PLAN DE PRUEBAS

Proyecto: Reservas Trenque

Desarrollador: Michael Jonathan Illescas

Fecha: 03/07/2025

# Índice

1. Objetivo del Plan de Pruebas

2. Alcance

3. Estrategia de Pruebas

4. Criterios de Entrada y Salida

5. Ambiente de Pruebas

6. Herramientas

7. Roles y Responsabilidades

8. Riesgos Identificados

9. Cronograma General

10. Métricas de Evaluación

# 1. Objetivo del Plan de Pruebas

Garantizar la calidad del sistema 'Reservas Trenque', verificando que todas las funcionalidades trabajen correctamente, que los usuarios puedan interactuar de forma fluida y que los datos se gestionen de forma segura, confiable y consistente.

# 2. Alcance

Este plan cubre todas las funcionalidades del sistema Reservas Trenque:  
- Registro e inicio de sesión de usuarios.  
- Visualización de alojamientos disponibles con sus detalles.  
- Búsqueda avanzada por fechas y filtros.  
- Gestión de favoritos para usuarios autenticados.  
- Reserva de alojamientos con selección de fechas.  
- Historial de reservas por usuario.  
- Calificación y reseñas de alojamientos.  
- Panel administrativo para gestionar contenido.  
- Notificaciones por correo electrónico.  
- Integración con WhatsApp.  
- Diseño responsivo.

# 3. Estrategia de Pruebas

Se aplicarán las siguientes pruebas:  
- Pruebas unitarias: métodos individuales del backend con JUnit.  
- Pruebas de integración: interacción entre controladores, servicios y repositorios.  
- Pruebas funcionales: validación de casos de uso desde el frontend.  
- Pruebas de interfaz: verificar usabilidad y diseño responsivo.  
- Pruebas de aceptación: validación del sistema completo.  
- Pruebas de correo: comprobar el correcto envío de emails de confirmación o notificación.

# 4. Criterios de Entrada y Salida

Criterios de Entrada:  
- Funcionalidades desarrolladas.  
- Entorno de pruebas operativo.  
- Datos de prueba disponibles.  
  
Criterios de Salida:  
- Todos los casos de prueba ejecutados.  
- Corrección de errores críticos.  
- Validación funcional y técnica satisfactoria.

# 5. Ambiente de Pruebas

- Backend: Spring Boot sobre MySQL  
- Frontend: React.js  
- Estilos: Bootstrap / CSS  
- Base de datos de pruebas con datos ficticios  
- Pruebas de correo mediante API SMTP

# 6. Herramientas

- JUnit / Mockito para pruebas unitarias en Java  
- Postman para pruebas de endpoints  
- React Testing Library (opcional)  
- Git + GitHub para versionado  
- Swagger UI para pruebas manuales de API

# 7. Roles y Responsabilidades

- Desarrollador principal: Michael Jonathan Illescas  
- Responsable de pruebas unitarias, funcionales y validaciones generales.

# 8. Riesgos Identificados

- Fallas en la validación de fechas durante reservas.  
- Problemas de compatibilidad en móviles.  
- Errores en la lógica de envío de correos.  
- Cambios de requerimientos sobre la marcha.

# 9. Cronograma General

- Semana 1: Pruebas de registro y visualización de alojamientos.  
- Semana 2: Pruebas de búsqueda, filtros y reservas.  
- Semana 3: Pruebas de panel admin, favoritos y notificaciones.  
- Semana 4: Pruebas integradas y validación final.

# 10. Métricas de Evaluación

- Porcentaje de casos de prueba exitosos.  
- Cantidad de errores encontrados por tipo.  
- Tasa de corrección de errores.  
- Tiempo promedio de resolución.