⚙️ CONFIGURACIÓN CORRECTA PARA EASYPANEL

🎯 PASO 1: Crear Aplicación en EasyPanel

1.1 Información Básica

```
Nombre: escalafin-mvp
Tipo: App
Deployment Method: GitHub
```

1.2 Configuración del Repositorio

```
Provider: GitHub
Owner: qhosting
Repository: escalafin-mvp
Branch: main
```

1.3 Configuración de Build

```
Build Method: Dockerfile
Dockerfile: Dockerfile.step3-full
Build Path: /app           ⚠️ CRÍTICO: NO DEJAR VACÍO
Build Context: .
```

🔐 PASO 2: Variables de Entorno

2.1 Variables Obligatorias

```
# === DATABASE ===
DATABASE_URL=postgresql://USER:PASSWORD@HOST:5432/escalafin?schema=public
# Reemplazar USER, PASSWORD, HOST con tus valores de EasyPanel PostgreSQL

# === NEXTAUTH ===
NEXTAUTH_URL=https://escalafin-mvp.TUDOMINIO.com
# Reemplazar con tu dominio de EasyPanel

NEXTAUTH_SECRET=GENERAR_SECRETO_SEGURO_AQUI
# Generar con: openssl rand -base64 32

# === AWS S3 ===
AWS_BUCKET_NAME=tu-bucket-name
AWS_FOLDER_PREFIX=escalafin/
AWS_REGION=us-east-1
AWS_ACCESS_KEY_ID=tu-access-key
AWS_SECRET_ACCESS_KEY=tu-secret-key

# === OPENPAY ===
OPENPAY_MERCHANT_ID=tu-merchant-id
OPENPAY_PRIVATE_KEY=tu-private-key
OPENPAY_PUBLIC_KEY=tu-public-key
OPENPAY_API_ENDPOINT=https://sandbox-api.openpay.mx
OPENPAY_IS_SANDBOX=true

# === EVOLUTIONAPI ===
EVOLUTION_API_URL=tu-url-evolutionapi
EVOLUTION_API_KEY=tu-api-key
EVOLUTION_API_INSTANCE=tu-instance

# === BUILD VARS ===
NODE_ENV=production
NEXT_OUTPUT_MODE=standalone
SKIP_ENV_VALIDATION=1
NEXT_TELEMETRY_DISABLED=1
PORT=3000
```

2.2 Generar NEXTAUTH_SECRET

En tu terminal local o servidor:

```
openssl rand -base64 32
```

Copiar el resultado a la variable `NEXTAUTH_SECRET`

🗄️ PASO 3: Crear Base de Datos PostgreSQL

3.1 En EasyPanel:

1. Ir a “Services” → “Add Service”
2. Seleccionar “PostgreSQL”

3. Configurar:

```
Name: escalafin-db
Version: 16
Database: escalafin
User: escalafin_user
Password: <generar-password-seguro>
```

3.2 Obtener DATABASE_URL

Después de crear, EasyPanel te dará la connection string:

```
postgresql://escalafin_user:PASSWORD@escalafin-db:5432/escalafin?schema=public
```

Copiar esto a la variable `DATABASE_URL` de tu app.

📦 PASO 4: Configuración de Recursos

4.1 Recursos Recomendados

```
CPU: 1-2 cores
Memory: 2GB minimum (recomendado 4GB)
Replicas: 1 (escalar después si es necesario)
```

4.2 Port Mapping

```
Container Port: 3000
Protocol: HTTP
```

4.3 Health Check

```
Path: /api/health
Port: 3000
Interval: 30s
Timeout: 10s
Initial Delay: 40s
```

🚀 PASO 5: Orden de Deployment

✅ ORDEN CORRECTO:

1. **Crear PostgreSQL** (escalafin-db)
2. **Esperar** a que PostgreSQL esté completamente iniciado
3. **Configurar todas las variables de entorno** en la app
4. **Verificar** que DATABASE_URL apunta correctamente a escalafin-db
5. **Deploy** de la aplicación

❌ ERRORES COMUNES:

- Deployar la app antes que la DB esté lista
- Olvidar configurar NEXTAUTH_URL con el dominio correcto
- No configurar NEXTAUTH_SECRET
- Dejar “Build Path” vacío
- Usar Dockerfile incorrecto

🔍 PASO 6: Verificación Post-Deploy

6.1 Logs del Contenedor

En EasyPanel, ir a tu app → “Logs” y verificar:

```
✅ Esperando PostgreSQL...
✅ Aplicando migraciones Prisma...
✅ Ejecutando seed inicial...
✅ Next.js started on http://0.0.0.0:3000
✅ Ready in XXXms
```

6.2 Health Check

```
curl https://tu-dominio.com/api/health
```

Debe responder:

```
{
  "status": "ok",
  "timestamp": "2025-10-18T..."
}
```

6.3 Página de Login

Visitar: `https://tu-dominio.com/login`

Debe mostrar la página de login sin errores.

🐛 TROUBLESHOOTING

Error: “Build Path required”

Solución: En configuración de build, colocar `/app` en “Build Path”

Error: “server.js not found”

Solución: Verificar que estás usando `Dockerfile.step3-full` y que NEXT_OUTPUT_MODE=standalone

Error: “Cannot connect to database”

Solución:
1. Verificar que PostgreSQL está running

2. Verificar DATABASE_URL correcto
3. Esperar 1-2 minutos para que DB termine de iniciar

Error: “NEXTAUTH_SECRET is not set”

Solución: Generar con `openssl rand -base64 32` y agregarlo a variables de entorno

Error: “Failed to generate Prisma Client”

Solución: Verificar que prisma/schema.prisma existe en el repo

---

✅ CHECKLIST FINAL ANTES DE DEPLOY

- [ ] PostgreSQL creado y running
- [ ] Todas las variables de entorno configuradas
- [ ] NEXTAUTH_SECRET generado
- [ ] DATABASE_URL apunta al servicio correcto
- [ ] NEXTAUTH_URL con dominio correcto
- [ ] Build Method: Dockerfile
- [ ] Dockerfile: Dockerfile.step3-full
- [ ] Build Path: /app
- [ ] Tests locales pasaron (test-dockerfiles.sh)
- [ ] Código pushed a GitHub main branch

🎉 ¡LISTO PARA DEPLOY!

Si todos los items del checklist están ✅, puedes proceder con confianza al deploy en EasyPanel.
