

# REFERENCIA RÁPIDA DE NPM

Gestión de paquetes, scripts, versionado, publicación, workspaces

## Instalación

### Instalar npm y Node

```
node -v && npm -v # check installed versions
npm install -g npm@latest # update npm itself
npm install --lts # install Node LTS via nvm
npm use 20 # switch to Node 20
```

### Comandos de instalación

**npm install** Instalar todas las dependencias de package.json

**npm install pkg** Agregar paquete como dependencia

**npm install -D pkg** Agregar paquete como devDependency

**npm install -g pkg** Instalar paquete globalmente

**npm install pkg@2.1.0** Instalar versión específica

**npm ci** Instalación limpia desde lock file (CI/CD)

**npm uninstall pkg** Eliminar paquete

## Gestión de paquetes

### Gestionar paquetes

```
npm ls # list installed packages
npm ls --depth=0 # top-level only
npm outdated # check for newer versions
npm update # update within server range
npm audit # check for vulnerabilities
```

### Comandos de gestión

**npm ls** Listar paquetes instalados en árbol

**npm outdated** Mostrar paquetes con versiones más nuevas

**npm update [pkg]** Actualizar paquetes dentro del rango semver

**npm audit** Auditar dependencias en busca de vulnerabilidades

**npm audit fix** Corregir dependencias vulnerables automáticamente

**npm prune** Eliminar paquetes extraños

**npm dedupe** Aplanar árbol de dependencias para reducir duplicación

## Scripts

### Ejecutar scripts

```
npm run build # run "build" script
npm test # shortcut for "test" script
npm start # shortcut for "start" script
npm run lint -- --fix # pass args to script
npm run dev & # run in background
```

### Ciclo de vida de scripts

**npm test / npm t** Ejecutar `scripts.test`

**npm start** Ejecutar `scripts.start`

**npm run <name>** Ejecutar cualquier script personalizado

**pre<name>** Se ejecuta automáticamente antes de <name>

**post<name>** Se ejecuta automáticamente después de <name>

**npm run** Listar todos los scripts disponibles

## package.json

### Inicializar y campos

```
npm init # interactive setup
npm init -y # accept all defaults
npm pkg set name="my-app" # set a field
npm pkg get version # read a field
```

### Campos clave

**name** Nombre del paquete (minúsculas, sin espacios)

**version** Versión actual (semver: mayor.minor.patch)

**main** Punto de entrada para CommonJS ( `require` )

**module** Punto de entrada para módulos ES (bundlers)

**type** `module` para ESM, `commonjs` para CJS (por defecto)

**scripts** Comandos con nombre ( `build` , `test` , `start` , etc.)

**dependencies** Dependencias de producción

**devDependencies** Dependencias solo para desarrollo

**engines** Rangos de versión requeridos de Node/npm

## Versionado

### Comandos de versión

```
npm version patch # 1.0.0 -> 1.0.1
npm version minor # 1.0.1 -> 1.1.0
npm version major # 1.1.0 -> 2.0.0
npm version 3.2.1 # set explicit version
npm version prerelease --preid=beta # 1.0.0-beta.0
```

### Rangos semver

**<1.2.3** Compatible: >=1.2.3 <2.0.0 (por defecto)

**~1.2.3** Nivel de parche: >=1.2.3 <1.3.0

**1.2.3** Solo versión exacta

**>=1.0.0 <2.0.0** Rango explícito

**\*** Cualquier versión

**1.x / 1.2.x** Rangos con comodín

**latest** Etiqueta de la versión más reciente publicada

## Publicación

### Flujo de publicación

```
npm login # authenticate to registry
npm publish # publish public package
npm publish --access public # scoped package as public
npm unpublish pkg@1.0.0 # remove specific version
npm deprecate pkg@<2* "Use v2+" # deprecate old versions
```

### Referencia de publicación

**npm login** Autenticarse con el registro npm

**npm publish** Publicar paquete en el registro

**npm pack** Crear tarball sin publicar

**npm unpublish** Eliminar versión publicada (dentro de 72h)

**npm deprecate** Marcar versiones como obsoletas

**.npmignore** Archivos a excluir del paquete publicado

**files (package.json)** Lista blanca de archivos a incluir en el paquete

## Workspaces

### Comandos de workspace

```
npm init -w packages/core # create workspace
npm install -w packages/core lodash # install in workspace
npm run build --workspaces # run in all workspaces
npm run test -w packages/api # run in specific workspace
npm ls --workspaces # list workspace deps
```

### Configuración de workspace

**workspaces (package.json)** Array de globs de workspace: [ `packages/\*` ]

**-w / --workspace** Apuntar a un workspace específico

**--workspaces** Ejecutar comando en todos los workspaces

**--include-workspace-root** Incluir el paquete raíz en operaciones de workspace

**npm install (root)** Instala todas las dependencias de workspace

**Hoisting** Deps compartidas elevadas al node\_modules raíz

## npm

### Ejecutar con npx

```
npx create-react-app my-app # run without installing
npx tsc --init # run local or remote bin
npx -p typescript tsc file.ts # specify package explicitly
npx --yes create-next-app # skip install prompt
npx node@18 -e "console.log('hi!')" # run with specific Node
```

### Opciones de npx

**npx cmd** Ejecutar cmd desde node\_modules/.bin local o remoto

**npx -p pkg cmd** Instalar pkg, luego ejecutar cmd

**npx --yes cmd** Confirmar automáticamente la instalación

**npx --no cmd** Rechazar instalación — falla si no local

**npx -c 'cmd'** Ejecutar comando shell con PATH de npx

**npx node@ver** Ejecutar versión específica de Node.js

## Configuración

### Comandos de configuración

```
npm config list # show current config
npm config set registry https://r.npmjs.com/
npm config set init-author-name "Name"
npm config get prefix # global install path
npm config delete key # remove a config value
```

### Referencia de configuración

**.npmrc (project)** Archivo de configuración por proyecto

**~/.npmrc** Archivo de configuración por usuario

**registry** URL del registro de paquetes

**save-exact** `true` para fijar versiones exactas al instalar

**engine-strict** `true` para forzar el campo `engines`

**fund** `false` para suprimir mensajes de financiamiento

**audit** `false` para omitir auditoría al instalar

## Patrones comunes

### Comandos en una línea

```
npm ls --depth=0 --json | jq '.dependencies | keys[]'
npm outdated --long # show type and homepage
npm cache clean --force # clear npm cache
npm explain pkg # why is pkg installed?
npm exec -- envinfo --system # system info for bug reports
```

## Recetas

**Lock file only** `npm ci` — instalación limpia desde package-lock.json

**Check licenses** `npx license-checker --summary`

**Find unused deps** `npx depcheck`

**Bundle size** `npx bundlephobia-cli pkg` — verificar tamaño del paquete

**Upgrade all** `npx npm-check-updates -u && npm install`

**Local registry** `npx verdaccio` — ejecutar registro privado