

# Guía Completa de Despliegue en Coolify - EscalaFin MVP

## Introducción

Esta guía te llevará paso a paso por el proceso de despliegue de EscalaFin MVP en Coolify, desde la configuración inicial hasta la puesta en producción.

## Requisitos Previos

- Servidor con Coolify instalado (mínimo 2GB RAM, 2 CPU cores)
- Dominio configurado (ej: `app.escalafin.com` , `adm.escalafin.com` )
- Acceso SSH al servidor
- Cuenta de GitHub con el repositorio configurado

## Paso 1: Preparar el Repositorio

### 1.1 Verificar Archivos Necesarios

Asegúrate de que tu repositorio tenga estos archivos:

```
escalafin_mvp/  
├── Dockerfile.production      # Dockerfile optimizado con standalone  
├── docker-compose.coolify.yml # Configuración de servicios  
├── start.sh                  # Script de inicio  
├── healthcheck.sh           # Health check endpoint  
├── app/  
│   ├── next.config.js       # Con output: "standalone"  
│   ├── package.json  
│   └── prisma/  
│       └── schema.prisma  
└── .gitignore
```

### 1.2 Actualizar next.config.js

Tu `next.config.js` debe incluir la configuración standalone:

```
const path = require('path');

/** @type {import('next').NextConfig} */
const nextConfig = {
  distDir: process.env.NEXT_DIST_DIR || '.next',
  output: 'standalone', // ← CRÍTICO para Coolify
  experimental: {
    outputFileTracingRoot: path.join(__dirname, '..'),
  },
  eslint: {
    ignoreDuringBuilds: true,
  },
  typescript: {
    ignoreBuildErrors: false,
  },
  images: { unoptimized: true },
};

module.exports = nextConfig;
```

## 1.3 Subir Cambios a GitHub

```
cd /home/ubuntu/escalafin_mvp
git add .
git commit -m "Configure for Coolify deployment with standalone output"
git push origin main
```



## Paso 2: Configurar Coolify

### 2.1 Crear Nuevo Proyecto

1. Accede a tu panel de Coolify: <https://adm.escalafin.com>
2. Click en **" + Add "** en Projects
3. Nombra tu proyecto: `escalafin-mvp`
4. Click en **"Create"**

### 2.2 Agregar Source (GitHub)

1. Ve a **Sources** en el sidebar
2. Click **" + Add Source "**
3. Selecciona **"GitHub"**
4. Autoriza Coolify en GitHub
5. Selecciona el repositorio `qhosting/escalafin-mvp`

### 2.3 Configurar la Aplicación

#### 2.3.1 Datos Básicos

- **Name:** `escalafin-app`
- **Build Type:** `Dockerfile`
- **Dockerfile:** `Dockerfile.production`
- **Branch:** `main`
- **Repository:** `qhosting/escalafin-mvp`
- **Build Pack:** `Docker`

- **Port:** 3000

### 2.3.2 Variables de Entorno

Configura estas variables en Coolify:

```
# Base de datos
DATABASE_URL=postgresql://escalafin:PASSWORD@db:5432/escalafin
POSTGRES_USER=escalafin
POSTGRES_PASSWORD=[GENERAR PASSWORD SEGURO]
POSTGRES_DB=escalafin
DATABASE_HOST=db
DATABASE_PORT=5432

# NextAuth
NEXTAUTH_URL=https://app.escalafin.com
NEXTAUTH_SECRET=[GENERAR CON: openssl rand -base64 32]

# OpenPay
OPENPAY_MERCHANT_ID=tu_merchant_id
OPENPAY_PRIVATE_KEY=tu_private_key
OPENPAY_PUBLIC_KEY=tu_public_key
OPENPAY_BASE_URL=https://sandbox-api.openpay.mx

# AWS S3
AWS_ACCESS_KEY_ID=tu_access_key
AWS_SECRET_ACCESS_KEY=tu_secret_key
AWS_BUCKET_NAME=escalafin-files
AWS_REGION=us-east-1
AWS_FOLDER_PREFIX=production/

# Evolution API (WhatsApp)
EVOLUTION_API_URL=https://tu-evolution-api.com
EVOLUTION_API_TOKEN=tu_token
EVOLUTION_INSTANCE_NAME=escalafin

# Configuración de deployment
NODE_ENV=production
NEXT_TELEMETRY_DISABLED=1
PORT=3000
HOSTNAME=0.0.0.0
SKIP_MIGRATIONS=false
RUN_SEED=false
```

## Paso 3: Configurar Base de Datos

### 3.1 Crear Servicio PostgreSQL

1. En tu proyecto, click **" + Service "**
2. Selecciona **"PostgreSQL"**
3. Configura:
  - **Name:** escalafin-db
  - **Version:** 14
  - **Username:** escalafin
  - **Password:** [mismo que en variables]
  - **Database:** escalafin
  - **Port:** 5432

4. Click **“Create”**

## 3.2 Crear Servicio Redis (Opcional)

1. Click **“+ Service”**
2. Selecciona **“Redis”**
3. Configura:
  - **Name:** escalafin-redis
  - **Version:** 7
  - **Password:** [generar password]
  - **Port:** 6379
4. Agrega a variables de entorno:

```
REDIS_URL=redis://:PASSWORD@escalafin-redis:6379
REDIS_PASSWORD=tu_password
```

## Paso 4: Configurar Dominio

### 4.1 Configurar DNS

En tu proveedor de DNS (Cloudflare, etc.):

Type	Name	Value	TTL
A	app.escalafin	[IP_DEL_SERVIDOR]	Auto
A	adm.escalafin	[IP_DEL_SERVIDOR]	Auto

### 4.2 Configurar en Coolify

1. Ve a tu aplicación en Coolify
2. Sección **“Domains”**
3. Agrega dominio: `app.escalafin.com`
4. Habilita **“Generate SSL”** (Let's Encrypt)
5. Click **“Save”**

## Paso 5: Desplegar

### 5.1 Primera Build

1. Ve a tu aplicación en Coolify
2. Click en **“Deploy”**
3. Observa los logs en tiempo real
4. Espera a que termine (puede tomar 5-10 minutos)

## 5.2 Verificar Deployment

```
# Verificar que los contenedores están corriendo
docker ps | grep escalafin

# Ver logs de la aplicación
docker logs -f [CONTAINER_ID]

# Verificar conectividad
curl -I https://app.escalafin.com/api/health
```

## 5.3 Ejecutar Migraciones Manualmente (si es necesario)

```
# Conectarse al contenedor
docker exec -it escalafin-app sh

# Ejecutar migraciones
npx prisma migrate deploy

# Verificar estado
npx prisma migrate status
```



## Paso 6: Monitoreo y Mantenimiento

### 6.1 Verificar Health Checks

Coolify

fy automáticamente verifica la salud de tu aplicación cada 30 segundos usando el endpoint `/api/health`.

### 6.2 Ver Logs

1. Ve a tu aplicación en Coolify
2. Click en **“Logs”**
3. Filtra por servicio (app, db, redis)

### 6.3 Configurar Backups de Base de Datos

1. Ve al servicio PostgreSQL
2. Click en **“Backups”**
3. Configura:
  - **Frequency:** Daily
  - **Retention:** 7 days
  - **Time:** 02:00 AM



## Paso 7: Despliegues Continuos

### 7.1 Configurar Auto-Deploy desde GitHub

1. Ve a tu aplicación en Coolify
2. Sección **“General”**
3. Habilita **“Automatic Deployment”**
4. Selecciona branch: `main`

5. Cada push a `main` desplegará automáticamente

## 7.2 Deploy Manual

```
# En Coolify UI, simplemente click en "Deploy"
# O vía API:
curl -X POST "https://adm.escalafin.com/api/v1/deploy" \
  -H "Authorization: Bearer YOUR_API_TOKEN" \
  -d '{"project": "escalafin-mvp"}'
```

## Troubleshooting

### Problema: Build falla

**Solución:**

```
# Verificar logs de build
docker logs escalafin-app-build

# Limpiar caché de Docker
docker builder prune -a

# Rebuild sin caché
# En Coolify: Settings > Clear Build Cache > Deploy
```

### Problema: Base de datos no conecta

**Solución:**

```
# Verificar que PostgreSQL está corriendo
docker ps | grep postgres

# Verificar conectividad desde el contenedor de la app
docker exec -it escalafin-app sh
pg_isready -h db -p 5432 -U escalafin

# Verificar DATABASE_URL en variables de entorno
docker exec escalafin-app env | grep DATABASE_URL
```

### Problema: 502 Bad Gateway

**Solución:**

```
# Verificar que la app está escuchando en el puerto correcto
docker exec escalafin-app netstat -tlnp | grep 3000

# Verificar health check
curl http://localhost:3000/api/health

# Reiniciar el servicio
# En Coolify: Click en "Restart"
```

### Problema: Migraciones fallan

**Solución:**

```
# Conectarse al contenedor
docker exec -it escalafin-app sh

# Ver estado de migraciones
npx prisma migrate status

# Resetear (⚠ SOLO EN DESARROLLO)
npx prisma migrate reset

# En producción, aplicar manualmente
npx prisma migrate deploy --schema=./prisma/schema.prisma
```



## Checklist de Verificación Post-Deployment

- [ ] Aplicación accesible en <https://app.escalafin.com>
- [ ] SSL/HTTPS funcionando correctamente
- [ ] `/api/health` retorna status 200
- [ ] Login/registro funcionan
- [ ] Base de datos con datos iniciales
- [ ] Subida de archivos a S3 funciona
- [ ] Notificaciones de WhatsApp (si aplica)
- [ ] Backups automáticos configurados
- [ ] Monitoreo activo
- [ ] Auto-deploy desde GitHub configurado



## Seguridad Post-Deployment

1. **Cambiar todas las contraseñas por defecto**
2. **Habilitar firewall:** Solo puertos 80, 443, 22
3. **Configurar fail2ban** para proteger SSH
4. **Habilitar HTTPS únicamente** (redirigir HTTP → HTTPS)
5. **Configurar CORS** apropiadamente
6. **Revisar variables de entorno** sensibles



## Soporte

- **Documentación Coolify:** <https://coolify.io/docs>
- **GitHub Issues:** <https://github.com/qhosting/escalafin-mvp/issues>
- **Email:** [soporte@escalafin.com](mailto:soporte@escalafin.com)

**Última actualización:** Octubre 2025

**Versión:** 1.0.0

**Autor:** DevOps Team - EscalaFin