

# NPM

## Comandos basicos

- \$ npm init
  - \$ npm init -y
  - \$ npm set init.author.email <email>
  - \$ npm set init.author.name <name>
- Sirve para crear el archivo package.json el cual contiene toda la información acerca de nuestro proyecto
- Asignar nuestros datos como en 'git', pero no es tan obligatorio hacer esto.

## Instalación de dependencias

Las dependencias se deben instalar en nuestra carpeta raíz de nuestro proyecto.

## \$ npm install <pkg> -g

Instalar un paquete de forma global. Esto permite que podamos utilizar este paquete en diferentes proyectos. Por lo general, se deben instalar estos paquetes con permisos de administrador.

Si no queremos colocar permisos de administrador a cada rato, podemos seguir esta guía: <https://docs.npmjs.com/resolving-eacces-permissions-errors-when-installing-packages-globally>

## \$ npm list -g --depth 0

Ver los paquetes que están instalados de forma global.

## \$ npm list

Para listar los paquetes que tiene un proyecto en específico

**\$ npm install <pkg> -O**

Podemos instalar de forma opcional un paquete con este comando.

**\$ npm install <pkg> --dry-run** ( simula la instalación )

Este flag indica que el paquete no va a ser instalado dentro del proyecto, simplemente es una simulación, nada mas nos muestra el output como si se fuese instalado

**\$ npm install <pkg> -f | npm install <pkg> --force**

Instalar algún paquete de forma forzada. Nos va a permitir instalar este paquete forzando esa instalación a que sea desde el ultimo recursos desde el servidor de NPM

**\$ npm install <pkg>@<versión>**

Para instalar algún paquete con una versión específica

## Actualizar y eliminar

**\$ npm outdated**

Para **ver** si se encuentran desactualizados los paquetes y, además nos indica la últimas versiones de esos paquetes

**\$ npm update**

Podemos actualizar los paquetes que se encuentran desactualizados.

**\$ npm install <pkg>@latest**

Podemos actualizar un paquete en específico a su última versión con este comando.

**\$ npm uninstall <pkg>**

Para desinstalar o eliminar un paquete en específico. Esto lo elimina del package.json y node\_modules

**\$ npm uninstall <pkg> --no-save**

Nos permite desinstalar o eliminar un paquete, pero sin eliminarlo del package.json pero si del node\_modules