

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

Al finalizarla, el estudiante:

- Habrá creado un paquete.
- Habrá enlazado el paquete al directorio global predeterminado.
- Habrá instalado el paquete en el directorio global predeterminado.

### Objetivos

---

El objetivo de la práctica es crear un paquete personalizado. Para ello, crearemos un paquete que simule las operaciones aritméticas de suma y resta.

### Preparación del entorno

---

Para ayudarnos a arrancar el proyecto del paquete, vamos a usar el generador de **Justo** para **Node**. Primero, vamos a instalarlo:

1. Abrir una consola.
2. Instalar el comando CLI de **Justo** y el generador, ambos globalmente:  

```
$ npm install -g justo-cli justo-generator-node
```
3. Comprobar que tenemos acceso al comando **justo** instalado mediante el paquete **justo-cli**:  

```
$ justo -h
```
4. Comprobar que tenemos acceso al generador **Justo** para **Node**:  

```
$ justo -g node help
```

### Creación del proyecto

---

Lo siguiente es crear la estructura de directorios del proyecto:

1. Ir a la consola.
2. Crear el directorio de la práctica e ir a él.  
Para esta práctica, llámelo paquete.
3. Crear la estructura de directorios inicial del proyecto mediante el generador de **Justo**:  

```
$ justo -g node
```

El generador necesita información sobre el nuevo paquete para rellenar algunos archivos y para generar la mejor estructura de directorios posible. Responda a las preguntas como si estuviera ante un proyecto real.

Cuando le pregunte el tipo de paquete a crear, aplicación o biblioteca, seleccione **lib**.
4. Echar un vistazo detenido a la estructura de directorios del proyecto generado.
5. Echar un vistazo al archivo de metadatos **package.json**.
6. Echar un vistazo al módulo principal del paquete, **index.js**.  
Recuerde que se indica mediante la propiedad **main** del archivo **package.json**. Por convenio y buenas prácticas, se usa **index.js**.
7. Instalar las dependencias del proyecto:  

```
$ npm install
```

Este comando lee el archivo de metadatos `package.json` e instala los paquetes indicados en las propiedades `dependencies` y `devDependencies`.

## Añadidura de módulo de cálculo

Una vez tenemos la estructura inicial del proyecto, lo siguiente es definir el módulo a través del cual realizar las operaciones de suma y resta:

1. Crear el archivo `calcul.js` en el directorio `lib`:

```
/**
 * Suma los valores pasados como argumentos.
 *
 * @param ...args:number[] Los valores a sumar.
 */
export function sum(...args) {
  var res = 0;
  for (let arg of args) res += arg;
  return res;
}

/**
 * Resta los valores pasados como argumentos.
 *
 * @param ...args:number[] Los valores a restar.
 */
export function sub(...args) {
  var res = 0;
  for (let arg of args) res -= arg;
  return res;
}
```

Como el código lo estamos desarrollando bajo la especificación `ES2015` o superior, la exportación de los objetos del módulo los podemos indicar mediante la sentencia `export`.

2. Definir la API del paquete.

Para ello, editar el archivo principal del paquete, recordemos, su punto de entrada, lo que ejecutará `node` cuando lo importe por primera vez, y exportar los elementos de la API:

```
export {sum, sub} from "../lib/calcul";
```

3. Ir a la consola.

4. Mostrar las tareas automatizadas del catálogo del proyecto:

```
$ justo -c
```

Se utiliza `Justo.js` como herramienta de automatización de las tareas de desarrollo.

5. Generar el paquete final del proyecto, en el directorio `dist`, mediante la tarea automatizada `build`:

```
$ justo build
```

6. Echar un vistazo a la carpeta `dist`.

El paquete a publicar se encuentra bajo la especificación `ES5` en la carpeta `dist/es5/nodejs/paquete`. `node` no soporta completamente la especificación `ES2015`, pero se recomienda encarecidamente desarrollar bajo esta especificación o una superior. De ahí que tengamos que compilar el paquete a una especificación soportada íntegramente por `node`, la `ES5`. El paquete compilado lo dejamos en el directorio `dist`.

7. Ir al directorio final del paquete, esto es, `dist/es5/nodejs/paquete`.

8. Instalar dependencias:

```
$ npm install --production
```

Recuerde que hay una diferencia entre el directorio del proyecto, desarrollado con herramientas de `Node` y el paquete final a usar los usuarios. Como vamos a probar el paquete, tenemos que instalar sus dependencias en el paquete final.

La opción `--production` instala sólo los paquetes indicados en la propiedad `dependencies` del archivo `package.json`.

9. Crear un enlace en el directorio predeterminado global para poder usarlo:

```
$ npm link
```

Recuerde que esto *no* instala el paquete en el directorio predeterminado. Simplemente, crea un enlace al directorio actual en el que se encuentra instalado el paquete. Por lo que el paquete debe encontrarse completamente instalado en el directorio referenciado, de ahí que hayamos tenido que instalar sus dependencias.

10. Comprobar la instalación del paquete mediante el enlace:

```
$ npm ls -g paquete
```

11. Abrir otra consola.

12. Ir al directorio *home* del usuario.

13. Abrir el intérprete de **node** en modo interactivo.

14. Importar el paquete:

```
> var calcul = require("paquete");  
undefined  
>
```

15. Mostrar el objeto API del paquete:

```
> calcul  
{ sum: [Getter], sub: [Getter] }  
>
```

16. Calcular la suma de los números 1, 3, 5, 7 y 9:

```
> calcul.sum(1, 3, 5, 7, 9)  
25  
>
```

17. Calcular la resta de los números 1, 3, 5, 7 y 9:

```
> calcul.sub(1, 3, 5, 7, 9)  
-25  
>
```

18. Salir del intérprete interactivo.

19. Suprimir el enlace:

```
> npm uninstall -g paquete
```

20. Comprobar que no existe el paquete en el directorio predeterminado:

```
> npm ls -g paquete
```

## Instalación del paquete

Ahora, vamos a instalar el paquete. No vamos a crear un enlace como antes, sino a instalarlo en el directorio global, sin necesidad de publicarlo previamente. En este caso, no es necesario tener en el paquete las dependencias instaladas. Se instalarán cuando lo instalemos en el directorio predeterminado.

1. Ir a la consola.
2. Ir al directorio que contiene el paquete final: `dist/es5/nodejs/paquete`.
3. Instalar el paquete globalmente:

```
> npm install -g .
```

4. Comprobar que se ha instalado:

```
> npm ls -g paquete
```

5. Probar el paquete instalado:

```
$ node -p "require('paquete').sum(1, 2, 3, 4, 5, 6)"  
21  
$
```

6. Desinstalar el paquete:

```
> npm uninstall -g paquete
```