pretende que este control se pueda acceder desde cualquier dispositivo con acceso web.

## **Requerimientos técnicos:**

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

## **Se pide que el programa presente las siguientes características:**

- Se le considera al personal ingresar su número de empleado y contraseña, los cuales serán consultados en la base de datos. De esta manera, si el usuario no existe o la contraseña no corresponde a este, se le informara 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 proporcionara 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.

**Actividad**

De acuerdo a la contextualización presentada realizar:

- Crear un plan de pruebas integral, incluir: cronograma y definición de roles del equipo de desarrollo. Además especificar estándares.
- Establecer las reglas para el versionado y manejo correcto de incidencias

## Justificación

En esta actividad diseñaremos un plan de prueba con su cronograma y roles, además explicaremos que tipo de versionado utilizaremos y que plan contra incidencias aplicaremos.

La finalidad de este ejercicio es poder realizar un plan de pruebas, entender su funcionamiento, finalidad, como se asignan roles y responsabilidades, además que métodos de versionamientos utilizaremos con nuestros sistemas para tener una mejor administración de ellos en todas sus etapas.

Y otro punto importante es como se trataran las incidencias, las cuales deben ser atendidas de modo serio y solucionarlas antes de que se materialicen y se conviertan en un problema para la compañía y eso genere una interrupción en los servicios, cosa que sería muy catastrófica para la empresa, ya que si los servicios se ven impactados y detenidos, eso generara perdidas no solo monetarias sino también afectara el prestigio no solo del servicio o sistema sino de la compañía, cosa que si es una empresa de servicios los clientes pueden verse afectados y cambiar de prestador de servicios.

Parte de este ejercicio me es muy familiar en este momento, ya que se está pasando una situación que aquí se pretende evitar y es una situación muy complicada de pasar.

Este trabajo fue subido al siguiente enlace de GitHub

<https://github.com/CarlosNico/Aseguramiento-de-la-Calidad>

# Desarrollo

## Plan de pruebas integral

En esta parte del proyecto se realizara un plan de pruebas integral al sistema multiplataforma en línea que se ha desarrollado para la compañía Cajas y derivados S.A de C.V. A continuación se incluirá el cronograma de actividades así mismo como los roles y responsabilidades dentro de mismo.

### Cronograma de pruebas

Fases	Actividades	encargado	Duración (semanas)				
			1	2	3	4	5
Objetivos	Objetivo del plan de pruebas	Lider de pruebas					
	Recursos disponibles	Lider de pruebas					
Planificacion	Asignacion de roles y responsabilidades	Lider de pruebas					
	Tipos de pruebas a realizar	Equipo de pruebas					
	Herramientas de pruebas a utilizar	Lider de pruebas					
	Definicion de tipos de pruebas	Lider de pruebas					
	Estimacion de tiempos	Lider de pruebas					
	Preparacionde ambiente de pruebas	Equipo de operaciones					
	Programacion de pruebas	Lider de pruebas					
Ejecucion	Ejecucion de pruebas	Equipo de pruebas					
	Registro de resultados	Analista					
	Identificacion y resolucion de riesgos encontrados	Analista - Equipo de pruebas					
	Seguimiento y resolucion de defectos encontrados	Equipo de desarrollo - Analista					
	Mitigacion de riesgos y vulnerabilidades	Equipo de desarrollo					
Seguimiento	documentación y resolucion de fallos	Analista					
Cierre	Resumen de los resultados de las pruebas.	Lider de pruebas					
	Entrega del sistema	Lider de pruebas					

### Roles y actividades

#### Líder de pruebas:

- Selecciona las actividades, técnicas, documentación, el enfoque de las pruebas y las estrategias que se ejecuten.
- Definir los criterios de terminación de las pruebas
- Presentación de informes de gestión.
- Gestionar el control sobre la planificación de pruebas

- Generar y actualizar el Test Summary del proyecto.

**Equipo de pruebas:**

- Analizar el requerimiento para obtener casos de prueba.
- Describir el proceso para la ejecución de las pruebas.
- Generar el diccionario de datos que será empleado en la ejecución de las pruebas.
- Identificar y priorizar las condiciones de los casos de prueba para su ejecución.
- Identificar los defectos que puedan existir en el objeto de pruebas.
- Registrar los defectos identificados de la ejecución de las pruebas en la matriz de defectos.
- Categorizar la severidad y prioridad de los incidentes detectados.
- Hacer del conocimiento de todos los involucrados en el proceso de desarrollo, el avance de la calidad del entregable, difundiendo el estatus de las pruebas ejecutadas y de los defectos detectados.
- Registrar en la matriz de defectos aquellos incidentes que sean detectados en las pruebas.

**Equipo de desarrollo:**

- Implementa el requerimiento solicitado asegurándose del cumplimiento de la calidad del mismo.
- Ejecuta pruebas unitarias.
- Deberá notificar obligatoriamente al equipo de pruebas, que ya puede iniciar la fase de ejecución de sus pruebas, especificando el alcance que se ha implementado y deberá ser evaluado.
- Realizar la corrección de los defectos encontrados.
- Mantener actualizado el estatus de atención en que se encuentran los defectos a su cargo.



**Analista:**

- Asegurar que el requerimiento contenga de manera clara y específica los criterios de aceptación del desarrollo solicitado por el cliente.
- De ser requerido deberá especificar o aclarar con el usuario solicitante, la información que le es proporcionada al equipo de pruebas y llegue a ser ambigua.
- Analizar todos aquellos defectos que sean detectados por el equipo de pruebas y los incidentes detectados en las pruebas de aceptación, para evaluar si procede o no como defecto.

**Equipo de operaciones:**

- Configura los ambientes de prueba, incluyendo la carga del conjunto de datos, la configuración de parámetros del sistema y la instalación de herramientas.

**Como estándares de calidad nos apoyamos:**

ISO 9001 – Gestión de calidad al ser el mejor reconocido y sus requisitos son genéricos y aplicables a cualquier organización.

ISO 12207 Modelos de ciclos de vida del software para hacer el recorrido desde que surge la necesidad o idea hasta la retirada del software.

IEEE 730 – 2002 estándar for software Quality Assurance Plants

# Reglas de versionado e incidencias

Para nuestro proyecto elegimos el tipo de versionado tradicional, el cual consiste en asignarle dos números (mayor.menor), que van incrementando conforme el desarrollo del software aumente y se requiera la asignación de un nuevo nombre, código o número único. Aunque menos habitual, podemos también indicarse con otro número más (micro) y la fase desarrollo en que se encuentra el software.

Se aumenta el número cuando:

- Mayor: el software sufre grandes cambios y mejoras.
- Menor: Cuando el software sufre pequeños cambios y/o correcciones de errores.
- Micro: Cuando se aplica una corrección al software, y a su vez sufre pocos cambios.
- Fase: Se indica si se encuentra en una fase de desarrollo que no sea final o estable, es decir, una fase inestable o en pruebas. Se suele indicar con un guion seguido de la fase correspondiente en minúsculas, o un espacio seguido de la fase. Puede haber varias versiones en una misma fase, para indicar el avance en el desarrollo del software pero manteniendo la fase para indicar que todavía es inestable, añadiendo un número al final del nombre de la fase que va incrementando conforme se publiquen nuevas versiones de esta fase.

## Manejo de incidencias

Una buena gestión de incidencias nos servirá para prevenir o restaurar tan pronto como sea posible cualquier interrupción o reducción no planificada en la calidad del servicio.

Para su buena ejecución se deben seguir los siguientes pasos:

- Detección: es crucial monitorizar los recursos con el fin de identificar las posibles incidencias con una detección temprana y normalizar el servicio antes de que cause un efecto negativo en los procesos o servicios.

- Registro: Todas las incidencias en el servicio deben ser registradas minuciosamente y cada incidencia debe ser registrada de forma independiente. Dicho registro debe tener toda la información necesaria.
- Categorización: En esta actividad se determina el tipo preciso de la incidencia.
- Priorización en la gestión de incidencias: La prioridad de la incidencia suele ser:
  - 1- La urgencia: Rapidez con la que la incidencia necesita ser resuelta.
  - 2-El impacto: Se determina por el número de usuarios o servicios que se ven afectados.
- Diagnóstico inicial: Cuando el personal de soporte de primer nivel recibe una incidencia, la diagnostica en base a los síntomas y si está capacitado para ello, la resuelve.
- Escalado: Existen dos tipos de escalado:
  - 1- Funcional: El soporte de primer nivel se ve incapaz de resolver la incidencia y la asigna al grupo resolutor correspondiente.
  - 2- Jerárquico: En caso de que se den cierta circunstancia (incidencias graves o críticas, riesgo de incumplimiento del SLA) que se deban notificar a los responsables del servicio correspondiente.
- Investigación y diagnóstico: Si la incidencia hace referencia a un fallo en el sistema, lo más probable es que se necesite investigar la causa del fallo.

Las tareas más comunes son las siguientes:

  - 1- Establecer que es lo que no funciona correctamente y para que secuencia de acciones del usuario (casuística).
  - 2- Establecer el impacto potencial de la incidencia.
  - 3- Determinar si la incidencia está producida por la implantación de un cambio.
  - 4- Busca en la base de datos de conocimiento (base de datos de errores conocidos, registro de incidencias, etc.) posibles soluciones y/o workarounds.

- Resolución: Una vez identificada una posible solución, se debe aplicar y someter a pruebas. Una vez confirmada su efectividad, la incidencia se considera resuelta y se le asigna al equipo de Service Desk para su cierre.

Además, es importante registrar todas las acciones llevadas a cabo para resolver la incidencia en su historial correspondiente.

- Cierre : Antes de cerrar la incidencia el equipo de Service Desk deberá validar lo siguiente:

- 1- El usuario está satisfecho con la resolución de la incidencia.
- 2- El cierre ha sido categorizado.
- 3- Se han cumplido con todos los datos necesarios.
- 4- Si es un problema recurrente. En este caso, generar un problema.
- 5- Se puede pasar una encuesta de satisfacción al usuario.

## Conclusión

En el mundo laboral es muy importante lo aprendido en esta clase, una buena elaboración de un software o servicio es primordial, pero no es posible sin ejecutar un plan de pruebas bien definido y ejecutado, el cual nos va a permitir tener un producto de calidad que le dé al cliente una excelente experiencia de usuario, ya que como dicen, el cliente solo va a conocer de nosotros por el producto presentado no por todo el trabajo que está detrás de él.

Además el uso de un correcto versionado evitara problemas y confusiones y es un método muy eficaz para categorizar y documentar software a través de su proceso de vida, desde que se crea, se prueba hasta que llega a su versión estable.

Y otro punto visto en este trabajo es el correcto tratamiento de las incidencias, una correcta detección y trato de una indecencia puede hacer mucha diferencia, en este momento en la empresa se está pasando por una situación donde no se aplicó correctamente ese proceso y una incidencia se materializo en un problema grave que hasta ahora no hemos podido solucionar de fondo y esta afectando de manera grave a la empresa.

El trabajo fue subido al siguiente enlace de Github

<https://github.com/CarlosNico/Aseguramiento-de-la-Calidad>

## Referencias

Ayala, C. (2020, October 23). Qué es la Gestión de incidencias ITIL y sus principales actividades.

ServiceTonic. <https://www.servicetonic.com/es/itil/itil-v3-gestion-de-incidencias/>

Crear un buen Plan de Pruebas. (2021, July 15). Software Testing Bureau.

<https://www.softwaretestingbureau.com/crear-un-buen-plan-de-pruebas/>

icr\_edit. (2023, May 25). Plan de pruebas de software: 8 pasos para realizarlas. icaria Technology.

<https://icariatechnology.com/plan-de-pruebas-de-software/>

Medina, I. F. (2022, December 5). Los estándares de calidad del software más importantes. Blog de

hiberus; Hiberus. <https://www.hiberus.com/crecemos-contigo/los-estandares-de-calidad-del-software-mas-importantes/>

Wikipedia contributors. (n.d.). Versionado de software. Wikipedia, The Free Encyclopedia.

[https://es.wikipedia.org/w/index.php?title=Versionado\\_de\\_software&oldid=155173463](https://es.wikipedia.org/w/index.php?title=Versionado_de_software&oldid=155173463)

(N.d.). Studocu.com. Retrieved May 13, 2024, from [https://www.studocu.com/es-](https://www.studocu.com/es-mx/document/universidad-tecnologica-de-la-mixteca/procesos-de-manufactura/roles-en-el-proceso-de-pruebas/8141013)

[mx/document/universidad-tecnologica-de-la-mixteca/procesos-de-manufactura/roles-en-el-proceso-de-pruebas/8141013](https://www.studocu.com/es-mx/document/universidad-tecnologica-de-la-mixteca/procesos-de-manufactura/roles-en-el-proceso-de-pruebas/8141013)