



Propuesta Técnica y Arquitectura del Sitio Web

Escuela Especial de Lenguaje “LALALA”

1. Introducción

La presente propuesta técnica describe el desarrollo del sitio web oficial de la **Escuela Especial de Lenguaje “LALALA”**, orientado a satisfacer las necesidades de una comunidad escolar rural en Chile. El objetivo es ofrecer una plataforma moderna, accesible y segura, con secciones dedicadas a personal docente/administrativo y un espacio público para apoderados. El proyecto se implementará con **Next.js 14+** (React) y **TypeScript**, tecnología que proporciona un rendimiento excepcional, excelente experiencia de desarrollo y gran escalabilidad ¹. Además se incorporarán prácticas de **diseño responsivo** (compatibilidad con dispositivos móviles) y medidas de **accesibilidad web** conforme a la normativa chilena vigente (WCAG 2.0 Nivel AA) ² ³. Se priorizará la optimización de la velocidad de carga, la seguridad (HTTPS, validación de datos, control de acceso) y la navegabilidad intuitiva del sitio.

2. Objetivos del Proyecto

- **Página principal atractiva:** Mostrar información institucional relevante (imagen o banner, nombre de la escuela, misión breve) y accesos directos al login de usuarios y a la sección pública.
 - **Autenticación segura por roles:** Implementar un sistema de inicio de sesión (correo electrónico y contraseña) que distinga permisos de **Administrador** (acceso total) y **Profesor** (acceso restringido).
 - **Panel privado para docentes:** Brindar a los profesores un espacio con módulos de planificación docente, agendamiento de reuniones con apoderados, visualización de horarios de clases y consulta del Proyecto de Mejoramiento Educativo (PME).
 - **Panel administrativo completo:** Proveer a los administradores herramientas para editar todo el contenido, gestionar usuarios y subir/editar/eliminar archivos (PDFs, imágenes).
 - **Sección pública informativa:** Crear un área abierta sin necesidad de autenticación que incluya el proyecto educativo de la escuela, documentos descargables (reglamentos, planes, protocolos), calendario escolar, datos del equipo multidisciplinario (con formulario de contacto/fonoaudiología) y galería multimedia. Además, se incluirá enlace oficial al portal del Ministerio de Educación (Mineduc).
 - **Gestión de documentos PDF:** Permitir la subida y organización de documentos en PDF desde el panel administrativo, clasificándolos en categorías lógicas (reglamentos, manuales, protocolos, etc.) para fácil acceso y descarga.
 - **Diseño responsivo:** Garantizar que el sitio se adapte correctamente a cualquier dispositivo (computadoras, tablets y teléfonos móviles) mediante un diseño web responsivo.
-

3. Alcance Funcional

3.1. Página Principal (/)

- Elementos visuales destacados (logo o imagen institucional) y el nombre completo de la escuela.
- Breve mensaje de bienvenida o descripción institucional.
- Formulario de inicio de sesión integrado para docentes y administradores.
- Botón o enlace claro que permita navegar al sitio público (área abierta).

3.2. Autenticación y Roles

- Acceso mediante correo electrónico y contraseña, con manejo seguro de sesiones.
- Definición de roles de usuario: **Administrador** (control total del sitio) y **Profesor** (funciones docentes).
- Protección de rutas privadas mediante middleware de Next.js, que redirige a login a usuarios no autenticados o sin permisos adecuados.

3.3. Panel del Profesor (/profesor)

- **Planificaciones:** Herramienta para crear, editar y consultar planificaciones de clases y actividades.
- **Reuniones con apoderados:** Agenda y registro de reuniones entre profesores y padres, con notificaciones o avisos.
- **Horarios de funcionamiento:** Visualización del horario de clases y disponibilidad de cada docente.
- **Proyecto de Mejoramiento Educativo (PME):** Sección con información y avances del PME institucional.

3.4. Panel Administrativo (/admin)

- Acceso exclusivo para administradores (usuarios con rol *admin*).
- Funciones de edición de todo el contenido público y privado del sitio (textos, imágenes, noticias).
- **Carga y gestión de archivos:** Subida, organización y eliminación de archivos PDF (reglamentos, planes, manuales, protocolos, etc.) desde el navegador.
- Administración de la galería multimedia (fotos y videos institucionales), con vista previa y categorización de medios.
- Gestión de usuarios: creación y asignación de roles para nuevos profesores o personal escolar.

3.5. Sitio Público (/public)

- Información sobre el **Proyecto Educativo** institucional, con documentos descargables en PDF.
- **Documentos oficiales** (reglamentos, proyectos, manuales) organizados para descarga.
- **Calendario escolar** actualizado (fechas de inicio/fin de clases, eventos importantes).
- **Centro de Padres/Apoderados:** sección con avisos y recursos útiles para las familias.
- **Equipo multidisciplinario:** presentación del equipo de apoyo (fonoaudiólogo, psicopedagogo, asistentes), con formulario de reserva de evaluación fonoaudiológica o diagnósticos.
- **Galería multimedia:** fotos y videos de actividades escolares.
- **Enlace al Mineduc:** redirección oficial al portal del Ministerio de Educación para información complementaria.

4. Arquitectura Técnica Propuesta

4.1. Tecnologías

- **Next.js 14+** (App Router): framework moderno basado en React que facilita el enrutamiento híbrido (SSR/SSG) y optimiza el rendimiento y SEO ¹.
- **TypeScript**: superset de JavaScript con tipado estático, que aumenta la robustez y mantenibilidad del código.
- **Tailwind CSS**: biblioteca de utilidades CSS para diseñar rápidamente interfaces adaptativas; permite aplicar clases condicionales en distintos puntos de quiebre, facilitando la creación de interfaces responsivas ⁴.
- **NextAuth.js** (o sistema equivalente): para autenticación segura y gestión de sesiones basadas en JWT/cookies.
- **Prisma ORM** con base de datos **PostgreSQL** (u otra base SQL): cliente ORM de alto rendimiento para modelar y consultar datos educativos.
- **Almacenamiento de archivos**: PDFs, imágenes y multimedia se almacenarán en el servidor o en un servicio de nube (AWS S3, Google Cloud Storage, etc.), con respaldo periódico.
- **CMS Headless (opcional)**: soluciones como Strapi o Sanity (versiones de código abierto) para permitir a administradores editar contenidos sin tocar código fuente.

4.2. Estructura Principal

Propuesta de estructura de archivos y directorios del proyecto:

```
/  
├── .env.local          # Variables de entorno (claves, tokens, DB URL)  
├── next.config.mjs     # Configuración general de Next.js (redirecciones,  
etc.)  
├── public/              # Recursos públicos (imágenes, PDFs, fuentes)  
│   ├── images/  
│   └── pdfs/  
└── src/  
    ├── app/                # Código principal y páginas (Next.js App Router)  
    │   ├── page.tsx         # Página principal (landing)  
    │   ├── login/            # Rutas de autenticación (login)  
    │   ├── admin/             # Panel administrativo (rutas privadas)  
    │   ├── profesor/          # Panel de profesor (rutas privadas)  
    │   └── public/            # Sección pública (páginas accesibles sin login)  
    ├── components/          # Componentes UI reutilizables  
    ├── lib/                 # Lógica central (autenticación, DB, utilidades)  
    └── styles/              # Estilos globales (Tailwind, CSS)  
└── package.json          # Dependencias y scripts del proyecto
```

4.3. Seguridad y Middleware

- Protección de rutas privadas usando **middleware** de Next.js: antes de cada solicitud a `/admin` o `/profesor` se verificará la sesión activa del usuario.
- Autorización por roles: si un usuario autenticado carece del rol necesario (por ejemplo, un profesor intenta acceder a `/admin`), será redirigido a una página de acceso denegado o al login.
- Validación de datos: todas las entradas de formularios se validarán tanto en el cliente como en el servidor para evitar inyecciones o datos inválidos.
- **HTTPS y buenas prácticas de seguridad:** se empleará cifrado SSL/TLS, protección de datos sensibles (p. ej. contraseñas hasheadas), y generación de tokens seguros para las sesiones.

4.4. Gestión de Archivos

- **Carga de PDFs:** desde el panel administrativo se podrá subir archivos PDF con validación de formato y tamaño máximos permitidos.
- **Organización:** los documentos se clasificarán automáticamente en carpetas o categorías (por ejemplo, "Reglamentos", "Proyectos", "Protocolos") para una gestión ordenada.
- **Almacenamiento seguro:** los archivos se guardarán en ubicaciones con copias de seguridad, ya sea en el servidor local o en una nube, para evitar pérdidas.
- **Acceso a documentos:** los archivos subidos estarán disponibles para descarga en la sección pública o privada correspondiente, respetando permisos (solo administradores y docentes pueden subir/borrar, pero apoderados pueden descargar los públicos).

5. Propuesta Económica

Concepto	Detalle	Valor CLP
Desarrollo web completo	Diseño, desarrollo, pruebas e implementación del sitio	\\$1.200.000
Funcionalidad de CMS (opc.)	Implementación de CMS para gestión autónoma de contenidos	+\\$300.000
Hosting, dominio y soporte	Configuración de servidor, subida del sitio e infraestructura básica (30 días de soporte)	Incluido

Nota: Los valores son estimados y pueden ajustarse según requerimientos adicionales o cambios en el alcance final.

6. Plan de Trabajo y Plazos

Etapa	Actividad	Duración estimada
1. Diseño y estructura básica	Maquetación inicial (landing page y rutas públicas)	1 semana

Etapa	Actividad	Duración estimada
2. Implementación de login	Desarrollo de autenticación y paneles privados	1 semana
3. Gestión de contenido	Construcción del panel administrativo y carga de archivos	1 semana
4. Ajustes finales y pruebas	Revisión, corrección de errores y documentación final	1 semana

7. Entregables

- Sitio web completamente funcional y desplegado en el entorno de producción.
- Código fuente organizado en repositorio privado (Git) para seguimiento de versiones.
- Documentación de usuario y administrador: guía básica para uso del sitio y carga de contenidos.
- Soporte técnico inicial para la puesta en marcha (configuración de dominio, resolución de dudas sobre el uso).

8. Consideraciones Finales

- El proyecto es **escalable**: la arquitectura propuesta permite agregar futuras funcionalidades (por ejemplo, noticias, inscripción en línea, comunicación con apoderados).
- Se priorizará la **accesibilidad y usabilidad**: el sitio cumplirá con los estándares WCAG 2.0 Nivel AA exigidos por la normativa chilena ² ³, de modo que cualquier persona (incluidas con discapacidad visual o motora) pueda utilizarlo con facilidad.
- El diseño será **responsive** y adaptado a la identidad institucional de la escuela.
- Se seguirán buenas prácticas de seguridad web actuales para proteger la información y la privacidad de los usuarios.

9. Contacto

Para consultas o mayor información, por favor comunicarse con:

[Tu Nombre Completo]

Correo electrónico: tu.email@dominio.cl

Teléfono: +56 9 XXXX XXXX

Agradecemos la oportunidad de colaborar con la **Escuela Especial de Lenguaje "LALALA"**. Quedamos a disposición para aclarar cualquier duda. Podemos entregar este documento en formato PDF o en un archivo editable (Word/Google Docs), según sus necesidades.

Fuentes: Recomendaciones de desarrollo web, frameworks modernos y estándares de accesibilidad (consultados en fuentes especializadas) 1 4 2 3 .

1 Next.js: La Guía Definitiva del Framework React más Popular - DEV Community

<https://dev.to/goaqidev/nextjs-la-guia-definitiva-del-framework-react-mas-popular-345f>

2 Accesibilidad web en Chile: evaluación de portal educativo | Educación y Tecnología

<https://revistas.umce.cl/index.php/edytec/article/view/1585>

3 Accesibilidad web en Chile: ¿Qué dice la ley y cuál es nuestra responsabilidad con las personas usuarias?

- IDA Blog | UX y Marketing para vincular marcas y audiencias.

<https://blog.ida.cl/accesibilidad/accesibilidad-web-chile/>

4 Responsive design - Core concepts - Tailwind CSS

<https://tailwindcss.com/docs/responsive-design>