
	<b>Acceso a datos</b> <b>Tema 4</b>		<b>Control</b>
	<b>API REST</b>		

## Ejercicio

Para la realización del ejercicio se permite acceder a las prácticas y apuntes realizados en clase. Además, se permite solo al acceso a la documentación oficial de Java.

**Documentación:** <https://docs.oracle.com/en/java/javase/19/>

Deberás crear una carpeta llamada **ControlT4**, dentro de ella deberás crear las carpetas **server** y **client**, que contendrán los proyectos del cliente y servidor respectivamente.

En el lado del servidor deberás crear una **API REST** en el lenguaje **JAVA** usando **Spring Framework**.

En el lado del cliente deberás crear una aplicación desarrollada en **React**, puedes utilizar cualquier librería que necesites para la realización de la actividad.

Los datos deben estar almacenados en una base de datos llamada **controlT4**. La tabla de la base de datos será creada por el propio programa y se llamará **Orders** con la estructura de la imagen. La tabla contiene información relacionada con los pedidos realizados a una empresa, donde un pedido puede tener estar formado por varios productos. El código de la estructura de la tabla y los datos a insertar se te han proporcionado.

Orders	
<b>PK</b>	<b>order_id BIGINT NOT NULL</b>
<b>PK</b>	<b>item_no BIGINT NOT NULL</b>
	product_id INT NOT NULL
	description VARCHAR(255)
	price DOBLE DEFAULT 0
	quantity DOBLE DEFAULT 0

Puntuación total de cada actividad **2,5p**. Desglosados de la siguiente manera:

Correcto funcionamiento de la actividad. (1,75p)

Código limpio siguiendo las buenas prácticas vistas en clase. (0,75p)

**Cuando finalices cada una de las actividades debes avisarme para comprobar su correcto funcionamiento.**

## **PARTE SERVIDOR**

Para organizar el código deberás crear las clases en su propio paquete siendo el paquete principal **com.accesodatos**.

### **Actividad 1:**

Crear el siguiente endpoint:

**localhost:8080/api/v1/orders/**

La información que devolverá es el **id** de cada pedido y el **importe total** de cada uno de ellos.

Se devolverá una lista de datos con una estructura como la que se muestra en la siguiente imagen:

```
[
  {
    "orderId": 1,
    "total": 2500.0
  },
  {
    "orderId": 2,
    "total": 4300.0
  },
  {
    "orderId": 3,
    "total": 350.0
  },
  {
    "orderId": 4,
    "total": 150.0
  }
]
```



### **Actividad 2:**

Crear el siguiente endpoint:

**localhost:8080/api/v1/orders/2**

Éste devolverá la lista de productos del pedido solicitado a través de la URL.

La estructura de datos es la mostrada en la siguiente imagen. En el ejemplo estamos solicitando los datos referentes al pedido número 2:

	<b>Acceso a datos</b> <b>Tema 4</b>		<b>Control</b>
	<b>API REST</b>		

```

{
  "orderId": 2,
  "itemNo": 1,
  "productId": 1,
  "description": "Spring FrameWork Book",
  "price": 100.0,
  "quantity": 1.0
},
{
  "orderId": 2,
  "itemNo": 2,
  "productId": 2,
  "description": "MacBook Pro",
  "price": 2000.0,
  "quantity": 2.0
},
{
  "orderId": 2,
  "itemNo": 3,
  "productId": 3,
  "description": "Windows 11",
  "price": 150.0,
  "quantity": 1.0
},
{
  .....
}

```

## **PARTE CLIENTE**

Deberás organizar el código siguiendo el código de buenas prácticas visto en clase. Para la gestión de las rutas de la aplicación deberás utilizar el paquete **react-router-dom**.

### **Actividad 3:**

En esta actividad deberás cargar en una tabla la lista de los pedidos con el importe total de cada uno de ellos, deberás solicitar dicha información al 1º endpoint creado en el servidor.

La url para mostrar los datos será:

**localhost:3000/orders/**

La tabla deberá aparecer de forma centrada en pantalla con la misma estructura como se muestra en la siguiente imagen. Tiene una 3ª columna con un enlace que servirá para realizar la siguiente actividad:

## Control Tema 4

Nº de Pedido	Importe Total	Acción
1	2500 €	<a href="#">Detalle</a>
2	4300 €	<a href="#">Detalle</a>
3	350 €	<a href="#">Detalle</a>
4	150 €	<a href="#">Detalle</a>

### Actividad 4:

En este caso, deberás mostrar los detalles del pedido seleccionado al hacer clic en el enlace **Detalle** de la actividad anterior. La url debería ser la siguiente (en el ejemplo para el pedido 2).

`localhost:3000/orders/2`

Se solicitará la información al 2º endpoint realizado en la parte del servidor. La información se mostrará mediante una tabla.

## Detalle del Pedido

Nº Item	Producto	Descripción	Precio	Cantidad	Importe
1	1	Spring FrameWork Book	100 €	1	100 €
2	2	MacBook Pro	2000 €	2	4000 €
3	3	Windows 11	150 €	1	150 €
4	4	RedHat Linux	50 €	1	50 €

Importe Total: 4300 €

[Volver](#)

Deberás mostrar el importe total del pedido, como se muestra en la imagen.

Además, tenemos un enlace **Volver** que nos dirigirá a la pantalla principal donde se muestran todos los pedidos.

Para subir los archivos del control deberás comprimir en formato **ZIP** el contenido de la carpeta del control, comprueba que esté todo correcto antes de subirlo a la plataforma elearning.