



• Requests

The image displays two screenshots of the Visual Studio Code editor, showing PHP code for two different request classes. The top screenshot shows the `StoreClientRequest.php` file, and the bottom screenshot shows the `StoreFornecedorRequest.php` file. Both files are part of a project structure visible in the Explorer sidebar.

StoreClientRequest.php

```
<?php
namespace App\Http\Requests;
use App\Rules\UpperCase;
use Illuminate\Foundation\Http\FormRequest;

class StoreClientRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'nome' => ['required', new UpperCase()],
            'sexo' => 'required',
            'endereco' => 'required',
            'debito' => 'required'
        ];
    }

    public function messages()
    {
        return [
            'nome.required' => 'Preencha o campo Nome',
            'sexo.required' => 'Preencha o campo Sexo',
            'endereco.required' => 'Preencha o campo Endereco',
            'debito.required' => 'Preencha o campo Débito'
        ];
    }
}
```

StoreFornecedorRequest.php

```
<?php
namespace App\Http\Requests;
use App\Rules\UpperCase;
use Illuminate\Foundation\Http\FormRequest;

class StoreFornecedorRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            'nome' => ['required', new UpperCase()],
            'cnpj' => 'required',
            'endereco' => 'required'
        ];
    }

    public function messages()
    {
        return [
            'nome.required' => 'Preencha o campo Nome',
            'cnpj.required' => 'Preencha o campo CNPJ',
            'endereco.required' => 'Preencha o campo Endereco'
        ];
    }
}
```

```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreProdutoRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
15     {
16         return true;
17     }
18
19     /**
20      * Get the validation rules that apply to the request.
21      *
22      * @return array
23      */
24     public function rules()
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()
```

```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreEntradaRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
15     {
16         return true;
17     }
18
19     /**
20      * Get the validation rules that apply to the request.
21      *
22      * @return array
23      */
24     public function rules()
25     {
26         return [
27             'idfornecedor' => 'required',
28             'valortotal' => 'required',
29             'datacompra' => 'required'
30         ];
31     }
32
33     public function messages(){
34         return [
35             'idfornecedor.required' => 'Preencha o campo ID Fornecedor*',
36             'valortotal.required' => 'Preencha o campo Valor Total*',
37             'datacompra.required' => 'Preencha o campo da Data da Compra*'
38         ];
39     }
40 }
41
```

```
StoreVendaRequest.php - codigo - Visual Studio Code

EXPLORER
...
StoreVendaRequest.php X
app > Http > Requests > StoreVendaRequest.php
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreVendaRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
15     {
16         return true;
17     }
18
19     /**
20      * Get the validation rules that apply to the request.
21      *
22      * @return array
23      */
24     public function rules()
25     {
26         return [
27             'idcliente' => 'required',
28             'valortotal' => 'required',
29             'datavenda' => 'required'
30         ];
31     }
32
33     public function messages(){
34         return [
35             'idcliente' => 'Preencha o campo com o ID do Cliente*',
36             'valortotal' => 'Preencha o campo Valor Total*',
37             'datavenda' => 'Preencha o campo com a Data da Venda*'
38         ];
39     }
40
41 }
```

```
StoreEnderecoRequest.php - codigo - Visual Studio Code

EXPLORER
...
StoreEnderecoRequest.php X
app > Http > Requests > StoreEnderecoRequest.php
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreEnderecoRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
15     {
16         return true;
17     }
18
19     /**
20      * Get the validation rules that apply to the request.
21      *
22      * @return array
23      */
24     public function rules()
25     {
26         return [
27             'logradouro' => 'required',
28             'bairro' => 'required',
29             'cidade' => 'required',
30             'uf' => 'required|min:2|max:2',
31             'cliente_id' => 'required'
32         ];
33     }
34
35     public function messages(){
36         return [
37             'logradouro.required' => 'Preencha o campo Logradouro*',
38             'bairro.required' => 'Preencha o campo Bairro*',
39             'cidade.required' => 'Preencha o campo Cidade*',
40             'uf.required' => 'Preencha o campo UF*',
41             'uf.min' => 'Campo UF deve ter no minimo uma letra*'
42         ];
43     }
44 }
```

- Colocando StoreRequest nos Controllers (demais telas também contém)

```
ClientController.php - codigo - Visual Studio Code

EXPLORER
  app
    Controllers
      ClientController.php
      Controller.php
      EnderecoCo...
      EntradaCon...
      Fornecedor...
      ItensEntrad...
      ItensVenda...
      ProdutoCon...
      VendaContr...
      Middleware
      Requests
      StoreClient...
      StoreEndere...
      StoreEntrad...
      StoreFornec...
      StoreItensEn...
      StoreItensVe...
      StoreProdu...
      StoreVenda...
      Kernel.php
      Models
      Providers
      Rules
      UpperCase.php
      bootstrap
      codigo
      config
      database
      public
      OUTLINE
      TIMELINE

ClientController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Http\Requests\StoreClientRequest;
7 use App\Models\Cliente;
8 class ClientController extends Controller
9 {
10     //
11     public function show(){
12         $clientes = Cliente::all();
13         echo $clientes;
14     }
15
16     public function index(){
17         $clientes = Cliente::all();
18         return view('clientes.index', ['clientes'=>$clientes]);
19     }
20
21     public function create(){
22         return view('clientes.create');
23     }
24
25     public function store(StoreClientRequest $request){
26         $cliente= new Cliente();
27         $cliente->nome= $request->nome;
28         $cliente->sexo= $request->sexo;
29         $cliente->endereco= $request->endereco;
30         $cliente->debito= $request->debito;
31         $cliente->save();
32         return redirect('/clientes/index');
33     }
34
35     public function edit($id){
36         $cliente = Cliente::findOrFail($id);
37         return view('clientes.edit', ['cliente'=>$cliente]);
38     }
39
40     public function update(Request $request){
41         $cliente= Cliente::find($request->id)->update($request->except('_method'));
42         return redirect('clientes/index')->with('msg', 'Cliente atualizado');
43     }
44 }
```

```
FornecedorController.php - codigo - Visual Studio Code

EXPLORER
  app
    Controllers
      FornecedorController.php
      Controller.php
      EnderecoCo...
      EntradaCon...
      Fornecedor...
      ItensEntrad...
      ItensVenda...
      ProdutoCon...
      VendaContr...
      Middleware
      Requests
      StoreClient...
      StoreEndere...
      StoreEntrad...
      StoreFornec...
      StoreItensEn...
      StoreItensVe...
      StoreProdu...
      StoreVenda...
      Kernel.php
      Models
      Providers
      Rules
      UpperCase.php
      bootstrap
      codigo
      config
      database
      public
      OUTLINE
      TIMELINE

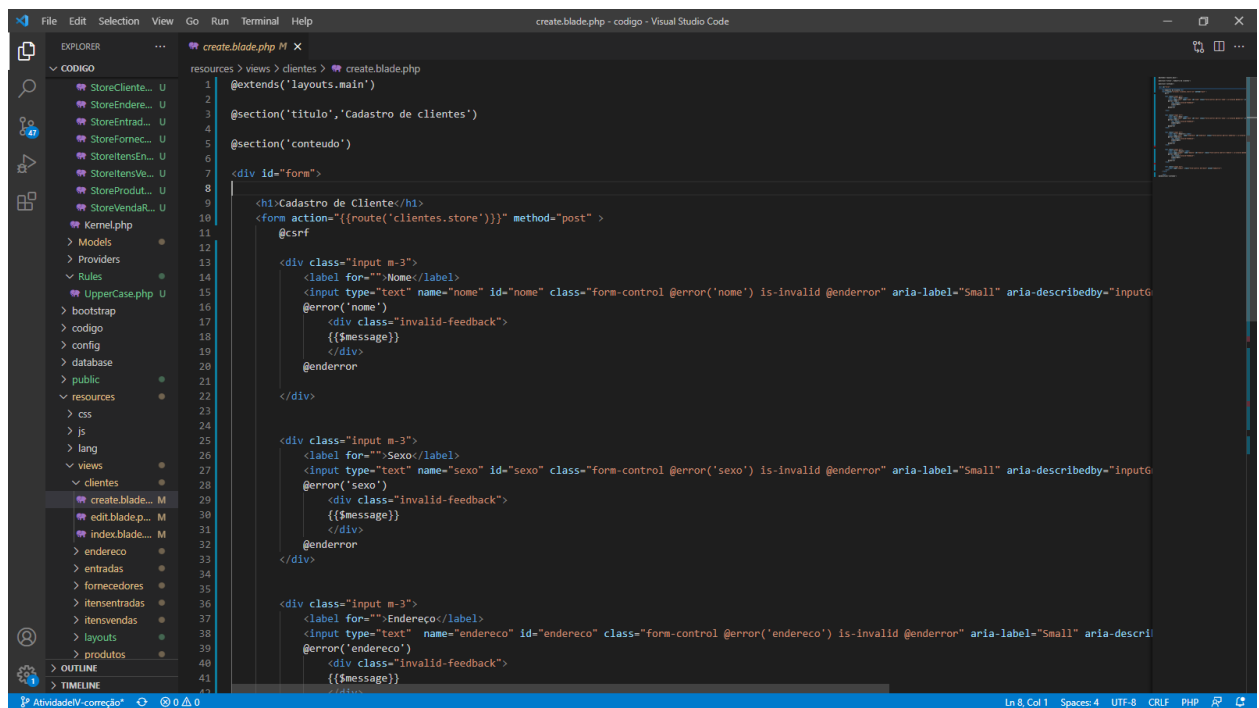
FornecedorController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Fornecedor;
7 use App\Http\Requests\StoreFornecedorRequest;
8 class FornecedorController extends Controller
9 {
10     //
11     public function show(){
12         $fornecedores = Fornecedor::all();
13         echo $fornecedores;
14     }
15
16     public function index(){
17         $fornecedores = Fornecedor::all();
18         return view('fornecedores.index', ['fornecedores'=>$fornecedores]);
19     }
20
21     public function create(){
22         return view('fornecedores.create');
23     }
24
25     public function store(StoreFornecedorRequest $request){
26         $fornecedor= new Fornecedor();
27         $fornecedor->nome= $request->nome;
28         $fornecedor->cnnpj= $request->cnnpj;
29         $fornecedor->endereco= $request->endereco;
30         $fornecedor->save();
31         return redirect('/fornecedores/index');
32     }
33
34     public function edit($id){
35         $fornecedor = Fornecedor::findOrFail($id);
36         return view('fornecedores.edit', ['fornecedor'=>$fornecedor]);
37     }
38
39     public function update(Request $request){
40         $fornecedor= Fornecedor::find($request->id)->update($request->except('_method'));
41         return redirect('fornecedores/index')->with('msg', 'Fornecedor atualizado');
42     }
43
44     public function destroy($id){
45         $fornecedor= Fornecedor::findOrFail($id)->delete();
46     }
47 }
```

```
File Edit Selection View Go Run Terminal Help
ProdutoController.php M X
EXPLORER
CODIGO
app
  Console
  Exceptions
  Http
    Controllers
      ProdutoController.php
      EnderecoCo... M
      EntradaCon... M
      Fornecedor... M
      ItensEntrad... M
      ItensVenda... M
      VendaContr... M
  Middleware
  Requests
    StoreCliente... U
    StoreEndere... U
    StoreEntrad... U
    StoreFornec... U
    StoreItensEn... U
    StoreItensVe... U
    StoreProdut... U
    StoreVendaR... U
  Kernel.php
  Models
  Providers
  Rules
    UpperCase.php U
  bootstrap
  codigo
  config
  database
  public
  OUTLINE
  TIMELINE
app > Http > Controllers > ProdutoController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Models\Produto;
7 use App\Http\Requests\StoreProdutoRequest;
8
9 class ProdutoController extends Controller
10 {
11     //
12     public function show(){
13         $produtos = Produto::all();
14         echo $produtos;
15     }
16
17     public function index(){
18         $produtos = Produto::all();
19         return view('produtos.index', ['produtos'=>$produtos]);
20     }
21     public function create(){
22         return view('produtos.create');
23     }
24
25     public function store(StoreProdutoRequest $request)
26     {
27         $produto = new Produto();
28         $produto->nome = $request->nome;
29         $produto->icms = $request->icms;
30         $produto->ipi = $request->ipi;
31         $produto->frete = $request->frete;
32         $produto->precofabrica = $request->precofabrica;
33         $produto->precocompra = $request->precocompra;
34         $produto->precovenda = $request->precovenda;
35         $produto->lucro = $request->lucro;
36         $produto->desconto = $request->desconto;
37         $produto->quantidade = $request->quantidade;
38         $produto->save();
39
40         return redirect('/produtos/index');
41     }
42 }
```

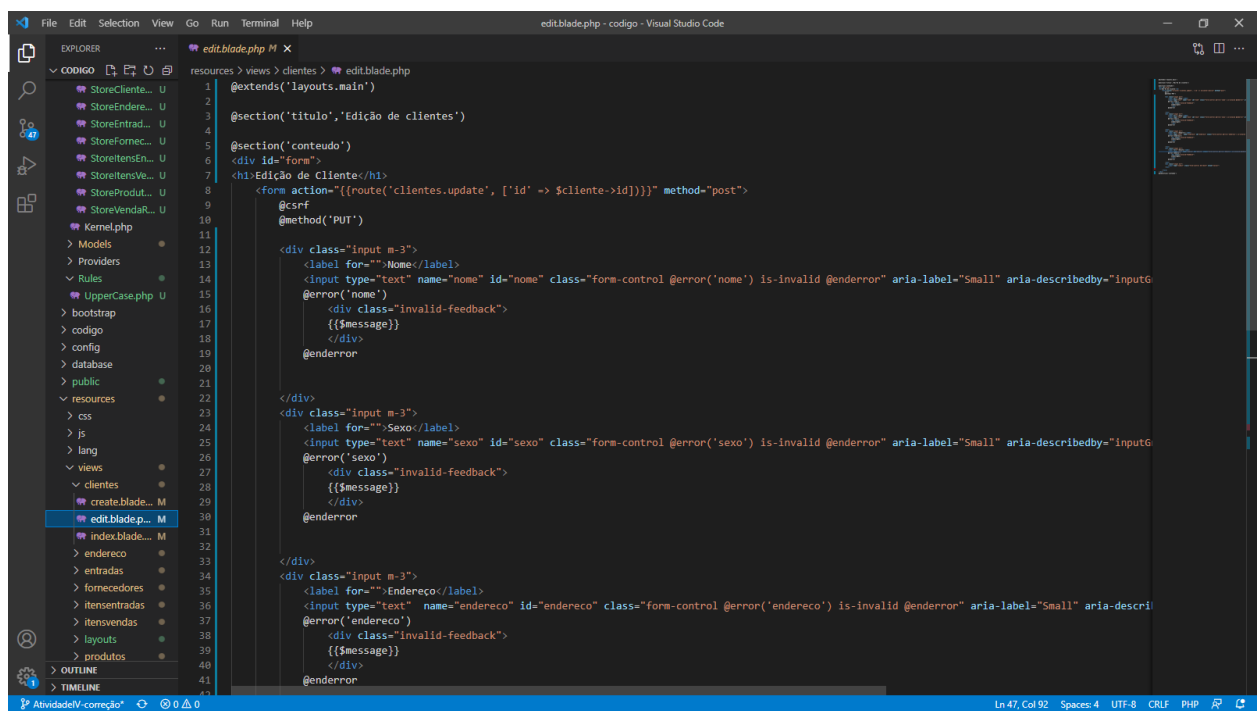
* Uppercase

```
File Edit Selection View Go Run Terminal Help
UpperCase.php U X
EXPLORER
CODIGO
app
  Console
  Exceptions
  Http
    Controllers
      ClienteCont... M
      Controller.php
      EnderecoCo... M
      EntradaCon... M
      Fornecedor... M
      ItensEntrad... M
      ItensVenda... M
      ProdutoCon... M
      VendaContr... M
  Middleware
  Requests
    StoreCliente... U
    StoreEndere... U
    StoreEntrad... U
    StoreFornec... U
    StoreItensEn... U
    StoreItensVe... U
    StoreProdut... U
    StoreVendaR... U
  Kernel.php
  Models
  Providers
  Rules
    UpperCase.php U
  bootstrap
  codigo
  config
  database
  public
  OUTLINE
  TIMELINE
app > Rules > UpperCase.php
1 <?php
2
3 namespace App\Rules;
4
5 use Illuminate\Contracts\Validation\Rule;
6
7 class UpperCase implements Rule
8 {
9     /**
10      * Create a new rule instance.
11      *
12      * @return void
13      */
14     public function __construct()
15     {
16         //
17     }
18
19     /**
20      * Determine if the validation rule passes.
21      *
22      * @param string $attribute
23      * @param mixed $value
24      * @return bool
25      */
26     public function passes($attribute, $value)
27     {
28         return str_word_count($value) > 1;
29     }
30
31     /**
32      * Get the validation error message.
33      *
34      * @return string
35      */
36     public function message()
37     {
38         return 'Insira o nome completo *';
39     }
40 }
41 }
```

- Views (demais telas também contém)



```
1 @extends('layouts.main')
2
3 @section('titulo','Cadastro de clientes')
4
5 @section('conteudo')
6
7 <div id="form">
8
9     <h1>Cadastro de Cliente</h1>
10     <form action="{{route('clientes.store')}}" method="post">
11         @csrf
12
13         <div class="input m-3">
14             <label for="">Nome</label>
15             <input type="text" name="nome" id="nome" class="form-control @error('nome') is-invalid @enderror" aria-label="Small" aria-describedby="inputG
16                 @error('nome')
17                     <div class="invalid-feedback">
18                         {{ $message }}
19                     </div>
20                 @enderror
21         </div>
22
23         <div class="input m-3">
24             <label for="">Sexo</label>
25             <input type="text" name="sexo" id="sexo" class="form-control @error('sexo') is-invalid @enderror" aria-label="Small" aria-describedby="inputG
26                 @error('sexo')
27                     <div class="invalid-feedback">
28                         {{ $message }}
29                     </div>
30                 @enderror
31         </div>
32
33         <div class="input m-3">
34             <label for="">Endereço</label>
35             <input type="text" name="endereco" id="endereco" class="form-control @error('endereco') is-invalid @enderror" aria-label="Small" aria-descri
36                 @error('endereco')
37                     <div class="invalid-feedback">
38                         {{ $message }}
39                     </div>
40                 @enderror
41         </div>
42     </form>
43 </div>
```



```
1 @extends('layouts.main')
2
3 @section('titulo','Edição de clientes')
4
5 @section('conteudo')
6
7 <div id="form">
8     <h1>Edição de Cliente</h1>
9     <form action="{{route('clientes.update', ['id' => $cliente->id])}}" method="post">
10         @csrf
11         @method('PUT')
12
13         <div class="input m-3">
14             <label for="">Nome</label>
15             <input type="text" name="nome" id="nome" class="form-control @error('nome') is-invalid @enderror" aria-label="Small" aria-describedby="inputG
16                 @error('nome')
17                     <div class="invalid-feedback">
18                         {{ $message }}
19                     </div>
20                 @enderror
21         </div>
22
23         <div class="input m-3">
24             <label for="">Sexo</label>
25             <input type="text" name="sexo" id="sexo" class="form-control @error('sexo') is-invalid @enderror" aria-label="Small" aria-describedby="inputG
26                 @error('sexo')
27                     <div class="invalid-feedback">
28                         {{ $message }}
29                     </div>
30                 @enderror
31         </div>
32
33         <div class="input m-3">
34             <label for="">Endereço</label>
35             <input type="text" name="endereco" id="endereco" class="form-control @error('endereco') is-invalid @enderror" aria-label="Small" aria-descri
36                 @error('endereco')
37                     <div class="invalid-feedback">
38                         {{ $message }}
39                     </div>
40                 @enderror
41         </div>
42     </form>
43 </div>
```

```
File Edit Selection View Go Run Terminal Help create.blade.php - codigo - Visual Studio Code

EXPLORER
  CODIGO
    Models
    Providers
    Rules
    UpperCase.php U
    bootstrap
    codigo
    config
    database
    public
    resources
    css
    js
    lang
    views
    clientes
    create.blade... M
    edit.blade.p... M
    index.blade... M
    endereco
    entradas
    fornecedores
    create.blade... M
    edit.blade.p... M
    index.blade... M
    itensentradas
    itensvendas
    layouts
    produtos
    vendas
    welcome.blade.php
    routes
    storage
    tests
    vendor
    OUTLINE
    TIMELINE

resources > views > fornecedores > create.blade.php
1 @extends('layouts.main')
2
3 @section('titulo','Cadastro de Fornecedor')
4
5 @section('conteudo')
6
7 <div id="form">
8   <h1>Cadastrar Fornecedor</h1>
9   <form action="{{route('fornecedores.store')}}" method="post">
10     @csrf
11     <div class="input m-3">
12       <label for="">Nome:</label>
13       <input type="text" name="nome" id="nome" class="form-control @error('nome') is-invalid @enderror" aria-label="Small" aria-describedby="inputGroup
14       @error('nome')
15         <div class="invalid-feedback">
16           {{ $message }}
17         </div>
18       @enderror
19     </div>
20     <div class="input m-3">
21       <p><label for="">CNPJ:</label>
22       <input type="text" name="cnpj" id="cnpj" class="form-control @error('cnpj') is-invalid @enderror" aria-label="Small" aria-describedby="inputGroup
23       @error('cnpj')
24         <div class="invalid-feedback">
25           {{ $message }}
26         </div>
27       @enderror
28     </div>
29     <div class="input m-3">
30       <p><label for="">Endereço:</label>
31       <input type="text" name="endereco" id="endereco" class="form-control @error('endereco') is-invalid @enderror" aria-label="Small" aria-describedby="
32       @error('endereco')
33         <div class="invalid-feedback">
34           {{ $message }}
35         </div>
36       @enderror
37     </div>
38     <div class="input m-3">
39       <input type="submit" class="form-control btn-dark" value="Cadastrar">
40     </div>
41   </form>
42 </div>
```

```
File Edit Selection View Go Run Terminal Help create.blade.php - codigo - Visual Studio Code

EXPLORER
  CODIGO
    Models
    Providers
    Rules
    UpperCase.php U
    bootstrap
    codigo
    config
    database
    public
    resources
    css
    js
    lang
    views
    clientes
    create.blade... M
    edit.blade.p... M
    index.blade... M
    endereco
    entradas
    fornecedores
    create.blade... M
    edit.blade.p... M
    index.blade... M
    itensentradas
    itensvendas
    layouts
    produtos
    vendas
    welcome.blade.php
    routes
    storage
    tests
    vendor
    OUTLINE
    TIMELINE

resources > views > produtos > create.blade.php
1 @extends('layouts.main')
2
3 @section('titulo','Cadastro de produtos')
4
5 @section('conteudo')
6
7 <div id="form">
8   <h1>Cadastrar Produto</h1>
9   <form action="{{route('produtos.store')}}" method="post">
10     @csrf
11     <div class="row">
12       <div class="col">
13         <label for="">Nome:</label>
14         <input type="text" name="nome" id="nome" class="form-control @error('nome') is-invalid @enderror" aria-label="Small" aria-describedby="inp
15         @error('nome')
16           <div class="invalid-feedback">
17             {{ $message }}
18           </div>
19         @enderror
20       </div>
21       <div class="col">
22         <p><label for="">ICMS:</label>
23         <input type="text" name="icms" id="icms" class="form-control @error('icms') is-invalid @enderror" aria-label="Small" aria-describedby="inp
24         @error('icms')
25           <div class="invalid-feedback">
26             {{ $message }}
27           </div>
28         @enderror
29       </div>
30       <p>
31       <div class="col">
32         <p><label for="">IPI:</label>
33         <input type="text" name="ipi" id="ipi" class="form-control @error('ipi') is-invalid @enderror" aria-label="Small" aria-describedby="inputG
34         @error('ipi')
35           <div class="invalid-feedback">
36             {{ $message }}
37           </div>
38         @enderror
39       </div>
40     </div>
41   </form>
42 </div>
```