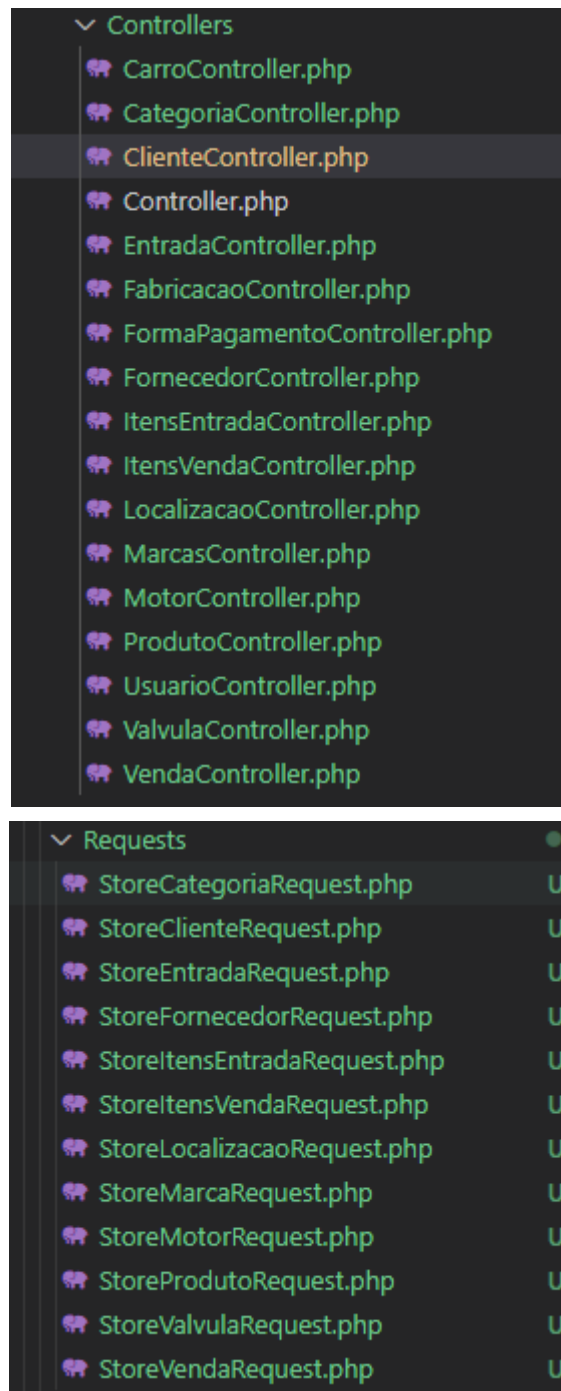


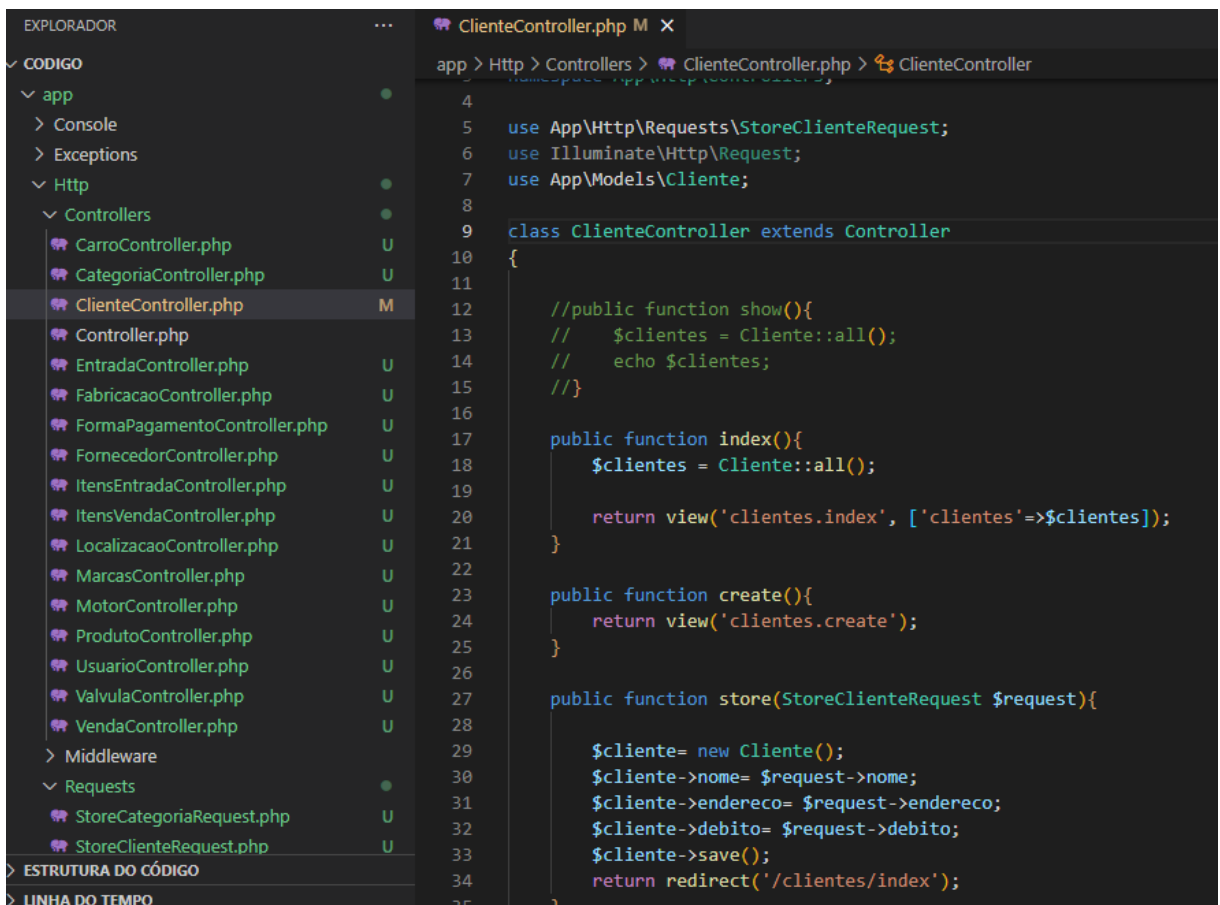
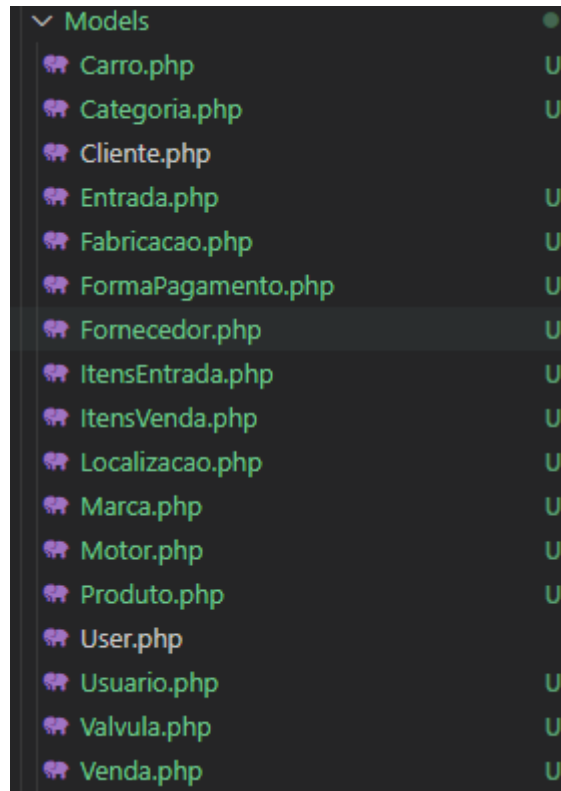
**Aluno:** Jefferson Batista Xavier

**Disciplina:** WEB II

**Professor:** Fábio Lima

### Controller, Request e Model





EXPLORADOR

CODIGO

- Controllers
  - CarroController.php U
  - CategoriaController.php U
  - ClienteController.php M
  - Controller.php
  - EntradaController.php U
  - FabricacaoController.php U
  - FormaPagamentoController.php U
  - FornecedorController.php U
  - ItensEntradaController.php U
  - ItensVendaController.php U
  - LocalizacaoController.php U
  - MarcasController.php U
  - MotorController.php U
  - ProdutoController.php U
  - UsuarioController.php U
  - ValvulaController.php U
  - VendaController.php U
- Middleware
- Requests
  - StoreCategoriaRequest.php U
  - StoreClienteRequest.php U
  - StoreEntradaRequest.php U
  - StoreFornecedorRequest.php U
  - StoreItensEntradaRequest.php U
  - StoreItensVendaRequest.php U

ESTRUTURA DO CÓDIGO

LINHA DO TEMPO

app > Http > Controllers > FornecedorController.php > ...

```
4 namespace App\Http\Controllers;
5 use Illuminate\Http\Request;
6 use App\Models\Fornecedor;
7 use App\Http\Requests\StoreFornecedorRequest;
8 class FornecedorController extends Controller
9 {
10     //
11     public function show(){
12         $fornecedores = Fornecedor::all();
13         echo $fornecedores;
14     }
15     public function index(){
16         $fornecedores = Fornecedor::all();
17         return view('fornecedores.index', ['fornecedores'=>$fornecedores]);
18     }
19     public function create(){
20         return view('fornecedores.create');
21     }
22
23     public function store(StoreFornecedorRequest $request){
24         $fornecedor= new Fornecedor();
25         $fornecedor->nome= $request->nome;
26         $fornecedor->cnpj= $request->cnpj;
27         $fornecedor->endereco= $request->endereco;
28         $fornecedor->telefone= $request->telefone;
29         $fornecedor->save();
30         return redirect('fornecedores.index');
31     }
32     public function edit($id){
33         $fornecedor = Fornecedor::findorFail($id);
34         return view('fornecedores.edit', ['fornecedor'=>$fornecedor]);
35     }
36 }
```

EXPLORADOR

CODIGO

- Controllers
  - EntradaController.php U
  - FabricacaoController.php U
  - FormaPagamentoController.php U
  - FornecedorController.php U
  - ItensEntradaController.php U
  - ItensVendaController.php U
  - LocalizacaoController.php U
  - MarcasController.php U
  - MotorController.php U
  - ProdutoController.php U
  - UsuarioController.php U
  - ValvulaController.php U
  - VendaController.php U
- Middleware
- Requests
  - StoreCategoriaRequest.php U
  - StoreClienteRequest.php U
  - StoreEntradaRequest.php U
  - StoreFornecedorRequest.php U
  - StoreItensEntradaRequest.php U
  - StoreItensVendaRequest.php U
  - StoreLocalizacaoRequest.php U
  - StoreMarcaRequest.php U
  - StoreMotorRequest.php U
  - StoreProdutoRequest.php U
  - StoreValvulaRequest.php U

ESTRUTURA DO CÓDIGO

app > Http > Requests > StoreClienteRequest.php > StoreClienteRequest > rules

```
14 /**
15  *
16  */
17 public function authorize()
18 {
19     return true;
20 }
21
22 /**
23  * Get the validation rules that apply to the request.
24  *
25  * @return array
26  */
27 public function rules()
28 {
29     return [
30         'nome' => 'required',
31         'cpf' => 'required',
32         'endereco' => 'required',
33         'debito' => 'required'
34     ];
35 }
36
37 public function messages(){
38     return [
39         'nome.required' => 'Preencha o campo Nome*',
40         'sexo.required' => 'Preencha o campo Sexo*',
41         'endereco.required' => 'Preencha o campo Endereço',
42         'debito.required' => 'Preencha o campo Débito*'
43     ];
44 }
```

EXPLORADOR

... StoreClienteRequest.php U StoreProdutoRequest.php U X

CODIGO

app > Http > Requests > StoreProdutoRequest.php > StoreProdutoRequest > messages

```
25 {
26     return [
27         'nome' => 'required',
28         'icms' => 'required',
29         'ipi' => 'required',
30         'frete' => 'required',
31         'precofabrica' => 'required',
32         'precocompra' => 'required',
33         'precovenda' => 'required',
34         'lucro' => 'required',
35         'desconto' => 'required',
36         'quantidade' => 'required'
37     ];
38 }
39
40 public function messages(){
41     return [
42         'nome.required' => 'Preencha o campo*',
43         'icms.required' => 'Preencha o campo*',
44         'ipi.required' => 'Preencha o campo*',
45         'frete.required' => 'Preencha o campo*',
46         'precofabrica.required' => 'Preencha o campo*',
47         'precocompra.required' => 'Preencha o campo*',
48         'precovenda.required' => 'Preencha o campo*',
49         'lucro.required' => 'Preencha o campo*',
50         'desconto.required' => 'Preencha o campo*',
51         'quantidade.required' => 'Preencha o campo*'
52     ];
53 }
```

ItensVendaController.php U 25

LocalizacaoController.php U 26

MarcasController.php U 27

MotorController.php U 28

ProdutoController.php U 29

UsuarioController.php U 30

ValvulaController.php U 31

VendaController.php U 32

> Middleware 33

Requests 34

StoreCategoriaRequest.php U 35

StoreClienteRequest.php U 36

StoreEntradaRequest.php U 37

StoreFornecedorRequest.php U 38

StoreItensEntradaRequest.php U 39

StoreItensVendaRequest.php U 40

StoreLocalizacaoRequest.php U 41

StoreMarcaRequest.php U 42

StoreMotorRequest.php U 43

StoreProdutoRequest.php U 44

StoreValvulaRequest.php U 45

StoreVendaRequest.php U 46

Kernel.php 47

Models 48

Carro.php U 49

50

51

52

53

54

EXPLORADOR ... StoreClienteRequest.php U StoreProdutoRequest.php U

CODIGO app > Models > Carro.php > ...

StoreLocalizacaoRequest.php	U	4
StoreMarcaRequest.php	U	5
StoreMotorRequest.php	U	6
StoreProdutoRequest.php	U	7
StoreValvulaRequest.php	U	8
StoreVendaRequest.php	U	9
Kernel.php	U	10
Models		11
Carro.php	U	12
Categoria.php	U	13
Cliente.php	U	14
Entrada.php	U	15
Fabricacao.php	U	16
FormaPagamento.php	U	17
Fornecedor.php	U	18
ItensEntrada.php	U	19
ItensVenda.php	U	20
Localizacao.php	U	21
Marca.php	U	22
Motor.php	U	23
Produto.php	U	24
User.php	U	25
Usuario.php	U	26
Valvula.php	U	27
Venda.php	U	28

```
4 {
5     private $idcarro;
6     private $modelo;
7
8     public function __construct($idcarro, $modelo)
9     {
10         $this->idcarro = $idcarro;
11         $this->modelo = $modelo;
12     }
13
14     public function getIdcarro()
15     {
16         return $this->idcarro;
17     }
18
19     public function setIdcarro($idcarro)
20     {
21         $this->idcarro = $idcarro;
22     }
23
24     public function getModelo()
25     {
26         return $this->modelo;
27     }
28
29     public function setModelo($modelo)
30     {
31         $this->modelo = $modelo;
32     }
33 }
```

> ESTRUTURA DO CÓDIGO

EXPLORADOR ... StoreClienteRequest.php U StoreProdutoRequest.php U Cliente.php X

CODIGO app > Models > Cliente.php > ...

StoreLocalizacaoRequest.php	U	1
StoreMarcaRequest.php	U	2
StoreMotorRequest.php	U	3
StoreProdutoRequest.php	U	4
StoreValvulaRequest.php	U	5
StoreVendaRequest.php	U	6
Kernel.php	U	7
Models		8
Carro.php	U	9
Categoria.php	U	10
Cliente.php	U	11
Entrada.php	U	12
Fabricacao.php	U	13
FormaPagamento.php	U	14
Fornecedor.php	U	15
ItensEntrada.php	U	16
ItensVenda.php	U	17
Localizacao.php	U	18
Marca.php	U	19

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Cliente extends Model
9 {
10     use HasFactory;
11     protected $table = "clientes";
12     protected $fillable = ['nome', 'endereço', 'debito'];
13
14     function contato(){
15         return $this->hasMany(Contato::class, 'id_fornecedor', 'id');
16     }
17     function venda(){
18         return $this->hasMany(Venda::class, 'id_fornecedor', 'id');
19     }
20 }
21
```

EXPLORADOR

... st.php U Categoria.php U Fabricacao.php U Produto.php U Localizacao.php

CODIGO

app > Models > Produto.php > Produto

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Produto extends Model
9 {
10     use HasFactory;
11
12     protected $table = "produtos";
13
14     protected $fillable = ['produto_id', 'icms', 'ipi', 'frete', 'precofabrica', ''];
15
16     function itensentrada(){
17         return $this->hasMany(ItensEntrada::class, 'produto_id', 'id');
18     }
19
20     function itensvenda(){
21         return $this->hasMany(ItensVenda::class, 'produto_id', 'id');
22     }
23 }
```

StoreMotorRequest.php U

StoreProdutoRequest.php U

StoreValvulaRequest.php U

StoreVendaRequest.php U

Kernel.php

Models

Carro.php U

Categoria.php U

Cliente.php

Entrada.php U

Fabricacao.php U

FormaPagamento.php U

Fornecedor.php U

ItensEntrada.php U

ItensVenda.php U

Localizacao.php U

Marca.php U

Motor.php U

Produto.php U

User.php

Usuario.php U

Valvula.php U

Venda.php U