🔧 Fix de Error de Build en EasyPanel

Fecha: 9 de Octubre, 2025
Error Original: `exit code: 1` en `build-with-standalone.sh`
Estado: ✅ RESUELTO

🔍 Problema Identificado

El build de Docker estaba fallando en EasyPanel con el siguiente error:

```
ERROR: failed to build: failed to solve: process "/bin/sh -c echo \"Force rebuild timestamp: $BUILD_TIMESTAMP\" && ./build-with-standalone.sh" did not complete successfully: exit code: 1
```

Causa Raíz

El script `build-with-standalone.sh` estaba intentando:
1. Modificar el `next.config.js` dinámicamente durante el build
2. Ejecutar el build de Next.js
3. Verificar la estructura del output

Este enfoque era **innecesariamente complejo** y propenso a fallos durante el proceso de build de Docker.

✅ Solución Implementada

Cambio en el Dockerfile

ANTES (Complejo):

```
# Copy and prepare the standalone build script
COPY build-with-standalone.sh ./
RUN chmod +x build-with-standalone.sh

# Build the application with standalone output - FORCE REBUILD NO CACHE
ENV NEXT_TELEMETRY_DISABLED=1
ENV NEXT_OUTPUT_MODE=standalone
ENV BUILD_TIMESTAMP=20250930_073500_PRISMA_BIN_FIX
RUN echo "Force rebuild timestamp: $BUILD_TIMESTAMP" && ./build-with-standalone.sh
```

DESPUÉS (Simplificado):

```
# Build the application with standalone output - DIRECT BUILD
ENV NEXT_TELEMETRY_DISABLED=1
ENV NEXT_OUTPUT_MODE=standalone
ENV BUILD_TIMESTAMP=20251009_013500_SIMPLIFIED_BUILD

# Build Next.js directly without intermediate script
RUN echo "🏗️ Building Next.js with standalone output..." && \
    yarn build && \
    echo "✅ Build completed!" && \
    ls -la .next/ && \
    ls -la .next/standalone/ || echo "⚠️  Standalone directory not found"
```

Beneficios del Cambio

✅ **Simplicidad:** Build directo con `yarn build`, sin scripts intermedios
✅ **Confiabilidad:** Menos puntos de fallo
✅ **Debugging:** Más fácil identificar errores
✅ **Mantenibilidad:** Código más limpio y estándar

🚀 Próximos Pasos en EasyPanel

1. Verificar que EasyPanel Detecte los Cambios

EasyPanel debería detectar automáticamente el nuevo commit en GitHub:

```
Commit: 6f675d8
Mensaje: "🔧 Fix: Simplificar build de Docker - eliminar script intermedio"
```

2. Hacer Deploy Manual (Si es necesario)

Si EasyPanel no detecta automáticamente los cambios:

1. **Ir a tu proyecto en EasyPanel**
2. **Ir a la pestaña “Build”**
3. **Click en “Rebuild”** o “Deploy”
4. **Esperar a que termine el build**

3. Verificar el Build Exitoso

El build debería mostrar:

```
🏗️ Building Next.js with standalone output...
✅ Build completed!
[Lista de archivos en .next/]
[Lista de archivos en .next/standalone/]
```

4. Verificar el Contenedor

Una vez que el build termine exitosamente:

```
# Verificar que el contenedor esté corriendo
docker ps | grep laeconomica

# Ver los logs del contenedor
docker logs <container_id>
```

5. Probar la Aplicación

Abrir en el navegador:

```
https://app.mueblerialaeconomica.com
```

Deberías ver la aplicación funcionando correctamente.

🔍 Diagnóstico si Persiste el Error

Si el Build Sigue Fallando

1. **Ver los logs completos del build en EasyPanel**
2. **Buscar el error específico** en el output de `yarn build`
3. **Verificar las variables de entorno** en la configuración de EasyPanel

Comandos para Debug Local

Puedes probar el build localmente para depurar:

```
cd /home/ubuntu/muebleria_la_economica

# Build de prueba local
docker build -t test-build \
  --build-arg DATABASE_URL=postgresql://postgres:password@localhost:5432/test \
  --build-arg NEXTAUTH_URL=http://localhost:3000 \
  --build-arg NEXTAUTH_SECRET=test-secret \
  --build-arg JWT_SECRET=test-jwt-secret \
  --build-arg NODE_ENV=production \
  --build-arg PORT=3000 \
  .

# Ver los logs del build
docker build --progress=plain -t test-build .
```

📋 Variables de Entorno Requeridas

Asegúrate de que estas variables estén configuradas en EasyPanel:

```
# Base de datos
DATABASE_URL=postgresql://postgres:PASSWORD@laeconomica-db:5432/laeconomica-db?
schema=public

# Autenticación
NEXTAUTH_URL=https://app.mueblerialaeconomica.com
NEXTAUTH_SECRET=<tu-secret-seguro>
JWT_SECRET=<tu-jwt-secret-seguro>

# Node
NODE_ENV=production
PORT=3000
```

🎯 Resumen del Fix

Aspecto	Antes	Después
Método de Build	Script intermedio	Directo con yarn
Complejidad	Alta (modificación dinámica)	Baja (configuración estática)
Puntos de Fallo	Múltiples	Mínimos
Debugging	Difícil	Fácil
Timestamp	20250930_073500	20251009_013500

✅ Checklist Post-Deploy

- [ ] Build de Docker completa exitosamente
- [ ] Contenedor arranca sin errores
- [ ] Prisma client se conecta a la base de datos
- [ ] La aplicación responde en el puerto 3000
- [ ] Traefik/Proxy enruta correctamente al contenedor
- [ ] La aplicación es accesible en https://app.mueblerialaeconomica.com
- [ ] Las migraciones de base de datos se ejecutan correctamente
- [ ] Los usuarios pueden hacer login

🆘 Soporte Adicional

Si después de este fix el build sigue fallando:

1. **Copia el log completo del build** desde EasyPanel
2. **Busca la línea específica del error** (después de `yarn build`)
3. **Comparte el error** para un diagnóstico más específico

El error más común sería relacionado con:
- TypeScript errors durante el build
- Dependencias faltantes
- Variables de entorno incorrectas

Commit del Fix: `6f675d8`
Branch: `main`
Repository: https://github.com/qhosting/muebleria-la-economica.git