



INFORME DE VERIFICACIÓN DE SYMLINKS

Fecha: 10 de diciembre de 2025

Repositorio: qhosting/aurum-control-center

Rama: master



RESUMEN EJECUTIVO

Se realizó una verificación exhaustiva del repositorio para identificar y corregir problemas con enlaces simbólicos (symlinks) que podrían estar causando el error reportado en Easypanel:

```
ERROR: failed to build: failed to solve: failed to read dockerfile:  
open Dockerfile: no such file or directory
```

✓ HALLAZGOS PRINCIPALES

1. Archivos Críticos - TODOS SON ARCHIVOS REALES

Archivo	Tipo	Estado Git	Tamaño	SHA
aurum-control-center/Docker-file	✓ Archivo Real	100644	997 bytes	1dd0e3f
aurum-control-center/.dockerignore	✓ Archivo Real	100644	557 bytes	57b0d57
aurum-control-center/docker-compose.yml	✓ Archivo Real	100644	871 bytes	a1caeed
aurum-control-center/package.json	✓ Archivo Real	100644	747 bytes	b4ba812
aurum-control-center/package-lock.json	✓ Archivo Real	100644	215,859 bytes	cc04d11
aurum-control-center/EASYPAN-EL_DEPLOY.md	✓ Archivo Real	100644	4,857 bytes	9d9b551
ANALIS-IS_DEPLOY.md	✓ Archivo Real	100644	10,420 bytes	c1b11ba

Nota: El modo Git `100644` confirma que son archivos regulares. Los symlinks tendrían modo `120000`.

2. Symlink Encontrado y Eliminado

- **Archivo:** `tmp` (en la raíz del repositorio)
- **Tipo:** Symlink → `/tmp/workspace_tmp`
- **Modo Git:** `120000` (confirmado como symlink)
- **Acción:** ✓ Eliminado del repositorio
- **Impacto:** Ninguno en archivos críticos de deployment

🔧 ACCIONES REALIZADAS

1. ✓ **Verificación completa de todos los archivos críticos**
 - Comando: `ls -la`, `file`, `git ls-files -s`
 - Resultado: Todos los archivos Docker y de deployment son archivos reales

2. **Búsqueda de symlinks en el repositorio**

- Comando: `find . -type l -ls`
- Resultado: Solo se encontró `tmp` (no crítico)

3. **Verificación en GitHub (repositorio remoto)**

- API: `/repos/qhosting/aurum-control-center/contents/...`
- Resultado: Todos los archivos existen y son tipo `"file"`

4. **Eliminación del symlink no crítico**

```
bash
git rm tmp
echo "tmp/" >> .gitignore
```

5. **Commit y push de cambios**

- Commit: `bf6835f - chore: Remove tmp symlink and update .gitignore`
- Push: Exitoso a `origin/master`



VERIFICACIÓN DEL DOCKERFILE

El Dockerfile en GitHub tiene contenido válido:

```
# Dockerfile optimizado para Next.js 14 con multi-stage build
FROM node:18-alpine AS base

# Instalar dependencias solo cuando sea necesario
FROM base AS deps
RUN apk add --no-cache libc6-compat
WORKDIR /app

# Copiar archivos de dependencias
COPY package.json package-lock.json* ./
RUN npm ci

# Builder stage
FROM base AS builder
WORKDIR /app
COPY --from=deps /app/node_modules ./node_modules
COPY . .
[...]
```

Confirmación: El archivo es válido y accesible en GitHub.



ANÁLISIS DEL ERROR DE EASYPANEL

Problema Reportado

```
ERROR: failed to build: failed to solve: failed to read dockerfile:
open Dockerfile: no such file or directory
```

Análisis

NO es un problema de symlinks - todos los archivos críticos son archivos reales tanto localmente como en GitHub.

Posibles Causas del Error en Easypanel

1. Configuración incorrecta del Build Context en Easypanel

- El Dockerfile está en: `aurum-control-center/Dockerfile`
- El Build Context debe ser: `aurum-control-center/`
- ⚠ Si Easypanel busca el Dockerfile en la raíz, no lo encontrará

2. Path del Dockerfile mal configurado

- Dockerfile path correcto: `aurum-control-center/Dockerfile`
- Build context: `aurum-control-center/`
- O alternativamente:
 - Dockerfile path: `Dockerfile`
 - Build context: `aurum-control-center/`

3. Caché de Easypanel

- Es posible que Easypanel tenga una versión cacheada del repositorio
- Solución: Forzar rebuild o limpiar caché

4. Permisos o sincronización de GitHub

- Verificar que Easypanel tenga acceso al repositorio
- Forzar re-fetch del repositorio

✓ ESTADO FINAL

Archivos Convertidos de Symlinks a Reales

NINGUNO - Todos los archivos críticos ya eran archivos reales desde el inicio.

Archivos que Eran Symlinks

- ✗ `tmp` → Eliminado (no era crítico para deployment)

Confirmación de Push

✓ Push exitoso al repositorio remoto:

```
To https://github.com/qhosting/aurum-control-center.git
a314472..bf6835f  master -> master
```

Verificación en GitHub

✓ Todos los archivos críticos verificados en GitHub:

- Tipo: `"file"` (no symlinks)
- Contenido: Válido y accesible
- Tamaño: Correcto

🎯 RECOMENDACIONES PARA EASYPANEL

1. Verificar configuración del Build Context:

```
yaml
```

```
Dockerfile path: Dockerfile
```

```
Build context: aurum-control-center/
```

2. O configurar paths absolutos:

```
yaml
Dockerfile path: aurum-control-center/Dockerfile
Build context: .
```

3. Limpiar caché de Easypanel:

- Forzar rebuild sin caché
- Re-sincronizar desde GitHub

4. Verificar que el repositorio en Easypanel apunte a:

- Repository: `ghosting/aurum-control-center`
- Branch: `master`
- Commit: `bf6835f` o posterior



CONCLUSIÓN

✓ Problema de symlinks: RESUELTO

- No existía un problema real de symlinks en los archivos críticos
- El único symlink (`tmp`) fue eliminado por precaución

⚠ Error de Easypanel: CONFIGURACIÓN

- El error reportado es muy probablemente un problema de configuración del path del Dockerfile en Easypanel
- Todos los archivos necesarios están correctamente en GitHub como archivos reales



RECURSOS

- Repositorio GitHub: <https://github.com/ghosting/aurum-control-center>
- Dockerfile: <https://github.com/ghosting/aurum-control-center/blob/master/aurum-control-center/Dockerfile>
- Guía de deployment: `aurum-control-center/EASYPANEL_DEPLOY.md`
- Análisis técnico: `ANALISIS_DEPLOY.md`

Última actualización: Commit `bf6835f`

Estado: ✓ Repositorio limpio y listo para deployment