

## 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 pasaje de datos se hará con JSON.

Las vistas (páginas .php o .html) deben respetar fielmente las formas, colores, iconos, etc..

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.

## Parte 1 BACKEND – Node.js (hasta un 5)

Diseñar un servidor Node.js (***http://localhost:2022***) con las siguientes rutas:

**listarUsuariosBD (GET):** mostrará el listado completo de los usuarios (obtenidos de la base de datos) en un array de Usuario (en formato de cadena JSON).

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

**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*.

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

**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).

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

**login (POST):** Se envía el correo y la clave en un JSON y retorna otro 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
- parcial: indicar el tipo de parcial

El token debe expirar en 120 segundos.

**login (GET):** Se envía el JWT (en el ***Bearer***) y retorna un JSON (éxito: true/false; mensaje: string; payload: datos (\*) / null; status: 200/403).

(\*) datos completos del payload

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

Agregar las siguientes rutas a la Api:

**toys (DELETE):** Borrado de juguetes por ID.

Recibe el ID del juguete a ser borrado (**id\_juguete**, cómo 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, 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.

**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)

Agregar un Middleware que verifique que el token sea válido.

Recibe el JWT (en el **Bearer**) a ser verificado.

Retorna un JSON (éxito: false; mensaje: string; status: 403) en el caso de que el token sea inválido.

Si es un token válido, pasar al siguiente callable.

Aplicarlo a los verbos POST y DELETE de la ruta **/toys**, a los verbos GET de las rutas **/listarUsuariosBD** y **/listarJuguetesBD** y al verbo POST de la ruta **/agregarJugueteBD**.

## Parte 1 FRONTEND – HTML5 y TypeScript (hasta un 8)

*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.

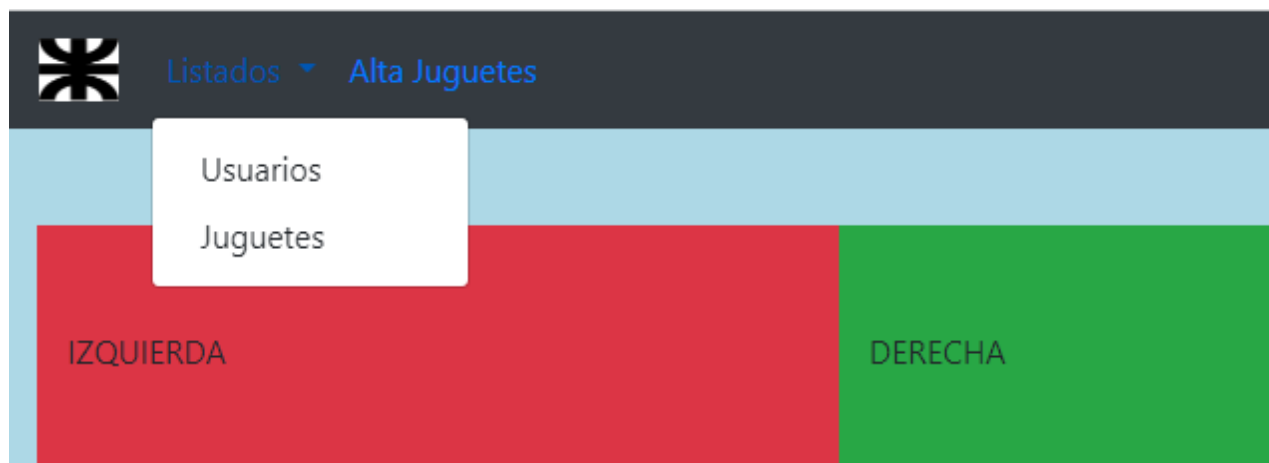
Asociar 'dinámicamente' al evento click del botón **btnEnviar**, de la página login.html, una función que recupere el correo y la clave para luego invocar (por **AJAX**) hacia al verbo POST de la ruta **/login** (de la Api Rest, **"http://localhost:2022/login"**).

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.php**.

The image shows a login form titled "LOGIN" in a pink box. Inside, there's a grey container with two input fields: "Correo" with an envelope icon and "Clave" with a key icon. Below these are two buttons: "Enviar" (blue) and "Limpiar" (yellow).

*principal.php – principal.ts*



*Principal.html* debe tener un menú (cómo muestra la imagen) y poseer una división en el cuerpo de dos secciones (izquierda y derecha).

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






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

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

Si es true, se armará (en el frontend) el listado que proviene del atributo **dato** del json de retorno.

Mostrarlo en la sección derecha.

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

CORREO	NOMBRE	APELLIDO	PERFIL	FOTO
a@a.com	a	a	propietario	
b@b.com	b	b	empleado	
c@c.com	c	c	supervisor	
d@d.com	d	d	empleado	
e@e.com	e	e	empleado	





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 2 FRONTEND – HTML5 y TypeScript (hasta un 10)

Al pulsar la opción de menú *Alta Juguetes*, se mostrará el formulario de alta de juguetes (cómo muestra la imagen) en la sección derecha.

### **Formulario alta / modificación**

El formulario está encapçalado por el título 'JUGUETES' en letras azules sobre un fondo verde. El formulario mismo tiene un fondo de color turquesa. Contiene tres campos de entrada: el primero para la 'Marca' con un icono de 'TM' a la izquierda; el segundo para el 'Precio' con un icono de '\$' a la izquierda; y el tercero para la selección de una foto, con un icono de cámara a la izquierda y un botón 'Sin ...dos' a la derecha. Debajo de estos campos hay dos botones: uno verde con el texto 'Agregar' y otro amarillo con el texto 'Limpiar'.

**Colores e iconos:** darkcyan – fas fa-trademark; fas fa-camera; fas fa-dollar-sign;

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.

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

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 izquierda.

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