

Guía de Seed en Producción - Mueblería La Económica

Índice

1. [El Problema](#)
 2. [Soluciones Disponibles](#)
 3. [Métodos de Ejecución](#)
 4. [Troubleshooting](#)
-

El Problema

El comando `yarn prisma db seed` falla en producción con el error:

```
/bin/sh: tsx: not found
```

Causa: `tsx` es una dependencia de desarrollo (`devDependencies`) que no se instala en producción por razones de optimización.

Soluciones Disponibles

Opción 1: Script Automatizado (Recomendado)

Hemos creado scripts que manejan automáticamente todas las alternativas:

A) Para entornos locales/servidor:

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

B) Para contenedores Docker:

```
./run-seed-docker.sh [nombre-contenedor]
```

Si no especificas el nombre del contenedor, el script lo detectará automáticamente.

Opción 2: Ejecución Manual

En el servidor (fuera del contenedor):

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

Dentro de un contenedor Docker:

```
# Primero, acceder al contenedor
docker exec -it nombre_contenedor sh

# Luego ejecutar el seed
npx tsx --require dotenv/config scripts/seed.ts
```

Opción 3: Usando Docker Exec Directo

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



Métodos de Ejecución

Para EasyPanel

1. Via SSH en el servidor:

```
# Conectar al servidor
ssh usuario@tu-servidor

# Ir al directorio del proyecto
cd /path/to/muebleria_la_economica

# Ejecutar el script
./run-seed-docker.sh nombre_contenedor_easypanel
```

1. Desde la terminal de EasyPanel:

```
# En el contenedor de la aplicación
npx tsx --require dotenv/config scripts/seed.ts
```

Para Coolify

1. Via interfaz web de Coolify:

- Ve a tu aplicación
- Click en "Execute Command"
- Ejecuta: `npx tsx --require dotenv/config scripts/seed.ts`

2. Via SSH en el servidor:

```
# Encontrar el nombre del contenedor
docker ps | grep muebleria

# Ejecutar el script
./run-seed-docker.sh nombre_contenedor_coolify
```

Para Docker Compose Local

```
# Método 1: Usando el script
./run-seed-docker.sh muebleria_app

# Método 2: Usando docker-compose exec
docker-compose exec app npx tsx --require dotenv/config scripts/seed.ts
```



Troubleshooting

Error: “npx: command not found”

Solución:

```
# Instalar npm en el contenedor (Alpine)
apk add --no-cache npm

# Luego ejecutar el seed
npx tsx --require dotenv/config scripts/seed.ts
```

Error: “DATABASE_URL not found”

Causa: El archivo `.env` no está presente o no se está cargando.

Solución:

```
# Verificar que existe el .env
ls -la .env

# Verificar que contiene DATABASE_URL
cat .env | grep DATABASE_URL

# Si no existe, crear uno con la variable correcta
echo "DATABASE_URL=postgresql://usuario:password@host:5432/database" > .env
```

Error: “P1001: Can’t reach database server”

Causa: La base de datos no es accesible desde el contenedor.

Solución:

```
# 1. Verificar que la DB está corriendo
docker ps | grep postgres

# 2. Verificar la conexión desde el contenedor
docker exec contenedor_app ping postgres_host

# 3. Verificar el DATABASE_URL
docker exec contenedor_app env | grep DATABASE_URL
```

El seed se ejecuta pero no crea usuarios

Verificación:

```
# Conectar a la base de datos
psql $DATABASE_URL

# Verificar usuarios
SELECT email, role FROM "User";

# Si la tabla no existe, ejecutar migraciones
npx prisma migrate deploy
```



Datos Creados por el Seed

El seed crea los siguientes usuarios:

Email	Password	Rol	Descripción
ad-min@economica.local	admin123	admin	Administrador del sistema
gestor@economica.local	gestor123	gestor_cobranza	Gestor de cobranza
co-brador@economica.local	cobrador123	cobrador	Cobrador de campo
repor-tes@economica.local	reportes123	reporte_cobranza	Usuario de reportes

Flujo Completo Recomendado

Para primera vez en producción:

```
# 1. Verificar que la aplicación está desplegada
docker ps

# 2. Ejecutar migraciones (si es necesario)
docker exec contenedor_app npx prisma migrate deploy

# 3. Generar Prisma Client (si es necesario)
docker exec contenedor_app npx prisma generate

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

# 5. Verificar que los usuarios fueron creados
# Acceder a la aplicación y probar login con admin@economica.local / admin123
```

Scripts de Ayuda

Ver logs del contenedor:

```
docker logs -f nombre_contenedor
```

Ver logs en tiempo real:

```
docker logs --tail=50 -f nombre_contenedor
```

Ejecutar comando interactivo en el contenedor:

```
docker exec -it nombre_contenedor sh
```

Verificar variables de entorno:

```
docker exec nombre_contenedor env
```

Notas Adicionales

1. **El seed es idempotente:** Puedes ejecutarlo múltiples veces sin problemas. Usa `upsert` para no duplicar datos.
2. **Limpieza de datos:** El seed elimina todos los datos existentes antes de crear nuevos. ¡Ten cuidado en producción!
3. **Alternativa sin limpieza:** Si necesitas solo crear usuarios sin borrar datos existentes, puedes modificar el script `seed.ts` comentando las líneas de `deleteMany()`.

4. **Backup recomendado:** Antes de ejecutar el seed en producción con datos reales, haz un backup de la base de datos:

```
bash
```

```
pg_dump $DATABASE_URL > backup_$(date +%Y%m%d_%H%M%S).sql
```

Enlaces Útiles

- [Documentación de Prisma Seed](https://www.prisma.io/docs/guides/database/seed-database) (https://www.prisma.io/docs/guides/database/seed-database)
- [npx Documentation](https://docs.npmjs.com/cli/v8/commands/npx) (https://docs.npmjs.com/cli/v8/commands/npx)
- [Docker exec Reference](https://docs.docker.com/engine/reference/commandline/exec/) (https://docs.docker.com/engine/reference/commandline/exec/)

Última actualización: 30 de Septiembre, 2025

Versión: 1.0.0

Mantenido por: Sistema de Gestión Mueblería La Económica