



# Fix: Scripts faltantes en package.json

**Fecha:** 28 de octubre de 2025

**Commit:** Por aplicar

**Tipo:** Corrección crítica de build



## Problema Identificado

El build de Docker fallaba con:

```
npm error Missing script: "build"
```

## Causa Raíz

Al intentar mover `prisma` de `dependencies` a `devDependencies` usando `jq`, accidentalmente eliminé toda la sección `scripts` del `package.json`.

**Comando problemático:**

```
jq 'del(.dependencies.prisma) | .devDependencies.prisma = "6.7.0"'
```

Este comando eliminó todos los campos excepto `dependencies` y `devDependencies`.



## Solución Aplicada

### 1. Recuperar package.json del commit anterior:

```
bash
```

```
git show 877f376:app/package.json > /tmp/package.json.backup
```

### 2. Aplicar cambio preservando todos los campos:

```
bash
```

```
cat /tmp/package.json.backup | \
```

```
jq 'del(.dependencies.prisma) | .devDependencies.prisma = "6.7.0"'
```

```
> app/package.json
```

### 3. Regenerar package-lock.json:






```
bash
```

```
cd app && npm install --package-lock-only --legacy-peer-deps
```

## Scripts Restaurados

```
{
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "prisma": {
    "seed": "tsx --require dotenv/config scripts/seed.ts"
  }
}
```

## Resultado Esperado

1.  `npm run build` ahora funciona correctamente
2.  Todos los scripts están presentes
3.  `prisma` está en `devDependencies` (correcto)
4.  `@prisma/client` está en `dependencies` (runtime)
5.  Build de Docker completa exitosamente

## Archivos Modificados

|                                      |  |
|--------------------------------------|--|
| app/ <code>package</code> .json      | - Scripts restaurados, prisma en devDependencies |
| app/ <code>package</code> -lock.json | - Regenerado                                     |

## Próximos Pasos

1. Commit y push de cambios
2. Rebuild en EasyPanel (limpiar caché)
3. Verificar logs de build
4. Confirmar que “npm run build” ejecuta correctamente

## Lección Aprendida

**NO usar `jq` para modificaciones estructurales complejas de `package.json`**

En su lugar:

- Usar `npm pkg set` / `npm pkg delete` (npm 7+)
- O editar manualmente el archivo
- O usar un script más robusto que preserve todos los campos

**Comando correcto con jq (preserva todo):**

```
jq '.dependencies.prisma as $p |  
  del(.dependencies.prisma) |  
  .devDependencies.prisma = $p' package.json
```

---

**Estado:**  Listo para deploy

**Prioridad:**  CRÍTICA

**Testing:** Pendiente en EasyPanel