🛡️ SISTEMA DE PERSISTENCIA Y RESPALDOS

Mueblería La Económica

Documentación completa del sistema de persistencia de datos y respaldos automáticos.

📋 Tabla de Contenidos

🔧 Configuración de Persistencia

Volúmenes Docker

La aplicación utiliza volúmenes persistentes de Docker para garantizar que los datos NO se pierdan durante los deploys:

```
volumes:
  postgres_data:      # Datos de PostgreSQL
  app-data:           # Cache de Next.js
```

Verificación

Para verificar que la persistencia está correctamente configurada:

```
./validate-persistence.sh
```

Este script verifica:
- ✅ Configuración de volúmenes en docker-compose.yml
- ✅ Estado de los volúmenes de Docker
- ✅ Conexión a la base de datos
- ✅ Estadísticas de datos almacenados
- ✅ Sistema de respaldos
- ✅ Scripts disponibles

🌱 Seed Inteligente

¿Qué es?

El **seed inteligente** (`seed-safe.ts`) es una versión mejorada del seed original que:

- ✅ **NO elimina datos existentes**
- ✅ Solo inserta datos si no existen
- ✅ Preserva toda la información de producción
- ✅ Crea usuarios esenciales si faltan
- ✅ Es seguro para ejecutar en producción

Diferencias con el Seed Original

Característica	Seed Original	Seed Inteligente
Elimina datos	❌ Sí (deleteMany)	✅ No
Verifica existencia	❌ No	✅ Sí
Seguro en producción	❌ No	✅ Sí
Datos de demo	✅ Sí	❌ No (solo usuarios esenciales)

Uso

```
# Opción 1: Desde el host
./run-seed-safe.sh

# Opción 2: Desde docker compose
docker compose exec app sh -c "cd /app && npx tsx scripts/seed-safe.ts"

# Opción 3: Dentro del contenedor
npx tsx scripts/seed-safe.ts
```

Usuarios Creados

El seed inteligente garantiza que existan estos usuarios:

Usuario	Email	Contraseña	Rol
Admin	admin@economica.local	admin123	admin
Gestor	gestor@economica.local	gestor123	gestor_cobranza
Cobrador	cobrador@economica.local	cobrador123	cobrador
Reportes	reportes@economica.local	reportes123	reporte_cobranza

---

💾 Sistema de Respaldos

Crear Respaldo Manual

```
./backup-database.sh
```

Esto creará:
- 📁 Un archivo comprimido en `backups/backup_YYYY-MM-DD_HH-MM-SS.sql.gz`
- 🔗 Un enlace simbólico `latest.sql.gz` al respaldo más reciente
- 🧹 Limpieza automática (mantiene últimos 30 respaldos)

Características

- ✅ Compresión GZIP para ahorrar espacio
- ✅ Nombres con timestamp para fácil identificación
- ✅ Incluye estructura completa de la base de datos
- ✅ Compatible con restauración limpia
- ✅ Rotación automática de respaldos antiguos

Estructura de Backups

```
backups/
├── backup_2025-10-09_10-30-00.sql.gz
├── backup_2025-10-09_14-30-00.sql.gz
├── backup_2025-10-09_18-30-00.sql.gz
└── latest.sql.gz -> backup_2025-10-09_18-30-00.sql.gz
```

⏰ Respaldos Automáticos

Configurar

```
./cron-backup.sh
```

Opciones Disponibles

1. **Diario a las 2:00 AM**
   - Ideal para: Producción con cambios moderados
   - Cron: `0 2 * * *`
2. **Cada 12 horas** (2:00 AM y 2:00 PM)
   - Ideal para: Producción activa
   - Cron: `0 2,14 * * *`
3. **Cada 6 horas**
   - Ideal para: Alta actividad
   - Cron: `0 */6 * * *`
4. **Personalizado**
   - Define tu propia expresión cron

Verificar Configuración

```
# Ver cron jobs activos
crontab -l

# Ver log de respaldos
tail -f backups/backup.log
```

Desactivar Respaldos Automáticos

```
crontab -e
# Elimina la línea que contiene: cron-backup-wrapper.sh
```

🔄 Restauración de Datos

Restaurar desde Respaldo

```
./restore-database.sh
```

Proceso Interactivo

1. Se muestra lista de respaldos disponibles
2. Selecciona el respaldo a restaurar
3. Confirma la operación (escribe “SI”)
4. La base de datos se restaura automáticamente

Opciones

- **Número**: Selecciona un respaldo específico
- **L**: Restaura el último respaldo (latest)

⚠️ ADVERTENCIA

La restauración **ELIMINA TODOS LOS DATOS ACTUALES** y los reemplaza con el respaldo seleccionado. Esta operación **NO se puede deshacer**.

Ejemplo

```
$ ./restore-database.sh

================================
  RESTAURAR BASE DE DATOS
================================

📋 Backups disponibles:

  [1] backup_2025-10-09_18-30-00.sql.gz (2.3M) - 2025-10-09 18:30:00
  [2] backup_2025-10-09_14-30-00.sql.gz (2.1M) - 2025-10-09 14:30:00
  [3] backup_2025-10-09_10-30-00.sql.gz (1.9M) - 2025-10-09 10:30:00

También puedes usar:
  [L] Último backup (latest)

Selecciona el número del backup a restaurar (o 'L' para el último): 1

⚠️  ADVERTENCIA:
   Esta operación eliminará TODOS los datos actuales
   y los reemplazará con el backup seleccionado:
   backup_2025-10-09_18-30-00.sql.gz

¿Estás seguro? (escribe 'SI' para continuar): SI

📦 Restaurando backup...
✅ Base de datos restaurada exitosamente
================================
```

📖 Mejores Prácticas

1. Respaldos Regulares

- ✅ Configura respaldos automáticos según tu nivel de actividad
- ✅ Verifica que los respaldos se están creando correctamente
- ✅ Prueba la restauración periódicamente

2. Antes de Cambios Importantes

```
# Siempre crea un respaldo antes de:
# - Migraciones de base de datos
# - Actualizaciones importantes
# - Cambios en el esquema

./backup-database.sh
```

3. Monitoreo

```
# Verifica el estado regularmente
./validate-persistence.sh

# Revisa los logs de respaldo
tail -f backups/backup.log
```

4. Almacenamiento Externo

Considera copiar los respaldos a un almacenamiento externo:

```
# Sincronizar con un servidor remoto
rsync -avz backups/ usuario@servidor:/ruta/backups/

# O usar un servicio en la nube
# aws s3 sync backups/ s3://mi-bucket/backups/
```

5. Seed en Producción

- ✅ Usa SIEMPRE `./run-seed-safe.sh` en producción
- ❌ NUNCA uses el seed original que elimina datos

🆘 Solución de Problemas

El contenedor no guarda los datos

Problema: Los datos se pierden al reiniciar los contenedores.

Solución:

```
# 1. Verificar configuración
./validate-persistence.sh

# 2. Asegurarte de que el volumen existe
docker volume ls | grep postgres_data

# 3. Verificar docker-compose.yml
grep -A 5 "postgres:" docker-compose.yml
```

Error al crear respaldo

Problema: `backup-database.sh` falla.

Solución:

```
# 1. Verificar que PostgreSQL esté corriendo
docker compose ps postgres

# 2. Verificar conexión
docker compose exec postgres pg_isready -U postgres

# 3. Verificar permisos del directorio
ls -la backups/
```

Error al restaurar

Problema: `restore-database.sh` falla.

Solución:

```
# 1. Verificar que el archivo existe
ls -la backups/

# 2. Verificar que no esté corrupto
gunzip -t backups/latest.sql.gz

# 3. Intentar con otro respaldo
./restore-database.sh
# Selecciona un respaldo diferente
```

Seed elimina datos

Problema: El seed está eliminando datos de producción.

Solución:

```
# 1. DETENER inmediatamente cualquier operación de seed

# 2. Restaurar desde el último respaldo
./restore-database.sh

# 3. En el futuro, usa SOLO el seed seguro
./run-seed-safe.sh
```

Cron job no se ejecuta

Problema: Los respaldos automáticos no se crean.

Solución:

```
# 1. Verificar que el cron job existe
crontab -l

# 2. Verificar el log
tail -f backups/backup.log

# 3. Verificar permisos del script
ls -la cron-backup-wrapper.sh

# 4. Reconfigurar si es necesario
./cron-backup.sh
```

🎯 Resumen de Comandos

Validación y Estado

```
./validate-persistence.sh     # Verificar configuración completa
docker compose ps              # Ver estado de contenedores
docker volume ls               # Ver volúmenes de Docker
```

Datos

```
./run-seed-safe.sh            # Ejecutar seed seguro (no elimina datos)
```

Respaldos

```
./backup-database.sh          # Crear respaldo manual
./restore-database.sh         # Restaurar desde respaldo
./cron-backup.sh              # Configurar respaldos automáticos
```

Logs y Monitoreo

```
docker compose logs -f postgres    # Ver logs de PostgreSQL
tail -f backups/backup.log         # Ver log de respaldos
crontab -l                         # Ver cron jobs activos
```

✅ Checklist de Implementación

- [ ] Ejecutar `./validate-persistence.sh` para verificar configuración
- [ ] Crear primer respaldo manual: `./backup-database.sh`
- [ ] Configurar respaldos automáticos: `./cron-backup.sh`
- [ ] Ejecutar seed seguro: `./run-seed-safe.sh`
- [ ] Probar restauración en un ambiente de prueba
- [ ] Configurar sincronización con almacenamiento externo (opcional)
- [ ] Documentar procedimientos específicos de tu organización

---

📞 Soporte

Si encuentras problemas:

1. Ejecuta `./validate-persistence.sh` y revisa el output
2. Revisa los logs: `docker compose logs -f postgres`
3. Verifica el log de respaldos: `cat backups/backup.log`
4. Consulta esta documentación

Creado: 9 de octubre de 2025
Proyecto: Mueblería La Económica
Versión: 1.0.0

END