

Checklist de Deployment - CitaPlanner con Persistencia

Pre-Deployment

1. Configuración de Volúmenes en Easypanel

- ☐ **Volumen PostgreSQL** configurado
- Nombre: `citaplanner-postgres-data`
- Mount Path: `/var/lib/postgresql/data`
- Tamaño: Mínimo 10GB
- ☐ **Volumen de Backups** configurado
- Nombre: `citaplanner-backups`
- Mount Path: `/backup-citaplanner`
- Tamaño: Mínimo 20GB

2. Variables de Entorno

- ☐ `DATABASE_URL` configurada correctamente
- ☐ `NEXTAUTH_URL` configurada con tu dominio
- ☐ `NEXTAUTH_SECRET` generado (mínimo 32 caracteres aleatorios)
- ☐ `MASTER_PASSWORD_HASH` (opcional, usa default si no se configura)

3. Código Actualizado

- ☐ Pull de la rama `feature/db-persistence`
 - ☐ Seed script idempotente actualizado
 - ☐ Script de backup con permisos de ejecución
 - ☐ `docker-compose.yml` con volumen de backups
 - ☐ Documentación revisada
-

Durante el Deployment

1. Deploy en Easypanel

- ☐ Push del código a GitHub
- ☐ Trigger deploy en Easypanel
- ☐ Esperar a que el build complete

2. Verificación de Logs

- ☐ Logs de build sin errores
- ☐ Migraciones aplicadas correctamente
- ☐ Seed ejecutado (solo si BD vacía)
- ☐ Aplicación iniciada correctamente

Post-Deployment

1. Verificación de Persistencia

- ☐ Aplicación accesible en el dominio
- ☐ Login funciona correctamente
- ☐ Datos de seed visibles (si es primera vez)

2. Prueba de Persistencia

- ☐ Crear un cliente de prueba
- ☐ Redeploy la aplicación
- ☐ Verificar que el cliente sigue existiendo
- ☒ Persistencia confirmada

3. Configuración de Backups

- ☐ Script de backup accesible en `/app/scripts/pg_backup.sh`
- ☐ Directorio de backups montado en `/backup-citaplanner`
- ☐ Ejecutar backup manual: `./scripts/pg_backup.sh`
- ☐ Verificar que se creó el backup en `/backup-citaplanner/daily/`

4. Configurar Cron Job

Opción A: En Easypanel

- ☐ Agregar cron job en servicio PostgreSQL
- ☐ Schedule: `0 2 * * *`
- ☐ Command: `/app/scripts/pg_backup.sh`

Opción B: Manual en Contenedor

- ☐ Conectar al contenedor
- ☐ Instalar cron: `apk add dcron`
- ☐ Configurar crontab
- ☐ Iniciar cron: `crond -b`

5. Verificación de Backups

- ☐ Esperar 24 horas para primer backup automático
- ☐ Verificar que se creó el backup
- ☐ Verificar integridad: `./scripts/pg_backup.sh --verify-only`
- ☐ Verificar logs en `/backup-citaplanner/logs/`

Pruebas Funcionales

1. Módulos Principales

- ☐ **Autenticación**
- Login con `admin@citaplanner.com`
- Login con otros usuarios
- Logout funciona

- ☐ **Clientes**

- Crear nuevo cliente
- Editar cliente existente
- Ver historial de cliente
- Agregar notas

- ☐ **Servicios**

- Ver catálogo de servicios
- Crear nuevo servicio
- Editar servicio

- ☐ **Citas**

- Crear nueva cita
- Ver calendario
- Editar cita
- Cancelar cita

- ☐ **Ventas/POS**

- Crear venta
- Registrar pago
- Ver historial de ventas

- ☐ **Inventario**

- Ver productos
- Crear producto
- Actualizar stock

- ☐ **Reportes**

- Ver dashboard
- Generar reportes
- Exportar datos

2. Interfaz en Español

- ☐ Todos los menús en español
- ☐ Formularios en español
- ☐ Mensajes de error en español
- ☐ Notificaciones en español

Monitoreo Continuo

Diario

- ☐ Verificar que la aplicación está corriendo
- ☐ Revisar logs de errores

- [] Verificar espacio en disco

Semanal

- [] Verificar que los backups se están creando
- [] Revisar logs de backup
- [] Verificar integridad de backups recientes

Mensual

- [] Probar restauración de backup en ambiente de prueba
- [] Revisar espacio usado por backups
- [] Limpiar logs antiguos si es necesario
- [] Actualizar documentación si hay cambios

Troubleshooting Rápido

Aplicación no inicia

```
# Ver logs del contenedor
docker logs <container-id>

# Verificar DATABASE_URL
echo $DATABASE_URL

# Verificar conexión a PostgreSQL
docker exec -it <container-id> pg_isready -h $PGHOST
```

Datos se pierden en redeploy

```
# Verificar volumen de PostgreSQL
docker exec -it <postgres-container> df -h | grep postgresql

# Verificar que el volumen es persistente
docker volume ls | grep postgres

# Verificar permisos
docker exec -it <postgres-container> ls -la /var/lib/postgresql/data
```

Backups no se crean

```
# Verificar permisos del script
docker exec -it <app-container> ls -la /app/scripts/pg_backup.sh

# Ejecutar backup manual
docker exec -it <app-container> /app/scripts/pg_backup.sh

# Ver logs
docker exec -it <app-container> tail -f /backup-citaplanner/logs/backup_*.log
```

Contactos de Emergencia

Documentación

- [Guía Completa de Persistencia](#) (./docs/DB-PERSISTENCIA.md)
- [Configuración de Volúmenes](#) (./docs/EASYPANEL-VOLUME-CONFIG.md)
- [Guía de Backup y Restauración](#) (./docs/BACKUP-RESTORE-GUIDE.md)

Comandos Útiles

```
# Ver estado de servicios
docker ps

# Ver logs en tiempo real
docker logs -f <container-id>

# Conectar a PostgreSQL
docker exec -it <postgres-container> psql $DATABASE_URL

# Ejecutar backup manual
docker exec -it <app-container> /app/scripts/pg_backup.sh

# Verificar espacio en disco
docker exec -it <app-container> df -h
```

Notas Finales

- ☒ El seed es idempotente - puede ejecutarse múltiples veces sin duplicar datos
- ☒ Los backups se rotan automáticamente (7 días, 4 semanas, 6 meses)
- ☒ Todos los datos persisten entre deployments
- ☒ La interfaz está completamente en español
- ☒ Las APIs retornan respuestas estandarizadas

¡Deployment exitoso! 🎉

Fecha: 9 de Octubre, 2025

Versión: 1.0.0

Branch: feature/db-persistence