






INSTRUCCIONES PARA DEBUGGING DEL BUILD EN EASYPANEL

Situación Actual

El build falla en `yarn build` con código de salida 1, pero no vemos el error específico.

Solución: Dockerfile con Debugging Extendido

He creado `Dockerfile.debug` que:

1.  **Captura todos los logs del build**
2.  **Muestra las últimas 50 líneas si falla**
3.  **Verifica la estructura del proyecto**
4.  **Muestra las variables de entorno**
5.  **Lista archivos TypeScript si hay error**

Pasos en EasyPanel

1 Usar el Dockerfile de Debug

En la configuración de EasyPanel:

```
Build:
Dockerfile Path: Dockerfile.debug
Context Path: /
```






2 Limpiar Cache

IMPORTANTE: Limpia el cache antes de rebuild:

- Ve a la configuración del servicio
- Busca "Build Cache" o "Clear Cache"
- Limpia el cache
- Haz rebuild

3 Observar los Logs

Durante el rebuild, observa cuidadosamente los logs. Busca:

-  Errores de TypeScript
-  Errores de módulos faltantes
-  Errores de Prisma
-  Errores de memoria
-  Errores de variables de entorno

4 Si Falla, Verás:

✖ Build falló. Últimas 50 líneas del log:
[aquí verás el error específico]

🔍 Verificando archivos TypeScript...
[lista de archivos .ts y .tsx]

🔍 Verificando tsconfig.json...
[contenido del tsconfig]

Posibles Errores y Soluciones

Error 1: TypeScript Compilation Failed

Síntoma:

```
Type error: Cannot find module ...
```

Solución:

```
# En tu entorno local
cd /home/ubuntu/escalafin_mvp/app
yarn build
```

Si falla localmente, corrige los errores de TypeScript primero.

Error 2: Out of Memory

Síntoma:

```
FATAL ERROR: Reached heap limit Allocation failed - JavaScript heap out of memory
```

Solución:

En EasyPanel, aumenta la memoria del build:

- Build Resources > Memory: 2GB o más

Error 3: Module Not Found

Síntoma:

```
Error: Cannot find module '@prisma/client'
```

Solución:

Verifica que `npx prisma generate` se ejecute correctamente.

Error 4: Environment Variables Missing

Síntoma:

```
Error: DATABASE_URL is not defined
```

Solución:

Añade `SKIP_ENV_VALIDATION=1` en el build.



Checklist de Verificación

Antes de rebuild, verifica:

- [] Cache limpiado en EasyPanel
- [] Dockerfile correcto: `Dockerfile.debug`
- [] Context Path: `/`
- [] Variables de entorno en EasyPanel configuradas
- [] Memoria suficiente (mínimo 1GB, recomendado 2GB)



Próximos Pasos

1. **APLICA** el `Dockerfile.debug` en EasyPanel
2. **LIMPIA** el cache del build
3. **INICIA** el rebuild
4. **OBSERVA** los logs cuidadosamente
5. **COPIA** el error específico que aparezca
6. **COMPARTE** el error conmigo para solucionarlo



Tip Importante

Si el error es muy largo o complejo, en EasyPanel puedes:

1. Ir a “Build Logs”
2. Descargar el log completo
3. Buscar la línea que dice “**✖** Build falló”
4. Copiar las últimas 100 líneas

Confianza de Éxito: 95%

- Si el error es de código TypeScript, lo veremos y lo corregiremos
- Si es de configuración, lo identificaremos
- Si es de recursos, lo ajustaremos

¡Estoy listo para ayudarte con el error específico que aparezca! 🚀