

El objeto de esta práctica es afianzar, reforzar y consolidar los conocimientos teóricos presentados en la lección.

Al finalizarla, el estudiante:

- Habrá editado los archivos de configuración.
- Habrá instalado paquetes globalmente.
- Habrá utilizado paquetes instalados globalmente.
- Habrá consultado la caché de **npm**.

Objetivos

El objetivo de la práctica es mostrar cómo trabajar con el comando **npm** de manera global.

Configuración de npm

Se recomienda configurar **npm** en el momento de la instalación de **Node**. Por favor, vuelva a la primera práctica y eche un vistazo al documento de instalación de **Node**. Concretamente, a la sección de configuración de **npm**. Si no ha realizado la configuración como se indica ahí, hágalo. Si ya lo ha hecho, basta con recordarlo.

Edición de los archivos de configuración

A continuación, vamos a editar los archivos de configuración de usuario y global mediante el propio comando **npm**. Así no es necesario recordar sus ubicaciones.

1. Abrir una consola.
2. Configurar el editor que debe usar **npm** para abrir los archivos de configuración cuando usemos el comando **npm config [-g] edit**. En nuestro caso, usaremos **Atom** de **GitHub**:

```
$ npm config set editor atom
```

De manera predeterminada, usa **vi** con el muchos usuarios no se sienten cómodos.

3. Mostrar la configuración que se aplicará al usuario, según su configuración específica y la heredada de la global y de instalación:

```
$ npm config list
```

Cuando a un parámetro se le asigna valores en distintos archivos de configuración, la última asignación sobrescribe a las anteriores.

4. Mostrar la configuración global:

```
$ npm config -g list
```

5. Editar el archivo de configuración del usuario:

```
$ npm config edit
```

Las líneas que comienzan por punto y coma (;) son comentarios. Observe los posibles parámetros configurables.

6. Observar la ubicación del archivo en el editor: **~/ .npmrc**.

7. Salir del editor.

8. Editar el archivo de configuración global:

```
$ npm config -g edit
```

9. Observar la ubicación del archivo en el editor: **~/ .npm/etc/npmrc**.

10. Salir del editor.

Consulta de paquetes instalables

Antes de instalar un paquete, vamos a ver cómo podemos consultar aquellos que se encuentran publicados en el repositorio oficial de **NPM**.

1. Abrir el navegador.
2. Ir a npmjs.org.
3. Escribir `lodash` en el cuadro de búsqueda:



4. Hacer clic en la lupa.
5. Echar un vistazo a la página de publicación del paquete.
6. Ir a la consola.
7. Realizar la búsqueda mediante el comando **npm**:

```
$ npm search lodash
```

Instalación de paquetes globales

Ahora, vamos a instalar paquetes globalmente, esto es, en el directorio predeterminado:

1. Ir a la consola.
2. Mostrar la información del directorio predeterminado:

```
$ npm prefix -g
/home/me/.npm
$ npm root -g
/home/me/.npm/lib/node_modules
$ npm bin -g
/home/me/.npm/bin
$
```

Recordemos que los paquetes instalados globalmente se ubican en la raíz (**root**). Y los *scripts* ejecutables en **bin**.

3. Instalar globalmente el paquete **justo-cli**:

```
$ npm install -g justo-cli
```

Este paquete instala un *script* ejecutable.

4. Comprobar que tenemos acceso a la carpeta **bin** invocando el *script* **justo** instalado por el paquete **justo-cli**:

```
$ justo -h
```

Si no tiene acceso al *script*, compruebe si la variable de entorno **PATH** contiene la ruta al subdirectorio **bin** del directorio global predeterminado. Y si el usuario tiene permiso de ejecución sobre este directorio.

5. Instalar el paquete **lodash** globalmente:

```
$ npm install -g lodash
```

6. Listar los paquetes instalados globalmente:

```
$ npm ls -g
```

7. Mostrar la ruta al directorio predeterminado:

```
$ npm prefix -g
```

8. Ir al directorio predeterminado.

9. Mostrar el contenido del directorio.

Debe aparecer los directorios **bin**, **etc** y **lib**.

10. Entrar en el directorio `bin`.
11. Listar el contenido del directorio.
En este directorio, `npm` instala los *scripts* ejecutables de los paquetes. No todos los paquetes tienen por qué instalar *scripts*. Pero si lo hacen, se instalarán aquí.
Debe haber una entrada asociada a `justo`. Este *script* es instalado por el paquete `justo-cli`. Recuerde que los paquetes pueden instalar paquetes reutilizables y *scripts* ejecutables.
12. Volver al directorio anterior.
13. Entrar en el directorio `lib`.
14. Listar el contenido del directorio.
Debe aparecer el directorio `node_modules`, donde se instalan los paquetes globalmente.
15. Ir al directorio `node_modules`.
16. Listar el contenido del directorio.
Observe los paquetes que se han instalado globalmente.
17. Volver al directorio predeterminado.
18. Entrar en el directorio `etc`.
19. Listar el contenido del directorio.
Aquí se encuentra el archivo de configuración `npmrc` global. El que se edita mediante el comando `npm config -g edit` y el que se actualiza cuando usamos `npm config -g set`.
20. Ir al directorio `home` del usuario.

Importación de paquetes

Ahora, vamos a usar el paquete `lodash` instalado globalmente:

1. Ir a la consola.
2. Abrir `node` en modo interactivo:

```
$ node
>
```
3. Importar el paquete `lodash`:

```
> const lodash = require("lodash");
undefined
>
```
4. Mostrar el objeto API del paquete:

```
> lodash
```
5. Reordenar aleatoriamente los elementos de un *array* mediante la función `shuffle()`:

```
> lodash.shuffle([1, 2, 3, 4, 5, 6])
[ 1, 3, 6, 5, 4, 2 ]
>
```
6. Salir del intérprete interactivo de `node`:

```
> .exit
$
```

También se puede usar `Ctrl+D` o dos `Ctrl+C` seguidos.

Caché de npm

A continuación, vamos a mostrar la caché de `npm`, donde se almacena los paquetes descargados del repositorio `NPM`:

1. Ir a la consola.
2. Mostrar la ubicación de la caché de `npm`:

```
$ npm config -g get cache
```

3. Listar los paquetes de la caché de **npm**:

```
$ npm cache ls
```