



Institut Sabadell

Distribució de paquets NPM, NVM i NG

Marc Sánchez López
23/05/2024
Programació web

Índex

Què són els paquets NPM, NVM i NG?.....	1
NPM.....	1
NVM.....	1
NG.....	1
Què fer si es vol tenir més d'una versió de NPM?.....	1
Comandes.....	2
NPM.....	2
NVM.....	3
Quines tecnologies utilitzen NPM?.....	4
Quins avantatges hi ha al utilitzar NPM?.....	4
Diferències entre dependències de producció i desenvolupament.....	4
Què és el fitxer package.json?.....	4
Distribució amb Angular.....	5
Instal·lar i desinstal·lar paquets.....	5
Mòduls globals i locals de NPM.....	5
Què és un script NPM?.....	6
Quins són els paquets més importants per a desenvolupament web?.....	6
Alternatives a NPM.....	6
Crea un paquet a npm.....	7
Crea un compte a npm.....	7
Crea un directori buit on posar el paquet.....	7
Crea el fitxer a pujar.....	7
Executa la comanda npm init.....	8
Executa ara npm login.....	8
I ara executa la comanda npm publish.....	9
I ara prova que funciona.....	10

Què són els paquets NPM, NVM i NG?

NPM

NPM és un gestor de paquets nadiu de NodeJS; un conjunt d'eines que ens permeten automatitzar els processos d'actualització, instal·lació, configuració i eliminació de paquets de programari. Permet l'empaquetatge de mòduls i administrar les dependències de les aplicacions i llibreries de codi.

També ens serveix com a gestor de paquets de Javascript.

NVM

NVM és el gestor de versions de NodeJS. Ens permet instal·lar i gestionar diverses versions dins d'un mateix sistema i la creació d'entorns amb versions de NodeJS específiques per a cada projecte.

NG

NG és el servei propi d'Angular que ens permet tota mena de coses com per exemple: executar el projecte, crear diversos components i serveis, actualitzar el propi Angular, etc.

Què fer si es vol tenir més d'una versió de NPM?

Primer de tot, hem d'instal·lar l'NVM en el nostre sistema (si no el tenim ja) des d'aquest repositori: <https://github.com/coreybutler/nvm-windows/releases/tag/1.1.12>

Una vegada tinguem nvm instal·lat, podem llistar les versions actuals que tenim en el nostre ordinador.

```
C:\Users\IES SABADELL>nvm list  
  
* 20.13.0 (Currently using 64-bit executable)
```

Ara podem instal·lar una nova versió de NodeJS amb la comanda nvm install «versió».

```
C:\Users\IES SABADELL>nvm install 18.20.2  
Downloading node.js version 18.20.2 (64-bit)...  
Extracting node and npm...  
Complete  
npm v10.5.0 installed successfully.  
  
Installation complete. If you want to use this version, type  
nvm use 18.20.2
```

Tornem a llistar les versions per comprovar que s'hagi instal·lat correctament i el gestor de versions la detecti.

```
C:\Users\IES SABADELL>nvm list

* 20.13.0 (Currently using 64-bit executable)
  18.20.2
```

Per a posar la versió instal·lada com la predefinida podem posar nvm use «versió» com ens indica anteriorment el mateix nvm.

```
C:\Users\IES SABADELL>nvm use 18.20.2
Now using node v18.20.2 (64-bit)

C:\Users\IES SABADELL>nvm list

  20.13.0
* 18.20.2 (Currently using 64-bit executable)
```

I ara per eliminar la versió instal·lada podem escriure npm uninstall «versió»

```
C:\Users\IES SABADELL>nvm uninstall 18.20.2
Uninstalling node v18.20.2... done
C:\Users\IES SABADELL>nvm list

  20.13.0
```

I només ens quedaria posar la versió anterior com la que volem usar.

```
C:\Users\IES SABADELL>nvm use 20.13.0
Now using node v20.13.0 (64-bit)

C:\Users\IES SABADELL>nvm list

* 20.13.0 (Currently using 64-bit executable)
```

Comandes

NPM

Per veure les comandes que podem executar amb npm podem executar la comanda npm -l i se'ns llistarà totes les comandes possibles amb la seva explicació de la seva funcionalitat i diversos usos que poden tenir.

```
C:\Users\IES SABADELL>npm -l
npm <command>

Usage:

npm install          install all the dependencies in your project
npm install <foo>    add the <foo> dependency to your project
npm test             run this project's tests
npm run <foo>        run the script named <foo>
npm <command> -h     quick help on <command>
npm -l              display usage info for all commands
npm help <term>      search for help on <term> (in a browser)
npm help npm         more involved overview (in a browser)

All commands:

  access              Set access level on published packages

                        Usage:
                        npm access list packages [<user>|<scope>|<scope:team>] [<package>]
                        npm access list collaborators [<package>] [<user>]
```

Les comandes bàsiques de npm serien:

- npm install (Ens serveix tant per instal·lar els paquets bàsics com d'específics posant «nom del paquet» al davant)
- npm update (Actualitza tots els paquets a les últimes versions disponibles)
- npm -h «comanda» (Ens mostra l'ajuda en línia d'ordres respecte a la comanda que li hem especificat) → També podem posar npm help «comanda» per a que ens obri en el navegador l'ajuda de la pàgina web amb més detalls.
- npm version (Ens mostra les versions dels paquets que tenim instal·lats)

NVM

Per veure les comandes de nvm només ens cal escriure la comanda nvm i se'ns llisten les possibles comandes que podem utilitzar amb aquest gestor de versions i una petita descripció de que fa cadascuna.

```
C:\Users\IES_SABADELL>nvm
Running version 1.1.12.
Usage:
  nvm arch           : Show if node is running in 32 or 64 bit mode.
  nvm current        : Display active version.
  nvm debug          : Check the NVM4W process for known problems (troubleshooter).
  nvm install <version> [arch] : The version can be a specific version, "latest" for the latest current version, or "lts" for the
                                most recent LTS version. Optionally specify whether to install the 32 or 64 bit version (defaults
                                to system arch). Set [arch] to "all" to install 32 AND 64 bit versions.
                                Add --insecure to the end of this command to bypass SSL validation of the remote download server.
  nvm list [available] : List the node.js installations. Type "available" at the end to see what can be installed. Aliased as ls.
  nvm on             : Enable node.js version management.
  nvm off            : Disable node.js version management.
  nvm proxy [url]    : Set a proxy to use for downloads. Leave [url] blank to see the current proxy.
                        Set [url] to "none" to remove the proxy.
  nvm node_mirror [url] : Set the node mirror. Defaults to https://nodejs.org/dist/. Leave [url] blank to use default url.
  nvm npm_mirror [url] : Set the npm mirror. Defaults to https://github.com/npm/cli/archive/. Leave [url] blank to default url.
  nvm uninstall <version> : The version must be a specific version.
  nvm use [version] [arch] : Switch to use the specified version. Optionally use "latest", "lts", or "newest".
                            "newest" is the latest installed version. Optionally specify 32/64bit architecture.
                            nvm use <arch> will continue using the selected version, but switch to 32/64 bit mode.
  nvm root [path]     : Set the directory where nvm should store different versions of node.js.
                        If <path> is not set, the current root will be displayed.
  nvm [--]version      : Displays the current running version of nvm for Windows. Aliased as v.
```

Les comandes bàsiques de nvm poden ser:

- nvm current (Ens mostra la versió que s'està utilitzant actualment de NodeJS)
- nvm install «versió» (Podem instal·lar la versió que vulguem)
- nvm list (Mostrem les versions instal·lades dins el sistema i quina s'està utilitzant)

NG

Per veure totes les comandes de ng podem anar a la pàgina web d'Angular on se'ns llisten amb les seves descripcions, exemples, etc. → <https://angular.dev/cli>

Les comandes bàsiques de ng poden ser:

- ng serve (Compila i executa el teu projecte per a mostrar-lo pel navegador. Si hi posem un -o al final se'ns obre directament)
- ng generate (Amb aquesta comanda podem crear tot tipus d'arxius per al nostre projecte)

- ng new «nom» (Creem un nou projecte d'Angular amb el nom especificat)
- ng update (Actualitzem la versió del nostre CLI a la versió que hi especifiquem)

Quines tecnologies utilitzen NPM?

NPM s'utilitza en tot tipus d'entorns, com projectes purament de Javascript, de Typescript o entorns de desenvolupament que utilitzin els dos llenguatges esmenats anteriorment com vue o Angular.

Quins avantatges hi ha al utilitzar NPM?

NPM és dels gestors de paquets més utilitzats del planeta, amb el repositori de paquets més gran de qualsevol tipus de llenguatge de programació (amb més de 2.1 milions de paquets llestos per instal·lar).

A part de tenir a l'abast tot tipus de paquets, amb npm és molt senzill manejar tot tipus de dependències entre ells evitant possibles conflictes. També ens permet manejar versions específiques de paquets que vulguem sense complicacions. Està integrat en diverses eines de desenvolupament facilitant encara més el seu ús.

Diferències entre dependències de producció i desenvolupament

Les dependències de producció són aquelles que són necessàries tant en l'execució de l'aplicació com en la seva creació i manteniment mentre que les dependències de desenvolupament són aquelles que només són necessàries per al programador mentre realitza el manteniment o creació d'aquesta.

Què és el fitxer package.json?

El fitxer package.json conté totes les dades importants d'un projecte que es realitzi utilitzant npm, ja que conté el nom, versió, descripció i les dependències d'aquest (tant les dependències de producció com les de desenvolupament, que s'anomenen devDependencies).

Distribució amb Angular

La comanda per generar una distribució seria ng build, la qual compila el projecte. Al compilar-lo, ha creat una nova carpeta anomenada dist, on hi ha diversos arxius i la imatge que he utilitzat en el projecte.



Instal·lar i desinstal·lar paquets

Per a instal·lar un paquet hem d'utilitzar la comanda npm install «nom del paquet» i per a desinstal·lar-ne un hem d'utilitzar l'altre comanda npm uninstall «nom del paquet». Quan es desinstal·la un paquet també s'eliminen totes les dependències que s'havien instal·lar amb aquest.

Mòduls globals i locals de NPM

Els mòduls locals són aquells mòduls que s'instal·len per al projecte actual, mentre que els globals són els paquets que s'instal·len en una carpeta específica del teu dispositiu per a tots els projectes que l'utilitzin, pel que pot dur a problemes i errors si diversos projectes hi ha conflictes entre els paquets globals i els locals.

Què és un script NPM?

Un script de npm és una comanda que podem executar des del mateix projecte on hi és (estan dins del package.json) que ens permet automatitzar una execució amb una única comanda. Lés més comunes i que venen per defecte en un projecte d'Angular són: start (ng start), build (ng build), watch (ng build --watch --configuration development) i test (ng test).

Quins són els paquets més importants per a desenvolupament web?

Els paquets més importants i més utilitzats de npm són:

- Express (<https://www.npmjs.com/package/express>)
- Socket.io (<https://www.npmjs.com/package/socket.io>)
- GraphQL (<https://www.npmjs.com/package/graphql>)
- Helmet (<https://www.npmjs.com/package/helmet>)
- Node-sass (<https://www.npmjs.com/package/node-sass>)
- Lodash (<https://www.npmjs.com/package/lodash>)
- Moment (<https://www.npmjs.com/package/moment>)

(Extret de: <https://www.bacancytechnology.com/blog/npm-packages>)

Alternatives a NPM

Dues alternatives a npm per a gestionar paquets a js serien: yarn i pnpm.

La primera alternativa, yarn, intenta solucionar la poca velocitat de descàrrega de npm i oferir execució sense xarxa de manera nativa.

Per a evitar velocitats de descàrrega excessivament lents, el que fa yarn es que evita descarregar paquets amb dependències que ja existeixen, és a dir, si un paquet necessita d'altres per a funcionar bé i part d'aquests ja estan descarregats no els descarrega ja que ja els té i pot fer que s'utilitzin entre sí.

La segona és pnpm, que és més ràpida que yarn evitant fer descàrregues de paquets sencers quan ja es té una versió antiga, només descarrega les actualitzacions dels paquets mateixos, el que estalvia temps i espai de disc.

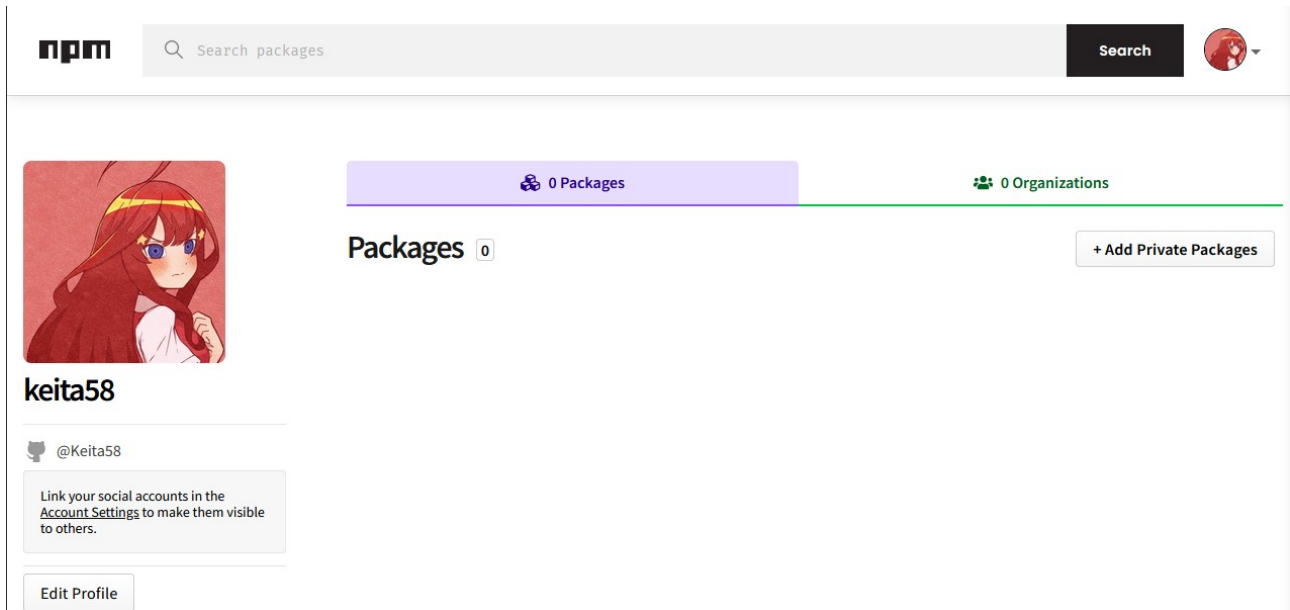
Un altre gestor de paquets (no de js) és el gestor de paquets d'Ubuntu, l'apt. Té les eines bàsiques; actualitzar paquets, descarregar-ne de nous i fins i tot actualitzar l'Ubuntu mateix. →

<https://ubuntu.com/server/docs/package-management>

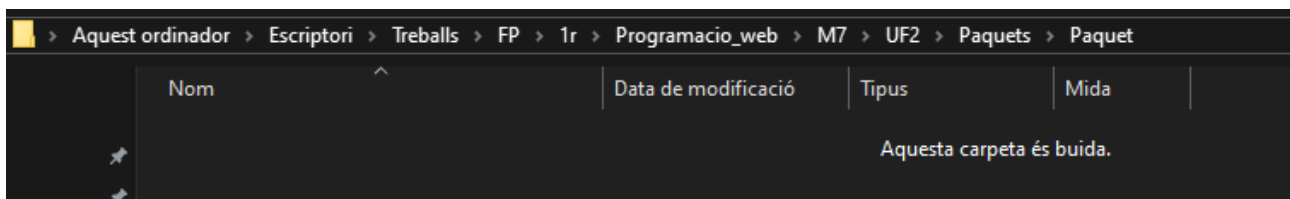
winget és el gestor de paquets nadiu de Windows 10 i 11. Realitza tasques molt semblants a l'apt d'Ubuntu, amb la diferència que winget només s'utilitza per a paquets, no per actualitzar el sistema.

Crea un paquet a npm

Crea un compte a npm



Crea un directori buit on posar el paquet



Crea el fitxer a pujar



Executa la comanda npm init

```
PS C:\Users\marc-\Desktop\Treballs\FP\1r\Programacio_web\M7\UF2\Paquets\Paquet> npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (paquet) suma
version: (1.0.0)
description: Suma bàsica per provar la creació de paquets a npm
entry point: (index.js)
test command:
git repository:
keywords: Tutorial NPM
author: Keita58
license: (ISC)
About to write to C:\Users\marc-\Desktop\Treballs\FP\1r\Programacio_web\M7\UF2\Paquets\Paquet\package.json:

{
  "version": "1.0.0",
  "description": "Suma bàsica per provar la creació de paquets a npm",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [
    "Tutorial",
    "NPM"
  ],
  "author": "Keita58",
  "license": "ISC"
}

Is this OK? (yes)

PS C:\Users\marc-\Desktop\Treballs\FP\1r\Programacio_web\M7\UF2\Paquets\Paquet> npm login
```

Executa ara npm login

```
PS C:\Users\marc-\Desktop\Treballs\FP\1r\Programacio_web\M7\UF2\Paquets\Paquet> npm login
npm notice Log in on https://registry.npmjs.org/
Login at:
https://www.npmjs.com/login?next=/login/cli/f2eeabd3-9089-43f3-a21e-dcfcb53de61e
Press ENTER to open in the browser...

Logged in on https://registry.npmjs.org/.
PS C:\Users\marc-\Desktop\Treballs\FP\1r\Programacio_web\M7\UF2\Paquets\Paquet> 
```

Per a executar la comanda npm publish

```
PS C:\Users\marc-\Desktop\Treballs\FP\1r\Programacio_web\M7\UF2\Paquets\Paquet> npm publish
npm notice
npm notice 📦 suma-proves@1.0.0
npm notice Tarball Contents
npm notice 104B index.js
npm notice 313B package.json
npm notice Tarball Details
npm notice name: suma-proves
npm notice version: 1.0.0
npm notice filename: suma-proves-1.0.0.tgz
npm notice package size: 418 B
npm notice unpacked size: 417 B
npm notice shasum: ebd960b1016d39700430d111d0ebe087a634ef86
npm notice integrity: sha512-ieNb49bj3scCA[...]9h9cwHbd6TSEw==
npm notice total files: 2
npm notice
npm notice Publishing to https://registry.npmjs.org/ with tag latest and default access
+ suma-proves@1.0.0
PS C:\Users\marc-\Desktop\Treballs\FP\1r\Programacio_web\M7\UF2\Paquets\Paquet>
```

He hagut de canviar el nom del paquet perquè hi havia un altre amb el nom de «suma» i no em deixava pujar-lo.

suma-proves

1.0.0 • Public • Published a few seconds ago

[Readme](#)[Code](#)[Beta](#)[0 Dependencies](#)[0 Dependents](#)[1 Versions](#)[Settings](#)

This package does not have a README. [Add a README](#) to your package so that users know how to get started.

Keywords

[Tutorial](#) [NPM](#)

Install

```
> npm i suma-proves
```

Version

1.0.0

License

ISC

Unpacked Size

417 B

Total Files

2

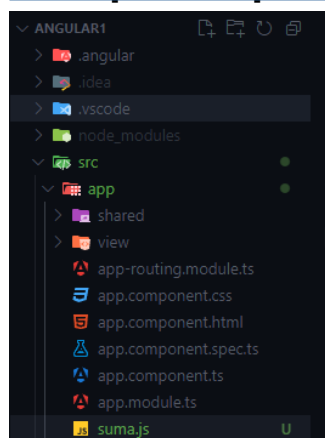
Last publish

a few seconds ago

Collaborators

[Try on RunKit](#)[Report malware](#)

I ara prova que funciona



He creat l'arxiu en el projecte d'Angular 1.

```
PS C:\Users\marc\Desktop\Treballs\FP\1r\Programacio_web\M7\UF1\Angular\Exercicis\Angular1> npm install suma-proves
added 1 package, and audited 929 packages in 3s
120 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
PS C:\Users\marc\Desktop\Treballs\FP\1r\Programacio_web\M7\UF1\Angular\Exercicis\Angular1>
```

He importat el paquet creat amb npm install suma-proves

```
package.json > ...
1 {
2   "name": "angular1",
3   "version": "0.0.0",
4   "scripts": {
5     "ng": "ng",
6     "start": "ng serve",
7     "build": "ng build",
8     "watch": "ng build --watch --configuration development",
9     "test": "ng test"
10  },
11  "private": true,
12  "dependencies": {
13    "@angular/animations": "^17.3.0",
14    "@angular/common": "^17.3.0",
15    "@angular/compiler": "^17.3.0",
16    "@angular/core": "^17.3.0",
17    "@angular/forms": "^17.3.0",
18    "@angular/platform-browser": "^17.3.0",
19    "@angular/platform-browser-dynamic": "^17.3.0",
20    "@angular/router": "^17.3.0",
21    "@ng-bootstrap/ng-bootstrap": "^16.0.0",
22    "@popperjs/core": "^2.11.8",
23    "bootstrap": "^5.3.2",
24    "rxjs": "~7.8.0",
25    "suma-proves": "^1.0.0",
26    "tslib": "^2.3.0",
27    "zone.js": "~0.14.3"
28  },
```

Es pot comprovar al package.json del projecte que s'ha importat correctament.

```
src > app > suma.js > ...
1 const suma = require('suma-proves');
2
3 console.log(suma(1,2))
```

En l'arxiu creat «suma.js» he importat el paquet creat i he fet un console.log per a comprovar que funciona correctament.

```
PS C:\Users\marc\Desktop\Treballs\FP\1r\Programacio_web\M7\UF1\Angular\Exercicis\Angular1> node .\src\app\suma.js
La teva suma és: 3
PS C:\Users\marc\Desktop\Treballs\FP\1r\Programacio_web\M7\UF1\Angular\Exercicis\Angular1>
```

I finalment podem comprovar que el que ens retorna el paquet és l'esperat i el que hem posat al codi original.