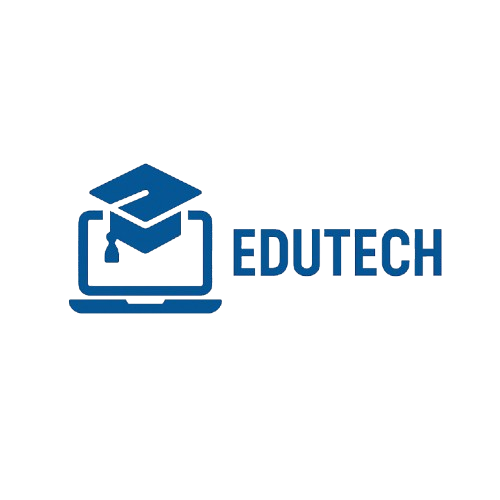
**Proyecto EDUTECH**

****

**Plan de Pruebas**

**Preparado por**: Tabatha Gamboa, Fabián Parraguez, Irina Martínez, Rahien Loncopan.

**Introducción**

El presente Plan de Pruebas está diseñado para asegurar el correcto funcionamiento del sistema EDUTECH, una plataforma educativa construida con arquitectura de microservicios. El objetivo es validar el comportamiento de los servicios, su comunicación a través del API Gateway y la generación de reportes. Este documento establece el alcance de las pruebas, roles involucrados y herramientas utilizadas en este proyecto que a su vez se decidió implementar una metodología ágil que combina elementos de Scrum y Kanban. Esto permite gestionar tareas de manera visual y dinámica.

**Recursos**

| **Tester** | **% Participación** |
| --- | --- |
| Tabatha Gamboa | 25% |
| Rahien Loncopan | 25% |
| Irina Martinez | 25% |
| Fabian Parraguez | 25% |

**Alcance**

Las pruebas cubrirán los siguientes aspectos del sistema desde el frontend:

* Validación de vistas y componentes (inicio, login, Pantalla estudiante, Pantalla Profesor, Catálogo,Contenido, Compra, Soporte).
* Comprobación de navegación y rutas internas.
* Validación de formularios: estructura, validaciones en campos, retroalimentación visual.
* Integración del frontend con las APIs expuestas por los microservicios.
* Experiencia de usuario (UX): fluidez, diseño responsivo, consistencia visual.
* Compatibilidad en navegadores comunes (Chrome, Firefox, Edge).

**Fuera del Alcance**

* Pruebas de rendimiento avanzadas o pruebas de carga.
* Validación interna de microservicios (ya abordada en el backend).
* Automatización de pruebas (queda como trabajo futuro).

**Pruebas de Rendimiento**

* Se realizaron pruebas empíricas observando los tiempos de carga visual y respuesta tras acciones del usuario.
* No se aplicaron herramientas de benchmarking o profiling para el frontend.

**Pruebas de Aceptación (UAT)**

* Se ejecutaron pruebas con distintos usuarios del equipo en roles de estudiante y profesor.
* Se simularon flujos completos: Pantalla inicial, login, catálogo, compra y soporte.

**Infraestructura**

* Se utilizó un entorno local con Node.js y React (Vite o Create React App).
* TypeScript desarrollo front, Tailwind CSS Diseño de Interfaz
* Herramientas: Git integrado con GitHub, Postman para validar respuestas,Visual Studio Code, Railway Implementación y Hosting.
* Gestión de datos: MySQL.

**Suposiciones**

* El backend está operativo y correctamente configurado en endpoints accesibles.
* La base de datos cuenta con datos de prueba suficientes para simular escenarios.
* Se asumió resolución estándar de pantalla y conexión estable durante pruebas.