

Parte 01

Aclaración: para todos los puntos se deberán crear o modificar las correspondientes clases de PHP.

POO: obligatorio el uso de POO. Todos los métodos deben estar declarados dentro de clases.

Base de Datos: crear y adjuntar el script de la base de datos (con sus tablas) para realizar la funcionalidad pedida.

Crear un **API Rest** para la óptica Coroline, que interactúe con la clase **Anteojos**, la clase **Usuario** y la base de datos **OpticaBD** (anteojos - usuarios - ventas_anteojos).

Crear los siguientes verbos:

A nivel de aplicación:

(POST) Alta de Anteojos (ID*, color, marca, precio y aumento). *

ID auto-incremental.

A nivel de ruta (**/anteojos**):

(GET) Listado de todos los anteojos (JSON - status 200).

A nivel de ruta (**/usuarios**):

(POST) Alta de usuario (ID*, correo, clave, nombre, apellido, perfil** y foto). * ID auto-incremental. ** propietario, encargado y empleado.

A nivel de aplicación:

(GET) Listado de usuarios (JSON - status 200)

A nivel de ruta (**/login**):

(POST) Se ingresa correo y clave. (retorna un **JWT** - status 200)

(GET) Verifica el **JWT**. En caso exitoso, retorna un JSON con mensaje y status 200. Caso contrario, retorna un JSON con mensaje y status 409.

A nivel de ruta (**/ventas**):

(POST) Alta de ventas (ID*, id_anteojos**, cantidad y fecha). * ID auto-incremental. ** ID de la tabla Anteojos.

(GET) Listado de todas las ventas (JSON - status 200).

Todos los verbos invocarán métodos de clase (Anteojos o Usuario) para realizar las acciones.

Crear los siguientes Middlewares (en la clase MW) para que:

1.- (instancia) Verifique que estén "seteados" el correo y la clave.

Si no existe alguno de los dos (o los dos) retorne un JSON con el mensaje de error correspondiente (y status 409).

Si existen, pasar al siguiente Middleware que verifique que:

2.- (clase) Si alguno está vacío (o los dos) retorne un JSON con el mensaje de error correspondiente (y status 409).

Caso contrario, pasar al siguiente Middleware.

3.-(instancia) Verificar que el correo y clave no existan en la base de datos. Si existen, retornar un JSON con el mensaje de error correspondiente (y status 409).
Caso contrario, acceder al verbo de la API.

Estos Middlewares aplicarlos al verbo POST de **/login**.

Parte 02

El token generado debe tener un tiempo de expiración no mayor a 20 segundos.

Crear, a nivel de aplicación, los verbos:
(DELETE) Borrado de anteojos por ID (con token: solo el propietario).
(PUT) Modificar los anteojos por ID (con token: solo propietario y encargado)

Agregar un Middleware (clase MW - método de instancia) que verifique que el token sea válido. Retornar un JSON con el mensaje de error correspondiente (y status 409), en caso de no ser válido.
Caso contrario, pasar al siguiente callable.

Agregar un Middleware (clase MW - método de clase) que verifique si es un "propietario" o no.
Si no lo es, retornar un JSON con el mensaje de error correspondiente (y status 409).

Agregar un Middleware (clase MW - método de instancia) que verifique si es un "encargado" o no.
Si no lo es, retornar un JSON con el mensaje de error correspondiente (y status 409).

Aplicar los Middlewares a los anteriores verbos.

Parte 03

Crear y aplicar Middlewares al grupo **/listados** que cumplan con las siguientes funcionalidades.

A partir del método que retorna el listado de los anteojos (NO hacer nuevos métodos!):

- 1.- Se necesita que, si el que accede al listado es un encargado, retorne todos los datos, menos el ID. (clase MW - método de instancia)
- 2.- Si es un encargado, muestre la cantidad de colores (distintos) que se tiene. (clase MW - método de instancia)
- 3.- Si es el propietario, muestre: todos o el antejo cuyo ID fue pasado como parámetro. (clase MW - método de clase).

A partir del método que retorna el listado de ventas (NO hacer nuevos métodos!):

- 1.- Si es empleado, que muestre el listado de las ventas. (clase MW - método de clase)
- 2.- Si es propietario, que muestre el listado de las ventas de la fecha dada. (clase MW - método de instancia)
- 3.- Si es empleado o encargado, que muestre el monto total de las ventas. (clase MW - método de clase)