



Estructura Técnica Actualizada - EscalaFin MVP



Fecha de Actualización: Septiembre 24, 2025



Resumen Ejecutivo

EscalaFin es una **PWA (Progressive Web Application)** completa para la gestión integral de créditos y préstamos, desarrollada con **Next.js 14**, **TypeScript**, y **PostgreSQL**. El sistema soporta múltiples roles, integración con servicios de pago (Openpay), notificaciones WhatsApp (EvolutionAPI), y almacenamiento en la nube (AWS S3).



Arquitectura General

Stack Tecnológico

```
Frontend: Next.js 14 (App Router) + TypeScript + Tailwind CSS + Shadcn/UI
Backend: Next.js API Routes + Prisma ORM + PostgreSQL
Auth: NextAuth.js (multi-role: ADMIN, ASESOR, CLIENTE)
Storage: AWS S3 para documentos
Payments: Openpay integration
Notifications: EvolutionAPI para WhatsApp
Analytics: Recharts + Chart.js
PWA: Service Workers + Push Notifications
```








Base de Datos

- **Motor:** PostgreSQL con Prisma ORM
- **Ubicación Schema:** /app/prisma/schema.prisma
- **Modelos:** 22 tablas principales con relaciones complejas
- **Características:** Triggers, índices optimizados, soft deletes



Estructura de Directorios

Root Directory (/home/ubuntu/escalafin_mvp/)

	app/	# Aplicación Next.js principal
	docs/	# Documentación del proyecto
	scripts/	# Scripts de utilidad y seed
	uploads/	# Archivos de ejemplo/temporal
	Dockerfile	# Configuración Docker
	docker-compose.yml	# Orquestación contenedores
	README.md	# Documentación principal

Aplicación Principal (/app/)

```

app/
├── app/                                # Next.js 14 App Router
│   ├── (auth)/                        # Rutas de autenticación
│   ├── admin/                        # Panel administrativo
│   ├── asesor/                       # Panel de asesores
│   ├── cliente/                     # Panel de clientes
│   ├── api/                         # API Routes (Backend)
│   ├── globals.css                  # Estilos globales
│   └── layout.tsx                   # Layout raíz
├── components/                      # Componentes React
│   ├── admin/                      # Componentes admin
│   ├── auth/                       # Componentes autenticación
│   ├── clients/                    # Gestión de clientes
│   ├── credit-applications/        # Solicitudes de crédito
│   ├── dashboards/                # Dashboards por rol
│   ├── files/                     # Gestión de archivos
│   ├── layout/                    # Componentes de layout
│   ├── loans/                    # Gestión de préstamos
│   ├── payments/                 # Sistema de pagos
│   ├── pwa/                      # Características PWA
│   └── ui/                        # Componentes base UI
├── lib/                             # Utilidades y servicios
│   ├── api/                      # Clientes API
│   ├── auth.ts                   # Configuración NextAuth
│   ├── aws-config.ts             # Configuración AWS S3
│   ├── db.ts                     # Conexión base de datos
│   ├── openpay.ts                # Integración Openpay
│   ├── evolution-api.ts          # WhatsApp API
│   └── utils.ts                  # Utilidades generales
├── prisma/                         # Configuración Prisma
│   ├── schema.prisma             # Esquema de BD
│   └── migrations/               # Migraciones
└── public/                         # Assets estáticos
    ├── icons/                   # Iconos PWA
    ├── sw.js                    # Service Worker
    └── manifest.json             # Manifiesto PWA
  
```

API Routes (Backend)

Estructura de APIs (/app/api/)







```

api/
├── admin/                                # APIs administrativas
│   ├── evolution-api/                  # Configuración WhatsApp
│   ├── message-recharges/              # Gestión recargas
│   ├── modules/                        # Gestión de módulos
│   ├── storage/                        # Configuración almacenamiento
│   ├── users/                          # Gestión de usuarios
│   └── whatsapp/                       # Configuración WhatsApp
├── analytics/                          # APIs de analíticas
│   ├── general/                       # Métricas generales
│   ├── kpis/                          # Indicadores clave
│   └── timeseries/                    # Datos temporales
├── auth/                               # Autenticación
│   └── [...nextauth]/                 # NextAuth endpoints
├── clients/                           # Gestión clientes
│   ├── [id]/                          # CRUD por ID
│   ├── migrate/                       # Migración datos
│   ├── route.ts                       # CRUD general
│   ├── search/                        # Búsqueda clientes
│   └── sync/                          # Sincronización
├── credit-applications/                # Solicitudes crédito
│   ├── [id]/                          # CRUD por ID
│   └── route.ts                       # CRUD general
├── files/                             # Gestión archivos
│   ├── [id]/                          # CRUD por ID
│   ├── list/                          # Listado archivos
│   ├── serve/                         # Servir archivos
│   └── upload/                        # Subida archivos
├── loans/                             # Gestión préstamos
│   ├── [id]/                          # CRUD por ID
│   ├── route.ts                       # CRUD general
│   └── search/                        # Búsqueda préstamos
├── notifications/                     # Sistema notificaciones
│   └── route.ts                       # CRUD notificaciones
├── payments/                          # Sistema de pagos
│   ├── cash/                          # Pagos en efectivo
│   ├── openpay/                       # Integración Openpay
│   ├── sync/                          # Sincronización
│   └── transactions/                  # Transacciones
├── personal-references/                # Referencias personales
│   ├── [id]/                          # CRUD por ID
│   └── route.ts                       # CRUD general
├── reports/                           # Generación reportes
│   ├── collections/                  # Reportes cobranza
│   ├── due-loans/                    # Préstamos vencidos
│   └── export/                        # Exportación datos
├── webhooks/                          # Webhooks externos
│   ├── evolution-api/                 # WhatsApp webhooks
│   └── openpay/                       # Openpay webhooks

```

 whatsapp/ # Notificaciones WhatsApp
 send-notification/ # Envío notificaciones

Características de las APIs

-  **Autenticación:** Middleware NextAuth en rutas protegidas
-  **Validación:** Zod schemas para request/response
-  **Error Handling:** Manejo consistente de errores
-  **Rate Limiting:** Control de solicitudes por IP
-  **CORS:** Configurado para dominios específicos
-  **Logging:** Logs estructurados con winston

Schema de Base de Datos

Modelos Principales

```
-- Usuarios y Autenticación
users (22 models relacionados)
accounts (NextAuth)
sessions (NextAuth)
verification_tokens (NextAuth)

-- Gestión de Clientes
clients (perfil completo + scoring)
personal_references (referencias personales)

-- Sistema de Préstamos
credit_applications (solicitudes)
loans (préstamos activos)
amortization_schedule (tabla amortización)

-- Sistema de Pagos
payments (registros de pago)
payment_transactions (transacciones)
cash_collections (cobranza efectivo)

-- Gestión de Archivos
file_uploads (archivos S3)
files (sistema dual archivos)

-- Notificaciones y WhatsApp
notifications (notificaciones sistema)
whatsapp_messages (historial WhatsApp)
evolution_api_config (configuración API)

-- Sistema y Auditoría
audit_logs (logs de auditoría)
system_config (configuración sistema)
report_generations (generación reportes)

-- PWA y Módulos
pwa_modules (módulos disponibles)
module_role_permissions (permisos por rol)
module_change_logs (historial cambios)
```

Enums Principales

```
UserRole: ADMIN | ASESOR | CLIENTE  
ClientStatus: ACTIVE | INACTIVE | BLACKLISTED  
LoanStatus: ACTIVE | PAID_OFF | DEFAULTED | CANCELLED  
PaymentStatus: PENDING | COMPLETED | FAILED | CANCELLED  
PaymentMethod: CASH | BANK_TRANSFER | DEBIT_CARD | CREDIT_CARD | ONLINE  
NotificationType: PAYMENT_DUE | LOAN_APPROVED | SYSTEM_ALERT | MARKETING
```

Componentes Frontend







Estructura de Componentes

```

components/
├── admin/                                # 8 componentes administrativos
│   ├── evolution-api-config.tsx
│   ├── message-recharge-management.tsx
│   ├── module-management.tsx
│   ├── storage-config.tsx
│   ├── user-management.tsx
│   └── whatsapp-*.tsx
├── auth/                                # Autenticación
│   └── login-form.tsx
├── clients/                             # 3 componentes clientes
│   ├── client-list.tsx
│   ├── client-migration.tsx
│   └── personal-references-form.tsx
├── credit-applications/                 # 4 componentes créditos
│   ├── credit-application-details.tsx
│   ├── credit-application-form.tsx
│   ├── credit-application-review.tsx
│   └── credit-applications-list.tsx
├── dashboards/                          # 6 dashboards especializados
│   ├── admin-dashboard.tsx
│   ├── asesor-dashboard.tsx
│   ├── cliente-dashboard.tsx
│   └── enhanced-*.tsx
├── files/                               # 3 componentes archivos
│   ├── document-manager.tsx
│   ├── file-manager.tsx
│   └── file-upload.tsx
├── layout/                              # Sistema de layout
│   ├── main-layout.tsx
│   ├── desktop-navbar.tsx
│   ├── mobile-sidebar.tsx
│   └── client-only-wrapper.tsx
├── loans/                              # 6 componentes préstamos
│   ├── amortization-schedule.tsx
│   ├── loan-detail.tsx
│   ├── loan-form.tsx
│   ├── loan-list.tsx
│   └── new-loan-form.tsx
├── payments/                            # 3 componentes pagos
│   ├── cash-payment-form.tsx
│   ├── openpay-integration.tsx
│   └── payment-history.tsx
├── pwa/                                 # 5 componentes PWA
│   ├── install-banner.tsx
│   ├── offline-indicator.tsx
│   ├── pwa-install-prompt.tsx
│   └── pwa-wrapper.tsx
└── ui/                                  # 50+ componentes base
    ├── shadcn/ui components
    ├── enhanced-*.tsx
    └── custom components

```


Características Destacadas

-  **TypeScript:** 100% tipado estático
-  **Responsive:** Mobile-first design
-  **Accessible:** WCAG 2.1 AA compliance
-  **Themed:** Dark/Light mode support
-  **Performance:** Lazy loading y code splitting
-  **SSR Compatible:** Server-side rendering ready

Servicios y Librerías

Servicios Core (/lib/)

```
// Autenticación y Sesiones
auth.ts           // NextAuth configuration
audit.ts          // Audit logging system






// Integrations externas
aws-config.ts     // AWS S3 configuration
s3.ts             // S3 file operations
openpay.ts        // Payment processing
evolution-api.ts  // WhatsApp integration

// Base de datos y APIs
db.ts             // Database connection
prisma.ts         // Prisma client setup
api/              // API client libraries

// Utilidades especializadas
analytics.ts      // Analytics tracking
scoring.ts        // Credit scoring system
notifications.ts  // Notification system
pwa-utils.ts      // PWA utilities
scheduled-tasks.ts // Background jobs

// Configuración y tipos
storage-config.ts // Storage configuration
types.ts          // TypeScript definitions
utils.ts          // General utilities
```

Características de los Servicios

-  **Singleton Pattern:** Instancias únicas para conexiones
 -  **Error Resilience:** Retry logic y circuit breakers
 -  **Type Safety:** Interfaces TypeScript completas
 -  **Environment Config:** Configuración por variables de entorno
 -  **Logging:** Structured logging con contexto
-

Seguridad Implementada

Autenticación y Autorización

```
// NextAuth.js con múltiples providers
- Email/Password authentication
- Role-based access control (RBAC)
- Session management con JWT
- Password hashing con bcrypt
- CSRF protection integrado
```

Validación de Datos

```
// Zod schemas en todas las APIs
- Request validation
- Response sanitization
- SQL injection prevention
- XSS protection con DOMPurify
- Input sanitization automática
```

Headers de Seguridad





```
// Next.js security headers
- Content Security Policy (CSP)
- X-Frame-Options: DENY
- X-Content-Type-Options: nosniff
- Referrer-Policy: strict-origin
- Strict-Transport-Security (HSTS)
```

Características PWA

Configuración PWA

```
// manifest.json
{
  "name": "EscalaFin - Gestión de Créditos",
  "short_name": "EscalaFin",
  "theme_color": "#1e40af",
  "background_color": "#ffffff",
  "display": "standalone",
  "orientation": "portrait",
  "scope": "/",
  "start_url": "/"
}
```

Service Worker Features

-  **Offline Caching:** Estrategia cache-first para assets
-  **Background Sync:** Sincronización cuando hay conexión
-  **Push Notifications:** Notificaciones push nativas
-  **Update Mechanism:** Auto-update con notificación al usuario

-  **Install Prompt:** Banner de instalación personalizado

Performance y Métricas

Optimizaciones Implementadas

```
// Build Optimizations
- Next.js Bundle Analyzer
- Tree shaking automático
- Code splitting por páginas
- Dynamic imports para componentes pesados
- Image optimization con Next/Image

// Runtime Optimizations
- SWR para cache de APIs
- React Query para estado del servidor
- Virtualized lists para datos grandes
- Debounced search inputs
- Memoization con React.memo y useMemo
```

Métricas de Performance Objetivo

```
Core Web Vitals:
├─ First Contentful Paint: < 1.5s
├─ Largest Contentful Paint: < 2.5s
├─ Cumulative Layout Shift: < 0.1
├─ First Input Delay: < 100ms
└─ Time to Interactive: < 3s
```

Variables de Entorno

Configuración Requerida (.env)

```
# Base de datos
DATABASE_URL="postgresql://user:pass@host:port/db"
DIRECT_URL="postgresql://user:pass@host:port/db"

# Autenticación
NEXTAUTH_URL="https://tu-dominio.com"
NEXTAUTH_SECRET="tu-secret-super-secreto"

# AWS S3 Storage
AWS_BUCKET_NAME="tu-bucket-s3"
AWS_FOLDER_PREFIX="escalafin/"

# Openpay (Pagos)
OPENPAY_API_KEY="sk_XXXXX"
OPENPAY_BASE_URL="https://sandbox-api.openpay.mx"
OPENPAY_CLIENT_ID="XXXXX"
OPENPAY_MERCHANT_ID="XXXXX"
OPENPAY_PRIVATE_KEY="XXXXX"
OPENPAY_PUBLIC_KEY="XXXXX"
OPENPAY_USERNAME="XXXXX"
OPENPAY_PASSWORD="XXXXX"

# EvolutionAPI (WhatsApp)
EVOLUTION_API_URL="https://tu-evolution-api.com"
EVOLUTION_API_KEY="XXXXX"
EVOLUTION_INSTANCE="instance_name"
```

Deployment

Opciones de Despliegue

```
# 1. Docker Local
docker build -t escalafin .
docker run -p 3000:3000 --env-file .env escalafin

# 2. Docker Compose
docker-compose up -d

# 3. EasyPanel (Recomendado)
# Ver: GUIA_DESPLIEGUE_EASYPANEL.md

# 4. Vercel/Netlify
# Ver: DEPLOYMENT.md
```

Scripts de Utilidad

```
# Desarrollo
yarn dev           # Servidor desarrollo
yarn build         # Build producción
yarn start         # Servidor producción

# Base de datos
yarn prisma:generate # Generar cliente
yarn prisma:migrate  # Ejecutar migraciones
yarn prisma:seed      # Sembrar datos

# Utilidades
yarn lint           # Linter
yarn type-check     # Verificación TypeScript
```



Roadmap Técnico

Q4 2025 - Optimizaciones

- [] **Performance:** Implement Redis caching layer
- [] **Security:** Add rate limiting per endpoint
- [] **PWA:** Implement background sync for payments
- [] **Analytics:** Enhanced reporting with real-time dashboards





Q1 2026 - Nuevas Características

- [] **Mobile App:** React Native companion app
- [] **AI/ML:** Credit scoring con machine learning
- [] **Multi-tenancy:** Support para múltiples instituciones
- [] **API Public:** RESTful API para integraciones externas



Soporte Técnico

Contacto

-  **Email:** soporte.tecnico@escalafin.com
-  **Discord:** [Canal de soporte técnico]
-  **Docs:** [Documentación completa]
-  **Issues:** [GitHub Issues]

Guías de Troubleshooting

- [TROUBLESHOOTING.md](#) (./docs/TROUBLESHOOTING.md)
- [API_REFERENCE.md](#) (./docs/API_REFERENCE.md)
- [DATABASE_GUIDE.md](#) (./docs/DATABASE_GUIDE.md)
- [DEPLOYMENT_ISSUES.md](#) (./docs/DEPLOYMENT_ISSUES.md)

🔥 ****Sistema Técnico Actualizado y Listo para Producción**** **📅 Última actualización****: Septiembre 24, 2025** **⚡ Estado****: Funcional y Optimizado** **🚀 Próximo release****: Q4 2025**