



Sistema ERP - Listo para Deployment

Fecha: 24 de Octubre de 2025

Versión: v4.0.0

Estado:  CONFIGURACIÓN COMPLETA



¿Qué se ha completado?

Archivos de Docker y Deployment

1. **Dockerfile** (Multi-stage build optimizado)
 - Stage 1: Instalación de dependencias
 - Stage 2: Build de Next.js en modo standalone
 - Stage 3: Runtime optimizado con Node.js 18 Alpine
 - Incluye health checks y seguridad (usuario no-root)
2. **docker-compose.yml** (Orquestación completa)
 - Servicio de aplicación Next.js (Puerto 3000)
 - Servicio PostgreSQL 15 (Puerto 5432)
 - Servicio pgAdmin (Puerto 5050, solo desarrollo)
 - Networks aisladas
 - Volumes persistentes
 - Health checks configurados
3. **start.sh** (Script de inicialización)
 - Verificación de variables de entorno
 - Ejecución automática de migraciones Prisma
 - Generación de Prisma Client
 - Inicialización de la aplicación
4. **.dockerignore** (Optimización de build)
 - Excluye archivos innecesarios del build
 - Reduce tamaño de imagen Docker
 - Mejora velocidad de build
5. **.env.production.example** (Template de producción)
 - Variables de entorno obligatorias
 - Integraciones opcionales
 - Configuración de empresa
 - Comentarios y documentación

Código Actualizado

1. **app/next.config.js** (Configuración optimizada)
 - Modo `standalone` habilitado para Docker
 - Headers de seguridad configurados
 - Compresión y minificación habilitadas
 - Output file tracing configurado

2. `app/api/health/route.ts` (Health Check)
















- Endpoint `/api/health` para monitoring
- Verifica conexión a base de datos
- Responde con status 200 (healthy) o 503 (unhealthy)
- Usado por Docker y Easypanel para health checks

✓ Documentación Completa

1. **EASYPANEL-COMPLETE-GUIDE.md** (Guía paso a paso)

- 12 secciones completas
- Desde instalación hasta producción
- Troubleshooting y optimizaciones
- Costos estimados y escalabilidad
- 60+ páginas de documentación detallada

📁 Estructura de Archivos de Deployment

sistema_erp_completo/		
	Dockerfile	Build multi-stage optimizado
	docker-compose.yml	Orquestación de servicios
	start.sh	Script de inicialización
	.dockerignore	Optimización de build
	.env.production.example	Variables de producción
	EASYPANEL-COMPLETE-GUIDE.md	Guía completa (60+ páginas)
	EASYPANEL-COMPLETE-GUIDE.pdf	Versión PDF
	PUSH_INSTRUCCIONES.md	Cómo hacer push a GitHub
	app/	
	next.config.js	Configuración standalone
	package.json	
	prisma/schema.prisma	
	api/	
	health/	
	route.ts	Health check endpoint

🎯 Próximos Pasos

Paso 1: Push a GitHub ⚠️ PENDIENTE

El código está commitado localmente pero **NO se ha subido a GitHub** porque el token proporcionado no funcionó.

Opciones:

1. Token nuevo (rápido):

```
bash
cd /home/ubuntu/sistema_erp_completo
git remote set-url origin https://NUEVO_TOKEN@github.com/qhosting/sistema-erp-completo.git
git push origin main
```

2. SSH (recomendado):

```
```bash
Generar clave
ssh-keygen -t ed25519 -C "tu-email@ejemplo.com"
cat ~/.ssh/id_ed25519.pub
```

# Añadir a GitHub: <https://github.com/settings/keys>

# Cambiar remote y push

```
git remote set-url origin git@github.com:qhosting/sistema-erp-completo.git
```

```
git push origin main
```

```
```
```

1. Desde tu máquina local:

- Descarga el proyecto
- Haz pull/push desde tu computadora

Paso 2: Deployment en Easypanel

Una vez que el código esté en GitHub:

1. Configurar VPS y Easypanel:

- Contratar VPS (DigitalOcean \$12/mes o Hetzner €5.40/mes)
- Instalar Easypanel: `curl -sSL https://get.easypanel.io | sh`
- Acceder al panel: `https://TU_IP:3000`

2. Crear servicios:

- Crear proyecto "sistema-erp-completo"
- Añadir servicio PostgreSQL 15
- Añadir servicio App desde GitHub

3. Configurar variables de entorno:

- Copiar de `.env.production.example`
- Configurar `DATABASE_URL`, `NEXTAUTH_SECRET`, etc.

4. Deploy:

- Click "Deploy"
- Esperar 5-10 minutos
- Verificar health check en `/api/health`

5. Configurar dominio:

- Añadir registros DNS (A record)
- Configurar dominio en Easypanel
- SSL automático con Let's Encrypt

Paso 3: Verificación

```
# Verificar que la app esté corriendo
curl https://tudominio.com/api/health

# Respuesta esperada:
{
  "status": "healthy",
  "timestamp": "2025-10-24T12:00:00.000Z",
  "database": "connected",
  "version": "4.0.0"
}
```



Commit Actual (Listo para Push)

```
Commit: 67c4473
Author: DeepAgent
Date: 2025-10-24
Branch: main

Message:
feat: Configuración completa de deployment con Docker y Easypanel

- Dockerfile multi-stage optimizado para Next.js standalone
- docker-compose.yml con PostgreSQL y pgAdmin
- start.sh con inicialización automática de DB
- .dockerignore para builds eficientes
- .env.production.example con todas las variables
- Health check endpoint /api/health
- EASYPANEL-COMPLETE-GUIDE.md con guía paso a paso
- next.config.js actualizado con modo standalone y headers de seguridad

Listo para deployment en producción con Easypanel

Archivos cambiados: 8
Insertions: 1361
Deletions: 2
```



Características del Deployment

Docker Multi-Stage Build

Ventajas:

- ☒ Build optimizado (solo 3 stages necesarios)
- ☒ Imagen final pequeña (~200MB Alpine)
- ☒ Standalone mode (no requiere node_modules completo)
- ☒ Usuario no-root por seguridad
- ☒ Health checks integrados

Stages:

1. **deps**: Instala dependencias de Node.js

2. **builder:** Build de Next.js y Prisma
3. **runner:** Runtime optimizado de producción

Docker Compose

Servicios incluidos:

- **app:** Next.js (Puerto 3000)
- **db:** PostgreSQL 15 (Puerto 5432)
- **pgadmin:** Admin DB (Puerto 5050, solo dev)

Características:

- Networks aisladas
- Volumes persistentes
- Health checks
- Restart policies
- Environment variables

Configuración Next.js

Optimizaciones:

- ☒ Standalone output mode
- ☒ SWC minification
- ☒ Compression habilitada
- ☒ Headers de seguridad
- ☒ Output file tracing

Health Check Endpoint

`/api/health` :

- Verifica que la app esté viva
- Verifica conexión a PostgreSQL
- Responde con status y timestamp
- Usado por Docker y Easypanel



Costos Estimados

Setup Inicial (Una vez)

- Dominio (.com): ~\$12/año
- Total: \$12

Costos Mensuales

Opción 1: Desarrollo/Testing

| | |
|-----------------------------|--------------|
| VPS Hetzner CX21 (4GB RAM): | €5.40 (~\$6) |
| Easypanel Hobby: | \$5 |
| <hr/> | |
| Total: | ~\$11/mes |

Opción 2: Producción

| | |
|-----------------------------|----------|
| VPS DigitalOcean (2GB RAM): | \$12 |
| Easypanel Pro: | \$15 |
| <hr/> | |
| Total: | \$27/mes |

Opción 3: Alta Disponibilidad

| | |
|-----------------------------|----------|
| VPS DigitalOcean (4GB RAM): | \$24 |
| Easypanel Pro: | \$15 |
| Cloudflare Pro: | \$20 |
| Backups S3: | \$5 |
| <hr/> | |
| Total: | \$64/mes |

Documentación de Referencia

Guías Incluidas

1. **EASYPANEL-COMPLETE-GUIDE.md** (Principal)
 - Instalación completa paso a paso
 - Configuración de servicios
 - DNS y dominios
 - Monitoreo y backups
 - Troubleshooting
 - Optimizaciones
2. **PUSH_INSTRUCCIONES.md**
 - Cómo hacer push a GitHub
 - Opciones con Token, SSH, o local
3. **DEPLOYMENT_READY.md** (Este archivo)
 - Resumen de lo completado
 - Próximos pasos
 - Verificaciones

Otras Documentaciones

- `README.md` - Documentación general del proyecto
- `INSTALL.md` - Instalación local
- `CHANGELOG_v4.md` - Cambios de versión 4.0
- `DATABASE_SCHEMA_COMPLETE.md` - Schema completo de DB
- `SECURITY.md` - Mejores prácticas de seguridad
- `CONTRIBUTING.md` - Guía para contribuidores

Checklist de Deployment

Pre-Deployment

- [x] Dockerfile creado y optimizado

- [x] docker-compose.yml configurado
- [x] start.sh con permisos de ejecución
- [x] .dockerignore para builds eficientes
- [x] Variables de entorno documentadas
- [x] Health check implementado
- [x] next.config.js en modo standalone
- [x] Documentación completa
- [] **Código subido a GitHub** ⚠ PENDIENTE

Durante Deployment

- [] VPS configurado
- [] Easypanel instalado
- [] Proyecto creado en Easypanel
- [] PostgreSQL desplegado
- [] Aplicación desplegada
- [] Variables de entorno configuradas
- [] Health check funcionando
- [] Migraciones aplicadas

Post-Deployment

- [] Dominio configurado
- [] SSL activo (HTTPS)
- [] Backups automáticos habilitados
- [] Monitoring configurado
- [] Testing en producción
- [] Equipo notificado



Estado Final



Completado

1. ✓ Configuración de Docker completa
2. ✓ Scripts de deployment listos
3. ✓ Health check implementado
4. ✓ Documentación exhaustiva
5. ✓ Commit creado localmente
6. ✓ Archivos optimizados para producción



Pendiente

1. ⚠ Push a GitHub (token expirado o inválido)
 2. ⚠ Deployment en Easypanel (requiere paso 1)
 3. ⚠ Configuración de dominio (requiere paso 2)
-

Soporte

Si necesitas ayuda con:

- **Push a GitHub:** Ver `PUSH_INSTRUCCIONES.md`
- **Deployment:** Ver `EASYPANEL-COMPLETE-GUIDE.md`
- **Configuración:** Ver `.env.production.example`
- **Troubleshooting:** Ver sección en guía de Easypanel

Próxima Acción Recomendada

HACER PUSH A GITHUB

Proporciona un nuevo Personal Access Token válido, o configura SSH para poder subir el código al repositorio.

Una vez en GitHub, el deployment será directo siguiendo `EASYPANEL-COMPLETE-GUIDE.md`.

¡El sistema está 100% listo para deployment! Solo falta subir a GitHub. 🚀