

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

Al finalizarla, el estudiante:

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

### Creación del proyecto

---

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

1. Abrir una consola.
2. Crear el directorio de la práctica e ir a él.
3. Crear el directorio de un módulo mediante el generador **justo-generator-justo**:  

```
$ justo -g justo module
```
4. Instalar dependencias:  

```
$ npm install
```

### Registro de flujos de trabajo

---

A continuación, vamos a simular la creación de cuentas de usuario. Para ello, crearemos dos tareas simples, una que comprueba si una cuenta de usuario existe y otra que la crea. En ambos casos, vamos a simular el entorno. Para ello, consideraremos que en nuestro entorno hay tres cuentas ya creadas, uno, dos y tres.

1. Editar el archivo **Justo.js**.
2. Crear la tarea simple **existe**, sin catalogarla, que comprueba y devuelve si existe una determinada cuenta:  

```
const existe = justo.simple(function(params) {  
  return ["uno", "dos", "tres"].indexOf(params[0]) >= 0;  
});
```
3. Crear la tarea simple **crea** que simula la creación de una cuenta, aunque no lo hará:  

```
const crea = justo.simple(function(params) {  
  //aquí vendría el código que crea la cuenta en el sistema  
});
```
4. Registrar el flujo de trabajo **crear** en el catálogo del proyecto, para añadir una cuenta al sistema si no existe:  

```
catalog.workflow({name: "crear", desc: "Crear cuenta de usuario."}, function(params) {  
  var cuenta = params[0];  
  
  if (!existe(`Comprobar existencia de '${cuenta}'.`, cuenta)) {  
    crea(`Crear cuenta '${cuenta}'.`, cuenta);  
  }  
});
```
5. Guardar cambios.
6. Ir a la consola.
7. Listar las tareas registradas en el catálogo:  

```
$ justo -c
```

Name	Description
crear	Crear cuenta de usuario.
default	Default task.

\$

8. Simular la creación de la cuenta uno:

\$ justo crear:uno

```
crear
  crear
    [ OK ] Comprobar existencia de 'uno'. (0 ms)
```

OK 1 | Failed 0 | Ignored 0 | Total 1

\$

Como supuestamente la cuenta uno *ya* existe, no se ejecuta la tarea simple crea.

9. Simular la creación de la cuenta cuatro:

\$ justo crear:cuatro

```
crear
  crear
    [ OK ] Comprobar existencia de 'cuatro'. (0 ms)
    [ OK ] Crear cuenta 'cuatro'. (0 ms)
```

OK 2 | Failed 0 | Ignored 0 | Total 2

\$

Ahora, en cambio, como cuatro supuestamente *no* existe, sí se ejecuta la tarea crea.

10. Cerrar la consola.