

# THERMOMIX

## Descripción

---

Nos vamos a encontrar que los Playmobil están organizados en series históricas.

Cada serie histórica ofrece una serie de productos en formato de caja con un precio de venta. La caja está formada por figuras en distintas cantidades.

Finalmente, las cajas están formadas por figuras que no se pueden adquirir de forma individual, siempre formarán parte de una caja.

## Arranque de los contenedores

---

Es preferible que se inicie todo con docker-compose a partir del fichero “docker-compose.yml”.

El comando

```
docker-compose up -d
```

descarga los contenedores y los pone en marcha, mientras que

```
docker-compose down
```

detiene los contenedores y los elimina.

## Endpoints

---

Los endpoints están programados para devolver una respuesta después de varios segundos. El primer endpoint genera error con una frecuencia del 50%. Hay que tener en cuenta ambas situaciones.

### Autenticación de usuarios

El usuario se autentica en el siguiente endpoint empleando el método de envío POST.

`http://127.0.0.1:8081/api/login_user`

```
### Retrieve token success
Send Request
POST http://127.0.0.1:8081/api/login_user
content-type: application/x-www-form-urlencoded

username=sonia.silverado
&password=s
```

### Libros de Thermomix

Un libro recoge una serie de platos. Este endpoint falla con una probabilidad del 50%.

`http://127.0.0.1:8082/api/books`

```
### Retrieve books success
```

```
Send Request
```

```
GET http://127.0.0.1:8082/api/books
```

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6I
```

## Platos de un determinado libro por id

Se suministra en la url el identificador del libro y obtenemos los platos incluidos en dicho libro.

<http://127.0.0.1:8082/api/book/6917/dishes>

```
### Retrieve dishes by book id success
```

```
Send Request
```

```
GET http://127.0.0.1:8082/api/book/6917/dishes
```

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC
```

## Plato por id

Se suministra en la url el identificador del plato y obtenemos información detallada del plato.

<http://127.0.0.1:8082/api/dish/691702>

```
### Retrieve dish by id success
```

```
Send Request
```

```
GET http://127.0.0.1:8082/api/dish/691702
```

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVC
```

## Receta de un determinado plato

Se suministra en la url el código de la receta, que coincide con el código del plato, y obtenemos el detalle de dicha receta.

<http://127.0.0.1:8082/api/recipe/254703>

```
### Retrieve recipe by id success
```

```
Send Request
```

```
GET http://127.0.0.1:8082/api/recipe/254703
```

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpX
```

# Proyecto

---

## Página de autenticación de usuarios

Desarrolla una página con un formulario para la autenticación de usuarios. Debe tener dos campos para solicitar el nombre de usuario y la contraseña.

Los ficheros se llamarán:

- login.html
- login.js

Emplea el endpoint de validación de usuarios consumiendo el endpoint correspondiente. Si la validación tiene éxito se recibe un token que habrá que emplear para poder consumir información de los endpoints anteriores. El token hay que almacenarlo en una variable de sesión y redireccionar a la página de selección de libros de Thermomix.

## Página de selección de un libro

Desarrolla una página en la que se muestren todas las series. Se mostrarán en una tabla con una única columna que muestra la denominación de la serie.

Los ficheros se llamarán:

- choose\_book.html
- choose\_book.js

Primero hay que comprobar que el usuario se ha autenticado consultando si el token está almacenado en una variable de sesión. Si no hay token hay que redireccionar al usuario a la página de login. Si hay token hay que extraer del token el nombre y el apellido para mostrarlo en la cabecera.

Emplea atributos inventados (data-) para colocar en cada fila el identificador del libro. Luego desde JavaScript tendrás que recuperar dicho atributo inventado (dataset) para saber sobre qué libro se ha clicado.

Al clicar en un libro debe ir a la página de selección de un plato. Hay que hacer una redirección con JavaScript.

Durante la carga de la información deben mostrarse mensajes informativos adecuados: espere a que la información esté disponible, error, ...

## Página de selección de un libro

Desarrolla una página en la que se muestren los platos de un determinado libro.

El identificador del libro debe ser extraída de la url. La url seguirá el siguiente formato (probado desde Live Server).

`http://127.0.0.1:5000/views/choose_dishes.html?book=6917`

Los ficheros se llamarán:

- choose\_dish.html
- choose\_dish.js

Los platos se muestran en una tabla con dos columnas, la primera con la denominación y la segunda con la imagen. Emplea atributos inventados (data-) para colocar en cada etiqueta información relevante.

### Página de receta de un plato

Desarrolla una página en la que se muestren las figuras de una determinada caja.

El identificador de la caja debe ser extraída de la url. La url seguirá el siguiente formato.

`http://127.0.0.1:5000/views/show_recipe.html?box=254703`

Los ficheros se llamarán:

- `show_recipe.html`
- `show_recipe.js`

La receta se muestra en un párrafo.