**Plan de Pruebas**

**Caso: Autodoc**

**Fecha: [12/11/2024]**

**Asignatura: Capstone**

**Integrantes:**

**Sebastián Aróstica**

**Bastián Espinosa**

**Diego Morán**

**Fecha de entrega: 22/11/2024**

**Profesor: Arturo Vargas**

**HISTORIAL DE MODIFICACIONES**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 12-11-2024 | 1.0 | Se empieza el documento | Diego Morán |
| **Fecha** | **Versión** | **Descripción** | **Autor** |
|  |  |  |  |

**ÍNDICE**

[**1. INTRODUCCIÓN 4**](#_heading=h.gjdgxs)

[**1.1 Resumen ejecutivo** 4](#_heading=h.fjhdonjhmbvf)

[**1.2 Alcance de las pruebas 5**](#_heading=h.30j0zll)

[**1.2.1 Elementos de pruebas 5**](#_heading=h.1fob9te)

[**1.2.2 Pruebas Funcionales 5**](#_heading=h.3znysh7)

[**1.2.3 Riesgos 5**](#_heading=h.2et92p0)

[**1.2.3.1 Matrices de riesgos 6**](#_heading=h.tyjcwt)

[**1.2.3.1.1 Pruebas 6**](#_heading=h.3dy6vkm)

[**1.3 Tipos de pruebas 7**](#_heading=h.1t3h5sf)

[**1.3.1 Pruebas Funcionales 7**](#_heading=h.4d34og8)

[**1.3.2 Herramientas involucradas 8**](#_heading=h.2s8eyo1)

[**1.3.3 Preparación del ambiente de pruebas 8**](#_heading=h.17dp8vu)

[**1.3.4 Diseño del ambiente de pruebas 8**](#_heading=h.3rdcrjn)

[**1.3.5 Diseño ambiente de pruebas 11**](#_heading=h.26in1rg)

[**2. Resultados de las pruebas 12**](#_heading=h.lnxbz9)

# INTRODUCCIÓN

## Resumen ejecutivo

En el presente documento se dará en detalle todo lo relacionado al plan de pruebas que está asociado al proyecto “Autodoc”. Este documento es necesario para atender los objetivos de calidad en el desarrollo de software, donde se definen aquellos aspectos como los módulos o funcionalidades sujetas a la verificación, tipos de pruebas, entornos, recursos asignados, entre otros.

En el plan de pruebas del proyecto “Autodoc” el cual consta de una aplicación móvil desarrollada para facilitar la gestión y cuidado de vehículos personales, mediante un asistente virtual que permitirá realizar consultas sobre el vehículo registrado y los documentos asociados a este, almacenados de forma manual en la aplicación, por el mismo usuario, tomando esto en cuenta, los principales actores dentro del sistema son:

* Conductor (registrado en APP)
* Administrador del sistema

Teniendo esto en cuenta, se considerarán los siguientes puntos que tendrán un enfoque principal en el ambiente de pruebas:

* Módulo Inicio de Sesión
  + Iniciar sesión
  + Registrar usuario
  + Recuperar contraseña
* Módulo CRUD Vehículo
  + Ingresar datos de vehiculo
  + Almacenar documentos:
    - Permiso de circulación
    - SOAP
    - Revisión técnica
  + Modificar Vehículo
  + Eliminar Vehículo
* Módulo Visualizar Perfil de Vehículo
  + Ver permiso de circulación
  + Ver SOAP
  + Ver Revisión técnica
  + Abrir chatbot
* Módulo de Chatbot
  + Realizar consultas
  + Historial de chats

## Alcance de las pruebas

Este plan de pruebas está diseñado para verificar y validar las funcionalidades y estabilidad de la aplicación **Autodoc**, desarrollada en **React Native**. Los sistemas y módulos que serán evaluados incluyen funcionalidades principales y servicios críticos que aseguran la experiencia del usuario y el cumplimiento de los objetivos de la aplicación.

### Elementos de pruebas

Dentro de los sistemas que serán evaluados en el plan de pruebas se encuentran los siguientes módulos de la aplicación:

1. **Login y Registro de Usuarios**: Pruebas relacionadas con la autenticación, recuperación de contraseñas y creación de cuentas.
2. **Gestión de Vehículos**: Creación, edición y eliminación de registros vehiculares en la app.
3. **Carga y Visualización de Documentos**: Validación de la funcionalidad para subir, descargar y visualizar documentos relacionados con el vehículo, como la revisión técnica o el SOAP.
4. **Notificaciones**: Correcto funcionamiento de notificaciones push para alertas de vencimientos o eventos relevantes.
5. **Pruebas de Integración con Backend**: Validación de la interacción entre la aplicación y el servidor para el envío y recepción de datos.
6. **Optimización de UI/UX**: Comprobación de que la interfaz sea intuitiva y funcional en diferentes dispositivos.

### Pruebas Funcionales

Para la validación funcional, se empleará el método de **testing de Caja Negra**, enfocándonos en la interacción del usuario con la aplicación sin considerar la estructura interna del código. Esto incluye:

* **Errores de interfaz**: Botones, enlaces y formularios que no funcionen como se espera.
* **Errores de flujo**: Funciones que interrumpan el flujo previsto, como problemas al completar el proceso de registro.
* **Errores de inicio/terminación**: Dificultades al abrir o cerrar la aplicación.
* **Pruebas de compatibilidad**: Asegurarse de que la app funciona correctamente en dispositivos Android e iOS de diferentes versiones y tamaños de pantalla.

### Riesgos

A continuación, se identifican los riesgos que pueden surgir durante las pruebas:

1. **Interrupción en la interacción entre frontend y backend** (Gravedad: Alta): Fallos en la conexión al servidor pueden impedir la carga de datos críticos.
2. **Problemas de rendimiento en dispositivos antiguos** (Gravedad: Media): Algunos dispositivos pueden no soportar completamente la aplicación, afectando la experiencia del usuario.
3. **Errores en notificaciones push** (Gravedad: Media): Alertas incorrectas o retrasadas pueden reducir la confiabilidad de la app.
4. **Vulnerabilidad a intentos de intrusión** (Gravedad: Alta): Ataques como denegación de servicio (DDoS) o intentos de acceso no autorizado al backend.

#### Matrices de riesgos

##### Pruebas

| **Número** | **Riesgo** | **Gravedad** | **Plan de mitigación** |
| --- | --- | --- | --- |
| 1 | El servidor backend sufre una caída o falla en la conexión | Alta | Implementar monitoreo constante del servidor, realizar pruebas de carga y tener un plan de contingencia con respaldo de servidores. |
| 2 | El usuario no puede iniciar sesión o recuperar su cuenta | Alta | Revisar los módulos de autenticación y recuperación de contraseñas. Implementar un sistema de verificación de acceso. |
| 3 | Los documentos no se cargan, descargan o visualizan correctamente | Alta | Verificar los módulos de carga y descarga de documentos, realizar pruebas de integración con el backend. |
| 4 | La aplicación no responde de manera óptima en dispositivos con recursos limitados | Media | Realizar pruebas de compatibilidad en diferentes dispositivos y optimizar el rendimiento en dispositivos de gama baja. |
| 5 | Error en las notificaciones push (retrasos o fallas) | Media | Monitorear los servicios de notificaciones, realizar pruebas en diferentes redes y garantizar una integración eficiente. |
| 6 | La aplicación es vulnerable a ataques de Denegación de Servicio (DDoS) o accesos no autorizados | Alta | Implementar medidas de seguridad, como firewalls y limitación de tráfico en el servidor, y realizar pruebas de penetración para detectar vulnerabilidades. |
| 7 | Los datos del vehículo no se registran o se pierden al cargarlos | Alta | Implementar procesos de validación de datos antes de ser almacenados en el backend, con copias de seguridad periódicas. |
| 8 | Fallos en la visualización de la interfaz de usuario (UI) | Baja | Realizar pruebas de usabilidad y asegurarse de que la interfaz sea consistente en diferentes dispositivos. |
| 9 | La aplicación presenta errores en la integración con otras plataformas (API externas) | Alta | Validar la integración de la app con APIs externas y garantizar la comunicación correcta entre sistemas. |
| 10 | La app no se actualiza correctamente en la tienda (App Store/Play Store) | Baja | Revisar los procesos de compilación y publicación de la app, garantizar la compatibilidad con la última versión de los dispositivos. |

## Tipos de pruebas

### Pruebas Funcionales

El testing funcional se realizará sobre los requerimientos funcionales y los casos de uso de la aplicación Autodoc. Estas pruebas tienen como finalidad comprobar que las funcionalidades principales, como el diagnóstico del vehículo y la gestión de documentos, operen conforme a las especificaciones establecidas.

* **Objetivo de la prueba:**  
  Validar que la aplicación cumpla con las necesidades de los usuarios reflejando correctamente las reglas del negocio y los requerimientos funcionales. Esto incluye validar las transacciones entre interfaces, el manejo de errores y el flujo correcto de cada funcionalidad, como el escaneo de datos y la visualización de documentos digitales.
* **Técnica a utilizar:**  
  Ejecutar cada caso de uso y su funcionalidad con datos válidos e inválidos para verificar:
* Que los resultados esperados ocurren cuando se utilicen datos válidos.
* Que los mensajes de error sean claros y correctos al usar datos inválidos, asegurando una buena experiencia de usuario.
* **Criterio de validación:**
* Todas las pruebas planificadas deben ejecutarse satisfactoriamente.
* Los defectos identificados durante las pruebas han sido registrados y asignados para su resolución.

### Herramientas involucradas

A continuación, se presentarán las diferentes herramientas que se utilizaron para probar las distintas funcionalidades que tiene el programa.

|  | **Herramienta** | **Versión** |
| --- | --- | --- |
| **Framework** | **React Native** |  |
| **Servicio de almacenamiento y autenticación** | **Firebase** |  |

### Preparación del ambiente de pruebas

El ambiente de pruebas se configuró para simular condiciones reales de uso de la aplicación, asegurando que las pruebas reflejen con precisión la experiencia del usuario final.

* **Configuración del Frontend y Backend:**Se utilizó React Native para ejecutar y probar las funcionalidades de la interfaz móvil.
* **Conexión al Servidor y Base de Datos:**La base de datos fue realizada con Firebase para sincronizar la información del usuario y los diagnósticos en tiempo real. Se realizaron configuraciones para garantizar una comunicación estable y segura entre el cliente y el servidor.

### Diseño del ambiente de pruebas

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| **HARDWARE** |  |  |
| **Estación de Pruebas (Administrador)** |  |  |
| Procesador | Intel Core i5-8300H CPU @ 2.30GHz 2.30 GHz |  |
| Memoria RAM | 8GB 3200MHz Kingston |  |
| Espacio en Disco | 225 GB |  |
| Tipo Monitor y Resolución | Monitor LG 1080p 22’’ |  |
| Unidad de Disquete | N/A |  |
| Tarjeta de red | Integrada |  |
| Mouse | Logitech G203 |  |
| Tipo Enlace | Canal |  |
| **Servidores** |  |  |
| ***Base de Datos (pruebas)*** |  |  |
| Procesador | Intel Core i5-8300H CPU @ 2.30GHz 2.30 GHz |  |
| Memoria RAM | 8GB 3200MHz Kingston |  |
| Espacio en Disco | 225 GB |  |
| Tipo Monitor y Resolución | Monitor LG 1080p 22’’ |  |
| Unidad de Disquete | N/A |  |
| Tarjeta de red | Integrada |  |
| Mouse | Logitech |  |
| Tipo Enlace | Switch / Hub |  |
| ***Aplicación (pruebas)*** |  |  |
| Procesador | Intel Core i5-8300H CPU @ 2.30GHz 2.30 GHz |  |
| Memoria RAM | 8GB 3200MHz Kingston |  |
| Espacio en Disco | 225 GB |  |
| Tipo Monitor y Resolución | Monitor LG 1080p 22’’ |  |
| Unidad de Disquete | N/A |  |
| Tarjeta de red | Integrada |  |
| Mouse | Logitech |  |
| Tipo Enlace | Switch Hub |  |
| **RED** |  |  |
| Medio | Fibra Óptica |  |
| Velocidad | 20Mb/s |  |
| Protocolo | TCP,UDP |  |
| Conexión Internet | Si |  |
| **SOFTWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Sistema Operativo | Windows 11 |  |
| Browser | Google Chrome |  |
| Software de Escritorio | Microsoft Office |  |
| **Base de Datos (pruebas)** |  |  |
| Sistema Operativo | Windows 11 |  |
| Software de almacenamiento | Firebase |  |
| Dominio/Cuenta | Admin |  |
|  |  |  |
| **Aplicación (pruebas)** |  |  |
| Sistema Operativo | Windows 11 |  |
| Software de almacenamiento | Firebase |  |
| Dominio/Cuenta | Admin |  |
|  |  |  |
| **Repositorios** |  |  |
| Servidor | Admin |  |
| Dominio / Cuenta | Admin |  |

### Diseño ambiente de pruebas

Para el siguiente diagrama se mostrará el ambiente de pruebas requerido para realizar las pruebas. La arquitectura del ambiente:

**Ventajas:**

**Desventajas:**

# Resultados de las pruebas

Durante la ejecución de las pruebas, el método elegido siendo pruebas de caja negra, se mostrará un registro a través de los casos de uso en cada tipo de prueba en el sistema, teniendo los siguientes resultados:

| **ÁREA FUNCIONAL: Autenticar Usuario** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Caso de prueba** | **Prioridad** | **Precondiciones** | **Datos de entrada** | **Pasos** | **Resultados Esperado** |
| 1 | Iniciar sesión dentro del sistema | Alta | El usuario debe ingresar las credenciales ya registradas para entrar en el sistema | Usuario: ba.espinosav@duocuc.cl  Contraseña: Bastian1920 | Login | El usuario logra entrar al sistema correctamente, entrando a la ventana principal del sistema. |
| 2 | Validaciones al momento de Iniciar sesión dentro del sistema | Alta | El usuario debe ingresar con credenciales que no se encuentran registradas en el sistema | Usuario: ba.espinosav@duocuc.cl  Contraseña: Bastian | Login | El sistema deniega la entrada. |

| **ÁREA FUNCIONAL : Pantalla principal administrador** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Caso de prueba** | **Prioridad** | **Precondiciones** | **Datos de entrada** | **Pasos** | **Resultados Esperado** |
| 1 | Iniciar sesión dentro del sistema | Alta | El usuario debe ingresar las credenciales ya registradas para entrar en el sistema | Usuario: ba.espinosav@duocuc.cl  Contraseña: Bastian1920 | Login | El usuario logra entrar al sistema correctamente, entrando a la ventana principal del sistema. |
| 2 | Validaciones al momento de Iniciar sesión dentro del sistema | Alta | El usuario debe ingresar con credenciales que no se encuentran registradas en el sistema | Usuario: ba.espinosav@duocuc.cl  Contraseña: Bastian | Login | El sistema deniega la entrada. |
| 3 | Iniciar sesión dentro del sistema como administrador | Alta | El administrador debe ingresar las credenciales ya registradas para entrar en el sistema | Usuario: admin@admin.cl  Contraseña: adminadmin | Login | Ingresar a la interfaz de administrador |
| 4 | Recuperar cantidad de usuarios | Alta | El administrador deberá hacer click en "Ver Total de Usuarios" para tener una ventana con el total de usuarios | Botón: Ver total | Ver Total de Usuarios | Ventana de Usuarios |
| 5 | Generar reporte de usuarios | Alta | El administrador deberá hacer click en "Descargar PDF" para Generar un reporte PDF con la cantidad de los usuarios | Botón: Descargar PDF | Descargar PDF | Descarga de pdf |
| 6 | Recuperar cantidad de chats por usuario | Alta | El administrador deberá hacer click en "Ver Chats por Usuarios" para tener una ventana con el total de Chats | Botón: Ver Chats por usuarios | Ver Chats por Usuarios | Ventana de Chats |
| 7 | Generar reporte de cantidad de chats por usuario | Alta | El administrador deberá hacer click en "Descargar PDF" para Generar un reporte PDF | Botón: Descargar PDF | Descargar PDF | Descarga de pdf |
| 8 | Recuperar cantidad de vehículos por usuarios | Alta | El administrador deberá hacer click en "Ver Vehiculos por Usuarios" para tener una ventana con el total de Vehículos | Botón: Ver vehículos por usuarios | Ver Vehiculos por Usuarios | Ventana de Vehiculos |
| 9 | Generar reporte de Vehículos | Alta | El administrador deberá hacer click en "Descargar PDF" para Generar un reporte PDF | Botón: Descargar PDF | Descargar PDF | Descarga de pdf |
| 10 | Cerrar sesión | ALTA | El administrador podrá cerrar su sesión de administrador | Botón: Cerrar sesión | Cerrar sesión | Enviar al inicio de la APP |

| **AREA FUNCIONAL : Pantalla principal de usuarios** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Caso de prueba** | **Prioridad** | **Precondiciones** | **Datos de entrada** | **Pasos** | **Resultados Esperado** |
| 11 | Iniciar sesión dentro del sistema | Alta | El usuario debe ingresar las credenciales ya registradas para entrar en el sistema | Usuario: ba.espinosav@duocuc.cl  Contraseña: Bastian1920 | 1. Seleccionar "Inicio de sesión"  2. Ingresar credenciales  3. Seleccionar "Iniciar Sesión" | El usuario logra entrar al sistema correctamente, entrando a la ventana principal del sistema. |
| 12 | Agregar Vehículo | Alta | El usuario podra agregar un vehiculo nuevo | -Marca:  -Modelo:  -Año:  -Patente:  -Seleccionar Permiso de Circulación  -Seleccionar SOAP  -Seleccionar Revisión Técnica | 1. Seleccionar "Agregar Vehículo"  2. Ingresar datos del vehículo  3. Subir archivos del vehículo  4. Seleccionar "Registrar Vehículo" | El sistema guardará en Firebase el vehículo que se acaba de registrar asociando una ID única. |
| 13 | Ver información del vehículo registrado por el usuario | Alta | El usuario deberá hacer click en un vehículo registrado para ver su información | Click en el vehículo | Click | La aplicación mostrara el vehículo previamente registrado por el usuario con los pdf subidos y sus datos registrados |
| 14 | Editar información del vehículo registrado por el usuario | Alta | El usuario deberá hacer click en Editar para actualizar la información | Click en el Editar | Click | La aplicación dejará al usuario actualizar los datos del vehículo registrado |
| 15 | Eliminar Vehículo | Alta | El usuario deberá hacer click en Eliminar para eliminar permanentemente su vehículo | Click en eliminar | Click | La aplicación eliminará toda la información del vehículo que registro el usuario |
| 16 | Iniciar Chat con Gemini | Alta | El usuario deberá tener registrado un vehículo con la información correcta para iniciar un chat con nuestro chatbot | Click en Chatbot | Click | La aplicación dejará iniciar un Chat nuevo o escoger un chat ya existente |
| 17 | Cerrar sesión | ALTA | El usuario podrá cerrar su sesión | Click | Cerrar sesión | Enviar al inicio de la APP |

| **ÁREA FUNCIONAL : Registrar usuario Nuevo** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Caso de prueba** | **Prioridad** | **Precondiciones** | **Datos de entrada** | **Pasos** | **Resultados Esperado** |
| 18 | Registrar usuario nuevo | Alta | Se deberá descargar la aplicación, abrirla y hacer click en Registrarse para poder registrar un usuario nuevo | -Nombre  -Apellido  -Nombre de usuario  -Correo electrónico  -Contraseña  -Confirmar Contraseña  Registrarse | LLenar los campos solicitados y hacer click en Registrarse | Enviar a pantalla de inicio y enviar alerta de usuario registrado |

**2.1 Reporte de defectos**

Por otro lado, al ejecutar las pruebas señaladas anteriormente nos encontramos con los siguientes defectos durante la ejecución:

| **ID** | **ID Caso Prueba** | **Nº Ciclo** | **Fecha** | **Módulo** | **Descripción Detallada** | **Tipo** | **Severidad** | **Estado** | ***Observación de equipo de desarrollo*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |