**Plan de Pruebas**

***EzyRide Project***

**Asignatura: Portafolio de Título**

**Integrantes:**

**Cristóbal Muñoz**

**Claudio Meza**

**Francisco James**

**Fecha de entrega: 18/10/2024**

**Profesor: Arturo Vargas**

**Historial de revisiones**

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 17/10/2024 | <1.0> | Se desarrolla el documento incluyendo una estructura abarcando puntos como tecnologías empleadas, hardware utilizado en pruebas, registro de defectos, etc. | Equipo |

**TABLA DE CONTENIDO**

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

[**1.2**](#_heading=h.30j0zll) **Alcance de las pruebas** 4

[**1.2.1**](#_heading=h.1fob9te) **Elementos de pruebas** 4

[**1.2.2**](#_heading=h.3znysh7) **Pruebas Funcionales** 4

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

[**1.2.3.1**](#_heading=h.tyjcwt) **Matrices de riesgos** 5

[**1.2.3.1.1**](#_heading=h.3dy6vkm) **Pruebas** 5

[**1.3**](#_heading=h.1t3h5sf) **Tipos de pruebas** 5

[**1.3.1**](#_heading=h.4d34og8) **Pruebas Funcionales** 5

[**1.3.2**](#_heading=h.2s8eyo1) **Herramientas involucradas** 5

[**1.3.3**](#_heading=h.17dp8vu) **Preparación del ambiente de pruebas** 6

[**1.3.4**](#_heading=h.3rdcrjn) **Diseño del ambiente de pruebas** 6

[**1.3.5**](#_heading=h.26in1rg) **Diseño ambiente de pruebas** 9

[**2.**](#_heading=h.lnxbz9) **Resultados de las pruebas** 9

[**2.1 Reporte de defectos** 10](#_heading=h.44sinio)

1. **Introducción**
   1. **Resumen ejecutivo**

En el presente documento se detalla el plan de pruebas asociado al proyecto **EzyDrive**, una aplicación diseñada para mejorar la experiencia de los pasajeros y conductores en el transporte privado. Este documento es esencial para asegurar que se cumplan los objetivos de calidad en el desarrollo de software, definiendo los módulos o funcionalidades que serán objeto de verificación, los tipos de pruebas a realizar, los entornos de prueba, los recursos asignados, entre otros aspectos relevantes.

El plan de pruebas del caso **EzyDrive** se centra en la gestión y optimización de reservas de transporte. La aplicación incorpora las siguientes funcionalidades clave:

* **Gestión de Reservas**: Permite a los pasajeros realizar el flujo de la reserva de transporte y a los conductores iniciar sesión, escoger un vehículo y aceptar o rechazar dichas reservas.
* **Notificaciones y Alertas**: Genera alertas en caso de demora o cambios en la situación.
  1. **Alcance de las pruebas**
     1. **Elementos de pruebas**

Dentro de los sistemas que serán que serán usados en el plan de pruebas nos encontramos con los siguientes:

Aplicación Móvil (ionic/Angular): Dentro de la aplicación móvil, serán testeados los Módulos de Login, pantallas conductor y pantallas pasajero.

El proyecto EzyRide involucra la gestión de reservas, usuarios, conductores y vehículos a través de una aplicación móvil. Los elementos clave a probar incluyen:

* **Autenticación y acceso seguro:** Login de conductores y pasajeros mediante email o teléfono.
* **Gestión de reservas:** Asignación, confirmación, y finalización de servicios.
* **Notificaciones en tiempo real:** Notificación de nuevas asignaciones y cambios de estado.
* **Capacidad de vehículos:** Actualización dinámica del espacio disponible en las vans.
* **Interfaces de usuario:** Formularios de login, selección de vehículos, confirmación de encuentros, etc.
  + 1. **Pruebas Funcionales**

Dentro de las pruebas funcionales del caso, se utilizará el método de testing “Caja negra”, Siendo se va a buscar la verificación de las funcionalidades del software sin tomar referente la estructura del código interno, además de que se caracteriza por su utilidad en los errores del interfaz, de rendimiento, de inicio y terminación.

Estas pruebas validan que las funcionalidades del sistema cumplan con los requerimientos establecidos y respondan según lo esperado. Las áreas cubiertas incluyen:

1. **Login y autenticación:** Verificación del acceso del usuario según credenciales y estado del email.
2. **Asignación automática de conductores:** Pruebas de la lógica FIFO (First In, First Out) para asegurar rotación equitativa.
3. **Confirmación de encuentros:** Validación del flujo de interacción entre el conductor y el pasajero.
4. **Actualización del estado de reservas:** Evaluación de los cambios de estado en tiempo real.
5. **Notificaciones:** Prueba de recepción de notificaciones en primer y segundo plano.
   * 1. **Riesgos**

Para poder identificar los riesgos que se pueden hallar dentro del sistema a través del respectivo testing, serán identificados a través del concepto de gravedad (bajo, medio y alto), estos mostraran la importancia de cada caso de uso con el cual se hará las pruebas de testing.

* + - 1. **Matrices de riesgos**
         1. **Pruebas**

A continuación, se detallan los riesgos identificados para el proyecto EzyRide, junto con sus respectivas acciones mitigantes.

| **N°** | **Riesgo** | **Gravedad** | **Acción** |
| --- | --- | --- | --- |
| 1 | Caída del servidor que soporta la aplicación | Alta | Implementar un plan de contingencia que garantice respaldo automático y servidores espejo. |
| 2 | El usuario no puede iniciar sesión | Alta | Revisar los módulos de autenticación y la configuración de Firebase. |
| 3 | Asignaciones de reservas fallan al procesarse | Alta | Verificar la lógica de asignación y la conexión con Firestore. |
| 4 | No se registra correctamente la capacidad del vehículo | Media | Revisar la lógica de actualización de capacidad disponible durante la asignación de reservas. |
| 5 | Las notificaciones no se entregan en tiempo real | Alta | Revisar la integración con Firebase Messaging y la configuración de permisos. |

* 1. **Tipos de pruebas**
     1. **Pruebas Funcionales**

#### **Objetivo de la Prueba:**

El objetivo de las pruebas es asegurar que la navegación y las funcionalidades dentro de la aplicación reflejen correctamente las reglas del negocio y los requerimientos definidos. Esto incluye la validación de los flujos entre ventanas, la interacción con los campos de entrada, y el uso adecuado de los métodos de acceso disponibles, tanto para pasajeros como para conductores.

#### **Técnica Utilizada:**

Para garantizar la precisión del sistema, cada caso de uso y flujo funcional se ejecuta utilizando tanto datos válidos como inválidos. La idea es comprobar lo siguiente:

1. **Resultados Esperados:**
   * Verificar que las funcionalidades operan según lo definido cuando se utilizan datos válidos.
   * Validar que los flujos y cambios de estado se realizan de forma precisa (ej.: asignación de reservas, confirmación de encuentros, etc.).
2. **Mensajes de Error Apropiados:**
   * Asegurar que se muestran mensajes de error claros y precisos cuando se ingresan datos inválidos (ej.: credenciales incorrectas, reserva inexistente).
   * Comprobar que las notificaciones y alertas se entregan en caso de errores operativos, como la falta de conexión a internet.

#### **Criterio de Validación:**

* Ejecución Completa: Todas las pruebas planificadas se llevaron a cabo conforme a lo establecido.
* Asignación de Defectos: Cada defecto identificado durante el proceso de pruebas ha sido registrado y asignado para su corrección.
* Resultados Consistentes: Los casos de prueba mostraron comportamientos consistentes en situaciones previstas y no previstas, asegurando la estabilidad del sistema ante distintos escenarios.

Este enfoque permite validar tanto los flujos funcionales como las excepciones, garantizando que la experiencia de usuario sea coherente con las reglas del negocio y que los errores se manejen de forma adecuada.

* + 1. **Herramientas involucradas**

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

|  | **Herramienta** | **Versión** |
| --- | --- | --- |
| **Firebase Cloud Storage** | **Base de Datos NoSQL** | **2024** |
| **Android Studio** | **Emulador Android** | **Koala** |
| **Ionic + Angular** | **Framework Ts + Framework Responsive** | **v8 + v18** |
| **Firebase Authentication** | **Manejo de autenticación por email y teléfono** | **2024** |

* + 1. **Preparación del ambiente de pruebas**

Para configurar el ambiente de pruebas, es fundamental definir los recursos de software utilizados y garantizar la correcta comunicación entre los distintos módulos de la aplicación. A continuación, se describe la estructura del ambiente de pruebas:

* **Aplicación Principal:**La app móvil se desarrolló utilizando **Ionic con Angular**, permitiendo pruebas tanto en dispositivos físicos como simulados para validar las funcionalidades de la interfaz y el comportamiento en distintos entornos.
* **Backend y Autenticación:**La autenticación y manejo de usuarios están gestionados mediante **Firebase**, asegurando una conexión estable y segura con la base de datos en tiempo real para la validación de credenciales y manejo de reservas.
* **Base de Datos:**El almacenamiento de datos utiliza **Cloud Firestore**, una base de datos NoSQL de Firebase, que permite la sincronización en tiempo real entre la app y los servicios backend. Las pruebas se realizarán para verificar que la recuperación y escritura de datos se lleven a cabo sin problemas durante los distintos casos de uso (como la gestión de reservas o la selección de vehículos).
* **Comunicación entre Módulos:**Se garantizará que la comunicación entre los diferentes módulos de la aplicación se ejecute correctamente. Esto incluye la interacción fluida entre:
  + **Módulo de autenticación:** para login y recuperación de contraseñas.
  + **Módulo de gestión de reservas:** para la visualización y confirmación de encuentros.
  + **Módulo de asignaciones:** para alertar a los conductores de nuevas asignaciones y cambios de estado en las reservas.
* **Pruebas en Diferentes Dispositivos:**El ambiente de pruebas incluye tanto dispositivos Android como iOS, además de simuladores, para garantizar que la aplicación funcione de manera uniforme en ambas plataformas.
  + 1. **Diseño del ambiente de pruebas**

| **ITEM** | **DESCRIPCIÓN** | **OBSERVACIONES** |
| --- | --- | --- |
| **HARDWARE** |  |  |
| **Estación de Pruebas (Administrador)** |  |  |
| Procesador | Intel Core i3-10100 |  |
| Memoria RAM | 12GB 2400MHz Kingston |  |
| Espacio en Disco | 256GB SSD |  |
| Tipo Monitor y Resolución | Monitor Samsung 1080p 27’’ |  |
| Unidad de Disquete | N/A |  |
| Tarjeta de red | Integrada |  |
| Modem | Huawei Generic |  |
| Mouse | Logitech G502 |  |
| Tipo Enlace | Canal |  |
| **Servidores** |  |  |
| ***Base de Datos (FIREBASE)*** |  | Integracion de Firebase Cloud |
| Procesador | N/A |  |
| Memoria RAM | N/A |  |
| Espacio en Disco | N/A |  |
| Tipo Monitor y Resolución | N/A |  |
| Unidad de Disquete | N/A |  |
| Tarjeta de red | N/A |  |
| Modem | N/A |  |
| Mouse | N/A |  |
| Tipo Enlace | N/A |  |
| ***Aplicación (pruebas)*** |  |  |
| Procesador | Intel Core i3-10100 |  |
| Memoria RAM | 12GB 2400MHz Kingston |  |
| Espacio en Disco | 256GB SSD |  |
| Tipo Monitor y Resolución | Monitor Samsung 1080p 27’’ |  |
| Unidad de Disquete | N/A |  |
| Tarjeta de red | Integrada |  |
| Módem | Huawei Generic |  |
| Mouse | Logitech G502 |  |
| Tipo Enlace | Canal |  |
| **Impresoras** |  |  |
| Marca y Modelo | N/A |  |
| Tipo | N/A |  |
| Resolución | N/A |  |
| Rendimiento | N/A |  |
| Dedicación | N/A |  |
| **RED** |  |  |
| Topología | Estrella |  |
| Medio | Fibra Óptica |  |
| Velocidad | 600Mb/s |  |
| Protocolo | TCP,UDP |  |
| Módems | Huawei Generic |  |
| Conexión Internet | Si |  |
| **Sistema de Respaldo / Restauración** |  |  |
| Unidad (Modelo y Marca) | N/A |  |
| Capacidad | N/A |  |
| Ubicación | N/A |  |
|  |  |  |
| **SOFTWARE** |  |  |
| **Estación de Pruebas (Cliente)** |  |  |
| Sistema Operativo | Windows 11 |  |
| Herramienta de *testing* | Android Studio |  |
| Herramienta de Modelamiento | Firebase Storage |  |
| BDMA | Firebase Cloud |  |
| Browser | Mozilla / Chrome |  |
| Software de Escritorio | N/A |  |
| **Base de Datos (Firebase)** |  | Integración de Firebase Cloud |
| Sistema Operativo | N/A |  |
| Software de Red | N/A |  |
| Dominio/Cuenta | N/A |  |
|  |  |  |
| **Aplicación (pruebas)** |  |  |
| Sistema Operativo | Windows 11 |  |
| Software de Red | PuTTY |  |
| Dominio/Cuenta | Admin |  |
|  |  |  |
| **Repositorios** |  | Repositorio Alojado en Github |
| Servidor | N/A |  |
| Dominio / Cuenta | N/A |  |

* + 1. **Diseño ambiente de pruebas**

El ambiente de pruebas fue diseñado para validar los siguientes flujos críticos:

1. **Login de usuarios:** Prueba de acceso y verificación de email.
2. **Asignación y confirmación de reservas:** Validación de lógica FIFO y cambios de estado.
3. **Notificaciones en tiempo real:** Confirmación de la entrega y recepción de notificaciones.
4. **Gestión de capacidad de vehículos:** Prueba de reducción automática del espacio disponible al asignar pasajeros.
5. **Resultados de las pruebas**

Se llevaron a cabo múltiples pruebas funcionales y de integración. Todos los casos críticos relacionados con la autenticación, asignación de reservas, y notificaciones fueron verificados con éxito, garantizando que el flujo del sistema opere como se esperaba.

| **Descripción** | **Condiciones** | **Resultado Esperado** | **Resultado Obtenido** | **Observaciones** |
| --- | --- | --- | --- | --- |
| **Iniciar Sesión en la Aplicación** | El usuario debe haber registrado previamente su email y contraseña.  Conexión a Internet disponible. | El usuario accede correctamente al dashboard tras ingresar credenciales válidas. | El usuario accede correctamente. |  |
| **Ingresar Credenciales no Registradas** | Conexión a Internet disponible.  Email y contraseña no registrados. | Aparece una alerta indicando “Credenciales incorrectas”. | Aparece una alerta indicando “Credenciales incorrectas”. |  |
| **Ingresar Pasajero con Teléfono** | El pasajero debe tener una reserva previamente registrada.  Conexión a Internet. | El sistema permite el ingreso del pasajero con su número de teléfono para confirmar su reserva. | El Pasajero ingresa con su teléfono. |  |
| **Selección de Vehículo** | El usuario debe haber iniciado sesión previamente.  Conexión a Internet disponible. | El usuario selecciona la van, y su estado cambia para que no pueda ser seleccionada por otro usuario. | El usuario selecciona la van. |  |
| **Mostrar Reservas** | El conductor debe haber iniciado sesión.  Conexión a Internet disponible. | Las reservas se muestran, y al asignar una reserva, su estado cambia a "asignado". | Se Muestran las reservas. |  |
| **Confirmar Encuentro** | El conductor debe haber iniciado sesión.  Conexión a Internet disponible. | Al confirmar el encuentro, la reserva cambia de estado a "en servicio". | Cambia el estado de la Reserva. |  |
| **Finalizar Servicio** | El conductor debe haber iniciado sesión.  Conexión a Internet disponible. | Al finalizar el servicio, la reserva cambia de estado a "completado" y desaparece de la lista activa de reservas. | Cambia el estado de la Reserva. |  |
| **Confirmación de Llegada a Punto** | El pasajero debe tener una reserva válida.  Conexión a Internet disponible. | Al confirmar la llegada, la reserva pasa a estado "pendiente de asignación". | Cambia el estado de la Reserva. |  |
| **Confirmación de Encuentro con Conductor** | El pasajero debe tener una reserva válida.  Conexión a Internet disponible. | Al confirmar el encuentro, la reserva cambia a "en servicio". | Cambia el estado de la Reserva. |  |
| **Formulario de Registro de Conductor** | Usuario con conexión a Internet.  Email válido para verificación. | El conductor completa el formulario, se registra y recibe un email para verificar su cuenta. | Se logra registrar el Usuario |  |
| **Recuperación de Contraseña** | Usuario con conexión a Internet.  Email válido. | El usuario recibe un correo para restablecer su contraseña y puede acceder con la nueva. | Recibe email de Recuperación. |  |
| **Selección de Conductor/Pasajero** | Usuario debe tener credenciales registradas (conductor) o número de teléfono asociado a una reserva (pasajero). | Dependiendo de la selección, el flujo cambia:  - Conductor inicia sesión.  - Pasajero ingresa con su teléfono. | Selecciona correctamente que usuario utilizará la app. |  |

**2.1 Reporte de defectos**

Los defectos encontrados durante las pruebas fueron registrados y priorizados para su corrección. Se identificaron principalmente errores relacionados con:

| **ID** | **ID Caso Prueba** | **Nº Ciclo** | **Fecha** | **Módulo** | **Descripción Detallada** | **Tipo** | **Severidad** | **Estado** | **Observación de equipo de desarrollo** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Inicio de Sesión | 1 | 17-10-2024 | Autenticación | Validar que el usuario pueda iniciar sesión con credenciales correctas y sea redirigido al dashboard. | Prueba Funcional | Alta | OK | N/A |
| 2 | Ingreso con Credenciales Inválidas | 1 | 17-10-2024 | Autenticación | Verificar que al intentar ingresar con email y contraseña no registrados, se muestre un mensaje de error. | Prueba Funcional | Alta | OK | N/A |
| 3 | Ingreso del Pasajero con Teléfono | 1 | 17-10-2024 | Acceso Pasajeros | Validar que el pasajero pueda ingresar con su número de teléfono para confirmar su reserva. | Prueba Funcional | Alta | OK | N/A |
| 4 | Selección de Vehículo | 1 | 17-10-2024 | Gestión de Vehículos | Verificar que el conductor pueda seleccionar un vehículo disponible, y que este no pueda ser seleccionado por otros usuarios. | Prueba Funcional | Alta | OK | N/A |
| 5 | Visualización de Reservas | 1 | 17-10-2024 | Gestión de Reservas | Comprobar que las reservas disponibles se muestren y que al asignar una, su estado cambie a "asignado". | Prueba Funcional | Alta | OK | N/A |
| 6 | Confirmación de Encuentro | 1 | 17-10-2024 | Gestión de Reservas | Validar que al confirmar el encuentro con el pasajero, la reserva cambie a estado "en servicio". | Prueba Funcional | Alta | OK | N/A |
| 7 | Finalización del Servicio | 1 | 17-10-2024 | Gestión de Reservas | Verificar que al finalizar el servicio, la reserva cambie a "completado" y desaparezca de la lista activa de reservas. | Prueba Funcional | Alta | OK | N/A |
| 8 | Confirmación de Llegada a Punto | 1 | 17-10-2024 | Acceso Pasajeros | Comprobar que el pasajero pueda confirmar su llegada y que el estado de la reserva pase a "pendiente de asignación". | Prueba Funcional | Alta | OK | N/A |
| 9 | Confirmación de Encuentro con Conductor | 1 | 17-10-2024 | Gestión de Reservas | Validar que el pasajero pueda confirmar el encuentro con el conductor y que la reserva pase a "en servicio". | Prueba Funcional | Alta | OK | N/A |
| 10 | Registro de Conductor | 1 | 17-10-2024 | Registro | Verificar que el conductor pueda registrarse y reciba un correo para verificar su cuenta. | Prueba Funcional | Alta | OK | N/A |
| 11 | Recuperación de Contraseña | 1 | 17-10-2024 | Recuperación | Validar que el usuario pueda recuperar su contraseña mediante un correo de restablecimiento. | Prueba Funcional | Alta | OK | N/A |
| 12 | Selección de Conductor/Pasajero | 1 | 17-10-2024 | Flujo de Usuario | Comprobar que el usuario pueda seleccionar si es conductor o pasajero y sea redirigido al flujo correspondiente. | Prueba Funcional | Alta | OK | N/A |