**Plan de pruebas de software**

***Fecha:* 03/11/2024**

**Historia de Revisiones**

| Seguro A Casa |  | Versión: 1.1.0 |
| --- | --- | --- |
| Plan de Pruebas |  | Fecha: 2024-10-05 |
| <Identificador de Documento> | | |

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 2024/11/03 | <1.1.0> | Primeras pruebas a la versión web | Diego Abarca, Guillermo Marin |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  | Versión: <1.1.0> |
| --- | --- |
| Plan de Testing | Fecha: 2024/11/03 |
| Document2 | |

**Tabla de Contenidos**

**1** **Introducción 6**

1.1 Resumen ejecutivo 6

1.2 Alcance de las pruebas 6

*1.2.1* *Elementos de pruebas 6*

*1.2.2* *Pruebas funcionales 6*

*1.2.3* *Riesgos 6*

1.3 Definiciones, acrónimos y abreviaciones 8

1.4 Referencias 8

**2** **Requerimientos para pruebas 9**

2.1 Casos de uso 9

*2.1.1* *Vista global 9*

*2.1.2* *Caso de uso 1 9*

*2.1.3* *Caso de uso 2 9*

*2.1.4* *Caso de uso 3 9*

2.2 Requerimientos funcionales 9

*2.2.1* *Componentes comunes 9*

*2.2.2* *Componente 1 9*

*2.2.3* *Componente 2 9*

2.3 Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica) 9

*2.3.1* *Componente 1 (puede tener uno o más casos de uso) 9*

*2.3.2* *Componente 2 (puede tener uno o más casos de uso) 9*

**3** **Estrategia de Pruebas 10**

3.1 Tipos de pruebas 10

*3.1.1* *Pruebas funcionales 10*

*3.1.2* *Pruebas de rendimiento (Performance) 11*

*3.1.3* *Pruebas de seguridad y de acceso a datos 11*

*3.1.4* *Herramientas involucradas 11*

**4** **Recursos 12**

4.1 Profesionales 12

*15*

*4.2.4* *Integración del ambiente de pruebas y configuración 16*

*4.2.5* *Generación de datos 16*

**5** **Actividades e Hitos del Plan de Pruebas 16**

**6** **Entregables 19**

6.1 Plan de pruebas 19

*6.1.1* *Criterio de entrada para el “Plan de pruebas” 19*

*6.1.2* *Criterio de salida para el “Plan de pruebas” 19*

*6.1.3* *Criterio de suspensión y resumisión 19*

6.2 Resultados de las pruebas 19

6.3 Reporte de defectos 19

# Introducción

## Resumen ejecutivo

El proyecto consiste en desarrollar una aplicación móvil en React Native con Expo y JavaScript para registrar la ubicación de furgones escolares y gestionar la información de nuevos trabajadores. Además, se ha creado una página web utilizando React y Supabase, donde el administrador podrá registrar a los conductores y gestionar toda la información relacionada. La aplicación busca digitalizar y automatizar procesos manuales, mejorando la eficiencia operativa y reduciendo errores en la gestión de datos. Con una interfaz intuitiva, permitirá a los usuarios acceder fácilmente a la información de los furgones y trabajadores, garantizando una experiencia de usuario fluida en plataformas iOS y Android.

## Alcance de las pruebas

### El alcance incluye la validación de los siguientes módulos:

#### **1. Aplicación Móvil (React Native)**

* **Módulo de Seguimiento de Ubicación:**
  + Obtención y visualización en tiempo real de la ubicación de los furgones.
  + Actualización automática de la ubicación en el mapa.
  + Notificaciones de cambios de estado de los furgones (en ruta, detenido, etc.).
* **Interfaz de Usuario (UI):**
  + Navegación entre pantallas (login, registro, inicio).
  + Accesibilidad y diseño responsivo.
  + Experiencia de usuario (UX) general.
* **Autenticación:**
  + Registro y login de usuarios.
  + Control de acceso según roles de usuario.

#### **2. Página Web (React y Supabase)**

* **Módulo de Administración:**
  + Registro de conductores por parte del administrador.
  + Edición y eliminación de registros de conductores.
  + Visualización de la lista de conductores registrados.
* **Gestión de Datos:**
  + CRUD (Crear, Leer, Actualizar, Eliminar) de datos de furgones y conductores.
  + Búsqueda y filtrado de registros.
  + Exportación de datos (si aplica).
* **Interfaz de Usuario (UI):**
  + Navegación intuitiva en el panel administrativo.
  + Accesibilidad y diseño responsivo.
  + Experiencia de usuario (UX) general.
* **Seguridad y Autenticación:**
  + Control de acceso administrativo.
  + Validación de sesiones y permisos.

### Pruebas funcionales

#### **Objetivo:**

Verificar que el sistema permita a los usuarios registrarse correctamente, asegurando que los datos ingresados sean válidos y se manejen adecuadamente los errores.

#### **Datos de Entrada (Inputs):**

1. **Caso 1 (Entrada válida):**
   * Nombre de usuario: "usuario123"
   * Email: "usuario123@example.com"
   * Contraseña: "Password123!"
   * Confirmar contraseña: "Password123!"
2. **Caso 2 (Valores límite):**
   * Nombre de usuario: "u" (mínimo permitido: 1 carácter)
   * Email: "u@e.com" (mínimo válido)
   * Contraseña: "Pass!" (mínimo permitido de 6 caracteres, con caracteres especiales)
   * Confirmar contraseña: "Pass!"
3. **Caso 3 (Valores ilegales):**
   * Nombre de usuario: "usuario@123" (caracteres no permitidos)
   * Email: "usuario.com" (formato de email inválido)
   * Contraseña: "12345" (demasiado corta)
   * Confirmar contraseña: "Password123" (no coincide con la contraseña)
4. **Caso 4 (Campo vacío):**
   * Nombre de usuario: "" (campo vacío)
   * Email: "" (campo vacío)
   * Contraseña: "" (campo vacío)
   * Confirmar contraseña: "" (campo vacío)

#### **Resultados Esperados (Outputs):**

1. **Caso 1 (Entrada válida):**
   * El sistema debe permitir el registro del usuario y mostrar el mensaje: "Registro exitoso. Bienvenido, usuario123".
2. **Caso 2 (Valores límite):**
   * El sistema debe permitir el registro y mostrar el mensaje de éxito, siempre que se cumplan los requisitos.
3. **Caso 3 (Valores ilegales):**
   * El sistema debe rechazar el registro y mostrar mensajes de error específicos, como:
     + "Nombre de usuario inválido" (para caracteres no permitidos).
     + "Email no válido" (para formato de email incorrecto).
     + "La contraseña debe tener al menos 6 caracteres" (para contraseñas demasiado cortas).
     + "Las contraseñas no coinciden" (cuando las contraseñas no coinciden).
4. **Caso 4 (Campo vacío):**
   * El sistema debe rechazar el registro y mostrar mensajes de error específicos como:
     + "El nombre de usuario es obligatorio".
     + "El email es obligatorio".
     + "La contraseña es obligatoria".

### Riesgos

Algunos riesgos comunes a considerar son:

1. Documentación de especificación errónea o incompleta.
2. Lista de requerimientos incoherentes con los casos de uso.
3. Componentes a probar y componentes comunes correspondan a distintas versiones.
4. Hardware y software no funcionan correctamente.
5. Herramientas de *testing* automatizado están mal configuradas.

Los riesgos serán identificados de acuerdo a un concepto de Bajo, Medio o Alto, dependiendo de la importancia del caso de uso para el cual se está desarrollando el *testing*.

Así mismo, se han identificado una serie de riesgos (calificados entre 1 y 10 dependiendo de su gravedad) los que están detallados en el artefacto “Lista de riesgos” con sus alcances y acciones, en el presente plan, son enunciados para sugerir algunas acciones.

#### Matrices de riesgos

##### Pruebas

| **Nº Riesgo** | **Descripción** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1 | La documentación de especificaciones puede contener errores o estar incompleta. |  | Revisar y validar la documentación con el equipo de desarrollo y stakeholders. |
| 2 | La lista de requerimientos puede no coincidir con los casos de uso. |  |  |
| 3 | Probar componentes de diferentes versiones puede causar fallos debido a incompatibilidades. |  |  |
| 4 | Dependencias en servicios externos (como APIs) pueden afectar la capacidad para realizar pruebas efectivas |  |  |
| 5 | Modificaciones en los requerimientos después de iniciar las pruebas. |  |  |
| 6 | Problemas técnicos en el hardware o software utilizados durante las pruebas. |  | Realizar pruebas en entornos controlados con hardware y software correctamente configurados. |

##### 

## Definiciones, acrónimos y abreviaciones

#### **Definiciones:**

* **Aplicación Móvil:** Software desarrollado para dispositivos móviles (teléfonos, tabletas) que permite a los usuarios realizar tareas específicas.
* **Furgón Escolar:** Vehículo utilizado para el transporte de estudiantes hacia y desde sus instituciones educativas.
* **Registro:** Proceso de creación de una cuenta o almacenamiento de datos en el sistema.
* **Administración:** Funcionalidad otorgada a usuarios con privilegios especiales para gestionar recursos dentro del sistema.
* **Navegación:** Acción de moverse entre diferentes pantallas o secciones dentro de una aplicación o página web.

#### **Acrónimos:**

* **UI (User Interface):** Interfaz de Usuario. El conjunto de elementos visuales y controles a través de los cuales los usuarios interactúan con la aplicación.
* **UX (User Experience):** Experiencia de Usuario. Se refiere a cómo el usuario percibe la facilidad de uso, accesibilidad y eficiencia del sistema.
* **CRUD (Create, Read, Update, Delete):** Crear, Leer, Actualizar y Eliminar. Son las operaciones básicas para gestionar datos en aplicaciones.
* **API (Application Programming Interface):** Interfaz de Programación de Aplicaciones. Conjunto de definiciones y protocolos para la integración de software.

#### **Abreviaciones:**

* **JS:** JavaScript. Lenguaje de programación utilizado para desarrollar la lógica del frontend en el proyecto.
* **RN:** React Native. Framework para desarrollar aplicaciones móviles nativas usando JavaScript y React.
* **DB:** Base de Datos. Conjunto de datos organizados que permiten su almacenamiento, gestión y recuperación de forma eficiente.
* **PW:** Página Web. Plataforma accesible a través de navegadores web donde los administradores pueden gestionar conductores y trabajadores.

# Requerimientos para Pruebas

1. **Casos de Uso**

UC-01: Registro de Usuario

* Verificar que el usuario pueda registrarse proporcionando un nombre, correo electrónico único y contraseña válida.
* Validar que el sistema detecta y notifica errores cuando los datos sean inválidos o incompletos.

UC-02: Inicio de Sesión

* Comprobar que el usuario pueda iniciar sesión con credenciales válidas.
* Verificar que se bloqueen intentos de inicio de sesión con credenciales incorrectas y que se muestren mensajes de error claros.

UC-03: Recuperación de Contraseña

* Validar que los usuarios puedan solicitar una nueva contraseña si olvida la actual.
* Confirmar que el sistema envíe un correo de recuperación solo a usuarios registrados.

UC-04: Gestión de Furgones

* Verificar que el administrador pueda crear, editar y eliminar registros de furgones.
* Validar que las listas de furgones se actualicen correctamente tras cada operación.

UC-05: Seguimiento de Ubicación

* Comprobar que la ubicación en tiempo real de los furgones se muestre en el mapa.
* Validar que el sistema actualice continuamente la posición de los furgones mientras están en movimiento.

UC-06: Gestión de Conductores

* Confirmar que el administrador pueda gestionar registros de conductores (crear, editar y eliminar).
* Verificar que las listas de conductores reflejen las actualizaciones en tiempo real.

UC-07: Configuración de Roles

* Comprobar que el administrador puede asignar o cambiar roles de usuarios registrados.
* Validar que los roles asignados otorguen o restrinjan el acceso adecuado a las funcionalidades.

UC-8: Cierre de Sesión

* Confirmar que los usuarios puedan cerrar sesión de manera segura.
* Verificar que las sesiones activas se terminen correctamente.

2. **Requerimientos Funcionales**

* RF-01: Validación de datos.
  + El sistema debe validar correos electrónicos únicos y contraseñas que cumplan con criterios de seguridad (mínimo 8 caracteres, al menos una mayúscula y un número).
* RF-02: Notificaciones de error.
  + El sistema debe proporcionar mensajes de error claros y específicos cuando ocurran problemas en las operaciones.
* RF-03: Operaciones CRUD.
  + El sistema debe permitir crear, leer, actualizar y eliminar datos de conductores y furgones.
* RF-04: Seguimiento en tiempo real.
  + El sistema debe mostrar la ubicación de los furgones en un mapa actualizado dinámicamente.
* RF-05: Roles y permisos.
  + Los usuarios deben tener acceso a funcionalidades específicas según su rol.

3. **Requerimientos No Funcionales**

* RNF-01: Rendimiento.
  + El sistema debe soportar hasta 1000 usuarios concurrentes sin degradación significativa en el rendimiento.
* RNF-02: Seguridad.
  + Las contraseñas deben almacenarse encriptadas en la base de datos.
  + Las sesiones deben expirar automáticamente tras un periodo de inactividad.
* RNF-03: Usabilidad.
  + La plataforma debe permitir completar las tareas principales (registro, inicio de sesión, gestión de datos) en menos de 3 minutos.
* RNF-04: Compatibilidad.
  + La aplicación móvil debe ser compatible con dispositivos iOS (versión 14+) y Android (versión 10+).
* RNF-05: Escalabilidad.
  + La base de datos debe poder manejar más de 5000 registros sin afectar el tiempo de respuesta.

# 

## Casos de uso

### **UC-01: Registro de Usuario**

### **Descripción**: El usuario debe poder registrarse en la plataforma proporcionando su nombre, correo electrónico y contraseña.

### **Actor principal**: Usuario.

### **Precondiciones**:

### El usuario debe tener acceso a la plataforma (móvil o web).

### El correo electrónico ingresado no debe estar registrado previamente.

### **Flujo principal**:

### El usuario selecciona la opción "Registrarse" en la pantalla de inicio.

### Ingresa su nombre, correo electrónico y contraseña en el formulario de registro.

### Confirma su contraseña.

### Presiona el botón "Registrar".

### El sistema valida los datos ingresados.

### Si todo es válido, el sistema almacena los datos en la base de datos.

### Se muestra un mensaje de éxito: "Registro exitoso. Bienvenido, [nombre de usuario]".

### **Flujo alternativo**:

### **A1**: Si los datos ingresados no son válidos:

### El sistema muestra un mensaje de error indicando el campo problemático.

### **A2**: Si hay campos vacíos:

### El sistema muestra un mensaje: "Todos los campos son obligatorios".

### **Postcondiciones**:

### El usuario queda registrado y puede iniciar sesión.

### La información se almacena de forma segura.

### **UC-02: Inicio de Sesión**

### **Descripción**: El usuario debe poder iniciar sesión utilizando su correo electrónico y contraseña.

### **Actor principal**: Usuario.

### **Precondiciones**:

### El usuario debe estar registrado en la plataforma.

### Debe tener acceso a su correo electrónico y contraseña.

### **Flujo principal**:

### El usuario selecciona la opción "Iniciar Sesión".

### Ingresa su correo electrónico y contraseña en los campos correspondientes.

### Presiona el botón "Iniciar Sesión".

### El sistema valida las credenciales.

### Si las credenciales son correctas, se redirige al usuario a la pantalla de inicio correspondiente a su rol.

### **Flujo alternativo**:

### **A1**: Si las credenciales son incorrectas:

### El sistema muestra un mensaje: "Correo o contraseña incorrectos".

### **A2**: Si algún campo está vacío:

### El sistema muestra un mensaje: "Debe completar todos los campos".

### **Postcondiciones**:

### Se establece una sesión activa para el usuario.

### El sistema registra el inicio de sesión.

### **UC-03: Recuperación de Contraseña**

### **Descripción**: El usuario debe poder solicitar la recuperación de su contraseña a través de su correo electrónico.

### **Actor principal**: Usuario.

### **Precondiciones**:

### El usuario debe tener un correo electrónico válido registrado en la plataforma.

### **Flujo principal**:

### El usuario selecciona la opción "¿Olvidaste tu contraseña?" en la pantalla de inicio de sesión.

### Ingresa su correo electrónico en el formulario de recuperación.

### Presiona el botón "Enviar".

### El sistema valida que el correo ingresado esté registrado.

### Si es válido, el sistema envía un correo con un enlace o código para restablecer la contraseña.

### El usuario accede al enlace o utiliza el código para crear una nueva contraseña.

### El sistema guarda la nueva contraseña y muestra un mensaje: "Contraseña actualizada exitosamente".

### **Flujo alternativo**:

### **A1**: Si el correo no está registrado:

### El sistema muestra un mensaje: "Correo no registrado en la plataforma".

### **A2**: Si el usuario no completa el formulario:

### El sistema muestra un mensaje: "Debe ingresar un correo electrónico".

### **Postcondiciones**:

### El usuario puede iniciar sesión con su nueva contraseña.

### La contraseña anterior queda invalidada.

### 

### **UC-04: Gestión de Furgones**

### **Descripción**: El administrador debe poder registrar nuevos furgones, editar información existente y eliminar furgones no requeridos.

### **Actor principal**: Administrador.

### **Precondiciones**: El administrador debe haber iniciado sesión.

### **Flujo principal**:

### El administrador selecciona "Gestión de Furgones" en el menú principal.

### Introduce los datos del furgón (placa, modelo, capacidad).

### Guarda los datos.

### **Resultado esperado**: El furgón se registra correctamente y aparece en la lista de furgones disponibles.

### **Postcondiciones**: La información se almacena en la base de datos.

### 

### **UC-05: Seguimiento de Ubicación**

### **Descripción**: Los usuarios deben poder visualizar la ubicación en tiempo real de los furgones escolares.

### **Actor principal**: Usuario registrado.

### **Precondiciones**: El usuario debe haber iniciado sesión.

### **Flujo principal**:

### El usuario selecciona "Seguimiento de Ubicación".

### El sistema muestra un mapa en tiempo real.

### El usuario selecciona un furgón específico para ver su ubicación actual.

### **Resultado esperado**: La ubicación actual del furgón seleccionado se muestra en el mapa en tiempo real.

### **Postcondiciones**: Los datos de ubicación se actualizan constantemente mientras el furgón esté en movimiento.

### 

### **UC-06: Gestión de Conductores**

### **Descripción**: El administrador debe poder registrar, editar y eliminar conductores del sistema.

### **Actor principal**: Administrador.

### **Precondiciones**: El administrador debe haber iniciado sesión.

### **Flujo principal**:

### El administrador selecciona "Gestión de Conductores".

### Introduce los datos del conductor (nombre, identificación, teléfono).

### Guarda los datos.

### **Resultado esperado**: Los datos del conductor se registran o actualizan correctamente.

### **Postcondiciones**: La información del conductor queda reflejada en la base de datos.

### **UC-09: Configuración de Roles**

### **Descripción**: El administrador debe poder asignar roles (por ejemplo, administrador o conductor) a los usuarios registrados.

### **Actor principal**: Administrador.

### **Precondiciones**: El administrador debe haber iniciado sesión.

### **Flujo principal**:

### El administrador accede a la configuración de usuarios.

### Selecciona un usuario registrado.

### Asigna o modifica el rol del usuario.

### **Resultado esperado**: El rol del usuario se actualiza correctamente.

### 

### **UC-10: Cierre de Sesión**

### **Descripción**: El usuario debe poder cerrar sesión de manera segura.

### **Actor principal**: Usuario registrado.

### **Precondiciones**: El usuario debe haber iniciado sesión.

### **Flujo principal**:

### El usuario selecciona la opción "Cerrar Sesión".

### El sistema elimina la sesión activa.

### **Resultado esperado**: El usuario es redirigido a la pantalla de inicio de sesión.

### 

### **Requerimientos Funcionales**

#### **Componentes Comunes**

### Estos son módulos o funcionalidades básicas que comparten tanto la plataforma web como la aplicación móvil:

### **Autenticación y Roles**:

### Registro de usuarios con validación de correo y contraseña.

### Inicio de sesión y cierre de sesión.

### Recuperación de contraseña.

### Configuración de roles para usuarios (administrador, conductor, usuario regular).

### **Notificaciones**:

### Envío de notificaciones en tiempo real sobre cambios en el estado de los furgones.

### Mensajes de error claros y específicos para acciones fallidas.

### **Interfaz de Usuario (UI)**:

### Diseño responsivo para dispositivos móviles y web.

### Navegación intuitiva entre pantallas y menús.

### Accesibilidad para cumplir con estándares básicos de diseño inclusivo.

### **Base de Datos**:

### CRUD para conductores y furgones.

### Encriptación de contraseñas y datos sensibles.

### Optimización para soportar grandes volúmenes de datos (más de 5000 registros).

### 

#### **Componente 1: Gestión de Conductores**

### **Descripción**: Permite administrar los datos de los conductores en la plataforma.

### **Funcionalidades**:

### Registro de nuevos conductores, incluyendo nombre, identificación, y datos de contacto.

### Edición de información existente.

### Eliminación de conductores inactivos.

### Búsqueda y filtrado de conductores por criterios (nombre, ID).

### **Flujo Principal**:

### El administrador selecciona "Gestión de Conductores".

### Introduce o edita la información requerida.

### Confirma la operación y visualiza la lista actualizada.

### **Restricciones**:

### Solo administradores pueden realizar modificaciones.

### 

#### **Componente 2: Seguimiento de Furgones**

### **Descripción**: Proporciona información en tiempo real sobre la ubicación de los furgones escolares.

### **Funcionalidades**:

### Visualización de un mapa interactivo que muestra la posición de cada furgón.

### Actualización en tiempo real de la ubicación.

### Cambio del estado del furgón (en ruta, detenido).

### Envío de notificaciones automáticas sobre eventos relevantes.

### **Flujo Principal**:

### El usuario selecciona "Seguimiento de Furgones".

### El sistema carga el mapa con las ubicaciones actuales.

### El usuario selecciona un furgón para obtener más detalles.

### **Restricciones**:

### Solo usuarios registrados pueden acceder a esta funcionalidad.

### Los datos de ubicación deben estar actualizados en intervalos no mayores a 10 segundos.

### 

## Requerimientos No-Funcionales (Matriz con Input/Output, solo si aplica)

### Componente 1: Gestión de Conductores

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Registro de Conductores | Datos del conductor: nombre, identificación, teléfono, correo electrónico. | Registro exitoso: "Conductor registrado correctamente." |
| 2 | Edición de Conductores | ID del conductor a editar, nuevos datos (opcional: nombre, identificación, teléfono, correo). | Actualización exitosa: "Información del conductor actualizada." |
| 3 | Eliminación de Conductores | ID del conductor a eliminar. | Eliminación exitosa: "Conductor eliminado." |
| 4 | Búsqueda de Conductores | Criterios de búsqueda: nombre, ID, u otros. | Lista de conductores que coinciden con los criterios de búsqueda. |

### Componente 2 (puede tener uno o más casos de uso)

| **Nro** | **Caso de Uso** | **Input** | **Output** |
| --- | --- | --- | --- |
| 1 | Visualización de Ubicación | Identificador único del furgón, coordenadas GPS obtenidas automáticamente por el sistema en tiempo real. | Mapa interactivo con la ubicación actual del furgón |
| 2 | Cambio de Estado | ID del furgón, nuevo estado (en ruta, detenido, mantenimiento) | Notificación enviada a los usuarios relacionados: "El furgón X ha cambiado su estado a [nuevo estado]." |
| 3 | Actualización en Tiempo Real | Coordenadas GPS recibidas a intervalos regulares (por ejemplo, cada 10 segundos). | Actualización del marcador en el mapa con la nueva posición del furgón. |
| 4 | Envío de Notificaciones | Eventos detectados automáticamente (por ejemplo, inicio de ruta, parada inesperada). | Notificaciones automáticas enviadas a los usuarios relacionados con el furgón afectado. |

# Estrategia de Pruebas

La estrategia de testing para el proyecto Seguro a Casa (SAC) está diseñada para garantizar que los requerimientos funcionales y no funcionales sean validados exhaustivamente, minimizando riesgos y garantizando la calidad del producto. Se realizan varios tipos de pruebas, seleccionados en función de los objetivos del sistema y la criticidad de sus componentes.

## Tipos de pruebas

### **Pruebas funcionales**

Las pruebas funcionales se centran en garantizar que todas las funcionalidades descritas en los requerimientos sean implementadas correctamente y cumplan con las expectativas del usuario final. Se basan en la técnica de **caja negra**, evaluando el sistema desde la perspectiva del usuario sin considerar su implementación interna.

| Objetivo de la prueba: | Validar que todas las funcionalidades operan según los requerimientos, asegurando la correcta ejecución de los **casos de uso** previamente definidos:   * Registro e inicio de sesión. * Gestión de conductores y furgones (CRUD). * Seguimiento de ubicación en tiempo real. * Configuración de roles y permisos.   Confirmar que los datos ingresados en el sistema son procesados correctamente, generando resultados esperados y manejando adecuadamente los errores.  Verificar que la navegación y las interacciones en la interfaz gráfica de usuario (GUI) reflejan fielmente las reglas del negocio.  Asegurar que las ventanas, campos, menús y objetos de la aplicación cumplen con los estándares de diseño y usabilidad. |
| --- | --- |
| Técnica a utilizar: | Ejecución de Casos de Uso:   * Analizar y ejecutar flujos principales y alternativos para todos los casos de uso descritos. * Usar datos válidos e inválidos para evaluar respuestas del sistema y manejo de errores.   Validaciones Específicas:   * Resultados esperados: Confirmar que el sistema genera salidas correctas con entradas válidas. * Mensajes de error: Asegurarse de que los errores son específicos y claros ante entradas inválidas. * Reglas de negocio: Validar que se aplican correctamente en cada flujo.   Procedimientos de Prueba:   * Diseñar y ejecutar procedimientos para cada pantalla, incluyendo: * Verificación de campos obligatorios. * Transiciones entre ventanas. * Estados de los objetos (habilitado, deshabilitado, foco, etc.). * Comprobar la consistencia del diseño en todo el sistema |
| Criterio de validación: | * **Ejecución Completa**:   + Todas las pruebas deben completarse sin interrupciones.   + Todos los casos de prueba deben ejecutarse según lo planificado. * **Asignación de Defectos**:   + Todos los defectos detectados deben documentarse y asignarse a los responsables. * **Conformidad con Estándares**:   + Cada ventana y funcionalidad debe cumplir con los estándares establecidos de usabilidad, accesibilidad y consistencia. |
| Consideraciones especiales: | No se probarán todas las propiedades de los objetos de terceros; se priorizaron las que impactan directamente la funcionalidad y experiencia del usuario.  Se debe registrar cualquier discrepancia entre los resultados esperados y los obtenidos para su posterior análisis y resolución. |
| Observaciones: | Las pruebas funcionales se realizan en un entorno de pruebas controlado que simula condiciones reales.  Se incluirán validaciones específicas para dispositivos móviles y navegadores web, garantizando una experiencia homogénea en todas las plataformas soportadas. |

### **Pruebas de rendimiento (Performance)**

El objetivo de las pruebas de rendimiento es garantizar que la aplicación **Seguro a Casa (SAC)** pueda manejar cargas de trabajo normales y de estrés sin degradación significativa en su desempeño. Estas pruebas están diseñadas para evaluar tiempos de respuesta, tasa de transacciones y la estabilidad general bajo diferentes condiciones.

| Objetivo de la prueba: | Validar que el sistema pueda soportar hasta 1000 usuarios concurrentes sin afectación significativa en el rendimiento.  Medir tiempos de respuesta para operaciones críticas como:   * Registro e inicio de sesión. * Actualización de datos en tiempo real (seguimiento de furgones). * Acceso y gestión de conductores y furgones.   Identificar posibles cuellos de botella y evaluar la capacidad del sistema para escalar según los requerimientos de uso. |
| --- | --- |
| Técnica a utilizar: | **Cargas Simuladas**:   * Usar herramientas como JMeter o Locust para simular usuarios concurrentes. * Ejecutar scripts que reproduzcan las operaciones más frecuentes del sistema.   **Escenarios de Prueba**:   * **Carga normal**: Simulación de 100 a 500 usuarios concurrentes realizando operaciones comunes. * **Sobrecarga**: Incremento progresivo hasta 1000 usuarios para evaluar el límite del sistema. * **Pruebas prolongadas**: Operaciones continuas durante un periodo extendido para verificar estabilidad. |
| Criterio de validación: | **Tiempos de Respuesta**:   * Operaciones críticas deben completarse en menos de 2 segundos bajo carga normal. * En sobrecarga, los tiempos no deben exceder los 5 segundos.   **Errores**:   * El porcentaje de errores no debe superar el 1% en ningún escenario.   **Uso de Recursos**:   * El consumo de CPU, memoria y disco debe mantenerse dentro de los límites aceptables para el entorno configurado. |
| Consideraciones especiales: | Las pruebas deben realizarse en un entorno que replique las condiciones de producción.  Se deben documentar métricas de uso de recursos, tiempos de respuesta y errores para cada escenario. |
| Observaciones: | Las pruebas funcionales se realizan en un entorno de pruebas controlado que simula condiciones reales.  Se incluirán validaciones específicas para dispositivos móviles y navegadores web, garantizando una experiencia homogénea en todas las plataformas soportadas. |

### 

### **Pruebas de seguridad y de acceso a datos**

Las pruebas de seguridad se enfocan en proteger la aplicación contra accesos no autorizados y garantizar la integridad de los datos sensibles. Este tipo de pruebas es crucial para sistemas que manejan información personal de usuarios, como contraseñas y datos de ubicación.

| Objetivo de la prueba: | Validar que solo los usuarios autorizados puedan acceder a funcionalidades específicas según sus roles.  Asegurar que la información confidencial (como contraseñas) esté protegida mediante encriptación.  Comprobar que las sesiones expiren tras un periodo de inactividad, previniendo accesos no autorizados. |
| --- | --- |
| Técnica a utilizar: | **Pruebas de Nivel de Aplicación**:   * Crear usuarios con diferentes roles y verificar que solo puedan acceder a funcionalidades asignadas. * Intentar acceder a funciones restringidas y comprobar que el sistema lo impida.   **Pruebas de Nivel de Sistema**:   * Simular intentos de inicio de sesión con credenciales incorrectas y comprobar que se limite el número de intentos fallidos. * Validar que las contraseñas almacenadas estén encriptadas en la base de datos. |
| Criterio de validación: | **Accesos Permitidos y Denegados:**   * Los usuarios deben acceder únicamente a las funcionalidades y datos que correspondan a su rol.   **Encriptación de Datos:**   * Las contraseñas deben estar cifradas y no ser legibles en texto plano.   **Autenticación:**   * El sistema debe bloquear cuentas tras múltiples intentos fallidos de inicio de sesión. * Las sesiones deben expirar automáticamente tras un periodo definido de inactividad. |
| Consideraciones especiales: | Las pruebas deben incluir escenarios reales y simulados para evaluar diferentes niveles de acceso.  La colaboración con el administrador de redes puede ser necesaria para garantizar que las configuraciones de seguridad sean consistentes. |

### 

### Herramientas involucradas

A continuación, se listan las herramientas que serán utilizadas para realizar las pruebas planificadas en el proyecto Seguro a Casa (SAC). Estas herramientas están seleccionadas en función de los objetivos de testing funcional, de rendimiento y de seguridad descritos anteriormente.

|  | **Herramienta** | **Vendedor/Tienda** | **Versión** |
| --- | --- | --- | --- |
| *Testing de rendimiento y carga.* | *JMeter* | *Apache Software Foundation* | *5.5* |
|  |  |  |  |
| *Debugging y testing del frontend y backend.* | *Visual Studio Code* | *Microsoft* | *1.83.0* |
| *Simulación de usuarios concurrentes* | *Locust* | *Open Source* | *2.16.3* |
| *Identificación de vulnerabilidades de seguridad.* | *OWASP ZAP* | *OWASP* | *2.13.0* |

# Recursos

## Profesionales

| **Recursos Humanos** | | |
| --- | --- | --- |
| **Rol** | **Recursos mínimos recomendados**  **(Número de personas full-time)** | **Responsabilidades específicas / Comentarios** |
| Diseñador de casos de prueba | 1 persona | Responsabilidades   * Identificar, priorizar e implementar casos de prueba. * Evaluar el esfuerzo necesario para cada prueba. * Diseñar los flujos principales y alternativos en función de los requerimientos. |
| Testeador | 1 persona | Responsabilidades:   * Ejecutar los casos de prueba diseñados. * Registrar los resultados y documentar errores. * Generar peticiones de cambios en la documentación y coordinar la corrección de defectos. |
| Administrador de sistema del pruebas | 1 persona | Responsabilidades   * Configurar y administrar el entorno de pruebas. * Gestionar herramientas y acceso a sistemas. * Asegurar la consistencia y disponibilidad de los datos en el entorno de pruebas. |

### Integración del ambiente de pruebas y configuración

La integración del ambiente de pruebas y su configuración son esenciales para asegurar que las pruebas se ejecuten en un entorno controlado, consistente y representativo del entorno de producción. Esto garantiza que los resultados obtenidos sean fiables y que el sistema cumpla con los requisitos definidos.

Las actividades generales a ser consideradas son:

| **Actividad** | **Responsable** | **Observaciones** |
| --- | --- | --- |
| Configuración inicial de servidores y red | Administrador de la Red | Garantizar conectividad segura y configuración adecuada para herramientas de prueba. |
| Instalación y configuración de herramientas | Administrador del Sistema | Implementar JMeter, Selenium, Postman y otras herramientas requeridas. |
| Implementación de la aplicación en el entorno | Administrador del Sistema | Subir versiones del frontend y backend al entorno de pruebas. |
| Generación de datos representativos | Administrador de Bases de Datos | Cargar datos relevantes para pruebas funcionales y de rendimiento. |

### Generación de datos

La generación de datos para las pruebas debe asegurar que cubra todos los casos de uso del sistema, tanto funcionales como no funcionales. Esto incluye la creación de datos válidos, inválidos, límites y negativos, además de asegurar que los datos sean representativos, variados y realistas, simulando diferentes perfiles y escenarios de usuarios. También se debe garantizar la integridad y consistencia de los datos, así como la seguridad, mediante la anonimización cuando sea necesario. Es clave que los datos sean adecuados para pruebas de rendimiento, con grandes volúmenes y distribuidos equitativamente, y que se mantengan actualizados conforme evolucionan los requisitos del sistema. La automatización de la generación de datos y su validación continua son esenciales para asegurar su calidad y cobertura.

#### Muestra de producción

Para que la muestra de datos sea realmente representativa, se deberá elegir una fecha testigo adecuada y que posibilite la mayor cobertura de datos. En este sentido se ha elegido como fecha testigo el

Para la obtención de datos por esta vía, se deberán definir las restricciones (por motivos de confidencialidad) y generar algún utilitario para filtrar los datos de tal forma de obtener la mayor variabilidad de datos posible.

# Actividades e Hitos del Plan de Pruebas

|  | **Tarea** | **Responsable** | **Esfuerzo** | **Fecha Inicio** | **Fecha Término** |
| --- | --- | --- | --- | --- | --- |
|  | **Plan de Pruebas** |  |  |  |  |
|  | Identificar el Proyecto | Líder del Proyecto | 1 día | 2024-10-09 | 2024-10-09 |
|  | Definir Estrategia | Analista de QA | 2 días | 2024-10-10 | 2024-10-11 |
|  | Estimar Actividades | Líder del Proyecto, QA | 1 día | 2024-10-12 | 2024-10-12 |
|  | Identificar Recursos | Líder del Proyecto | 1 día | 2024-10-13 | 2024-10-13 |
|  | Documentar el “Plan de Pruebas” | Analista de QA | 3 días | 2024-10-14 | 2024-10-16 |
|  | Agendar de Actividades | Líder del Proyecto | 1 día | 2024-10-17 | 2024-10-17 |
|  | Revisar el “Plan de Pruebas” | Líder del Proyecto, QA | 2 días | 2024-10-18 | 2024-10-19 |
|  | **Diseño de las Pruebas** |  |  |  |  |
|  | Analizar Requerimientos | Analista de QA | 2 días | 2024-10-20 | 2024-10-21 |
|  | Especificar Procedimientos de prueba | Diseñador de Pruebas | 3 días | 2024-10-22 | 2024-10-24 |
|  | Especificar casos de prueba | Diseñador de Pruebas | 4 días | 2024-10-25 | 2024-10-28 |
|  | Revisar Cobertura de los requerimientos de prueba | QA y Equipo de Desarrollo | 2 días | 2024-10-29 | 2024-10-30 |
|  | **Implementación de las Pruebas** |  |  |  |  |
|  | Establecer Ambiente de Implementación | Administrador de Sistemas | 2 días | 2024-10-31 | 2024-11-01 |
|  | Desarrollar los Procedimientos de Prueba | Diseñador de Pruebas | 2 días | 2024-11-02 | 2024-11-03 |
|  | Probar y depurar los procedimientos de prueba | Tester | 2 días | 2024-11-04 | 2024-11-05 |
|  | Modificar los procedimientos de prueba | Tester y QA | 1 días | 2024-11-06 | 2024-11-06 |
|  |  |  |  |  |  |
|  | **Ejecución de las pruebas** |  |  |  |  |
|  | Ejecutar pruebas | Tester | 4 días | 2024-11-07 | 2024-11-10 |
|  | Comprobar resultados esperados | Tester y QA | 1 días | 2024-11-11 | 2024-11-11 |
|  | Investigar resultados inesperados | Tester y QA | 2 días | 2024-11-12 | 2024-11-13 |
|  | Registrar defectos (log) | Tester | continuo | 2024-11-07 |  |
|  | Re-Ejecutar las pruebas | Tester | 2 días | 2024-11-14 | 2024-11-15 |
|  | **Evaluación de las pruebas** |  |  |  |  |
|  | Revisar el Log de pruebas | QA | 1 día | 2024-11-16 | 2024-11-16 |
|  | Evaluar cobertura de los casos de prueba | QA | 1 día | 2024-11-16 | 2024-11-16 |
|  | Evaluar defectos | QA y Desarrollo | 1 día | 2024-11-16 | 2024-11-16 |
|  | Reportar defectos | QA | 1 día | 2024-11-16 | 2024-11-16 |

# Entregables

Documentos:

* Plan de Pruebas: Documento detallado con la estrategia, casos de prueba, cronograma, y recursos asignados.
  + Autor: Analista de QA.
  + Destinatario: Líder del Proyecto y equipo de desarrollo.
  + Fecha de Entrega: 16/10/2024.
* Planilla de Casos de Prueba: Contiene el diseño de los casos de prueba, con inputs, outputs esperados, y pasos a seguir.
  + Autor: Diseñador de Pruebas.
  + Destinatario: Tester y QA.
  + Fecha de Entrega: 30/10/2024.

Herramientas**:**

* **JMeter:** Para pruebas de rendimiento.
* **OWASP ZAP:** Para identificación de vulnerabilidades de seguridad.
* **Visual Studio Code:** Para debugging y testing en frontend y backend.
* **Locust:** Para simular usuarios concurrentes.

## Plan de pruebas

**Criterio de Entrada para el “Plan de Pruebas”**

* Documentación de requisitos aprobada y revisada por el equipo.
* Disponibilidad del entorno de pruebas configurado correctamente.
* Datos de prueba generados y validados.
* Herramientas de pruebas instaladas y funcionales.
* Disponibilidad de los recursos humanos asignados (QA, Tester, Administrador del Sistema).

**Criterio de Salida para el “Plan de Pruebas”**

* Ejecución completa de todos los casos de prueba diseñados, incluyendo flujos principales y alternativos.
* Todos los defectos críticos y de alta prioridad deben haber sido resueltos y validados.
* Cobertura del 100% de los requerimientos funcionales y no funcionales.
* Elaboración y aprobación del sumario de evaluación de pruebas.

**Criterio de Suspensión y Resumisión**

**Suspensión:**

* Fallos críticos en el entorno de pruebas que impidan la ejecución (por ejemplo, indisponibilidad de servidores).
* Ausencia de datos necesarios para las pruebas.
* Identificación de defectos bloqueantes que requieran solución inmediata antes de continuar.

1. **Resumisión:**

* Corrección de los defectos bloqueantes y validación de los mismos.
* Restauración del entorno de pruebas al estado funcional.
* Disponibilidad de los recursos necesarios para continuar la ejecución.

## Resultados de las pruebas

#### **1. Pruebas Funcionales**

**Objetivo:** Validar que las funcionalidades principales del sistema operan correctamente según los requerimientos.

1. **Prueba: Registro de Usuarios**
   1. **Resultado Esperado:**Los usuarios deben registrarse correctamente cuando ingresan datos válidos. Se deben mostrar mensajes de error claros para entradas inválidas (correo ya registrado, contraseñas no coincidentes, etc.).
   2. **Resultado Obtenido:**El sistema permitió el registro exitoso para entradas válidas y generó mensajes de error adecuados para datos incorrectos.
2. **Prueba: Inicio de Sesión**
   1. **Resultado Esperado:**Los usuarios pueden iniciar sesión con credenciales válidas. Se deben bloquear intentos fallidos repetidos y mostrar mensajes de error claros.
   2. **Resultado Obtenido:**Inicio de sesión exitoso para credenciales válidas. Se bloqueó la cuenta después de 5 intentos fallidos, mostrando un mensaje adecuado.
3. **Prueba: Gestión de Conductores y Furgones (CRUD)**
   1. **Resultado Esperado:**Se permite crear, leer, actualizar y eliminar registros correctamente. Los datos deben reflejarse inmediatamente tras cada operación.
   2. **Resultado Obtenido:**Todas las operaciones CRUD funcionaron correctamente y los datos se reflejaron en tiempo real.
4. **Prueba: Seguimiento de Ubicación**
   1. **Resultado Esperado:**La ubicación en tiempo real de los furgones debe actualizarse cada 10 segundos en el mapa interactivo.
   2. **Resultado Obtenido:**El sistema actualizó la ubicación en tiempo real con un margen de error de menos de 1 segundo.

#### **2. Pruebas de Rendimiento**

**Objetivo:** Evaluar la capacidad del sistema bajo diferentes niveles de carga.

1. **Prueba: Carga Normal (500 usuarios concurrentes)**
   * **Resultado Esperado:**Operaciones clave (registro, inicio de sesión, seguimiento en tiempo real) deben completarse en menos de 2 segundos.
   * **Resultado Obtenido:**Tiempos de respuesta promedio: 1.7 segundos. Todos los usuarios experimentaron fluidez en las operaciones.
2. **Prueba: Sobrecarga (1000 usuarios concurrentes)**
   * **Resultado Esperado:**El sistema debe soportar la carga sin errores críticos. Los tiempos de respuesta no deben exceder los 5 segundos.
   * **Resultado Obtenido:**Tiempos de respuesta promedio: 4.3 segundos. No se registraron errores críticos.

#### **3. Pruebas de Seguridad**

**Objetivo:** Garantizar el control de acceso y protección de datos sensibles.

1. **Prueba: Roles y Permisos**
   * **Resultado Esperado:**Los usuarios solo deben acceder a funcionalidades específicas según su rol (administrador, conductor, usuario).
   * **Resultado Obtenido:**Todos los accesos estuvieron correctamente restringidos según los roles asignados.
2. **Prueba: Encriptación de Contraseñas**
   * **Resultado Esperado:**Las contraseñas deben almacenarse encriptadas y no ser legibles en texto plano.
   * **Resultado Obtenido:**Validación exitosa. Todas las contraseñas se almacenan utilizando un algoritmo de encriptación seguro (bcrypt).
3. **Prueba: Expiración de Sesión**
   * **Resultado Esperado:**Las sesiones deben expirar automáticamente tras 15 minutos de inactividad.
   * **Resultado Obtenido:**Las sesiones expiraron correctamente tras el periodo definido.