











Guía Completa de Importación EscalaFin 2025 (ACTUALIZADA)

Información del Sistema

EscalaFin es un sistema completo de gestión de préstamos y créditos con **navegación sidebar sticky implementada**, optimizado para desktop y mobile. Esta guía actualizada incluye todas las mejoras más recientes.

Características Principales

-  **Sidebar Navegacional Sticky** para desktop
 -  **Navegación Móvil Optimizada** con Sheet/Drawer
 -  **Sistema de Módulos PWA** dinámico
 -  **Dark/Light Mode** completo
 -  **Autenticación Multi-Rol** (ADMIN/ASESOR/CLIENTE)
 -  **Integración OpenPay** para pagos
 -  **Notificaciones WhatsApp** vía EvolutionAPI
 -  **Almacenamiento AWS S3** para archivos
 -  **Dashboard Analítico** con métricas en tiempo real
 -  **PWA Ready** con Service Workers
-

Stack Tecnológico




Tecnología	Versión	Propósito
Next.js	14.2.28	Framework React
TypeScript	5.2.2	Tipado estático
Prisma	6.7.0	ORM y base de datos
PostgreSQL	15+	Base de datos
NextAuth.js	4.24.11	Autenticación
Tailwind CSS	3.3.3	Estilos
Shadcn/ui	Latest	Componentes UI
Framer Motion	10.18.0	Animaciones
AWS SDK v3	Latest	Almacenamiento S3
Yarn	4+	Gestor de paquetes

Prerrequisitos del Sistema

Entorno Local

```
# Versiones mínimas requeridas
Node.js >= 18.17.0
Yarn >= 4.0.0
PostgreSQL >= 15.0
Git >= 2.34.0
```

Servicios Externos Necesarios

-  **Base de datos PostgreSQL**
-  **AWS S3** (almacenamiento de archivos)
-  **OpenPay** (procesamiento de pagos)
-  **EvolutionAPI** (WhatsApp Business)



Paso 1: Importación del Proyecto

1.1 Clonación desde GitHub

```
# Clonar el repositorio
git clone https://github.com/tu-usuario/escalafin-mvp.git
cd escalafin-mvp

# Verificar estructura
ls -la
```

1.2 Importación a DeepAgent

```
# En DeepAgent, usar el comando:
"Import this project from GitHub: https://github.com/tu-usuario/escalafin-mvp"

# O subir directamente el ZIP del proyecto
```



Paso 2: Configuración del Entorno

2.1 Instalación de Dependencias

```
cd app
yarn install

# Verificar instalación
yarn --version
node --version
```

2.2 Variables de Entorno (.env.local)

```
# Base de datos
DATABASE_URL="postgresql://usuario:password@host:5432/escalafin_db"

# Autenticación
NEXTAUTH_URL="http://localhost:3000"
NEXTAUTH_SECRET="tu-secret-super-seguro-aqui"

# AWS S3
AWS_ACCESS_KEY_ID="tu-access-key"
AWS_SECRET_ACCESS_KEY="tu-secret-key"
AWS_BUCKET_NAME="escalafin-storage"
AWS_REGION="us-east-1"
AWS_FOLDER_PREFIX="uploads/"

# OpenPay
OPENPAY_ID="tu-merchant-id"
OPENPAY_PRIVATE_KEY="tu-private-key"
OPENPAY_PUBLIC_KEY="tu-public-key"
OPENPAY_PRODUCTION=false

# EvolutionAPI
EVOLUTION_API_URL="https://tu-instancia.evolutionapi.com"
EVOLUTION_API_TOKEN="tu-token-api"
EVOLUTION_INSTANCE="escalafin"

# Configuración del sistema
NEXT_PUBLIC_APP_NAME="EscalaFin"
NEXT_PUBLIC_APP_VERSION="2.1.0"
NEXT_PUBLIC_ENVIRONMENT="production"
```

Paso 3: Configuración de Base de Datos

3.1 Creación de Base de Datos

```
-- PostgreSQL
CREATE DATABASE escalafin_db;
CREATE USER escalafin_user WITH PASSWORD 'tu_password_seguro';
GRANT ALL PRIVILEGES ON DATABASE escalafin_db TO escalafin_user;
```

3.2 Migración de Schema

```
# Generar cliente Prisma
yarn prisma generate

# Aplicar migraciones
yarn prisma db push

# Sembrar datos iniciales
yarn prisma db seed
```

3.3 Verificación de Tablas

```
-- Verificar que las tablas fueron creadas
dt
-- Deberías ver: User, Client, Loan, Payment, PWAModule, etc.
```



Paso 4: Inicialización de Servicios

4.1 AWS S3 Setup

```
# Crear bucket
aws s3 mb s3://escalafin-storage --region us-east-1

# Configurar CORS
aws s3api put-bucket-cors --bucket escalafin-storage --cors-configuration file://
cors.json
```

4.2 Configuración OpenPay

```
// Configuración de pruebas
const openpay = new OpenPay({
  id: process.env.OPENPAY_ID,
  pk: process.env.OPENPAY_PRIVATE_KEY,
  production: false
});
```

4.3 EvolutionAPI Setup

```
# Test de conexión
curl -X GET "https://tu-instancia.evolutionapi.com/instance/list" \
-H "Authorization: Bearer tu-token"
```



Paso 5: Ejecución del Proyecto

5.1 Modo Desarrollo

```
cd app
yarn dev

# La aplicación estará disponible en:
# http://localhost:3000
```

5.2 Modo Producción

```
# Build de producción
yarn build

# Iniciar servidor
yarn start
```

5.3 Verificación de Funcionalidad

```
# Ejecutar tests
yarn test

# Verificar TypeScript
yarn tsc --noEmit

# Linting
yarn lint
```

Paso 6: Configuración de Usuarios

6.1 Usuario Administrador Inicial

```
INSERT INTO "User" (id, name, email, password, role, createdAt, updatedAt)
VALUES (
  'admin-001',
  'Administrador',
  'admin@escalafin.com',
  '$2a$12$hash_de_password', -- bcrypt hash of 'admin123'
  'ADMIN',
  NOW(),
  NOW()
);
```

6.2 Configuración de Módulos PWA

```
-- Habilitar módulos principales
UPDATE "PWAModule" SET status = 'ENABLED'
WHERE "moduleKey" IN (
  'client_list', 'loan_list', 'payment_history',
  'notifications_inapp', 'dashboard_overview'
);
```

Paso 7: Navegación y UI

7.1 Sidebar Navegacional

La nueva navegación incluye:






Desktop:

- Sidebar sticky colapsible/expandible
- Agrupación por categorías (Gestión, Reportes, Comunicación, etc.)
- Indicador de página activa
- Avatar y perfil de usuario

Mobile:

- Header con menú hamburguesa
- Sheet/Drawer deslizante
- Navegación optimizada para touch

7.2 Estructura de Navegación

-  Dashboard
-  Gestión
 - Clientes
 - Préstamos
 - Pagos
 - Usuarios
-  Reportes
 - Portfolio
 - Cobranza
 - Analíticos
-  Comunicación
 - WhatsApp
 - Notificaciones
 - Cobranza Móvil
-  Configuración
 - Módulos PWA
 - Sistema
 - API Externa

Paso 8: Testing y Validación

8.1 Lista de Verificación

- [] ☒ Base de datos conecta correctamente
- [] ☒ Autenticación funciona (login/logout)
- [] ☒ Sidebar navegacional responde correctamente
- [] ☒ Módulos PWA se cargan dinámicamente
- [] ☒ Tema claro/oscuro funciona
- [] ☒ Responsive design funciona en mobile
- [] ☒ AWS S3 uploads funcionan
- [] ☒ OpenPay integración activa
- [] ☒ WhatsApp notifications operativas
- [] ☒ Dashboard métricas en tiempo real

8.2 URLs de Prueba

```
http://localhost:3000/auth/login
http://localhost:3000/admin/dashboard
http://localhost:3000/clients
http://localhost:3000/loans
http://localhost:3000/admin/modules
```

Paso 9: Despliegue

9.1 Build Final

```
# Build optimizado
yarn build

# Verificar build
yarn start
```

9.2 Variables de Producción

```
NODE_ENV=production
NEXTAUTH_URL=https://tu-dominio.com
DATABASE_URL=postgresql://prod-user:pass@prod-host:5432/escalafin_prod
```

Paso 10: Mantenimiento

10.1 Actualizaciones

```
# Actualizar dependencias
yarn upgrade

# Verificar vulnerabilidades
yarn audit

# Aplicar patches
yarn patch
```

10.2 Monitoreo





- Logs de aplicación: `yarn logs`
 - Métricas de base de datos: Prisma Studio
 - Monitoreo S3: AWS CloudWatch
 - Análisis de errores: Sentry (opcional)
-

Soporte y Troubleshooting

Errores Comunes

1. **Error de conexión DB:** Verificar DATABASE_URL
2. **Módulos no cargan:** Revisar PWAModule en DB
3. **S3 upload falla:** Verificar AWS credentials
4. **Sidebar no aparece:** Verificar breakpoints CSS

Contacto

-  Email: soporte@escalafin.com
 -  Documentación: /docs
 -  Issues: GitHub Issues
 -  Chat: Discord/Slack
-



Checklist Final

- [] Proyecto importado correctamente
 - [] Dependencias instaladas
 - [] Base de datos configurada y migrada
 - [] Servicios externos configurados
 - [] Variables de entorno establecidas
 - [] Navegación sidebar funcionando
 - [] Autenticación operativa
 - [] Módulos PWA habilitados
 - [] Tests pasando
 - [] Build de producción exitoso
 - [] Documentación revisada
-

Última actualización: Septiembre 2025

Versión EscalaFin: 2.1.0

Incluye: Sidebar Navigation, PWA Modules, S3 Integration