

# JAVASCRIPT FRONTEND

## AVANZADO

### CLASE 12 : NUEVAS TECNOLOGÍAS

## NPM

npm abre un mundo de talentos de JavaScript para usted y su equipo. Es el registro de software más grande del mundo, con aproximadamente 3 mil millones de descargas por semana. El registro contiene más de 600,000 paquetes (bloques de construcción de código). Los desarrolladores de código abierto de todos los continentes usan npm para compartir y tomar prestados paquetes, y muchas organizaciones usan npm para administrar el desarrollo privado también. <sup>1</sup>

npm consta de tres componentes distintos:

1. el sitio web
2. la interfaz de línea de comando (CLI)
3. el registro

Use **el sitio web** para descubrir paquetes, configurar perfiles y administrar otros aspectos de su experiencia npm. Por ejemplo, puede configurar Orgs (organizaciones) para gestionar el acceso a paquetes públicos o privados.

El **CLI** se ejecuta desde una terminal. Así es como la mayoría de los desarrolladores interactúan con npm.

El **registro** es una gran base de datos pública de software JavaScript y la metainformación que lo rodea.

Use npm para. . .

- Adapte paquetes de código a sus aplicaciones o incorpore paquetes tal como están.
- Descargue herramientas independientes que puede usar de inmediato.
- Ejecute paquetes sin descargar usando npx.
- Compartir código con cualquier usuario de npm, en cualquier lugar.
- Restringir código a desarrolladores específicos.
- Form Orgs (organizaciones) para coordinar el mantenimiento, la codificación y los desarrolladores de paquetes.
- Forme equipos virtuales usando Orgs.

- Administre múltiples versiones de código y dependencias de código.
- Actualice las aplicaciones fácilmente cuando se actualice el código subyacente.
- Descubre múltiples formas de resolver el mismo rompecabezas.
- Encuentre otros desarrolladores que estén trabajando en problemas y proyectos similares.

Estos son solo algunos ejemplos de formas en que los desarrolladores usan npm.

## NODE

Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El ecosistema de paquetes de Node.js, npm, es el ecosistema más grande de librerías de código abierto en el mundo.

Lo curioso de la instalación de Node es que ya viene incluida con NPM por lo tanto podemos hacer uso de este manager desde el primer momento en que empezamos a trabajar con Node.js

### Trabajando con NPM Y NODE

La mejor forma de administrar los paquetes npm instalados localmente es crear un archivo **package.json**.

Un archivo **package.json**:

- enumera los paquetes de los que depende su proyecto.
- le permite especificar las versiones de un paquete que su proyecto puede usar utilizando reglas de control de versiones semánticas.
- hace que tu compilación sea reproducible y, por lo tanto, mucho más fácil de compartir con otros desarrolladores.

### Requisitos

Un paquete.json debe tener:

- "Nombre"
  - todo en minúsculas
  - una palabra, sin espacios
  - guiones y guiones bajos permitidos

- "Versión"
  - en la forma de x.x.x
  - sigue la especificación de Semver

Por ejemplo:

```
{  
  "nombre": "my-awesome-package",  
  "versión": "1.0.0"  
}
```

## Creando un paquete.json

Hay dos formas básicas de crear un archivo **package.json**.

### 1. Ejecute un cuestionario CLI

Para crear un paquete.json ejecute:

```
> npm init
```

Esto iniciará un cuestionario de línea de comando que concluirá con la creación de un paquete.json en el directorio en el que inició el comando.

## Especificando Dependencias

Para especificar los paquetes de los que depende su proyecto, debe enumerar los paquetes que desea usar en su archivo package.json. Hay 2 tipos de paquetes que puede enumerar:

- **"dependencias"**: la aplicación requiere estos paquetes en producción.
- **"devDependencies"**: estos paquetes solo son necesarios para el desarrollo y las pruebas.

## Instalando Dependencias

Hay dos formas de instalar paquetes npm: localmente o globalmente. Elija qué tipo de instalación usar en función de cómo desea usar el paquete.

- Si desea depender del paquete de su propio módulo, utilizando algo como Node.js 'require, entonces desea instalarlo localmente. Este es el comportamiento predeterminado de la instalación de npm.
- Si desea utilizar un paquete como una herramienta de línea de comandos (como grunt CLI), instálelo de manera global.

Un paquete se puede descargar con el comando:

```
> npm install <package_name>
```

Esto creará el directorio node\_modules en su directorio actual (si aún no existe uno) y descargará el paquete a ese directorio.

Para descargar paquetes de forma global, use el comando `npm install -g <paquete>`, por ejemplo <sup>4</sup>:

```
> npm install -g <paquete>
```

## JSON-SERVER

El paquete o módulo de JSON-SERVER nos sirve para “obtener una API de REST completa y falsa con cero codificación en menos de 30 segundos (en serio)” <sup>5</sup>. Es ideal para desarrolladores front-end que necesitan un back-end rápido para comenzar a prototipar el front-end.

```
> npm install -g json-server
```

## USO

1. Crear un archivo extensión .json (db.json)

```
{
  "posts": [
    { "id": 1, "title": "json-server", "author": "typicode" }
  ],
  "comments": [
    { "id": 1, "body": "some comment", "postId": 1 }
  ]
}
```

```
],  
  "profile": { "name": "typicode" }  
}
```

2. Iniciar el servidor desde la línea de comandos

```
> json-server --watch db.json
```

3. Ir a <http://localhost:3000>

Ahora, si vas a <http://localhost:3000/posts/1>, obtendrás

```
{ "id": 1, "title": "json-server", "author": "typicode" }
```

También al hacer solicitudes, es bueno saber que:

- Si realiza solicitudes POST, PUT, PATCH o DELETE, los cambios se guardarán de forma automática y segura en db.json utilizando lowdb.
- El JSON de su cuerpo de solicitud debe estar cerrado por el objeto, al igual que la salida GET. (por ejemplo {"nombre": "Foobar"})
- Los valores de Id no son mutables. Cualquier valor de identificación en el cuerpo de su solicitud PUT o PATCH será ignorado. Solo se respetará un valor establecido en una solicitud POST, pero solo si no se ha realizado.
- Una solicitud POST, PUT o PATCH debe incluir un encabezado application / type: application / json para usar el JSON en el cuerpo de la solicitud. De lo contrario, dará como resultado un 200 OK pero sin cambios en los datos.

### Servidor de archivos estáticos

Puede usar el servidor JSON para servir su HTML, JS y CSS, simplemente cree un directorio ./public o use --static para establecer un directorio diferente de archivos estáticos.

1. <https://docs.npmjs.com/getting-started/what-is-npm>
2. <https://nodejs.org/es/>
3. <https://docs.npmjs.com/getting-started/installing-npm-packages-locally>

4. <https://docs.npmjs.com/getting-started/installing-npm-packages-globally>
5. <https://www.npmjs.com/package/json-server>