



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

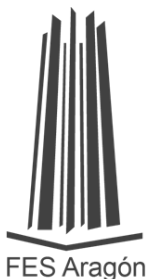
## INGENIERÍA EN COMPUTACIÓN INGENIERIA DE SOFTWARE 2 PROYECTO FINAL

PROFESOR:

JESUS HERNANDEZ CABRERA

INTEGRANTES:

- Badillo Mendoza Jaime Yair
- Galindo Rosales Erik Eduardo
- González Corona Adrián
- González Valadez Javier
- Martínez Bautista Luis Ángel
- Rosales Lázaro José Eduardo
- Sánchez Vera Erick



GRUPO: 2008

FECHA DE ENTREGA: 08/06/2022

# Documentación Proyecto Final

## Introducción

En el presente proyecto se desarrollará una aplicación por medio de Nodejs, MongoDB y Docker. La aplicación constará de dos secciones: la primera será un contenido estático, con conceptos de Docker e información sobre el equipo de desarrollo, todo ello empleando la máquina de render Handle Bars y estilos CSS. La segunda parte será una API REST con el tema de automóviles, en donde se implementarán los métodos GET, POST, PATCH, PUT y DELETE. Además, el proyecto se realizará mediante la metodología ágil SCRUM.

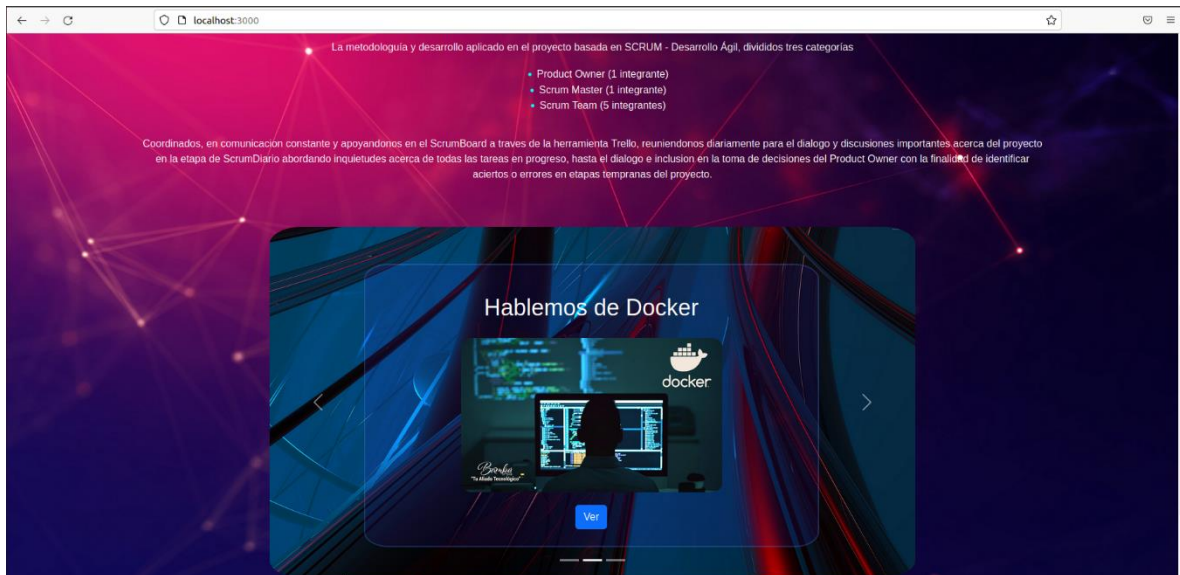
## Desarrollo

En este documento no se profundizará en cuanto a la programación del proyecto, para ello el código se encuentra en el siguiente repositorio: [https://github.com/Javier-Gonzalez-Valadez/IS2\\_PROYECTO\\_TESLAS.git](https://github.com/Javier-Gonzalez-Valadez/IS2_PROYECTO_TESLAS.git)

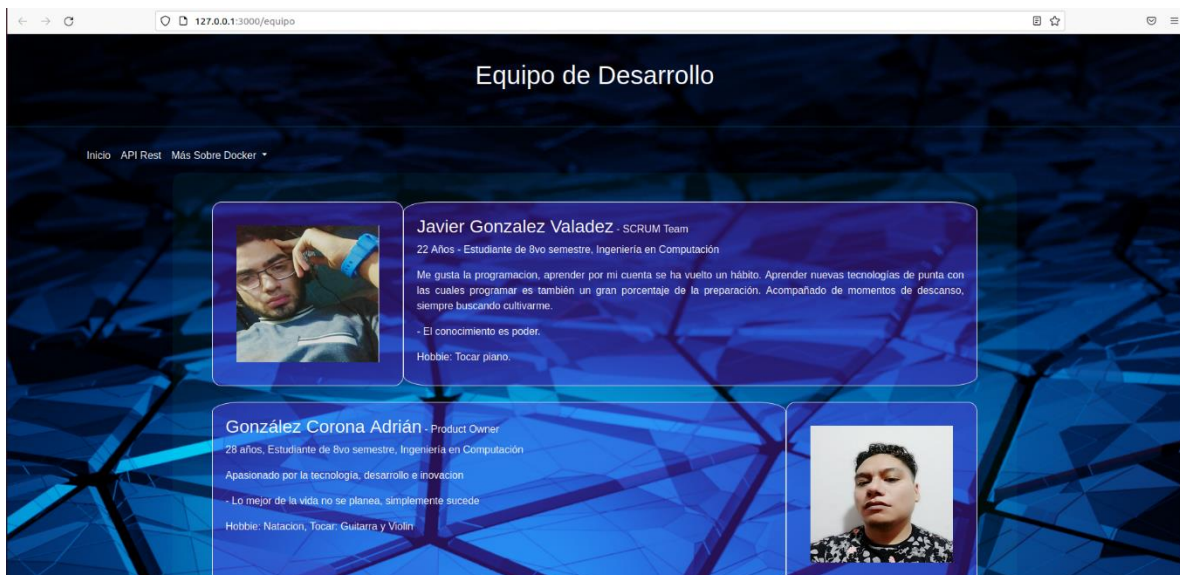
La parte estática consta de las siguientes secciones:

- **Pagina principal:** en donde se da una breve descripción de cada una de los temas a tratar, con sus respectivos redireccionamientos.





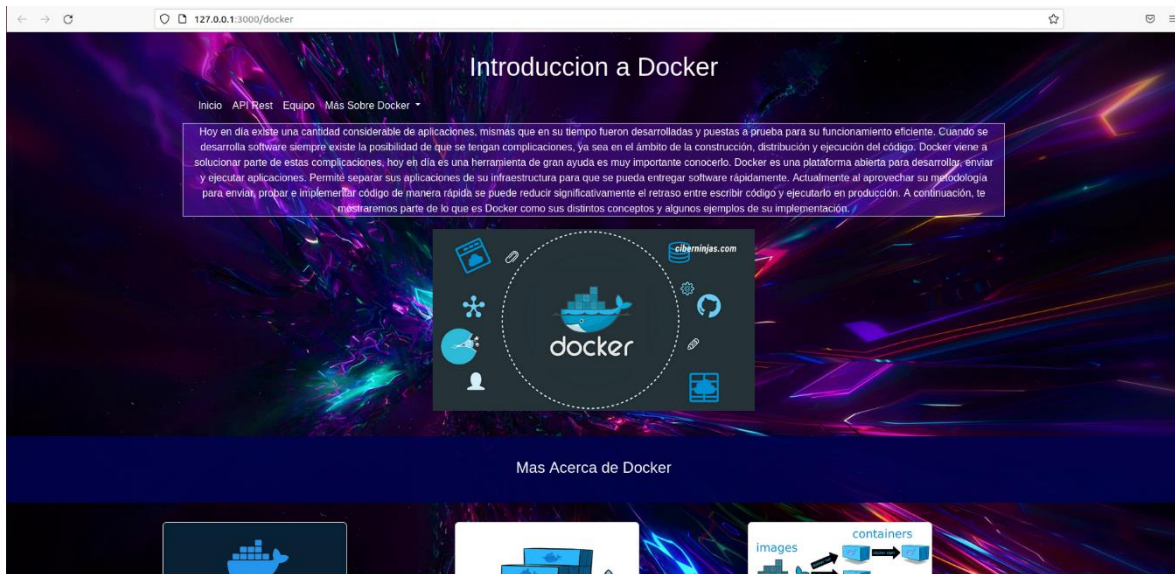
- **Pagina con información sobre el equipo de desarrollo:** Aquí se muestran los integrantes que conformaron el equipo de desarrollo, con información acerca de cada uno.



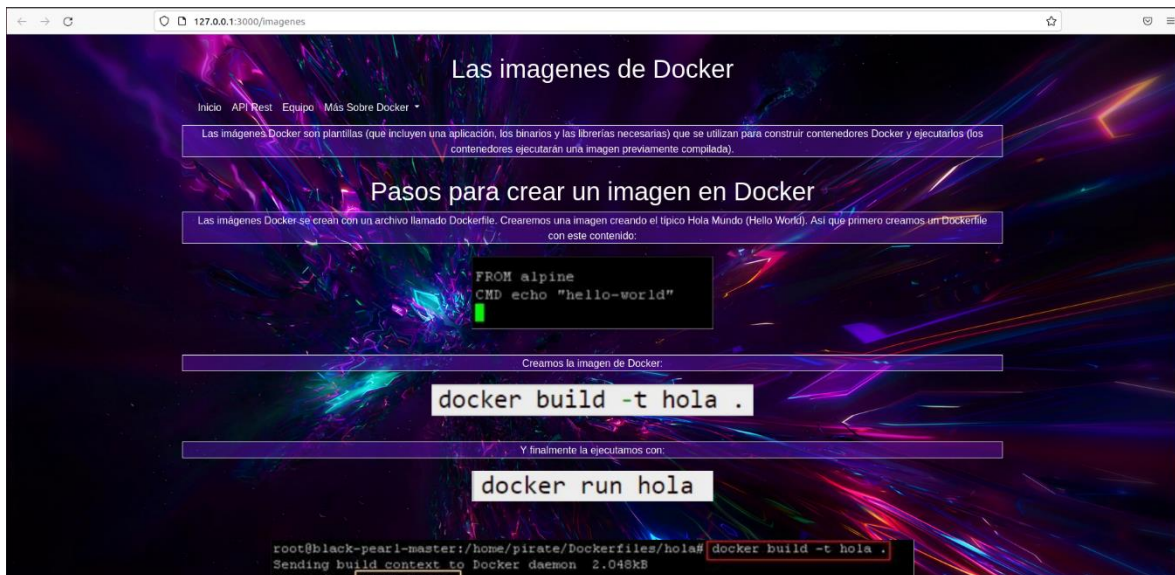
**Conceptos de Docker:** En este apartado se recabo información fundamental acerca de lo que es Docker. Se crearon paginas con los siguientes conceptos:



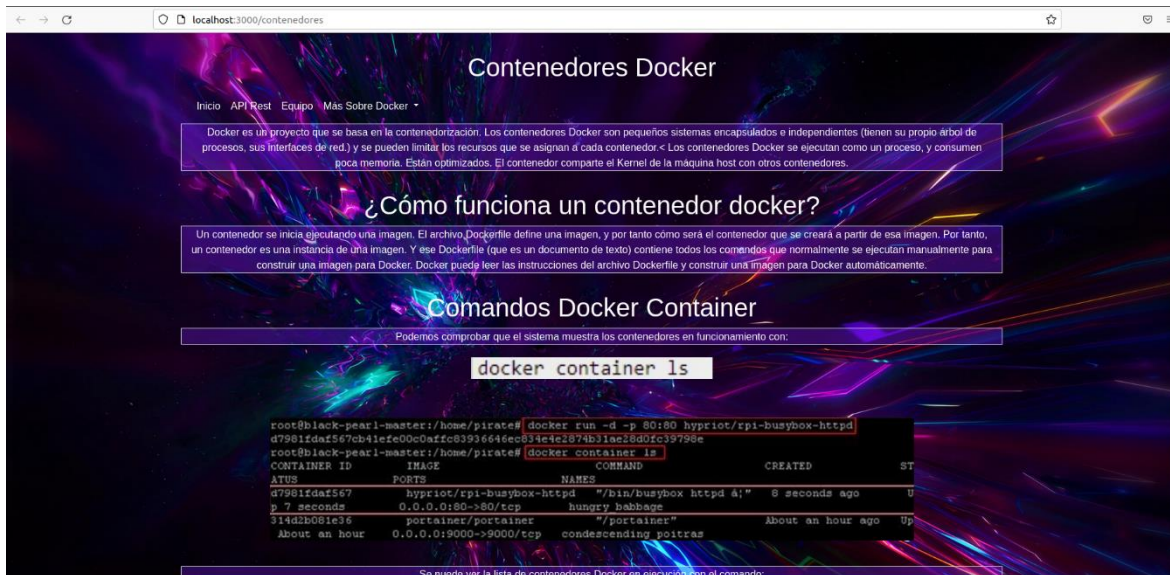
- Introducción a Docker



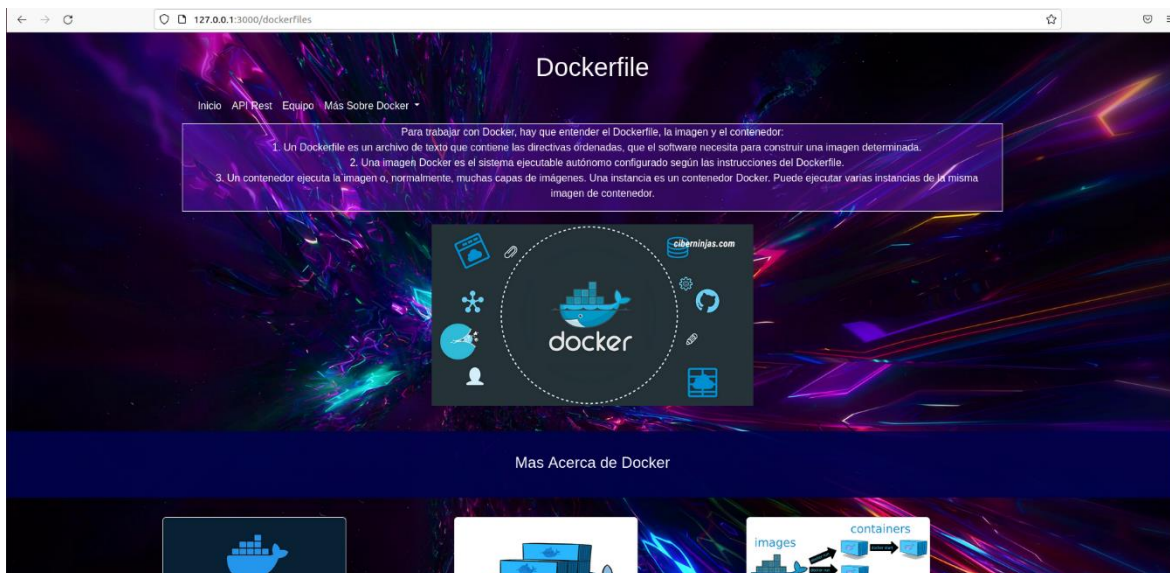
- Imágenes Docker



- Contenedores

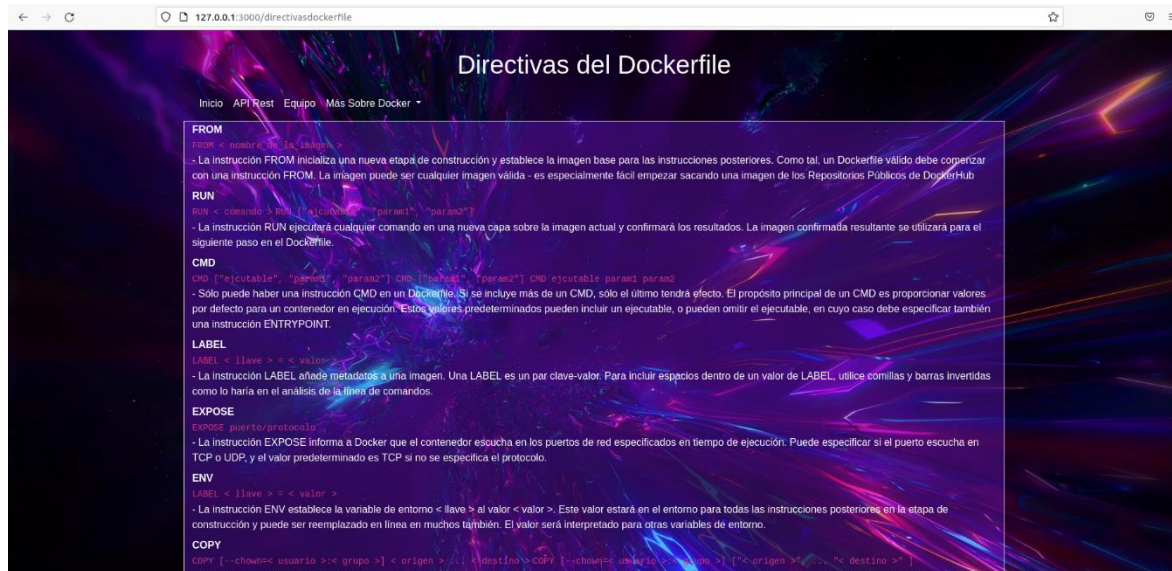


- Archivo Dockerfile

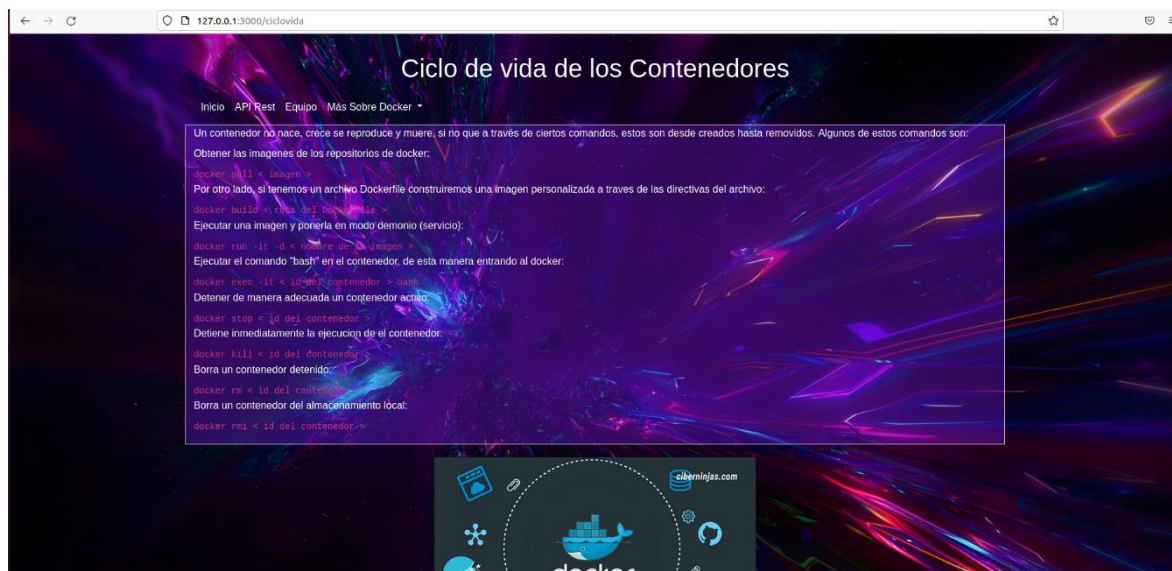




- Directivas Dockerfile



- Ciclo de vida de los contenedores.



Cada una de ellas cuenta con redireccionamiento entre ellas para una mejor navegación.

En el apartado de la API REST el tema asignado fue automóviles, aquí se realizó la base de datos y se implementarán los métodos GET, POST, PATCH, PUT y DELETE.

Los métodos funcionan de la siguiente manera:

- **Método GET:** Por medio de este método podemos obtener los recursos o información del servidor.

#### Estructura de consulta:

- GET servidor/directorio/
- GET servidor/directorio/archivo (ID).

Prueba del método GET:

Se traerá toda la información que contiene el directorio autos para ello, se accede mediante la siguiente ruta: GET 127.0.0.1:3000/autos.

GET	▼	127.0.0.1:3000/autos
-----	---	----------------------

Obteniendo el siguiente resultado:

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   {
3     "_id": "629842203b8b7a9071884ed6",
4     "id": 1,
5     "marca": "BMW",
6     "modelo": 2022,
7     "tipoDeDiseno": "i9",
8     "tipoMotor": "Diesel",
9     "numeroPlazas": "4",
10    "__v": 0
11  },
12  {
13    "_id": "6298430f3b8b7a9071884edd",
14    "id": 2,
15    "marca": "BMW",
16    "modelo": 2022,
17    "tipoDeDiseno": "ISVB",
18    "tipoMotor": "Hibrido",
19    "numeroPlazas": "5",
20    "__v": 0
21  },
22 }
```

127.0.0.1:3000/autos

GET 127.0.0.1:3000/autos

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 96 ms Size: 1.67 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "_id": "629842203b8b7a9071884ed6",
4     "id": 1,
5     "marca": "BMW",
6     "modelo": 2022,
7     "tipoDeDiseno": "i9",
8     "tipoMotor": "Diesel",
9     "numeroPlazas": "4",
10    "__v": 0
11  },
12  {
13    "_id": "6298430f3b8b7a9071884edd",
14    "id": 2,
15    "marca": "BMW",
16    "modelo": 2022,
17    "tipoDeDiseno": "ISVB",
18    "tipoMotor": "Hibrido",
19    "numeroPlazas": "5",
20    "__v": 0
21  },
22 }
```

Para consultar solo información en particular se agrega el ID: como se muestra a continuación:

Get 127.0.0.1:3000/autos/2



GET

127.0.0.1:3000/autos/2

En la salida muestra solo la información del ID seleccionado.

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "6298430f3b8b7a9071884edd",
3   "id": 2,
4   "marca": "BMW",
5   "modelo": 2022,
6   "tipoDeDiseno": "ISVB",
7   "tipoMotor": "Hibrido",
8   "numeroPlazas": "5",
9   "__v": 0
10 }
```

Prueba en página completa

127.0.0.1:3000/autos/2

GET 127.0.0.1:3000/autos/2

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "6298430f3b8b7a9071884edd",
3   "id": 2,
4   "marca": "BMW",
5   "modelo": 2022,
6   "tipoDeDiseno": "ISVB",
7   "tipoMotor": "Hibrido",
8   "numeroPlazas": "5",
9   "__v": 0
10 }
```

Otras consultas realizadas:

GET 127.0.0.7:3000/autos/5

GET



127.0.0.1:3000/autos/5

Resultado:

Body Cookies Headers (7) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {
2   "_id": "6298446d3b8b7a9071884ee9",
3   "id": 5,
4   "marca": "Aston Martin",
5   "modelo": 2021,
6   "tipoDeDiseno": "DBS Superleggera",
7   "tipoMotor": "Diesel",
8   "numeroPlazas": "2",
9   "__v": 0
10 }
```

Overview GET 127.0.0.1:3000/autos/5 + ... No Environment

127.0.0.1:3000/autos/5 Save

GET 127.0.0.1:3000/autos/5 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 65 ms Size: 396 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "6298446d3b8b7a9071884ee9",
3   "id": 5,
4   "marca": "Aston Martin",
5   "modelo": 2021,
6   "tipoDeDiseno": "DBS Superleggera",
7   "tipoMotor": "Diesel",
8   "numeroPlazas": "2",
9   "__v": 0
10 }
```

GET 127.0.0.7:3000/autos/10

GET	127.0.0.1:3000/autos/10
-----	-------------------------

Resultado:

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON 

```
1  {
2      "_id": "629846723b8b7a9071884ef4",
3      "id": 10,
4      "marca": "McLaren",
5      "modelo": 2021,
6      "tipoDeDiseno": "600LT",
7      "tipoMotor": "Diesel",
8      "numeroPlazas": "2",
9      "__v": 0
10 }
```

Save

Send

## Cookies

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Status: 200 OK Time: 62 ms Size: 381 B Save Response

Pretty Raw Preview Visualize JSON 

```
1 1
2 "id": "6298467238b7a9071884ef4",
3 "id": 10,
4 "marca": "McLaren",
5 "modelo": 2021,
6 "tipoDeOiseneno": "680LT",
7 "tipoMotor": "Diesel",
8 "numeroPlazas": "2",
9 "v": 0
10 2
```

- **Método POST:** Por medio de este método podemos insertar los recursos o información al servidor.

### Estructura de consulta:



- POST servidor/directorio/, este es el directorio que almacenara nuevos datos.

Prueba del método POST:

Seleccionamos la ruta en la que queremos insertar los datos:

GET 127.0.0.1:3000/autos.

POST

127.0.0.1:3000/autos

Por medio de un formulario realizaremos la inserción:

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	id	12
<input checked="" type="checkbox"/>	marca	Porsche
<input checked="" type="checkbox"/>	modelo	2021
<input checked="" type="checkbox"/>	tipoDeDiseno	911 GT3
<input checked="" type="checkbox"/>	tipoMotor	Diesel
<input checked="" type="checkbox"/>	numeroPlazas	2

Al dar send muestra la inserción:

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1
2   "id": 12,
3   "marca": "Porsche",
4   "modelo": 2021,
5   "tipoDeDiseno": "911 GT3",
6   "tipoMotor": "Diesel",
7   "numeroPlazas": "2",
8   "_id": "629cfed0898aa5bda400e34f",
9   "__v": 0
10
```

Comprobamos por medio de un GET y verificamos que si agregó los datos.

```
Cookies Headers (7) Test Results
ty Raw Preview Visualize JSON
{
  "tipoMotor": "Diesel",
  "numeroPlazas": "2",
  "__v": 0
},
{
  "_id": "629cfed0898aa5bda400e34f",
  "id": 12,
  "marca": "Porsche",
  "modelo": 2021,
  "tipoDeDiseno": "911 GT3",
  "tipoMotor": "Diesel",
  "numeroPlazas": "2",
  "__v": 0
}
```

Ejemplo dos:

Definimos los valores a insertar:

POST 127.0.0.1:3000/autos

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none
 ☐ form-data
 ☒ x-www-form-urlencoded
 ☐ raw
 ☐ binary
 ☐ GraphQL

	KEY	VALUE
<input checked="" type="checkbox"/>	id	11
<input checked="" type="checkbox"/>	marca	Tesla Motors
<input checked="" type="checkbox"/>	modelo	2022
<input checked="" type="checkbox"/>	tipoDeDiseno	Modelo S
<input checked="" type="checkbox"/>	tipoMotor	Electrico
<input checked="" type="checkbox"/>	numeroPlazas	4

Se insertan los valores correctamente:

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "id": 11,
3    "marca": "Tesla Motors",
4    "modelo": 2022,
5    "tipoDeDiseno": "Modelo S",
6    "tipoMotor": "Electrico",
7    "numeroPlazas": "4",
8    "_id": "629cff98898aa5bda400e352",
9    "__v": 0
10 }
  
```



POST 127.0.0.1:3000/autos Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	11	
<input checked="" type="checkbox"/> marca	Tesla Motors	
<input checked="" type="checkbox"/> modelo	2022	
<input checked="" type="checkbox"/> tipoDeDiseno	Modelo S	
<input checked="" type="checkbox"/> tipoMotor	Electrico	
<input checked="" type="checkbox"/> numeroPlazas	4	
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 72 ms Size: 392 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "id": 11,
3    "marca": "Tesla Motors",
4    "modelo": 2022,
5    "tipoDeDiseno": "Modelo S",
6    "tipoMotor": "Electrico",
7    "numeroPlazas": "4",
8    "_id": "629c9ff98898aa5bda40e352",
9    "__v": 0
10 }

```

## Verificación:

GET 127.0.0.1:3000/autos Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (7) Test Results Status: 200 OK Time: 66 ms Size: 1.72 KB Save Response

Pretty Raw Preview Visualize JSON

```

85  {
86    {
87      "_id": "629c9ff98898aa5bda40e34f",
88      "id": 12,
89      "marca": "Porsche",
90      "modelo": 2021,
91      "tipoDeDiseno": "911 GT3",
92      "tipoMotor": "Diesel",
93      "numeroPlazas": "2",
94      "__v": 0
95    },
96    {
97      "_id": "629c9ff98898aa5bda40e352",
98      "id": 11,
99      "marca": "Tesla Motors",
100     "modelo": 2022,
101     "tipoDeDiseno": "Modelo S",
102     "tipoMotor": "Electrico",
103     "numeroPlazas": "4",
104     "__v": 0
105   }
106 }

```

- **Método PATCH:** Por medio de este método podemos actualizar los recursos o información del servidor.

## Estructura de consulta:

- PATCH, servidor/directorio/archivo (ID). Para actualizar un archivo.

## Prueba del método PATCH:

Primero se obtiene el contenido del directorio autos:

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 [
2   {
3     "_id": "629842203b8b7a9071884ed6",
4     "id": 1,
5     "marca": "BMW",
6     "modelo": 2022,
7     "tipoDeDiseno": "i9",
8     "tipoMotor": "Diesel",
9     "numeroPlazas": "5",
10    "__v": 0
11  },
12  {
13    "_id": "6298438b3b8b7a9071884ee5",
14    "id": 3,
15    "marca": "BMW",
16    "modelo": 2022,
17    "tipoDeDiseno": "z4",
18    "tipoMotor": "Diesel",
19    "numeroPlazas": "2",
20    "__v": 0
21  },
22  ]
```

Para realizar la actualización usamos: PATCH 127.0.0.1:3000/autos/1

PATCH 127.0.0.1:3000/autos/1

Para modificar los datos es necesario hacerlo mediante el formato Json como se muestra a continuación:

```
Params Authorization Headers (8) Body ● Pre-request Script Tests Settings
● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON
1 .....
2 ..... "_id": "629842203b8b7a9071884ed6",
3 ..... "id": 1,
4 ..... "marca": "BMW",
5 ..... "modelo": 2023,
6 ..... "tipoDeDiseno": "i9s",
7 ..... "tipoMotor": "Diesel",
8 ..... "numeroPlazas": "4",
9 ..... "__v": 0
10 .....
```

Al registrarlo nos manda una notificación de actualización exitosa.

```
1  ...
2  ... "_id": "629842203b8b7a9071884ed6",
3  ... "id": 1,
4  ... "marca": "BMW",
5  ... "modelo": 2023,
6  ... "tipoDeDiseno": "i9s",
7  ... "tipoMotor": "Diesel",
8  ... "numeroPlazas": "4",
9  ... "__v": 0
10 ... }
```

Body Cookies Headers (7) Test Results

Pretty

Raw

Preview

Visualize

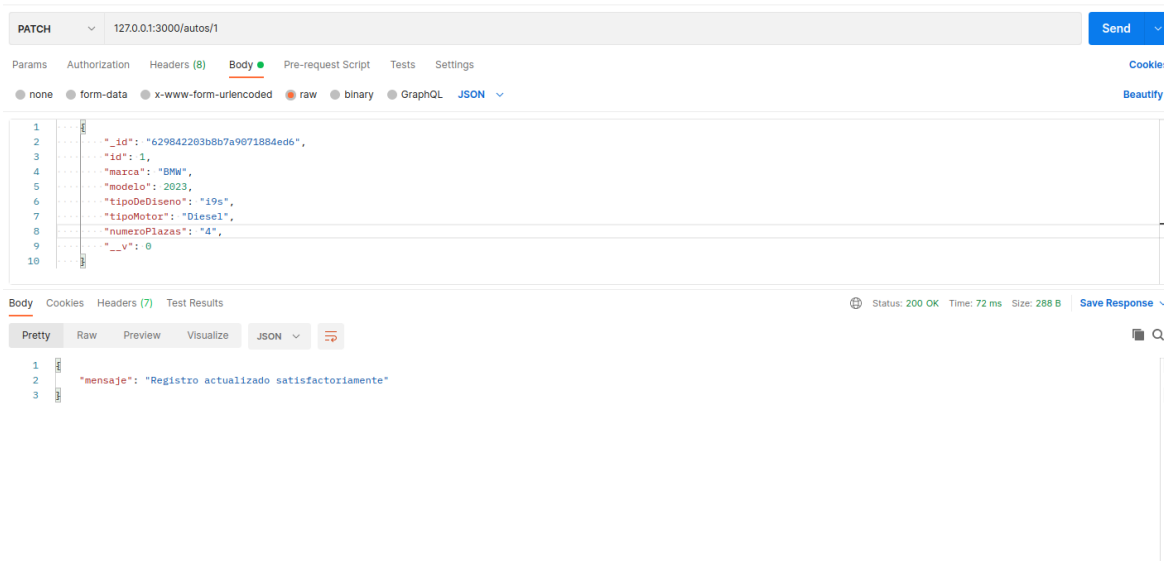
JSON

⌵



```
1  {
2  }
3  }
```

"mensaje": "Registro actualizado satisfactoriamente"



Comprobamos que su actualización fue correcta, para ello se ocupó de nueva cuenta el método GET y se obtiene que es correcta:



Body Cookies Headers (7) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 [
2   {
3     "_id": "629842203b8b7a9071884ed6",
4     "id": 1,
5     "marca": "BMW",
6     "modelo": 2023,
7     "tipoDeDiseno": "i9s",
8     "tipoMotor": "Diesel",
9     "numeroPlazas": "4",
10    "__v": 0
11  },
12  {
13    "_id": "6298438b3b8b7a9071884ee5",
14    "id": 3,
15    "marca": "BMW",
16    "modelo": 2022,
17    "tipoDeDiseno": "z4",
18    "tipoMotor": "Diesel",
19    "numeroPlazas": "2",
20    "__v": 0
21  },
22 ]
```

GET 127.0.0.1:3000/autos/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 67 ms Size: 1.41 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "_id": "629842203b8b7a9071884ed6",
3   "id": 1,
4   "marca": "BMW",
5   "modelo": 2023,
6   "tipoDeDiseno": "i9s",
7   "tipoMotor": "Diesel",
8   "numeroPlazas": "4",
9   "__v": 0
10 },
11 {
12   "_id": "6298438b3b8b7a9071884ee5",
13   "id": 3,
14   "marca": "BMW",
15   "modelo": 2022,
16   "tipoDeDiseno": "z4",
17   "tipoMotor": "Diesel",
18   "numeroPlazas": "2",
19   "__v": 0
20 },
21 ]
```

Prueba dos, método PATCH:

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
37 "tipoDeDiseno": "CT6",
38 "tipoMotor": "Diesel",
39 "numeroPlazas": "4",
40 "__v": 0
41 },
42 {
43   "_id": "629845703b8b7a9071884eee",
44   "id": 7,
45   "marca": "Jaguar",
46   "modelo": 2015,
47   "tipoDeDiseno": "XF",
48   "tipoMotor": "Diesel",
49   "numeroPlazas": "4",
50   "__v": 0
51 },
52 {
53   "_id": "629845d43b8b7a9071884ef0",
54   "id": 8,
55   "marca": "Land Rover",
56   "modelo": 2022,
57   "tipoDeDiseno": "Discovery Sport",
58   "tipoMotor": "Diesel"
```

Actualización: PACH 127.0.0.1:3000/autos/7

PATCH	127.0.0.1:3000/autos/7
-------	------------------------

Actualización exitosa:

```
1 {
2   "_id": "629845703b8b7a9071884eee",
3   "id": 7,
4   "marca": "Jaguar Super Sport",
5   "modelo": 2018,
6   "tipoDeDiseno": "XF",
7   "tipoMotor": "Diesel",
8   "numeroPlazas": "4",
9   "__v": 0
10 }
```

Body Cookies Headers (7) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "mensaje": "Registro actualizado satisfactoriamente"  
3 }
```

Resultado:

Body Cookies Headers (7) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
37   "tipoDeDiseno": "CT6",  
38   "tipoMotor": "Diesel",  
39   "numeroPlazas": "4",  
40   "__v": 0  
41 },  
42 {  
43   "_id": "629845703b8b7a9071884eee",  
44   "id": 7,  
45   "marca": "Jaguar Super Sport",  
46   "modelo": 2018,  
47   "tipoDeDiseno": "XF",  
48   "tipoMotor": "Diesel",  
49   "numeroPlazas": "4",  
50   "__v": 0  
51 },  
52 {  
53   "_id": "629845d43b8b7a9071884ef0",  
54   "id": 8,  
55   "marca": "Land Rover",  
56   "modelo": 2022,  
57   "tipoDeDiseno": "Discovery Sport",  
58   "tipoMotor": "Diesel",
```



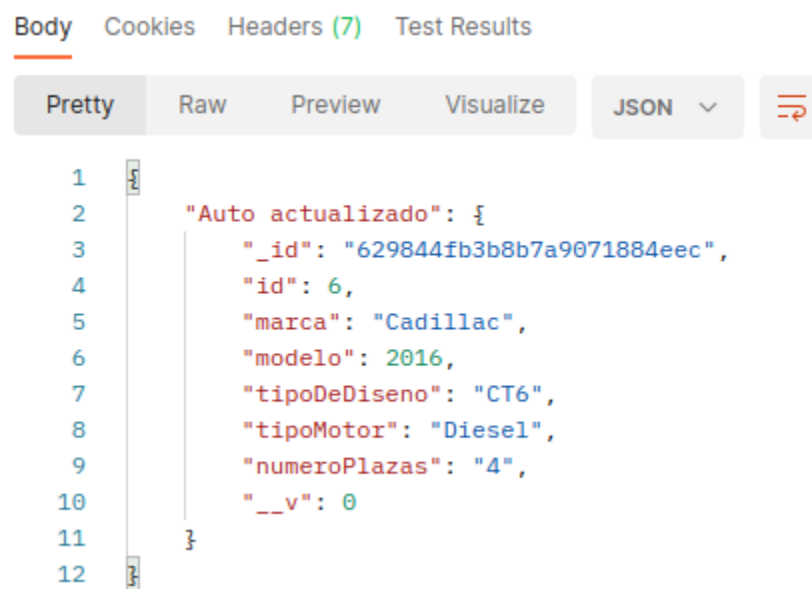
- **Método PUT:** Por medio de este método podemos modificar parcial o totalmente los recursos o información del servidor.

### Estructura de consulta:

- PUT, servidor/directorio/archivo (HASH). Para actualizar un archivo.

Prueba del método PUT:

Primero se obtiene el contenido del directorio autos:



```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "Auto actualizado": {
3     "_id": "629844fb3b8b7a9071884eec",
4     "id": 6,
5     "marca": "Cadillac",
6     "modelo": 2016,
7     "tipoDeDiseno": "CT6",
8     "tipoMotor": "Diesel",
9     "numeroPlazas": "4",
10    "__v": 0
11  }
12 }
```

Para realizar la modificación:

PUT 127.0.0.1:3000/autos/629844fb3b8b7a9071884eec

PUT	127.0.0.1:3000/autos/629844fb3b8b7a9071884eec
-----	---

Introducimos los datos a modificar en formato Json.

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL JSON ▾

```
1  {
2    "_id": "629844fb3b8b7a9071884eec",
3    "id": 6,
4    "marca": "McLaren",
5    "modelo": 2018,
6    "tipoDeDiseno": "Senna",
7    "tipoMotor": "Diesel",
8    "numeroPlazas": 2,
9    "__v": 0
10 }
```

Obtenemos el siguiente resultado

Body Cookies Headers (7) Test Results

Pretty

Raw

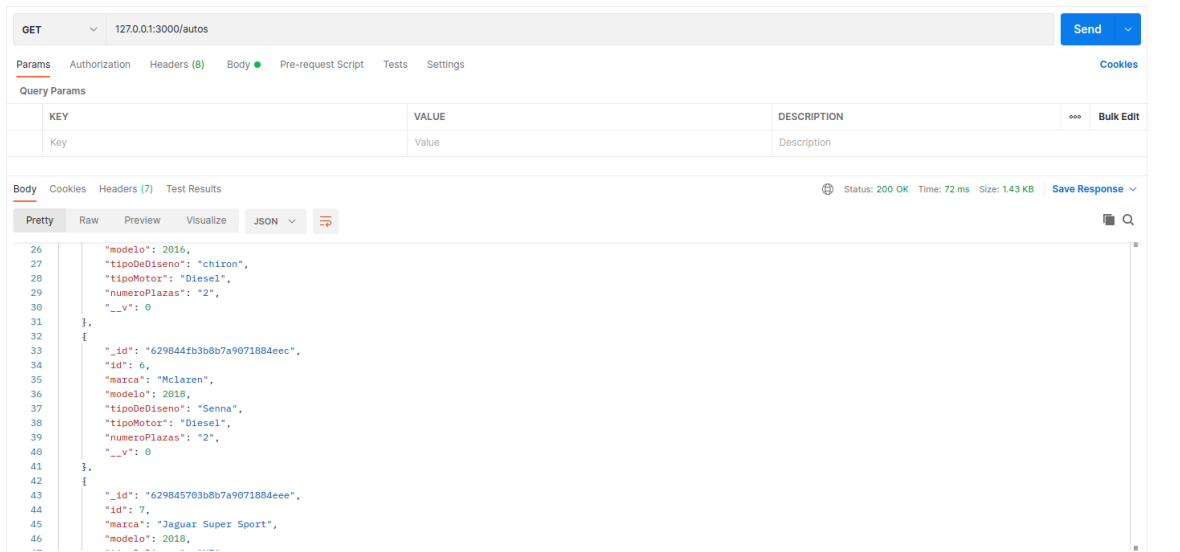
Preview

Visualize

JSON ▾



```
26    "modelo": 2016,
27    "tipoDeDiseno": "chiron",
28    "tipoMotor": "Diesel",
29    "numeroPlazas": 2,
30    "__v": 0
31  },
32  {
33    "_id": "629844fb3b8b7a9071884eec",
34    "id": 6,
35    "marca": "McLaren",
36    "modelo": 2018,
37    "tipoDeDiseno": "Senna",
38    "tipoMotor": "Diesel",
39    "numeroPlazas": 2,
40    "__v": 0
41  },
42  {
43    "_id": "629845703b8b7a9071884eee",
44    "id": 7,
45    "marca": "Jaguar Super Sport",
46    "modelo": 2018,
```



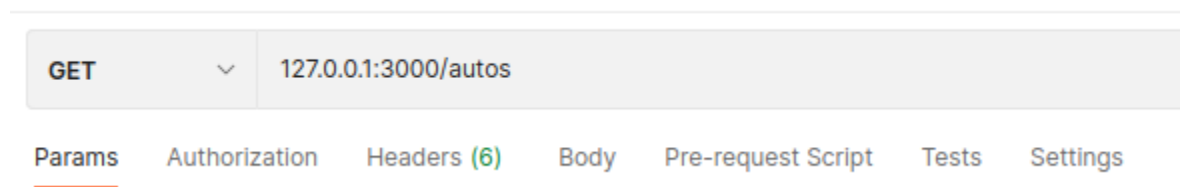
- **Método DELETE:** Por medio de este método podemos eliminar los recursos o información del servidor.

### Estructura de consulta:

- DELETE servidor/directorio/archivo (ID). Para eliminar un archivo.

### Prueba del método DELETE:

Primero se obtiene el contenido del directorio autos:



Comprobamos que contiene archivos:

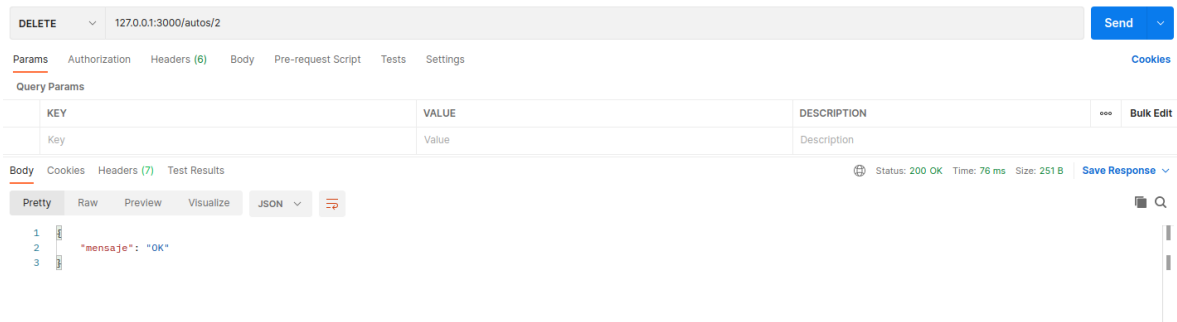
```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 [
2   {
3     "_id": "629842203b8b7a9071884ed6",
4     "id": 1,
5     "marca": "BMW",
6     "modelo": 2022,
7     "tipoDeDiseno": "i9",
8     "tipoMotor": "Diesel",
9     "numeroPlazas": "4",
10    "__v": 0
11  },
12  {
13    "_id": "6298430f3b8b7a9071884edd",
14    "id": 2,
15    "marca": "BMW",
16    "modelo": 2022,
17    "tipoDeDiseno": "ISVB",
18    "tipoMotor": "Hibrido",
19    "numeroPlazas": "5",
20    "__v": 0
21  },
```

Se eliminará la información con el ID dos, el cual se encuentra en el directorio autos, se realiza lo siguiente: DELETE 127.0.0.1:3000/autos/2.

```
DELETE 127.0.0.1:3000/autos/2
```

Posteriormente nos da un mensaje de “OK” con lo cual comprobamos que la eliminación fue completada.

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "mensaje": "OK"
3 }
```



Volvemos a consultar el directorio autos, podemos observar que ya no contiene la información con el ID “2”.





GET 127.0.0.1:3000/autos/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 63 ms Size: 1.57 KB Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "_id": "629842203b8b7a9071884ee6",
3   "id": 1,
4   "marca": "BMW",
5   "modelo": 2022,
6   "tipoDeDiseno": "19",
7   "tipoMotor": "Diesel",
8   "numeroPlazas": "4",
9   "___v": 0
10 }
11
12 {
13   "_id": "6298438b3b8b7a9071884ee5",
14   "id": 3,
15   "marca": "BMW",
16   "modelo": 2022,
17   "tipoDeDiseno": "24",
18   "tipoMotor": "Diesel",
19   "numeroPlazas": "2",
20   "___v": 0
21 }
22
23 {
24   "_id": "629843f43b8b7a9071884ee7",
25   "id": 4,
  
```

Eliminaciones realizadas:

DELETE 127.0.1:3000/autos/5

DELETE 127.0.0.1:3000/autos/5

Notificación de realizado:

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```

1 {
2   "mensaje": "OK"
3 }
  
```

## Comprobación:

```
Body Cookies Headers (7) Test Results
Pretty Raw Preview Visualize JSON ↕

20  __v: 0
21  },
22  {
23    "_id": "629843f43b8b7a9071884ee7",
24    "id": 4,
25    "marca": "Bugatti",
26    "modelo": 2016,
27    "tipoDeDiseno": "chiron",
28    "tipoMotor": "Diesel",
29    "numeroPlazas": "2",
30    "__v": 0
31  },
32  {
33    "_id": "629844fb3b8b7a9071884eec",
34    "id": 6,
35    "marca": "Cadillac",
36    "modelo": 2016,
37    "tipoDeDiseno": "CT6",
38    "tipoMotor": "Diesel",
39    "numeroPlazas": "4",
40    "__v": 0
41  },
42  {
```

OET 127.0.0.1:3000/autos/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (7) Test Results Status: 200 OK Time: 66 ms Size: 1.41 KB Save Response

Pretty Raw Preview Visualize JSON ↕

```
20  __v: 0
21  },
22  {
23    "_id": "629843f43b8b7a9071884ee7",
24    "id": 4,
25    "marca": "Bugatti",
26    "modelo": 2016,
27    "tipoDeDiseno": "chiron",
28    "tipoMotor": "Diesel",
29    "numeroPlazas": "2",
30    "__v": 0
31  },
32  {
33    "_id": "629844fb3b8b7a9071884eec",
34    "id": 6,
35    "marca": "Cadillac",
36    "modelo": 2016,
37    "tipoDeDiseno": "CT6",
38    "tipoMotor": "Diesel",
39    "numeroPlazas": "4",
40    "__v": 0
41  },
42  {
43    "_id": "629845703b8b7a9071884eee",
44    "id": 7
```