# **TIPOS DE DOCUMENTOS**

## **PREPARACION DE PRUEBAS**

### 

### **PLAN DE PRUEBAS**

De los siguientes elementos se compone el plan de pruebas:

• Definición del alcance

• Definición de los recursos

• Criterios de calidad

• Definición de ítems a probar

• Tipos de pruebas escogidas para validar los criterios

• Riesgos

#### **ALCANCE**

Verificar el cumplimiento de las normas y estándares de calidad parametrizados por los requerimientos solicitados en el desarrollo, con el fin de que sea un aplicativo que cumpla su funcionalidad, se manejara la validación concreta de los estándares de calidad con planes de pruebas aplicados a este desarrollo obteniendo pruebas exactas de tolerancias a fallos y de más criterios manejados en las normas de calidad implementadas.

El documento de desarrollo de pruebas se elabora con el nombre de aplicación de modelo calidad para el módulo de comunicación de traspaso de la empresa San Ambiente, y en él se especifica paso a paso las etapas para que lograr hacer una prueba exacta definiendo el alcance, enfoque y recursos utilizados en la etapa de prueba, ítems a probar, herramientas que se usaran, alcance de cada prueba y los riesgos

#### **DEFINICION DE LOS RECURSOS DE PRUEBA**

Se nombran algunas herramientas para la realización de pruebas

* Herramientas para pruebas unitarias:

Tabla 1: Tipos de herramientas para la elaboración de pruebas

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tabla de tipos de herramientas para elaboración de pruebas | | | | |
| Tipo de prueba | Herramienta | Descripción | Adecuada para este proyecto | Facilidad de uso |
| Pruebas de código | Visual Studio | Identificar e interactuar con objetos gráficos | Si |  |
| Visual Studio | Interfaz web | Si |  |
| Visual Studio | Pruebas unitarias | Si | [Esta foto](https://commons.wikimedia.org/wiki/File:Approve.svg) de Autor desconocido está bajo licencia [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/) |
| JMeter | Prueba de stress | Si |  |
| Análisis de errores de código | Visual Studio Debugging | Recorrido de código y análisis de errores | Si |  |

• PostgreSQL

Es un gestor de base de datos relacional orientado a objetos y libre, que va dirigido a una comunidad de desarrolladores, no tiene gestor de defectos haciendo muy difícil conocer el estado de sus defectos.

• Angular Material

Es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

• Node.js

Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript

* Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft para Windows , Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

#### **CRITERIOS DE CALIDAD**

Para el plan de pruebas de este proyecto se tomarán algunos criterios de calidad según la norma ISO 25010

**Adecuación Funcional:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades cuando se usa en condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Completitud funcional. Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
* Corrección funcional. Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
* Pertinencia funcional. Capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Para la adecuación funcional se realizarán las siguientes pruebas:

* Pruebas Unitarias
* Casos de uso

**Eficiencia:** Asegura que la aplicación logre cumplir los objetivos, pero en un tiempo óptimo y sin retrasos.

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Comportamiento temporal. Los tiempos de respuesta y procesamiento.
* Utilización de recursos. Las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.
* Capacidad. Grado en que los límites máximos de un parámetro de un producto o sistema software cumplen con los requisitos.

Para la eficiencia dentro del software se realizarán las siguientes pruebas:

* Pruebas de carga o consulta masiva
* Tiempos de respuesta del aplicativo

**Usabilidad:** Es la medida de la calidad de la experiencia que tiene un usuario cuando interactúa con un producto o sistema

Capacidad del producto software para ser entendido, aprendido, usado por el usuario bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* Capacidad para reconocer su adecuación. Capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
* Capacidad de aprendizaje. Capacidad del producto que permite al usuario aprender su aplicación.
* Capacidad para ser usado. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
* Protección contra errores de usuario. Capacidad del sistema para proteger a los usuarios de hacer errores.
* Estética de la interfaz de usuario. Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
* Accesibilidad. Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Para la elaboración de pruebas de usabilidad se realizaron formatos para usuarios donde se mostrarán los resultados de la revisión y el refinamiento del trabajo realizado

**Seguridad**: Es la medida de la confidencialidad que tienen los archivos del aplicativo

Capacidad de protección de la información y los datos para que personas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

* **Confidencialidad.** Protección contra el acceso de datos e información no autorizados.
* **Integridad.** Componente para prevenir accesos o modificaciones no autorizados.
* **No repudio.** Demostrar las acciones o eventos que han tenido lugar, y no puedan ser repudiados posteriormente.
* **Responsabilidad.** Rastrear de forma inequívoca las acciones de una entidad.
* **Autenticidad.** Demostrar la identidad de un sujeto o un recurso.

Para la seguridad dentro del software se realizarán pruebas para validar la autenticación de los usuarios al sistema se realizarán las siguientes pruebas:

* Casos de uso
* Pruebas Unitarias

#### **DEFINICION DE ITEMS A PROBAR**

Los ítems para probar se dividen en varios módulos:

• **Interfaz gráfica:** Se tendrá en cuenta cada una de las vistas del aplicativo.

**Pruebas de usabilidad:** Se prueba el ítem mediante encuestas realizadas a diferentes usuarios ajenos al software desarrollado.

**Pruebas Eficiencia:** Se prueba el comportamiento del aplicativo y los tiempos de respuesta a las peticiones realizadas por los usuarios.

• **lógica del aplicativo:**

**Pruebas Unitarias:** Se prueban métodos de clases, por ejemplo: métodos get/set, se realizarán pruebas en las líneas de código validando que el método codificado reciba y entregue valores correctos e incorrectos.

• **base de datos:**

**Pruebas de consultas masivas**: Se realizan pruebas de carga donde se simula la conexión de múltiples usuarios consultando la misma base de datos.

**Pruebas Eficiencia:** Se prueba el comportamiento del aplicativo y los tiempos de respuesta a las peticiones realizadas por los usuarios.

* **Login (Seguridad):** Se probará la validación de los datos, autenticación de los usuarios y encriptación de contraseñas en base de datos, se evitará que los datos personales como contraseñas no se ingresen en texto plano.

#### **TIPOS DE PRUEBAS ESCOGINAS PARA VALIDAR LOS CRITERIOS DE CALIDAD.**

Con las siguientes pruebas se quiere satisfacer los criterios de calidad que se plantearan

• Pruebas Unitarias

• Pruebas de usabilidad

• Pruebas eficiencia

• Pruebas de seguridad

#### **RIESGOS**

Los riesgos que se pueden tener en este plan de pruebas son que al ejecutar las pruebas con las herramientas planteadas anteriormente no se esperen los resultados que se tienen planeados, ya que el volumen de la aplicación puede ser un poco grande para las herramientas, y que puede que no sean óptimas para ejecutarse con este tipo de proyectos o que al ser ejecutadas tenga errores en el código y dificulte el proceso de pruebas.