



TECNOLOGO EN ANÁLISIS Y DESARROLLO DE SOFTWARE

EVIDENCIA GA9-220501096- AA1-EV02

PLAN DE PRUEBAS DE SOFTWARE

APRENDICES:

JULIO CESAR HERNANDEZ

DIEGO DAVID MURILLO

DIANA LIZETH MAHECHA

CINDY GONZALEZ

FICHA: 2627090

INSTRUCTOR: CARLOS ALBERTO FUEL

CENTRO DE COMERCIO Y TURISMO -REGIONAL QUINDIO

MAYO 2024

SENA



Contenido

INTRODUCCION	3
OBJETIVOS DE LAS PRUEBAS	3
LAS PRUEBAS DE SOFTWARE Y SUS CARACTERISTICAS.....	3
MERA (2016), LISTA LOS SIETE PRINCIPIOS:.....	4
TIPOS DE PRUEBAS	5
FASES EN LAS PRUEBAS DEL SOFTWARE.....	6
LOS CASOS DE PRUEBA.....	7
ELEMENTOS DE UN CASO DE PRUEBA	7
VENTAJAS Y DESVENTAJAS DE LOS TIPOS DE PRUEBAS	8
RELACION DE PRUBA.....	11
MÓDULOS DE REQUERIMIENTOS	11
MODULO DE DISPOSITIVOS.....	12
ENTORNO Y CONFIGURACION DE LAS PRUEBAS	12
CRITERIOS DE INICIO	13
BASES DE DATOS DE PRUEBAS	13
CRITERIOS DE RECHAZO	14
REGISTRO DE LOS RESULTADOS A LAS PRUEBAS	15
PRUEBAS ALFA.....	22
ANALISIS DE LOS RESULTADOS DE LAS PRUEBAS.....	24
FORMATO DEL INSTRUMENTO DE VALIDACION DE USABILIDAD	25
CONCLUSION	28
BIBLIOGRAFIA.....	29



INTRODUCCION

La prueba de software es el proceso de evaluar y verificar que un producto o aplicación de software hace lo que se supone que debe hacer. Los beneficios de las pruebas incluyen la prevención de errores, la reducción de los costos de desarrollo y la mejora del rendimiento.

Un plan de gestión de pruebas ayuda a priorizar qué tipos de pruebas proporcionan el mayor valor, dado el tiempo y los recursos disponibles. La efectividad de las pruebas se optimiza ejecutando la menor cantidad de pruebas para encontrar la mayor cantidad de defectos.

OBJETIVOS DE LAS PRUEBAS

En el siguiente documento vamos a tomar un software de prueba donde realizaremos revisión de los requerimientos, realización de análisis documentales, identificación de defectos, pruebas funcionales y no funcionales y un plan de mejora continua.

LAS PRUEBAS DE SOFTWARE Y SUS CARACTERISTICAS

En los últimos años se han propuesto algunos principios con el fin de establecer pautas universales para que los programadores de software las adapten a sus procesos de pruebas.



NERA (2016), LISTA LOS SIETE PRINCIPIOS:

- **Las pruebas muestran la presencia de defectos:** Las pruebas son herramientas que facilitan detectar defectos; no obstante, no garantizan que no haya defectos ocultos y no es una evidencia de que el software esté óptimo.
- **Pruebas tempranas:** Encontrar los errores en etapas tempranas, permite un importante ahorro de recursos.
- **No es posible realizar pruebas exhaustivas:** Hay que realizar un análisis de riesgos, para identificar prioridades y tener una óptima toma de decisiones, centralizando esfuerzos y utilizando el talento humano y recursos de manera correcta.
- **Las pruebas dependen del contexto:** Estas dependen del contexto en el cual se ejecutan; por lo cual se debe dar más énfasis en las que sean para un sistema crítico como para un software financiero, para el cual se requiere realizar más pruebas, en comparación con otras aplicaciones con nivel críticos bajos.
- **Agrupación de defectos:** Las pruebas deben agruparse, la mayoría de fallos operativos se enfocan en un número reducido de módulos.
- **La paradoja de los pesticidas:** No hay que repetir la misma prueba varias veces, por lo cual es importante que los casos de prueba se revisen constantemente y escribir nuevos con el fin de encontrar más defectos.



- **Ausencia de error:** De nada sirve corregir en el sistema si este no cumple con los requerimientos o necesidades del usuario.

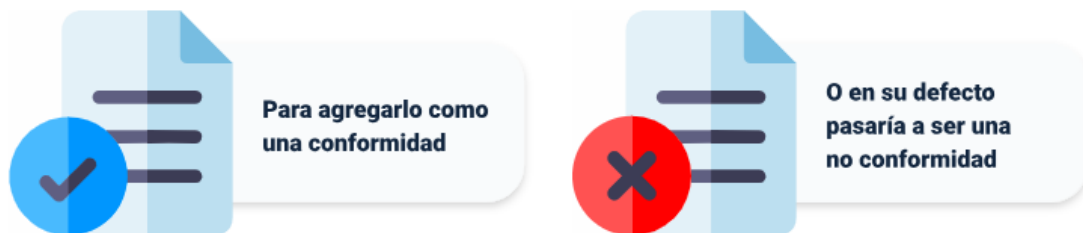
TIPOS DE PRUEBAS

- **Prueba de aceptación:** verifica si todo el sistema funciona según lo previsto.
- **Pruebas de integración:** asegura que los componentes o funciones del software operen juntos.
- **Pruebas de unidad:** valida que cada unidad de software funcione según lo esperado. Una unidad es el componente de prueba más pequeño de una aplicación.
- **Pruebas funcionales:** verifica funciones mediante la emulación de escenarios de negocio, en función de los requisitos funcionales. La prueba de caja negra es una forma común de verificar funciones.
- **Pruebas de rendimiento:** prueba cómo funciona el software bajo diferentes cargas de trabajo. Las pruebas de carga, por ejemplo, se utilizan para evaluar el rendimiento en condiciones de carga reales.
- **Pruebas de regresión:** verifica si las nuevas características rompen o degradan la funcionalidad. Las pruebas de cordura se pueden utilizar para verificar menús, funciones y comandos a nivel superficial, cuando no hay tiempo para una prueba de regresión completa.
- **Pruebas de estrés:** prueba cuánta tensión puede soportar el sistema antes de que falle. Considerado como un tipo de prueba no funcional.



- **Pruebas de usabilidad:** valida qué tan bien un cliente puede usar un sistema o una aplicación web para completar una tarea.

Es de suma importancia realizar una bitácora y una matriz con el fin de ejecutar los casos de prueba y así concluir si el caso funciona como se esperaba:



La matriz ayuda a establecer métricas que ayudan a definir tiempos, esfuerzos, asignaciones, y riesgos logrando optimizar recursos y talento humano.

FASES EN LAS PRUEBAS DEL SOFTWARE

- El Control
- Comprobación De Resultados
- Elaboración De Informes De Proceso
- Aplicación Objetivo De Las Pruebas
- Bitácoras De Experiencias



Según la organización ISO (2020):

Estas pruebas se regulan por la norma ISO/IEC 25000 Sistemas y Requisitos de Calidad y Evaluación de *software* (SQuaRE).

Pruebas de *software* aplicables a todos aquellos que deseen crear:

- Especificaciones de prueba.
- Marcos correspondientes.
- O crear automatización de pruebas.

LOS CASOS DE PRUEBA

Son los documentos de los productos que se generan al momento de realizar una prueba de software, en el cual se plasma las precondiciones entradas y resultados esperados, incluyendo la implementación correcta, la identificación de errores, el chequeo de la calidad y otras informaciones consideradas importantes, en general no hay una plantilla específica solo la bitácora, listas de chequeo, e imágenes en la cual se registran las no conformidades que se encuentran en las pruebas.

ELEMENTOS DE UN CASO DE PRUEBA

- **Identificador:** Puede ser alfanumérico o numérico.
- **Nombre:** Nombre del caso de prueba de manera concisa.
- **Descripción:** Objetivo del caso de prueba, también describe qué probará, en ciertas ocasiones se incluye el ambiente de pruebas.



- **Número de orden Ejecución:** Orden en la cual se ejecuta el caso de prueba, en la situación de que se tengan múltiples casos de prueba.
- **Requerimiento asociado:** Si se plantea un caso de prueba se debe saber a qué requerimiento va asociado para mantener la trazabilidad.
- **Precondición:** Estado en la cual se debe encontrar el sistema antes de comenzar la prueba.
- **Postcondición:** El estado en que debe encontrarse el sistema luego de ejecutar la prueba.
- **Resultado esperado:** Objetivo que debe ser alcanzado posterior a ejecutarse la prueba.

VENTAJAS Y DESVENTAJAS DE LOS TIPOS DE PRUEBAS

Todo proceso tiene sus ventajas y desventajas, a continuación, se presentan cada una de ellas.

Pruebas funcionales

- Optimiza las funcionalidades y la calidad del producto.
- Verifica que la seguridad y la arquitectura sean las adecuadas.
- Asegura que el comportamiento del software y de todas sus funciones sea el esperado. Se asegura que el software o aplicación está libre de errores.



Pruebas unitarias

- Simplifica la integración y permite una buena documentación.
- Funciona con el proceso de desarrollo ágil.
- Detección temprana de errores en sus funcionalidades lo que minimiza tiempo y costos. Optimiza la efectividad del código.

Pruebas de integración

- Detalla los errores y conflictos en cada módulo y en sus conexiones.
- Mejora la calidad del código.
- Garantiza que las funciones del software serán confiables y estables entre sus módulos.
- Garantiza que todos los módulos del software estén correctamente integrados y funcionen en conjunto como se necesita.

Pruebas de Rendimiento

- Mejora las características del software o aplicación perfeccionando el sistema.
- Identifica errores que no se detectan en las pruebas funcionales.
- Localiza los cuellos de botella para solucionarlos y optimizar su rendimiento.
- Valora la escalabilidad y velocidad de la aplicación o website.



Desventajas de estos tipos de pruebas:

Por lo general, todas las pruebas de software traen beneficios al desarrollador, dependiendo del tipo de la prueba los beneficios pueden ser más altos y enfocados en un fin o propósito específico, dando como resultado una mejor calidad de producto, mejorando su funcionalidad, reduciendo costos, tiempos, recursos humanos y mejorando la experiencia del usuario. Se podría considerar como una desventaja, si se comparan los tipos de prueba ya que puede variar en:

- Tiempo, Costo, Esfuerzo, Recursos humanos, Exactitud de las pruebas
- Sin embargo, la única desventaja de las pruebas de software son no hacerlas ya que el omitir este paso desencadena una nefasta serie de errores como un producto no funcional y colocarían en tela de juicio la profesionalidad de programador y su equipo de trabajo.



RELACION DE PRUBA

Nombre	Descripción	Fecha de Aprobación
Informe técnico	Metodología de desarrollo y análisis de prueba	
Documento	Informe de requisitos	

MÓDULOS DE REQUERIMIENTOS

Componentes	<ul style="list-style-type: none">• Registro de usuario• Tipo de Evento• consultar• Modificar• Eliminar• Buscar
Objetivos	<ul style="list-style-type: none">• Verificar conectividad• Usabilidad• seguridad• Realización de Procesos
Responsable	<ul style="list-style-type: none">• Equipo de proyecto en conjunto con el programa de formación



MODULO DE DISPOSITIVOS

Componentes	
<ul style="list-style-type: none">• Monitoreo de servicios y Dispositivos	
Objetivos	<ul style="list-style-type: none">• Verificar conectividad de los servicios y aplicaciones• Dar una respuesta rápida al usuario de cualquier fallo• Una interfaz sencilla de comprender del usuario• Información detallada de dispositivo y aplicación
Responsable	<ul style="list-style-type: none">• Equipo de proyecto en conjunto con el programa de formación

ENTORNO Y CONFIGURACION DE LAS PRUEBAS

Para el proceso de pruebas del proyecto se requiere de la disponibilidad de los siguientes entornos a saber:

Hardware:

- Memoria RAM: 16 GB o 32 GB de RAM.
- Procesador: Gama alta, como un Intel Core i7 o i9, o un procesador AMD Ryzen.



- SSD: Con al menos 512 GB, para almacenar archivos de audio y video.
- Tarjeta gráfica: se recomienda una tarjeta gráfica dedicada de gama media o alta para manejar aplicaciones de video en tiempo real.

Software:

- Licencia: Propietario.
- Sistema Operativo: Windows 10 pro.
- Base de datos: MYSQL.
- Servidor: Hosting calidad certificada ISO 9001.

CRITERIOS DE INICIO

- Aceptación del plan de pruebas: Revisión y aceptación del documento que contiene los casos de pruebas para la certificación del proyecto.
- Aceptación de paquetes: Revisión y aceptación de los softwares de desarrollo y que estos cumplan con las condiciones.
- Aceptación de ambiente: Revisión y aceptación del ambiente de certificación y que estos cumplan con las condiciones de aceptación.

BASES DE DATOS DE PRUEBAS

- Bases de datos: Data base
- Servidor: MYSQL
- Datos: Aleatorios



CRITERIOS DE RECHAZO

- Errores graves: información presentada erróneamente, información mal registrada en la base de datos, caídas de programa, incumplimiento de objetivos en funciones principales, entre otros.
- Errores medios: Error en presentación de datos, incumplimiento de objetivos en funciones secundarias, caídas de programas auxiliares.
- Errores leves: Errores en presentación de datos secundarios, no adecuación a estándares, comportamientos correctos pero diferentes, en situaciones similares de operación entre otros.

Criterio	Descripción
APROBACION	Se aprobará con el 100% de la prueba ejecutada, pero con un 90% de aceptación. Esto quiere decir que pueden existir errores medios o bajos equivalentes al 10%.
RECHAZO	En caso de ocurrir que el proyecto no cumpla con el nivel exigido, el proyecto se rechaza completo en su etapa de certificación.



REGISTRO DE LOS RESULTADOS A LAS PRUEBAS

PRUEBAS DE COMPONENTES

Modulo N°1 Registro de Usuario, Tipo de Evento.

Objetivos de la Prueba		Comprobar servicios y dispositivos
Técnicas		Dispositivos de prueba y datos falsos
Casos de Prueba		Tipo de prueba Unitaria (Caja Blanca)
		Objetivo: Validar si el registro se guarda correctamente
		Caso N°1: Registro de clientes
		Entradas: Nombre: "Gabriel Torres" Tipo de evento: "Cumpleaños" Fecha: "10-agosto-2024"
		Salida Esperada: Registro Exitoso.
Resultados		Todos los casos de prueba se han ejecutado
Observaciones		

Formato de casos de Prueba	
Tipo de prueba: Unitaria caja blanca	
Objetivo: Verificar si el formulario registra sus respectivas validaciones en los campos obligatorios y no registra.	
Caso N°2	
Descripción: Datos incorrectos, se dejarán las etiquetas en blanco	



Entradas: Nombre: “”, Tipo de evento: “”, Fecha: “”.

Salida esperada: Por favor, rellene este campo.

Modulo N°2 Consultar.

Objetivos de la Prueba		Comprobar servicios y dispositivos
Técnicas		Dispositivos de prueba y datos repetidos
Casos de Prueba		Tipo de prueba Unitaria (Caja Blanca)
		Objetivo: Validar si el registro ya esta guardado
		Caso N°1: Consultar Registro
		Entradas: Nombre:” Gabriel Torres” Tipo de evento: “Cumpleaños” Fecha: “10-agosto-2024”
		Salida Esperada: Registro Existente.
Resultados		Todos los casos de prueba se han ejecutado
Observaciones		

Formato de casos de Prueba	
Tipo de prueba: Unitaria caja blanca	
Objetivo: Verificar si el formulario registra sus respectivas validaciones en los campos obligatorios.	
Caso N°2	
Descripción: Datos Existentes, se pondrá un registro repetido	
Entradas: Nombre: “”, Tipo de evento: “”, Fecha: “”.	
Salida esperada: Registro Existente	



Modulo N°3 Modificar, Eliminar, Buscar.

Objetivos de la Prueba		Comprobar servicios y dispositivos
Técnicas		Dispositivos de prueba, Modificar, Eliminar, Buscar
Casos de Prueba		Tipo de prueba Unitaria (Caja Blanca)
		Objetivo: Navegar por las opciones de modificar, eliminar, buscar y verificar si cada una de ellas cumple con su función.
		Caso N°1: Seleccionar la opción sobre un registro existente. Entradas: Tipo de evento: “Matrimonio” Fecha: “20-Julio-2024”
		Salida Esperada: Registro Modificado
Resultados		Todos los casos de prueba se han ejecutado
Observaciones		

Formato de casos de Prueba	
Tipo de prueba: Unitaria caja blanca	
Objetivo: Verificar si el formulario registra sus respectivas validaciones en los campos obligatorios.	
Caso N°2	
Descripción: Datos Existentes, colocar datos validados.	
Entradas: Tipo de evento: “”, Fecha: “”.	
Salida esperada: Mostrar el registro con sus respectivas opciones de modificar y eliminar.	



PRUEBAS FUNCIONALES

Modulo N°1 Pruebas de funcionalidad Caso de uso Registro y Tipo de evento.

Objetivos de la Prueba	Comprobar el funcionamiento del registro de datos en los formularios de dispositivos y servicios de manera eficiente.
Técnicas	Evaluar la funcionalidad, introduciendo datos validos he inválidos. Cuando se ingresan datos correctos, se obtienen resultados deseados. Cuando se utilizan datos incorrectos se obtienen mensajes de error o advertencias.
Caso de uso involucrado	Caso de uso Registro y Tipo de evento

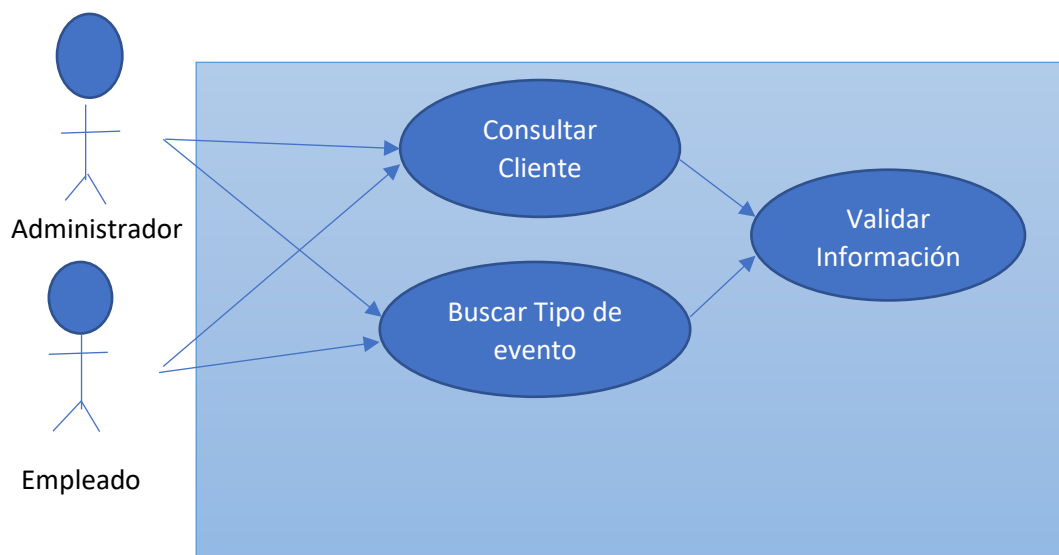


Resultados:	Todas las pruebas planificadas se han ejecutado Todos los defectos identificados se han considerado
-------------	--



Modulo N°2 Pruebas de funcionalidad Caso de uso Consultar.

Objetivos de la Prueba	Comprobar el funcionamiento de la opción consultar y que cumpla su funcionamiento
Técnicas	Evaluar la funcionalidad, introduciendo datos validos he inválidos. Cuando se ingresan datos correctos, se obtienen resultados deseados. Cuando se utilizan datos incorrectos se obtienen mensajes de error o advertencias.
Caso de uso involucrado	Caso de uso Consultar cliente y Tipo de servicio

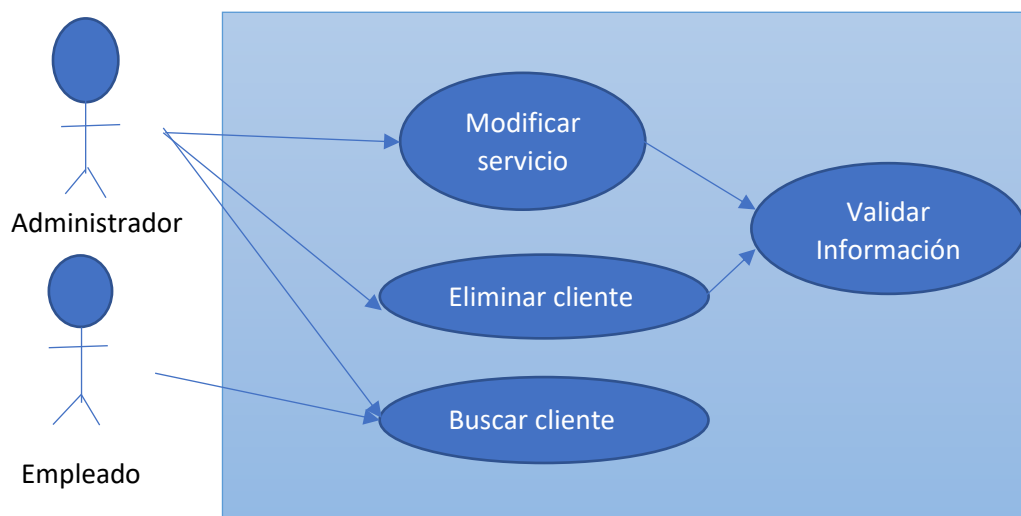


Resultados:	Todas las pruebas planificadas se han ejecutado Todos los defectos identificados se han considerado
--------------------	--



Modulo N°3 Pruebas de funcionalidad Caso de uso Modificar, Eliminar, Buscar.

Objetivos de la Prueba	Comprobar el funcionamiento de las opciones Modificar, Eliminar, Buscar y que cumpla su funcionamiento
Técnicas	Cuando se selecciones Modificar el servicio deseado, cambia automáticamente al guardar los cambios. Al eliminar un registro, se debe borrar del sistema permitentemente. Colocando el nombre del cliente debe mostrar los resultados esperados.
Caso de uso involucrado	Caso de uso modificar, eliminar, buscar.

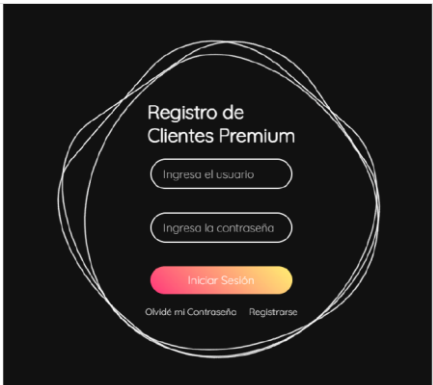


Resultados:	Todas las pruebas planificadas se han ejecutado Todos los defectos identificados se han considerado
--------------------	--



PRUEBAS ALFA

Modulo N°1 Pruebas de funcionalidad Caso de uso Registro y Tipo de evento.

Objetivos de la Prueba	Ingresar los datos correctos al formulario
Técnicas	Cuando se ingresan los datos, automáticamente ingresa al sistema Cuando se ingresan los datos incorrectos muestra un mensaje de error
Interfaz Asociada	
Resultados	

Modulo N°2 Pruebas de funcionalidad Caso de uso Consultar.

Objetivos de la Prueba	Ingresar los datos correctos al formulario
Técnicas	Cuando se ingresan los datos de consulta muestra automáticamente el registro en la BD. Cuando se ingresan los datos incorrectos muestra un mensaje de error Al buscar un dato incorrecto el sistema no arroja ningún resultado



Interfaz Asociada	Formulario Para Agendar Evento <input type="text" value="Nombre"/> <input type="text" value="Apellido"/> <input type="text" value="Teléfono"/> <input type="text" value="Correo"/> <input type="text" value="Tipo Evento"/> <input type="text" value="dd/mm/aaaa"/> <input type="button" value="Agendar"/> Volver a la página principal
	Resultados

Modulo N°3 Pruebas de funcionalidad Caso de uso Modificar, Eliminar, Buscar.

Objetivos de la Prueba

Al Buscar desde el formulario saldrá la opción correspondiente para Modificar o Eliminar

Técnicas

Al presionar modificar y guardar quedara la acción automáticamente

Al presionar eliminar se borrará del sistema

Al elegir volver, el software volverá a la pantalla anterior.

Interfaz Asociada

Registro

Código	Nombre	Apellido	Telefono	Correo	Tipo de Evento	Cantidad de Invitados	Acciones	
1	Jairo	Hernandez	3195669281	yuaib27@gmail.com	cumpleaños	50	Actualizar	Eliminar
2	diego	murillo	3005764784	diego-dj@gmail.com	quince años	45	Actualizar	Eliminar
3	diana	mahecha	3001234567	dianamahe@gmail.com	empresarial	80	Actualizar	Eliminar
4	andres	bernal	3005824652	felipe@gmail.com	cumpleaños	30	Actualizar	Eliminar
5	stephania	hernandez	3154687913	tefa@gmail.com	empresarial	50	Actualizar	Eliminar
6	mario	hernandez	3045381746	mario@gmail.com	cumpleaños	50	Actualizar	Eliminar
7	nicolas	hernandez	3004658625	rico@gmail.com	despedida	50	Actualizar	Eliminar

Agregar

Limpiar



Resultados	

ANALISIS DE LOS RESULTADOS DE LAS PRUEBAS

Tipo de Prueba	Caso de Prueba	Resultados	Criterio de Finalización
Componentes	Registro de clientes Caso N°1 Caso N°2	Esperado	Aprobado
Componentes	Consultar Caso N°1 Caso N°2	Esperado	Aprobado
Componentes	Modificar, Eliminar, Buscar Caso N°1 Caso N°2	Esperado	Aprobado
Funcionales	Todos los casos han sido probados	Esperado	Aprobado
Interfaz de usuario	Los problemas de usabilidad fueron corregidos gracias a la aplicación de las heurísticas de Nielsen	Esperado	Aprobado
Alfa	Todas las funcionalidades del módulo se ajustan a los requerimientos	Esperado	Aprobado



FORMATO DEL INSTRUMENTO DE VALIDACION DE USABILIDAD

Este documento a sido creado con la finalidad de llevar un registro de las opiniones de los usuarios finales acerca de la usabilidad del sistema, es decir, las interfaces del software lo que incluye, el tamaño de la letra, los colores utilizados en las pantallas, la navegabilidad apropiada en cada uno de los módulos, entre otros.

La información proporcionada por los actores de las pruebas será de gran ayuda para mejorar el sistema en cuanto a los problemas de usabilidad.

Nombre	Descripción	Fecha de aplicación



Seleccione la opción que mas se acerque a su opinión:



ASPECTOS A EVALUAR EN LA APLICACIÓN

FECHA DE LA EVALUACION: _____

NOMBRE DEL SISTEMA EVALUADO: CIJOMADI

VERSION DEL SISTEMA: V1.0

ITEM	CUMPLE	NO CUMPLE	NO APLICA	OBSERVACIONES
INTERFAZ AMIGABLE ¿Se cuenta con una interfaz amigable y basada en los requerimientos?	x			
COMPONENTES DE AREA ¿Cada componente tiene un área destinada a la ayuda?	x			
PARAMETROS DE CONSULTA ¿Los resultados se muestran ordenados según la importancia de los parámetros de consulta?		x		
AREA DE REGISTRO ¿Cada resultado tiene un enlace al detalle del registro seleccionado?	x			
BOTONES DE BUSQUEDA ¿Se muestran botones de búsqueda que ayuden al usuario?		x		
AGIL NAVEGACION ¿El sistema posee menús que ayuden con una ágil navegación?	x			



SEGURIDAD DE LA INFORMACION ¿No se genera duplicidad de información?		x		
Observaciones				

FIRMA DEL EVALUADOR: _____

NOMBRE DEL EVALUADOR: _____



CONCLUSION

Todas las pruebas de software traen beneficios al desarrollador, dependiendo del tipo de la prueba los beneficios pueden ser más altos y enfocados en un fin o propósito específico, dando como resultado una mejor calidad de producto, mejorando su funcionalidad, reduciendo costos, tiempos, recursos humanos y mejorando la experiencia del usuario.

Aunque las pruebas en sí cuestan dinero, las empresas pueden ahorrar millones al año en desarrollo y soporte si cuentan con una buena técnica de pruebas y procesos de control de calidad. Las pruebas tempranas de software revelan problemas antes de que un producto salga al mercado. Cuanto antes reciban los equipos de desarrollo los comentarios sobre las pruebas, antes podrán abordar cuestiones como:

- Defectos en la arquitectura
- Decisiones de diseño poco acertadas
- Funcionalidades no válidas o incorrectas
- Vulnerabilidades de seguridad
- Problemas de escalabilidad

Cuando el desarrollo deja un amplio margen para las pruebas, mejora la fiabilidad del software y se entregan aplicaciones de alta calidad con pocos errores. Un sistema que cumpla o incluso supere las expectativas de los clientes conduce a un aumento potencial de las ventas y a una mayor cuota de mercado.



BIBLIOGRAFIA

- Nos apoyamos con la grabación de la explicación con el profesor y una plantilla de ejemplo
- <https://drive.google.com/file/d/1kzYInby-VRUVODalmCVaW2KeOireHEn5/view?usp=sharing>
- Con material de apoyo
- <https://zajuna.sena.edu.co/zajuna/mod/page/view.php?id=275406>
- He información tomada de internet
- <https://www.ibm.com/es-es/topics/software-testing>
- <https://www.ibm.com/es-es/topics/software-testing>