

Aclaración:

Las partes se corregirán de manera secuencial (ascendentemente). Si están bien todos los puntos de una parte, habilita la corrección de la parte posterior.

Se pueden usar templates o reutilizar código hecho, pero en ningún caso, se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.

Toda comunicación con el backend se realizará con AJAX. Todo el pasaje de datos se hará con JSON.

Se deben respetar los nombres de los archivos, de las clases, de los métodos y de los parámetros de las peticiones.

La Api Rest se creará en Node.js. La misma tendrá que interactuar con la base de datos *jugueteria_bd* (juguetes – usuarios) y gestionar JWT.

NO subir el *node_modules*.

Parte 1 BACKEND – Node.js (hasta un 4)

Diseñar un servidor Node.js (***http://localhost:2023***) con las siguientes rutas y middlewares:

verificar_usuario (MW): Recibe el correo y la clave en un JSON y retorna toda la información del registro de la tabla **usuarios**, de la base de datos *jugueteria_bd*, al siguiente callable.

Si el usuario no existe, retornará un JSON (éxito: false; mensaje: "Correo y/o clave incorrectas"; jwt: null; status: 403).

login (POST): Recibe, del middleware anterior, los datos del usuario en un JSON. Se generará un JWT y se retornará un JSON (éxito: true/false; mensaje: string; jwt: JWT (*) / null; status: 200/403).

(*) el payload del JWT tendrá la siguiente estructura:

- usuario: con todos los datos del usuario, a excepción de la clave.
- alumno: colocar su nombre y apellido
- dni_alumno: indicar el número de DNI del alumno

El token creado deberá ser firmado con el apellido.nombre del alumno y debe expirar en 120 segundos.

verificar_jwt (MW): Recibe el JWT (en el ***Bearer***) y retorna toda la información del JWT decodificada al siguiente callable.

Si el JWT no es válido, retornará un JSON (éxito: false; mensaje: string; status: 403).

login (GET): Recibe, del middleware anterior, el JWT decodificado y retorna un JSON (éxito: true/false; mensaje: string; payload: datos (*) / null; status: 200/403).

(*) datos completos del payload

Parte 1 FRONTEND – HTML5 y TypeScript (hasta un 5)

login.html – login.ts

Desarrollar un proyecto, que desde **http://localhost/lab_3/sp**, se acceda a las siguientes páginas que interactúen con el servidor Node.js.

Asociar 'dinámicamente' al evento click del botón ***btnEnviar***, de la página login.html, una función que recupere el correo y la clave para luego invocar (por ***AJAX***) hacia al verbo POST de la ruta ***/login*** (de la Api Rest, "***http://localhost:2023/login***").

Si el atributo éxito del json de retorno es ***false***, se mostrará un mensaje que indique lo acontecido (por consola y alert).

Si es true, se guardará en el *LocalStorage* el ***JWT*** obtenido y se redireccionará hacia ***principal.html***.

A login form with a pink header containing the word "LOGIN" in blue. Below the header is a light gray form container. Inside the container, there are two input fields. The first is labeled "Correo" and has a mail icon to its left. The second is labeled "Clave" and has a key icon to its left. At the bottom of the form container are two buttons: a blue "Enviar" button and a yellow "Limpiar" button.

LOGIN

Correo

Clave

Enviar Limpiar

Parte 2 BACKEND – Node.js (hasta un 6)

Agregar las siguientes rutas y middlewares a la Api Rest:

agregarJugueteBD (POST): recibirá un JSON → **juguete_json** (marca y precio) y **foto** para agregar un registro en la tabla **juguetes**, de la base de datos *jugueteria_bd*, más el JWT (en el **Bearer**).

Se retornará un **JSON** que contendrá: éxito(bool) y mensaje(string) indicando lo acontecido.

Agregar el middleware **verificar_jwt**.

Nota: La foto guardarla en *./public/juguetes/fotos/*, con el nombre formado por la marca punto extensión.

Ejemplo: mattel.jpg

listarJuguetesBD (GET): Listado de juguetes. Obtendrá el listado completo de los juguetes (array JSON). Recibe el JWT (en el **Bearer**).

Retorna un JSON (éxito: true/false; mensaje: string; dato: arrayJSON / null; status: 200/424).

Agregar el middleware **verificar_jwt**.

toys (DELETE): Borrado de juguetes por ID.

Recibe el ID del juguete a ser borrado (**id_juguete**, como parámetro JSON) más el JWT (en el **Bearer**).

Nota: La foto debe ser borrada.

Retorna un JSON (éxito: true/false; mensaje: string; status: 200/418)

toys (POST): Modificar juguetes por ID.

Recibe el JSON del juguete a ser modificado → **juguete** (id_juguete, marca y precio), la **foto** y el JWT (en el **Bearer**).

El id_juguete será el id del juguete a ser modificado, mientras que el resto, serán los valores a ser modificados.

Agregar el middleware **verificar_jwt**.

Nota: La foto guardarla en *./public/juguetes/fotos/*, con el siguiente formato:

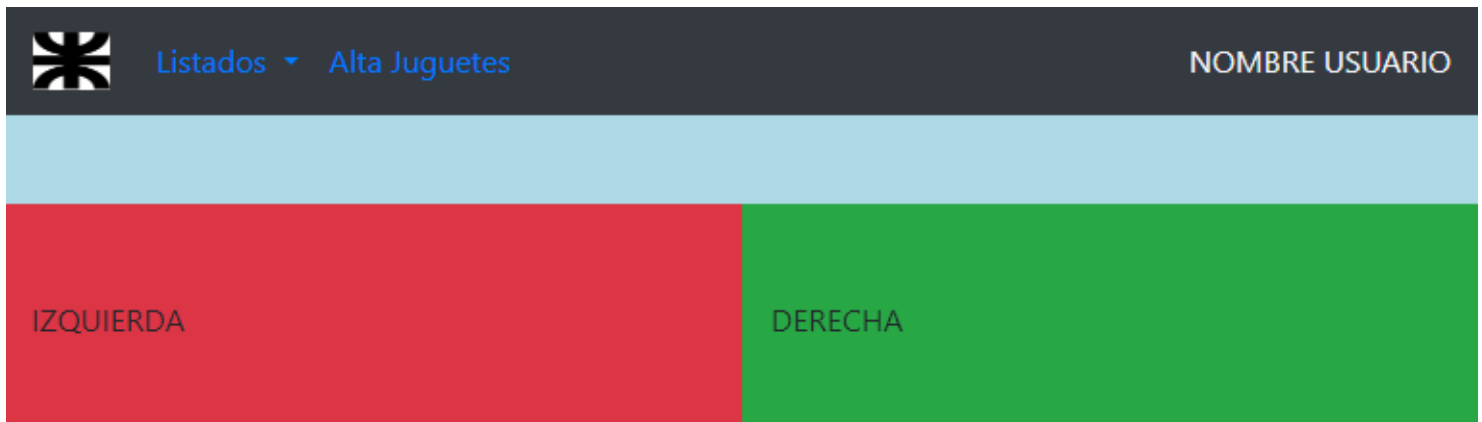
marca_modificacion.extension.

Ejemplo: ./public/juguetes/fotos/mattel_modificacion.jpg

Retorna un JSON (éxito: true/false; mensaje: string; status: 200/418)

Parte 2 FRONTEND – HTML5 y TypeScript (hasta un 7)

principal.html – principal.ts



Al terminar de cargar la página *principal.html*, se deberá verificar que el JWT sea válido. Para ello se deberá recuperar el JWT del `LocalStorage` y se invocará (por **AJAX**) al verbo GET (ruta **/login**). Si el atributo éxito del json de retorno es **false**, redirigir hacia *login.html*. Caso contrario, colocar en `` el nombre del usuario recuperado del JWT.

Al pulsar el submenú *Juguetes* del menú Listados, se mostrará el listado de los juguetes en una tabla de HTML.

Invocar (por **AJAX**) al verbo GET (ruta **/listarJuguetesBD**, de la Api Rest).

Si el atributo éxito del json de retorno es **false**, se mostrará (en un alert y en consola) el mensaje recibido.

Si es true, se armará (en el frontend) el listado que proviene del atributo **dato** del json de retorno. Mostrarlo en la sección izquierda. Las fotos tendrán un ancho y alto de 50px.

Nota: Si el JWT no es válido, redirigir hacia *login.html*.

ID	MARCA	PRECIO	FOTO
1	Mattel	3500	
2	He-man	4000	
3	HotWheels	1550	
4	BJ	8550	

Parte 3 FRONTEND – HTML5 y TypeScript (hasta un 8)

Al pulsar la opción de menú *Alta Juguetes*, se mostrará el formulario de alta de juguetes en la sección derecha (armarlo en el front-end).

Formulario alta / modificación



El formulario se encuentra dentro de un contenedor con un encabezado verde que contiene el título "JUGUETES" en letras azules. El cuerpo del formulario tiene un fondo azul oscuro. Hay tres campos de entrada con iconos grises a la izquierda: "TM" para el campo "Marca", "\$" para el campo "Precio", y una cámara para el campo de selección de archivos. El campo de archivos muestra "Seleccionar archivo" y "Sin ...dos". En la parte inferior hay dos botones: "Agregar" (verde) y "Limpiar" (naranja).

Asociar al evento click del botón **btnAgregar** una función que recupere la marca, el precio y la foto. Invocar (por **AJAX**) al verbo POST (de la ruta **/agregarJugueteBD** de la Api Rest).

Se mostrará (en un alert y por consola) el mensaje recibido. Actualizar el listado de juguetes.

Nota: Si el JWT no es válido, redirigir hacia *login.html*.

Parte 4 FRONTEND – HTML5 y TypeScript (hasta un 10)

Agregar al listado de juguetes, una columna extra, con dos botones.

El primer botón (*btnDelete*) permitirá el borrado del juguete, previa confirmación del usuario, preguntando si el juguete con tal marca y precio se quiere borrar de la base de datos.

Si se confirma, se eliminará el juguete, invocando (por **AJAX**) al verbo DELETE (de la ruta **/toys** de la Api Rest).

Si el atributo éxito del json de retorno es **false**, se mostrará (en un alert y por consola) el mensaje recibido.

Si es true, refrescar el listado de juguetes.

Nota: Si el JWT no es válido, redirigir hacia *login.html*.

El segundo botón (*btnUpdate*) permitirá la modificación del juguete seleccionado. Para ello, se cargarán todos los datos del juguete en el formulario (formulario alta / modificación). Mostrarlo en la sección derecha.

Al pulsar el botón **btnModificar** (cambiarlo en el formulario) se invocará (por **AJAX**) al verbo POST (de la ruta **/toys** de la Api Rest).

Si el atributo éxito del json de retorno es **false**, se mostrará (en un alert y por consola) el mensaje recibido.

Si es true, refrescar el listado de juguetes.

Nota: Si el JWT no es válido, redirigir hacia *login.html*.

ID	MARCA	PRECIO	FOTO	ACCIONES	
1	Mattel	3500		Eliminar	Modificar
2	He-man	4000		Eliminar	Modificar
3	HotWheels	1550		Eliminar	Modificar
4	BJ	8550		Eliminar	Modificar