



Scripts para Agregar al package.json

Instrucciones

Agrega estos scripts manualmente a tu archivo `nextjs_space/package.json` en la sección `"scripts"`.

Scripts Recomendados

```
{
  "scripts": {
    // ... scripts existentes ...

    // Scripts de Base de Datos
    "postinstall": "prisma generate",
    "db:migrate": "prisma migrate deploy",
    "db:seed": "prisma db seed",
    "db:studio": "prisma studio",
    "db:reset": "prisma migrate reset",

    // Scripts de Deployment
    "check:env": "node scripts/check-env.js",
    "vercel:build": "prisma generate && next build",
    "deploy:check": "node scripts/check-env.js && yarn build"
  }
}
```

Descripción de Scripts

Scripts de Base de Datos

postinstall

```
yarn install # Ejecuta automáticamente prisma generate después
```

Uso: Se ejecuta automáticamente después de `yarn install`

Descripción: Genera el cliente de Prisma

db:migrate

```
yarn db:migrate
```

Uso: Ejecutar migraciones en producción

Descripción: Aplica todas las migraciones pendientes sin crear nuevas

db:seed

```
yarn db:seed
```

Uso: Cargar datos de prueba

Descripción: Ejecuta el archivo `scripts/seed.ts` para poblar la base de datos

db:studio

```
yarn db:studio
```

Uso: Explorar la base de datos visualmente

Descripción: Abre Prisma Studio en `http://localhost:5555`

db:reset

```
yarn db:reset
```

Uso: Resetear la base de datos en desarrollo

Descripción: Elimina y recrea la base de datos, ejecuta migraciones y seeds

Scripts de Deployment

check:env

```
yarn check:env
```

Uso: Verificar variables de entorno antes de desplegar

Descripción: Verifica que todas las variables requeridas estén configuradas

vercel:build

```
yarn vercel:build
```

Uso: Build para Vercel

Descripción: Genera el cliente de Prisma y luego hace el build de Next.js

deploy:check

```
yarn deploy:check
```

Uso: Verificación completa antes de deployment

Descripción: Verifica variables de entorno y hace un build de prueba

Cómo Agregar los Scripts

Paso 1: Abrir el archivo

```
cd nextjs_space  
nano package.json # o usa tu editor favorito
```

Paso 2: Localizar la sección “scripts”

Busca la sección que empieza con `"scripts": {`

Paso 3: Agregar los nuevos scripts

Agrega cada script en una nueva línea, asegurándote de:

- Usar comas al final de cada línea (excepto la última)
- Mantener la indentación consistente

Ejemplo Completo

```
{
  "name": "app",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "test": "jest --watch",
    "test:ci": "jest --ci --coverage",
    "test:unit": "vitest",
    "test:unit:ui": "vitest --ui",
    "test:e2e": "playwright test",
    "test:e2e:ui": "playwright test --ui",
    "test:e2e:debug": "playwright test --debug",
    "test:all": "yarn test:ci && yarn test:e2e",
    "postinstall": "prisma generate",
    "db:migrate": "prisma migrate deploy",
    "db:seed": "prisma db seed",
    "db:studio": "prisma studio",
    "db:reset": "prisma migrate reset",
    "check:env": "node scripts/check-env.js",
    "vercel:build": "prisma generate && next build",
    "deploy:check": "node scripts/check-env.js && yarn build"
  },
  "prisma": {
    "seed": "tsx --require dotenv/config scripts/seed.ts"
  },
  ..
}
```

Paso 4: Guardar y verificar

```
# Verificar que el JSON es válido
yarn install

# Probar un script
yarn check:env
```

Uso en Vercel

Si agregaste `vercel:build`, actualiza la configuración en Vercel:

1. Ve a tu proyecto en Vercel
2. Settings → General → Build & Development Settings
3. Build Command: `cd nextjs_space && yarn vercel:build`

O actualiza el `vercel.json`:

```
{  
  "buildCommand": "cd nextjs_space && yarn vercel:build",  
  ...  
}
```

Troubleshooting

Error: “Cannot find module ‘scripts/check-env.js’”

Solución: Asegúrate de que el archivo `scripts/check-env.js` existe

```
ls nextjs_space/scripts/check-env.js
```

Error: “prisma: command not found”

Solución: Instala las dependencias

```
cd nextjs_space  
yarn install
```

Error al ejecutar seed

Solución: Verifica que el archivo `scripts/seed.ts` existe y está bien configurado

```
ls nextjs_space/scripts/seed.ts
```

Recursos Adicionales

- [Documentación de Prisma](https://www.prisma.io/docs) (https://www.prisma.io/docs)
 - [Documentación de Vercel](https://vercel.com/docs) (https://vercel.com/docs)
 - [npm scripts documentation](https://docs.npmjs.com/cli/v9/using-npm/scripts) (https://docs.npmjs.com/cli/v9/using-npm/scripts)
-

Última actualización: Diciembre 2024