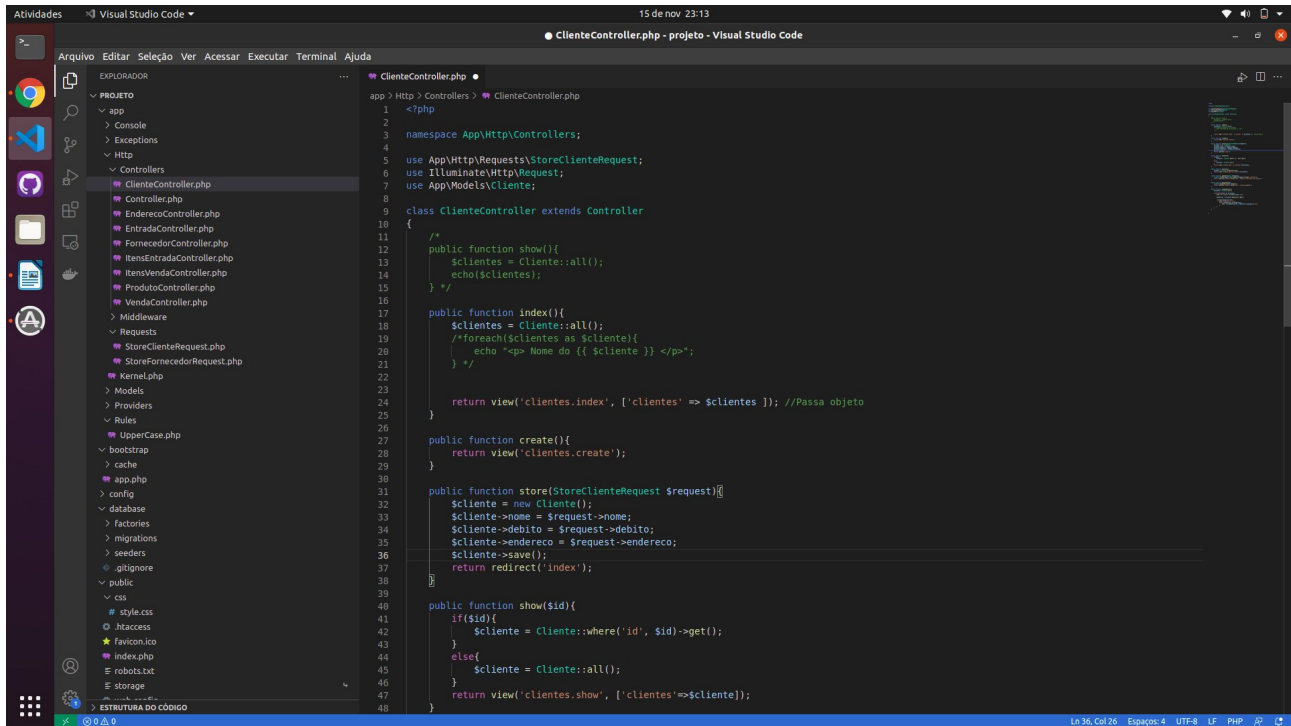


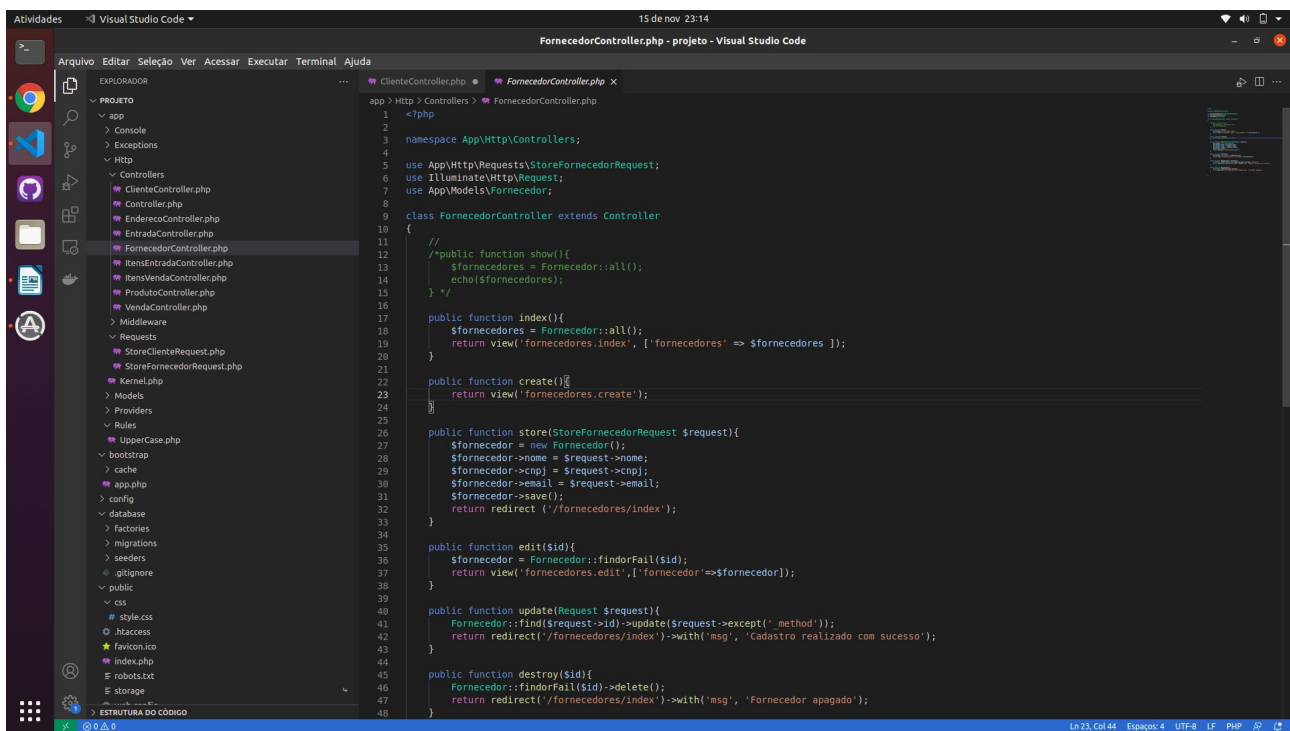
Monalyza da Silva Lima

Atividade VII

A partir do sistema desenvolvido (branch VI) Fazer validação de todos os formulários- conforme aula gravada.



```
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 show(){
13          $clientes = Cliente::all();
14          echo($clientes);
15      } */
16
17     public function index(){
18         $clientes = Cliente::all();
19         /*foreach($clientes as $cliente){
20             echo "<p> Nome do {{ $cliente }} </p>";
21         } */
22         return view('clientes.index', ['clientes' => $clientes ]); //Passa objeto
23     }
24
25     public function create(){
26         return view('clientes.create');
27     }
28
29     public function store(StoreClienteRequest $request){
30         $cliente = new Cliente();
31         $cliente->nome = $request->nome;
32         $cliente->debito = $request->debito;
33         $cliente->endereco = $request->endereco;
34         $cliente->save();
35         return redirect('index');
36     }
37
38     public function show($id){
39         if($id){
40             $cliente = Cliente::where('id', $id)->get();
41         }
42         else{
43             $cliente = Cliente::all();
44         }
45         return view('clientes.show', ['clientes'=>$cliente]);
46     }
47
48 }
```



```
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 show(){
13          $fornecedores = Fornecedor::all();
14          echo($fornecedores);
15      } */
16
17     public function index(){
18         $fornecedores = Fornecedor::all();
19         return view('fornecedores.index', ['fornecedores' => $fornecedores ]);
20     }
21
22     public function create(){
23         return view('fornecedores.create');
24     }
25
26     public function store(StoreFornecedorRequest $request){
27         $fornecedor = new Fornecedor();
28         $fornecedor->nome = $request->nome;
29         $fornecedor->cnnpj = $request->cnnpj;
30         $fornecedor->email = $request->email;
31         $fornecedor->save();
32         return redirect ('/fornecedores/index');
33     }
34
35     public function edit($id){
36         $fornecedor = Fornecedor::findorFail($id);
37         return view('fornecedores.edit', ['fornecedor'=>$fornecedor]);
38     }
39
40     public function update(Request $request){
41         $fornecedor = Fornecedor::find($request->id)->update($request->except('method'));
42         return redirect('fornecedores/index')->with('msg', 'Cadastro realizado com sucesso');
43     }
44
45     public function destroy($id){
46         $fornecedor = Fornecedor::findorFail($id)->delete();
47         return redirect('fornecedores/index')->with('msg', 'Fornecedor apagado');
48     }
49 }
```

Visual Studio Code - 15 de nov 23:26

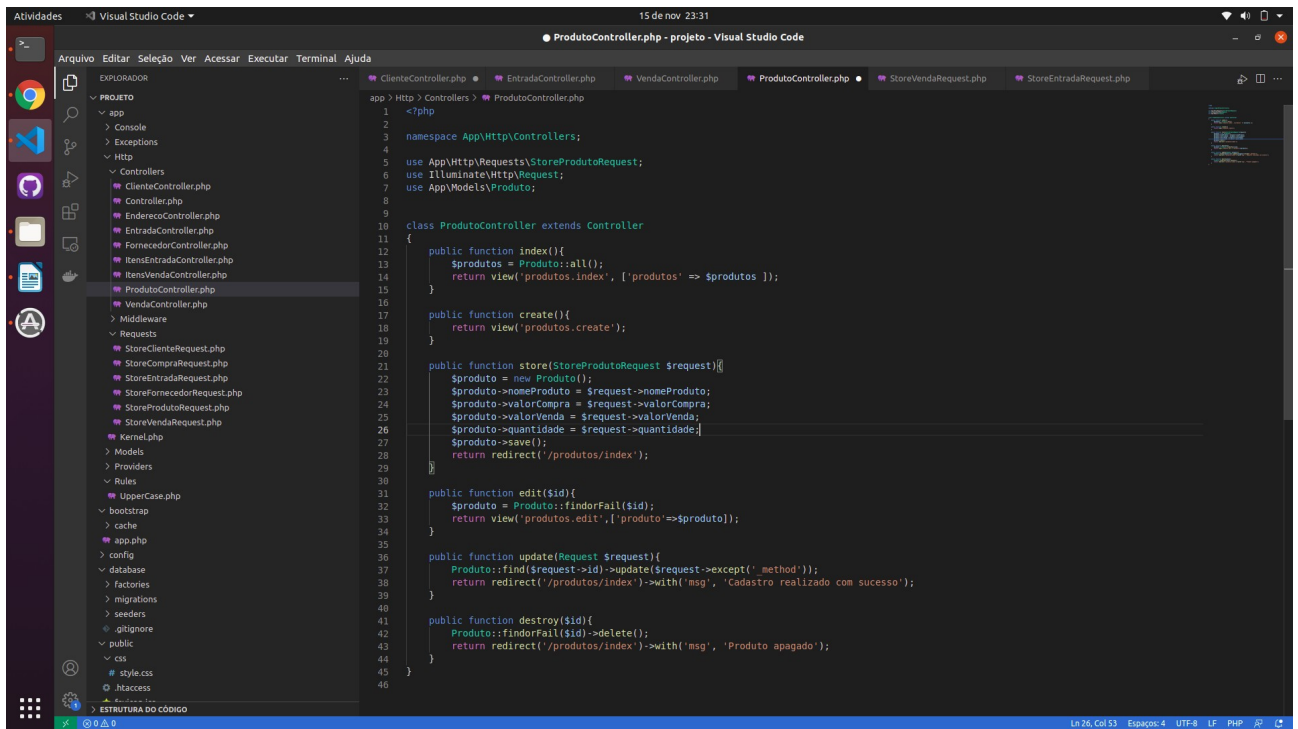
EntradaController.php - projeto - Visual Studio Code

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Http\Requests\StoreEntradaRequest;
6 use Illuminate\Http\Request;
7 use App\Models\Entrada;
8
9 class EntradaController extends Controller
10 {
11     public function index(){
12         $entradas = Entrada::all();
13         return view('entradas.index', ['entradas' => $entradas]);
14     }
15     public function create(){
16         return view('entradas.create');
17     }
18
19     public function store(StoreEntradaRequest $request){
20         $entrada = new Entrada();
21         $entrada->notaFiscal = $request->notaFiscal;
22         $entrada->valorTotal = $request->valorTotal;
23         $entrada->save();
24         return redirect('/entradas/index');
25     }
26
27     public function edit($id){
28         $entrada = Entrada::findOrFail($id);
29         return view('entradas.edit', ['entrada' => $entrada]);
30     }
31
32     public function update(Request $request){
33         $entrada = Entrada::findOrFail($request->id);
34         $entrada->update($request->except('method'));
35         return redirect('/entradas/index')->with('msg', 'Cadastro realizado com sucesso');
36     }
37
38     public function destroy($id){
39         $entrada = Entrada::findOrFail($id);
40         $entrada->delete();
41         return redirect('/entradas/index')->with('msg', 'Entrada apagada');
42     }
43
44     public function showEntrada(){
45         $entradas = Entrada::all();
46
47         foreach($entradas as $entrada){
48             echo "<p> Entrada: $entrada->notaFiscal </p>";
49         }
50     }
51 }
```

Visual Studio Code - 15 de nov 23:29

VendaController.php - projeto - Visual Studio Code

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Http\Requests\StoreVendaRequest;
6 use Illuminate\Http\Request;
7 use App\Models\Venda;
8
9 class VendaController extends Controller
10 {
11     public function index(){
12         $vendas = Venda::all();
13         return view('vendas.index', ['vendas' => $vendas]);
14     }
15     public function create(){
16         return view('vendas.create');
17     }
18
19     public function store(StoreVendaRequest $request){
20         $venda = new Venda();
21         $venda->notaFiscal = $request->notaFiscal;
22         $venda->valorTotal = $request->valorTotal;
23         $venda->save();
24         return redirect('/vendas/index');
25     }
26
27     public function edit($id){
28         $venda = Venda::findOrFail($id);
29         return view('vendas.edit', ['venda' => $venda]);
30     }
31
32     public function update(Request $request){
33         $venda = Venda::findOrFail($request->id);
34         $venda->update($request->except('method'));
35         return redirect('/vendas/index')->with('msg', 'Cadastro realizado com sucesso');
36     }
37
38     public function destroy($id){
39         $venda = Venda::findOrFail($id);
40         $venda->delete();
41         return redirect('/vendas/index')->with('msg', 'Venda apagada');
42     }
43
44     public function showVenda(){
45         $vendas = Venda::all();
46
47         foreach($vendas as $venda){
48             echo "<p> Venda: $venda->notaFiscal </p>";
49         }
50     }
51 }
```



Atividades Visual Studio Code 15 de nov 23:39

StoreEntradaRequest.php - projeto - Visual Studio Code

EXPLORADOR

- PROJETO
 - app
 - Console
 - Exceptions
 - Http
 - Controllers
 - ClienteController.php
 - Controller.php
 - EnderecoController.php
 - EntradaController.php
 - FornecedorController.php
 - ItemEntradaController.php
 - ItemVendaController.php
 - ProdutoController.php
 - VendaController.php
 - Middleware
 - Requests
 - Kernel.php
 - Models
 - Providers
 - Rules
 - UpperCase.php
 - bootstrap
 - cache
 - app.php
 - config
 - database
 - factories
 - migrations
 - seeders
 - .gitignore
 - public
 - css
 - style.css
 - .htaccess
 - favicon.ico
 - index.php
 - robots.txt
 - storage
 - web.config
 - resources
 - ...

ESTRUTURA DO CÓDIGO

```
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             'notaFiscal' => 'required|min:10',
28             'valorTotal' => 'required',
29         ];
30     }
31
32     public function messages()
33     {
34         return [
35             'notaFiscal.required' => 'Campo nota fiscal obrigatório',
36             'notaFiscal.min' => 'Campo nome deve ter no mínimo :min',
37             'valorTotal.required' => 'Campo valor total obrigatório'
38         ];
39     }
40 }
```

Ln 21, Col 7 Espaço: 4 UTF-8 LF PHP

Atividades Visual Studio Code 15 de nov 23:39

StoreClienteRequest.php - projeto - Visual Studio Code

EXPLORADOR

- PROJETO
 - app
 - Console
 - Exceptions
 - Http
 - Controllers
 - ClienteController.php
 - Controller.php
 - EnderecoController.php
 - EntradaController.php
 - FornecedorController.php
 - ItemEntradaController.php
 - ItemVendaController.php
 - ProdutoController.php
 - VendaController.php
 - Middleware
 - Requests
 - Kernel.php
 - Models
 - Providers
 - Rules
 - UpperCase.php
 - bootstrap
 - cache
 - app.php
 - config
 - database
 - factories
 - migrations
 - seeders
 - .gitignore
 - public
 - css
 - style.css
 - .htaccess
 - favicon.ico
 - index.php
 - robots.txt
 - storage
 - web.config
 - resources
 - ...

ESTRUTURA DO CÓDIGO

```
1 <?php
2
3 namespace App\Http\Requests;
4
5 use Illuminate\Foundation\Http\FormRequest;
6 use App\Rules\UpperCase;
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', 'min:10', new UpperCase()],
29             'debito' => 'required',
30             'endereco' => 'required',
31         ];
32     }
33
34     public function messages()
35     {
36         return [
37             'nome.required' => 'Campo nome obrigatório',
38             'nome.min' => 'Campo nome deve ter no mínimo :min',
39             'debito.required' => 'Campo débito obrigatório',
40             'endereco.required' => 'Campo endereço obrigatório'
41         ];
42     }
43 }
```

Ln 34, Col 32 Espaço: 4 UTF-8 LF PHP

Visual Studio Code interface showing the file explorer on the left with the project structure. The main editor displays the code for `StoreProdutoRequest.php`. The code defines a `StoreProdutoRequest` class extending `FormRequest` with `authorize()` and `rules()` methods.

```
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 false;
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             'nomeProduto' => 'required|min:10',
28             'valorCompra' => 'required',
29             'valorVenda' => 'required',
30             'quantidade' => 'required',
31         ];
32     }
33
34     public function messages()
35     {
36         return [
37             'nomeProduto.required' => "Campo nome obrigatório",
38             'nomeProduto.min' => "Campo nome deve ter no mínimo :min",
39             'valorCompra.required' => "Campo valor de compra obrigatório",
40             'valorVenda.required' => "Campo valor de venda obrigatório",
41             'quantidade.required' => "Campo quantidade obrigatório"
42         ];
43     }
44 }
45
46
```

Visual Studio Code interface showing the file explorer on the left with the project structure. The main editor displays the code for `StoreFornecedorRequest.php`. The code defines a `StoreFornecedorRequest` class extending `FormRequest` with `authorize()` and `rules()` methods.

```
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             'nome' => 'required|min:10',
28             'cnpj' => 'required',
29             'email' => 'required|email:rfc,dns',
30         ];
31     }
32
33     public function messages()
34     {
35         return [
36             'nome.required' => "Campo nome obrigatório",
37             'nome.min' => "Campo nome deve ter no mínimo :min",
38             'cnpj.required' => "Campo CNPJ obrigatório",
39             'email.required' => "Campo E-mail obrigatório"
40         ];
41     }
42 }
43
44
```


Visual Studio Code interface showing the file explorer on the left with the project structure. The main editor displays the code for `StoreVendaRequest.php`. The code defines a namespace, uses `Illuminate\Foundation\Http\FormRequest`, and implements `authorize()` and `rules()` methods. The `rules()` method includes validation rules for `notaFiscal` and `valorTotal`.

```
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 false;
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             'notaFiscal' => 'required|min:10',
28             'valorTotal' => 'required',
29         ];
30     }
31
32     public function messages()
33     {
34         return [
35             'notaFiscal.required' => 'Campo nota fiscal obrigatório',
36             'notaFiscal.min' => 'Campo nome deve ter no mínimo :min',
37             'valorTotal.required' => 'Campo valor total obrigatório'
38         ];
39     }
40 }
41
```

Visual Studio Code interface showing the file explorer on the left with the project structure. The main editor displays the code for `create.blade.php`. The code is a Blade template for creating a new client, featuring form fields for name, debt, and address, along with validation error handling and submit buttons.

```
1 <link rel="stylesheet" href="{{ URL::asset('css/style.css')}}">
2
3 @extends('layouts.main')
4
5 @section('titulo','Cadastro de clientes')
6
7 @section('conteudo')
8     <form action="{{route('cliente.store')}}" method="post">
9         @csrf
10         <div class="mb-3">
11             <label form="" class="form-label">Nome do Cliente</label>
12             <input type="text" class="form-control @error('nome') is-invalid @enderror" name="nome" id="nome">
13             @error('nome')
14                 <div class="invalid-feedback">
15                     {{ $message }}
16                 </div>
17             @enderror
18         </div>
19         <div class="mb-3">
20             <label form="" class="form-label">Débito</label>
21             <input type="text" class="form-control @error('debito') is-invalid @enderror" name="debito" id="debito">
22             @error('debito')
23                 <div class="invalid-feedback">
24                     {{ $message }}
25                 </div>
26             @enderror
27         </div>
28         <div class="mb-3">
29             <label form="" class="form-label">Endereço</label>
30             <input type="text" class="form-control" name="endereco" id="endereco">
31         </div>
32         <button type="submit" class="btn btn-primary">Cadastrar</button>
33     </form>
34     <div class="botoes">
35         <form action="{{route('cliente.index')}}">
36             <button type="submit" class="btn btn-primary">Cancelar</button>
37         </form>
38     </div>
39 @endsection('conteudo')

```

Visual Studio Code - 16 de nov 00:30

create.blade.php - projeto - Visual Studio Code

```
1 <link rel="stylesheet" href="{{ URL::asset('css/style.css')}}">
2
3 @extends('layouts.main')
4
5 @section('titulo','Cadastro de fornecedores')
6
7 @section('conteudo')
8     <form action="{{route('fornecedor.store')}}" method="post">
9         @csrf
10         <div class="mb-3">
11             <label form = "" class="form-label">Nome do fornecedor</label>
12             <input type="text" class="form-control @error('nome') is-invalid @enderror" name="nome" id="nome">
13             @error('nome')
14                 <div class="invalid-feedback">
15                     {{ $message }}
16                 </div>
17             @enderror
18         </div>
19         <div class="mb-3">
20             <label form = "" class="form-label">CNPJ</label>
21             <input type="text" class="form-control @error('cnpj') is-invalid @enderror" name="cnpj" id="cnpj">
22             @error('cnpj')
23                 <div class="invalid-feedback">
24                     {{ $message }}
25                 </div>
26             @enderror
27         </div>
28         <div class="mb-3">
29             <label form = "" class="form-label">E-mail</label>
30             <input type="text" class="form-control @error('email') is-invalid @enderror" name="email" id="email">
31             @error('email')
32                 <div class="invalid-feedback">
33                     {{ $message }}
34                 </div>
35             @enderror
36         </div>
37         <button type="submit" class="btn btn-primary">Cadastrar</button>
38     </form>
39     <div class="botoes">
40         <form action="{{route('fornecedor.index')}}">
41             <button type="submit" class="btn btn-primary">Cancelar</button>
42         </form>
43     </div>
44 @endsection('conteudo')
```

Visual Studio Code - 16 de nov 00:34

create.blade.php - projeto - Visual Studio Code

```
1 <link rel="stylesheet" href="{{ URL::asset('css/style.css')}}">
2
3 @extends('layouts.main')
4
5 @section('titulo','Cadastro de vendas')
6
7 @section('conteudo')
8     <form action="{{route('entrada.store')}}" method="post">
9         @csrf
10         <div class="mb-3">
11             <label form = "" class="form-label">Nota Fiscal</label>
12             <input type="text" class="form-control @error('notaFiscal') is-invalid @enderror" name="notaFiscal" id="notaFiscal">
13             @error('notaFiscal')
14                 <div class="invalid-feedback">
15                     {{ $message }}
16                 </div>
17             @enderror
18         </div>
19         <div class="mb-3">
20             <label form = "" class="form-label">Valor Total</label>
21             <input type="text" class="form-control @error('valorTotal') is-invalid @enderror" name="valorTotal" id="valorTotal">
22             @error('valorTotal')
23                 <div class="invalid-feedback">
24                     {{ $message }}
25                 </div>
26             @enderror
27         </div>
28         <div class="mb-3">
29             <label form = "" class="form-label">Nº Fornecedor</label>
30             <input type="text" class="form-control" name="fornecedor_id" id="fornecedor_id">
31         </div>
32         <button type="submit" class="btn btn-primary">Cadastrar</button>
33     </form>
34     <div class="botoes">
35         <form action="{{route('entrada.index')}}">
36             <button type="submit" class="btn btn-primary">Cancelar</button>
37         </form>
38     </div>
39 @endsection('conteudo')
```

Visual Studio Code - create.blade.php - projeto - Visual Studio Code

16 de nov 00:21

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda

EXPLORADOR

PROJETO

- index.php
- robots.txt
- storage
- web.config
- resources
 - css
 - js
 - views
 - clientes
 - endereco
 - entradas
 - fornecedores
 - create.blade.php
 - edit.blade.php
 - index.blade.php
 - itensentrada
 - itensvenda
 - layouts
 - produtos
 - vendas
 - aluno.blade.php
 - welcome.blade.php
- routes
- storage
- tests
- vendor
- editorconfig
- env
- env.example
- gitattributes
- gitignore
- styleci.yml
- artisan
- composer.json
- composer.lock
- index.html
- package.json
- phpunit.xml
- README.md

ESTRUTURA DO CÓDIGO

create.blade.php

```
1 <link rel="stylesheet" href="{{ URL::asset('css/style.css'); }}">
2
3 @extends('layouts.main')
4
5 @section('titulo','Cadastro de fornecedores')
6
7 @section('conteudo')
8     <form action="{{route('fornecedor.store')}}" method="post">
9         @csrf
10         <div class="mb-3">
11             <label form = "" class="form-label">Nome do fornecedor</label>
12             <input type="text" class="form-control @error('nome') is-invalid @enderror" name="nome" id="nome">
13             @error('nome')
14                 <div class="invalid-feedback">
15                     {{ $message }}
16                 </div>
17             @enderror
18         </div>
19         <div class="mb-3">
20             <label form = "" class="form-label">CNPJ</label>
21             <input type="text" class="form-control @error('cnpj') is-invalid @enderror" name="cnpj" id="cnpj">
22             @error('cnpj')
23                 <div class="invalid-feedback">
24                     {{ $message }}
25                 </div>
26             @enderror
27         </div>
28         <div class="mb-3">
29             <label form = "" class="form-label">E-mail</label>
30             <input type="text" class="form-control @error('email') is-invalid @enderror" name="email" id="email">
31             @error('email')
32                 <div class="invalid-feedback">
33                     {{ $message }}
34                 </div>
35             @enderror
36         </div>
37         <button type="submit" class="btn btn-primary">Cadastrar</button>
38     </form>
39     <div class="botoes">
40         <form action="{{route('fornecedor.index')}}">
41             <button type="submit" class="btn btn-primary">Cancelar</button>
42         </form>
43     </div>
44 @endsection('conteudo')
```

Visual Studio Code - create.blade.php - projeto - Visual Studio Code

16 de nov 00:30

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda

EXPLORADOR

PROJETO

- index.php
- robots.txt
- storage
- web.config
- resources
 - css
 - js
 - views
 - clientes
 - endereco
 - entradas
 - fornecedores
 - itensentrada
 - itensvenda
 - layouts
 - produtos
 - vendas
 - aluno.blade.php
 - welcome.blade.php
- routes
- storage
- tests
- vendor
- editorconfig
- env
- env.example
- gitattributes
- gitignore
- styleci.yml
- artisan
- composer.json
- composer.lock
- index.html
- package.json
- phpunit.xml
- README.md

ESTRUTURA DO CÓDIGO

create.blade.php

```
1 <link rel="stylesheet" href="{{ URL::asset('css/style.css'); }}">
2
3 @extends('layouts.main')
4
5 @section('titulo','Cadastro de vendas')
6
7 @section('conteudo')
8     <form action="{{route('venda.store')}}" method="post">
9         @csrf
10         <div class="mb-3">
11             <label form = "" class="form-label">Nota Fiscal</label>
12             <input type="text" class="form-control @error('notaFiscal') is-invalid @enderror" name="notaFiscal" id="notaFiscal">
13             @error('notaFiscal')
14                 <div class="invalid-feedback">
15                     {{ $message }}
16                 </div>
17             @enderror
18         </div>
19         <div class="mb-3">
20             <label form = "" class="form-label">Valor Total</label>
21             <input type="text" class="form-control @error('valorTotal') is-invalid @enderror" name="valorTotal" id="valorTotal">
22             @error('valorTotal')
23                 <div class="invalid-feedback">
24                     {{ $message }}
25                 </div>
26             @enderror
27         </div>
28         <button type="submit" class="btn btn-primary">Cadastrar</button>
29     </form>
30     <div class="botoes">
31         <form action="{{route('venda.index')}}">
32             <button type="submit" class="btn btn-primary">Cancelar</button>
33         </form>
34     </div>
35 @endsection('conteudo')
```