

# TALLER DE TESTING Y CALIDAD DE SOFTWARE

U2. Desarrollo y ejecución de casos de prueba. Semana 6



# **ESCUELA DE CONSTRUCCIÓN E INGENIERIA**

Director: Marcelo Lucero

# **ELABORACIÓN**

Experto disciplinar: Aída Villamar Gallardo.

Diseño instruccional: Carla Silva Alvarado.

# **VALIDACIÓN**

Experto disciplinar: Andrés del Alcázar

Jefa de Diseño Instruccional: Alejandra San Juan Reyes.

#### **EQUIPO DE DESARROLLO**

**AIEP** 

AÑO

2021



Aprendizaje esperado de la semana	4
ntroducción	4
1. Plan de pruebas de sistema, según estándares de la industria	5
1.1 Concepto de plan de pruebas de sistema	5
1.2 Objetivos	5
1.3 Estructura	6
1.4 Estándares de la industria	7
2. Relación de función de plan de pruebas con proceso de desarrollo de software, considerando análisis de planes de pruebas de proyectos en el mercado	
2.1 Análisis de planes de pruebas	1
3. Diseño de plan de pruebas de sistema para proyecto de software, según requerimientos del cliente	2
3.1Diseño de planes de prueba1	2
3.2Requerimientos de clientes	3
4. Implementación de plan de pruebas para proyecto de desarrollo de software según estándares de la industria	
4.1Implementación de planes de pruebas1	4
Conclusiones1	9
Referencias bibliográficas2	20



## Aprendizaje esperado de la semana

Implementan plan de pruebas de sistema para proyecto de desarrollo de software, considerando función y diseño, según estándares de la industria.

#### Introducción

- ¿Qué es un plan de pruebas?
- ¿Qué puedo obtener al considerar un plan de pruebas en mi proyecto de desarrollo?
- ¿Cuáles son los estándares asociados a las pruebas de software?

Hasta ahora, hemos ido conociendo las distintas pruebas que podemos aplicar a nuestros proyectos para obtener información acerca de la calidad del sistema.

Esta semana veremos, qué podemos definir en un plan de pruebas. Por medio del cual abarcaremos funcionalidades completas, planificando y detallando las pruebas que se utilizaran, incorporando además estándares que nos entregaran los lineamientos que debemos seguir tanto en los conceptos, definiciones, procesos de prueba, documentación, técnicas de prueba y en lo que son los modelos de evaluación de procesos para pruebas de software.



### 1. Plan de pruebas de sistema, según estándares de la industria.

#### 1.1.- Concepto de plan de pruebas de sistema

El concepto de plan de pruebas de sistema está asociado a la definición y planificación de un plan de trabajo que considera todos los pasos en cuanto a las actividades de prueba se trata, pudiendo por medio de este plan de pruebas dar a conocer el alcance de estas.

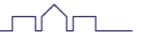
Dentro del plan de pruebas la planificación se considera como uno de los ítems con mayor relevancia al momento de probar un proyecto, es necesario tener bajo control la totalidad de los aspectos que comprende, como son: la manera en la que se va a probar, cuáles son los recursos que se tendrán disponibles, considerar las pruebas de regresión, la documentación asociada al proyecto y todo aquello que se encuentre relacionado.

Un punto importante es el relativo al presupuesto ya que si bien es cierto se debe destinar parte de él, la omisión de las pruebas puede generar que los problemas se arrastren hasta el entorno de producción, y en ese caso será más caro y lento solucionarlo.

La planificación de las pruebas es uno de los principales puntos a la hora de probar el software con éxito. En ausencia de un buen test plan es muy poco probable que las pruebas se ejecuten en un plazo estipulado e incluso se incrementara el costo del proyecto al consumir más tiempo y recursos. Esta planificación tiene que estar incluida dentro de la planificación general de un proyecto de software. Fuente: (Montecinos, 2020)

# 1.2.- Objetivos

Durante el proceso de desarrollo de un sistema es posible que existan errores que no hayan sido detectados, por ejemplo, en los requerimientos, o bien, en el diseño o en la funcionalidad. Por medio de la definición del plan de pruebas encontramos que los objetivos están enfocados a la detección antes de que el sistema sea entregado al cliente, apuntando siempre a la calidad del sistema y teniendo en consideración que entre más nos demoremos en la detección desde que se producen los defectos, el costo y el tiempo en solucionarlo van creciendo.



Según (Montecinos, 2020), los objetivos principales al realizar pruebas serán los siguientes:

- o Detectar un error específico.
- o Descubrir errores no descubiertos anteriormente.
- o Tener un buen caso de prueba.

Además, los atributos que debería tener una buena prueba son:

- o Intentar obtener la más alta probabilidad de encontrar un error.
- No debe ser redundante.
- o No debe ser ni demasiado sencilla ni demasiado compleja.

#### 1.3.- Estructura

Un plan de prueba es un documento por medio del cual se definen y consideran todos los elementos que están asociados con las actividades para asegurar la calidad, debemos considerar en la preparación un ambiente de pruebas que sea lo más parecido posible a lo que se tendrá al pasar a producción.

Si consideramos lo relacionado a la estructura del plan de prueba, debemos partir por señalar que el enfoque siempre debe ir direccionado por los requisitos que se han definido al comenzar el proceso de desarrollo de nuestro proyecto. Una vez que hemos definido lo anterior, pasamos a trabajar en cada uno de los puntos que serán incluidos dentro de la estructura del plan de pruebas.

Según (Montecinos, 2020) se indica que hay varias maneras de realizar un plan de pruebas. Una de ellas es aplicar la norma —ISO/IEC 29119 Software testing que sustituye a la norma —IEEE 829. En esta norma se indican los puntos que debe considerar un plan de prueba:

- o Identificador del plan de pruebas.
- o Introducción.
- o Elementos de la prueba.
- Características que probar y no probar.
- Método/enfoque de prueba (implementación de la estrategia de prueba).
- o Criterios de paso/fallo de la prueba.



- Criterios de entrada, salida, suspensión y reanudación de las pruebas.
- Entregables de las pruebas.
- o Tareas/hitos clave de las pruebas.
- o Cronograma.
- o Necesidades del entorno.
- Responsabilidades.
- o Dotación de personal y necesidades de capacitación.
- o Riesgos.
- o Aprobaciones.

#### 1.4.- Estándares de la industria

En la actualidad, podemos encontrar estándares que se encuentran regulados por diversos órganos internacionales, ellos han documentado dentro de lo que son sus normas, estándares o lineamientos para las pruebas que realizamos dentro del ámbito del software, entre estas organizaciones tenemos:

- ISO (Internacional Organization for Standardization) es la Organización Internacional de Normalización
- o IEEE (Institute of Electrical and Electronics Engineers) Instituto de Ingenieros Eléctricos y Electrónicos

Además de las organizaciones mencionadas podemos encontrar aquellas que se han dedicado a crear el ámbito de las pruebas y divulgar sus beneficios:

- ISTQB (International Software Testing Qualifications Board) Comité Internacional de Certificaciones de Pruebas de Software
- ISST (International Society for Software Testing) Sociedad Internacional para Pruebas de Software
- o TMMI (Test Maturity Model Integration) Modelo de Madurez de Pruebas

Dentro de las principales certificaciones están: ISO, ISTQB ISST, TMMI.

Dentro de las publicaciones de estandarización ISO podemos encontrar las siguientes normas asociadas:

**ISO / IEC / IEEE 29119-1: 2013** Fuente: (Standardization, 2013)



Ingeniería de software y sistemas. Pruebas de software. Parte 1: Conceptos y definiciones.

#### **ABSTRACTO**

El propósito de la serie ISO / IEC / IEEE 29119 de estándares de prueba de software es definir un conjunto de estándares acordados internacionalmente para las pruebas de software que puede ser utilizado por cualquier organización al realizar cualquier forma de prueba de software. ISO / IEC / IEEE 29119-1: 2013 facilita el uso de los demás estándares ISO / IEC / IEEE 29119 al introducir los conceptos y el vocabulario sobre los que se basan estos estándares, además de proporcionar ejemplos de su aplicación en la práctica. ISO / IEC / IEEE 29119-1: 2013 es informativo y proporciona un punto de partida, contexto y orientación para las otras partes.

**ISO / IEC / IEEE 29119-2: 2013** Fuente: (Standardization, ISO International Organization for Standardization, 2013)

Ingeniería de software y sistemas. Pruebas de software. Parte 2: Procesos de prueba.

#### **ABSTRACTO**

El propósito de la serie ISO / IEC / IEEE 29119 de estándares de prueba de software es definir un conjunto de estándares acordados internacionalmente para las pruebas de software que puede ser utilizado por cualquier organización al realizar cualquier forma de prueba de software. ISO / IEC / IEEE 29119-2: 2013 comprende descripciones de procesos de prueba que definen los procesos de prueba de software a nivel organizacional, nivel de gestión de prueba y niveles de prueba dinámica. Admite pruebas dinámicas, pruebas funcionales y no funcionales, pruebas manuales y automatizadas, y pruebas con y sin script. Los procesos definidos en ISO / IEC / IEEE 29119-2: 2013 se pueden utilizar junto con cualquier modelo de ciclo de vida de desarrollo de software.

Dado que las pruebas son un enfoque clave para la mitigación de riesgos en el desarrollo de software, ISO / IEC / IEEE 29119-2: 2013 sigue un enfoque de pruebas basado en riesgos. Las pruebas basadas en riesgos son un enfoque común de la industria para la elaboración de estrategias y la gestión de pruebas. Las pruebas basadas en riesgos permiten priorizar las pruebas y centrarlas en las características y funciones más importantes.

**ISO / IEC / IEEE 29119-3: 2013** Fuente: (Standardization I. O., 2013)



Ingeniería de software y sistemas. Prueba de software. Parte 3: Documentación de prueba.

#### **ABSTRACTO**

El propósito de la serie ISO / IEC / IEEE 29119 de estándares de prueba de software es definir un conjunto de estándares acordados internacionalmente para las pruebas de software que puede ser utilizado por cualquier organización al realizar cualquier forma de prueba de software.

ISO / IEC / IEEE 29119-3: 2013 incluye plantillas y ejemplos de documentación de prueba. Las plantillas están organizadas en cláusulas que reflejan la estructura general de descripción del proceso de prueba en ISO / IEC / IEEE 29119-2, es decir, por el proceso de prueba en el que se están produciendo. El anexo A contiene esquemas del contenido de cada documento. El anexo B contiene asignaciones de ISO / IEC / IEEE 29119-2. El anexo C contiene una descripción general de los ejemplos. Los anexos D a S contienen ejemplos de la aplicación de las plantillas. El anexo T proporciona asignaciones a los estándares existentes. La bibliografía para ISO / IEC / IEEE 29119-3: 2013 se encuentra al final del documento. ISO / IEC / IEEE 29119-3: 2013 admite pruebas dinámicas, pruebas funcionales y no funcionales, pruebas manuales y automatizadas y pruebas con secuencias de comandos y sin secuencias de comandos. Las plantillas de documentación definidas en ISO / IEC / IEEE 29119-3:

**ISO / IEC / IEEE 29119-4: 2015** Fuente: (Standardization I. O., International Organization for Standardization, 2015)

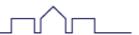
Ingeniería de software y sistemas. Pruebas de software. Parte 4: Técnicas de prueba.

#### **ABSTRACTO**

ISO / IEC / IEEE 29119-4: 2015 define técnicas de diseño de prueba que se pueden utilizar durante el proceso de diseño e implementación de prueba que se define en ISO / IEC / IEEE 29119-2.

ISO / IEC / IEEE 29119-4: 2015 está destinado, entre otros, a probadores, administradores de pruebas y desarrolladores, en particular a los responsables de la gestión e implementación de pruebas de software.

**ISO / IEC 33063: 2015** Fuente: (Standardization I. O., International Organization for Standardization, 2015)



Tecnología de la información - Evaluación de procesos - Modelo de evaluación de procesos para pruebas de software.

#### **ABSTRACTO**

ISO / IEC 33063: 2015 define un modelo de evaluación de procesos que cumple con los requisitos de ISO / IEC 33004 y que respalda el desempeño de una evaluación de la capacidad del proceso utilizando el marco de medición de procesos definido en ISO / IEC 33020. El modelo de evaluación de procesos proporciona indicadores para orientación sobre la interpretación de los propósitos y resultados del proceso como se definen en ISO / IEC / IEEE 29119-2 y los atributos del proceso como se definen en ISO / IEC 33020, y proporciona orientación, por ejemplo, sobre la definición, selección y uso de indicadores de evaluación.

Un modelo de evaluación de procesos comprende un conjunto de indicadores del desempeño y la capacidad del proceso. Los indicadores se utilizan como base para recopilar la evidencia objetiva que permite a un evaluador asignar calificaciones, siguiendo los requisitos de ISO / IEC 33002. El conjunto de indicadores incluidos en esta Norma Internacional no pretende ser un conjunto completo ni es pretendía ser aplicable en su totalidad. Deben seleccionarse subconjuntos que sean apropiados para el contexto y el alcance de la evaluación.

El modelo de evaluación de procesos en esta Norma Internacional está dirigido a patrocinadores de evaluación y evaluadores competentes que deseen seleccionar un modelo y método de proceso documentado asociado para la evaluación (para la determinación de la capacidad o la mejora del proceso).

Se puede utilizar para la evaluación cualquier modelo de evaluación de procesos para pruebas de software que cumpla con los requisitos definidos en ISO / IEC 33004 con respecto a los modelos para evaluación de procesos. Es posible que se necesiten diferentes modelos y métodos para abordar las diferentes necesidades comerciales y de prueba. Este modelo de evaluación se proporciona como un ejemplo de un modelo que cumple con todos los requisitos expresados en ISO / IEC 33004.



# 2. Relación de función de plan de pruebas con proceso de desarrollo de software, considerando análisis de planes de pruebas de proyectos en el mercado.

#### 2.1.- Análisis de planes de pruebas

La relación del plan de pruebas con el proceso de desarrollo de software nos hace ver que podemos encontrar distintos tipos de plan de pruebas, ya que pueden ser diferentes según la etapa de prueba. Si estamos en el comienzo del proyecto podemos crear un plan maestro que va a considerar toda la ejecución del proyecto en el que estamos trabajando. Adicionalmente a esto, podemos crear planes de pruebas de manera separada considerando tipos de pruebas de manera específica, como pueden ser las pruebas de integración, las pruebas de sistema. O, si estamos cercanos al final pruebas de aceptación o pruebas de usuario.

Está la posibilidad de crear también, un plan para las pruebas funcionales y de forma separada uno para las pruebas no funcionales. Las pruebas de rendimiento contarán con un plan de pruebas independiente.

Como vimos anteriormente un plan de pruebas estándar es el que define IEEE-829

A continuación, revisaremos los objetivos de algunos tipos de planes de prueba disponibles, considerando distintos enfoques.

Si lo que necesitamos es un plan de pruebas funcionales, podemos basarnos en la estructura que plantea (ANDALUCÍA, s.f.), en el que señalan con respecto al modelo propuesto:

El objetivo de este documento es recoger los casos de pruebas que verifican que el sistema satisface los requisitos especificados. Deberá contener la definición de los casos de prueba, la matriz de trazabilidad entre casos de pruebas y requisitos, y la estrategia a seguir en la ejecución de las pruebas.



Con respecto a las pruebas de integración la estructura que plantea y que nos entrega una referencia de los contenidos de los planes de prueba disponibles en el mercado en este contexto (ANDALUCÍA, JUNTA DE ANDALUCÍA, s.f.) Considera:

El objetivo de este documento es recoger la definición de las pruebas que se deberán realizar para asegurar la correcta integración entre todos los componentes/módulos del sistema.

Si lo que se necesita es crear un plan de prueba considerando pruebas unitarias (ANDALUCÍA, JUNTA DE ANDALUCÍA, s.f.) Propone:

El objetivo de este documento es recoger la definición del conjunto de pruebas que aseguren el correcto funcionamiento de los componentes de código de forma aislada.

Para lo que son las pruebas de aceptación o pruebas de usuario (ANDALUCÍA, JUNTA DE ANDALUCÍA, s.f.) Indica:

El objetivo de este documento es definir el conjunto de pruebas que deberán ser ejecutadas por los usuarios del sistema para validar si el sistema cumple con los requisitos de funcionamiento esperado y proceder así a la aceptación del sistema.

# 3. Diseño de plan de pruebas de sistema para proyecto de software, según requerimientos del cliente.

## 3.1.-Diseño de planes de prueba

Para realizar el diseño del plan de pruebas deberemos considerar al menos los siguientes puntos que se detallan a continuación:

1. Realizar en base a los requerimientos un análisis para poder comprender ya sea el proyecto o una parte de él, entendiendo que serán sometidos a verificación de su calidad. En este punto debemos saber entre otras cosas, por ejemplo, si es un sistema nuevo o no, establecer cuáles son las funcionalidades presentes que se están modificando, determinar los requerimientos no funcionales.



- 2. Reconocer cuales son todas aquellas funcionalidades nuevas que se probarán, con la finalidad de incluir todas aquellas características en el plan de pruebas.
- 3. Reconocer cuales son todas aquellas funcionalidades que deben ser probadas dentro de los sistemas existentes, en este caso podemos dividirlo en aquellas funcionalidades que se modifican y están ante el usuario (como agregar formularios), o modificar funcionalidades en sus componentes de forma interna, en este caso el usuario no aprecia los cambios de manera directa.
- 4. Es necesario definir cuál será la estrategia de prueba. Es decir, que tipos de prueba se van a realizar.
- 5. Se deben establecer cuáles serán los criterios para el inicio o reanudación, criterios para la suspensión de las pruebas y criterios para aceptación o rechazo considerando cual será el porcentaje de tolerancia en cuanto a los casos de prueba sin imprevistos.
- 6. Reconocer cuáles serán los entornos de prueba necesarios, tanto de hardware como de software que se van a necesitar para poder ejecutar las pruebas que se han planificado. En este punto lo que se recomienda es que esa configuración sea lo más parecida a la que se tendrá en el ambiente de producción del sistema.
- 7. Se establecen las necesidades en cuanto a personal y entrenamiento.
- 8. Se determinan la metodología a utilizar y los procedimientos de prueba.
- 9. Elaboración de la planificación de las pruebas que se van a ejecutar.
- 10. Finalmente se identifican cuáles son los riesgos y se definen los planes de respuesta ante la ocurrencia de ellos para cada caso específico.

#### 3.2.-Requerimientos de clientes

Sabemos que todo proyecto de desarrollo debe comenzar a partir de la captura de requerimientos de los clientes, existen diversos tipos de requerimientos, los que deben ser implementados en el sistema.

Con respecto a las pruebas, encontramos aquellas que están basadas en requerimientos se indica en (SOMMERVILLE, 2005) que:

 Un principio general de ingeniería de requerimientos es que estos deberían poder probarse. Es decir, los requerimientos deberían ser escritos de tal forma que se pueda diseñar una



prueba para que un observador pueda comprobar que los requerimientos se satisfacen. Las pruebas basadas en requerimientos, por lo tanto, son una aproximación sistemática al diseño de casos de prueba en donde el usuario considera cada requerimiento y deriva un conjunto de pruebas para cada uno de ellos. Las pruebas basadas en requerimientos son pruebas de validación en lugar de pruebas de defectos --el usuario intenta demostrar que el sistema ha implementado sus requerimientos de forma adecuada.

Pruebas basadas en requerimientos, en donde los casos de prueba se diseñan para probar los requerimientos del sistema. Esta aproximación se utiliza principalmente en la etapa de pruebas del sistema. Ya que los requerimientos del sistema normalmente se implementan por varios componentes. Para cada requerimiento, se identifica casos de prueba que puedan demostrar que el sistema satisface ese requerimiento.

Se puede ver a partir de esto que las pruebas de un requerimiento no significan escribir sólo una única prueba. Normalmente tienen que escribirse varias pruebas para asegurar que cubre por completo el requerimiento.

# 4. Implementación de plan de pruebas para proyecto de desarrollo de software, según estándares de la industria.

# 4.1.- -Implementación de planes de pruebas

Según lo señalado por (Montecinos, 2020), una vez definidos la estructura y el diseño del plan de pruebas, se procede a la implementación, para lo cual se deberán tener presente algunas reglas y/o recomendaciones, como se detalla en el siguiente recuadro:



#	Regla	Descripción y recomendaciones	Ejemplos
1	Utilizar lenguaje	1 Los casos de prueba deben estar escritos en	
	activo	voz activa y en tiempo presente.	abre la ventana "Nombre de Ventana"
		<ul><li>2 Decirle al tester qué es lo que tiene que hacer.</li><li>3 No usar voz pasiva.</li></ul>	Evitar oraciones como: "La ventana es abierta del lado izquierdo de la aplicación"
2	Identificar Acciones v	1 El tester hace una acción.	Descripción del paso
	Resultados	<ul><li>2 La acción debe iniciar con un verbo, seguida del objeto sobre el cual se realiza la acción.</li><li>3 La aplicación genera un resultado.</li></ul>	
		4 El resultado esperado debe iniciar con: "La aplicación" muestra/despliega etc	- "El Sistema" despliega la pantalla "Administrar Poliza"
		5 No usar voz pasiva en acciones ni en	Descripción del paso
		resultados esperados.	- Presionar la tecla [Enter]
			Resultado esperado - La aplicación muestra la sección
			"Sección X"
3	Manejo de Objetos	1 Las acciones sobre objetos dinámicos de la aplicación deben ser referenciadas como "Ir a".	- Ir al menú "Facturación > Ordenes > Ordenes Abiertas"
		2 Los objetos de la aplicación deben ser descritos exactamente como son percibidos por el tester, usando comillas, mayúsculas y minúsculas.	- Ir a la sección "Ordenes de Pago" Ingresar cualquier valor en el campo "Cliente"
		3 Cuando un objeto este anidado sobre otro hay que utilizar ">"	Presionar las teclas [Ctrl + X] Clic en el botón "Guardar" Ingresar el valor "1001" en el campo
		4 Al mencionar una tecla o combinación de teclas se debe indicar entre "[]", en caso que sea combinación se debe indicar como "[tecla + tecla]"	"Nombre del campo"
		5 Acciones sobre objetos estáticos de la aplicación (botones, mensajes, etc.) deben de iniciar con "Clic"	
		6 Valores de constantes deben estar entre comillas.	



<u> </u>			I
4	Diseño de alto nivel	1 Evitar pasos detallados adicionales, a menos	Utilizar sentencias como:
		que el caso de prueba esté probando un objeto	
		en particular (pantalla, botón, menú, etc.).	Ejecutar "Word" desde la barra de
			Tareas
		2 Identificar lo que hace la aplicación, no lo que	
		hace el objeto.	Ejecutar "Word" desde el menú inicio
			Figure "Mard" doods to correcte do
			Ejecutar "Word" desde la carpeta de instalación
			Instalacion
			Evitar oraciones como:
			"Ir al escritorio y buscar el icono que
			ejecuta la aplicación, dar doble clic sobre el icono para ejecutar la
			aplicación"
			aplicacion
5	Evitar descripciones	1 Evitar mencionar características que no se	Si el caso de prueba está probando una
)	irrelevantes	relacionan con el objetivo del caso de prueba.	pantalla, evita describir lo que este
	irrelevantes	relacionan con el objetivo del caso de prueba.	alrededor de ella.
6		1 Los casos de prueba deben tener bien	
		definida la dependencia con otro caso de prueba	completo la funcionalidad a probar.
	funcionalidad	a ser ejecutado.	
			Generar una factura.
			Ir a la pantalla "Envío de Factura".
7	Dependencias de los	1 Las dependencias con otros casos de prueba	Este caso de prueba depende del caso
	casos de prueba	deben de ser claramente definidas en el campo	"Generar una factura".
		de precondiciones del caso de prueba.	
8	Clonación	1 Antes de crear un caso de prueba, verificar si	Un buen caso de prueba para clonar es
		no existe otro del cual se pueda clonar.	aquel que sigue el paso a paso de
			manera muy similar.
			Si las variables pueden ser fácilmente
			sustituidas y el flujo solo un poco
			modificado, entonces es un buen
			candidato a ser clonado.



	B 11.1		I
9	Precondiciones	·	Este caso de prueba depende del caso
		configuraciones/condiciones que son requeridas	
		para ejecutar un caso de prueba.	Es necesario tener la parametrización
			"XYZ" para este caso de prueba.
		2 El caso de prueba debe de tener claramente	
		identificado y enumerado todos los casos de	
		prueba dependientes, necesidades de ambiente y	
		datos externos requeridos, así como cualquier	
		otra precondición.	
		3 Las precondiciones deben estar enumeradas.	
10	Datos	1 En caso de que sea requerido, el caso de	Matriz de Datos de Prueba
		prueba debe especificar las variables a utilizar y el	
		origen de las mismas.	
		2 La matriz de datos de prueba debe contener	
		todos los datos relevantes a ser utilizados en las	
		pruebas.	
11	Maneio de datos en	1 Cuando tengamos que llenar información de	Descripción del paso
''	caso de prueba	varios campos, utilizar la frase: "Llenar los	
	caso de pideba	siguientes campos con los valores indicados:"	valores indicados:
		signification con los valores indicados.	valores mulcados.
		2 Los campos y sus valores (en caso de que	"Nombre Completo" con el valor
		sean definidos previamente debido a las	
		precondiciones del caso de prueba) deben estar	
		listados.	"Dirección" con el valor <direccion></direccion>
		3 Los nombres de campos deberán estar entre	"Teléfono" con el valor <telefono></telefono>
		"Comillas".	
			Clic en el botón "Guardar"
		4 Los valores de los campos que apliquen	
		deberán especificarse entre ángulos:	Resultado esperado
		<dato variable=""></dato>	
			El sistema muestra un mensaje con un
		5 Si algún dato en específico será requerido	número de seguimiento.
		posteriormente en el caso de prueba, utilizar la	
		frase: "Anotar el valor <dato> para posterior</dato>	Anotar el valor del <numero de<="" td=""></numero>
		verificación"	SEGUIMIENTO> para posterior
			verificación.
1			



12	Puntos de verificación	1 Un punto de verificación asegura que un caso de prueba cumple con el objetivo para el cual fue	
		creado.  2. Cada caso de prueba debe tener como mínimo un punto de verificación.  3 La palabra "Verificar" estará reservada para los puntos de verificación, y no podrá ser utilizada para otro fin dentro de los casos de prueba.  4 No se puede utilizar la palabra "Verificar" en la descripción del paso del caso de prueba, únicamente se utilizara en los resultados esperados para medir los puntos de verificación.	Resultado esperado Verificar que "La Aplicación" despliegue el siguiente mensaje:
13	Descripción	1 El caso de prueba debe tener una descripción.	Nombre del caso de prueba
		2 La descripción debe dar información adicional para los objetivos del caso de prueba.	Descripción del caso de prueba
		3 Se puede agregar información sobre el ambiente, el caso de prueba o su impacto.	
		4 No es aceptable tener el nombre del caso de prueba como descripción.	
14	Pasos enumerados	1 Todos los pasos del caso de prueba deben estar enumerados en orden secuencial. Si en algún momento es necesario agregar pasos a los casos de prueba se deben renombrar los números de pasos con 1.1, 1.2, 1.3 etc., esto con la finalidad de evitar tener que reenumerar todos los pasos del caso de prueba.	enumerados
15	Resultado Esperado	1 Debe existir un "Resultado Esperado" para cada descripción de paso.	El resultado esperado debe expresar como la aplicación reacciona al estímulo descrito en la descripción del paso.
		2 Los Resultados Esperados deben escribirse empezando con la frase: "La Aplicación muestra/confirma/imprime/etc.", seguido de la salida esperada y su criterio de aceptación.	
		3 No usar voz pasiva.	
		4 No usar la palabra "Debe/Debería"	
16	Complejidad del caso de prueba	1 Los casos de prueba deben incluir el tamaño, el número de datos ingresados y los puntos de verificación, esto con el fin de obtener el grado de complejidad y realizar estimaciones.	

 Tabla 1: Implementación de plan de pruebas
 Fuente: (Montecinos, 2020)



## **Conclusiones**

En todo desarrollo de software, ya sea que estemos trabajando en un proyecto completo o una parte él, es necesario considerar la incorporación de las pruebas. Desarrollando un plan de pruebas, nos permitirá establecer si se cumple con lo que el cliente requiere y de acuerdo con la calidad que se ha especificado inicialmente.

El plan de pruebas de software se puede aplicar a todo proyecto de software, considerando las necesidades específicas de cada caso. Así como cada empresa puede aplicar un plan que se ajuste a sus necesidades de mejora continua.



# Referencias bibliográficas

- 1. Mera Paz, J. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería Solidaria*, 12 (20), 163-176.
- 2. Pressman, RS y P. (2010). INGENIERÍA DEL SOFTWARE (SADEC V, Ed.).
- 3. Rodríguez, FT (2014). INTRODUCCIÓN A LAS PRUEBAS DE SISTEMAS DE INFORMACIÓN. Montevideo, Uruguay: Abstracta.