

Arquitectura y Estructura de Carpetas - Escuela de Musica Barrial

1. Stack Tecnologico

Este proyecto utiliza una arquitectura moderna "Serverless First" con Next.js 15, garantizando escalabilidad y performance.

- Framework: Next.js 15 (App Router)
- Lenguaje: TypeScript
- Estilos: Tailwind CSS + Shadcn/UI
- Backend: Firebase (Firestore, Auth, Storage)
- Pagos: Mercado Pago SDK
- Emails: Resend

2. Estructura de Directorios (Source Code)

```
src/
|-- app/                                # Next.js App Router (Rutas y Layouts)
|   |-- (public)/                       # Grupo de rutas para el sitio abierto
|   |   |-- layout.tsx                 # Layout con Navbar y Footer publico
|   |   |-- page.tsx                  # Home Page dinamica
|   |   `-- [slug]/                   # RUTA DINAMICA: Procesa /nosotros, /clases, etc.
|   |       `-- page.tsx              # Page Template Unico
|   |-- (admin)/                       # Grupo de rutas protegidas para el Panel Admin
|   |   |-- layout.tsx                # Layout del Admin (Sidebar diferente)
|   |   `-- dashboard/               # Metricas y vista general
|   |-- api/                           # Route Handlers (Backend)
|   |   |-- payments/                 # Webhooks y Checkout de Mercado Pago
|   |   `-- email/                   # Endpoints de envio de correo
|   `-- layout.tsx                    # Root Layout (Fuentes, SEO base, AuthProvider)
|
|-- components/                        # Componentes de React
|   |-- sections/                     # Secciones dinamicas (Hero, Form, ClassList)
|   |-- shared/                       # SEOHead, Navbar, Footer, PaymentButton
|   `-- ui/                           # Componentes atomicos (Shadcn/UI)
|
|-- hooks/                             # Logica reutilizable (useAuth, useCompression)
|
|-- lib/                               # Clientes y configuraciones de SDKs
|   |-- firebase.ts                   # Firebase (Auth, DB, Storage)
|   |-- mercadopago.ts                # Mercado Pago Config
|   `-- resend.ts                     # Resend Config
|
|-- services/                          # LOGICA DE NEGOCIO (Constructores)
|   |-- pages.service.ts              # Fetch de data por slug
|   |-- payments.service.ts           # Logica de Mercado Pago
|   `-- upload.service.ts             # Compresion WebP y subida a Storage
|
|-- types/                             # Definiciones de TypeScript (English Keys)
|   `-- index.ts                      # Interfaces de la Base de Datos
```

Arquitectura y Estructura de Carpetas - Escuela de Musica Barrial

```
|  
`-- middleware.ts          # Seguridad y proteccion de rutas /admin
```

Arquitectura y Estructura de Carpetas - Escuela de Musica Barrial

3. Detalle de Servicios Clave

A. Dynamic Page Template:

Utiliza el slug de la URL para consultar el documento correspondiente en Firestore. Si el documento tiene un array de "sections", el componente itera sobre ellos y renderiza el contenido.

B. Pipeline de Imagenes:

El upload.service.ts intercepta el archivo en el navegador, lo procesa mediante Canvas o librerias de compresion, lo transforma a WebP y lo sube a Storage, retornando la URL final y el alt_text.

C. Flujo de Mercado Pago:

1. El usuario inicia el pago -> Se crea orden en Firestore (pending).
2. Redireccion a Mercado Pago.
3. Notificacion via Webhook (POST /api/payments/webhook).
4. El webhook valida el pago, actualiza Firestore (approved) y dispara mail confirmatorio.