

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

Al finalizar la práctica, el estudiante:

- Habrá creado macros.
- Habrá registrado macros en el catálogo.
- Habrá invocado macros del catálogo de un proyecto.

creación del proyecto

Para comenzar, vamos a crear el directorio de un proyecto **Justo.js**:

1. Abrir una consola de **PowerShell**.
2. Crear el directorio de la práctica e ir a él.
3. Crear los archivos **Justo.json**, **Justo.js** y **package.json**.
4. Instalar las dependencias:

```
> npm install
```

registro de macros de tareas

A continuación, vamos a crear una macro que tiene como objeto, por un lado, crear el directorio **backup** y, como segunda función, copiar el contenido del archivo **Justo.json** en él. Para ello, usaremos las tareas simples **create** y **copy** del *plugin* **justo-plugin-fs**. Más adelante en el curso, veremos detenidamente qué son los *plugins*; grosso modo, un componente de tareas reutilizables.

1. Ir a la consola de **PowerShell**.
2. Instalar, localmente al proyecto, el *plugin* **justo-plugin-fs** para poder usarlo:

```
> npm install --save-dev justo-plugin-fs
```

3. Editar el archivo **Justo.js**, por ejemplo, mediante **Atom**:

```
> atom Justo.js
```

4. Importar las tareas simples **create** y **copy** del *plugin*:

```
//imports
const justo = require("justo");
const catalog = justo.catalog;
const copy = require("justo-plugin-fs").copy;
const create = require("justo-plugin-fs").create;
```

5. Añadir la macro de tareas **backup** con las tareas de creación y copia como, por ejemplo:

```
//catalog
catalog.macro({name: "backup", desc: "Hacer copia de seguridad de Justo.json."}, [
  {
    title: "Crear directorio de backup",
    task: create,
    params: [{
      dir: "backup"
    }]
  },
],
{
  title: "Copiar Justo.json",
```

```

    task: copy,
    params: [{
      src: "Justo.json",
      dst: "backup/"
    }]
  }
});

```

6. Guardar cambios.

7. Ir a la consola de **PowerShell**.

8. Listar las tareas registradas en el catálogo del proyecto:

```

> justo -l
Name      Description
backup    Hacer copia de seguridad de Justo.json.
default    Default task.
>

```

9. Ejecutar la macro backup:

```

> justo backup

backup
backup
  V Crear directorio de backup (5 ms)
  V Copiar Justo.json (7 ms)

OK 2 | Failed 0 | Ignored 0 | Total 2
>

```

10. Comprobar que se ha copiado el archivo mostrando su contenido:

```

> cat backup/Justo.json

```

TAREA predeterminada

Finalmente, vamos a configurar la tarea predeterminada para que ejecute la tarea backup como predeterminada:

1. Editar el archivo **Justo.js**.

2. Modificar la tarea catalogada **default** para que ejecute backup:

```

catalog.macro({name: "default", desc: "Default task."}, ["backup"]);

```

3. Guardar cambios.

4. Ir a la consola de **PowerShell**.

5. Ejecutar la tarea predeterminada:

```

> justo

default
default
  backup
    V Crear directorio de backup (0 ms)
    V Copiar Justo.json (16 ms)

OK 2 | Failed 0 | Ignored 0 | Total 2
>

```

6. Cerrar la consola.

No es necesario que la tarea predeterminada, recordemos, aquella que se registra en el catálogo como **default**, sea una macro de tareas, pero por convenio y buenas prácticas, así se hace.