

RECURSOS COMPLETOS - Seed en Producción

ARCHIVOS DISPONIBLES

muebleria_la_economica/	
├──  SCRIPTS EJECUTABLES	
├── run-seed-production.sh	★ Para servidor local/VPS
├── run-seed-docker.sh	★ Para contenedores Docker
├──  DOCUMENTACIÓN	
├── README-SEED.md	 Guía rápida (START HERE)
├── INSTRUCCIONES-SEED-PRODUCCION.md	 Instrucciones paso a paso
├── SEED-PRODUCTION-GUIDE.md	 Guía completa y detallada
├── SEED-SOLUTION-SUMMARY.md	 Resumen técnico de la solución
└──  VERSIONES PDF (para lectura offline)	
├── README-SEED.pdf	
├── INSTRUCCIONES-SEED-PRODUCCION.pdf	
├── SEED-PRODUCTION-GUIDE.pdf	
└── SEED-SOLUTION-SUMMARY.pdf	

INICIO ULTRA-RÁPIDO

Para Docker (EasyPanel, Coolify, etc.)

```
./run-seed-docker.sh
```

Para Servidor Local

```
./run-seed-production.sh
```

Comando Manual Universal

```
npx tsx --require dotenv/config scripts/seed.ts
```

CUÁL DOCUMENTO LEER SEGÚN TU NECESIDAD

“Solo quiero ejecutar el seed YA”

Lee: README-SEED.md (2 minutos)

```
cat README-SEED.md
```

“Necesito instrucciones paso a paso para mi plataforma”

Lee: INSTRUCCIONES-SEED-PRODUCCION.md (5 minutos)

```
cat INSTRUCCIONES-SEED-PRODUCCION.md
```

“Quiero entender todo el proceso y solucionar problemas”

Lee: SEED-PRODUCTION-GUIDE.md (10 minutos)

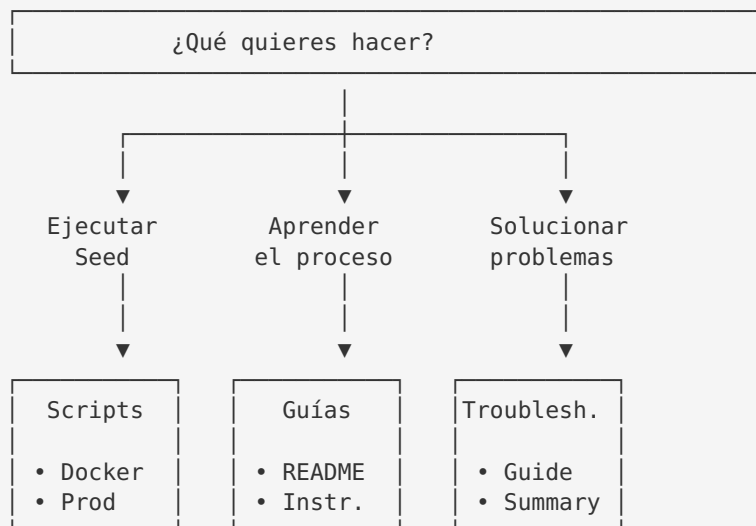
```
cat SEED-PRODUCTION-GUIDE.md
```

“Soy desarrollador y necesito detalles técnicos”

Lee: SEED-SOLUTION-SUMMARY.md (7 minutos)

```
cat SEED-SOLUTION-SUMMARY.md
```

MAPA VISUAL DE USO



COMANDOS POR PLATAFORMA

EasyPanel

```
# Opción 1: Terminal de EasyPanel
npx tsx --require dotenv/config scripts/seed.ts

# Opción 2: SSH + Script
./run-seed-docker.sh nombre_contenedor

# Opción 3: Docker directo
docker exec contenedor npx tsx --require dotenv/config scripts/seed.ts
```

Coolify

```
# Opción 1: Execute Command (UI)
npx tsx --require dotenv/config scripts/seed.ts

# Opción 2: SSH + Script automatizado
./run-seed-docker.sh

# Opción 3: Docker directo
docker exec $(docker ps | grep coolify-app | awk '{print $1}') \
npx tsx --require dotenv/config scripts/seed.ts
```

Docker Compose

```
# Opción 1: Script (detecta automáticamente)
./run-seed-docker.sh

# Opción 2: Docker Compose
docker-compose exec app npx tsx --require dotenv/config scripts/seed.ts

# Opción 3: Docker directo
docker exec muebleria_app_1 npx tsx --require dotenv/config scripts/seed.ts
```

VPS/Servidor Directo

```
# Opción 1: Script automatizado
./run-seed-production.sh

# Opción 2: Manual en app/
cd app && npx tsx --require dotenv/config scripts/seed.ts
```



CARACTERÍSTICAS DE LOS SCRIPTS

run-seed-docker.sh

Características:

- 🔍 Detecta contenedores automáticamente
- ✅ Verifica estructura de archivos
- ↺ 3 métodos de fallback

- 🎨 Output colorizado
- 📝 Mensajes claros de error
- 🛡️ Validaciones exhaustivas

Uso:

```
# Detección automática
./run-seed-docker.sh

# Manual
./run-seed-docker.sh nombre_contenedor
```

Métodos que prueba:

1. `npx tsx` (principal)
 2. `yarn prisma db seed` (fallback)
 3. Instrucciones manuales (si falla)
-

✨ **run-seed-production.sh**

Características:

- 🔧 3 métodos diferentes
- ✅ Verifica configuración
- 📦 Instala dependencias si es necesario
- 🎨 Output colorizado
- 📝 Mensajes de ayuda
- 🛡️ Validaciones de `.env`

Uso:

```
./run-seed-production.sh
```

Métodos que prueba:

1. `npx tsx` (recomendado)
 2. `ts-node` (si está disponible)
 3. `yarn add tsx` + ejecutar (último recurso)
-

EJEMPLOS PRÁCTICOS

Ejemplo 1: Primera vez en EasyPanel

```
# 1. SSH al servidor
ssh root@mi-servidor.com

# 2. Ir al proyecto (o clonar si no existe)
cd /opt
git clone https://github.com/qhosting/muebleria-la-economica.git
cd muebleria-la-economica

# 3. Encontrar contenedor
docker ps | grep muebleria
# Output: abc123  easypanel-muebleria...

# 4. Ejecutar seed
./run-seed-docker.sh abc123

# 5. Verificar
curl -X POST https://app.mueblerialaeconomica.com/api/auth/signin \
  -H "Content-Type: application/json" \
  -d '{"email":"admin@economica.local","password":"admin123"}'
```

Ejemplo 2: Coolify con detección automática

```
# 1. Clonar repo en servidor
ssh root@coolify-server.com
cd /opt && git clone https://github.com/qhosting/muebleria-la-economica.git
cd muebleria-la-economica

# 2. Ejecutar (detecta automáticamente)
./run-seed-docker.sh

# Output:
# 🚀 SEED EN CONTENEDOR DOCKER
# 📦 Buscando contenedores...
# ✓ Usando contenedor: coolify-app-xyz
# 🌱 Ejecutando seed...
# ✅ ¡Seed completado exitosamente!
```

Ejemplo 3: Docker Compose local

```
# 1. Levantar servicios
docker-compose up -d

# 2. Esperar a que estén listos
docker-compose ps

# 3. Ejecutar seed (método 1 - automático)
./run-seed-docker.sh

# 0 método 2 - explícito
docker-compose exec app npx tsx --require dotenv/config scripts/seed.ts

# 4. Verificar
open http://localhost:3000
# Login: admin@economica.local / admin123
```

VERIFICACIÓN COMPLETA

Check 1: Salud de la aplicación

```
curl https://app.mueblerialaeconomica.com/api/health
# Esperado: {"status":"ok"}
```

Check 2: Login API

```
curl -X POST https://app.mueblerialaeconomica.com/api/auth/signin \
-H "Content-Type: application/json" \
-d '{"email":"admin@economica.local","password":"admin123"}'
# Esperado: {"token":"...", "user":{"..."}}
```

Check 3: Base de datos

```
docker exec postgres_container psql -U usuario -d database -c \
"SELECT email, rol FROM \"User\";"
# Esperado: Lista de 4 usuarios
```

Check 4: Interface web

1. Abrir: <https://app.mueblerialaeconomica.com>
2. Login: admin@economica.local / admin123
3. Verificar: Dashboard se carga correctamente
4. Verificar: Menú muestra todas las opciones de admin

TROUBLESHOOTING RÁPIDO

“tsx: not found”

✓ Solución: Usar npx tsx en lugar de tsx
npx tsx --require dotenv/config scripts/seed.ts

“Container not found”

✓ Solución: Ver contenedores y usar nombre exacto
docker ps
./run-seed-docker.sh nombre_exacto

“DATABASE_URL not defined”

✓ Solución: Verificar y agregar .env
docker exec contenedor cat .env | grep DATABASE_URL
docker exec contenedor sh -c 'echo "DATABASE_URL=..." >> .env'

“Can’t reach database”

✓ Solución: Verificar red y conectividad
docker network ls
docker network inspect nombre_red
docker exec app_container ping postgres_host

“Prisma Client not found”

✓ Solución: Generar cliente
docker exec contenedor npx prisma generate
./run-seed-docker.sh contenedor

DESCARGA Y USO

Clonar el repositorio

```
git clone https://github.com/qhosting/muebleria-la-economica.git
cd muebleria-la-economica
```

Dar permisos de ejecución (si es necesario)

```
chmod +x run-seed-production.sh
chmod +x run-seed-docker.sh
```

Ejecutar

```
# Para Docker
./run-seed-docker.sh





# Para servidor local
./run-seed-production.sh
```

USUARIOS CREADOS





Email	Password	Rol	Permisos
ad-min@economica.local	admin123	admin	Todos
gestor@economica.local	gestor123	gestor_cobranza	Gestión de cobranza
co-brador@economica.local	cobrador123	cobrador	Cobros y rutas
repor-tes@economica.local	reportes123	reporte_cobranza	Solo reportes

ÍNDICE DE DOCUMENTACIÓN






README-SEED.md

-  Guía rápida de inicio
-  Comandos esenciales
-  Ejemplos básicos
-  Problemas comunes
- **Tiempo de lectura:** 2-3 minutos
- **Ideal para:** Primera vez usando los scripts





INSTRUCCIONES-SEED-PRODUCCION.md

-  Instrucciones paso a paso
-  Casos de uso específicos por plataforma
-  Ejemplos reales
-  Checklist de despliegue
- **Tiempo de lectura:** 5-7 minutos
- **Ideal para:** Seguir un proceso estructurado

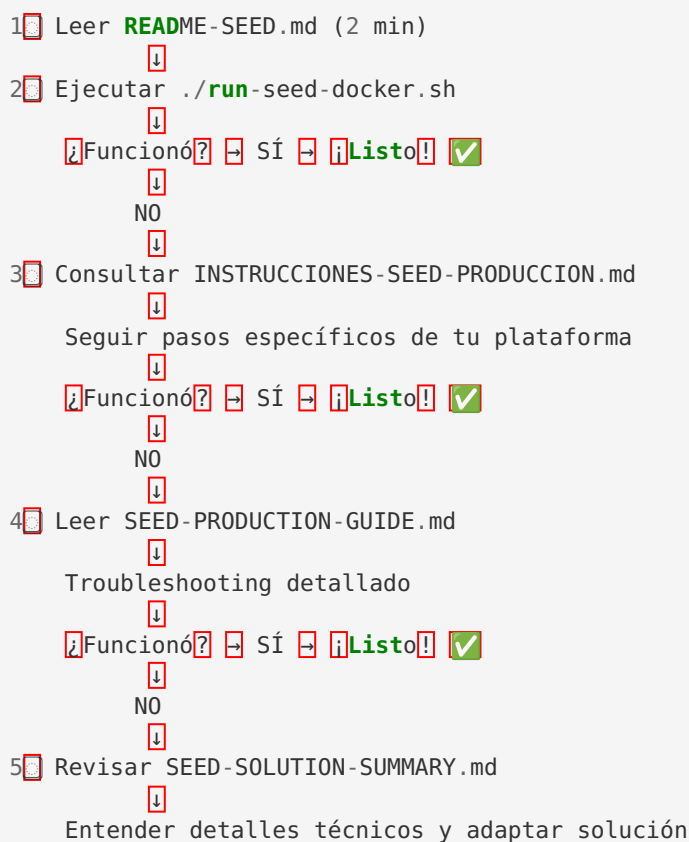
SEED-PRODUCTION-GUIDE.md

-  Guía completa y exhaustiva
-  Explicación del problema
-  Todas las soluciones posibles
-  Troubleshooting detallado
-  Enlaces a recursos externos
- **Tiempo de lectura:** 10-15 minutos
- **Ideal para:** Entender todo el contexto

SEED-SOLUTION-SUMMARY.md

-  Resumen técnico de la solución
-  Arquitectura de los scripts
-  Flujos de ejecución
-  Detalles de implementación
- **Tiempo de lectura:** 7-10 minutos
- **Ideal para:** Desarrolladores que quieren detalles técnicos

FLUJO RECOMENDADO



ENLACES IMPORTANTES

- **Repositorio:** <https://github.com/qhosting/muebleria-la-economica>
- **Issues:** <https://github.com/qhosting/muebleria-la-economica/issues>
- **Documentación Prisma:** <https://www.prisma.io/docs/guides/database/seed-database>
- **Documentación npx:** <https://docs.npmjs.com/cli/v8/commands/npx>

CHECKLIST FINAL

Antes de considerar que todo está listo:

- ☐ Scripts ejecutables descargados
- ☐ Permisos de ejecución configurados
- ☐ Contenedor/servidor identificado
- ☐ Seed ejecutado exitosamente
- ☐ Login probado con `admin@economica.local`
- ☐ Usuarios verificados en base de datos
- ☐ Dashboard accesible y funcionando
- ☐ Passwords cambiados en producción
- ☐ Backup de base de datos realizado
- ☐ Documentación revisada

TIPS Y MEJORES PRÁCTICAS

1. **Siempre usar** `npx tsx` en producción, no `tsx` directamente
2. **Hacer backup** antes de ejecutar seed en producción con datos reales
3. **Cambiar passwords** inmediatamente después del seed
4. **Verificar el login** de todos los usuarios después del seed
5. **Revisar logs** si algo falla: `docker logs -f contenedor`
6. **Usar scripts automatizados** en lugar de comandos manuales
7. **Documentar cambios** si personalizas el seed
8. **Probar en staging** antes de producción si es posible

¡TODO LISTO!

Con estos recursos tienes todo lo necesario para:

- ☒ Ejecutar seed en cualquier plataforma
- ☒ Solucionar problemas comunes
- ☒ Entender el proceso completo
- ☒ Personalizar según tus necesidades
- ☒ Mantener y documentar tu sistema

¡Feliz deployment! 

Fecha: 30 de Septiembre, 2025

Versión: 1.0.0

Mantenido por: Sistema de Gestión Mueblería La Económica

Licencia: MIT

Estado:  Producción - Probado y Funcionando