**TABLA DE CONTENIDO**

#### INTRODUCCIÓN

Este documento muestra el plan de pruebas y los pasos a seguir para la ejecución de pruebas de la aplicación Precise Estimate en cuanto a Front-End y Back-End. El propósito es saber cómo se comporta la aplicación recibiendo un tráfico moderado de usuarios concurrentes (20 – 50 usuarios) dentro de un ambiente publico web.

#### ESQUEMA DE TRABAJO

Para especificar el esquema de trabajo detallamos cada uno de los actores encargados del diseño y la ejecución de las pruebas en las diferentes partes de la aplicación, de la misma manera la especificación de los canales de comunicación entre los encargados de las pruebas y la publicación de los resultados.

#### Actores Encargados

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Cargo** | **Función** |
| Sebastian Aguirre | Desarrollador Backend | Creación datos de prueba |
| Publicación de base de datos con datos de prueba |
| Creación de escenario de pruebas de CRUD de proyecto |
| Creación de escenario de pruebas de CRUD de historia épica |
| Creación de escenario de pruebas de CRUD de historia característica |
| Creación de escenario de pruebas de CRUD de historia historia |
| Creación de escenario de pruebas de CRUD de historia tarea |
| Creación de escenario de pruebas de asignación de tareas a usuario |
| Ejecución de pruebas vinculadas a proyecto |
| Ejecución de pruebas vinculadas a historias (épica, característica, historia y tarea) |
| John Jairo Isaza | Desarrollador Backed | Creación datos de prueba |
| Diseño de pruebas de CRUD de proyecto |
| Diseño de pruebas de CRUD de historia épica |
| Diseño de pruebas de CRUD de historia característica |
| Diseño de pruebas de CRUD de historia historia |
| Diseño de pruebas de CRUD de historia tarea |
| Diseño de pruebas de CRUD de sprint y release del proyecto |
| Creación de escenario de pruebas de CRUD de usuario |
| Ejecución de pruebas de usuario |
| Felipe Cano | Desarrollador Frontend | Creación datos de prueba |
| Revisión de pantallas de proyecto (varios navegadores) |
| Creación de proyecto desde front end (varios navegadores) |
| Revisión de tamaños y estilos para la pantalla de presentación de historias y proyectos (varios navegadores) |
| Asociación de historias a un usuario desde front end (varios navegadores) |
| Ejecución de prueba de click sobre el login de usuario |
| Ejecución de prueba de carga y estrés sobre la información completa de un proyecto |
| Jaime Wilchez | Desarrollador Frontend | Creación datos de prueba |
| Revisión de pantallas de reportes (varios navegadores) |
| Diseño de pruebas de reportes |
| Ejecución de pruebas de generación de reportes |
| Ejecución de reportes desde front end (varios navegadores) |
| Revisión de toda la aplicación en múltiples dispositivos a diferentes resoluciones |
| Diseño de pruebas no funcionales |

## Canales de comunicación

Los canales de comunicación establecidos para la etapa de pruebas son los mismos que se han trabajado a lo largo de todo el diseño y la ejecución del proyecto:

* GitHub como repositorio de información en donde se creará cada una de las pruebas como una tarea asociada al proyecto y dentro de la misma se publicará el resultado de la misma.
* Google Drive como repositorio intermedio de documentos para almacenar los detalles documentales de las pruebas realizadas y donde se especificará el diseño de cada una de las pruebas. El diseño y la ejecución de cada prueba en específico se deberá almacenar en las tareas del repositorio.
* Skype para comunicación con algún integrante del equipo en caso de ser necesario un apoyo adicional.

#### ENTORNO DE PRUEBAS

## Herramientas para pruebas

* 1. **Herramientas de software**

| **Herramienta** | **Descripción** |
| --- | --- |
| POSTMAN | Herramienta para la ejecución de pruebas automáticas programadas en servicios REST. Permite la creación ágil de peticiones y flujos de trabajo al sistema. |
|  |  |

* 1. **Herramientas de hardware**

| **Elemento** | **Descripción** |
| --- | --- |
| Dispositivo Android 4.0 – Superior IOS 7.0 – Superior | Navegador web capaz de procesar código en JavaScript |

## CONFIGURACIÓN DE LOS AMBIENTES DE PRUEBAS

* 1. **Ambientes de pruebas**

Las características requeridas para el ambiente de pruebas son:

* Gestor de paquetes NPM versión 4 o superior
* Node.Js Versión 7.6 o superior
* Framework Angular V2
* Angular Core 2.4 o superior
* MySql Database Engine V 5.6 o superior
  1. **Entorno de usuario**

Las características que debe tener el usuario son

* Google Chrome V 55 o superior
* Sistema operativo Windows 7 o superior.

#### ENFOQUE DE LAS PRUEBAS

## Tipos de pruebas a ejecutar

* 1. **Pruebas estáticas**

|  |  |
| --- | --- |
| **Objetivo** | Verificar la calidad del código fuente mediante una evaluación de aspectos concernientes con la sintaxis, semántica, estilo e implementación de buenas prácticas de programación |
| **Componente a probar** | Código fuente |
| **Enfoque metodológico** | Caja blanca  Existen dos mecanismos para efectuar estas pruebas:   * Inspecciones de código, siguiendo el proceso formal de revisiones de pares * Análisis estático de código, mediante una herramienta de apoyo seleccionada para tal fin   Los parámetros de evaluación vienen dados por los estándares seleccionados para la aplicación y el diseño detallado del componente a revisar. |
| **Herramientas de soporte** | Ninguna |

* 1. **Pruebas dinámicas**

**- Pruebas funcionales**

|  |  |
| --- | --- |
| **Objetivo** | Verificar la satisfacción de los requerimientos funcionales del sistema. |
| **Componente a probar** | Clase, método o paquete.  Forma, listado y reporte del sistema  Algoritmo de cálculo o cómputo de operaciones |
| **Enfoque metodológico** | Caja blanca.  Contempla la ejecución de pruebas unitarias para verificar el correcto funcionamiento de los elementos (clases, métodos, paquetes) que integran un componente de software. Los parámetros de evaluación vienen definidos por el diseño detallado de cada componente.  Caja negra.  Contempla la ejecución de pruebas unitarias, funcionales y de aceptación, con base en casos de prueba puntuales. Los parámetros de evaluación vienen definidos por los requerimientos del sistema. Este enfoque incluirá un análisis de resultados para componentes del software como algoritmos o cómputo de operaciones. Los parámetros de evaluación vienen dados por los puntos de validación específicos determinados para cada componente probado. |
| **Herramientas de soporte** | Postman |

**- Pruebas no funcionales**

|  |  |
| --- | --- |
| **Objetivo** | Verificar el cumplimiento de los requerimientos no funcionales y de atributos de calidad establecidos para el sistema |
| **Componente a probar** | Módulos del sistema.  Interfaces que interactúan con el sistema  Algoritmos o cómputo de operaciones |
| **Enfoque metodológico** | Caja blanca y caja negra  Se definen y diseñan escenarios de operación de la aplicación dadas las medidas de respuesta de los escenarios de atributos de calidad identificados. Los parámetros de evaluación son las medidas de respuesta asociadas a cada escenario. |
| **Herramientas de soporte** | Ninguna |

## Criterios del proceso de pruebas

**Inicio**

| **Criterio** | **Fases del proceso** |
| --- | --- |
| Los datos para la ejecución de pruebas se encuentran disponibles. | Pruebas funcionales y de aceptación |
| El ambiente de pruebas se encuentra totalmente configurado y listo para entrar en funcionamiento. | Pruebas funcionales y de aceptación |
| La última versión de los objetos a probar se encuentra aplicada en el ambiente de pruebas. | Pruebas funcionales y de aceptación |
| Se encuentran disponibles las herramientas de software necesarias para la ejecución de las pruebas. | Pruebas funcionales y de aceptación |
| El entregable a probar está completo en cuanto al alcance esperado para el ciclo de pruebas. | Pruebas de aceptación |

**Finalización**

| **Criterio** | **Fases del proceso** |
| --- | --- |
| Existen como máximo 10 defectos triviales por módulo o función, pendiente de resolución. | Pruebas funcionales |

**Suspensión**

| **Criterio** | **Fases del proceso** |
| --- | --- |
| Cambios a los requerimientos, no documentados, que afecten la especificación de los casos de prueba. | Pruebas funcionales y de aceptación |
| Modificación del ambiente de pruebas o de los datos para pruebas, sin previa autorización del responsable de las pruebas. | Pruebas funcionales y de aceptación |
| Se detectan más de [6] defectos críticos para un mismo módulo o componente. | Pruebas funcionales |

**Reanudación**

| **Criterio** | **Fases del proceso** |
| --- | --- |
| Notificación y documentación de los cambios realizados con la entrega de la nueva versión para pruebas. | Pruebas funcionales y de aceptación |
| Restauración del ambiente de pruebas y sus datos. | Pruebas funcionales y de aceptación |
| Corrección de los defectos críticos y no triviales que generaron la suspensión del ciclo de pruebas. | Pruebas funcionales y de aceptación |

#### ENTREGABLES DEL PROCESO

| **Entregable** | **Descripción** |
| --- | --- |
| Plan de Pruebas General | Documento donde se presenta la descripción detallada de la estrategia de pruebas a ejecutar. Incluye entre otros aspectos el esquema de trabajo, y el alcance de las pruebas. |
| Plan de Pruebas Detallado | Documento donde se especifica la información pertinente de cada prueba, por cada característica a ser probada. |