



Resumen de Completación del Deployment - CitaPlanner

Fecha: 7 de Octubre, 2025

Repositorio: qhosting/citaplanner

Branch Principal: main



Tareas Completadas

1. Merge del PR #59 (Logging Verbose)

- **Estado:** ❌ No se pudo mergear directamente (conflictos de rebase)
- **Solución:** Se creó PR #60 con los mismos cambios
- **PR #60:** ✅ Mergeado exitosamente
- **Commit SHA:** 2f7fc253edcf72d6bcb4574ed5cf1a92abcf3f6

Cambios incluidos en PR #60:

- Agregado `set -e` y `set -x` al build script para mejor debugging
- Logging comprehensivo en cada paso del build
- Verificación de archivos esenciales antes de proceder
- Verificación de versiones de Node y Yarn
- Mostrar contenido de `next.config.js` durante el build
- Manejo detallado de errores para fallos en yarn build
- Actualizado `BUILD_TIMESTAMP` para forzar invalidación de cache

2. Corrección de Configuración Standalone (PR #61)



- **Estado:** ✅ Creado y mergeado exitosamente
- **PR #61:** <https://github.com/qhosting/citaplanner/pull/61>
- **Commit SHA:** 8265bcb2d55b0ae7ccd35f0f914db77d746faa80

Cambios incluidos en PR #61:

- ✅ Agregado fallback a `'standalone'` en `next.config.js`
- Antes: `output: process.env.NEXT_OUTPUT_MODE`
- Ahora: `output: process.env.NEXT_OUTPUT_MODE || 'standalone'`
- ✅ Simplificado `build-with-standalone.sh` (eliminada manipulación redundante)
- ✅ Actualizado `BUILD_TIMESTAMP` a `20251007_FINAL_FIX_STANDALONE`
- ✅ Reducida complejidad del build process

3. Verificación de Archivos Críticos

- ✅ Todos los archivos necesarios están presentes:
- ✅ `app/prisma/schema.prisma` - Schema de base de datos
- ✅ `app/scripts/seed.ts` - Script de seed
- ✅ `app/package.json` - Configuración correcta de prisma seed
- ✅ `Dockerfile` - Configuración multi-stage optimizada
- ✅ `start.sh` - Script de inicio con manejo de Prisma

-  `build-with-standalone.sh` - Script de build simplificado
-  `next.config.js` - Configuración con fallback standalone





Estado Actual del Repositorio

Commits Recientes en Main

```
8265bcb - fix: ensure standalone output in next.config.js and simplify build script
2f7fc25 - Merge pull request #60 (Add verbose logging)
6349e02 - Adopt working Easypanel configuration from muebleria-la-economica (#57)
ea579ee - fix: use npx tsx for prisma seed command (#56)
```

PRs Mergeados

-  PR #60: Add verbose logging and error handling to build script
-  PR #61: Fix: Ensure standalone output and simplify build process

PRs Pendientes

-  PR #59: Cerrado/Obsoleto (reemplazado por PR #60)



Configuración del Deployment

Dockerfile

- **Build Method:** Multi-stage con standalone output
- **Base Image:** node:18-alpine
- **Build Timestamp:** 20251007_FINAL_FIX_STANDALONE
- **Output Mode:** standalone (con fallback garantizado)

Next.js Configuration

```
output: process.env.NEXT_OUTPUT_MODE || 'standalone'
```

- Garantiza que siempre se genere standalone output
- Fallback robusto si la variable de entorno falla

Prisma Configuration

```
"prisma": {
  "seed": "npx tsx --require dotenv/config scripts/seed.ts"
}
```

- Seed script configurado correctamente
- Usa tsx para ejecutar TypeScript directamente

Webhook de Deployment

Problema Identificado

El webhook proporcionado no responde:

```
URL: https://easypanel.abacus.ai/webhooks/deploy/clzwwqxqy0001uo8aqvvvvvvv
Error: Could not resolve host: easypanel.abacus.ai
```

Soluciones Alternativas

Opción 1: Auto-Deploy desde GitHub (Recomendado)

Si Easypanel está configurado con auto-deploy desde GitHub:

1. Los cambios ya están en `main`
2. Easypanel debería detectar automáticamente el push
3. El deployment se iniciará automáticamente

Opción 2: Deploy Manual desde Easypanel UI

1. Acceder a Easypanel dashboard
2. Navegar al proyecto CitaPlanner
3. Hacer clic en "Deploy" o "Redeploy"
4. Monitorear los logs del build

Opción 3: Verificar Webhook URL

El webhook URL podría estar incorrecto o el servicio podría estar en un dominio diferente:

- Verificar en Easypanel dashboard la URL correcta del webhook
- Podría ser un dominio diferente (no easypanel.abacus.ai)

Archivos Modificados en Esta Sesión

1. app/next.config.js

```
- output: process.env.NEXT_OUTPUT_MODE,
+ output: process.env.NEXT_OUTPUT_MODE || 'standalone',
```

2. build-with-standalone.sh



- Eliminadas ~45 líneas de código redundante
- Simplificado para confiar en next.config.js
- Mantiene logging verbose para debugging

3. Dockerfile

```
- ENV BUILD_TIMESTAMP=20251007_VERBOSE_LOGGING_DEBUG
+ ENV BUILD_TIMESTAMP=20251007_FINAL_FIX_STANDALONE
```

Próximos Pasos Recomendados

Inmediatos

1. **Verificar Auto-Deploy:**
 - Revisar si Easypanel detectó el push a main
 - Verificar logs de deployment en Easypanel dashboard
2. **Deploy Manual (si auto-deploy no funciona):**
 - Acceder a Easypanel
 - Hacer clic en "Deploy" manualmente
 - Monitorear logs del build
3. **Verificar Build Logs:**
 - Con el logging verbose, los logs mostrarán exactamente dónde falla (si falla)
 - Buscar mensajes con  o  en los logs







Después del Deployment Exitoso

1. **Verificar la aplicación:**
 - Acceder a la URL de la aplicación
 - Verificar que carga correctamente
 - Probar login y funcionalidades básicas
2. **Verificar Base de Datos:**
 - Confirmar que las migraciones se aplicaron
 - Verificar que el seed se ejecutó (si es primera vez)
3. **Monitorear Errores:**
 - Revisar logs de la aplicación en runtime
 - Verificar conexión a base de datos

Debugging Guide

Si el Build Falla

Con el logging verbose del PR #60, los logs mostrarán:

1.  Qué archivos existen y cuáles faltan
2.  Versiones de Node y Yarn
3.  Contenido de next.config.js
4.  Estado de node_modules
5.  Salida completa de yarn build
6.  Estructura del directorio .next/standalone

Comandos Útiles para Debugging

```
# Ver logs del container
docker logs [CONTAINER_ID]

# Acceder al container
docker exec -it [CONTAINER_ID] sh

# Verificar archivos
ls -la /app/
ls -la /app/.next/standalone/




# Verificar Prisma
npx prisma generate
npx prisma db push
```

Documentación Relacionada




- **EASYPANEL-DEPLOYMENT-GUIDE.md** - Guía completa de deployment
- **DEPLOYMENT.md** - Documentación general de deployment
- **PR #60** - <https://github.com/qhosting/citaplanner/pull/60>
- **PR #61** - <https://github.com/qhosting/citaplanner/pull/61>

Mejoras Implementadas




Robustez

-  Fallback garantizado para standalone output
-  Logging verbose para debugging rápido
-  Verificaciones de archivos esenciales

Simplicidad

-  Código más limpio y mantenible
-  Menos manipulación de archivos
-  Configuración centralizada

Debugging

-  Logs detallados en cada paso
-  Mensajes de error claros
-  Verificaciones de estado

Conclusión

Estado Final:  Repositorio listo para deployment

Todos los cambios necesarios han sido implementados y mergeados a `main`. El repositorio ahora tiene:

- ☒ Configuración robusta de standalone output
- ☒ Logging verbose para debugging
- ☒ Scripts simplificados y mantenibles
- ☒ Todos los archivos críticos presentes

Acción Requerida del Usuario:

1. Verificar si el auto-deploy de Easypanel se activó
 2. Si no, hacer deploy manual desde Easypanel dashboard
 3. Monitorear los logs del build
 4. Reportar cualquier error encontrado (los logs verbose ayudarán)
-

Nota Importante sobre GitHub App:

Para operaciones futuras con repositorios privados, asegúrate de dar permisos a la [GitHub App de Abacus.AI](https://github.com/apps/abacusai/installations/select_target) (https://github.com/apps/abacusai/installations/select_target) para acceder a todos los repositorios necesarios.