

Estructura del Proyecto EscalaFin MVP

Arquitectura General

escalafin_mvp/	
app/	# Aplicación Next.js
app/	# App Router de Next.js
layout.tsx	# Layout principal
page.tsx	# Página home (redirecciona)
auth/	# Páginas de autenticación
login/page.tsx	# Página de login
register/page.tsx	# Página de registro
admin/	# Dashboard Admin
dashboard/page.tsx	
asesor/	# Dashboard Asesor
dashboard/page.tsx	
cliente/	# Dashboard Cliente
dashboard/page.tsx	
api/	# API Routes
auth/	# NextAuth endpoints
signup/	# Registro de usuarios
components/	# Componentes React
auth/	# Componentes de autenticación
dashboards/	# Dashboards por rol
ui/	# Componentes UI reutilizables
providers.tsx	# Providers de contexto
auth-wrapper.tsx	# Wrapper de autenticación
lib/	# Utilidades y configuración
auth.ts	# Configuración NextAuth
db.ts	# Cliente Prisma
prisma/	# Configuración de base de datos
schema.prisma	# Esquema de la BD
scripts/seed.ts	# Datos de prueba
middleware.ts	# Middleware de autenticación
.env	# Variables de entorno
package.json	# Dependencias
README.md	# Documentación principal
GUIA_LOGIN_Y_USO.md	# Esta guía
ESTRUCTURA_PROYECTO.md	# Este archivo

Componentes Clave

Autenticación

- **NextAuth.js:** Sistema de autenticación
- **Roles:** ADMIN, ASESOR, CLIENTE
- **Protección:** Middleware y AuthWrapper
- **Base de datos:** PostgreSQL con Prisma

Dashboards

- **AdminDashboard:** Métricas generales, solicitudes pendientes
- **AsesorDashboard:** Cartera de clientes, solicitudes enviadas
- **ClienteDashboard:** Préstamos activos, historial de pagos



Base de Datos

```
-- Tablas principales
users          # Usuarios del sistema
clients        # Información de clientes
credit_applications # Solicitudes de crédito
loans          # Préstamos activos
amortization_schedule # Tabla de amortización
payments       # Historial de pagos
```



UI/UX

- **Framework:** Tailwind CSS
- **Componentes:** Shadcn UI
- **Iconos:** Lucide React
- **Responsive:** Mobile-first design
- **Notificaciones:** Sonner



Tecnologías Utilizadas

Frontend:

- **Next.js 14** (App Router)
- **React 18** (TypeScript)
- **Tailwind CSS**
- **Shadcn UI**
- **Framer Motion** (animaciones)

Backend:

- **Next.js API Routes**
- **NextAuth.js** (autenticación)
- **Prisma ORM** (base de datos)
- **PostgreSQL**
- **bcryptjs** (hashing passwords)

Desarrollo:

- **TypeScript** (tipado estático)
- **ESLint** (linting)
- **Prettier** (formateo)
- **Yarn** (gestor de paquetes)







Dependencias Principales

```
{
  "next": "14.2.28",
  "react": "18.2.0",
  "next-auth": "4.24.11",
  "@prisma/client": "6.7.0",
  "tailwindcss": "3.3.3",
  "@radix-ui/react-*": "varias",
  "lucide-react": "0.446.0",
  "bcryptjs": "2.4.3"
}
```

Scripts Disponibles

```
yarn dev      # Desarrollo
yarn build    # Build de producción
yarn start    # Servidor de producción
yarn lint     # Linting
yarn prisma   # Comandos de Prisma
```

Seguridad Implementada

-  **Hash de contraseñas** (bcrypt)
-  **Middleware de autenticación**
-  **Protección de rutas por rol**
-  **Validación de entrada**
-  **Cookies seguras**
-  **Sanitización de datos**

Escalabilidad

Preparado para:

- Integración con APIs externas
- Sistema de notificaciones
- PWA para móviles
- Reportes avanzados
- Automatización con webhooks

Estructura modular permite:

- Agregar nuevos roles
- Extender funcionalidades
- Integrar servicios externos
- Crear módulos adicionales

Rutas del Sistema

Públicas:

- `/` → Redirige según autenticación
- `/auth/login` → Página de login
- `/auth/register` → Página de registro

Protegidas:

- `/admin/dashboard` → Solo ADMIN
- `/asesor/dashboard` → ASESOR + ADMIN
- `/cliente/dashboard` → CLIENTE + ADMIN

API:

- `/api/auth/*` → NextAuth endpoints
- `/api/signup` → Registro de usuarios

Datos y Estados

Gestión de Estado:

- **NextAuth Session** → Estado de usuario
- **React State** → Estado local
- **Prisma Client** → Estado de BD

Persistencia:

- **PostgreSQL** → Datos principales
- **Cookies** → Sesión de usuario
- **Local Storage** → Preferencias UI

Flujo de Datos

```

Usuario → Login → NextAuth → Prisma → PostgreSQL
                        ↓
Dashboard → React Components → API Routes → DB
                        ↓
UI Updates ← State Management ← Response
  
```

Este proyecto está diseñado para ser fácil de entender, mantener y expandir, siguiendo las mejores prácticas de desarrollo web moderno.