
A large, dark blue ink splash or blotch is centered on a white background. The splash has irregular, organic edges with some lighter blue and white speckling around it, suggesting a liquid spill or a paint splatter. The text 'Criação de APIs' is written in white, sans-serif font across the middle of the dark blue area.

Criação de APIs

Alterando um carro

Vamos criar a rota para alterar um carro, no arquivo routes, faça:

A função alterar será criada no controller.

```
src > JS routes.js > ...
1  const express = require('express');
2  const router = express.Router();
3
4  const CarroController = require('./controllers/CarroController');
5
6  router.get('/carros', CarroController.buscarTodos);
7  router.get('/carro/:codigo', CarroController.buscarUm);
8  router.post('/carro', CarroController.inserir);
9  router.put('/carro/:codigo', CarroController.alterar);
10
11 module.exports = router;
```

Alterando um carro

Vamos criar nossa rota de alterar

Vamos trabalhar com o metodo put.

Vamos começar criando o edit no CarroController.js

Ele é igual o inserir, porém também recebe o id.

```
52  alterar: async(req, res) => {  
53      let json = {error:'', result:{}};  
54  
55      let codigo = req.params.codigo;  
56      let modelo = req.body.modelo;  
57      let placa = req.body.placa;  
58  
59      if (codigo && modelo && placa){  
60          await CarroService.alterar(codigo, modelo, placa);  
61          json.result = {  
62              codigo,  
63              modelo,  
64              placa  
65          };  
66      }else{  
67          json.error = 'Campos não enviados';  
68      }  
69      res.json(json);  
70  }
```

Alterando um carro

Agora vamos criar o alterar no arquivo CarroService.js

```
38     },
39     alterar:(codigo, modelo, placa)=> {
40         return new Promise((aceito, rejeitado)=> {
41             db.query('UPDATE carros SET modelo = ?, placa = ? WHERE codigo = ?',
42                 [modelo, placa, codigo],
43                 (error, results) => {
44                     if(error){ rejeitado(error); return; }
45                     aceito(results);
46                 }
47             );
48         });
49     },
50 };

```

Alterando um carro

Agora vamos fazer o teste passando os parâmetros que queremos que sejam alterados e o código do carro que iremos alterar.

HTTP request options

Method Endpoint

Method and Endpoint are required. Click below to add additional parameters.

[Add authentication](#)

[+ Add header](#)

Parameter Name	Parameter Value	
<input type="text" value="modelo"/>	<input type="text" value="Honda Civic"/>	×
<input type="text" value="placa"/>	<input type="text" value="EMO4949"/>	×

HTTP 200 OK

```
{"error":"","result":{"codigo":"4","modelo":"Honda Civic","placa":"EMO4949"}}
```

Alterando um carro

Agora vamos verificar se a alteração foi realizada no banco de dados.

	codigo	modelo	placa
▶	4	Honda Civic	EMO4949
	5	Fusca2	gjg8778
	6	Honda Fit	GJG3563
✱	NULL	NULL	NULL