

# Checkpoint Guardado Exitosamente

---

**Fecha:** 24 de Octubre, 2025

**Checkpoint:** "Configuración deployment Docker y Easypanel"

**Estado:**  Build exitoso, tests pasados, proyecto listo

---






## Resumen de Cambios Completados

---






### 1. Configuración de Deployment (Docker + Easypanel)

#### Archivos Creados/Actualizados:





**Dockerfile** - Multi-stage build optimizado

-  Stage 1: Instalación de dependencias
-  Stage 2: Build de producción
-  Stage 3: Runtime optimizado
-  Uso de standalone output de Next.js
-  Usuario no-root para seguridad




**docker-compose.yml** - Orquestación de servicios

-  Servicio web (Next.js)
-  Servicio PostgreSQL
-  Volúmenes persistentes
-  Health checks configurados
-  Red interna para comunicación




**start.sh** - Script de inicialización

-  Espera de disponibilidad de base de datos
-  Migraciones automáticas de Prisma
-  Seed de datos inicial
-  Inicio del servidor





**.dockerignore** - Optimización de build

-  Exclusión de node\_modules
-  Exclusión de archivos de desarrollo
-  Reducción de tamaño de imagen

**.env.production.example** - Template de variables

-  Documentación de variables requeridas
-  Ejemplos de configuración
-  Organización por categorías

**EASYPANEL-COMPLETE-GUIDE.md** - Guía completa

-  Instrucciones paso a paso
-  Configuración de servicios
-  Variables de entorno
-  Troubleshooting

## 2. Configuración de Health Check

### app/api/health/route.ts

- ☒ Endpoint de monitoreo
- ☒ Verificación de base de datos
- ☒ Información del sistema
- ☒ Status codes apropiados

## 3. Fixes de Compilación

### app/app/layout.tsx

- ☒ Service Worker temporalmente deshabilitado
- ☒ Eliminación de errores de consola
- ☒ Build limpio sin warnings críticos



## Estado del Repositorio Git

### Commits Pendientes para Push:

```
1e5bef7 - fix: Deshabilitar Service Worker para evitar errores de redirect
bcc5c7a - docs: Add DEPLOYMENT_READY PDF documentation
2a307af - Configuración de deployment
67c4473 - feat: Configuración completa de deployment con Docker y Easypanel
a2bc768 - Actualización de documentación
d748244 - Commit inicial
```

**Total:** 6 commits listos para push

**Rama:** main

**Remote:** [https://github.com/qhosting/sistema\\_erp\\_completo](https://github.com/qhosting/sistema_erp_completo)



## Acción Requerida: Push a GitHub

El token de GitHub anterior ha expirado. Necesitas generar un nuevo token para completar el push.

### Opción 1: Nuevo Token de GitHub (Recomendado)

#### 1. Generar token:

- Ve a: <https://github.com/settings/tokens>
- Click en "Generate new token" → "Generate new token (classic)"
- Configuración:
  - Name: ERP System - Deployment 2025
  - Expiration: 90 days
  - Scopes: ☒ repo + ☒ workflow
  - Click "Generate token"
  - **COPIA EL TOKEN INMEDIATAMENTE**

#### 2. Hacer push con el nuevo token:

```
cd /home/ubuntu/sistema_erp_completo

# Configurar remote con el token
git remote set-url origin https://TU_NUEVO_TOKEN@github.com/qhosting/sistema_erp_completo.git

# Push con tags
git push origin main --tags

# Limpiar token por seguridad
git remote set-url origin https://github.com/qhosting/sistema_erp_completo.git
```

## Opción 2: SSH (Más Seguro)

```
# Generar clave SSH
ssh-keygen -t ed25519 -C "tu_email@ejemplo.com"

# Agregar a GitHub: https://github.com/settings/keys
cat ~/.ssh/id_ed25519.pub

# Cambiar remote a SSH
git remote set-url origin git@github.com:qhosting/sistema_erp_completo.git

# Push
git push origin main --tags
```

## Opción 3: GitHub CLI

```
# Instalar gh
curl -fsSL https://cli.github.com/packages/githubcli-archive-keyring.gpg | sudo dd of=/usr/share/keyrings/githubcli-archive-keyring.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/githubcli-archive-keyring.gpg] https://cli.github.com/packages stable main" | sudo tee /etc/apt/sources.list.d/github-cli.list > /dev/null
sudo apt update && sudo apt install gh

# Autenticar y push
gh auth login
git push origin main --tags
```



## Siguiente Paso: Deployment en Easypanel

Una vez que hagas push al repositorio, sigue estos pasos:

### 1. Acceder a Easypanel

- URL: Tu instancia de Easypanel
- Login con tus credenciales

### 2. Crear Nuevo Proyecto

- Click en "Create Project"
- Nombre: `sistema-erp-completo`
- Seleccionar fuente: GitHub Repository

### 3. Conectar Repositorio

- Autorizar GitHub si es necesario
- Seleccionar: `ghosting/sistema_erp_completo`
- Branch: `main`

### 4. Configurar Servicios

#### Servicio 1: Database (PostgreSQL)

- Image: `postgres:15-alpine`
- Variables de entorno:
  - `POSTGRES_USER=erp_user`
  - `POSTGRES_PASSWORD=tu_password_seguro`
  - `POSTGRES_DB=erp_database`
- Volume: `/var/lib/postgresql/data`
- Port: 5432 (interno)

#### Servicio 2: Web (Next.js)

- Build method: Dockerfile
- Port: 3000
- Variables de entorno (ver `.env.production.example`):
  - `DATABASE_URL=postgresql://erp_user:password@postgres:5432/erp_database`
  - `NEXTAUTH_URL=https://tu-dominio.com`
  - `NEXTAUTH_SECRET=genera_un_secret_aleatorio`

### 5. Deploy

- Click en "Deploy"
- Esperar a que el build complete
- Verificar logs para confirmar éxito

### 6. Verificación Post-Deploy

- ☒ Acceder a la URL pública
- ☒ Verificar health check: `/api/health`
- ☒ Probar login con usuario de prueba
- ☒ Verificar conectividad a base de datos



## Checklist Completo



### Desarrollo

- [x] Todos los módulos implementados (FASE 1-4)
- [x] Autenticación configurada (NextAuth.js)
- [x] Base de datos configurada (PostgreSQL + Prisma)
- [x] APIs implementadas y funcionando
- [x] UI/UX completa con Shadcn/UI
- [x] Responsive design



### Documentación

- [x] README completo

- [x] Guías de instalación
- [x] Documentación de API
- [x] Esquema de base de datos
- [x] Guía de deployment
- [x] Guía de Easypanel
- [x] Changelog
- [x] PDFs generados

## ✓ Deployment

- [x] Dockerfile multi-stage
- [x] docker-compose.yml
- [x] Scripts de inicialización
- [x] Health checks
- [x] Variables de entorno documentadas

## ✓ Testing

- [x] Compilación TypeScript exitosa
- [x] Build de producción exitoso
- [x] Dev server funcionando
- [x] Checkpoint guardado

## ↻ Pendiente

- [ ] Push a GitHub (requiere nuevo token)
- [ ] Deployment en Easypanel
- [ ] Configuración de dominio personalizado
- [ ] SSL/TLS configurado
- [ ] Pruebas en producción

## 📦 Archivos Listos en el Repositorio

```
sistema_erp_completo/
├── app/                                # Aplicación Next.js
│   ├── app/                          # Pages y API routes
│   ├── components/                  # Componentes React
│   ├── lib/                        # Utilidades y configs
│   ├── prisma/                     # Esquema de base de datos
│   └── public/                      # Assets estáticos
├── docs/                            # Documentación detallada
├── Dockerfile                       # ✨ NUEVO
├── docker-compose.yml               # ✨ NUEVO
├── start.sh                         # ✨ NUEVO
├── .dockerignore                    # ✨ NUEVO
├── .env.production.example          # ✨ NUEVO
├── EASYPANEL-COMPLETE-GUIDE.md     # ✨ NUEVO
├── DEPLOYMENT_READY.md              # ✨ NUEVO
├── README.md                        # Actualizado
└── [15+ archivos de docs...]       # Completos
```

## Sistema ERP - Características Principales

---

### Módulos Implementados:

1. **Dashboard** - Visualización ejecutiva de KPIs
2. **Clientes** - Gestión completa de clientes
3. **Ventas** - Sistema de ventas y cotizaciones
4. **Pedidos** - Gestión de pedidos
5. **Cobranza** - Sistema de cobranza tradicional
6. **Cobranza Móvil** - PWA para cobradores en campo
7. **Pagares** - Gestión de pagarés y financiamiento
8. **Productos** - Catálogo e inventario
9. **Almacén** - Control de inventario
10. **Compras** - Módulo de compras y proveedores
11. **Notas de Crédito/Cargo** - Ajustes contables
12. **Garantías** - Sistema de garantías
13. **Reestructuras** - Reestructuración de deudas
14. **Facturación Electrónica** - Integración con PAC
15. **Comunicación** - SMS y WhatsApp
16. **Automatización** - Tareas y workflows
17. **Business Intelligence** - Analytics avanzados
18. **Auditoría** - Registro de cambios
19. **Reportes** - Sistema de reportes

### Tecnologías:

- **Frontend:** Next.js 14, React 18, TypeScript
- **UI:** Shadcn/UI, Tailwind CSS, Radix UI
- **Backend:** Next.js API Routes, NextAuth.js
- **Base de datos:** PostgreSQL, Prisma ORM
- **Deployment:** Docker, Easypanel, PostgreSQL
- **Integraciones:** Evolution API (WhatsApp), LabsMobile (SMS)



### Notas Importantes

---

1. **Service Worker:** Temporalmente deshabilitado para evitar errores de redirect. Puedes rehabilitarlo más tarde cuando configures una página de inicio que no redirija.
2. **Variables de Entorno:** Asegúrate de configurar todas las variables requeridas en Easypanel antes del deployment.
3. **Seguridad:**
  - Genera un NEXTAUTH\_SECRET fuerte
  - Usa contraseñas seguras para PostgreSQL
  - Configura CORS apropiadamente
  - Implementa rate limiting en producción

#### 4. **Base de Datos:**

- Las migraciones se ejecutan automáticamente en start.sh
- El seed crea usuarios de prueba
- Respalda la base de datos regularmente

#### 5. **Monitoreo:**

- Usa el endpoint /api/health para health checks
- Configura logs en Easypanel
- Monitorea el uso de recursos

---

## Soporte

Para cualquier problema durante el deployment:

1. Revisa la documentación en `EASYPANEL-COMPLETE-GUIDE.md`
2. Verifica los logs del contenedor
3. Consulta el endpoint de health check
4. Revisa la configuración de variables de entorno

---

**¡El proyecto está completamente listo para deployment!** 🎉

Solo falta hacer el push a GitHub con un nuevo token de autenticación.