

Proyecto final - Plan de pruebas y reglas para el versionado e incidencias

Aseguramiento de la Calidad

Ingeniería en Desarrollo de Software

Tutor: Jessica Hernández Romero

Alumno: Fernando Pedraza Garate

Fecha: 30 de Abril del 2023

Índice

Etapa 1 – Análisis de Atributos y Plan de Pruebas de Caja Negra.

- Introducción. Pág. 4
- Descripción Pág. 5
- Justificación Pág. 6
- Desarrollo Pág. 7-10
 - Atributos de calidad
 - Plan de pruebas de caja negra
 - Lista de verificación de requerimientos

Etapa 2 – Casos y Matriz de Pruebas

- Introducción. Pág. 11
- Descripción Pág. 12
- Justificación Pág. 13
- Desarrollo Pág. 14-18
 - Diagrama de Casos de Uso
 - Casos de Prueba
 - Matriz de Prueba

Proyecto final - Etapa 3 – Plan de pruebas y reglas para el versionado e incidencias.

- Introducción. Pág. 19
- Descripción Pág. 20
- Justificación Pág. 21
- Desarrollo Pág. 22-26
 - Plan de pruebas integral
 - Reglas para el versionado e incidencias
- Conclusión Pág. 27-29
- Referencias. pág. 30-32

Introducción

De acuerdo con la Sociedad Americana para el Control de la Calidad, la calidad es “el conjunto de características de un producto, proceso o servicio que le confieren aptitud para satisfacer las necesidades de un usuario o cliente”, sin embargo, la calidad es subjetiva, porque depende de los atributos elegidos para medirla; y es circunstancial, ya que el conjunto de atributos elegidos puede variar en diferentes situaciones, por ejemplo, cuando se aplica el concepto de calidad al software, esta deja de ser subjetiva, debido a que se determinan los atributos de calidad; pero no deja de ser circunstancial, ya que, en ciertas situaciones, un determinado conjunto de características de calidad pueden ser más importantes que en otras, razón por la que es importante definir los atributos que permitirán medir este tipo de calidad, tales como “Confiabilidad, Usabilidad, Adaptabilidad, Funcionalidad, Eficiencia, y Portabilidad”, donde además de la calidad del software, es importante abordar el SQA, de sus siglas en inglés (Aseguramiento de la calidad de Software), ya que no puede existir uno sin el otro, en donde el aseguramiento de la calidad del software es un conjunto de actividades planificadas y ejecutadas sistemáticamente que garantiza que el software que se está construyendo sea de alta calidad.

Descripción.

En base al contexto de la compañía Cajas y Derivados S.A. de C.V. debido al crecimiento que ha tenido como empresa, requiere la implementación de un sistema multiplataforma en línea, que les permita acceder desde cualquier dispositivo con acceso web para su control de inventarios y clientes, tomando en cuenta los requerimientos técnicos este, deberá ser desarrollado con los lenguajes HTML5, CSS, Python y un patrón de arquitectura MVC, así como el modelo encargado de conectarse a la base de datos sea postgresQL, contemplando las siguientes características:

El control de acceso será por número de empleado y contraseña, los cuales serán consultados en la base de datos y si el usuario no existe o la contraseña no corresponde a este, se le informe mediante un mensaje de error, que toda la información presentada por la aplicación se muestre conforme a los formatos corporativos definidos por la organización, que el sistema permita tener dos perfiles, uno como administrador para que el sistema le permita dar de alta los diferentes tipos de empleados, contemplando que no se permita duplicidad de información, de presentarse dicha situación el sistema mostrara un error, y el otro como empleado, donde el sistema les permita dar de alta los productos del inventario, evitando duplicidad de información, en caso de que el producto ya existiera, mostrará un mensaje de error, así como la generación de reportes de inventarios, de clientes y de empleados. Contemplando los requerimientos funcionales el sistema deberá ser veloz, fácil de entender, manteniendo la información encriptada y deberá crear respaldados cada siete días.

Justificación.

Se recomienda emplear esta solución ya que como toda empresa que tiene excelentes resultados y busca crecer o está en pro de expandirse deberá contemplar y estar dispuesta a invertir en el cambio cultural y organizacional de la misma, contemplando en todo momento el aseguramiento de la calidad de sus servicios o productos, con las ventajas y desventajas que esto le conlleva, tal como, “El aumento de la productividad como de sus ganancias, llevando de la mano la automatización de sus procesos, la implementación de nuevas tecnologías, permitiendo mejorar la calidad de sus productos, logrando captar nuevos clientes y mercados, aumentando el empleo y la competitividad, obteniendo un equipo de trabajo de calidad, sin perder de vista que se puede presentar la falta de comunicación con el equipo de trabajo, mayores costos de producción y mantenimiento, cambio en la legislación y pagos de impuestos, así como el impacto ambiental y el incremento de desperdicios derivado de su crecimiento”.

Desarrollo.

Atributos de Calidad

Al realizar esta implementación se observa que contará con los siguientes atributos:

Confiabilidad, al mantener encriptada la información, al no permitir duplicidad, así como accesos no permitidos y la generación de respaldos que eviten la pérdida de información.

Usabilidad, al ser desarrollado de forma sencilla para un mejor entendimiento por parte de los usuarios o clientes.

Adaptabilidad, al contemplar su funcionamiento desde cualquier dispositivo con acceso web.

Funcionalidad, al contemplar que sea veloz, fácil de entender, manteniendo la encriptación de la información, y la creación de respaldos.

Eficiencia, al contemplar que sea veloz para obtener el rendimiento apropiado.

Portabilidad, al contemplar su ejecución desde cualquier dispositivo con acceso web.

Plan de Pruebas de Caja Negra

Descripción del caso: El control de acceso será por número de empleado y contraseña, los cuales serán consultados en la base de datos y si el usuario no existe o la contraseña no corresponde a este, se le informe mediante un mensaje de error, que el sistema permita tener dos perfiles, uno como administrador para que el sistema le permita dar de alta los diferentes tipos de empleados, contemplando que no se permita duplicidad de información, de presentarse dicha situación el sistema mostrara un error, y el otro como empleado, donde el sistema les permita dar de alta los productos del inventario, evitando duplicidad de información, en caso de que el producto ya existiera, mostrará un mensaje de error, así como la generación de reportes de inventarios, de clientes y de empleados.

Técnica de pruebas de caja negra: Requerimiento funcional / Caso de uso

Caso 1.1: Datos de entrada: Ingresar número de empleado y contraseña, el sistema consulta información en BD.

Resultado esperado (Salida): Información inexistente o errónea, el sistema manda mensaje de error, la información es correcta, se consigue el acceso al sistema.

Caso 1.2: Datos de entrada: Ingreso como administrador, el sistema permitirá el alta, edición, y eliminación de empleados en el sistema y la base de datos.

Resultado esperado (Salida): Información duplicada o errónea, el sistema manda mensaje de error, si la información es nueva o correcta, se consigue crear nuevos registros y el mantenimiento de la misma.

Caso 1.3: Datos de entrada: Ingreso como empleado, el sistema permitirá el alta, edición, y eliminación de los productos del inventario en el sistema.

Resultado esperado (Salida): Información duplicada o errónea, el sistema manda mensaje de error, si la información es nueva o correcta, se consigue crear nuevos registros y el mantenimiento de la misma.

Lista de verificación de requerimientos

Requerimientos Técnicos	Requerimientos Funcionales
Desarrollado en HTML5 y Python	Velocidad de transferencia de archivos
Utiliza CSS	Cifrado seguro
Gestor PostgresQL	Contraseñas encriptadas
MVC (Modelo, Vista, Controlador)	Patrón de arquitectura o diseño

Introducción.

Los casos de prueba son los escenarios que se utilizan para medir la funcionalidad de una aplicación a través de un conjunto de ciertas acciones o condiciones para verificar los resultados esperados, constando de varios pasos, como pasos de prueba, datos de prueba, condiciones previas y posteriores, mismas que serán desarrolladas para un escenario de prueba en particular, que se puedan aplicar a cualquier aplicación de software, y se realizan a través de pruebas manuales y automatizadas, o cualquier herramienta de gestión de pruebas. En otras palabras, un caso de prueba es un conjunto de acciones ejecutadas para autenticar la funcionalidad de una aplicación de software.

Por otro lado las matrices de prueba son una herramienta eficaz para tener una mejor cobertura de todos los requerimientos a través de los casos de pruebas, que nos permiten obtener trazabilidad, así como, la colectividad de pruebas, la trazabilidad permitirá identificar los requisitos con otros artefactos, como pruebas, validaciones, casos de uso, de código, etc., permitiendo responder a los cambios de una forma más controlada y con mayor información, en consecuencia, anticiparse a lo que un cambio puede significar, haciendo más sencillo el trabajo en equipo y de colaboración, cubriendo escenarios que afectan otros requerimientos, pudiendo modificar o adicionar tareas nuevas para mejorar el software, además de poder crear y establecer prioridades.

Descripción.

*

Continuando con el contexto de la etapa 1 de la compañía Cajas y Derivados S.A. de C.V. se reconocerá la importancia de los casos de prueba, los cuales constan a partir de crear una entrada o ingreso de solicitud de información, que deberán crear o realizar alguna función dentro del sistema, validando que se cumpla con las precondiciones establecidas anteriormente y en caso de cumplirse, permitirán dicha entrada, originando la salida o respuesta a lo solicitado dentro del sistema, siempre y cuando también se cumpla con la post condición establecida, de igual forma validando que también se cumpla, de ser así, se mostrara en pantalla o se realizara una acción si este cumple con el resultado de salida. Y en el caso de las matrices nos permitirán ir armando el sistema, como si fuera un rompecabezas, con la identificación de los casos de prueba que se elaboren de manera aislada, con el objetivo claro de escenificar el software permitiendo medir la funcionalidad del mismo.

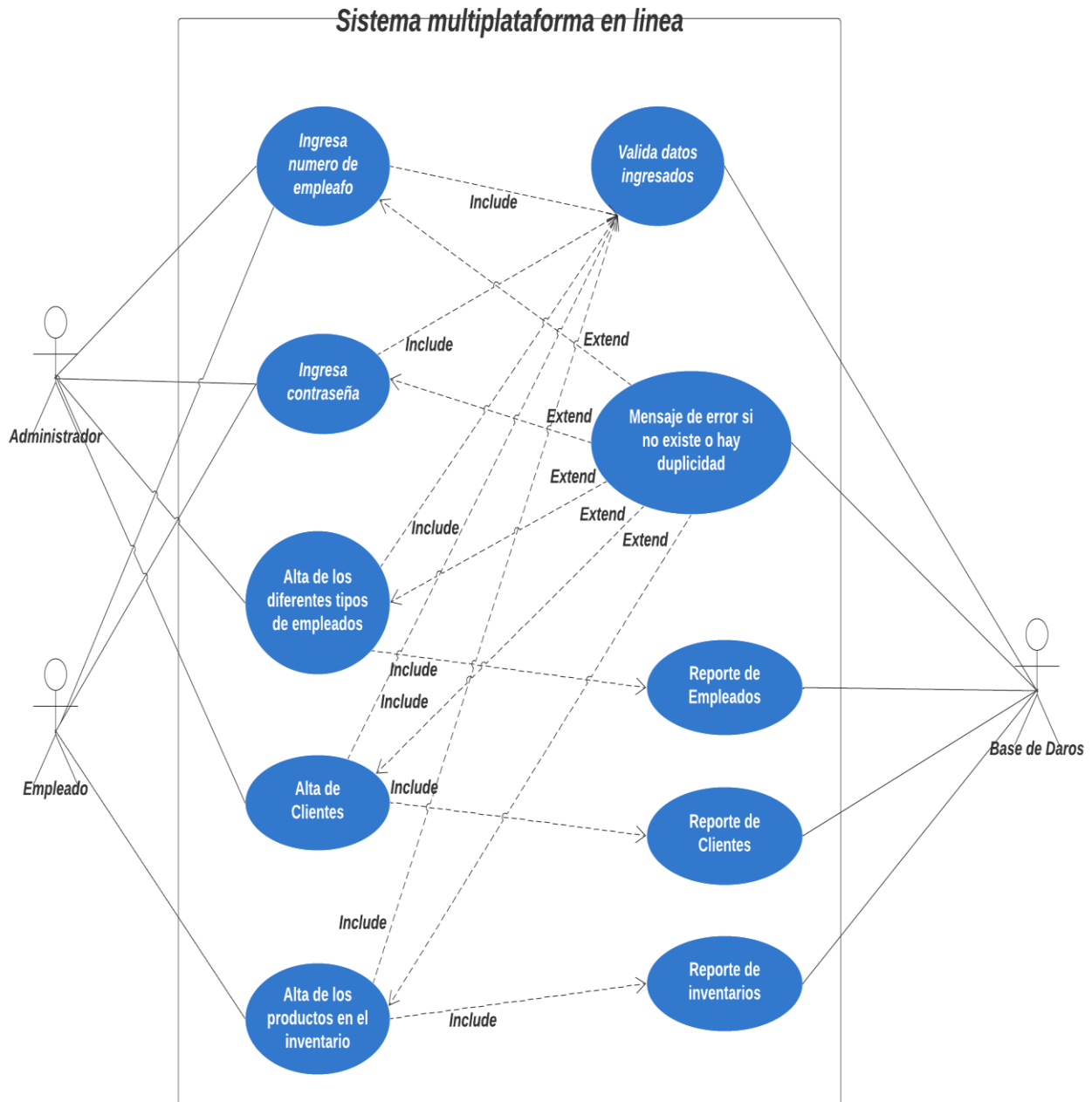
Justificación.

Se recomienda realizar en este y todos los casos referidos para el aseguramiento de la calidad en el desarrollo de software, ejecutar los casos de prueba por ser una parte crítica de los servicios en el desarrollo de software, para así poder alcanzar un alto nivel de calidad que no solo beneficiara al usuario final, sino que también evitara errores en el funcionamiento del software, evitando que perjudiquen el propósito del mismo, y es en estas circunstancias donde la automatización de pruebas de software juega un rol crucial, debido a que este tipo de pruebas implican un gran inversión de recursos que se podrían dedicar y enfocar a otros aspectos específicos del software, que pudieran ser más complejos, teniendo como beneficios el ahorro de tiempo y esfuerzo, verificar que los errores pasados no se reproduzcan, garantizar que el producto sea funcional para el usuario, y asegurar que no hay ninguna funcionalidad rota en el “build” o proceso de creación y compilación de software.

Desarrollo.

Diagrama de caso de uso

fernando pedraza | April 24, 2023



Tablas de casos de prueba

Nombre del Proyecto: SML (Sistema Multiplataforma en línea)						
Caso de Prueba						
Prioridad de Prueba: Alta			Prueba diseñada por: Fernando Pedraza Garate			
Nombre del módulo: Caso 1.1			Fecha del diseño de prueba: Abril 24, 2023			
Nombre de la prueba: Control de acceso			Prueba ejecutada por: Fernando Pedraza Garate			
Descripción: validación de información			Fecha de la ejecución de la prueba: Abril 24, 2023			
Pasos	Pasos de Prueba	Datos de Prueba	Resultado esperado	Resultado actual	Estatus (Califico / No califico)	Observaciones
1	Ingreso de número de empleado	Permitir el ingreso de texto alfa numérico a través del teclado	Visualizar en el campo seleccionado el dato ingresado	Si permite el ingreso alfanumérico	No	En caso de no existir el numero ingresado debería poner el texto escrito en rojo
2	Ingreso de la contraseña	Permitir el ingreso de texto alfa numérico a través del teclado	Ocultar los caracteres ingresados con asteriscos	Si se oculta el texto ingresado con asteriscos	No	Evitar que el carácter escrito se visualice antes de sustituirlo por el asterisco
3	El sistema valida los datos ingresados con los de la BD	La validación de información, antes de permitir el acceso al sistema	Mandar mensaje de error en caso de información errónea o inexistente	Se genera la validación con éxito	Si	
4	Se permite el acceso al sistema	Acceso correcto al sistema	No permitir accesos que no cumplan con los criterios establecidos	Gestión correcta de la información ingresada	Si	

Nombre del Proyecto: SML (Sistema Multiplataforma en línea)						
Caso de Prueba						
Prioridad de Prueba: Alta			Prueba diseñada por: Fernando Pedraza Garate			
Nombre del módulo: Caso 1.2			Fecha del diseño de prueba: Abril 24, 2023			
Nombre de la prueba: Ingreso como administrador al sistema			Prueba ejecutada por: Fernando Pedraza Garate			
Descripción: Alta, modificación y eliminación de información de los diferentes tipos de empleados			Fecha de la ejecución de la prueba: Abril 24, 2023			
Pasos	Pasos de Prueba	Datos de Prueba	Resultado esperado	Resultado actual	Estatus (Califico / No califico)	Observaciones
1	Ingreso al sistema como administrador	Que se tenga activa la opción para el alta, modificación y eliminación de información de empleados	Mantenimiento adecuado de la información según el perfil de cada registro	Si cuenta con las opciones para el alta, modificación y eliminación de registros de empleados	Si	
2	Creación de un registro nuevo	Tener activa la ventana para la creación de empleados nuevos	Mandar mensaje de error en caso de ya existir el registro	Se creó satisfactoriamente el registro como nuevo	No	Crear un flag para que sea autorizado en sistema el registro nuevo por parte del gerente a cargo de RH
3	Actualización de información existente de un registro	Tener activa la ventana para la modificación de información en registros existentes	Mandar mensaje de error en caso de información errónea o incompleta	Se genera la modificación con éxito	No	Crear un flag para que sea autorizada la modificación del registro en sistema por parte del gerente a cargo de RH
4	Eliminación de registros existentes	Tener activa la ventana para la eliminación de registros existentes	Mandar mensaje de confirmación para la eliminación del registro	Se elimina el registro de forma correcta	Si	

Nombre del Proyecto: SML (Sistema Multiplataforma en línea)						
Caso de Prueba						
Prioridad de Prueba: Media			Prueba diseñada por: Fernando Pedraza Garate			
Nombre del módulo: Caso 1.3			Fecha del diseño de prueba: Abril 24, 2023			
Nombre de la prueba: Ingreso como empleado al sistema			Prueba ejecutada por: Fernando Pedraza Garate			
Descripción: Alta, modificación y eliminación de los productos del inventario			Fecha de la ejecución de la prueba: Abril 24, 2023			
Pasos	Pasos de Prueba	Datos de Prueba	Resultado esperado	Resultado actual	Estatus (Califico / No califico)	Observaciones
1	Ingreso al sistema como empleado	Que se tenga activa la opción para el alta, modificación y eliminación de productos del inventario	Mantenimiento adecuado de la información según el producto	Si cuenta con las opciones para el alta, modificación y eliminación de productos del inventario	Si	
2	Creación de un registro nuevo	Tener activa la ventana para la creación de productos nuevos	Mandar mensaje de error en caso de ya existir el registro	Se creó satisfactoriamente el registro como nuevo	Si	
3	Actualización de información existente de un registro	Tener activa la ventana para la modificación de información en registros existentes	Mandar mensaje de error en caso de información errónea o duplicada	Se genera la modificación con éxito	Si	
4	Eliminación de registros existentes	Tener activa la ventana para la eliminación de registros existentes	Mandar mensaje de confirmación para la eliminación del registro	Se elimina el registro de forma correcta	Si	

Matriz de Prueba

Nombre del sistema: SML (Sistema multiplataforma en línea)							
Fecha de revisión: Abril 25, 2023				Encargado de la revisión: Jessica Hernández Romero			
No.	Módulo	Descripción	Núm. de pasos que aprobaron	Núm. de pasos que no aprobaron	Resultado esperado del módulo en general	Resultado real del módulo en general	Califico / no califico
CP1	Caso 1.1	Control de acceso	2	2	Satisfactorio	Funcional	No
CP2	Caso 1.2	Ingreso como administrador	2	2	Satisfactorio	Funcional	No
CP3	Caso 1.3	Ingreso como empleado	4	0	Satisfactorio	Completo	Si

Introducción.

Para desarrollar software de calidad y libre de errores, el plan de pruebas y los casos de prueba son muy importantes, ya que el plan de pruebas de software STP (Software Test Plan) se diseña para determinar el ambiente de aplicación de los recursos, así como el calendario de las actividades de las pruebas, este debe identificar el dominio y sus características a probar, lo mismo que el tipo de pruebas a realizar, tomando como base el estándar de la norma 829-2008 del IEEE (Institute of Electrical and Electronics Engineers), también conocida como estándar 829 para software y prueba del sistema de documentación, actualmente es la norma establecida y la más utilizadas en el área de las ingenierías, sobre todo en informática y telecomunicación.

Una regla de versionado es la práctica de rastrear y gestionar los cambios en el código del software, estas son herramientas que ayudan a los equipos de software a gestionar los cambios realizados en el software a lo largo del tiempo, a medida que los entornos de desarrollo se aceleran, los sistemas de control de versiones ayudan a los equipos de software a trabajar de forma más rápida e inteligente, son especialmente útiles para para los equipos de DevOps (es un marco de trabajo entre los equipos de desarrollo de aplicaciones “Dev” y sus homólogos en operaciones tecnológicas “Ops”), y por medio de esto les ayuda a reducir el tiempo de desarrollo y a aumentar las implementaciones de forma exitosa.

Descripción.

Para este caso se deberá diseñar para las actividades propuestas, un plan integral a ejecutar en las pruebas que se van a aplicar, estas deberán incluir un cronograma, el cual permitirá establecer tiempos específicos para el cumplimiento de cada tarea identificada, permitiendo la entrega en tiempo y forma de cada una de ellas, así como también, la definición de los roles del equipo de desarrollo, ya que es de suma importancia identificar el “quien tiene que” delegando las tareas a realizar de forma adecuada, permitiendo cumplir con los tiempos establecidos para la entrega de cada una de las mismas así como del software mismo, esta deberá de especificar los estándares que se deben cumplir, basados en la norma de calidad 829-2008 establecida por el IEEE (Institute of Electrical and Electronics Engineers), estableciendo las reglas de versionado que permitirán trabajar más rápido y de forma inteligente, dando un buen seguimiento a cada una de las actualizaciones permitiendo el manejo correcto de las incidencias presentadas.

Justificación.

Basados en la necesidad de la compañía Cajas y Derivados, S.A. de C.V. para el control de su inventario y registro de clientes, derivado del crecimiento que ha tenido, y al tener la necesidad de implementar un sistema multiplataforma, se recomienda crear un plan de pruebas integral, que incluya un cronograma y definición de los roles del equipo de desarrollo, para poder tener un registro de los cambios realizados o por realizar en su sistema, ya que estos permitirán trabajar más rápido y de forma inteligente en cualquier cambio o actualización de su software, tomando en cuenta la especificación de los estándares establecidos, para así dar cumplimiento a la norma 829 establecida por la IEEE, por lo que se deberán establecer reglas para el versionado, el cual permitirá llevar un registro y control de los cambios realizados para su mejora o corrección de errores de software, así como el correcto manejo de incidencias.

Desarrollo.

Plan de Pruebas Integral

Cronograma de Actividades								
Fases	Actividades	Responsable	Fecha de entrega	Semana compromiso de entrega				
				1	2	3	4	5
Establecer objetivos	Analizar el problema	AC	05/05/2023					
	Identificar nuevas funcionalidades a probar	AC	05/05/2023					
	Identificación de funcionalidades existentes a probar	AC	05/05/2023					
	Propuesta de solución	AC	05/05/2023					
Planificación	Asignación de roles	AC / AS	10/05/2023					
	Gestión de tiempos	AC / AS	10/05/2023					
	Diseño de la Base de Datos	DS / PS	12/05/2023					
Ejecución	Diseño del sistema	DS / PS	15/05/2023					
	Desarrollo del sistema	DS / PS	16/05/2023					
	Pruebas funcionales definidas por el ISTQB, prueba de caja negra	Tester	17/05/2023					
	Pruebas no funcionales	Tester	18/05/2023					
	Pruebas de caja blanca	Tester	19/05/2023					
	Pruebas de regresión	Tester	19/05/2023					
	Determinación de entornos requeridos	AS	19/05/2023					
	Pruebas y testeo	Tester	25/05/2023					
Seguimiento y control	Revisiones finales	Todos	26/05/2023					
	Identificación de riesgos	AC	26/05/2023					
	Planes de respuesta	AC / AS	29/05/2023					
Cierre	Evacuación final	CAB	30/05/2023					
	Entrega del sistema	AS	31/05/2023					

ISTQB (International Software Testing Qualifications Board).

Organización dedicada a la calificación y certificación de profesionales y empresas en el área de pruebas de software

AS - Arquitecto de Software
AC - Analista de Calidad
DS - Desarrollador de Software
PS - Programador de Software
TS - Tester de Software
T - Todos
CAB - Comité de cambios

Reglas para el versionado e incidencias

Existen diferentes tipos de reglas que se pueden utilizar, empezando con las versiones por número, algo común es realizar el manejo de versiones mediante 3 números: X.Y.Z y cada uno indica una cosa diferente:

El primero (X) se le conoce como versión mayor y nos indica la versión principal del software.

Ejemplo: 1.0.0, 3.0.0

El segundo (Y) se le conoce como versión menor y nos indica nuevas funcionalidades. Ejemplo:

1.2.0, 3.3.0

El tercero (Z) se le conoce como revisión y nos indica que se hizo una revisión del código por algún fallo. Ejemplo: 1.2.2, 3.3.4

Una vez que se conoce el significado de cada número, viene una pregunta importante:

¿Cómo saber cuándo cambiarlos y cuál cambiar?

La versión mayor o “X” es cuando se agregan nuevas funcionalidades importantes, puede ser como un nuevo módulo o característica clave para la funcionalidad, la versión menor o “Y”, cuando se hacen correcciones menores, cuando se arregla un error y se agregan funcionalidades que no son cruciales para un proyecto, la revisión o “Z”, cambia cada vez que se entrega el proyecto, y para las versiones por estabilidad, además de tener las versiones por números se puede agregar una clasificación por estabilidad del proyecto. Las opciones que tenemos para esto son: Alpha, Beta.

Alpha es una versión inestable la cual es muy probable que tenga muchas opciones que mejorar, pero se requiere sea probada para poder encontrar errores y se puedan poner a prueba las funcionalidades, en la mayoría de los casos se puede decir que está casi listo el producto.

Ejemplo: 1.0Alpha, 1.0a1, 1.0a2.

Beta una versión más estable que Alpha en la que se cuenta con el producto en su totalidad, y se desea realizar pruebas de rendimiento, usabilidad y funcionamiento de algunos módulos para ver cómo funciona bajo un ambiente no tan controlado. Aquí aparece el nombre de Beta Tester que se escucha a menudo en el mundo del software. Ejemplo: 2.0Beta, 2.0b, 2.0b1

El siguiente paso es RC (Release Candidate), que es el último toque fino del software antes de salir y después de pasar por Beta. Ejemplo: 3.0-RC o también 3.0-RC1.

Existen más cosas que se manejan con versiones, en el mundo del software se sabe que además de los programas que se usan habitualmente existen softwares más complejos como los videojuegos y sistemas operativos, en ambos casos se necesitan considerar los casos de los parches y las fechas de lanzamientos, ambos elementos también pueden influir en el control de versiones.

En la versión de parche, se agrega un dígito para señalar dicho parche, y se obtiene un resultado como el siguiente: X.Y.Z, para posteriormente obtener el siguiente resultado de la versión: X.Y.Z.P así que P sería el número del parche:

Ejemplo: 1.2.5.2, 02.03.03.01

Para la versión por fecha, en algunos casos se necesita saber exactamente la fecha en que se publicó el software, entonces se podrá utilizar el manejo de versiones por fecha. Este tiene muchas variaciones, se puede tener diferente orden del año, mes y día, por ejemplo: 1.2.3.1543 donde 15 es el año 2015, 4 es el mes y 3 el día, como ya se mencionó anteriormente se podrían obtener diferentes acomodados y formatos: 1.2.3.4315 o 1.2.3.201543, 1.2.3.1534

Versión	Fecha	Descripción	Autor
V.0.1	Abril 12, 2023	Primera versión del documento	Fernando Pedraza Garate / Analista de Calidad
V.0.2	Abril 23, 2023	Mostrar el texto en color rojo como indicativo de que el dato ingresado no existe o hay un error	Fernando Pedraza Garate / Analista de Calidad
V.0.3	Abril 23, 2023	Evitar que se muestren los caracteres ingresados como contraseña	Fernando Pedraza Garate / Analista de Calidad
V.0.4	Abril 23, 2023	Crear un apartado de autorización por parte del departamento correspondiente para el alta de empleados en la BD.	Fernando Pedraza Garate / Analista de Calidad
V.0.5	Abril 23, 2023	Crear un apartado de autorización por parte del departamento correspondiente para la modificación o eliminación de empleados en la BD.	Fernando Pedraza Garate / Analista de Calidad
V.1.0	Abril 30, 2023	Aprobación de la DTI y CAB	Jessica Hernández Romero / DTI / CAB

Conclusión.

En conclusión la elaboración de este tipo de análisis permite identificar de forma sencilla los requerimientos que se deben de cubrir en la elaboración de un producto o servicio, en el cual se debe garantizar que el procedimiento para la elaboración de los mismos sea el correcto, cubriendo todos los requerimientos de calidad establecidos en las normas de calidad establecidas para la producción o elaboración de los mismos, asegurando que sean de gran calidad para el usuario final o cliente, permitiendo identificar de forma sencilla los pros y contras en su funcionalidad y el proceso de elaboración, evitando errores que pudieran generar gastos extraordinarios en el proceso, y por medio de la prueba de caja negra que se basa en la prueba lógica se logra identificar si se está cumpliendo con los requisitos establecidos o no, logrando identificar las áreas de oportunidad para mejorarlos e innovar nuevos productos o servicios.

La elaboración de los casos de prueba permite evaluar de forma independiente cada módulo del sistema, fraccionándolo de forma que se pueda hacer una revisión puntual en cada paso del proceso, ya sea para su corrección o mejora, y así obtener el funcionamiento esperado del mismo, permitiendo inspeccionar de forma adecuada su funcionalidad, así como la forma en que interactúa entre cada módulo, y sobre todo la usabilidad que se tendrá por parte de los usuarios finales, permitiendo que sea fácil de entender, manejar y administrar la información almacenada o registrada con la seguridad esperada. En el caso de la matriz de prueba nos permite identificar los casos puntuales de forma simple en base a los casos de prueba realizados con anterioridad de cada módulo revisado, abordándolos de forma que se puedan hacer los cambios o mejoras pertinentes al sistema permitiendo mejorar su funcionalidad y asegurando la calidad del mismo.

Es de suma importancia llevar un registro de cada cambio realizado en la actualización o mantenimiento de software para así evitar, como vulgarmente se dice: “reinventar el hilo negro”, en cada parte del proceso a cambiar, modificar, o crear, sin tener que partir de cero, a menos de que sea una implementación nueva, evitando pérdida de información ya sea por cambio o ausencia de personal, tomando en cuenta que no debe existir la exclusividad de algunas personas en los procesos del sistema y en la elaboración de los cambios a realizar, contando con la información por escrito en todo momento, por lo que la elaboración del plan de pruebas integral es primordial ya que ayudara en la gestión de las tareas, asignándoles un responsable para su elaboración, así como tiempos de entrega, permitiendo entregar el proyecto en las fechas establecidas con los clientes y por medio del versionado se podrá mantener un registro resumido de los cambios realizados al programa o sistema.

Referencias

Atributos de la calidad del software, (2023)

16 de Abril 2023, de Universidad México Internacional, sitio web:

<https://umi.edu.mx/coppel/IDS/mod/scorm/player.php>

Crecimiento de una empresa, ventajas y desventajas, (2023)

16 de Abril 2023, de Romina Maejo, sitio web:

<https://blog.nubox.com/empresas/crecimiento-de-una-empresa#:~:text=El%20crecimiento%20de%20una%20empresa%20aumenta%20el%20empleo%20y%20la, trabajo%20que%20antes%20no%20exist%C3%ADa.>

Pruebas de caja negra, (Febrero 20, 2017)

16 de Abril 2023, de PMOinformatica.com, sitio web:

<http://www.pmoinformatica.com/2017/02/pruebas-de-caja-negra-ejemplos.html>

¿Qué son los casos de prueba?, (Abril 21, 2023)

21 de Abril 2023, de visure, sitio web:

<https://visuresolutions.com/es/what-are-test-cases-how-to-write-software-related-test-cases/#:~:text=Los%20casos%20de%20prueba%20son,para%20verificar%20los%20resultados%20esperados.>

¿Qué es un caso de prueba de software? (Octubre 10, 2016)

23 de Abril 2023, de The QA Testing Channel, sitio web:

<https://www.youtube.com/watch?v=Jegndzw3DEs>

Testing de software: la importancia de automatizar los casos de prueba (Marzo 29, 2022)

23 de Abril 2023, de encora, sitio web:

<https://www.encora.com/es/blog/testing-de-software-la-importancia-de-automatizar-los-casos-de-prueba#:~:text=Entre%20los%20beneficios%20de%20la,funcionalidad%20rota%20en%20el%20build.>

Plan de pruebas (Abril 30, 2023)

30 de Abril 2023, de umi.edu.mx, sitio web:

https://umi.edu.mx/coppel/IDS/pluginfile.php/1894/mod_scorm/content/5/res/data/img10.png

¿Qué es el control de versiones? (Abril 30, 2023)

30 de Abril 2023, de ATLASIAN Bitbucket, sitio web:

<https://www.atlassian.com/es/git/tutorials/what-is-version-control#:~:text=El%20control%20de%20versiones%2C%20tambi%C3%A9n,en%20el%20c%C3%B3digo%20de%20software.>

Prueba de Software: 10 pasos para elaborar el plan de pruebas, (Enero 18. 2016)

30 de Abril 2023, de PMOinformatica.com, sitio web:

<http://www.pmoinformatica.com/2016/01/elaborar-plan-pruebas-software.html>

¿Cómo se definen las versiones del software?, (2017)

30 de Abril 2023, de EDteam, sitio web:

<https://ed.team/blog/como-se-deciden-las-versiones-del-software>