

Atividade Projeto1-1Unidade

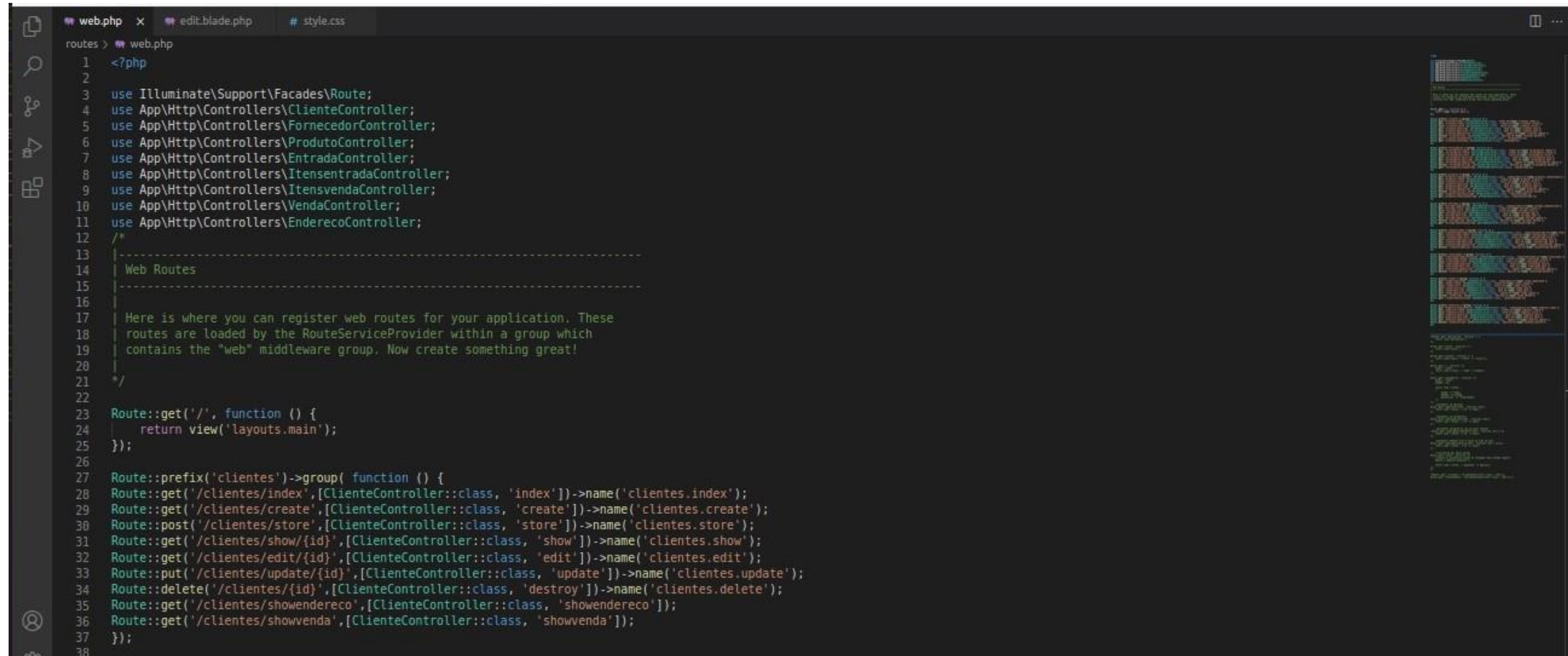
Aluno: John Junqueira Aranha

Disciplina: Laboratório de Programação Web2

Professor: Fábio Lima

Data: 23/11/2021

Rotas



```
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\ClienteController;
5  use App\Http\Controllers\FornecedorController;
6  use App\Http\Controllers\ProdutoController;
7  use App\Http\Controllers\EntradaController;
8  use App\Http\Controllers\ItensentradaController;
9  use App\Http\Controllers\ItensvendaController;
10 use App\Http\Controllers\VendaController;
11 use App\Http\Controllers\EnderecoController;
12 /*
13 |-----|
14 | Web Routes |
15 |-----|
16 |
17 | Here is where you can register web routes for your application. These
18 | routes are loaded by the RouteServiceProvider within a group which
19 | contains the "web" middleware group. Now create something great!
20 |
21 */
22
23 Route::get('/', function () {
24     return view('layouts.main');
25 });
26
27 Route::prefix('clientes')->group( function () {
28     Route::get('/clientes/index',[ClienteController::class, 'index'])->name('clientes.index');
29     Route::get('/clientes/create',[ClienteController::class, 'create'])->name('clientes.create');
30     Route::post('/clientes/store',[ClienteController::class, 'store'])->name('clientes.store');
31     Route::get('/clientes/show/{id}',[ClienteController::class, 'show'])->name('clientes.show');
32     Route::get('/clientes/edit/{id}',[ClienteController::class, 'edit'])->name('clientes.edit');
33     Route::put('/clientes/update/{id}',[ClienteController::class, 'update'])->name('clientes.update');
34     Route::delete('/clientes/{id}',[ClienteController::class, 'destroy'])->name('clientes.delete');
35     Route::get('/clientes/showendereco',[ClienteController::class, 'showendereco']);
36     Route::get('/clientes/showvenda',[ClienteController::class, 'showvenda']);
37 });
38
```

```
web.php x edit.blade.php # style.css
routes > web.php
39 Route::prefix('fornecedores')->group( function () {
40 Route::get('/fornecedores/indexs',[FornecedorController::class, 'indexs'])->name('fornecedores.indexs');
41 Route::get('/fornecedores/create',[FornecedorController::class, 'create'])->name('fornecedores.create');
42 Route::post('/fornecedores/store',[FornecedorController::class, 'store'])->name('fornecedores.store');
43 Route::get('/fornecedores/show/{id}',[FornecedorController::class, 'show'])->name('fornecedores.show');
44 Route::get('/fornecedores/edit/{id}',[FornecedorController::class, 'edit'])->name('fornecedores.edit');
45 Route::put('/fornecedores/update/{id}',[FornecedorController::class, 'update'])->name('fornecedores.update');
46 Route::delete('/fornecedores/{id}',[FornecedorController::class, 'destroy'])->name('fornecedores.delete');
47 Route::get('/fornecedores/showentrada',[FornecedorController::class, 'showentrada']);
48 });
49
50 Route::prefix('produtos')->group( function () {
51 Route::get('/produtos/indexproduto',[ProdutoController::class, 'indexproduto'])->name('produtos.indexproduto');
52 Route::get('/produtos/create',[ProdutoController::class, 'create'])->name('produtos.create');
53 Route::post('/produtos/store',[ProdutoController::class, 'store'])->name('produtos.store');
54 Route::get('/produtos/show/{id}',[ProdutoController::class, 'show'])->name('produtos.show');
55 Route::get('/produtos/edit/{id}',[ProdutoController::class, 'edit'])->name('produtos.edit');
56 Route::put('/produtos/update/{id}',[ProdutoController::class, 'update'])->name('produtos.update');
57 Route::delete('/produtos/{id}',[ProdutoController::class, 'destroy'])->name('produtos.delete');
58 Route::get('/produtos/showitensvenda',[ProdutoController::class, 'showitensvenda']);
59 Route::get('/produtos/showitensentrada',[ProdutoController::class, 'showitensentrada']);
60 });
61
62 Route::prefix('entradas')->group( function () {
63 Route::get('/entrada/indexentrada',[EntradaController::class, 'indexentrada'])->name('entrada.indexentrada');
64 Route::get('/entrada/create',[EntradaController::class, 'create'])->name('entrada.create');
65 Route::post('/entrada/store',[EntradaController::class, 'store'])->name('entrada.store');
66 Route::get('/entrada/show/{id}',[EntradaController::class, 'show'])->name('entrada.show');
67 Route::get('/entrada/edit/{id}',[EntradaController::class, 'edit'])->name('entradas.edit');
68 Route::put('/entrada/update/{id}',[EntradaController::class, 'update'])->name('entrada.update');
69 Route::delete('/entrada/{id}',[EntradaController::class, 'destroy'])->name('entradas.delete');
70 Route::get('/entrada/showitensentrada',[EntradaController::class, 'showitensentrada']);
71 });
72
```

web.php x edit.blade.php # style.css

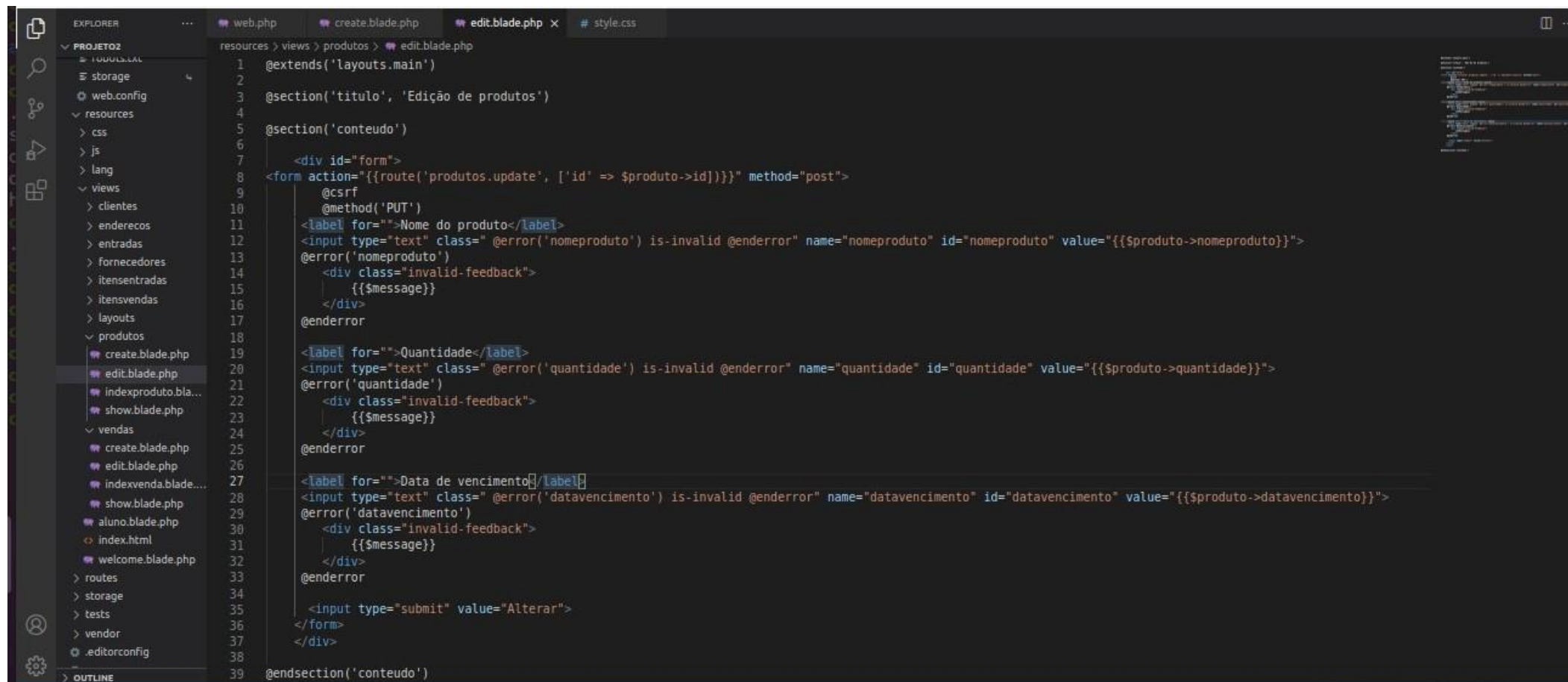
routes > web.php

```
73 Route::prefix('itensentradas')->group( function () {
74 Route::get('/itensentrada/indexitensentrada',[ItensentradaController::class, 'indexitensentrada'])->name('itensentrada.indexitensentrada');
75 Route::get('/itensentrada/create',[ItensentradaController::class, 'create'])->name('itensentrada.create');
76 Route::post('/itensentrada/store',[ItensentradaController::class, 'store'])->name('itensentrada.store');
77 Route::get('/itensentrada/show/{id}',[ItensentradaController::class, 'show'])->name('itensentrada.show');
78 Route::get('/itensentrada/edit/{id}',[ItensentradaController::class, 'edit'])->name('itensentrada.edit');
79 Route::put('/itensentrada/update/{id}',[ItensentradaController::class, 'update'])->name('itensentrada.update');
80 Route::delete('/itensentrada/{id}',[ItensentradaController::class, 'destroy'])->name('itensentrada.delete');
81 });
82
83 Route::prefix('itensvendas')->group( function () {
84 Route::get('/itensvenda/indexitensvenda',[ItensvendaController::class, 'indexitensvenda'])->name('itensvenda.indexitensvenda');
85 Route::get('/itensvenda/create',[ItensvendaController::class, 'create'])->name('itensvenda.create');
86 Route::post('/itensvenda/store',[ItensvendaController::class, 'store'])->name('itensvenda.store');
87 Route::get('/itensvenda/show/{id}',[ItensvendaController::class, 'show'])->name('itensvenda.show');
88 Route::get('/itensvenda/edit/{id}',[ItensvendaController::class, 'edit'])->name('itensvenda.edit');
89 Route::put('/itensvenda/update/{id}',[ItensvendaController::class, 'update'])->name('itensvenda.update');
90 Route::delete('/itensvenda/{id}',[ItensvendaController::class, 'destroy'])->name('itensvenda.delete');
91 });
92
93 Route::prefix('vendas')->group( function () {
94 Route::get('/venda/indexvenda',[VendaController::class, 'indexvenda'])->name('venda.indexvenda');
95 Route::get('/venda/create',[VendaController::class, 'create'])->name('venda.create');
96 Route::post('/venda/store',[VendaController::class, 'store'])->name('venda.store');
97 Route::get('/venda/show/{id}',[VendaController::class, 'show'])->name('venda.show');
98 Route::get('/venda/edit/{id}',[VendaController::class, 'edit'])->name('venda.edit');
99 Route::put('/venda/update/{id}',[VendaController::class, 'update'])->name('venda.update');
100 Route::delete('/venda/{id}',[VendaController::class, 'destroy'])->name('venda.delete');
101 Route::get('/venda/showitensvenda',[VendaController::class, 'showitensvenda']);
102 });
103
104 Route::prefix('enderecos')->group( function () {
105 Route::get('/endereco/indexendereço',[EnderecoController::class, 'indexendereço'])->name('endereço.index');
106 Route::get('/endereço/create',[EnderecoController::class, 'create'])->name('endereço.create');
107 Route::post('/endereço/store',[EnderecoController::class, 'store'])->name('endereço.store');
108 Route::get('/endereço/show/{id}',[EnderecoController::class, 'show'])->name('endereço.show');
109 Route::get('/endereço/edit/{id}',[EnderecoController::class, 'edit'])->name('endereço.edit');
110 Route::put('/endereço/update/{id}',[EnderecoController::class, 'update'])->name('endereço.update');
111 Route::delete('/endereço/{id}',[EnderecoController::class, 'destroy'])->name('endereço.delete');
```


Views – (Produtos “create.blade.php”)

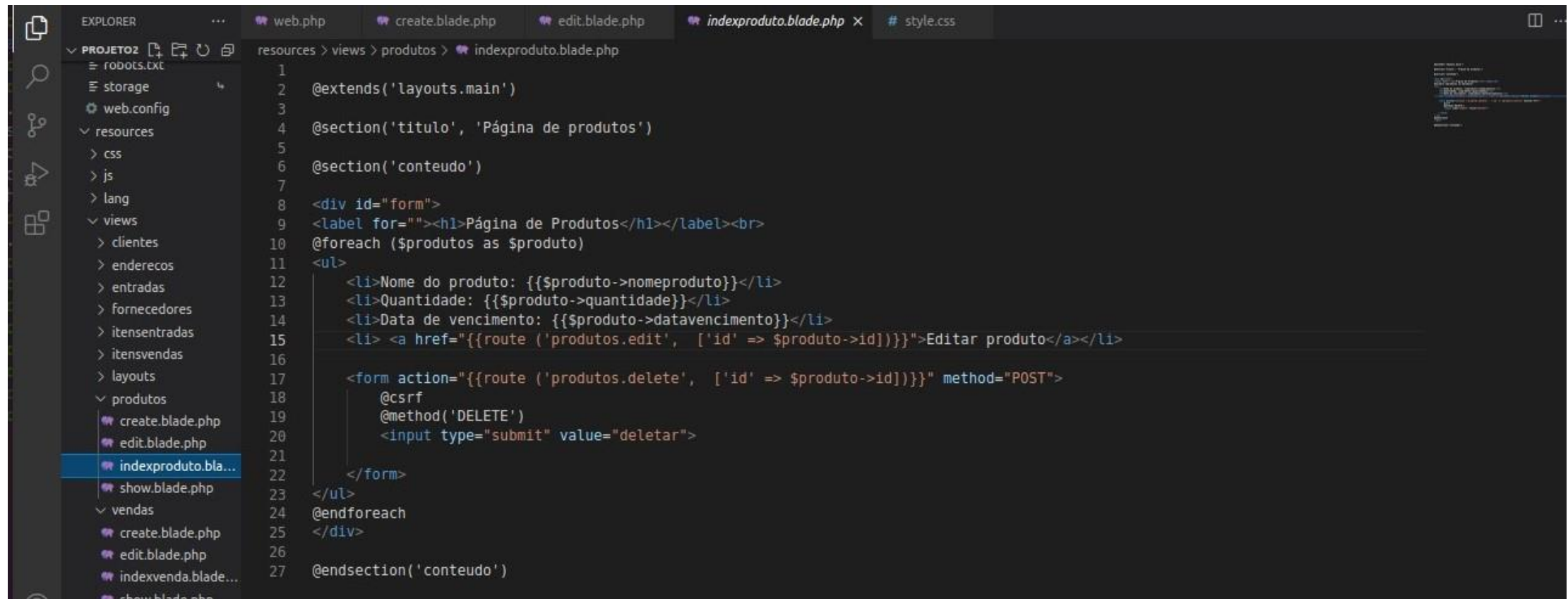
```
resources > views > produtos > create.blade.php
1
2 @extends('layouts.main')
3
4 @section('titulo', 'Cadastro de produtos')
5
6 @section('conteudo')
7
8     <div id="form">
9         <form action="{{route('produtos.store')}}" method="post">
10             @csrf
11             <div class="title">
12                 <label for=""><h3>Cadastrar Produto</h3></label><br><br>
13             </div>
14             <div class="input">
15                 <label for="">Nome do produto</label>
16                 <input type="text" class="" @error('nomeproduto') is-invalid @enderror" name="nomeproduto" id="nomeproduto"><br><br>
17                 @error('nomeproduto')
18                 <div class="invalid-feedback">
19                     {{ $message }}
20                 </div>
21                 @enderror
22             </div>
23             <div class="input">
24                 <label for="">Quantidade</label>
25                 <input type="text" class="" @error('quantidade') is-invalid @enderror" name="quantidade" id="quantidade"><br><br>
26                 @error('quantidade')
27                 <div class="invalid-feedback">
28                     {{ $message }}
29                 </div>
30             </div>
31             @enderror
32             <div class="input">
33                 <label for="">Data de vencimento</label>
34                 <input type="text" class="" @error('datavencimento') is-invalid @enderror" name="datavencimento" id="datavencimento"><br><br>
35                 @error('datavencimento')
36                 <div class="invalid-feedback">
37                     {{ $message }}
38                 </div>
39             </div>
40             @enderror
41             <div class="inputcad">
42                 <input type="submit" value="Cadastrar"><br><br>
43             </div>
44         </form>
45     </div>
46
47 @endsection('conteudo')
```

(Produtos “edit.blade.php”)



```
1 @extends('layouts.main')
2
3 @section('titulo', 'Edição de produtos')
4
5 @section('conteudo')
6
7     <div id="form">
8         <form action="{{route('produtos.update', ['id' => $produto->id])}}" method="post">
9             @csrf
10             @method('PUT')
11             <label for="">Nome do produto</label>
12             <input type="text" class=" @error('nomeproduto') is-invalid @enderror" name="nomeproduto" id="nomeproduto" value="{{ $produto->nomeproduto }}">
13             @error('nomeproduto')
14                 <div class="invalid-feedback">
15                     {{ $message }}
16                 </div>
17             @enderror
18
19             <label for="">Quantidade</label>
20             <input type="text" class=" @error('quantidade') is-invalid @enderror" name="quantidade" id="quantidade" value="{{ $produto->quantidade }}">
21             @error('quantidade')
22                 <div class="invalid-feedback">
23                     {{ $message }}
24                 </div>
25             @enderror
26
27             <label for="">Data de vencimento</label>
28             <input type="text" class=" @error('datavencimento') is-invalid @enderror" name="datavencimento" id="datavencimento" value="{{ $produto->datavencimento }}">
29             @error('datavencimento')
30                 <div class="invalid-feedback">
31                     {{ $message }}
32                 </div>
33             @enderror
34
35             <input type="submit" value="Alterar">
36         </form>
37     </div>
38 @endsection('conteudo')
```

(Produtos “index.blade.php”)



The image shows a code editor interface with a dark theme. On the left, the 'EXPLORER' sidebar displays a file tree for a project named 'PROJETO2'. The tree includes folders like 'storage', 'resources', and 'views', with 'views' expanded to show subfolders like 'clientes', 'enderecos', and 'produtos'. The 'produtos' folder is selected, showing files like 'create.blade.php', 'edit.blade.php', and 'indexproduto.blade.php'. The main editor area shows the content of 'indexproduto.blade.php' with line numbers 1 through 27. The code uses Blade templating syntax to extend a layout, define sections, and iterate over a list of products to display their details and provide edit/delete links.

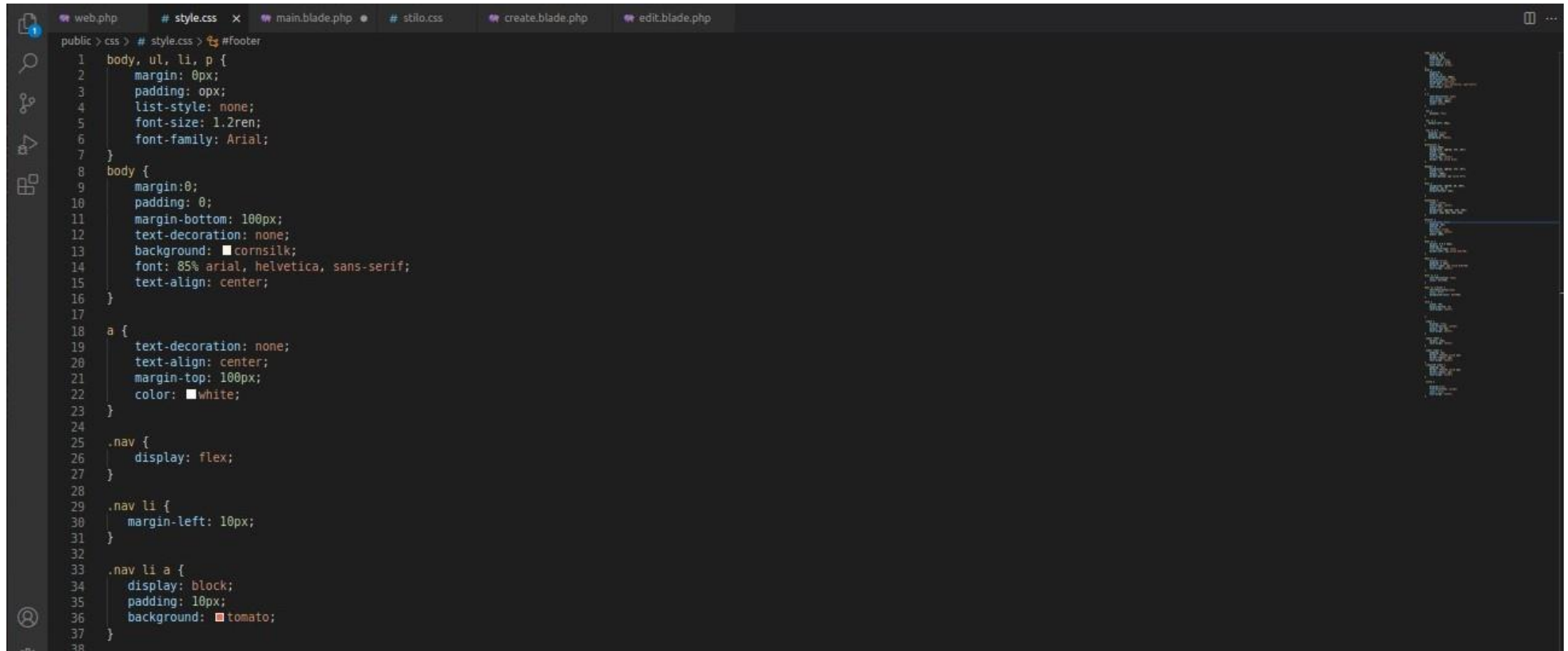
```
1 @extends('layouts.main')
2
3
4 @section('titulo', 'Página de produtos')
5
6 @section('conteudo')
7
8 <div id="form">
9 <label for=""><h1>Página de Produtos</h1></label><br>
10 @foreach ($produtos as $produto)
11 <ul>
12 <li>Nome do produto: {{ $produto->nomeproduto }}</li>
13 <li>Quantidade: {{ $produto->quantidade }}</li>
14 <li>Data de vencimento: {{ $produto->datavencimento }}</li>
15 <li><a href="{{ route('produtos.edit', ['id' => $produto->id]) }}">Editar produto</a></li>
16
17 <form action="{{ route('produtos.delete', ['id' => $produto->id]) }}" method="POST">
18 @csrf
19 @method('DELETE')
20 <input type="submit" value="deletar">
21 </form>
22 </ul>
23 @endforeach
24 </div>
25
26
27 @endsection('conteudo')
```

Layouts-main.blade

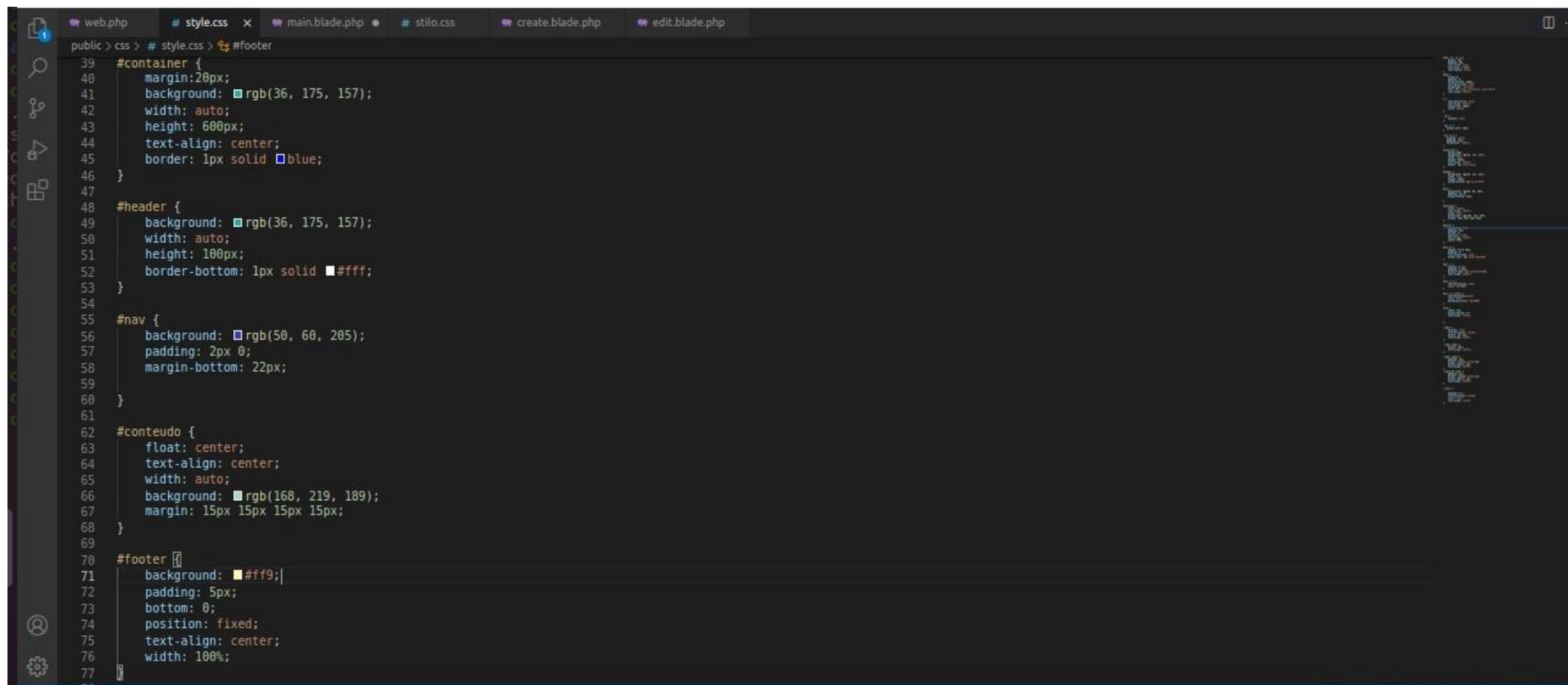
```
resources > views > layouts > main.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <!-- CSS only -->
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWbQ78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6j"
9   <!-- CSS Local -->
10  <link rel="stylesheet" href="{{URL::asset('css/style.css')}}">
11
12  <title>@yield('titulo')</title>
13 </head>
14 <body>
15   <div id="container">
16     <div id="header">
17       <h1>Sistema de vendas</h1>
18     </div>
19     <div id="nav">
20       <ul>
21         <li><a href="{{route('produtos.indexproduto')}}">Produtos</a></li>
22         <li><a href="{{route('fornecedores.indexs')}}">Fornecedores</a></li>
23         <li><a href="{{route('entrada.indexentrada')}}">Entradas</a></li>
24         <li><a href="{{route('itensentrada.indexitensentrada')}}">Itens Entrada</a></li>
25         <li><a href="{{route('venda.indexvenda')}}">Vendas</a></li>
26         <li><a href="{{route('itensvenda.indexitensvenda')}}">Itens Venda</a></li>
27         <li><a href="{{route('clientes.index')}}">Clientes</a></li>
28         <li><a href="{{route('endereco.index')}}">Endereço</a></li>
29       </ul>
30     </div>
31     <div id="conteudo">
32       @yield('conteudo')
33     </div>
34     <div id="footer">
35       Copyright @ WEB II - 2021.1
36     </div>
37   </div>
38   <!-- JavaScript Bundle with Popper -->
39   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikTlWxgYs0g+OMhuP+IlRH9sENB00LRn5q+8nbTov4+1p"
40   crossorigin="anonymous"></script>
41
42 </body>
43 </html>
```


- A codificação das demais Views, seguem o mesmo padrão para o 'create', 'edit' e 'index'; sendo das tabelas clientes, fornecedores, entradas, itensentradas, vendas, itensvendas, endereço, além de produtos.

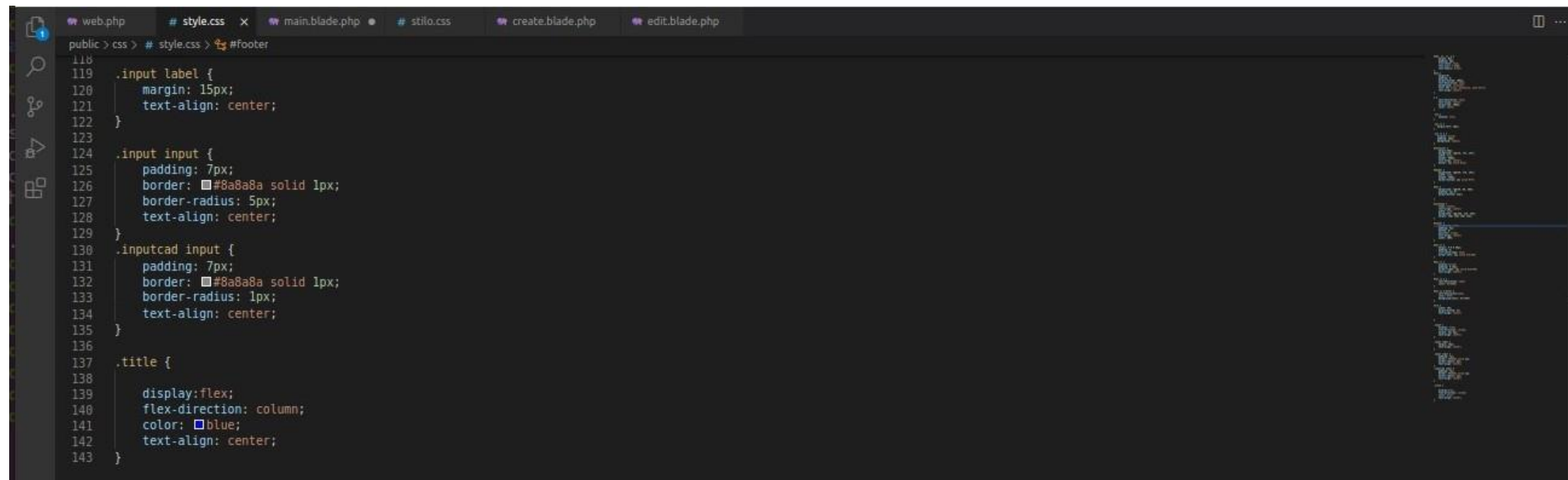
CSS-style.css

A screenshot of a code editor with a dark theme. The editor has several tabs at the top: 'web.php', '# style.css' (active), 'main.blade.php', '# stilo.css', 'create.blade.php', and 'edit.blade.php'. The active tab shows CSS code for 'style.css'. The code is as follows:

```
public > css > # style.css > #footer
1  body, ul, li, p {
2      margin: 0px;
3      padding: 0px;
4      list-style: none;
5      font-size: 1.2rem;
6      font-family: Arial;
7  }
8  body {
9      margin: 0;
10     padding: 0;
11     margin-bottom: 100px;
12     text-decoration: none;
13     background: #f0f0f0;
14     font: 85% arial, helvetica, sans-serif;
15     text-align: center;
16 }
17
18 a {
19     text-decoration: none;
20     text-align: center;
21     margin-top: 100px;
22     color: #fff;
23 }
24
25 .nav {
26     display: flex;
27 }
28
29 .nav li {
30     margin-left: 10px;
31 }
32
33 .nav li a {
34     display: block;
35     padding: 10px;
36     background: #f08080;
37 }
38
```

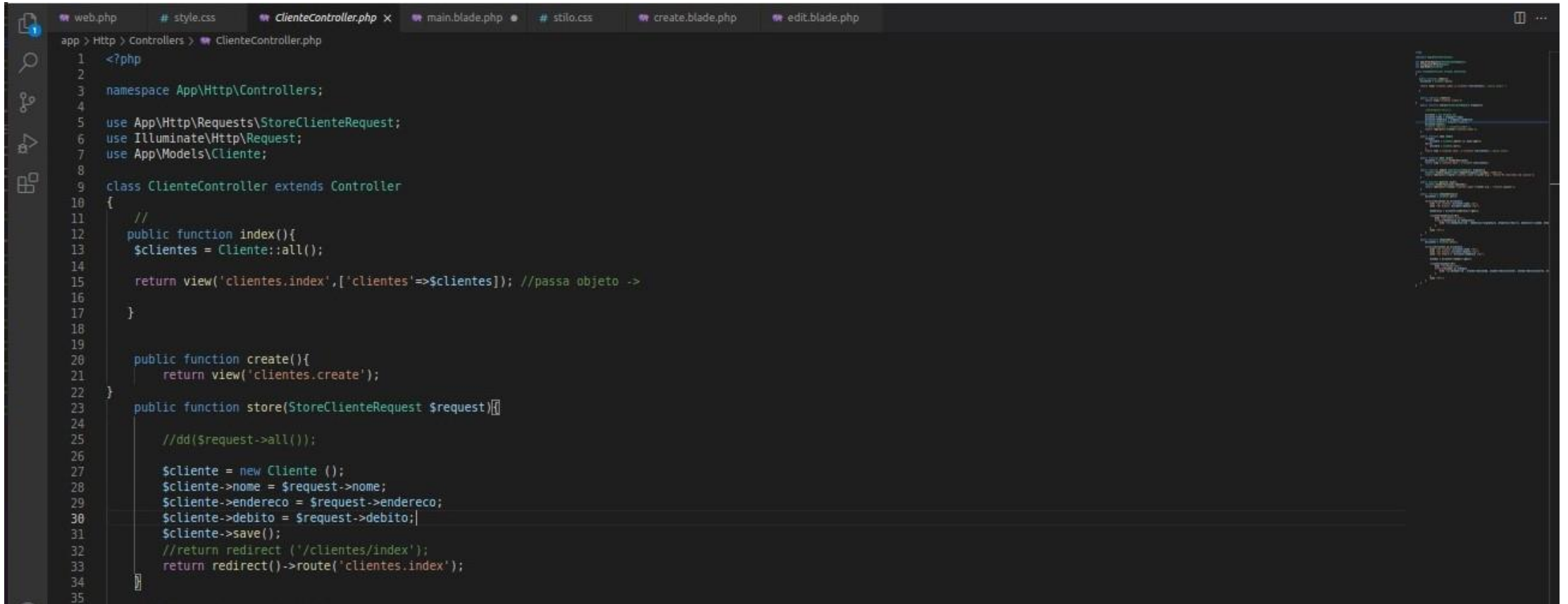


```
public > css > # style.css > #footer
39 #container {
40     margin: 20px;
41     background: rgb(36, 175, 157);
42     width: auto;
43     height: 600px;
44     text-align: center;
45     border: 1px solid blue;
46 }
47
48 #header {
49     background: rgb(36, 175, 157);
50     width: auto;
51     height: 100px;
52     border-bottom: 1px solid #fff;
53 }
54
55 #nav {
56     background: rgb(50, 60, 205);
57     padding: 2px 0;
58     margin-bottom: 22px;
59 }
60
61
62 #conteudo {
63     float: center;
64     text-align: center;
65     width: auto;
66     background: rgb(168, 219, 189);
67     margin: 15px 15px 15px 15px;
68 }
69
70 #footer {
71     background: #ff9;
72     padding: 5px;
73     bottom: 0;
74     position: fixed;
75     text-align: center;
76     width: 100%;
77 }
```

Controllers

- ClienteController.php

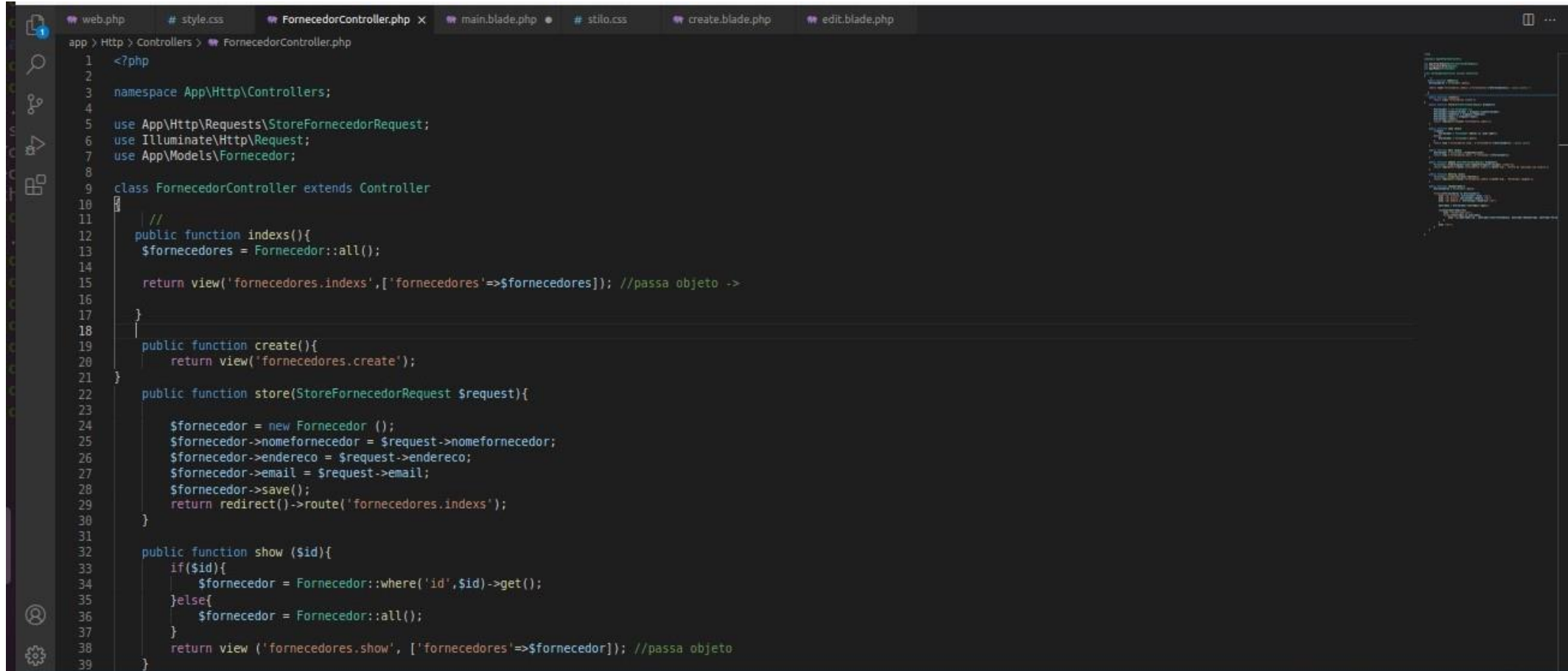


```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Http\Requests\StoreClienteRequest;
6  use Illuminate\Http\Request;
7  use App\Models\Cliente;
8
9  class ClienteController extends Controller
10 {
11     //
12     public function index(){
13         $clientes = Cliente::all();
14
15         return view('clientes.index', ['clientes'=>$clientes]); //passa objeto ->
16     }
17
18
19
20     public function create(){
21         return view('clientes.create');
22     }
23
24     public function store(StoreClienteRequest $request){
25
26         //dd($request->all());
27
28         $cliente = new Cliente ();
29         $cliente->nome = $request->nome;
30         $cliente->endereco = $request->endereco;
31         $cliente->debito = $request->debito;
32         $cliente->save();
33         //return redirect ('/clientes/index');
34         return redirect()->route('clientes.index');
35     }
36 }
```

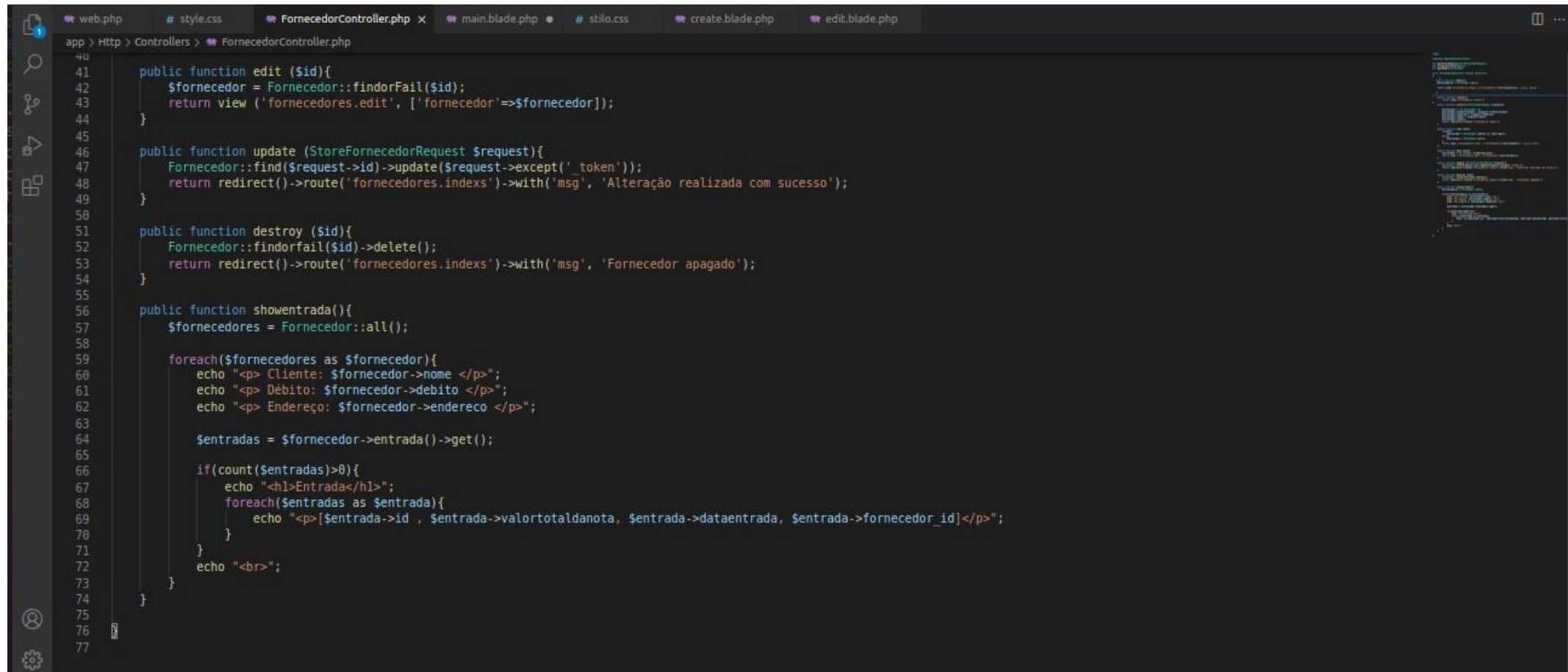
```
web.php # style.css ClienteController.php x main.blade.php # stilo.css create.blade.php edit.blade.php
app > Http > Controllers > ClienteController.php
36 public function show ($id){
37     if($id){
38         $cliente = Cliente::where('id',$id)->get();
39     }else{
40         $cliente = Cliente::all();
41     }
42     return view ('clientes.show', ['clientes'=>$cliente]); //passa objeto
43 }
44
45 public function edit ($id){
46     $cliente = Cliente::findorFail($id);
47     return view ('clientes.edit', ['cliente'=>$cliente]);
48 }
49
50 public function update (StoreClienteRequest $request){
51     Cliente::find($request->id)->update($request->except(' token'));
52     return redirect()->route('clientes.index')->with('msg', 'Alteração realizada com sucesso');
53 }
54
55 public function destroy ($id){
56     Cliente::findorfail($id)->delete();
57     return redirect()->route('clientes.index')->with('msg', 'Cliente apagado');
58 }
59
60 public function showendereco(){
61     $clientes = Cliente::all();
62
63     foreach($clientes as $cliente){
64         echo "<p> Cliente: $cliente->nome </p>";
65         echo "<p> Debito: $cliente->debito </p>";
66
67         $enderecos = $cliente->endereco()->get();
68     }
69 }
```

```
web.php # style.css ClienteController.php main.blade.php # stilo.css create.blade.php edit.blade.php
app > Http > Controllers > ClienteController.php
68
69     if(count($enderecos)>0){
70         echo "<h1>Endereço</h1>";
71         foreach($enderecos as $endereço){
72             echo "<p>[ $endereço->id , $endereço->logradouro, $endereço->bairro, $endereço->cidade, $endereço->uf]</p>";
73         }
74     }
75     echo "<br>";
76 }
77
78
79 public function showvenda(){
80     $clientes = Cliente::all();
81
82     foreach($clientes as $cliente){
83         echo "<p> Cliente: $cliente->nome </p>";
84         echo "<p> Debito: $cliente->debito </p>";
85         echo "<p> Endereço: $cliente->endereco </p>";
86
87         $vendas = $cliente->venda()->get();
88
89         if(count($vendas)>0){
90             echo "<h1>Venda</h1>";
91             foreach($vendas as $venda){
92                 echo "<p>[ $venda->id , $venda->datavenda, $venda->descontototal, $venda->descontoacerto, $venda->valortotal]</p>";
93             }
94         }
95         echo "<br>";
96     }
97 }
98 }
```


- FornecedorController.php

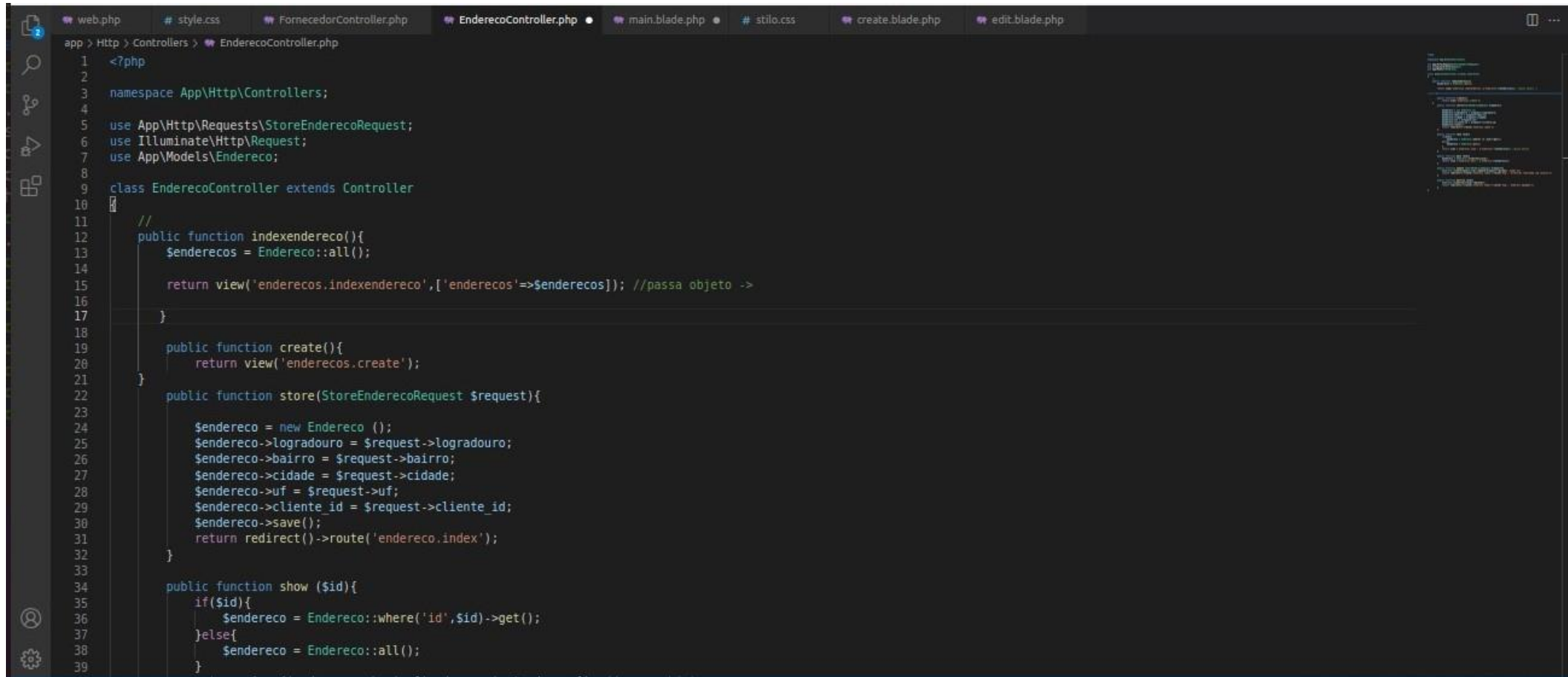


```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Http\Requests\StoreFornecedorRequest;
6 use Illuminate\Http\Request;
7 use App\Models\Fornecedor;
8
9 class FornecedorController extends Controller
10 {
11     //
12     public function indexs(){
13         $fornecedores = Fornecedor::all();
14
15         return view('fornecedores.indexs', ['fornecedores'=>$fornecedores]); //passa objeto ->
16     }
17
18     public function create(){
19         return view('fornecedores.create');
20     }
21
22     public function store(StoreFornecedorRequest $request){
23
24         $fornecedor = new Fornecedor ();
25         $fornecedor->nomefornecedor = $request->nomefornecedor;
26         $fornecedor->endereco = $request->endereco;
27         $fornecedor->email = $request->email;
28         $fornecedor->save();
29         return redirect()->route('fornecedores.indexs');
30     }
31
32     public function show ($id){
33         if($id){
34             $fornecedor = Fornecedor::where('id',$id)->get();
35         }else{
36             $fornecedor = Fornecedor::all();
37         }
38         return view ('fornecedores.show', ['fornecedores'=>$fornecedor]); //passa objeto
39     }
40 }
```

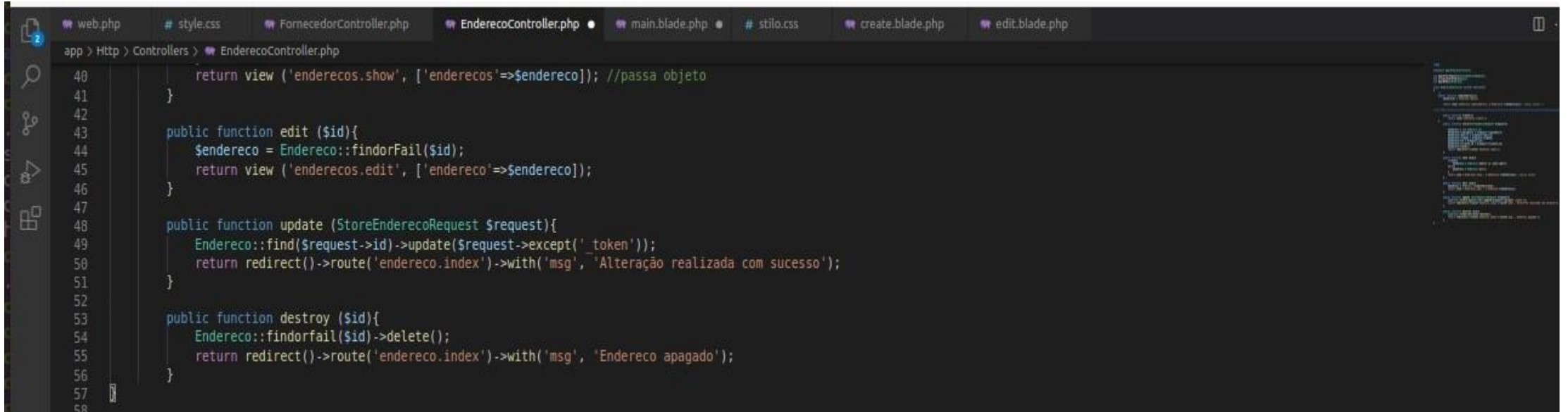


```
40
41 public function edit ($id){
42     $fornecedor = Fornecedor::findorFail($id);
43     return view ('fornecedores.edit', ['fornecedor'=>$fornecedor]);
44 }
45
46 public function update (StoreFornecedorRequest $request){
47     Fornecedor::find($request->id)->update($request->except(' token'));
48     return redirect()->route('fornecedores.indexs')->with('msg', 'Alteração realizada com sucesso');
49 }
50
51 public function destroy ($id){
52     Fornecedor::findorfail($id)->delete();
53     return redirect()->route('fornecedores.indexs')->with('msg', 'Fornecedor apagado');
54 }
55
56 public function showentrada(){
57     $fornecedores = Fornecedor::all();
58
59     foreach($fornecedores as $fornecedor){
60         echo "<p> Cliente: $fornecedor->nome </p>";
61         echo "<p> Débito: $fornecedor->debito </p>";
62         echo "<p> Endereço: $fornecedor->endereco </p>";
63
64         $entradas = $fornecedor->entrada()->get();
65
66         if(count($entradas)>0){
67             echo "<h1>Entrada</h1>";
68             foreach($entradas as $entrada){
69                 echo "<p>[$entrada->id , $entrada->valortotaldanota, $entrada->dataentrada, $entrada->fornecedor_id]</p>";
70             }
71         }
72         echo "<br>";
73     }
74 }
75
76
77
```

-EnderecoController



```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Http\Requests\StoreEnderecoRequest;
6 use Illuminate\Http\Request;
7 use App\Models\Endereco;
8
9 class EnderecoController extends Controller
10 {
11     //
12     public function indexendereco(){
13         $enderecos = Endereco::all();
14
15         return view('enderecos.indexendereco',['enderecos'=>$enderecos]); //passa objeto ->
16     }
17
18
19     public function create(){
20         return view('enderecos.create');
21     }
22
23     public function store(StoreEnderecoRequest $request){
24
25         $endereco = new Endereco ();
26         $endereco->logradouro = $request->logradouro;
27         $endereco->bairro = $request->bairro;
28         $endereco->cidade = $request->cidade;
29         $endereco->uf = $request->uf;
30         $endereco->cliente_id = $request->cliente_id;
31         $endereco->save();
32         return redirect()->route('endereco.index');
33     }
34
35     public function show ($id){
36         if($id){
37             $endereco = Endereco::where('id',$id)->get();
38         }else{
39             $endereco = Endereco::all();
40         }
41     }
42 }
```

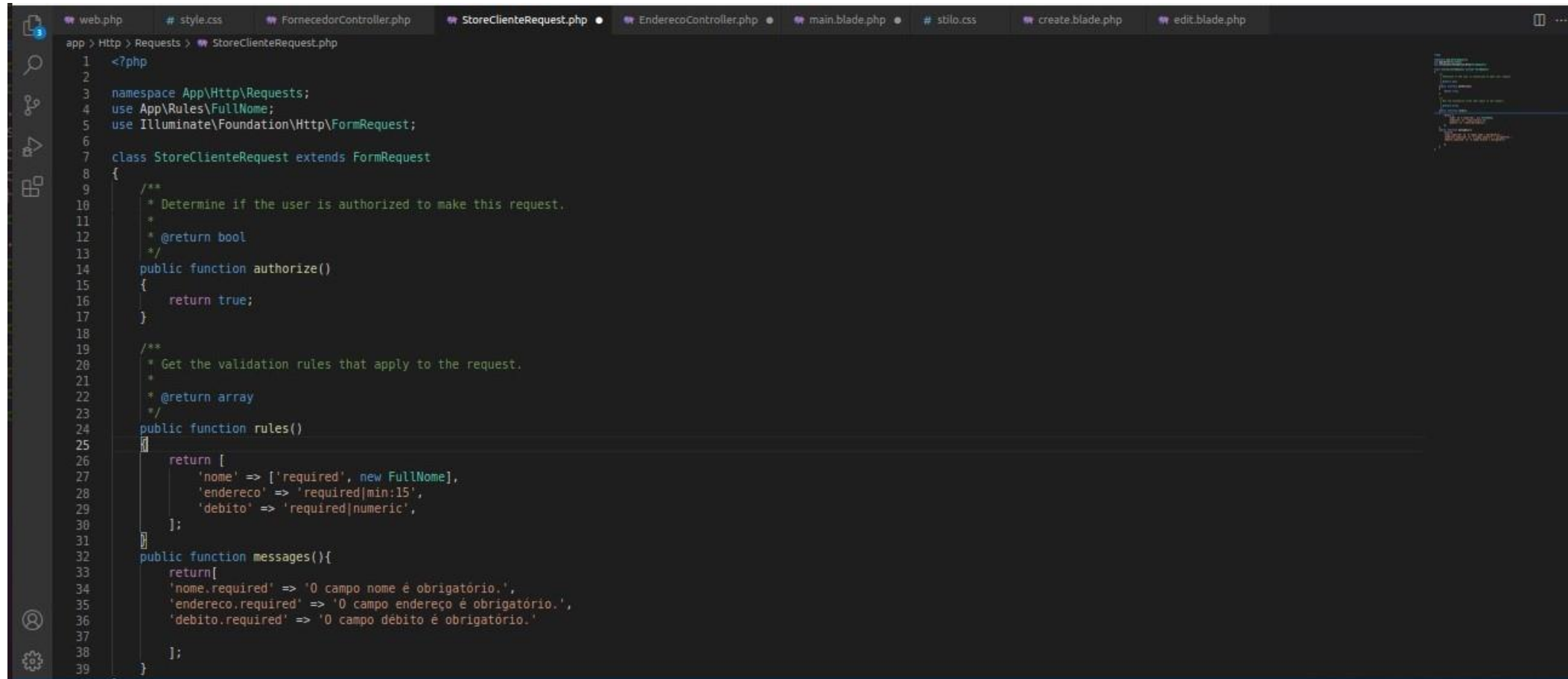


```
web.php # style.css FornecedorController.php EnderecoController.php main.blade.php stilo.css create.blade.php edit.blade.php
app > Http > Controllers > EnderecoController.php
40     return view ('enderecos.show', ['enderecos'=>$endereco]); //passa objeto
41 }
42
43 public function edit ($id){
44     $endereco = Endereco::findorFail($id);
45     return view ('enderecos.edit', ['endereco'=>$endereco]);
46 }
47
48 public function update (StoreEnderecoRequest $request){
49     Endereco::find($request->id)->update($request->except('_token'));
50     return redirect()->route('endereco.index')->with('msg', 'Alteração realizada com sucesso');
51 }
52
53 public function destroy ($id){
54     Endereco::findorfail($id)->delete();
55     return redirect()->route('endereco.index')->with('msg', 'Endereco apagado');
56 }
57
58
```

- Os demais controladores, possuem o mesmo padrão de codificação comparado aos códigos das últimas imagens.

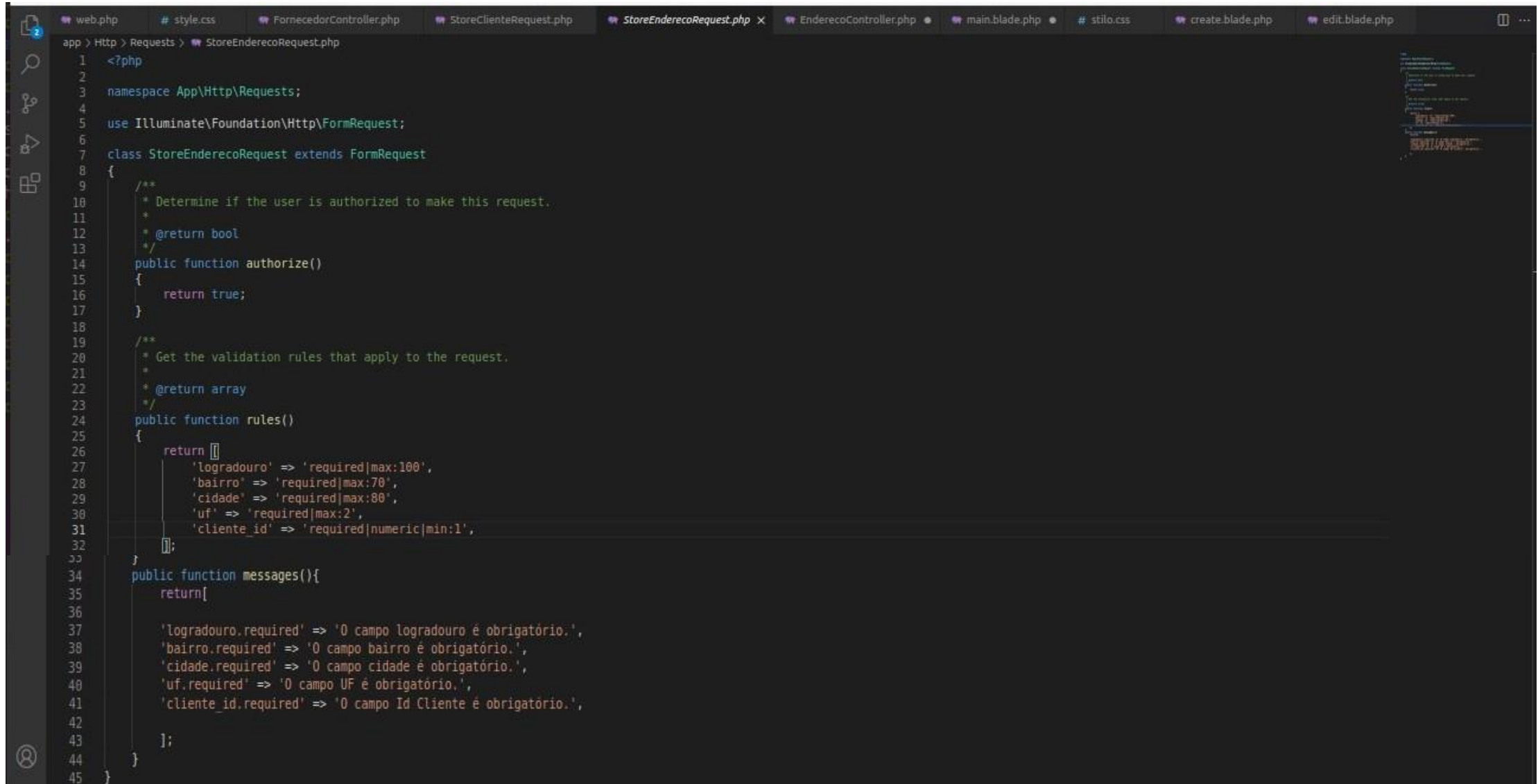
Requests:

StoreClienteRequest.php



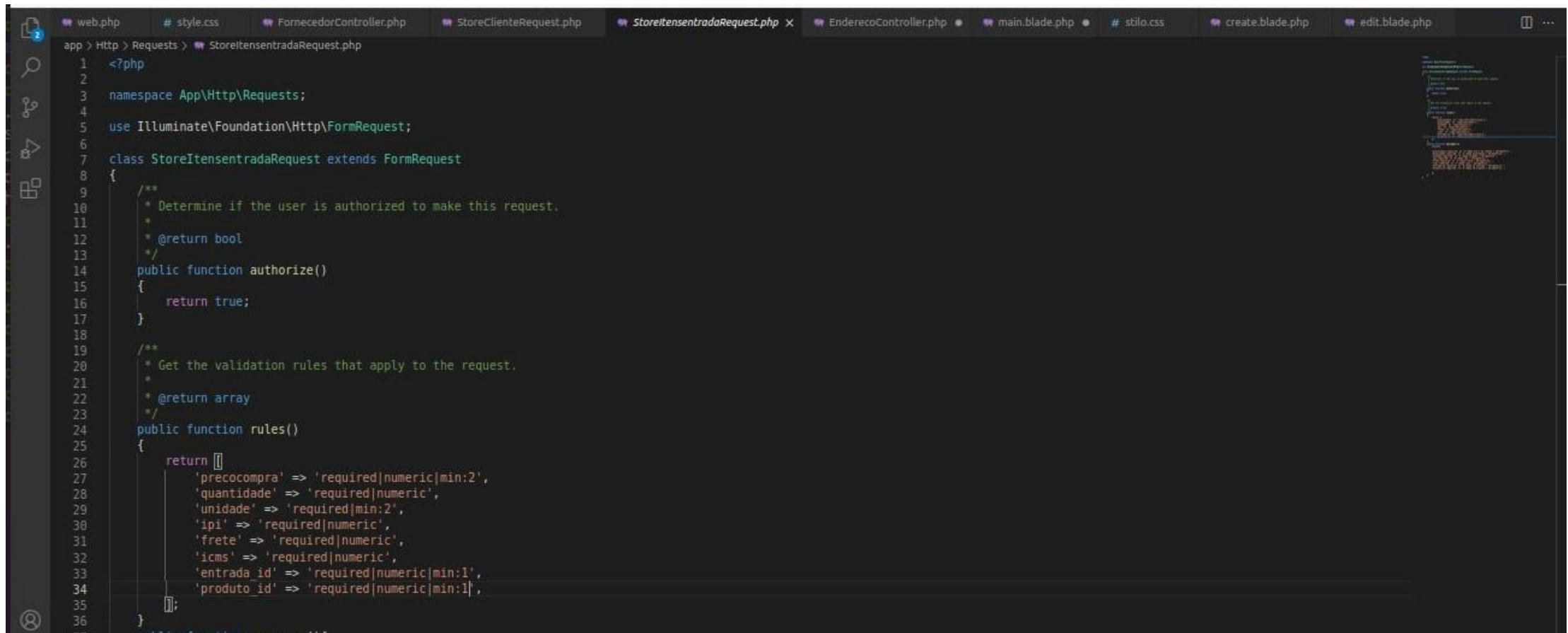
```
1 <?php
2
3 namespace App\Http\Requests;
4 use App\Rules\FullName;
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreClienteRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public 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', new FullName],
28             'endereco' => 'required|min:15',
29             'debito' => 'required|numeric',
30         ];
31     }
32
33     public function messages(){
34         return[
35             'nome.required' => 'O campo nome é obrigatório.',
36             'endereco.required' => 'O campo endereço é obrigatório.',
37             'debito.required' => 'O campo débito é obrigatório.'
38         ];
39     }
40 }
```

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|max:100',
28             'bairro' => 'required|max:70',
29             'cidade' => 'required|max:80',
30             'uf' => 'required|max:2',
31             'cliente_id' => 'required|numeric|min:1',
32         ];
33     }
34
35     public function messages(){
36         return[
37
38             'logradouro.required' => 'O campo logradouro é obrigatório.',
39             'bairro.required' => 'O campo bairro é obrigatório.',
40             'cidade.required' => 'O campo cidade é obrigatório.',
41             'uf.required' => 'O campo UF é obrigatório.',
42             'cliente_id.required' => 'O campo Id Cliente é obrigatório.',
43
44         ];
45     }
46 }
```

StoreItensEntradaRequest



```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreItensEntradaRequest 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             'precocompra' => 'required|numeric|min:2',
28             'quantidade' => 'required|numeric',
29             'unidade' => 'required|min:2',
30             'ipi' => 'required|numeric',
31             'frete' => 'required|numeric',
32             'icms' => 'required|numeric',
33             'entrada_id' => 'required|numeric|min:1',
34             'produto_id' => 'required|numeric|min:1',
35         ];
36     }
37 }
```

```
36 }
37 public function messages(){
38     return[
39
40         'precompra.required' => 'O campo preço de compra é obrigatório.',
41         'quantidade.required' => 'O campo quantidade é obrigatório.',
42         'unidade.required' => 'O campo unidade é obrigatório.',
43         'ipi.required' => 'O campo ipi é obrigatório.',
44         'frete.required' => 'O campo frete é obrigatório.',
45         'icms.required' => 'O campo icms é obrigatório.',
46         'entrada_id.required' => 'O campo Id entrada é obrigatório.',
47         'produto_id.required' => 'O campo Id produto é obrigatório.',
48
49     ];
50 }
51 }
```

- Todos os Requests atendem a codificação de validação com códigos respectivamente semelhantes a estrutura mostrada nos StoreClienteRequest, StoreEnderecoRequest e StoreItensEntradaRequest.