

En las lecciones anteriores, aprendimos a trabajar con **NPM** para reutilizar y crear paquetes. Ha llegado el momento de ver cómo podemos publicar nuestros paquetes en el repositorio oficial de **NPM**. Siendo ésta la lección que cierra la parte específica de **NPM**.

La lección comienza con una introducción del concepto de publicación. A continuación, mostramos cómo crear una cuenta de usuario en el repositorio oficial de **NPM**, a través de la cual realizar nuestras publicaciones. Antes de adentrarnos en la manera de realizar las publicaciones, mostramos cómo abrir sesiones con el repositorio. Finalmente, explicamos cómo publicar en el repositorio.

Al finalizar la lección, el estudiante sabrá:

- Cómo publicar paquetes en el repositorio oficial de **NPM**.
- Cómo suprimir paquetes del repositorio oficial de **NPM**.
- Cómo crear la cuenta de usuario en el repositorio oficial de **NPM**.
- Cómo abrir sesiones **NPM** para realizar publicaciones.
- Cómo asociar etiquetas a publicaciones.

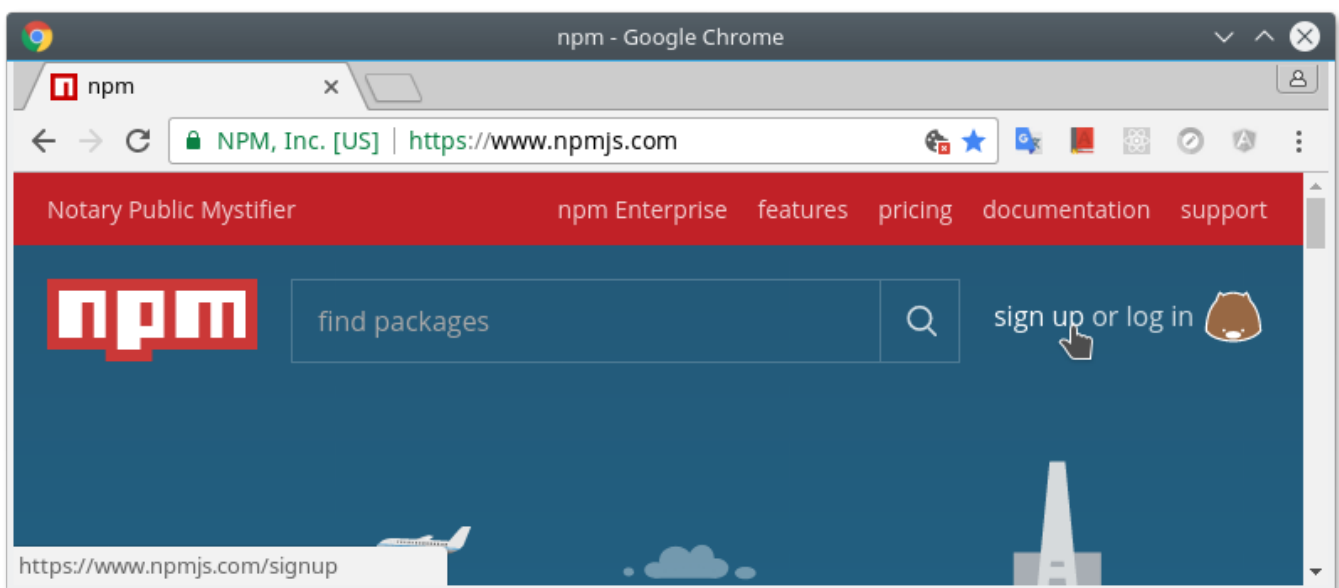
### Introducción

La **publicación** (*publish*) es la operación mediante la cual registramos un paquete en el repositorio **NPM**. Para poder realizar publicaciones, es necesario disponer de una cuenta de usuario en el repositorio. De esta manera, sólo el propietario del paquete podrá realizar publicaciones.

Una vez publicado un paquete en el repositorio, podremos instalarlo fácilmente mediante el comando **npm install**.

### Cuenta de usuario NPM

Para poder publicar paquetes en el repositorio de **NPM**, necesitamos estar dados de alta en el repositorio. Esto se puede conseguir mediante su sitio web oficial, [npmjs.org](https://www.npmjs.org), o el comando **npm adduser**. Recomendamos hacerlo mediante el sitio web. Para ello, vaya a [npmjs.org](https://www.npmjs.org) y haga clic en el enlace **sign up**:



Y a continuación, rellene el formulario:

npm - Google Chrome

NPM, Inc. [US] | <https://www.npmjs.com/signup>

## Create a profile *or* Log in

**First Name**

**Last Name**

**Public Email**

**Username**

username

<https://www.npmjs.com/~username>

**Password**

☒ Sign up for the **npm Weekly**

☐ I agree to the **End User License Agreement** and the **Privacy Policy**.

Your email address will show on your profile page, but npm will never share or sell it.

Create an Account

Haga clic en **Create an Account** cuando haya terminado.

## Sesiones NPM

Una **sesión** (*session*) es una conexión con el repositorio o registro de **NPM**. Para poder publicar paquetes en **NPM**, además de una cuenta de usuario, es necesario abrir una sesión con el servidor a través de la cual realizar las publicaciones. Esta sesión se asocia a una determinada cuenta y se abre mediante **npm login**:

```
npm login [--registry=url]
```

El comando le solicitará el nombre de usuario de su cuenta, la contraseña y la dirección de correo electrónico pública.

Para cerrar la sesión, se utiliza el comando **npm logout**:

```
npm logout [--registry=url]
```

Para saber bajo qué cuenta de usuario está abierta la sesión, se utiliza el comando `npm whoami`:

```
npm whoami [--registry=url]
```

## Publicación de paquetes

Recordemos que cada publicación está relacionada con un paquete, su nombre y su versión. Nunca puede haber dos publicaciones en el repositorio con el mismo nombre de paquete y la misma versión. Si realizó una publicación equivocada, tendrá que publicar el paquete de nuevo con un número de versión distinto.

Para publicar un paquete, hay que utilizar el comando `npm publish`:

```
npm publish
npm publish archivo.tgz
npm publish directorio
```

La primera variante publica el paquete del directorio actual. Mediante la segunda sintaxis, el paquete se encuentra comprimido en un archivo `.tgz`. Finalmente, la tercera opción publica el paquete del directorio indicado.

Por ejemplo, supongamos que nos encontramos en el directorio del proyecto. El cual se encuentra escrito bajo la especificación `ES2017`, no entendible en estos momentos por el motor de `Node`. En este caso, lo mejor es compilar la aplicación a `ES5` y publicar el contenido bajo esa especificación. Generalmente, esto se hace depositando el código compilado del paquete en una carpeta especial, `dist/`, dentro del proyecto. Si deseamos publicar el contenido de `dist`, no habrá más que hacer lo siguiente:

```
npm publish dist
```

## Etiquetas

Una *etiqueta* (*tag*) es un nombre que identifica una determinada versión de un paquete. Un paquete puede tener tantas etiquetas como sea necesario, pero en todo momento una etiqueta sólo puede asociarse a una determinada versión. Se puede utilizar etiquetas para identificar de manera más fácil versiones alfa, beta y *RC* (*release candidate*) del paquete. Así, si deseamos instalar la última versión beta de un paquete, podríamos usar lo siguiente:

```
npm install -g react@beta
```

Observe que en vez de indicar el número de versión, se indica la etiqueta. Cada vez que se publica una nueva versión bajo la etiqueta `beta`, la etiqueta deja de apuntar a la versión anterior y pasa a hacerlo a la última. Es más fácil recordar etiquetas que números de versión.

La etiqueta `latest` se reserva para identificar la última versión publicada en el repositorio `NPM`.

## Asociación de etiquetas a versiones

Una versión se puede etiquetar durante la publicación o una vez publicado. Para hacerlo durante la publicación, es muy fácil hacerlo, no hay más que añadir la opción `--tag` y su nombre al comando `npm publish`. He aquí un ejemplo:

```
npm publish --tag beta dist/
```

Si ya hemos publicado el paquete, podemos usar el comando `npm dist-tag`:

```
npm dist-tag add paquete@versión etiqueta
```

## Supresión de etiquetas

Para suprimir una etiqueta, se puede usar la siguiente sintaxis del comando `npm dist-tag`:

```
npm dist-tag rm paquete etiqueta
```

## Consulta de etiquetas

Para conocer qué etiquetas están asociadas a un determinado paquete, se puede utilizar el comando `npm dist-tag` como sigue:

```
npm dist-tag ls paquete
```

Veamos un ejemplo de su salida:

```
$ npm dist-tag ls react
0.10.0-rc1: 0.10.0-rc1
0.11.0-rc1: 0.11.0-rc1
latest: 15.3.2
next: 15.4.0-rc.4
$
```

## Supresión de paquetes del repositorio

---

Es posible suprimir paquetes del repositorio. Esta supresión es parcial. Se suprime el paquete del repositorio, pero no se podrá publicar ningún paquete con el mismo nombre y las versiones publicadas. El paquete ya no será instalable desde el repositorio, pero no se borra los identificadores formados por el nombre y las versiones de todas las publicaciones realizadas.

Esta supresión se hace mediante el comando `npm unpublish`:

```
npm unpublish --force paquete
npm unpublish paquete@versión
```

La primera versión suprime todas las publicaciones. La segunda sólo la de la versión indicada.