

C. NPMS

1 ¿QUÉ ES NPM?

npm es parte esencial de Node.js, el entorno de ejecución de JavaScript en el lado del servidor basado en el motor V8 de Google. Es muy seguramente la principal razón del gran éxito de Node permitiendo que cientos de desarrolladores puedan compartir paquetes de software entre distintos proyectos.

npm responde a las siglas de Node Package Manager o manejador de paquetes de node, es la herramienta por defecto de JavaScript para la tarea de compartir e instalar paquetes.

Tal como reza su documentación, npm se compone de al menos dos partes principales.

Un repositorio online para publicar paquetes de software libre para ser utilizados en proyectos Node.js

Una herramienta para la terminal (command line utility) para interactuar con dicho repositorio que te ayuda a la instalación de utilidades, manejo de dependencias y la publicación de paquetes.

Es decir, en tu proyecto basado en Node — que actualmente incluye los proyectos de aplicaciones web que utilizan Node para su proceso de compilación y generación de archivos — utilizarás la utilidad de línea de comandos (cli) para consumir paquetes desde el repositorio online, un listado gigantesco de soluciones de software para distintos problemas disponibles públicamente en npmjs.com y para manejar dependencias, y para ello necesitas un archivo de configuración que le diga a npm que este es un proyecto node.

1.1 NPM SCRIPTS

Una importante sección de este archivo es `scripts`. Esta sección define un listado de propiedades que permiten ejecutar comandos dentro del contexto de tu proyecto incluyendo: comandos de otros paquetes listados como dependencias, scripts personalizados, scripts bash, etc.

Por defecto se crea un script para ejecutar el comando de **test** que, si no agregaste nada personalizado en el proceso de inicialización sólo tendrá una llamada al comando **echo**, es decir, ****al ejecutar en la terminal `npm run test` verás en la consola `Error: no test specified`

1.2 DEPENDENCIAS Y DEPENDENCIAS DE DESARROLLO

Estas dependencias son instaladas al ejecutar ***npm install <pkg> --save*** y en el caso de las dependencias de desarrollo utilizando ***npm install <pkg> --save-dev***.

La diferencia de estos listados es que `dependencies` está destinado a ser utilizado en producción y `devDependencies` define paquetes que son necesarios sólo durante el desarrollo de tu proyecto.

Es importante conocer cómo se definen las versiones a utilizar en estas dependencias. Cada una de ellas muestra un número basado en semver en la forma **mayor.minor.patch**.

- **mayor**: Representa una versión mayor que genera cambios en la API del producto.
- **minor**: Representa un valor que aumenta cuando se hacen cambios retro-compatibles.
- **patch**: Un valor que aumenta cada vez que se hacen reparaciones de errores o mejoras sutiles.

También es posible encontrar algunos símbolos frente a la numeración de la versión, estos son:

- **^**: latest minor release. Por ejemplo `^1.0.4` indica que 1.0.4 es la versión más "baja" que se puede instalar, pero permite que se instale cualquier versión superior a esa pero que se encuentre dentro de la versión 1.
- **~**: latest patch release. Esta es la forma contraria a **^**. Esta especificación `~1.0.4` puede instalar la versión 1.0.7 si es que esta es la última versión del patch.

Una vez instalados los paquetes de tus dependencias la información de las versiones instaladas queda almacenada en un archivo llamado **package-lock.json**

1.3 PACKAGE-LOCK.JSON

Este archivo es auto generado por ***npm install*** y es una lista descriptiva y exacta de las versiones instaladas durante tu proceso. No está destinado a ser leído ni manipulado por los desarrolladores, si no, para ser un insumo del proceso de manejo de dependencias.

2 ¿CÓMO TRABAJAR CON NPM?

Lo más usual que harás con npm es la instalación de dependencias, esto se hace mediante `npm install` para instalar todas las dependencias listadas en el archivo `package.json` o utilizando `npm install <pkg>` para instalar algún paquete en particular.

2.1 NPM INSTALL

Este script nativo de npm tiene algunas otras opciones a la hora de hacer la instalación de paquetes.

Por defecto al ejecutar `npm install <pkg>` la última versión disponible en el repositorio agregando el símbolo **^** a la versión.

Al ejecutar `npm install <pkg>` el paquete se instalará dentro del directorio `node_modules` que está dentro de tu proyecto.

Puedes usar algunos parámetros tales como

- `-g` para indicar que quieres que el paquete se instale globalmente
- `--production` indica que la ejecución de `npm install` solo instalará las dependencias listadas en el apartado `dependencies` dejando de lado las dependencias de desarrollo.

2.2 NPM AUDIT

npm tiene una multitud disponibles lo que es una gran característica indicando lo saludable del ecosistema que es capaz de generar nuevas librerías a gran velocidad, pero también puede ser un problema. **npm** puede albergar paquetes maliciosos o con problema de seguridad.

La organización tras **npm** mantiene una lista de agujeros de seguridad y puedes utilizar este comando para revisar tus dependencias.

`npm audit` te entregará información de las vulnerabilidades encontradas en tus dependencias junto con una breve descripción de cómo resolverlo indicando la versión que corrige el defecto.

3 NPMS INTERESANTES

3.1 POSTHTML-INCLUDE

Introduce una nueva etiqueta en HTML llamada `<include src=""></include>`, que nos permite incluir fragmento de código HTML dentro de otros.

Por ejemplo, podríamos tener un fichero `header.html` con la cabecera de nuestra página, y otro `footer`, independientemente del número de páginas que tenga nuestro sitio web. Si queremos modificar una de ellas, tan solo debemos modificar un fichero y no todos.

Instalación: **`npm i -D posthtml-include`**

```
added 1180 packages, removed 1 package, and audited 1181 packages
106 packages are looking for funding
  run `npm fund` for details

56 vulnerabilities (44 moderate, 12 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

Debemos instalar también Parcel: ***npm i -D Parcel***

```
C:\Users\Juan\Desktop\node4>npm i -D Parcel
```

En la carpeta raíz del proyecto debemos añadir un fichero llamado **.posthtmlrc** (cuidado con él .). con el siguiente contenido:

```
{  
  "plugins": {  
    "posthtml-include": {  
      "root": "./src"  
    }  
  }  
}
```

Ahora ya podemos utilizar la etiqueta **include** dentro de nuestros ficheros **HTML**:

```

<> index.html x
src > <> index.html > html > body > main > h1
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9   <include src="./header.html"></include>
10
11   <main>
12     <h1>Esto es el main</h1>
13   </main>
14   <include src="./footer.html"></include>
15   <script src="eje.js"></script>
16 </body>
17 </html>

```

```

<> header.html x
src > <> header.html > header
1 <header>
2   <nav>
3     <h3>Esto es el header</h3>
4     <hr/>
5   </nav>
6 </header>

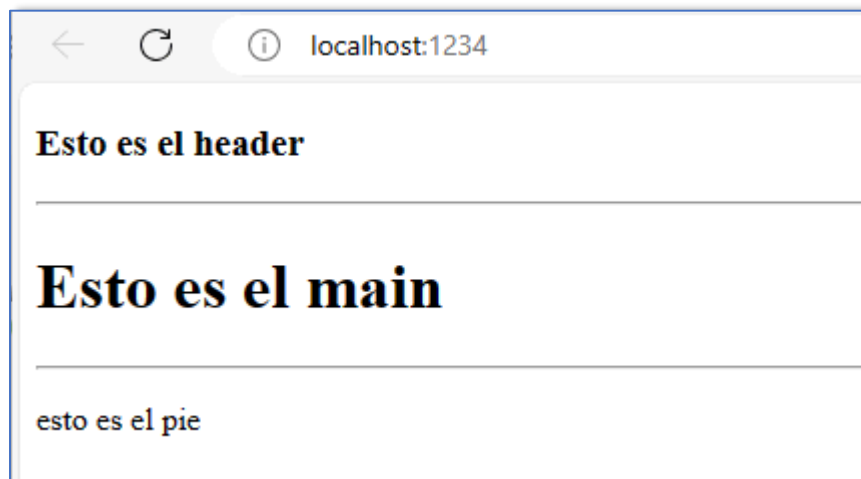
```

```

<> footer.html x
src > <> footer.html > footer
1 <footer>
2   <hr/>
3   <p>esto es el pie</p>
4 </footer>

```

El resultado es el siguiente:



3.2 FONTAWESOME

FontAwesome es un conjunto de herramientas de fuentes e íconos basado en CSS y Less. La versión 6 tiene dos paquetes: FontAwesome Free y el propietario FontAwesome Pro (disponible por una tarifa). Para instalar FontAwesome free, lo hacemos con el siguiente comando: ***npm i --save @fontawesome/fontawesome-free***

```

C:\Users\Juan\Desktop\node4>npm i --save @fortawesome/fontawesome-free
npm WARN ERESOLVE overriding peer dependency
npm WARN While resolving: htmlnano@2.1.0
npm WARN Found: svgo@2.8.0
npm WARN   node_modules/@parcel/optimizer-htmlnano/node_modules/svgo
npm WARN   svgo@"^2.4.0" from @parcel/optimizer-htmlnano@2.10.3
npm WARN   node_modules/@parcel/optimizer-htmlnano
npm WARN     @parcel/optimizer-htmlnano@"2.10.3" from @parcel/config-default@
2.10.3
npm WARN   node_modules/@parcel/config-default

```

Para poder usar FontAwesome en nuestro proyecto debemos importarlo tanto en nuestro CSS (main.css) como en el fichero JS (main.js)

main.css

```

# style.css  ×
src > # style.css
1  @import '../node_modules/@fortawesome/fontawesome-free/css/all.css';

```

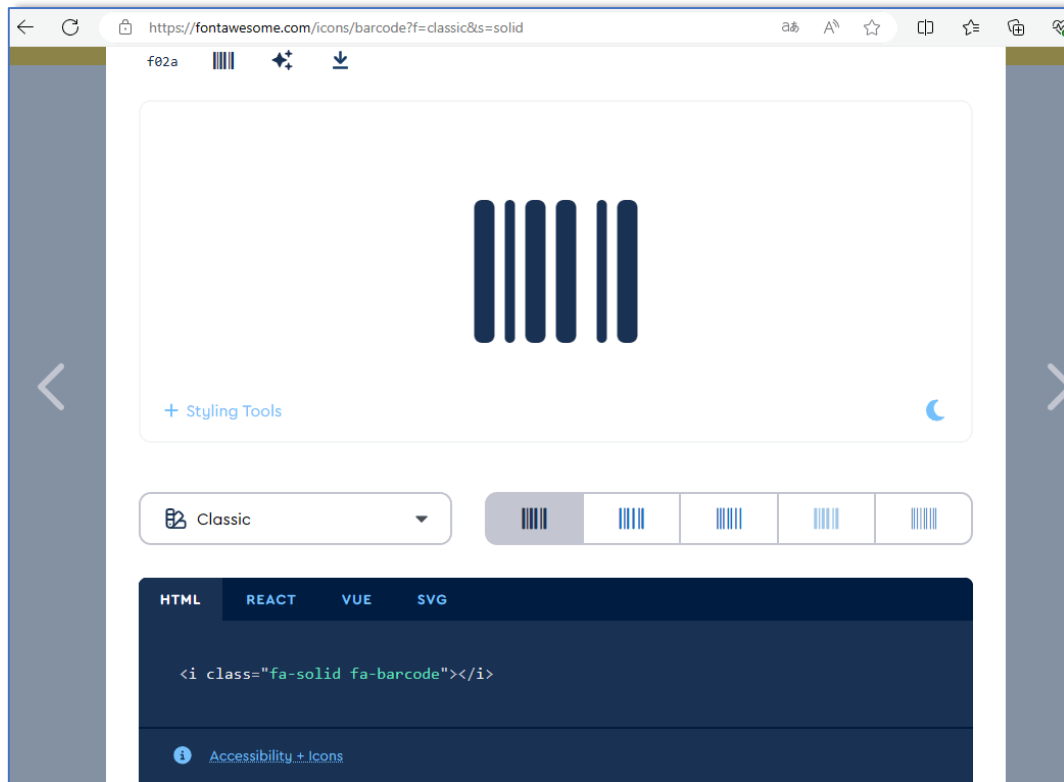
eje.js

```

JS eje.js  ×
src > JS eje.js
1  import '@fortawesome/fontawesome-free/js/all';

```

Vamos a la página de fontwasome y seleccionamos un icono (o varios):



Veamos un ejemplo de fontawesome:

index.html

```
<> index.html x
src > <> index.html > ...
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Document</title>
7      <link rel="stylesheet" href="style.css" />
8    </head>
9    <body>
10     <main>
11       <h1>Esto es el main</h1>
12       <i class="fa-solid fa-barcode"></i>
13     </main>
14
15     <script type="module" src="eje.js"></script>
16   </body>
17 </html>
18
```

Este es el resultado:



3.3 SIMPLE SLIDER

Simple-slider es una micro biblioteca de carrusel basada en la API requestAnimationFrame.

Es una implementación a prueba de test y menos dependiente de CSS.

Para poder utilizar simple slider en nuestro proyecto debemos, en primer lugar, instalarlo:

```
\uoc-boilerplate-master>npm install --save simple-slider
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#depre
recated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-res
olve#deprecated
```

Ahora debemos añadir las líneas oportunas a nuestro fichero main.js

main.js

```
/**
 * Import dependencies from node_modules
 * see commented examples below
 */

import "@fortawesome/fontawesome-free/js/all";
import "cookieconsent";

import { getSlider } from 'simple-slider';
```

```
getSlider({
  container: document.getElementById('myslider'),
  transitionTime:1,
  delay:3.5
});
```

Para que simple-slider entre en acción tendremos una zona con id='myslider' para que este paquete pueda "hacer su magia".


```
<div id="myslider" class="uoc-container3 galeria">
  
  
  
  
  
```

3.4 BOOTSTRAP

Bootstrap es un framework front-end de código abierto para el desarrollo de sitios web y aplicaciones móviles. Se basa en HTML, CSS y JavaScript.

Bootstrap proporciona un conjunto de herramientas, componentes y estilos predefinidos que permiten a los desarrolladores crear interfaces de usuario atractivas y responsive de manera rápida y sencilla.

Bootstrap también incluye una serie de plugins para funcionalidades como carruseles, ventanas modales, menús desplegables, entre otros.

Para instalar Bootstrap, vamos a instalar a este y también instalaremos Popper, ya que nuestros menús desplegables, y otros dependen de él para su posicionamiento.

```
\PEC2>npm i --save bootstrap @popperjs/core
added 2 packages, and audited 386 packages in 2s
171 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

El fichero package.json se verá modificado con este hecho.

```
...
  "sharp": "0.31.1",
  },
  "dependencies": {
    "@popperjs/core": "^2.11.7",
    "bootstrap": "^5.2.3"
  }
}
```

Ahora vamos a importar Bootstrap en nuestro fichero **main.css**

```
// @import "some-node-module";
// @import "@some-company/some-node-module";
|
@import "bootstrap/scss/bootstrap";
```

También debemos añadir al fichero **main.js**

```
//import 'some-node-module';
//import SomeModule from 'some-node-module';

import 'bootstrap';
```

Ya está todo completo, ya podemos usar Bootstrap en nuestro proyecto.

3.5 DATATABLES

DataTables es una biblioteca de JavaScript que permite agregar funciones avanzadas a las tablas HTML estándar, como paginación, ordenamiento, búsqueda, filtrado y mucho más.

Permite crear tablas interactivas y dinámicas que mejoran la experiencia de usuario al interactuar con grandes cantidades de datos.

También es altamente personalizable y se integra fácilmente con otros complementos y bibliotecas de JavaScript.

Un ejemplo de DataTables:

Las 10 primeras jugadoras de ajedrez del mundo en abril de 2023:

Mostrar registros Buscar:

Posición.	Jugadora	Federación	ELO
1	Yifan Hou	China	2628
2	Humpy Koneru	India	2576
3	Aleksandra Goryachkina	Rusia	2565
4	Kateryna Lagno	Rusia	2560
5	Ju Wenjun	China	2550
6	Lei Tingjie	China	2545
7	Alexandra Kosteniuk	Suiza	2535
8	Tan Zhongyi	China	2526
9	Nana Dzagnidze	Georgia	2525
10	Mariya Muzychuk	Ucrania	2515

Mostrando 1 a 10 de 10 registros Anterior Siguiente

Para instalar DataTables debemos usar:

```
PEC2>npm install datatables.net-dt --save-dev
```

```
PEC2> npm install datatables.net-responsive-dt --save
```

Hacemos las importaciones oportuna en los ficheros main.scss y main.js

main.css

```
/* see: https://v2.parceljs.org/features/module-resolution/
 * see commented examples below
 */

@import "bootstrap/scss/bootstrap";
@import "bootstrap-icons/font/bootstrap-icons";
@import "npm:cookieconsent/build/cookieconsent.min.css";
@import "npm:datatables.net-dt/css/jquery.dataTables.css";
```

main.js

```
import DataTable from 'datatables.net-dt';
import 'datatables.net-responsive-dt';
```

Añadiremos la secuencia de inicialización para dos tablas que vamos a utilizar en la página dentro de nuestro fichero main.js

```
let table = new DataTable('#myTable', {
  language: {
    url: 'https://cdn.datatables.net/plug-ins/1.12.1/i18n/es-ES.json'
  }
});

let table2 = new DataTable('#myTable2', {
  language: {
    url: 'https://cdn.datatables.net/plug-ins/1.12.1/i18n/es-ES.json'
  }
});
```

3.6 WEBS CON ENLACES

[Las 40 mejores bibliotecas y frameworks de JavaScript para 2023 \(kinsta.com\)](https://kinsta.com/es/javascript-frameworks/)

[Bundlephobia | Size of npm dependencies](https://bundlephobia.com/)

[npm | Home \(npmjs.com\)](https://npmjs.com/)