**Plan de Pruebas de Software**

***ROUTESOLUTION***

***Fecha: 19/06/2021***

**Tabla de contenido**

Historial de Versiones 4

Información del Proyecto 4

Aprobaciones 4

Resumen Ejecutivo 5

Alcance de las Pruebas 5

Elementos de Pruebas 5

Nuevas Funcionalidades a Probar 6

Pruebas de Regresión 6

Funcionalidades a No Probar 7

Enfoque de Pruebas (Estrategia) 7

Criterios de Aceptación o Rechazo 8

Criterios de Aceptación o Rechazo 8

Criterios de Suspensión 8

Criterios de Reanudación 9

Entregables 9

Recursos 10

Requerimientos de Entornos – Hardware 10

Requerimientos de Entornos – Software 10

Herramientas de Pruebas Requeridas 11

Personal 11

Entrenamiento 12

Planificación y Organización 12

Procedimientos para las Pruebas 12

Matriz de Responsabilidades 13

Cronograma 13

Premisas 14

Dependencias y Riesgos 14

Referencias 15

Glosario 15

**Historial de Versiones**

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| **11/05/2017** | **01** | **GAES 2** | **SENA** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Información del Proyecto**

| Empresa / Organización | TRANSPORTE LA RUTA S.A.S |
| --- | --- |
| Proyecto | ROUTESOLUTION |
| Fecha de preparación | 20/06/2017 |
| Cliente | TRANSPORTE LA RUTA S.A.S |
| Patrocinador principal | SENA |
| Gerente / Líder de Proyecto | RONALD ANTONIO ZEA |
| Gerente / Líder de Pruebas de Software | ANDRES GIOVANNY LINARES |

**Aprobaciones**

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| **RONALD ANTONIO ZEA** | **Analista** | **ROUTESOLUTION** | **20/06/2017** |  |
| **ANDRES GIOVANNY LINARES** | **Analista** | **ROUTESOLUTION** | **20/06/2017** |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Resumen Ejecutivo**

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

**Alcance de las Pruebas**

**Elementos de Pruebas**

* **Módulo de gestión de usuarios**
* **Módulo de gestión de vehículos**
* **Módulo de gestión de agentamientos**
* **Módulo de gestión de suministros**
* **Módulo de reportes**

**Nuevas Funcionalidades a Probar**

* **Validar que el sistema permita registrar personal de la compañía**
* **Validar que el sistema permita crear nuevos usuarios para que interactúen con el sistema**
* **Validar que el sistema permita actualizar la información del personal como: información personal, fotos, etc.**
* **Validar que el sistema permita registrar suministros en el módulo de suministros**
* **Validar que el sistema permita realizar el descuento de suministros en el inventario al momento de suministrarlos para un mantenimiento**
* **Validar que el sistema genere reportes de los servicios del sistema de información**
* **Validar que el sistema cumpla con el 100% de utilidad respecto a los procesos de la compañía**
* **Validar que el sistema permita crear los registros de los vehículos de la compañía**
* **Validar que el sistema genere a tiempo las alertas de documentos próximos a vencer**
* **Validar que el sistema permita cargar documentos como digitalizaciones o imágenes de respectivos documentos.**
* **Validar que el sistema permita ingresar al sistema de manera simultánea varios usuarios al tiempo y que no se cuelgue en ningún proceso.**

**Pruebas de Regresión**

Se realizarán pruebas de regresión ya que se han realizado mejoras utilizando otros frameworks o modificado controladores y métodos para mejoras del sistema.

* Validar que el sistema cuenta con validación de campos.
* Validar que el sistema realice los procesos en tiempos cortos.
* Realizar pruebas de autenticación en el sistema permitiendo pintar el menú correspondiente a cada usuario rol.
* Verificar que el sistema funcione en el navegador de google y Firefox de manera adecuada.

**Funcionalidades a No Probar**

No Aplica

**Enfoque de Pruebas (Estrategia)**

Describe el enfoque general de las pruebas. Para cada grupo o combinación de características hay que especificar el enfoque que asegurará que ese grupo de características se va a probar adecuadamente. Hay que especificar las actividades, técnicas y herramientas principales que se van a usar para cada uno de los grupos de pruebas diseñados.

Existen diferentes grupos de pruebas que se pueden llevar a cabo. Para cada uno de ellos hay que hacer una descripción, indicar las personas que van a llevar a cabo las pruebas y la metodología que se va a seguir. Entre estos grupos están los siguientes:

* Pruebas unitarias
* Pruebas de Integración
* Pruebas de sistema
* Pruebas de carga
* Pruebas de aceptación de usuario
* Pruebas de regresión
* Etc.

**Prueba funcional.**

Es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente [diseñadas](https://es.wikipedia.org/wiki/Dise%C3%B1o_orientado_a_objetos) para el [software](https://es.wikipedia.org/wiki/Software). Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete [informático](https://es.wikipedia.org/wiki/Inform%C3%A1tica). Dicho de otro modo son pruebas específicas, concretas y exhaustivas para probar y validar que el software hace lo que debe y sobre todo, lo que se ha especificado.

**Pruebas de interface.**

Pruebas de interfaz son pruebas de integración que se ocupan de probar las interfaces entre componentes o sistemas.

**Pruebas de desempeño.**

Son un tipo de pruebas que permiten analizar y evaluar las características del software relacionadas con el desempeño,

por ejemplo:

* Tiempos de respuesta
* Throughput (tasa de atención de peticiones)
* Capacidad (máximo trabajo útil por unidad de tiempo)

**Pruebas no funcionales.**

Las pruebas no funcionales se enfocan en validar un sistema o aplicación por medio de sus [requerimientos no funcionales](http://www.pmoinformatica.com/2015/04/requerimientos-no-funcionales-una.html), es decir, la forma en que el sistema funciona y no por medio de comportamientos específicos.  
  
Las características no funcionales de un sistema o aplicación con frecuencia se cuantifican en escalas variables, como por ejemplo los tiempos de respuesta en pruebas de desempeño.

**Criterios de Aceptación o Rechazo**

**Criterios de Aceptación o Rechazo**

A continuación, se listan algunos posibles criterios de aceptación a modo de ejemplo:

1. Un usuario no puede enviar un formulario sin completar todos los campos obligatorios.
2. La información del formulario se almacena en la base de datos de participantes.
3. Está funcionando el filtro Anti-SPAM.
4. Se envía un mensaje de “Registración recibida” luego de recibir la información del formulario.
5. Un usuario no puede registrarse con la misma información de otro usuario.
6. Se deberá chequear por dirección de mail.

**Criterios de Suspensión**

Existen varios tipos de criterios de salida dentro de los cuales se pueden mencionar: cubrimiento de funcionalidades en general, cubrimiento de funcionalidades críticas para el sistema, Número de defectos críticos y mayores detectados, etc.

Lista todos los criterios que se han de satisfacer para que una fase de pruebas se de por finalizada. Entre los posibles ítems que se pueden incluir están:

* Completitud de los casos de pruebas
* Números y severidad de los defectos abiertos
* Paso de los objetivos de pruebas

**Criterios de Reanudación**

Lista todos los criterios que se han de satisfacer para empezar la ejecución de las pruebas. Entre los posibles ítems que se pueden incluir están los siguientes:

* Aprobación del plan de pruebas
* Entorno de pruebas estable y preparado
* Casos de pruebas escritos y aprobados
* Herramientas de pruebas preparadas
* Recursos para las pruebas disponibles

**Entregables**

Los documentos que entregaremos en parte de evidencia seria el formatos de plan de pruebas, formato casos de pruebas, matriz de resultados, códigos o pantallazos de pruebas de jUnit.

**Recursos**

**Requerimientos de Entornos – Hardware**

* 2 computadores para pruebas
* Interfaz X10 para computador CM11A
* Sensor de movimiento X10 MS14A
* Transmisor/Receptor X10 TM751
* Transmisor Universal X10 PF284/PSC01
* Modulo de Aparato X10 AM486
* Webcam Logitech QuickCam Express
* Webcam 3Com

**Requerimientos de Entornos – Software**

* Apache Tomcat 5.5
* Apache DS
* Windows 1 pro
* Internet Explorer 11
* Mozilla Firefox 2
* Java Developer Kit 1.6
* Java Runtime Environment 1.6
* Flash Media Player 11
* [JUnit](https://es.wikipedia.org/wiki/JUnit): Entorno de pruebas para Java creado por [Erich Gamma](https://es.wikipedia.org/wiki/Erich_Gamma) y [Kent Beck](https://es.wikipedia.org/wiki/Kent_Beck). Se encuentra basado en SUnit creado originalmente para realizar pruebas unitarias para el lenguaje Smalltalk.
* [TestNG](https://es.wikipedia.org/wiki/TestNG): Creado para suplir algunas *deficiencias* en JUnit.
* [JTiger](https://es.wikipedia.org/w/index.php?title=JTiger&action=edit&redlink=1): Basado en anotaciones, como TestNG.
* [MOQ](https://es.wikipedia.org/w/index.php?title=MOQ&action=edit&redlink=1): Entorno para la creación dinámica de objetos simuladores (mocks). [«MOQ»](https://github.com/Moq/moq4/).
* [QUnit](https://es.wikipedia.org/w/index.php?title=QUnit&action=edit&redlink=1): Librería para pruebas unitarias en Javascript. Creada por la fundación jQuery, ha sido reescrita para ser independiente de la librería jQuery.
* [libunittest](http://libunittest.sourceforge.net/): Librería portable para pruebas unitarias en C++ que usa el nuevo estándar C++11.
* Sql
* Mysql
* oracle

**Herramientas de Pruebas Requeridas**

**1.- Selenium**

Es un framework para pruebas de aplicaciones Web, descargable de forma gratuita desde su sitio web. Proporciona una herramienta de gabración y playback, que permite desarrollar pruebas sin necesidad de aprender un lenguaje de Scripting.  
  
Incluye características como grabación, playback, selección de campos, auto completar formularios, pruebas de recorrido (Walkthrough), debug, puntos de control, scripts ruby y otros formatos.

**Personal**

* Coordinador de pruebas
* Analista de pruebas

**Entrenamiento**

**-Apropiación de pruebas**

**-Desarrollos de la lista de casos de pruebas**

**-Desarrollo de los casos de pruebas**

**-Desarrollo de los scripts de pruebas**

**-Preparación del entorno de Pruebas**

**-Fechas de ejecución de pruebas**

**Planificación y Organización**

**Procedimientos para las Pruebas**

Esta sección debería incluir una lista de hitos clave en las pruebas. Puede incluir:

* Aprobación del plan de pruebas
* Desarrollos de la lisara de casos de pruebas
* Desarrollo de los casos de pruebas
* Desarrollo de scripts de pruebas
* Preparación del entorno de pruebas
* Fechas de ejecución de pruebas

**Matriz de Responsabilidades**



**Cronograma**

Debe estar basado en estimaciones de actividades realizadas por el equipo de prueba. En él se Identifican los hitos relevantes en las pruebas de software, se establecen las dependencias (actividades predecesoras) y demás aspectos componentes de un cronograma.

**Premisas**

* Falta de recursos y baja competencia en pruebas -
* Falta de los recursos necesarios para ejecutar las pruebas según el plan+
* Tiempo reducido asignado a la fase de pruebas +
* Cambios frecuentes en la definición de los objetivos y alcance del plan de pruebas +
* Falta de coordinación entre los equipos de desarrollo y testing -
* Falta de experiencia con nuevas tecnologías, herramientas, lenguajes de programación, … +

**Dependencias y Riesgos**

**1. Intervención temprana del equipo de pruebas en el proyecto**

La inclusión del equipo de pruebas en las etapas iniciales del desarrollo del producto ayudará a obtener mayor conocimiento del mismo así como permitirá detectar posibles defectos en etapas tempranas, por lo que el coste de resolución de los mismos será inferior.

**2. Preparación de las pruebas**

Antes de comenzar el desarrollo del producto, el equipo de pruebas podrá comenzar a diseñar el plan a seguir, así como identificar futuras necesidades. Herramientas a utilizar, configuración de entornos, …

**3. Definición de los criterios de entrada – salida**

No refiriéndose a los datos, sino los puntos de unión con otras plataformas e integraciones con terceros. Es muy útil definir y mantener las interfaces y mecanismos de comunicación con terceros para evitar futuros problemas.

**4. Requerimientos de pruebas**

Desde el equipo de pruebas, se fomentará el uso de estándares, tecnologías abiertas, así como buenas prácticas de desarrollo (por ejemplo TDD, integración continua, etc)

**5. Gestión de defectos**

Una tarea de gran importancia es el seguimiento y priorización de los defectos encontrados. Estos deben ser incluidos en los planings de siguientes iteraciones para que sean resueltos. Además, deben ser trazados para conocer cuando y en qué versión han sido resueltos.

Siguiendo estos puntos, conseguiremos reducir en gran medida los riesgos más comunes durante el desarrollo de software. Hay que tener en cuenta que se debe trabajar en sincronía con los demás grupos implicados, desde la parte de gestión, pasando por desarrollo, pruebas, despliegue, … Unos dependen de otros y los problemas de unos se propagan a otros.

**Referencias**

Lista de todos los documentos que pueden citarse como apoyo o para ampliar el contenido del plan de pruebas. Algunos ejemplos de lo que se puede hacer referencia aquí son:

* Plan de Proyecto.
* Especificaciones de Requerimientos.
* Diseño General.
* Diseño Detallado.
* Procedimientos y estándares de Desarrollo.
* Procedimientos y estándares de Pruebas.
* Metodologías, Procedimientos y estándares corporativos.

**Glosario**

Definiciones de términos usados en la documentación, y general sobre el área de pruebas.

**Filtro:** Materia porosa, a través de la cual se hace pasar un fluido para clarificarlo o depurarlo.

**Proyecto:** Idea de una cosa que se piensa hacer y para la cual se establece un modo determinado y un conjunto de medios necesarios.

**Diseño:** Actividad creativa que tiene por fin proyectar objetos que sean útiles y estéticos.

**Defectos:** Se llama defecto a la carencia o falta de las [cualidades](https://es.wikipedia.org/wiki/Cualidad) propias y naturales de una cosa o a cualquier imperfección natural o moral.

**Estándares:** Consiste en el establecimiento de normas a las que debe ajustarse la información geográfica, los procesos de intercambio de ésta y la interoperación de los sistemas que deben manejarla.

**Tester:** Un tester investiga un producto de software con el objetivo de obtener información acerca de su calidad y del valor que representa para quienes lo utilizan.