



ALUMNO:

María Fernanda Fuentes Abascal

DOCENTE:

José Miguel Carrera Pacheco

ACTIVIDAD:

Navegación avanzada y manejo de errores

MATERIA:

Desarrollo Web Profesional

FECHA:

01/02/2026

# Plataforma Web de Apoyo Académico

**Proyecto:** Plataforma Web de Apoyo Académico

**Tipo de sistema:** Aplicación Web

## 1. INTRODUCCIÓN GENERAL

El presente documento define la arquitectura, navegación, experiencia de usuario, accesibilidad y manejo de errores de la Plataforma Web de Apoyo Académico. El objetivo es establecer una base técnica sólida antes de la implementación, siguiendo prácticas utilizadas en proyectos reales de la industria del software.

Este documento funciona como una fuente única de verdad, permitiendo que todos los roles del equipo trabajen bajo los mismos criterios técnicos, de diseño y de accesibilidad.

## 2. DEFINICIÓN DEL PROBLEMA

En el contexto académico, muchos estudiantes presentan dificultades para acceder a contenido de apoyo confiable, organizado y accesible. La información suele encontrarse dispersa, sin jerarquía clara y con plataformas poco intuitivas o inaccesibles.

Esto genera:

- Pérdida de tiempo
- Frustración
- Bajo aprovechamiento del contenido
- Abandono de plataformas educativas

## 3. USUARIOS REALES IDENTIFICADOS

- Estudiantes de nivel medio superior y superior
- Usuarios con diferentes niveles de alfabetización digital
- Usuarios que navegan sin mouse (teclado)
- Usuarios con limitaciones visuales leves

## 4. JUSTIFICACIÓN DE LA SOLUCIÓN (APLICACIÓN WEB)

Se justifica el uso de una aplicación web debido a que:

- Es accesible desde cualquier dispositivo con navegador
- No requiere instalación
- Permite actualización continua de contenido
- Facilita control de usuarios y roles
- Es escalable a futuro (cursos, evaluaciones, seguimiento)

## 5. ALCANCE INICIAL (MVP)

El MVP del sistema incluye:

- Visualización de cursos académicos
- Lectura de contenidos
- Búsqueda de cursos
- Registro e inicio de sesión
- Navegación accesible
- Manejo de errores controlado (404 y 500)

## 6. ARQUITECTURA GENERAL DEL SISTEMA

### 6.1 Capas del sistema

- **Frontend:** Interfaz, navegación, accesibilidad, UX
- **Backend:** API REST, lógica de negocio, manejo de errores
- **Base de datos:** Almacenamiento estructurado de usuarios y cursos
- **Infraestructura:** Docker, control de versiones y CI/CD

## 7. ARQUITECTURA DE INFORMACIÓN Y NAVEGACIÓN

### 7.1 Arquitectura de información

La información se organiza jerárquicamente:

- Inicio
- Cursos
- Detalle de curso
- Usuario (perfil)
- Autenticación

Esto permite una comprensión rápida del sistema.

### 7.2 Sitemap y flujos de navegación

El sistema se diseña pensando en flujos, no solo en pantallas:

- Usuario entrá
- Explora cursos
- Consulta contenido
- Regresa o continúa navegando

### 7.3 Rutas del sistema

**Rutas públicas:**

- Inicio
- Cursos
- Detalle de curso
- Login / Registro
- Página 404
- Página 500

### **Rutas privadas:**

- Perfil de usuario
- Gestión de cursos (futuro)

## **8. UX/UI PROFESIONAL Y ACCESIBILIDAD**

### **8.1 Principios aplicados**

- Heurísticas de Nielsen
- Principios de Steve Krug
- WCAG 2.1 nivel básico

### **8.2 Decisiones UX**

- Menú persistente
- Jerarquía visual clara
- Botones visibles
- Mensajes claros
- Lenguaje no técnico

### **8.3 Accesibilidad aplicada**

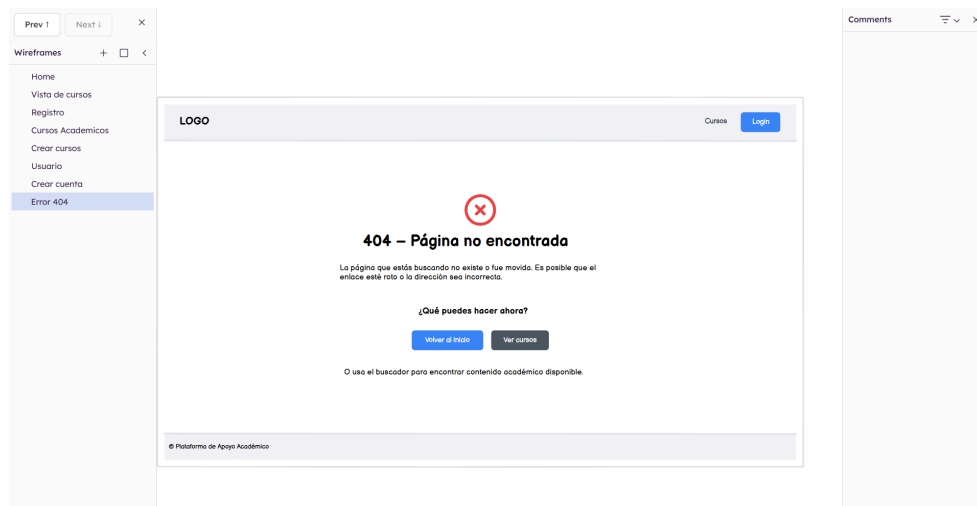
- HTML semántico
- Navegación completa por teclado
- Orden lógico de tabulación
- Foco visible
- Contraste adecuado
- Uso de ARIA cuando es necesario

## **9. NAVEGACIÓN AVANZADA Y MANEJO DE ERRORES**

### **9.1 Importancia del manejo de errores**

Un sistema profesional no oculta errores, los explica y guía al usuario.

## 9.2 Pantalla 404 – Recurso no encontrado



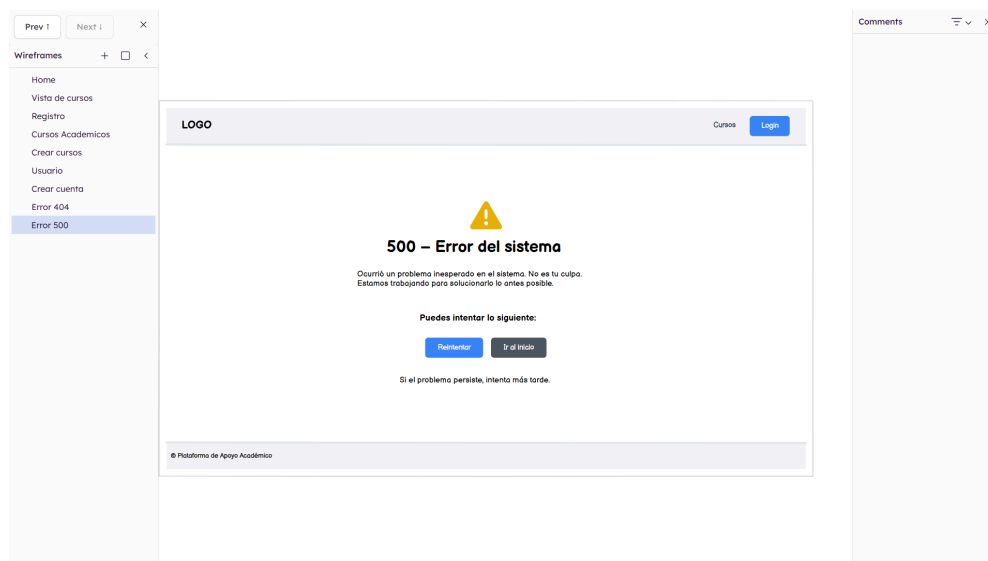
### Objetivo:

- Informar que la página no existe
- Evitar que el usuario se pierda
- Guiar al siguiente paso

### Características:

- Mensaje claro
- Botón para volver al inicio
- Acceso a cursos
- Navegación por teclado
- Diseño consistente

## 9.3 Pantalla 500 – Error interno del servidor



**Objetivo:**

- Informar del fallo sin culpar al usuario
- Reducir frustración
- Ofrecer una alternativa

**Características:**

- Mensaje empático
- Botón de reintento
- Enlace al inicio
- `aria-live` para lectores de pantalla

**9.4 Herramientas utilizadas (Frontend)**

- HTML5 semántico
- CSS accesible
- ARIA (`aria-live`, roles)
- Router Frontend
- Lighthouse para validación

**10. BACKEND: MANEJO DE ERRORES Y RUTAS**

El backend implementa:

- Arquitectura MVC
- Middlewares de errores
- Rutas controladas
- Respuestas HTTP estandarizadas
- Errores 404 y 500 con contratos JSON

Estas respuestas se comunican al frontend para mostrar las pantallas correspondientes.

**11. TESTING Y VALIDACIÓN**

Se realizan pruebas de:

- Navegación por teclado
- Accesibilidad
- Rutas inexistentes
- Errores del servidor
- Flujo completo del usuario

**12. BASE TÉCNICA Y HERRAMIENTAS**

- Git y GitHub
- Docker y Docker Compose
- CI/CD básico

- Estructura profesional de carpetas
- Documentación desde el inicio