

Crear modulo NPM.

Primero configuramos nuestra cuenta de npm:

```
npm set init.author.name "Selene Night"
npm set init.author.email "selene@loquesea.com"
npm set init.author.url "https://algo.com"
```

hay que crear cuenta en el sitio de npm

npm login...

Para la versión es importante usar correctamente la semantica de versionado. (semver)

<https://semver.org/>

a.b.c → a= version de la API, b= cambios en la interfaz, c= corrigiendo bugs o sumando parches.

Vamos a crear la siguiente estructura:

```
--> miModulo(master)$ tree
.
├── LICENSE
├── package.json
├── README.md
└── src
```

Donde README.md

Platzom

Idioma inventado para el curso de [Fundamentos de JavaScript](https://platzy.com/js) de [Platzy](https://platzy.com).

Descripción del idioma.

- + si la palabra termina en "ar" se le quitan.
- + Si la palabra inicia con z, se añade "pe" al final-
- + Si la palabra tiene 10 o mas letras se parte a la mitad, y se une con un guion.
- + Si es un palindromo, se retorna la misma palabra intercalando mayusculas y minusculas.

Instalación.

...

```
npm install platzom
```

...

Uso

```
...  
import platzom from 'platzom';  
  
platzom('Programar'); // Program  
...  
  
## Creditos.  
  
- [Sacha Lifszyc](http://twiter.com/@slifszyc)  
  
## Licencia  
  
[MIT](https://opensource.org/licenses/MIT)
```

La licencia puede ser del MIT.

Dentro de src tenemos el código fuente del módulo que vamos a crear.

Instalamos:

```
$ npm install -D babel-cli  
$ npm install babel-preset-env -D
```

y debemos crear el archivo .babelrc con:

```
{  
  "presets": ["env"]  
}
```

También debemos crear tarea en el package.json:

```
"build": "babel src/platzom.js -d dist"
```

El archivo que queda en la carpeta “dist” es el que babel transpila para ser llevado a producción.

TESTEANDO

```
$ npm install mocha chai -D
```

mocha es para hacer los test, y chaid para hacer aserciones, osea, decir cual es el valor que se espera sea retornado.

Ahora creamos una carpeta llamada test con un archivo llamado test.js

en este escribimos nuestro test. (revisar archivo.)

Ahora lo publicamos en github y en NPM.

Se sube correctamente a un repositorio de github, ignorando archivos innecesario, en especial el directorio node_modules. (.gitignore) y después del push, ejecutamos:

```
$ npm publish
```