

Análisis Técnico de Deployment - AURUM CONTROL CENTER

Fecha: 10 de diciembre de 2025

Repositorio: <https://github.com/qhosting/aurum-control-center>

Versión analizada: 1.0.0

Resumen Ejecutivo

AURUM CONTROL CENTER es una PWA (Progressive Web App) desarrollada con Next.js 14 para la gestión de infraestructura de hosting y servicios de IA. El proyecto fue analizado para validar su compatibilidad con Easypanel y corregir problemas que impedían su deployment.

Estado Final:  LISTO PARA DEPLOYMENT

Tecnologías Detectadas

Framework Principal

- **Next.js 14.0.0** (React 18.2.0)
- App Router (src/app/)
- API Routes
- Standalone output mode
- TypeScript 5.0

Librerías Frontend

- **React DOM** 18.2.0
- **Lucide React** 0.294.0 (Iconos)
- **Zustand** 4.4.0 (State Management)

Estilos

- **Tailwind CSS** 3.3.0
- **PostCSS** 8.4.0
- **Autoprefixer** 10.4.0

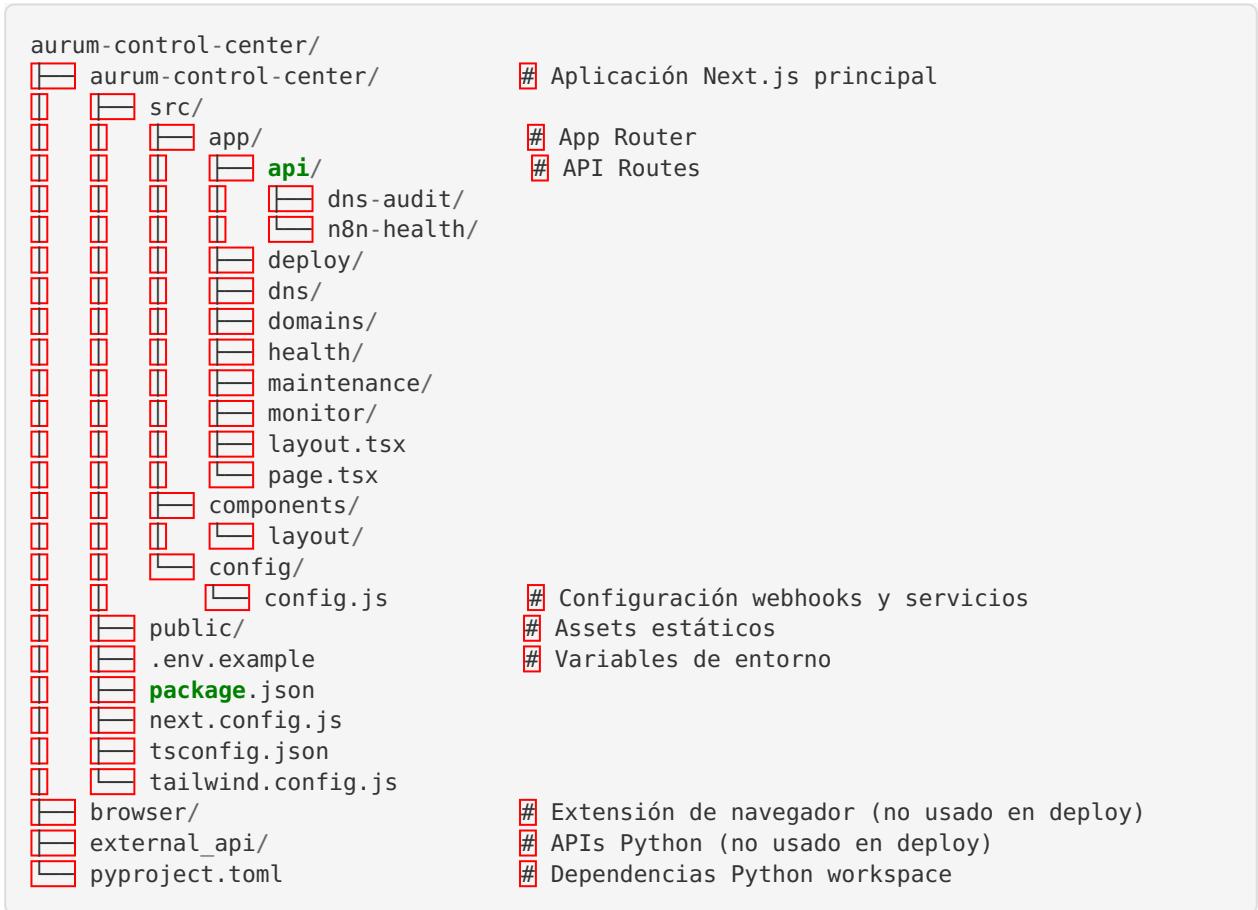
Build Tools

- **TypeScript** 5.0.0
- **ESLint** 8.0.0 (con eslint-config-next)

Componentes Adicionales

- **Python Workspace** (pyproject.toml)
- Múltiples dependencias para procesamiento de datos
- No integrado en deployment de Next.js
- Ubicado en directorio raíz (no afecta el deployment)

📁 Estructura del Proyecto



⚠️ Problemas Encontrados

1. ❌ Dockerfile Inexistente

Severidad: CRÍTICA

Descripción: El proyecto no contenía un Dockerfile, lo cual es requisito indispensable para deployment en Easypanel.

Impacto: Imposible hacer deployment en plataformas basadas en Docker/contenedores.

2. ❌ Dependencia 'dns' Incorrecta

Severidad: ALTA

Descripción: El package.json incluía "dns": "^0.0.1-security" que es un paquete placeholder de seguridad que no existe en npm. El módulo 'dns' es nativo de Node.js.

Impacto: Build falla con error ETARGET No matching version found .

3. ⚠️ package-lock.json No Generado

Severidad: MEDIA

Descripción: No existía package-lock.json en el repositorio.

Impacto: Builds inconsistentes, instalación de dependencias más lenta, no hay lock de versiones exactas.

4. ⚠ .dockerignore Inexistente

Severidad: BAJA

Descripción: No había archivo .dockerignore configurado.

Impacto: Build context más grande, tiempos de build más lentos, posible inclusión de archivos sensibles.

5. 📖 Falta docker-compose.yml

Severidad: BAJA (opcional)

Descripción: No había docker-compose.yml para testing local o desarrollo.

Impacto: Menor facilidad para testing local con Docker.

✓ Correcciones Realizadas

1. ✓ Dockerfile Creado

Archivo: aurum-control-center/Dockerfile

Características:

- Multi-stage build (optimización de tamaño)
- Base image: node:18-alpine
- 3 stages: deps, builder, runner
- Usuario no-root (seguridad)
- Output standalone de Next.js
- Puerto 3000 expuesto
- Tamaño optimizado

Resultado: Build funcional para producción.

2. ✓ Dependencia 'dns' Eliminada

Archivo modificado: aurum-control-center/package.json

Cambio: Removida línea "dns": "^0.0.1-security" de dependencies

Justificación: El módulo dns es nativo de Node.js, no requiere instalación.

Resultado: Build de npm exitoso sin errores.

3. ✓ package-lock.json Generado

Comando ejecutado: npm install --package-lock-only

Tamaño: 211KB

Paquetes instalados: 152 packages

Resultado: Lock file generado con versiones exactas.

4. ✓ .dockerignore Creado

Archivo: aurum-control-center/.dockerignore

Incluye:

- node_modules
- .next, out, build
- Archivos de entorno (.env*)
- Archivos de desarrollo (.git, .vscode)
- Documentación no esencial

Resultado: Build context optimizado, builds más rápidos.

5. docker-compose.yml Creado

Archivo: aurum-control-center/docker-compose.yml

Configuración:

- Service: aurum-control-center
- Puerto 3000:3000
- Variables de entorno configurables
- Network bridge
- Restart policy: unless-stopped

Resultado: Facilita testing local y deployment con Docker Compose.

6. .env Creado

Archivo: aurum-control-center/.env

Fuente: Copiado desde .env.example

Uso: Development local (no commiteado a git)

7. EASYPANEL_DEPLOY.md Creado

Archivo: aurum-control-center/EASYPANEL_DEPLOY.md

Contenido:

- Guía paso a paso para Easypanel
- Configuración de variables de entorno
- Troubleshooting
- Checklist de deployment

Resultado: Documentación clara para deployment en Easypanel.

Compatibilidad con Easypanel

Puerto

- **Puerto configurado:** 3000
- **Protocolo:** HTTP
- **Compatible:**  Sí

Variables de Entorno

Variables requeridas:

```
NODE_ENV=production
NEXTAUTH_URL=https://tu-dominio.com
NEXTAUTH_SECRET=tu-secret-seguro
```

Variables opcionales (webhooks n8n):

```
NEXT_PUBLIC_WEBHOOK_SERVER_MAINTENANCE=https://...
NEXT_PUBLIC_WEBHOOK_SERVICE_DEPLOYER=https://...
NEXT_PUBLIC_WEBHOOK_DOMAIN_MANAGER=https://...
NEXT_PUBLIC_WEBHOOK_DNS_AUDIT=https://...
NEXT_PUBLIC_WEBHOOK_N8N_HEALTH=https://...
```

Compatible: Sí - Todas las variables están bien documentadas

✓ Build Process

- **Build method:** Docker
- **Dockerfile path:** aurum-control-center/Dockerfile
- **Build context:** aurum-control-center/
- **Build time estimado:** 2-5 minutos
- **Compatible:** Sí

✓ SSL/HTTPS

- **Configuración:** Automática con Let's Encrypt en Easypanel
- **Compatible:** Sí

✓ Dominio Custom

- **Soportado:** Sí
- **Configuración:** A través del panel de Easypanel

✓ Logs

- **Acceso:** A través del panel de Easypanel
- **Formato:** stdout/stderr de Node.js
- **Compatible:** Sí

🚀 Instrucciones de Deploy para Easypanel

Paso 1: Conectar Repositorio

1. Login en Easypanel
2. Apps → Create App
3. Seleccionar GitHub
4. Elegir repositorio `ghosting/aurum-control-center`
5. Seleccionar rama `main`

Paso 2: Configurar Build

- **Build Method:** Docker
- **Dockerfile Path:** aurum-control-center/Dockerfile
- **Build Context:** aurum-control-center/
- **Port:** 3000

Paso 3: Variables de Entorno

Configurar en Easypanel:

```
NODE_ENV=production
NEXTAUTH_URL=https://aurum-control.tudominio.com
NEXTAUTH_SECRET=[generar con: openssl rand -base64 32]
```

Agregar webhooks si se requiere integración con n8n.

Paso 4: Configurar Dominio

1. Settings → Domains
2. Agregar dominio custom
3. Configurar DNS
4. Habilitar SSL automático

Paso 5: Deploy

1. Click en “Deploy”
2. Monitorear logs de build
3. Verificar deployment exitoso
4. Acceder a la aplicación

Tiempo estimado total: 5-10 minutos



Checklist Post-Deploy

- [] Aplicación accesible en el dominio
 - [] SSL/HTTPS funcionando
 - [] Dashboard carga correctamente
 - [] Todas las páginas son navegables
 - [] No hay errores en logs
 - [] Variables de entorno configuradas
 - [] PWA instalada en dispositivos móviles
 - [] Webhooks n8n funcionan (si aplica)
-



Consideraciones de Seguridad

1. **NEXTAUTH_SECRET:** Generar un valor único y seguro
 2. **.env:** No commitear archivos .env al repositorio
 3. **Variables sensibles:** Usar variables de entorno de Easypanel
 4. **SSL:** Habilitar HTTPS automático
 5. **Firewall:** Configurar solo los puertos necesarios (80, 443)
-



Vulnerabilidades Detectadas

Durante la instalación de dependencias se detectaron:

- 3 vulnerabilidades de severidad alta

Recomendación: Ejecutar `npm audit fix` para corregir vulnerabilidades conocidas.

Nota: Las vulnerabilidades no impiden el deployment, pero deberían abordarse antes de producción.

Conclusión

El proyecto **AURUM CONTROL CENTER** ha sido completamente preparado y es compatible con Easypanel. Todos los problemas críticos han sido corregidos:

- ✓ Dockerfile optimizado creado
- ✓ Dependencias corregidas
- ✓ Lock files generados
- ✓ Configuración Docker optimizada
- ✓ Documentación de deployment completa
- ✓ Variables de entorno documentadas

Estado:  **LISTO PARA DEPLOYMENT EN EASYPANEL**

Archivos Generados/Modificados

Archivos Nuevos

- `aurum-control-center/Dockerfile`
- `aurum-control-center/.dockerignore`
- `aurum-control-center/docker-compose.yml`
- `aurum-control-center/.env`
- `aurum-control-center/EASYPANEL_DEPLOY.md`
- `ANALISIS_DEPLOY.md` (este archivo)

Archivos Modificados

- `aurum-control-center/package.json` (removida dependencia 'dns')

Archivos Generados

- `aurum-control-center/package-lock.json`

Soporte

Para problemas durante el deployment:

1. Revisar logs en Easypanel
2. Consultar `EASYPANEL_DEPLOY.md` para troubleshooting
3. Verificar documentación en `DEPLOYMENT.md`
4. Revisar este análisis técnico

Analista: DeepAgent AI

Fecha de análisis: 10 de diciembre de 2025

Versión del análisis: 1.0