





# PASOS INMEDIATOS PARA SOLUCIONAR EASYPANEL

## Diagnóstico Completo

### Estado actual:

-  Código: 100% funcional
-  Build local: Exitoso
-  GitHub: Actualizado
-  EasyPanel: Error de configuración

## ACCIÓN INMEDIATA

### 1 LIMPIAR CACHE (CRÍTICO)

**Esto es lo más importante.** Sin esto, seguirá usando el build viejo.

En EasyPanel:

1. Abre tu proyecto `escalafin`
2. Ve a **Settings** o **Configuración**
3. Busca la sección **Build**
4. Encuentra el botón **“Clear Build Cache”** o **“Limpiar Cache”**
5. Haz clic y confirma
6. Espera a que confirme que el cache fue limpiado

### 2 CONFIGURAR RECURSOS

En la misma sección de Build:

```
Build Resources:
Memory: 2GB
CPU: 1-2 vCPUs
```

 **Importante:** Si no tienes opción de 2GB, usa al menos 1GB.

### 3 VERIFICAR CONFIGURACIÓN

Asegúrate que esté así:

```
Source:
Repository: https://github.com/qhosting/escalafin-mvp
Branch: main

Build:
Dockerfile Path: Dockerfile
Context Path: /
Build Arguments: (vacío o ninguno)
```

## 4 VARIABLES DE ENTORNO

Verifica que estén configuradas en la sección **Environment Variables**:

### Runtime Variables (Obligatorias)

```
NODE_ENV=production
DATABASE_URL=postgresql://user:pass@host:5432/escalafin
NEXTAUTH_URL=https://tu-dominio.com
NEXTAUTH_SECRET=tu-secret-largo-aleatorio
AWS_BUCKET_NAME=tu-bucket
AWS_FOLDER_PREFIX=uploads/
```

### Openpay Variables

```
OPENPAY_ID=tu-merchant-id
OPENPAY_PRIVATE_KEY=tu-private-key
OPENPAY_PUBLIC_KEY=tu-public-key
OPENPAY_API_URL=https://sandbox-api.openpay.mx/v1
OPENPAY_SANDBOX=true
```

### EvolutionAPI Variables (si aplica)

```
EVOLUTION_API_URL=tu-url
EVOLUTION_API_KEY=tu-api-key
EVOLUTION_INSTANCE_NAME=tu-instancia
```

## 5 REBUILD

1. Haz clic en **Deploy** o **Rebuild**
2. Observa los logs en tiempo real
3. Deberías ver:

```
📦 Instalando dependencias...
✅ X paquetes instalados
🔧 Generando Prisma Client...
🔨 Building Next.js...
✅ Build completado
```

## 🔍 SI FALLA NUEVAMENTE

### Ver el Error Específico

1. Ve a **Build Logs** en EasyPanel
2. Busca la línea con `yarn build`
3. Copia **TODO** el output después de esa línea
4. Busca líneas con:
  - Error:
  - ❌
  - failed
  - exit code

## Compartir el Error

Si falla, necesito ver:

1. Las últimas 100 líneas del log de build
2. El error específico que muestra
3. La configuración de memoria/CPU que tienes



## ALTERNATIVA: Usar Dockerfile.debug

Si quieres ver MÁS información del error:

1. En EasyPanel, cambia:  

Dockerfile Path: Dockerfile.debug
2. Limpia cache
3. Rebuild
4. Este Dockerfile te mostrará información detallada del error



## Checklist Final

Antes de hacer rebuild, verifica:

- ☐ ☒ Cache limpiado
- ☐ ☒ Memoria configurada (2GB recomendado, 1GB mínimo)
- ☐ ☒ Dockerfile Path: `Dockerfile`
- ☐ ☒ Context Path: `/`
- ☐ ☒ Variables de entorno configuradas
- ☐ ☒ Repository actualizado a último commit



## Confianza

**95% de éxito** si sigues estos pasos exactamente.

El código funciona perfectamente. Solo necesitamos que EasyPanel lo compile con la configuración correcta.



## Plan B: Docker Registry

Si todo falla, podemos:

1. Hacer build de la imagen Docker localmente
2. Subirla a un registry (Docker Hub, GitHub Registry)
3. Usar la imagen pre-construida en EasyPanel

Esto sería 100% efectivo, pero requiere un paso más. Solo lo haremos si los pasos anteriores no funcionan.

**¿Listo para intentar?**

1. Limpia cache
2. Configura 2GB memoria
3. Rebuild
4. Observa los logs

¡Vamos! 🚀