🔧 Fix: Prisma Generate en Build de Docker con NPM

Fecha: 28 de octubre de 2025
Commit: Por aplicar
Tipo: Corrección crítica de build

📋 Problema Identificado

El build de Docker fallaba en el paso de generación de Prisma Client:

```
Error: Command failed with exit code 1: npm i @prisma/client@6.7.0 --silent
```

Causas Raíz

1. `prisma` **CLI en dependencies incorrectas:**
   - Estaba en `dependencies` cuando debería estar en `devDependencies`
   - Prisma CLI solo se necesita durante el desarrollo y build, no en runtime

2. **Paso de limpieza innecesario en Dockerfile:**
   - Se borraba `node_modules/@prisma/client` antes de generar
   - Esto causaba que Prisma intentara reinstalarlo, fallando por conflictos de lockfile
   - `npm ci` ya instala todo correctamente, no es necesario limpiar

---

✅ Soluciones Aplicadas

1. Corregir ubicación de `prisma` en package.json

Antes:

```
{
  "dependencies": {
    "@prisma/client": "6.7.0",
    "prisma": "6.7.0"  // ❌ Ubicación incorrecta
  }
}
```

Después:

```
{
  "dependencies": {
    "@prisma/client": "6.7.0"  // ✅ Solo el runtime client
  },
  "devDependencies": {
    "prisma": "6.7.0"  // ✅ CLI en devDependencies
  }
}
```

2. Simplificar generación de Prisma en Dockerfile

Antes:

```
RUN echo "🔧 Limpiando y generando Prisma Client..." && \
    rm -rf node_modules/.prisma node_modules/@prisma/client && \
    npx prisma generate && \
    echo "✅ Prisma Client generado" && \
    # ... validaciones complejas ...
```

Después:

```
RUN echo "🔧 Generando Prisma Client..." && \
    npx prisma generate && \
    echo "✅ Prisma Client generado correctamente"
```

3. Regenerar package-lock.json

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

🎯 Resultado Esperado

1. ✅ `npm ci` instala todas las dependencias correctamente
2. ✅ `npx prisma generate` genera el cliente sin intentar reinstalar
3. ✅ El build de Docker completa exitosamente
4. ✅ La app inicia sin errores de Prisma Client

📦 Archivos Modificados

```
app/package.json          - Movido prisma a devDependencies
app/package-lock.json     - Regenerado con --legacy-peer-deps
Dockerfile                - Simplificado paso de Prisma generate
```

🚀 Próximos Pasos

1. Commit y push de cambios

2. Rebuild en EasyPanel (limpiar caché)
3. Verificar logs de build
4. Confirmar que la app inicia correctamente

📝 Notas Técnicas

- **NPM vs Yarn:** Proyecto migrado completamente a npm
- **Legacy Peer Deps:** Necesario por conflictos con @typescript-eslint
- **Prisma CLI:** Solo necesario durante build, no en runtime
- **Dockerfile Alpine:** Compatible con npm ci sin problemas

✨ Mejoras Adicionales

Este fix también:
- Reduce el tamaño de la imagen Docker (prisma CLI no se copia al runtime)
- Simplifica el proceso de build (menos pasos, menos puntos de fallo)
- Mejora la velocidad de build (sin limpieza innecesaria)
- Alinea mejor con las mejores prácticas de Prisma + Docker

Estado: ✅ Listo para deploy
Prioridad: 🔴 CRÍTICA
Testing: Pendiente en EasyPanel