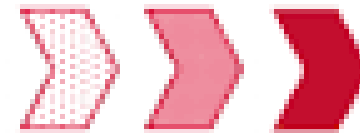


INTRODUCCIÓN A LA CREACIÓN Y PUBLICACIÓN DE LIBRERÍAS EN ANGULAR



¿ QUE ES UNA LIBRERÍA ?



- *Una librería es una **colección de componentes**, que podemos usar en cualquier otro proyecto simplemente con su instalación y que una vez instalada y publicada será totalmente funcional pudiendo aprovechar todos los recursos de esta*



¿ POR QUE USAR UNA LIBRERÍA ?

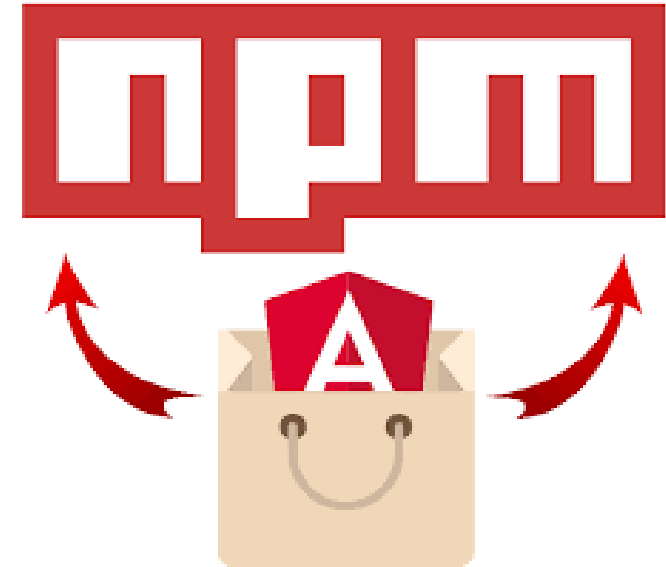


- *Para que los desarrolladores puedan realizar su trabajo de una forma mas rápida y eficiente.*
- *Para que el trabajo sea lo mas modular ,eficiente y mantenible, ya que al crear librerías con nuevas funcionalidades será posible utilizar estas nuevas funcionalidad en los desarrollos mientras estas se siguen desarrollando.*



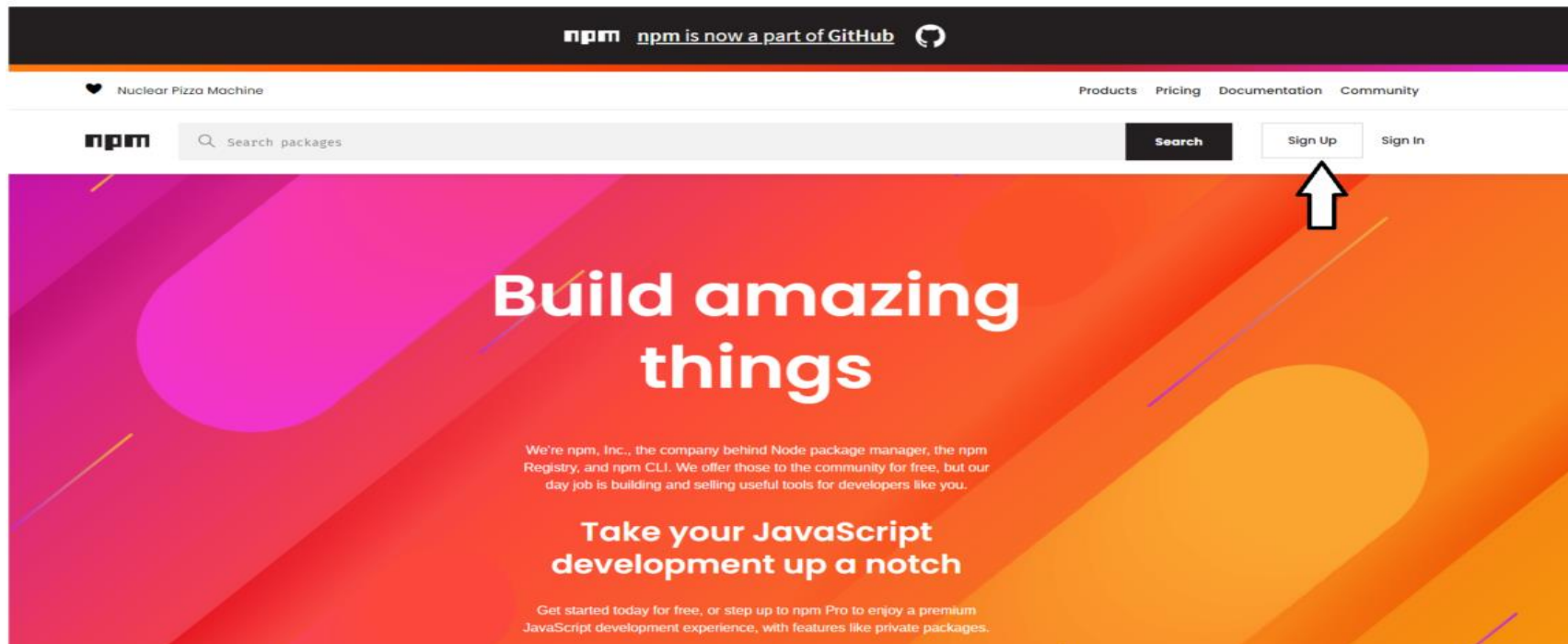
¿ QUE ES LO QUE VAMOS A HACER ?

- *Nosotros vamos a crearnos una librería en Angular , publicar esta en npm y así utilizarla desde cualquier proyecto que estemos realizando*
- *¿ Que es npm ? **npm** es el sistema de gestión de paquetes por defecto para Node.js, un entorno de ejecución para JavaScript.*



CREACIÓN DE UNA CUENTA EN NPM

- Entramos en npmjs.com y pulsamos sobre Sign Up



- Rellenamos los datos para crear la cuenta donde luego publicaremos nuestra librería

The image displays two side-by-side screenshots of the npm sign-up page, illustrating the process of creating an account.

Left Screenshot (Empty Form):

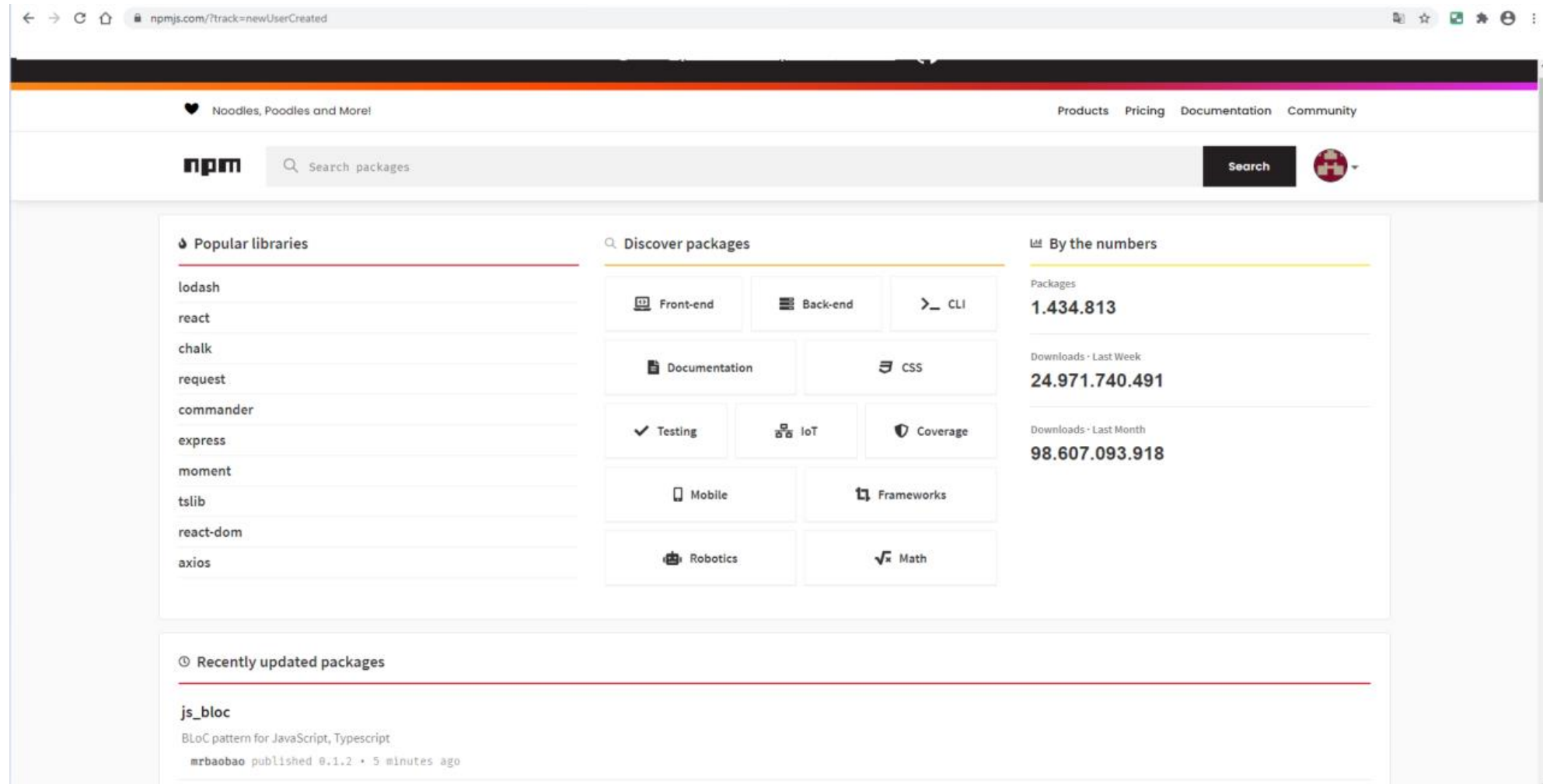
- Sign Up** header.
- Username** field.
- Email address** field.
- Password** field.
- Note:** Your email address will be added to the metadata of packages that you publish, so it may be seen publicly.
- Your password should be at least 10 characters.** [Learn more](#)
- ☐ Agree to the [End User License Agreement](#) and the [Privacy Policy](#).
- Create an Account** button.
- [or Login](#) link.

Right Screenshot (Filled Form):

- Sign Up** header.
- Username** field: `davidcgijon` (indicated by a left-pointing arrow).
- Email address** field: `davidcgijon@gmail.com` (indicated by a left-pointing arrow).
- Password** field: masked with dots (indicated by a left-pointing arrow).
- Note:** Your email address will be added to the metadata of packages that you publish, so it may be seen publicly.
- Your password should be at least 10 characters.** [Learn more](#)
- ☒ Agree to the [End User License Agreement](#) and the [Privacy Policy](#) (indicated by a right-pointing arrow).
- Create an Account** button.
- [or Login](#) link (indicated by an up-pointing arrow).



- Una Vez nos hayamos logueado , nos mandaran un mail y una vez verificado tendremos activa nuestra cuenta en npm para poder trabajar con ella



The screenshot shows the npm website interface. At the top, there's a navigation bar with links for Products, Pricing, Documentation, and Community. Below this is a search bar with the text "Search packages" and a "Search" button. The main content area is divided into three columns:

- Popular libraries:** A list of popular packages including lodash, react, chalk, request, commander, express, moment, tslib, react-dom, and axios.
- Discover packages:** A section with various category tiles such as Front-end, Back-end, CLI, Documentation, CSS, Testing, IoT, Coverage, Mobile, Frameworks, Robotics, and Math.
- By the numbers:** A section showing statistics:
 - Packages: 1.434.813
 - Downloads · Last Week: 24.971.740.491
 - Downloads · Last Month: 98.607.093.918

At the bottom, there's a section for "Recently updated packages" featuring the package "js_bloc" by "mrbaobao", which was published 0.1.2 · 5 minutes ago.



CREACIÓN DE UNA LIBRERÍA EN ANGULAR

- Lo primero que tenemos que hacer es crear un nuevo proyecto *Angular*, en este caso yo le llamaré “**my-project**”, para eso utilizamos el comando:

```
| $ ng new my-project
```

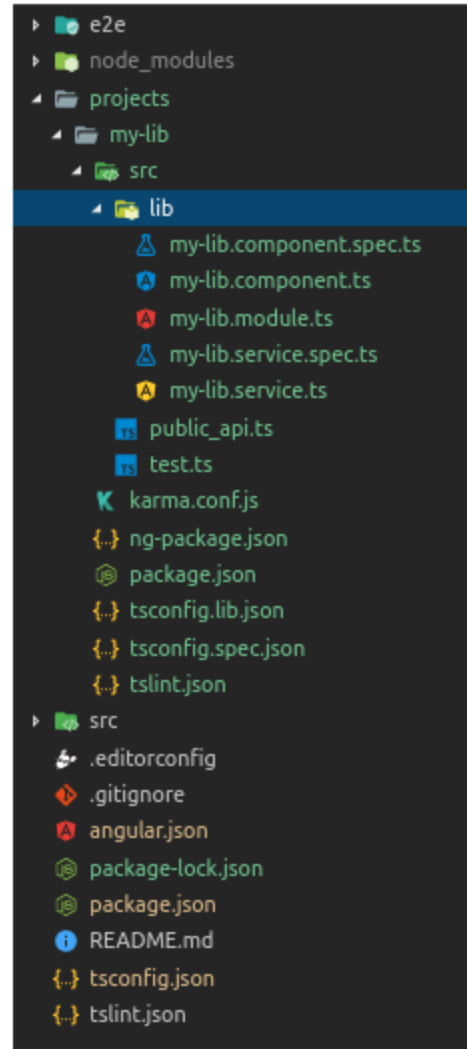
- Una vez creado el proyecto, nos posicionamos en la raíz, y en la terminal escribimos lo siguiente (el CLI de *Angular* nos proporciona un comando para crear una librería):

```
-
```

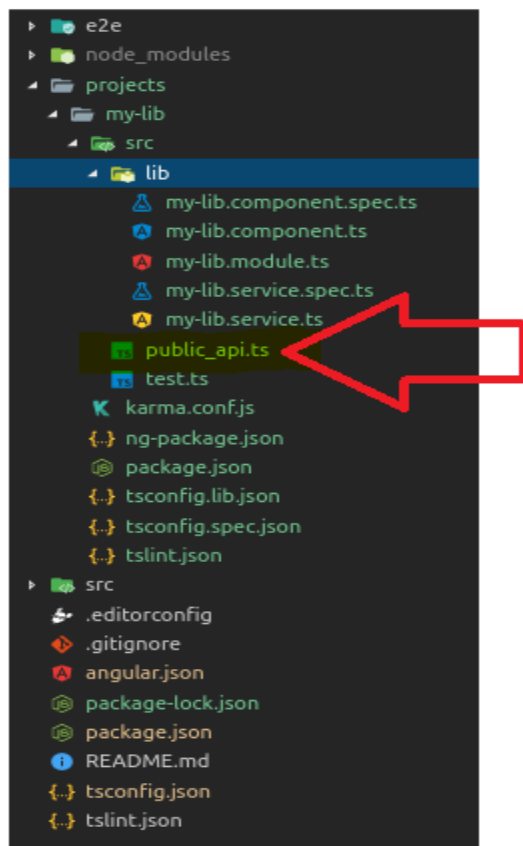
```
| $ ng generate library my-lib
```



- La estructura que tendrá nuestro proyecto debería ser la siguiente:



- Un archivo importante de nuestra librería es el llamado **public_api.ts** ahí es donde se encuentran todas las clases, componentes, directivas, etc.. que nuestra librería va a compartir. Si agregamos algún otro archivo, como por ejemplo una interfaz que se va a compartir con el usuario que ocupe nuestra librería, la agregamos a nuestro **public_api.ts**. Los archivos que son de uso interno para la librería y que no serán compartidos con el usuario, simplemente no se agregan.



```

1  /*
2   * Public API Surface of my-lib
3   */
4
5  export * from './lib/my-lib.module';
6

```

public_api.ts



- Una vez escrita nuestra librería Transpilarla ,es decir convertir nuestro código Typescript en código Javascript, para eso nos vamos a apoyar de **ng-packagr**, es un paquete que nos simplifica la creación/transpilación de librerías para *Angular*.
- Instalamos **ng-packagr**.

| `$ npm install ng-packagr — save-dev`

- En nuestro **package.json** del proyecto root **my-project** creamos un nuevo script, que nos ayudará a transpilar nuestra librería usando **ng-packagr**.

```
"scripts": {
  "ng": "ng",
  "start": "ng serve",
  "build": "ng build",
  "test": "ng test",
  "lint": "ng lint",
  "e2e": "ng e2e",
  "build:lib": "ng-packagr -p projects/my-lib/ng-package.json"
},
```

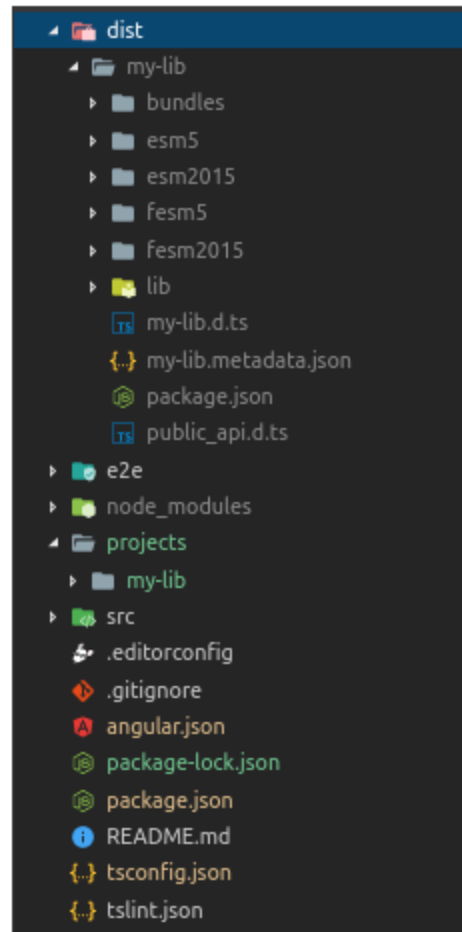
package.json of the my-project root

- El nombre del script puede ser cualquiera, en este caso se ha puesto “**build:lib**” y apunta al archivo **ng-package.json** que es el que le indica dónde está la librería y como la va a transpilar. Ahora ejecutamos nuestro nuevo comando:

| `$ npm run build:lib`



- Esto generará una carpeta **dist** en el proyecto raíz (my-project) con la librería transpilada.



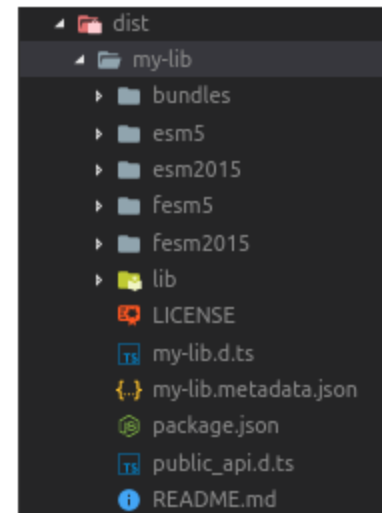
transpiled lib



¿ QUE SUBIR A NPM ?

- Para NPM entramos a la carpeta **dist** que se generó al correr el comando **npm run build:lib** y entramos a la carpeta **my-lib**.

```
| $ cd dist/my-lib
```



transpiled lib



- Esto es lo que se subirá a NPM.
- Una vez posicionado en la carpeta **dist/my-lib**, en nuestra terminal accedemos a nuestra cuenta de NPM:

```
| $ npm login
```

- les pedirá las credenciales y hacemos la publicación de nuestro paquete:

```
| $ npm publish
```

- Recuerdar que cada modificación a nuestro paquete hay que cambiar la versión del mismo , esto lo podemos encontrar en el package.json y habrá que modificarlo cada vez que volvamos a publicar una versión estable de nuestra librería

```
"version": "1.0.1",
```



- Para verificar que se ha publicado nuestra librería en npm:

```
https://npmjs.com/package/<package>
```

- ¿ Como Usamos la librería en otro proyecto ?
- Instalandola como cualquier otra librería

npm install <nombre_librería> --save

- E importando el modulo como cualquier otra

Referencias:

<https://www.ceadesc.org/article/how-to-develop-consume-and-publish-an-angular-library-to-npm-202aaa/>

<https://medium.com/angular-in-depth/complete-beginner-guide-to-publish-an-angular-library-to-npm-d42343801660>

<https://medium.com/@hernandez.hs/creando-una-biblioteca-simple-usando-cli-angular-7-y-desplegando-en-azure-d8c9d6668bd>

