

## Plantilla de Plan Maestro de Pruebas

### Preparado por:

Adrián Ricardo González Muralles - 23152

Jose Pablo Ordoñez Barrios – 231329

Marcos Rodrigo Ambrocio Larios - 231140

José Alejandro Antón Escobar - 221041

### Introducción

- Informar el alcance de las pruebas para el proyecto en cuestión, dando una breve explicación o resumen de todas las secciones que integrará el plan. Además de señalar cuáles son los riesgos y las suposiciones.

### Recursos

Tester	% de participación
Marcos Rodrigo Ambrocio Larios	25%
José Alejandro Antón Escobar	25%
Jose Pablo Ordoñez Barrios	25%
Adrián Ricardo González Muralles	25%

### Alcance

Las pruebas incluyen todas las funcionalidades principalmente las funcionalidades del backend, como las apis y controladores que son las áreas de mayor estrés en el desarrollo de la aplicación y en el front-end las pruebas están en las funcionalidades como el boton que se nesecitan pruebas para su correcto funcionamiento

### Fuera del Alcance

Se debe incluir los elementos del sistema que no se probarán como parte del plan.

### Características a probar

- Descripción de las pruebas a realizar
  - Se establecerá el rango de las fechas en las que el sistema será probado.
  - Se describirán las pruebas que se realizarán al sistema utilizando las plantillas siguientes:
  - Pruebas de Funcionalidad:

ID Caso Prueba	Escenario	Variable 1	Variable 2	...	Resultado esperado
01	Registro	Usuario nuevo	Datos válidos		Registro exitoso
02	Login	Email y contraseña	Correctos		Inicio de sesión exitoso
03	Solicitar un viaje	Ubicación origen	Ubicación destino		Se muestra lista de conductores
04	Confirmar Viaje	Conductor disponible	Tipo de vehículo		Viaje confirmado y en progreso
05	Calificación	Viaje completado	5 estrellas		Calificación registrada exitosamente

○ Pruebas de Carga:

Identificador de la prueba	Parte de la aplicación probada	Condición	Resultado Esperado	Método o herramienta a utilizar
C01	API de login	1000 usuarios en simultaneo	Responde en < 3s	
C02	Crear viaje	500 solicitudes concurrentes	95% respuestas exitosas	

○ Pruebas de Seguridad:

Identificador de la prueba	Condición	Elemento a probar	Resultado esperado
S01	Acceso sin token	Endpoint /viajes	Retorna 401 No autorizado
S02	Token expirado	API/usuario	Retorna 403 Token inválido
S03	Entrada maliciosa	Formulario Login	Acceso denegado + log de alerta

## Criterios de aceptación o fallo

Son los criterios que serán considerados para dar por completado el Plan de Pruebas, por ejemplo: Porcentaje de la cobertura de las pruebas esperado, cierto porcentaje de casos exitosos, cobertura de todos los componentes, porcentaje de defectos corregidos, todo error debe de ir acompañado de un mensaje de validación, entre otros. Cuando un criterio de aprobación fue rechazado se toma acción para el criterio utilizando el criterio de fallo para dicho criterio. Todo criterio de aprobación lo debe de acompañar un criterio de fallo, el cual es la acción que se tomara en la implementación del plan, cuando se ejecuten las pruebas sobre el proyecto.

Id criterio	Descripción	Aprobación	Fallo
A1	90% de los casos funcionales pasan	Se aprueba el módulo	Se revisan errores y se rehace la prueba
A2	80% cobertura de código con pruebas unitarias	Continúa con integración	Se exige mayor cobertura antes del despliegue
A3	0 vulnerabilidades críticas detectadas	Se aprueba despliegue en producción	Se bloquea paso a producción

## Criterios de suspensión y reanudación

En esta sección se describen los criterios de suspensión los cuales establecen claramente bajo qué condiciones se detienen un conjunto de casos de pruebas, por ejemplo, en caso de existir defectos que impidan la ejecución de más casos de pruebas, cierto porcentaje de casos fallidos, o cualquier otro que se especifique. Los criterios de reanudación establecen bajo qué criterios se reanudarán las pruebas, por ejemplo: cuando nos entreguen una nueva versión de pruebas, cuando nos realicen las configuraciones necesarias, etc. Es importante considerar que para cada criterio de suspensión debe contar su criterio de reanudación.

Criterio de suspensión	Criterio de reanudación
Fallo crítico impide que el sistema inicie sesión	Una nueva versión corrige el bug
> 50% de los casos fallan en pruebas funcionales	El equipo de desarrollo entrega una nueva versión corregida
API deja de responder durante pruebas de carga	Se reinicia el servicio y se verifican los bugs

## Infraestructura

### Lenguajes de Programación

Frontend: JavaScript con React Native (para aplicaciones móviles).

Backend: JavaScript con Express.js (Node.js).

Base de Datos: PostgreSQL.

### 2. Frameworks y Librerías

React Native: Framework para desarrollo de aplicaciones móviles multiplataforma.

Express.js: Framework para creación de APIs RESTful en el backend.

Jest: Framework de pruebas unitarias y de integración para JavaScript y React.

Supertest: Utilizado para pruebas de endpoints en Express.js.

React Testing Library: Librería para pruebas de componentes en React Native.

### 3. Gestores y Control de Versiones

Git: Sistema de control de versiones distribuido.

### Suposiciones

Para llevar a cabo las pruebas del sistema, se requiere contar con un entorno de pruebas configurado de forma idéntica al de producción, con acceso a la base de datos poblada con datos de prueba, servicios externos simulados o en modo sandbox, y herramientas como Jest, Postman y emuladores móviles. Es necesario asumir que el desarrollo de las funcionalidades está completo, que los entornos son estables, y que los testers tienen los permisos adecuados. Además, deben realizarse acciones previas como cargar datos, ejecutar migraciones y validar la conectividad de los servicios.

### Riesgos

En esta sección se especificarán los riesgos que pueden afectar directa o indirectamente a los resultados de las pruebas. Identificar y tener las acciones preventivas y correctivas de los riesgos anteriormente definidos, nos permiten tomar decisiones rápidas y eficientes, porque anteriormente ya se realizó el análisis de los riesgos y sus acciones a tomar. Es importante que al documentar los riesgos sea de manera organizada y entendible para todos los involucrados del proyecto.

No	Riesgos	Probabilidad (1-5)	Impacto (1-5)	Severidad (Prob*Impct)	Plan de Mitigación
1	<i>Retrasos en el desarrollo de funcionalidades</i>	2	5	10	<i>Replanificar de acuerdo al avance</i>
2	<i>Usuarios no disponibles para pruebas UAT</i>	1	5	5	<i>Contactar con anticipación a usuarios clave</i>

3	<i>API de terceros no responde (ej. Google Maps)</i>	2	4	8	<i>Mockear respuestas para pruebas independientes</i>
4	<i>Fallos de red o infraestructura externa</i>	3	4	12	<i>Tener entornos locales simulados para pruebas críticas</i>

<sup>i</sup> Plantilla adaptada de

---

<sup>i</sup> Crispin, L., & Gregory, J. (2009). Agile Testing. A practical Guide for Tester and Agile Teams. Angewandte Chemie International Edition, 6(11), 951–952. Pearson Education, Inc. y Plantilla de Plan de Pruebas. Universidad Nacional Andrés Bello