**Plan de pruebas de software**

***[Proyecto CarMotorFix]***

***Fecha: [04/09/2024]***

**Historial de versiones**

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| 27/10 | 1.0 | Esteban Inzunza | CarMotorFix |  |
|  |  |  |  |  |

Información del proyecto

| Empresa / Organización | N/A |
| --- | --- |
| Proyecto | CarMotorFix |
| Fecha de preparación | 04-09-2024 |
| Cliente | N/A |
| Patrocinador principal | N/A |
| Gerente / Líder de proyecto | Paulo Escobar |
| Gerente / Líder de pruebas de software | Esteban Inzunza |

Aprobaciones

| **Nombre y Apellido** | **Cargo** | **Departamento u organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| Paulo Escobar | Co-Líder Pruebas |  | 26/10/2024 | 20232450-9 |
| Esteban Inzunza | Lider de pruebas |  | 26/10/2024 | 20698320-5 |
| Diego Lastra | Tester |  | 26/10/2024 | 21276258-k |

Resumen ejecutivo

Alcance de las pruebas

Elementos de pruebas

Las pruebas cubrirán los requerimientos funcionales, incluyendo la creación, modificación y finalización de órdenes de trabajo (OT), gestión de usuarios (clientes y mecánicos), registro de vehículos, generación de cotizaciones, gestión de servicios y autenticación. También se realizarán pruebas para verificar la seguridad de los datos, autenticación de usuarios, roles y permisos, y notificaciones. En cuanto a los requerimientos no funcionales, se validará la usabilidad, compatibilidad con dispositivos comunes en talleres mecánicos y cumplimiento con normativas de privacidad. No se realizarán pruebas en dispositivos fuera de los sistemas operativos asumidos ni en funcionalidades no incluidas en el alcance del proyecto.

Nuevas funcionalidades a probar

Al ser el primer software entregado para la realización de pruebas, no existen parámetros de comparación anteriores, como versiones alpha o beta.

Pruebas de regresión

En este programa será innecesarias por que el programa está en una versión muy temprana del producto final

**Funcionalidades a no probar**

* Modo oscuro: Es una funcionalidad estética que no afecta la lógica del sistema ni la interacción con el usuario.
* Visualización de la calificación del mecánico: Aunque es importante para la experiencia del usuario, no tiene impacto en los procesos principales como la gestión de OT.

Enfoque de pruebas (estrategia)

Se utilizarán dos estrategias principales: **pruebas funcionales** y **pruebas de humo**, orientadas a garantizar la calidad del software.

### Pruebas Funcionales

* Estas pruebas se centran en verificar que cada funcionalidad cumple con los requerimientos del negocio definidos para el sistema.
* Permiten validar que el software realiza correctamente las tareas para las que fue diseñado.
* Funcionalidades a evaluar:
  + Creación, asignación y modificación de órdenes de trabajo (OT).
  + Autenticación de usuarios y roles.
  + Generación de cotizaciones.
  + Registro de usuarios
  + Registro de Vehículos (Cliente, Mecanico)
  + Actualización de estado del vehículo
  + Creación de Servicio (Admin)
  + Deshabilitar un Servicio (Admin)
  + Asignación de Roles (Admin)

**Objetivo:** Confirmar que todas las características funcionales del sistema cumplen con las expectativas y los casos de uso definidos.

### Pruebas de Humo

* Estas pruebas son esenciales para validar el **funcionamiento básico del software** antes de continuar con pruebas más detalladas.
* Se enfocan en las **funcionalidades críticas y principales**, asegurando que el sistema está estable y que los módulos clave operan como se espera.
* Áreas clave a evaluar:
  + Inicio de sesión de usuarios (clientes y mecánicos).
  + Creación de órdenes de trabajo básicas.
  + Actualización de estados en las órdenes y vehículos.
  + Generación y visualización de cotizaciones.

**Objetivo:** Detectar problemas graves que puedan impedir el funcionamiento básico del software, asegurando que los componentes principales están operativos.

Criterios de aceptación o rechazo

**Cumplimiento de Requerimientos:**

* La funcionalidad debe satisfacer completamente los requerimientos funcionales y no funcionales especificados en la **Planilla de Requerimientos**.
* Ejemplo: La creación de órdenes de trabajo debe permitir agregar todos los datos requeridos y actualizar correctamente el estado.

**Resultados Esperados:**

* Las pruebas deben producir resultados exactos y consistentes en cada ejecución.
* Ejemplo: Una cotización generada debe reflejar los precios correctos, descuentos aplicables y totales calculados sin errores.

**Cumplimiento de Casos de Prueba:**

* Al menos el 95% de los casos de prueba relacionados con la funcionalidad deben ser aprobados exitosamente.
* Los defectos menores o aceptables deben ser documentados y aprobados por los interesados.

| **Criterio** | **Descripción** |
| --- | --- |
| Aprobación | Como criterio de aceptación se podría considerar que el software pase las pruebas funcionales (60%) y las pruebas de humo (90%), pasando las 2 pruebas definidas ya se podría considerar un de aceptación del software. |
| Rechazado | En caso de que las pruebas de Humo y posteriormente las pruebas funcionales no cumplan con el porcentaje establecido se dará como rechazado el proyecto |

Criterios de suspensión

**Pruebas Funcionales:**

* Si el porcentaje de fallos supera el 40% del total de casos de prueba ejecutados, se pausará la labor de pruebas funcionales. En este caso, el equipo de desarrollo deberá corregir los defectos identificados antes de continuar con las pruebas.
* Justificación: Un nivel de fallos tan alto indica problemas significativos en el cumplimiento de los requisitos, lo que hace ineficiente continuar probando sin resolver los problemas subyacentes.

**Pruebas de Humo:**

* Al ser estas pruebas esenciales para validar el funcionamiento básico y las características críticas del sistema, se establece un umbral máximo de 10% de fallos. Si se supera este límite, las pruebas se detendrán y se solicitará una revisión inmediata de los módulos o funcionalidades afectados.
* Justificación: Las pruebas de humo validan la estabilidad del sistema; cualquier fallo significativo en esta etapa pone en duda la viabilidad de continuar con pruebas más detalladas.

Criterios de reanudación

Las pruebas se reanudarán únicamente cuando se cumplan las siguientes condiciones para garantizar que el sistema esté en un estado adecuado para ser evaluado nuevamente:

**Pruebas Funcionales:**

* Se requerirá que al menos el **80% de los casos de prueba funcionales** previamente fallidos hayan sido solucionados y verificados con éxito.

**Pruebas de Humo:**

* Al ser estas pruebas esenciales para validar las funcionalidades básicas del sistema, al menos el **90% de los casos de prueba de humo** deben haber sido corregidos y verificados.

**Revisión de Impacto:**

* Se deberá realizar un análisis posterior a las correcciones para evaluar si los cambios realizados han afectado otras áreas del sistema.

Entregables

**Plantilla de casos de Prueba**

[Casos de Prueba](https://docs.google.com/spreadsheets/d/1aqnF2LIhtcizM0i9sFdYm-OZ73EpdynB/edit?gid=70891680#gid=70891680)

**Matriz de Trazabilidad**[Matriz de Trazabilidad](https://docs.google.com/spreadsheets/d/103gUfJC94vEVxE2yZaj8-gisogkvSGqY/edit?gid=13984877#gid=13984877)

Recursos

Requerimientos de entornos – Hardware

* Procesador Intel Celeron o superior
* RAM 4,00 GB DDR3 1333 mhz
* Tipo de sistema Sistema operativo de 32 bits, procesador x32
* Disco duro externo de 100gb

Requerimientos de entornos – Software

* Windows 10 o superior

Personal

El personal necesario sería :

-Jefe de Proyecto : Paulo Escobar

-Líder de Pruebas : Esteban Inzunza

-Tester : Diego Lastra

**Entrenamiento**

El personal deberá pasar por una capacitación en la cual se enseñara a utilizar lo básico del software para así levantar el servidor, además, se demostrará el funcionamiento del software para que puedan aplicar las pruebas

Planificación y organización

**Procedimientos para las pruebas**

Las pruebas que se realizarán se usarán en base a la metodología top-down

**Secuencias de pasos para la Creación de Orden de Trabajo.**

Ejecución manual del proceso de creación de OT, donde los campos requeridos serán llenados con datos válidos y guardados. Posteriormente, se intentará acceder a la OT para confirmar que la información registrada coincide con los datos ingresados.

**Secuencias de pasos para la Asignación de Orden de Trabajo.**

Ejecución manual del proceso de asignación de OT, seleccionando un mecánico disponible. Se verificará que la OT se actualice correctamente con el mecánico asignado y que la información sea visible en la lista de OT.

**Secuencias de pasos de la Modificación de Orden de Trabajo**

Ejecución manual del proceso de edición de una OT existente, modificando campos específicos como servicios o fechas. Se guardarán los cambios y se verificará que se reflejan en el sistema correctamente.

**Secuencias de pasos para la Actualización de Estados de OT.**

Ejecución manual del proceso de cambio de estado en una OT, seleccionando estados diferentes (pendiente, en proceso, completada). Se confirmará que el sistema registre correctamente el cambio y lo muestre en el historial.

**Secuencias de pasos para el Inicio de Sesión de Usuarios (Clientes)**

Ejecución manual del proceso de ingreso al sistema, donde un cliente utilizará credenciales válidas para acceder. También se probarán credenciales incorrectas para confirmar que el sistema arroje un mensaje de error y no permita el acceso.

**Secuencias de pasos para la Autenticación de Usuarios**

Ejecución manual del proceso de autenticación, verificando que los usuarios con credenciales válidas accedan según su rol. En caso de credenciales incorrectas, se confirmará que el sistema no permita el acceso y arroje un mensaje de error.

**Secuencias de pasos para la Asignación de Roles**

Ejecución manual del proceso de edición de usuarios para asignar roles. Se verificará que los permisos asociados al nuevo rol sean aplicados y que el usuario pueda realizar las funciones correspondientes.

**Secuencias de pasos para la Actualización de Estado del Vehículo**

Ejecución manual del proceso de cambio de estado en un vehículo asociado a una OT. Se verificará que los estados actualizados sean registrados y reflejados en la interfaz de usuario.

Matriz de responsabilidades

[MATRIZ DE RESPONSABILIDADES.xlsx](https://docs.google.com/spreadsheets/d/1IPadkCX4exzxZd25HzyLJ_hMAPjXwtNJ/edit?usp=drive_link&ouid=108819358622315356188&rtpof=true&sd=true)

Premisas

Las siguientes son las premisas o suposiciones consideradas en este plan de pruebas, cuya alteración podría generar impactos significativos en la planificación y ejecución del proyecto:

* El personal encargado de las pruebas deberá tener conocimientos básicos en la ejecución de pruebas manuales y manejo de herramientas básicas de soporte técnico.
* Se utilizará un enfoque manual para las pruebas, sin herramientas de automatización, lo que requiere mayor tiempo y dedicación en la ejecución de las mismas.
* Se seguirá estrictamente el cronograma establecido para evitar retrasos en la entrega del proyecto.
* El equipo de pruebas tendrá acceso completo al entorno de pruebas, incluyendo credenciales para usuarios de distintos roles (cliente, mecánico, administrador).
* Se garantiza que el entorno de desarrollo estará disponible para realizar las pruebas, con todas las funcionalidades requeridas implementadas previamente.
* Los casos de prueba se diseñarán en función de los requerimientos funcionales y no funcionales aprobados en la etapa de análisis.

Dependencias y Riesgos

Estos son algunos de los riesgos asociados con el proceso de prueba de software con ejemplos de factores de riesgo:

* Los riesgos no son identificados correctamente por lo tanto no se puede llevar un correcto control de estos ni pueden ser solucionados a tiempo
* No cumplir con el tiempo indicado
* No capacitar al personal a tiempo para probar el software de manera correcta
* Los recursos del sistema no son suficientes por lo tanto el sistema no funciona en su totalidad
* Personal sin conocimiento de la base de datos lo que imposibilita evaluar su seguridad, tipo de datos y tiempos de respuesta

Referencias

[**MATRIZ DE TRAZABILIDAD DE REQUISITOS – Proyectos de ingenieria de sistemas (wordpress.com)**](https://prosistemas85133078.wordpress.com/matriz-de-trazabilidad-de-requisitos/)

Glosario

**Pruebas funcionales**: Pruebas orientadas a validar que el sistema cumple con los requisitos definidos por el negocio. Evalúan el resultado de acciones específicas, sin analizar los estados intermedios del sistema durante la ejecución.

**Pruebas de humo**: Pruebas iniciales y básicas diseñadas para verificar que los componentes principales del software funcionan correctamente. Son rápidas de ejecutar y aseguran que el sistema sea apto para pruebas más detalladas.

**Orden de Trabajo (OT)**: Documento digital o registro que detalla las tareas asignadas a un mecánico, incluyendo información sobre el cliente, el vehículo y las reparaciones o mantenimientos a realizar.

**Tester**: Persona encargada de planificar y ejecutar pruebas en el software para identificar errores, riesgos potenciales y garantizar que el sistema cumpla con los estándares de calidad establecidos.

**Roles de usuario**: Clasificación de usuarios dentro del sistema según sus permisos y responsabilidades, tales como administrador, cliente o mecánico.

**Gestión de estados**: Proceso en el que se actualiza el estado de una entidad del sistema, como una OT o un vehículo, reflejando el progreso o cambios realizados en las operaciones.

**Autenticación de usuarios**: Proceso que verifica la identidad de un usuario para garantizar que solo personas autorizadas puedan acceder al sistema, generalmente mediante credenciales como correo electrónico y contraseña.

**Matriz de trazabilidad**: Herramienta que relaciona los requerimientos del sistema con los casos de prueba, asegurando que todos los aspectos importantes han sido cubiertos durante el proceso de pruebas.