

Parte 02

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.
Utilizar Slim framework para la creación de la Api Rest

En la creación del JWT, establecerle un valor de expiración de **30 segundos**.

Crear, a **nivel de aplicación**, los verbos:

(DELETE) Borrado de autos por ID.

Recibe el ID del auto a ser borrado más el **JWT**. Si el perfil es '**propietario**' se borrará de la base de datos. Caso contrario, se mostrará el mensaje correspondiente (indicando que usuario intentó realizar la acción).

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

(PUT) Modificar los autos por ID.

Recibe el JSON del auto a ser modificado más el **JWT**. Si el perfil es '**propietario**' o '**encargado**' se modificará de la base de datos. Caso contrario, se mostrará el mensaje correspondiente (indicando que usuario intentó realizar la acción).

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

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

1.- (método de instancia) verifique que el **token** sea válido.

Recibe el JWT a ser verificado.

Retorna un JSON con el mensaje de error correspondiente (y status 403), en caso de no ser válido.

Caso contrario, pasar al siguiente callable.

2.- (método de clase) verifique si es un '**propietario**' o no.

Recibe el JWT a ser verificado.

Retorna un JSON con propietario: true/false; mensaje: string (mensaje correspondiente); status: 200/409.

3.- (método de instancia) verifique si es un '**encargado**' o no.

Recibe el JWT a ser verificado.

Retorna un JSON con encargado: true/false; mensaje: string (mensaje correspondiente); status: 200/409.

Aplicar los Middlewares a los verbos PUT y DELETE.