

# DATOS PERSISTENTES EN PRODUCCIÓN

## Índice

1. Configuración Actual
  2. Cómo Limpiar Datos Demo
  3. Garantía de Persistencia
  4. Verificación
  5. Troubleshooting
- 

## Configuración Actual

### Cambios Implementados

1. **Seed Automático DESACTIVADO**
    - El script `start.sh` ya NO ejecuta seed en cada deploy
    - Los datos existentes se preservan entre deploys
    - Solo se ejecuta `prisma db push` (sincroniza esquema sin borrar datos)
  2. **Scripts de Limpieza Creados**
    - `clean-demo-data.sql` - Script SQL para limpiar datos demo
    - `clean-demo-data.sh` - Script automatizado con confirmación
  3. **Modo Producción**
    - Todos los datos capturados son persistentes
    - La base de datos PostgreSQL mantiene los datos entre deploys
    - Solo el esquema se actualiza, no los datos
- 

## Cómo Limpiar Datos Demo

### Método 1: Script Automatizado (RECOMENDADO)

Este script eliminará TODOS los datos demo excepto el usuario admin.

### En tu Servidor de Producción (Coolify):

1. Conectarse al container:

```
# Desde Coolify, ve a "Console" o usa SSH
docker exec -it [nombre-container-muebleria] sh
```

2. Configurar `DATABASE_URL` (si no está ya configurada):

```
export DATABASE_URL="postgres://usuario:password@host:5432/database"
```

3. Ejecutar el script de limpieza:

```
cd /app
sh clean-demo-data.sh
```

4. **Confirmar:**

- Escribe SI cuando se te pida confirmación
- El script eliminará todos los datos demo
- Solo se mantendrá el usuario admin (admin@admin.com)

## Método 2: Script SQL Manual

Si prefieres más control, ejecuta el SQL directamente:

```
# Conectarse a la base de datos
psql "$DATABASE_URL"

# O copiar y pegar el contenido de clean-demo-data.sql
```

## Método 3: Desde Prisma Studio

1. **Iniciar Prisma Studio** (en desarrollo local):

```
cd app
npx prisma studio
```

2. **Eliminar registros manualmente:**

- Abre cada tabla
  - Selecciona y elimina los registros demo
  - Guarda los cambios
- 

## Garantía de Persistencia

### ¿Cómo se Garantiza la Persistencia?

1. **Base de Datos Externa**

- PostgreSQL está en un servidor separado (no en el container)
- Los datos están en el servidor de base de datos, no en Docker
- El container solo se conecta a la BD, no la contiene

2. **Prisma db push**

- Solo actualiza el ESQUEMA (estructura de tablas)
- **NO borra datos existentes**
- Solo agrega/modifica columnas si es necesario

3. **Seed Desactivado**

- El script `start.sh` **NO** ejecuta seed
- Mensaje en logs: **Seed automático DESACTIVADO** (modo producción)
- Los datos solo se crean cuando TÚ los creas manualmente

## Qué Pasa en Cada Deploy

*# Coolify hace deploy:*

1. Construye nueva imagen Docker
  2. Crea nuevo container
  3. Container se conecta a la MISMA base de datos
  4. Prisma sincroniza esquema (sin borrar datos)
  5. Container inicia y lee los datos existentes
  6. TUS DATOS SIGUEN AHÍ
- 

## Verificación

### Verificar que Seed NO se Ejecuta

1. Revisa los logs de Coolify después de un deploy:

CORRECTO:

Seed automático DESACTIVADO (modo producción)  
Los datos existentes serán preservados

INCORRECTO (si ves esto, hay un problema):

Seeding database...  
Database seeded successfully

2. Verifica el contenido de start.sh:

```
cat start.sh | grep -A 5 "PRODUCCIÓN"
```

Debería mostrar:

```
# PRODUCCIÓN: NO EJECUTAR SEED AUTOMÁTICAMENTE  
echo " Seed automático DESACTIVADO (modo producción)"
```

### Verificar Persistencia de Datos

1. Crea un cliente de prueba:
  - Ve a tu app: <https://app.mueblerialaeconomica.com>
  - Crea un cliente llamado “TEST PERSISTENCIA”
  - Anota el ID o nombre
2. Haz un redeploy en Coolify:
  - Click en “Redeploy” o “Force Deploy”
  - Espera a que termine
3. Verifica que el cliente sigue ahí:
  - Abre la app nuevamente
  - Ve a la lista de clientes
  - El cliente “TEST PERSISTENCIA” debe seguir existiendo

---

## Troubleshooting

**Problema:** Los datos se borran en cada deploy

**Causas posibles:**

1. El seed se está ejecutando automáticamente
  - **Solución:** Verifica `start.sh` y confirma que dice “Seed automático DESACTIVADO”
  - **Revisa logs:** Busca “ Seeding database” en los logs
  - **Si aparece seed:** Hay un problema con el script
2. Prisma está usando `db push --force-reset`
  - **Solución:** Verifica que `start.sh` use solo `prisma db push --skip-generate`
  - **NO debe tener:** `--force-reset` o `--accept-data-loss`
3. Base de datos incorrecta
  - **Solución:** Verifica que `DATABASE_URL` apunte a la BD de producción
  - **Comando:** `docker exec -it [container] env | grep DATABASE_URL`

**Problema:** No puedo limpiar datos demo

**Solución 1:** Permisos

```
# Dar permisos de ejecución
chmod +x clean-demo-data.sh
```

```
# Ejecutar nuevamente
sh clean-demo-data.sh
```

**Solución 2:** `DATABASE_URL` no configurada

```
# Configurar la variable
export DATABASE_URL="postgresql://user:password@host:5432/database"
```

```
# Ejecutar script
sh clean-demo-data.sh
```

**Solución 3:** Usar SQL directamente

```
# Conectarse a la BD
psql "$DATABASE_URL"
```

```
# Copiar el contenido de clean-demo-data.sql
# y ejecutarlo en el prompt de psql
```

**Problema:** Error al ejecutar `clean-demo-data.sh`

**Error:** “psql: command not found”

```
# El script intentará usar Prisma automáticamente
# Asegúrate de estar en el directorio correcto
cd /app
sh clean-demo-data.sh

Error: "Prisma Client not found"

# Generar Prisma Client primero
npx prisma generate

# Ejecutar script nuevamente
sh clean-demo-data.sh
```

---

## Resumen de Archivos

### Archivos Modificados

Archivo	Descripción	Cambio
start.sh	Script de inicio del container	Seed desactivado, mensaje de producción
clean-demo-data.sql	Script SQL para limpiar datos	Nuevo archivo creado
clean-demo-data.sh	Script automatizado de limpieza	Nuevo archivo creado

### Archivos SIN Modificar

Archivo	Razón
package.json	Prisma seed aún existe pero NO se ejecuta (start.sh lo previene)
scripts/seed.ts	Archivo de seed existe pero NO se usa en producción
prisma/schema.prisma	Esquema de BD sin cambios

---

## Checklist de Producción

Antes de empezar a usar la app en producción, verifica:

- ☐ **Variables de Entorno Configuradas:**
  - DATABASE\_URL apunta a PostgreSQL de producción
  - NEXTAUTH\_URL es <https://app.mueblerialaeconomica.com>

- NEXTAUTH\_SECRET configurado correctamente
  - ❑ **Datos Demo Limpiados:**
    - Ejecutar `clean-demo-data.sh` para limpiar datos demo
    - Solo usuario admin debe existir
    - Confirmar en la interfaz que no hay datos demo
  - ❑ **Seed Desactivado:**
    - Verificar logs: debe decir “ Seed automático DESACTIVADO”
    - NO debe aparecer “ Seeding database”
  - ❑ **Persistencia Verificada:**
    - Crear un registro de prueba
    - Hacer redeploy
    - Verificar que el registro sigue existiendo
  - ❑ **Documentación Actualizada:**
    - Este documento está en el repositorio
    - Scripts de limpieza están en el repositorio
    - Equipo sabe cómo limpiar datos si es necesario
- 

## Empezar a Usar en Producción

### Paso 1: Limpiar Datos Demo

*# Conectarse al container de Coolify*  
`docker exec -it [nombre-container] sh`

*# Ejecutar script de limpieza*  
`cd /app`  
`sh clean-demo-data.sh`

*# Confirmar con "SI"*

### Paso 2: Verificar Usuario Admin

1. Ve a: <https://app.mueblerialaeconomica.com/login>
2. Inicia sesión con:
  - Email: `admin@admin.com`
  - Password: `admin123` (o el que hayas configurado)

### Paso 3: Comenzar a Capturar Datos Reales

1. **Clientes:** Agrega tus clientes reales
2. **Productos:** Agrega tu inventario real
3. **Proveedores:** Agrega tus proveedores reales
4. **Ventas:** Registra ventas reales
5. **Gastos:** Registra gastos reales

#### Paso 4: Verificar Persistencia

1. Haz un redeploy en Coolify
  2. Verifica que todos los datos siguen ahí
  3. **¡Listo! Tu sistema está en producción con datos persistentes**
- 

### Soporte

Si tienes problemas con la persistencia de datos:

1. **Revisa los logs** de Coolify:
    - Busca mensajes de error
    - Confirma que seed NO se ejecuta
    - Verifica que `db push` se completa exitosamente
  2. **Verifica la base de datos:**
    - Confirma que `DATABASE_URL` es correcta
    - Asegúrate de que la BD está accesible
    - Revisa permisos de usuario de BD
  3. **Contacta al equipo técnico:**
    - Proporciona los logs completos
    - Describe qué datos se perdieron
    - Indica cuándo ocurrió el problema
- 

**Fecha:** 2025-10-11

**Versión:** 1.0

**Estado:** PRODUCCIÓN - Datos Persistentes Activados

---

### Seguridad y Backups

#### Recomendaciones

1. **Backups Regulares:**
    - Configura backups automáticos de PostgreSQL
    - Frecuencia recomendada: Diario
    - Retención: Mínimo 7 días
  2. **Punto de Restauración:**
    - Antes de hacer cambios grandes, haz backup manual
    - Guarda el backup antes de ejecutar `clean-demo-data.sh`
  3. **Monitoreo:**
    - Revisa logs después de cada deploy
    - Verifica que los datos persisten
    - Alerta si aparece “Seeding database” en logs de producción
-

**¡Tu sistema ahora está configurado para mantener datos persistentes en producción!**