

# Paquetes NPM

## 0) Requisitos

- Node.js (ideal LTS)
- npm / pnpm (el que uses)
- Git

## 1) Crea carpeta y repo

```
mkdir chess-ai  
cd chess-ai  
git init
```

## 2) Inicializa package.json

```
npm init -y
```

## 3) Instala TypeScript y tooling mínimo

```
npm i -D typescript tsup
```

- **TypeScript**: compila TS → JS.
- **tsup**: empaquetador/build simple (genera `dist/`, soporta ESM/CJS fácil).

## 4) Crea tsconfig (para librería)

```
npx tsc --init
```

Luego ajusta lo importante (lo típico):

- `declaration: true` (para `.d.ts`)
- `sourceMap: true`
- `outDir: "dist"`

- `moduleResolution: "Bundler"` (o `NodeNext` si prefieres)
- `target: "ES2020"` (o ES2022)

*(No te pego un tsconfig gigante aquí; si quieres, te dejo uno “golden” listo.)*

## 5) Estructura base

Crea:

```
src/
  index.ts
```

En `src/index.ts` exporta tu clase:

```
export class ChessEngine {
  constructor() {}
}
```

## 6) Configura build + exports (ESM-only recomendado)

Edita `package.json` y deja algo así (lo esencial):

- `"type": "module"` → ESM
- `exports` → entrada oficial
- `types` → tipos

Ejemplo mínimo:

```
{
  "name": "chess-ai",
  "version": "0.1.0",
  "type": "module",
  "main": "./dist/index.js",
  "types": "./dist/index.d.ts",
  "exports": {
    ".": {
      "types": "./dist/index.d.ts",
      "default": "./dist/index.js"
    }
  }
}
```

```
    },
  },
  "files": ["dist"],
  "scripts": {
    "build": "tsup src/index.ts --format esm --dts",
    "dev": "tsup src/index.ts --format esm --dts --watch",
    "prepack": "npm run build"
  },
  "devDependencies": {
    "tsup": "^x",
    "typescript": "^x"
  }
}
```

✓ Con esto:

- cuando alguien instala tu paquete, solo se publica `dist`
- `npm publish` fuerza build antes (por `prepack`)
- Typescript users reciben tipos automáticamente

## 7) Compila por primera vez

```
npm run build
```

Deberías ver:

```
dist/index.js  
dist/index.d.ts
```

## 8) Prueba el paquete localmente (sin publicar)

Dos opciones:

Opción A: `npm pack` (la forma más real)

```
npm pack
```

Eso crea un `.tgz`. Luego en otro proyecto:

```
npm i ./ruta/chess-ai-0.1.0.tgz
```

## Opción B: `npm link` (rápido, pero más "mágico")

```
npm link  
# en tu app:  
npm link chess-ai
```

Yo prefiero `npm pack` para evitar sorpresas.

## 9) Añade README y licencia

- `README.md` con ejemplo mínimo de uso.
- `LICENSE` (MIT suele ser lo estándar).

## 10) Publicación

Cuando estés listo:

```
npm login  
npm publish --access public
```

Si el nombre está pillado, tendrás que cambiar "name" (por ejemplo @tu-scope/chess-ai).

## Mi recomendación para tu caso (IA ajedrez)

Empieza con esto:

1. **ESM-only** (lo que te puse).
2. API mínima: `ChessEngine`, `Position`, `Move`, `search(options)`.
3. Luego ya expandes a submódulos (`/fen`, `/pgn`) cuando el core sea estable.