

Atividade VII

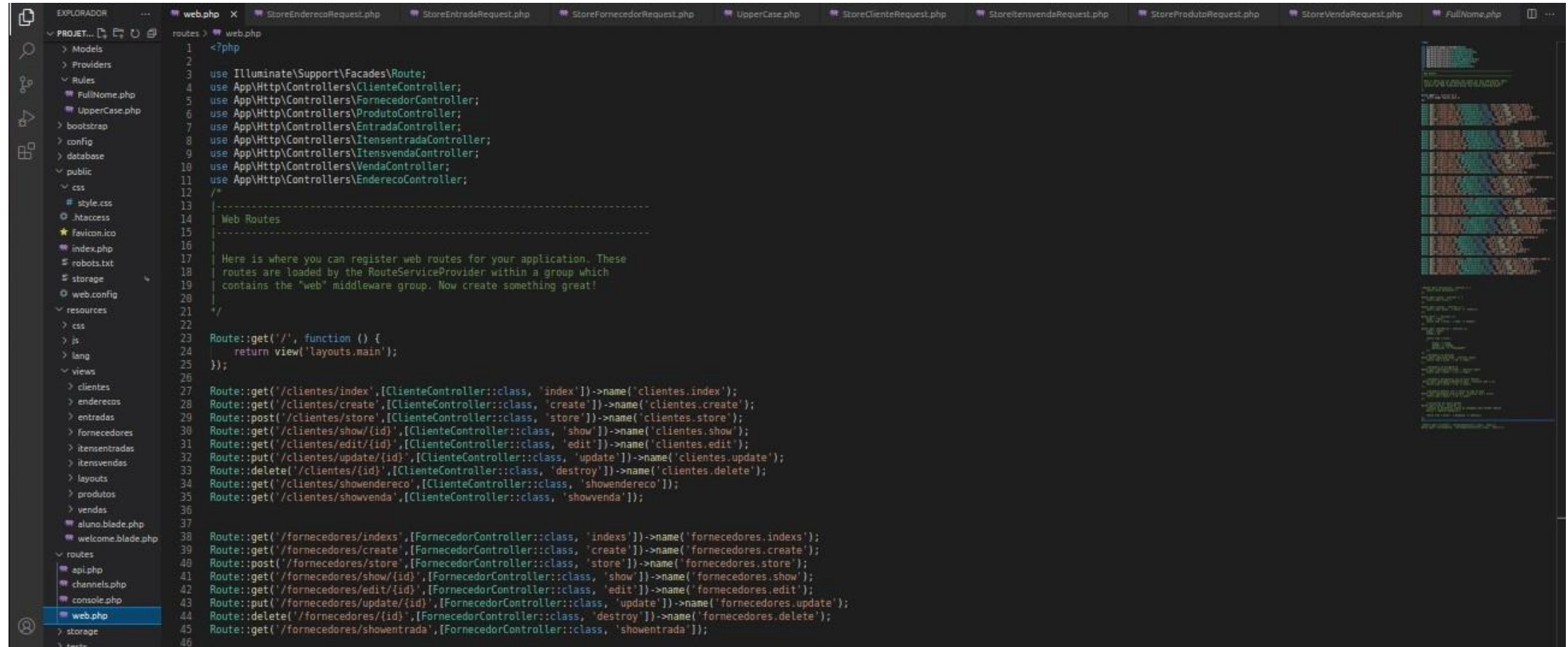
Aluno: John Junqueira Aranha

Disciplina: Laboratório de Programação Web2

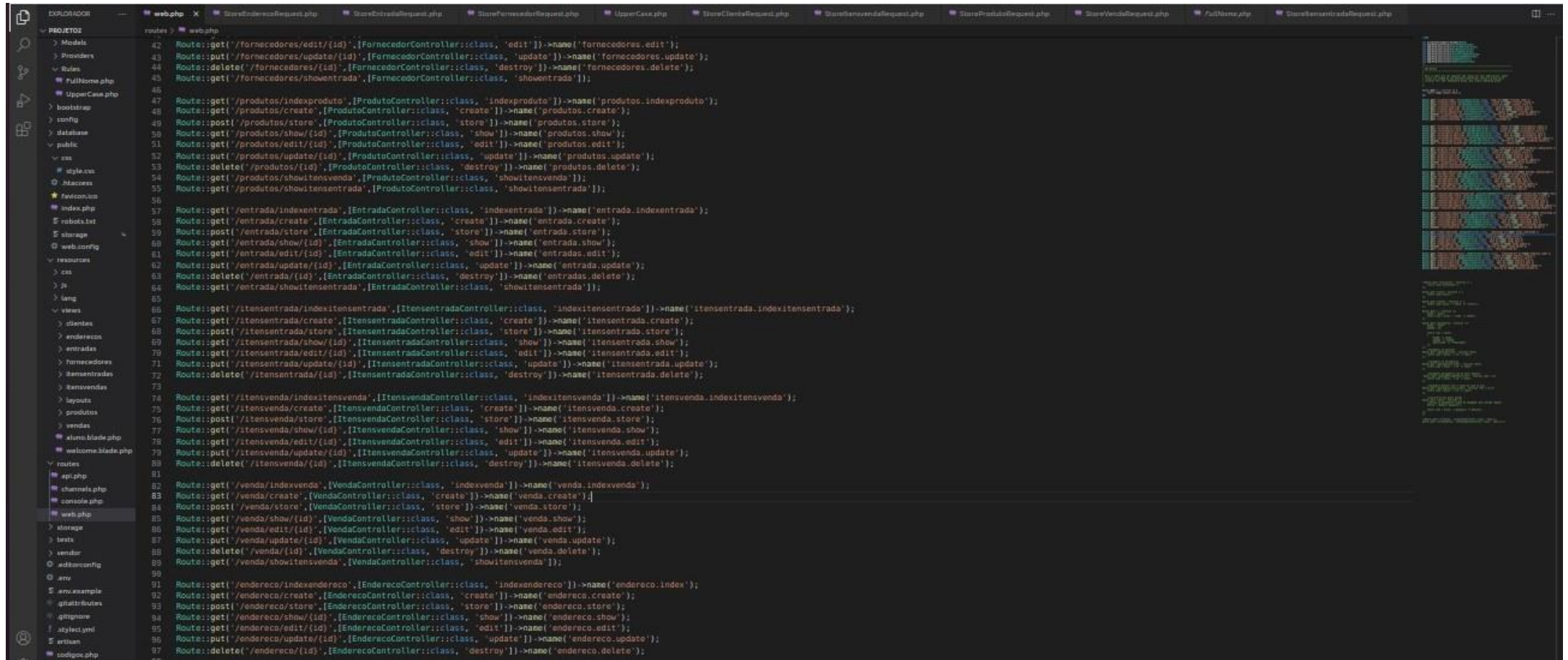
Professor: Fábio Lima

Data: 16/11/2021

Rotas



```
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::get('/clientes/index',[ClienteController::class, 'index'])->name('clientes.index');
28 Route::get('/clientes/create',[ClienteController::class, 'create'])->name('clientes.create');
29 Route::post('/clientes/store',[ClienteController::class, 'store'])->name('clientes.store');
30 Route::get('/clientes/show/{id}',[ClienteController::class, 'show'])->name('clientes.show');
31 Route::get('/clientes/edit/{id}',[ClienteController::class, 'edit'])->name('clientes.edit');
32 Route::put('/clientes/update/{id}',[ClienteController::class, 'update'])->name('clientes.update');
33 Route::delete('/clientes/{id}',[ClienteController::class, 'destroy'])->name('clientes.delete');
34 Route::get('/clientes/showendereco',[ClienteController::class, 'showendereco']);
35 Route::get('/clientes/showvenda',[ClienteController::class, 'showvenda']);
36
37
38 Route::get('/fornecedores/index',[FornecedorController::class, 'index'])->name('fornecedores.index');
39 Route::get('/fornecedores/create',[FornecedorController::class, 'create'])->name('fornecedores.create');
40 Route::post('/fornecedores/store',[FornecedorController::class, 'store'])->name('fornecedores.store');
41 Route::get('/fornecedores/show/{id}',[FornecedorController::class, 'show'])->name('fornecedores.show');
42 Route::get('/fornecedores/edit/{id}',[FornecedorController::class, 'edit'])->name('fornecedores.edit');
43 Route::put('/fornecedores/update/{id}',[FornecedorController::class, 'update'])->name('fornecedores.update');
44 Route::delete('/fornecedores/{id}',[FornecedorController::class, 'destroy'])->name('fornecedores.delete');
45 Route::get('/fornecedores/showentrada',[FornecedorController::class, 'showentrada']);
46
```

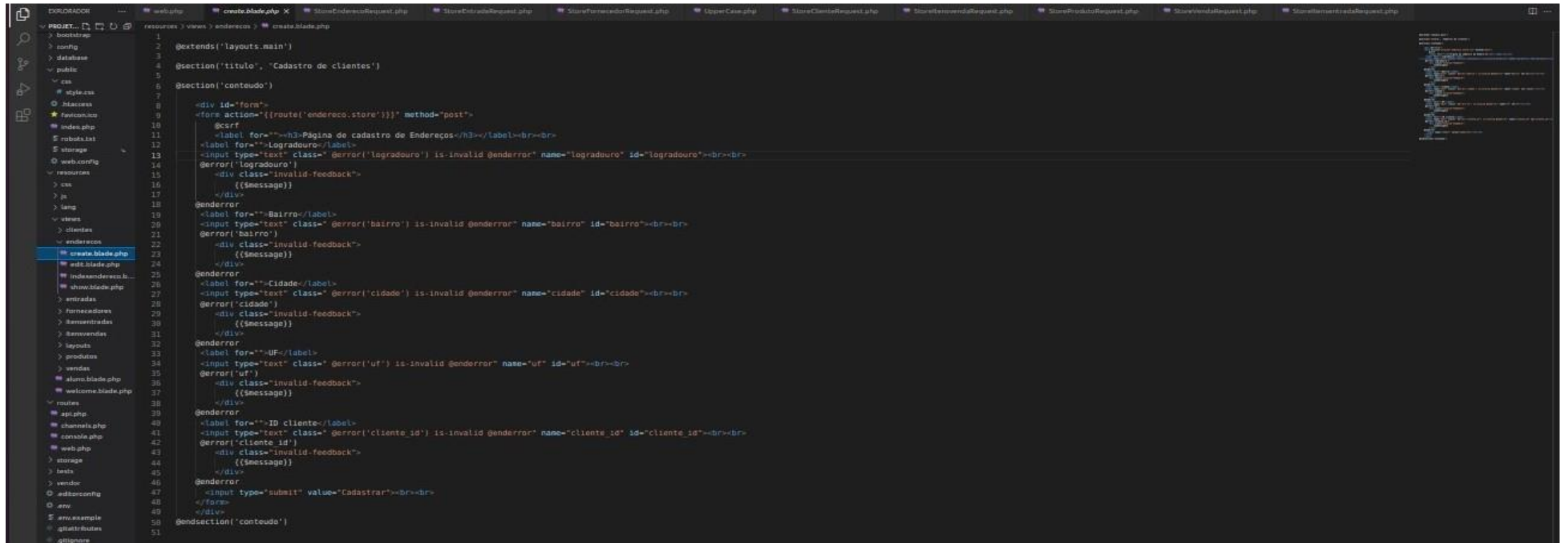


Layouts.main

The screenshot shows a web development IDE with the following components:

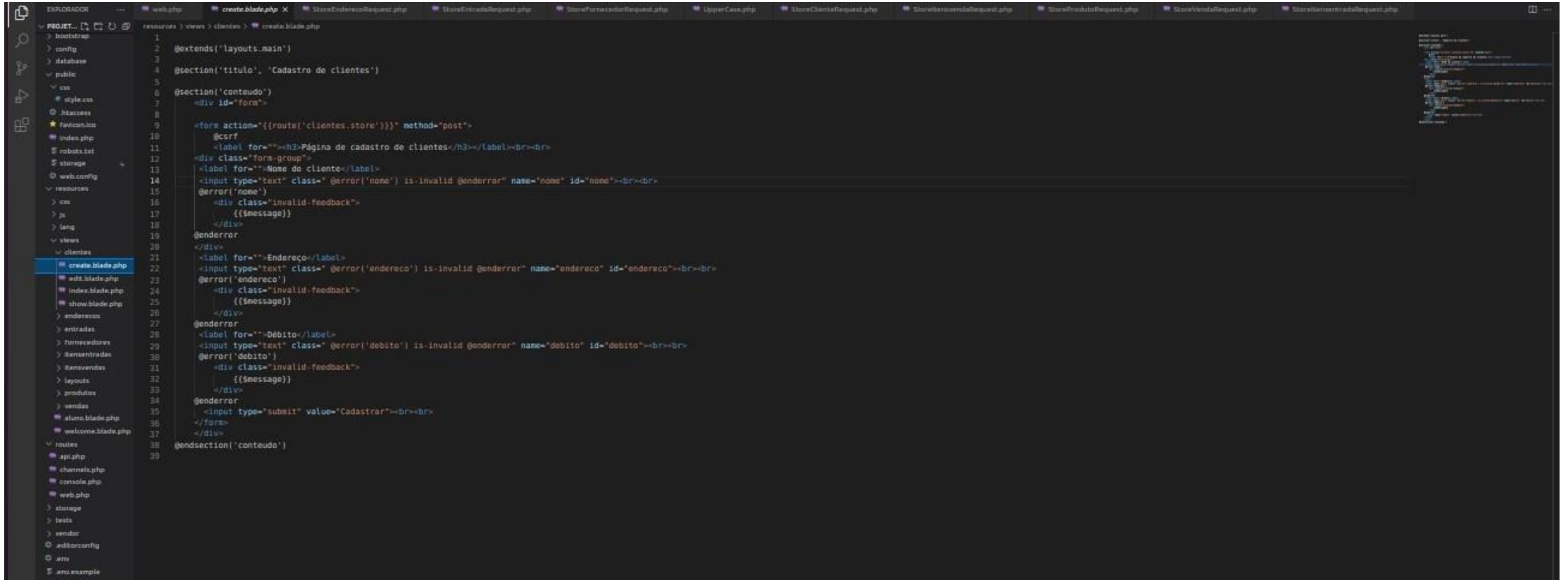
- File Explorer (Left):** Displays a project structure with folders like `config`, `database`, `public`, and `resources`. The `main.blade.php` file is selected.
- Code Editor (Center):** Shows the content of `main.blade.php`, which is an HTML template using Bootstrap 5.1.3 and jQuery. The template includes a header, a navigation bar, a main content area, and a footer.
- Preview Window (Right):** Shows the rendered HTML output of the `main.blade.php` file.

Create Endereço



```
1 @extends('layouts.main')
2
3 @section('titulo', 'Cadastro de clientes')
4
5 @section('conteudo')
6
7     <div id="form">
8         <form action="{{route('enderecos.store')}}" method="post">
9             @csrf
10             <label for="">Página de cadastro de Endereços</label><br><br>
11             <label for="">Logradouro</label>
12             <input type="text" class="@error('logradouro') is-invalid @enderror" name="logradouro" id="logradouro"><br>
13             @error('logradouro')
14                 <div class="invalid-feedback">
15                     {{ $message }}
16                 </div>
17             @enderror
18             <label for="">Bairro</label>
19             <input type="text" class="@error('bairro') is-invalid @enderror" name="bairro" id="bairro"><br>
20             @error('bairro')
21                 <div class="invalid-feedback">
22                     {{ $message }}
23                 </div>
24             @enderror
25             <label for="">Cidade</label>
26             <input type="text" class="@error('cidade') is-invalid @enderror" name="cidade" id="cidade"><br>
27             @error('cidade')
28                 <div class="invalid-feedback">
29                     {{ $message }}
30                 </div>
31             @enderror
32             <label for="">UF</label>
33             <input type="text" class="@error('uf') is-invalid @enderror" name="uf" id="uf"><br>
34             @error('uf')
35                 <div class="invalid-feedback">
36                     {{ $message }}
37                 </div>
38             @enderror
39             <label for="">ID cliente</label>
40             <input type="text" class="@error('cliente_id') is-invalid @enderror" name="cliente_id" id="cliente_id"><br>
41             @error('cliente_id')
42                 <div class="invalid-feedback">
43                     {{ $message }}
44                 </div>
45             @enderror
46             <input type="submit" value="Cadastrar"><br></form>
47         </div>
48     @endsection('conteudo')
```


Create Clientes

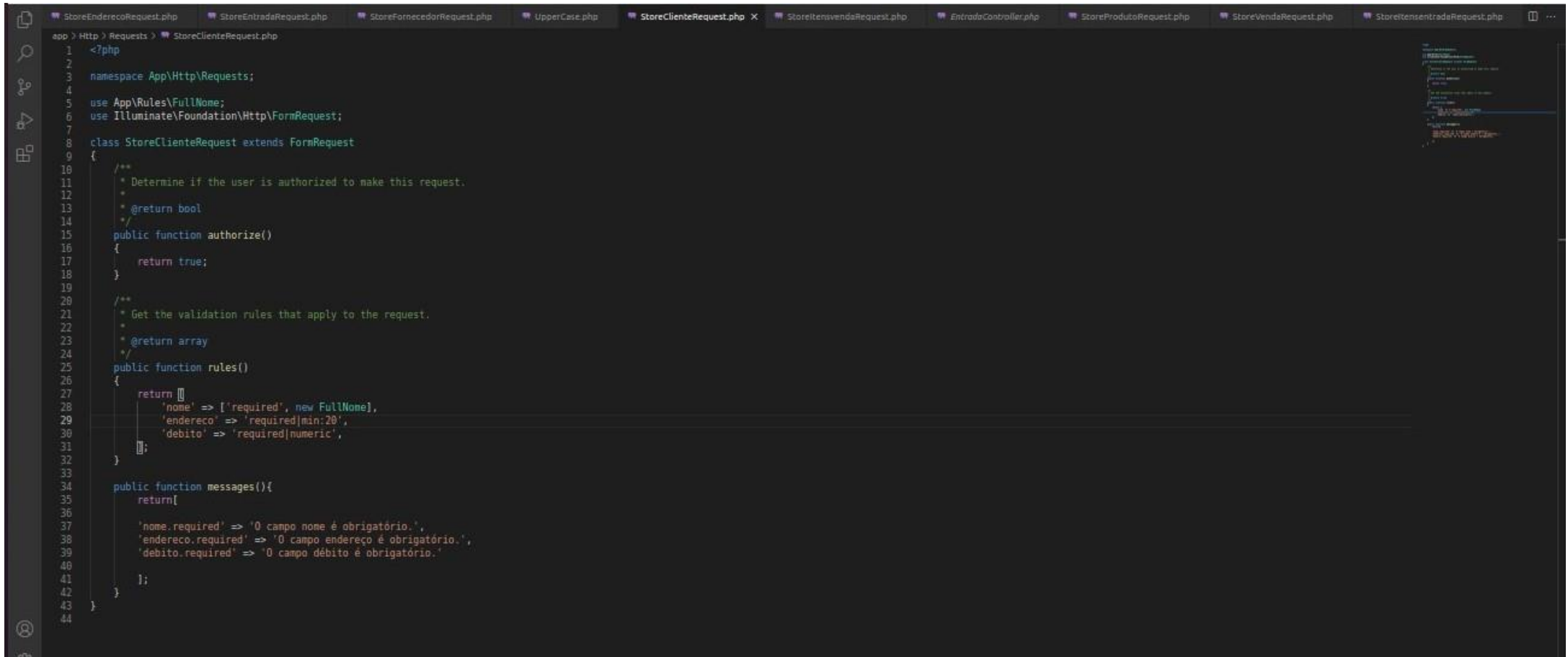


```
1 @extends('layouts.main')
2
3 @section('titulo', 'Cadastro de clientes')
4
5 @section('conteudo')
6     <div id="form">
7
8         <form action="{{route('clientes.store')}}" method="post">
9             @csrf
10             <label for=""><h3>Página de cadastro de clientes</h3></label><br><br>
11             <div class="form-group">
12                 <label for="">Nome do cliente</label>
13                 <input type="text" class="@error('nome') is-invalid @enderror" name="nome" id="nome"><br><br>
14                 @error('nome')
15                     <div class="invalid-feedback">
16                         {{ $message }}
17                     </div>
18                 @enderror
19             </div>
20             <label for="">Endereço</label>
21             <input type="text" class="@error('endereco') is-invalid @enderror" name="endereco" id="endereco"><br><br>
22             @error('endereco')
23                 <div class="invalid-feedback">
24                     {{ $message }}
25                 </div>
26             @enderror
27             <label for="">Débito</label>
28             <input type="text" class="@error('debito') is-invalid @enderror" name="debito" id="debito"><br><br>
29             @error('debito')
30                 <div class="invalid-feedback">
31                     {{ $message }}
32                 </div>
33             @enderror
34             <input type="submit" value="Cadastrar"><br><br>
35         </form>
36     </div>
37 @endsection('conteudo')
```

Os demais formulários, seguem o mesmo padrão de create clientes e endereço.

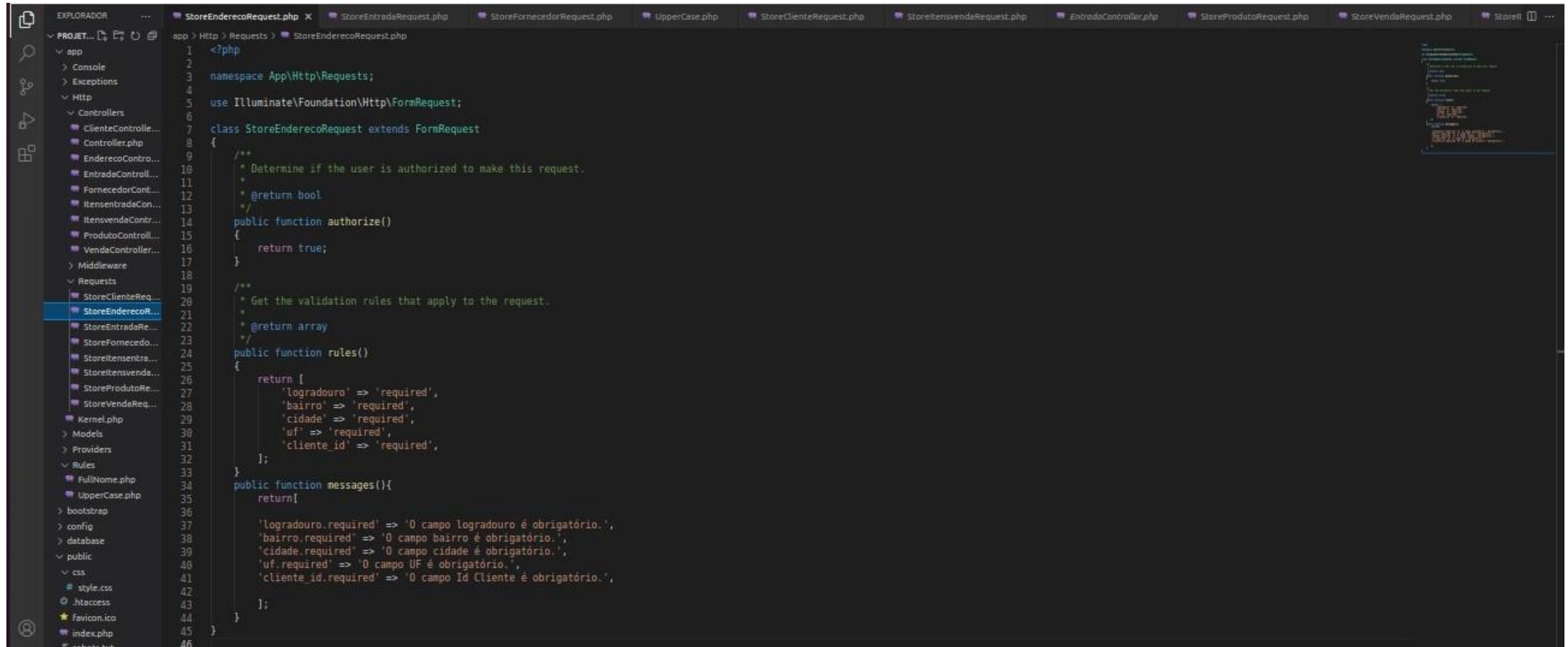
Requests:

StoreClienteRequest



```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use App\Rules\FullName;
6 use Illuminate\Foundation\Http\FormRequest;
7
8 class StoreClienteRequest extends FormRequest
9 {
10     /**
11      * Determine if the user is authorized to make this request.
12      *
13      * @return bool
14      */
15     public function authorize()
16     {
17         return true;
18     }
19
20     /**
21      * Get the validation rules that apply to the request.
22      *
23      * @return array
24      */
25     public function rules()
26     {
27         return [
28             'nome' => ['required', new FullName],
29             'endereco' => 'required|min:20',
30             'debito' => 'required|numeric',
31         ];
32     }
33
34     public function messages(){
35         return[
36
37             'nome.required' => 'O campo nome é obrigatório.',
38             'endereco.required' => 'O campo endereço é obrigatório.',
39             'debito.required' => 'O campo débito é obrigatório.'
40
41         ];
42     }
43 }
44
```

StoreEnderecoRequest

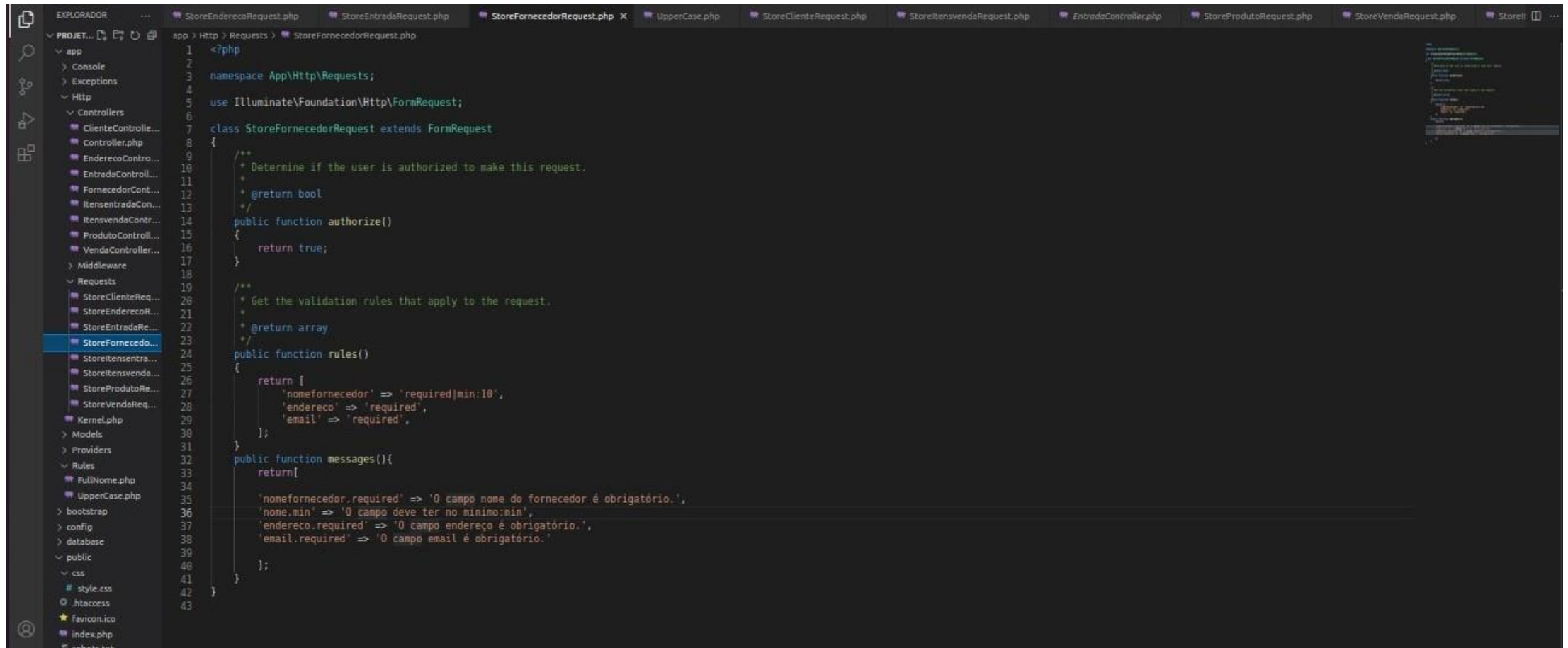


```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreEnderecoRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
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',
31             'cliente_id' => 'required',
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 }
```


StoreEntradaRequest

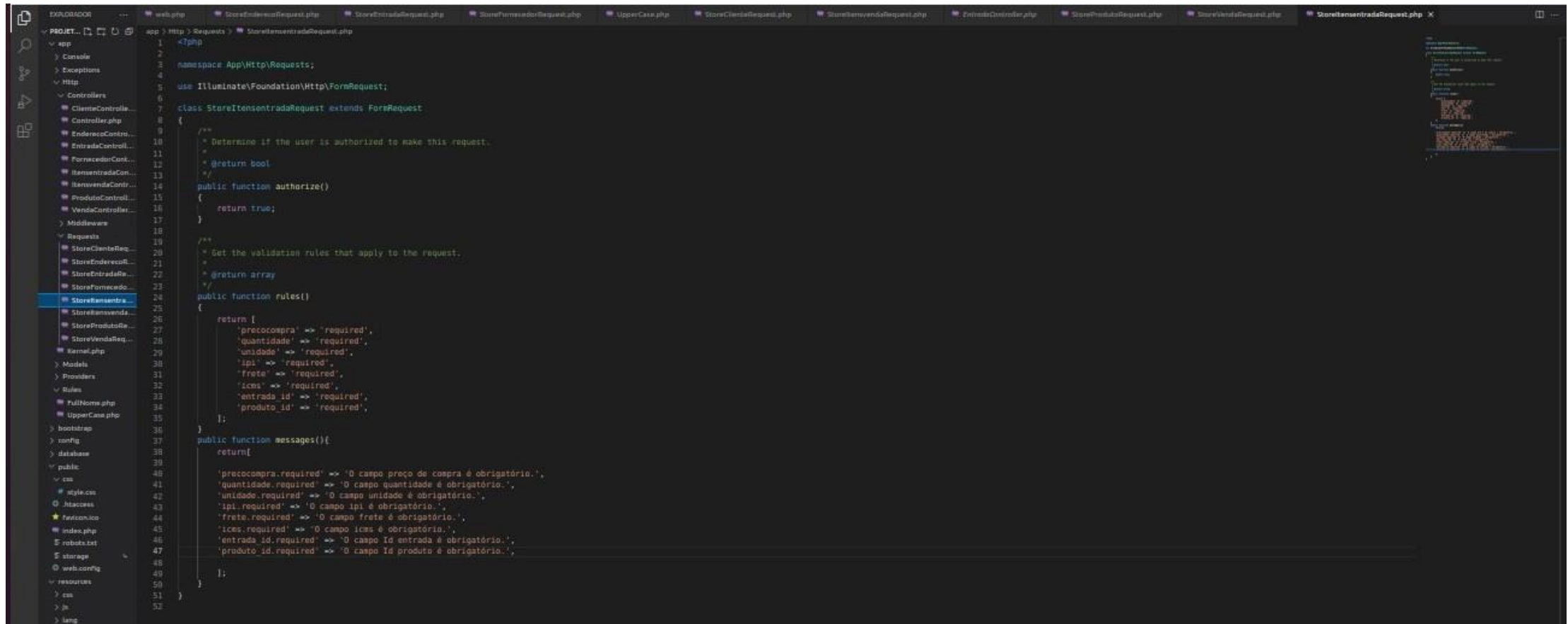
```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreEntradaRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
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             'valortotaldanota' => 'required',
28             'dataentrada' => 'required',
29             'fornecedor_id' => 'required',
30         ];
31     }
32
33     public function messages(){
34         return[
35             'valortotaldanota.required' => 'O campo Valor total da nota é obrigatório.',
36             'dataentrada.required' => 'O campo data entrada é obrigatório.',
37             'fornecedor_id.required' => 'O campo Id fornecedor é obrigatório.'
38         ];
39     }
40 }
41
42
```

StoreFornecedorRequest



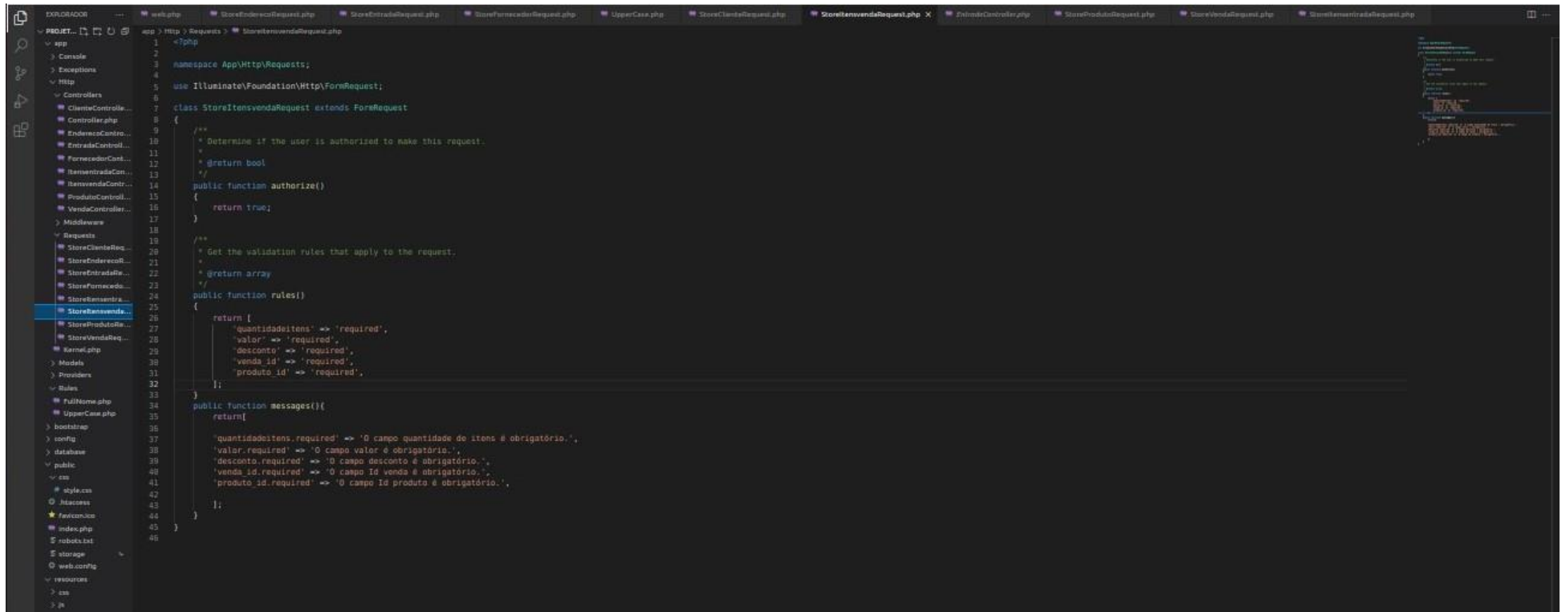
```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreFornecedorRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
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             'nomefornecedor' => 'required|min:10',
28             'endereco' => 'required',
29             'email' => 'required',
30         ];
31     }
32
33     public function messages(){
34         return[
35             'nomefornecedor.required' => 'O campo nome do fornecedor é obrigatório.',
36             'nome.min' => 'O campo deve ter no minimo:min',
37             'endereco.required' => 'O campo endereço é obrigatório.',
38             'email.required' => 'O campo email é obrigatório.'
39         ];
40     }
41 }
42
43
```

StoreItensentradaRequest

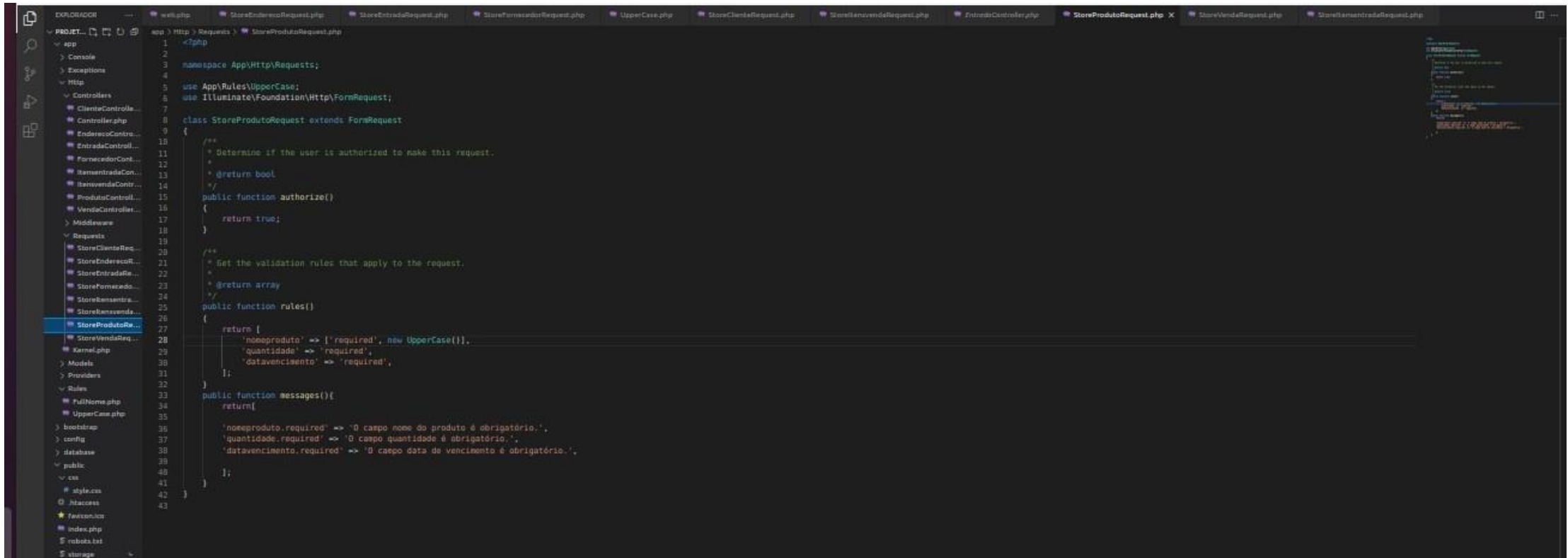


```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreItensentradaRequest extends FormRequest
8 {
9     /**
10      * Determine if the user is authorized to make this request.
11      *
12      * @return bool
13      */
14     public function authorize()
15     {
16         return true;
17     }
18
19     /**
20      * Set the validation rules that apply to the request.
21      *
22      * @return array
23      */
24     public function rules()
25     {
26         return [
27             'preco compra' => 'required',
28             'quantidade' => 'required',
29             'unidade' => 'required',
30             'ipi' => 'required',
31             'frete' => 'required',
32             'icms' => 'required',
33             'entrada_id' => 'required',
34             'produto_id' => 'required',
35         ];
36     }
37
38     public function messages()
39     {
40         return [
41             'preco compra.required' => 'O campo preço de compra é obrigatório.',
42             'quantidade.required' => 'O campo quantidade é obrigatório.',
43             'unidade.required' => 'O campo unidade é obrigatório.',
44             'ipi.required' => 'O campo ipi é obrigatório.',
45             'frete.required' => 'O campo frete é obrigatório.',
46             'icms.required' => 'O campo icms é obrigatório.',
47             'entrada_id.required' => 'O campo Id entrada é obrigatório.',
48             'produto_id.required' => 'O campo Id produto é obrigatório.',
49         ];
50     }
51 }
52
```

StoreItensvendaRequest

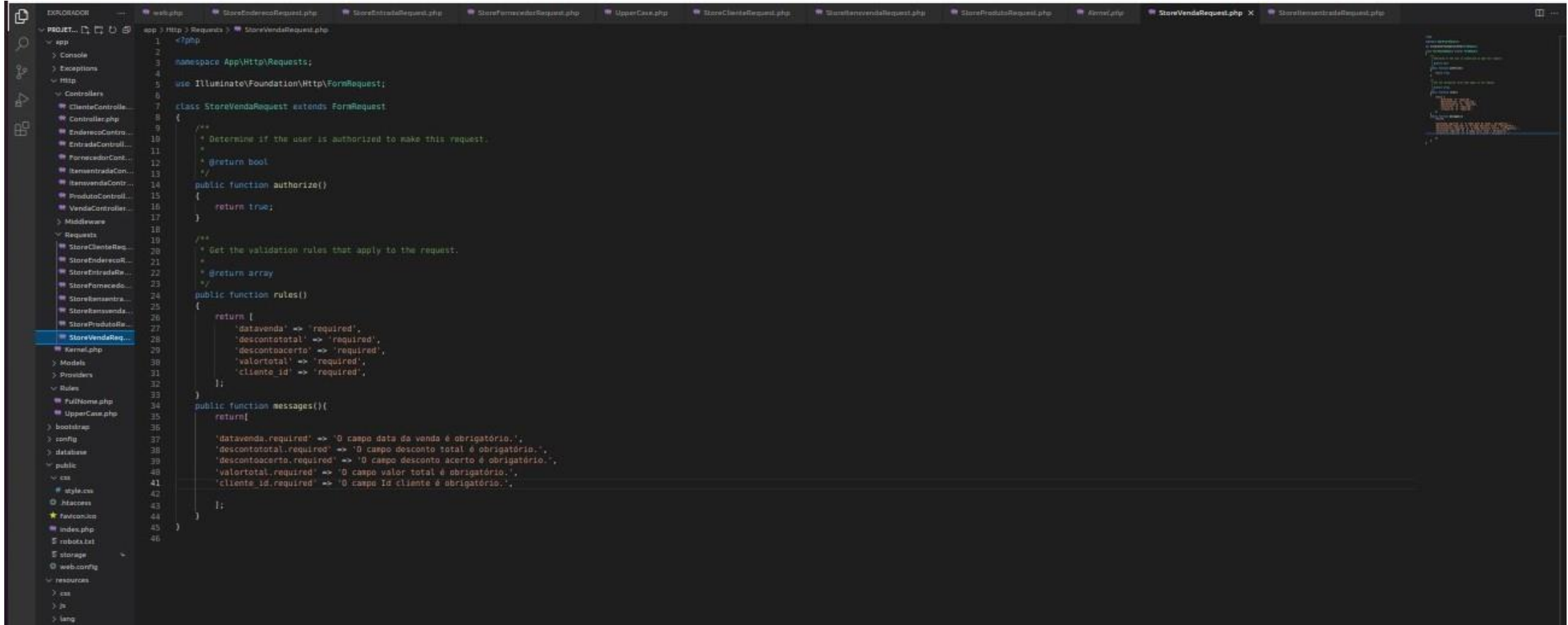


StoreProdutoRequest



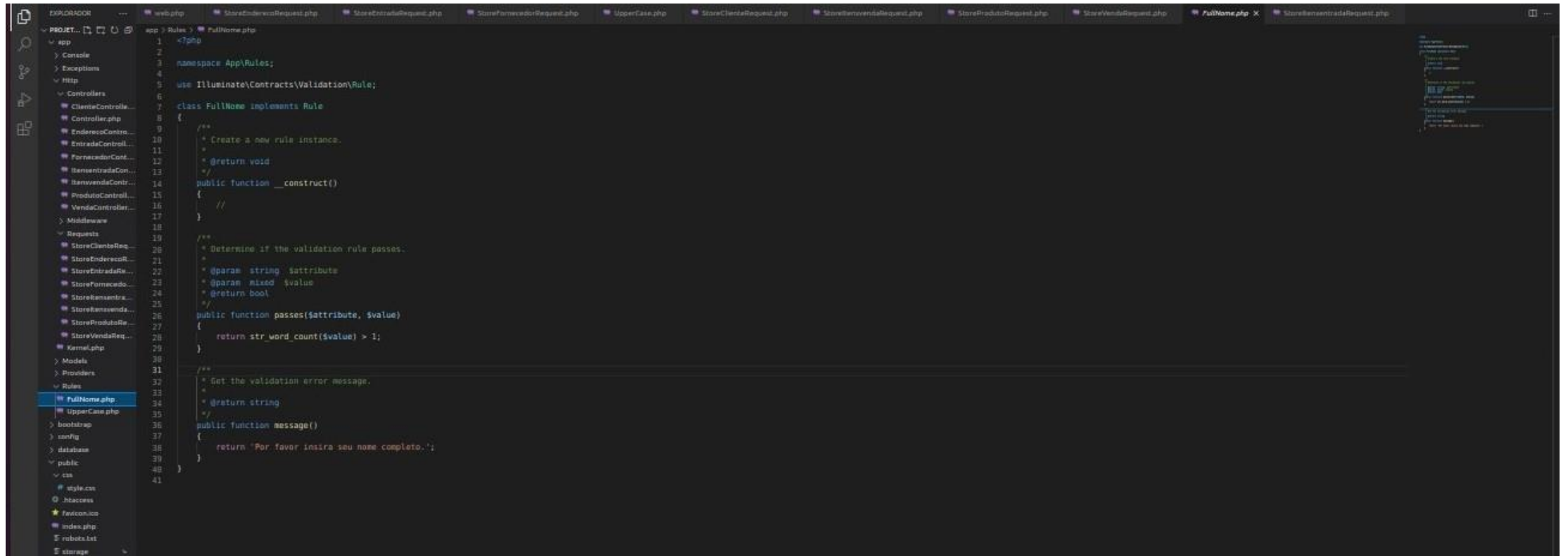
```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use App\Rules\UpperCase;
6 use Illuminate\Foundation\Http\FormRequest;
7
8 class StoreProdutoRequest extends FormRequest
9 {
10     /**
11      * Determine if the user is authorized to make this request.
12      *
13      * @return bool
14      */
15     public function authorize()
16     {
17         return true;
18     }
19
20     /**
21      * Get the validation rules that apply to the request.
22      *
23      * @return array
24      */
25     public function rules()
26     {
27         return [
28             'nomeproduto' => ['required', new UpperCase()],
29             'quantidade' => 'required',
30             'datavencimento' => 'required',
31         ];
32     }
33
34     public function messages()
35     {
36         return [
37             'nomeproduto.required' => 'O campo nome do produto é obrigatório.',
38             'quantidade.required' => 'O campo quantidade é obrigatório.',
39             'datavencimento.required' => 'O campo data de vencimento é obrigatório.',
40         ];
41     }
42 }
43
```

StoreVendaRequest



```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6
7 class StoreVendaRequest extends FormRequest
8 {
9
10     /**
11      * Determine if the user is authorized to make this request.
12      *
13      * @return bool
14      */
15     public function authorize()
16     {
17         return true;
18     }
19
20     /**
21      * Get the validation rules that apply to the request.
22      *
23      * @return array
24      */
25     public function rules()
26     {
27         return [
28             'datavenda' => 'required',
29             'descontototal' => 'required',
30             'descontoacerto' => 'required',
31             'valortotal' => 'required',
32             'cliente_id' => 'required',
33         ];
34     }
35
36     public function messages()
37     {
38         return [
39             'datavenda.required' => 'O campo data da venda é obrigatório.',
40             'descontototal.required' => 'O campo desconto total é obrigatório.',
41             'descontoacerto.required' => 'O campo desconto acerto é obrigatório.',
42             'valortotal.required' => 'O campo valor total é obrigatório.',
43             'cliente_id.required' => 'O campo Id cliente é obrigatório.',
44         ];
45     }
46 }
```

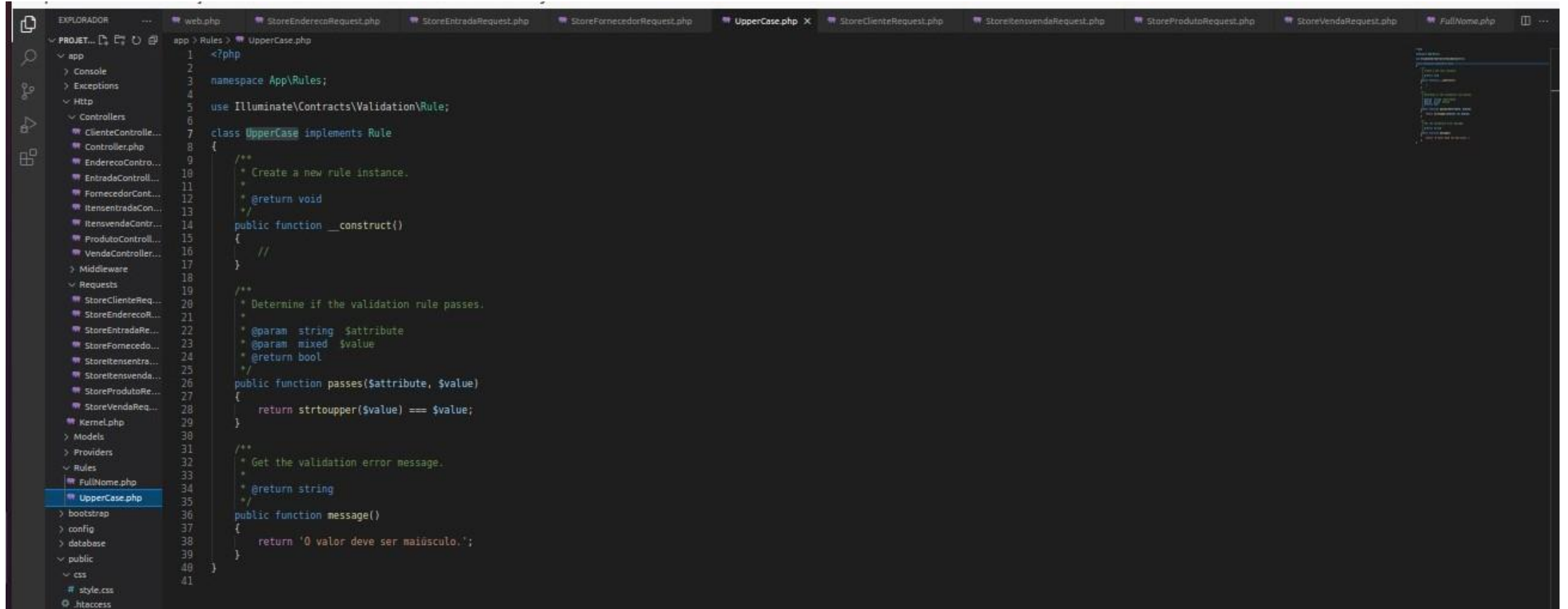

Rules: FullName



The screenshot shows a code editor with a dark theme. The left sidebar displays a file explorer with a project structure including folders for 'app', 'bootstrap', 'config', 'database', 'public', and 'css'. The 'app' folder is expanded, showing sub-folders like 'Controllers', 'Requests', 'Models', and 'Rules'. The 'Rules' folder is selected, and the file 'FullName.php' is open in the editor. The code defines a validation rule class 'FullName' that implements the 'Rule' interface from the 'illuminate/contracts/validation' namespace. The class includes a constructor, a 'passes' method that checks if the word count of the value is greater than 1, and a 'message' method that returns a custom error message in Portuguese: 'Por favor insira seu nome completo.'.

```
1 <?php
2
3 namespace App\Rules;
4
5 use Illuminate\Contracts\Validation\Rule;
6
7 class FullName 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 'Por favor insira seu nome completo.';
39     }
40 }
41
```

UpperCase



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'app', 'Controllers', 'Requests', 'Models', 'Providers', 'Rules', 'bootstrap', 'config', 'database', 'public', 'css', and 'style.css'. The 'Rules' folder is expanded, showing 'UpperCase.php' selected. The code editor displays the following PHP code:

```
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, $value)
27     {
28         return strtoupper($value) === $value;
29     }
30
31     /**
32      * Get the validation error message.
33      *
34      * @return string
35      */
36     public function message()
37     {
38         return 'O valor deve ser maiúsculo.';
39     }
40 }
41
```