



# Solución Error: tsx ENOENT - Seed



## Problema

```
Error: Command failed with ENOENT: tsx --require dotenv/config scripts/seed.ts
spawn tsx ENOENT
```

**Causa:** `tsx` es una devDependency y no está disponible en producción (Docker).



## Soluciones

### Solución 1: Usar npx (Recomendado)

En lugar de ejecutar el seed directamente, usa `npx` :

```
# Dentro del contenedor Docker
docker exec -it muebleria-app-1 sh -c "cd /app/app && npx tsx --require dotenv/config
scripts/seed.ts"
```

O si estás dentro del contenedor:

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

### Solución 2: Instalar tsx en Producción

Modifica el `Dockerfile` para incluir `tsx` en las dependencias de producción:

#### Opción A - Instalar solo para seed:

```
# En el Dockerfile, después de copiar los archivos
RUN npm install tsx --save
```

#### Opción B - Mover tsx a dependencies en package.json:

```
"dependencies": {
  "...": "...",
  "tsx": "4.20.3"
}
```

### Solución 3: Ejecutar con ts-node

Si `ts-node` está disponible:

```
docker exec -it muebleria-app-1 sh -c "cd /app/app && npx ts-node -r dotenv/config scripts/seed.ts"
```

## Solución 4: Script de Seed Simplificado

Crea un script shell que no dependa de tsx:

```
#!/bin/sh
# seed-prod.sh

cd /app/app
node -r dotenv/config -r ts-node/register scripts/seed.ts
```



## Solución Rápida Ahora Mismo

Si necesitas ejecutar el seed **ahora** sin modificar el Dockerfile:

```
# 1. Entrar al contenedor
docker exec -it muebleria-app-1 sh

# 2. Ir al directorio de la app
cd /app/app

# 3. Instalar tsx temporalmente
npm install tsx

# 4. Ejecutar el seed
npx tsx --require dotenv/config scripts/seed.ts

# 5. Salir
exit
```



## Prevenir Este Error en el Futuro

### Actualizar Dockerfile

Agrega la instalación de tsx para operaciones de mantenimiento:

```
# En el Dockerfile, después de COPY y antes de CMD
RUN npm install --only=production && npm install tsx
```

### Crear Script de Mantenimiento

Crea un script `maintenance.sh` :

```
#!/bin/sh
# maintenance.sh

echo "🔧 Instalando herramientas de mantenimiento..."
npm install tsx ts-node

echo "🌱 Ejecutando seed..."
npx tsx --require dotenv/config scripts/seed.ts

echo "✅ Mantenimiento completado"
```

Luego:

```
chmod +x maintenance.sh
docker exec -it muebleria-app-1 /app/maintenance.sh
```



## Alternativa: Seed Script en JavaScript Puro

Si quieres evitar dependencias de TypeScript, convierte el seed a JavaScript:

```
# Compilar el seed a JavaScript una vez
npx tsc scripts/seed.ts --outDir scripts/compiled

# Ejecutar el seed compilado
node -r dotenv/config scripts/compiled/seed.js
```



## Recomendación Final

### Para Desarrollo:

- Mantén `tsx` en devDependencies
- Usa `npm run seed` localmente

### Para Producción:

- Usa `npx tsx` cuando necesites ejecutar el seed
- O modifica el Dockerfile para incluir `tsx`
- O crea un script de mantenimiento dedicado



## Si Nada Funciona

Si ninguna solución funciona, puedes ejecutar el seed desde **fuera del contenedor**:

```
# 1. Tener las dependencias instaladas localmente
cd /home/ubuntu/muebleria_la_economica/app
npm install

# 2. Configurar la conexión a la base de datos
export DATABASE_URL="postgresql://user:password@host:5432/database"

# 3. Ejecutar el seed
npm run seed
```

---

## ✓ Verificación

Después de ejecutar el seed exitosamente:

```
# Verificar que se crearon los usuarios
docker exec -it muebleria-app-1 sh -c "cd /app/app && npx prisma studio"

# 0 desde la línea de comandos de Prisma
docker exec -it muebleria-app-1 sh -c "cd /app/app && npx prisma db pull"
```

---

## 📞 Notas Importantes

1. **Backup Primero:** Antes de ejecutar seed en producción, haz backup:

```
bash
```













```
docker exec -it postgres pg_dump -U postgres -d muebleria_db > backup_$(date +
%Y%m%d_%H%M%S).sql
```


2. **El Seed Limpia Datos:** El script de seed **elimina todos los datos existentes** antes de crear nuevos.
3. **Datos de Demo:** El seed crea 200 clientes de demostración y 4 usuarios base.


---






## 🎉 Resultado Esperado

Después de ejecutar el seed correctamente:

```
 Iniciando seeders...
 Limpiando datos existentes...
 Creando usuarios esenciales...
 Usuarios creados exitosamente
 Creando plantillas de ticket...
 Plantillas de ticket creadas
 Creando clientes...
 200 clientes creados exitosamente
 Creando pagos de ejemplo...
 Pagos de ejemplo creados
 Creando rutas de cobranza...
 Rutas de cobranza creadas

 ¡Seeders completados exitosamente!

 Resumen de datos creados:
- 4 usuarios
- 200 clientes
- ~50 pagos
- 2 plantillas de ticket
- 10 rutas de cobranza

 Credenciales de acceso:
 Admin: admin@economica.local / admin123
 Gestor: gestor@economica.local / gestor123
 Cobrador: cobrador@economica.local / cobrador123
 Reportes: reportes@economica.local / reportes123
```