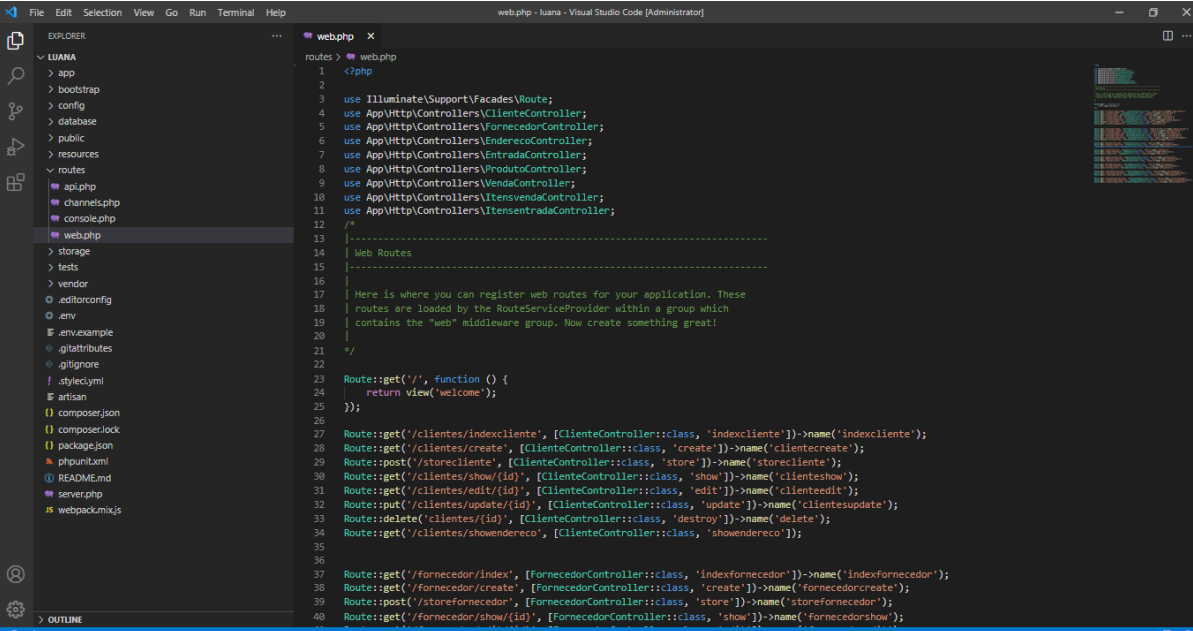
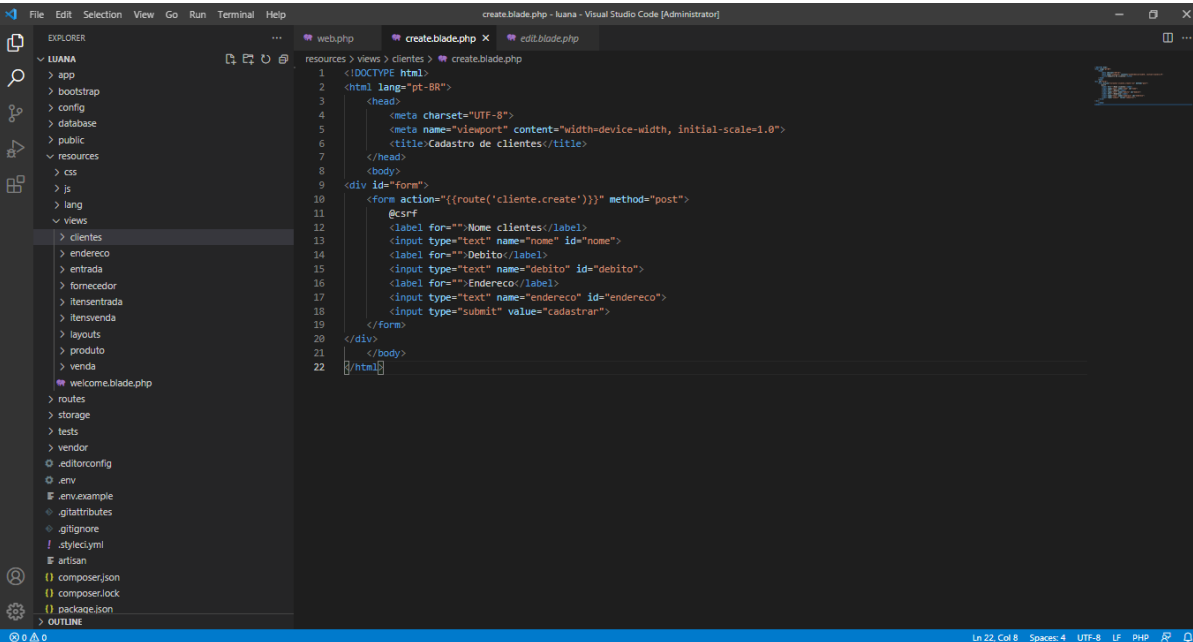


1 - Corrija o arquivo de rotas nomeando as rotas exemplo:



```
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\EnderecoController;
7 use App\Http\Controllers\EntradaController;
8 use App\Http\Controllers\ProdutoController;
9 use App\Http\Controllers\VendaController;
10 use App\Http\Controllers\ItensentradaController;
11 use App\Http\Controllers\ItensvendaController;
12
13 /*
14 |-----|
15 | Web Routes
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('welcome');
25 });
26
27 Route::get('/clientes/indexcliente', [ClienteController::class, 'indexcliente'])->name('indexcliente');
28 Route::get('/clientes/create', [ClienteController::class, 'create'])->name('clientecreate');
29 Route::post('/storecliente', [ClienteController::class, 'store'])->name('storecliente');
30 Route::get('/clientes/show/{id}', [ClienteController::class, 'show'])->name('clientesshow');
31 Route::get('/clientes/edit/{id}', [ClienteController::class, 'edit'])->name('clienteedit');
32 Route::put('/clientes/update/{id}', [ClienteController::class, 'update'])->name('clientesupdate');
33 Route::delete('/clientes/{id}', [ClienteController::class, 'destroy'])->name('delete');
34 Route::get('/clientes/showendereco', [ClienteController::class, 'showendereco']);
35
36
37 Route::get('/fornecedor/index', [FornecedorController::class, 'indexfornecedor'])->name('indexfornecedor');
38 Route::get('/fornecedor/create', [FornecedorController::class, 'create'])->name('fornecedorcreate');
39 Route::post('/storefornecedor', [FornecedorController::class, 'store'])->name('storefornecedor');
40 Route::get('/fornecedor/show/{id}', [FornecedorController::class, 'show'])->name('fornecedorschow');
```

2 - Nas Views modifique as Actions substituido:



```
1 <DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Cadastro de clientes</title>
7 </head>
8 <body>
9
10 <div id="form">
11     <form action="{{route('cliente.create')}}" method="post">
12         @csrf
13         <label for="">Nome clientes</label>
14         <input type="text" name="nome" id="nome">
15         <label for="">Debito</label>
16         <input type="text" name="debito" id="debito">
17         <label for="">Endereco</label>
18         <input type="text" name="endereco" id="endereco">
19         <input type="submit" value="cadastrar">
20     </form>
21 </div>
22 </body>
23 </html>
```

create.blade.php - luana - Visual Studio Code [Administrator]

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Endereco</title>
7 </head>
8 <body>
9 <div>
10 <form action="{{route('endereco.store')}}" method="post">
11 @csrf
12 <label for="logradouro">Logradouro</label>
13 <input type="text" name="logradouro" id="logradouro">
14 <label for="bairro">Bairro</label>
15 <input type="text" name="bairro" id="bairro">
16 <label for="cidade">Cidade</label>
17 <input type="text" name="cidade" id="cidade">
18 <label for="uf">UF</label>
19 <input type="text" name="uf" id="uf">
20 <input type="hidden" name="idcliente" value="{{idcliente}}>
21 <input type="submit" value="Cadastrar">
22 </form>
23 </div>
24 </body>
25 </html>
```

Ln 26, Col 8 Spaces: 4 UTF-8 LF PHP

create.blade.php - luana - Visual Studio Code [Administrator]

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Entrada</title>
7 </head>
8 <body>
9 <div>
10 <form action="{{route('entrada.store')}}" method="post">
11 @csrf
12 <label for="valortotal">Valor total</label>
13 <input type="text" name="valortotal" id="valortotal">
14 <input type="hidden" name="idfornecedor" value="{{idfornecedor}}>
15 <input type="submit" value="Cadastrar">
16 </form>
17 </div>
18 </body>
19 </html>
```

Ln 1, Col 1 Spaces: 4 UTF-8 LF PHP

Visual Studio Code interface showing the Explorer sidebar with the project structure. The file explorer is expanded to show the 'venda' directory, which contains 'create.blade.php' and 'welcome.blade.php'. The main editor displays the content of 'create.blade.php' in the 'venda' view.

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>venda</title>
7 </head>
8 <body>
9 <div id="form">
10 <form action="{{route('venda.store')}}" method="post">
11 @csrf
12 <label for="valor">Valor</label>
13 <input type="text" name="valor" id="valor">
14 <input type="hidden" name="idcliente" value="{{idcliente}}>
15 <input type="submit" value="Cadastrar">
16 </form>
17 </div>
18 </body>
19 </html>
```

Visual Studio Code interface showing the Explorer sidebar with the project structure. The file explorer is expanded to show the 'itensentrada' directory, which contains 'create.blade.php' and 'welcome.blade.php'. The main editor displays the content of 'create.blade.php' in the 'itensentrada' view.

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Itensentrada</title>
7 </head>
8 <body>
9 <div id="form">
10 <form action="{{route('itensentrada.store')}}" method="post">
11 @csrf
12 <label for="precocompra">Preço</label>
13 <input type="text" name="precocompra" id="precocompra">
14 <input type="hidden" name="identrada" value="{{identrada}}>
15 <input type="hidden" name="idproduto" value="{{idproduto}}>
16 <input type="submit" value="Cadastrar">
17 </form>
18 </div>
19 </body>
20 </html>
```

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The file explorer is expanded to the 'resources' directory, specifically the 'views' folder, where 'create.blade.php' is selected. The main editor window shows the content of 'create.blade.php' for the 'itensvenda' view. The code is a Blade template that defines an HTML form with a text input for 'valorvenda', a hidden input for 'idproduto', and a submit button labeled 'Cadastran'. The form is associated with the route 'itensvenda.store' and uses the POST method. The template also includes a CSRF token and a meta tag for the viewport.

```

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>itensvenda</title>
7 </head>
8 <body>
9 <div id="form">
10 <form action="{{route('itensvenda.store')}}" method="post">
11 @csrf
12 <label for="valorvenda">Valor </label>
13 <input type="text" name="valorvenda" id="valorvenda">
14 <input type="hidden" name="idproduto" value="{{idproduto}}>
15 <input type="hidden" name="idvenda" value="{{idvenda}}>
16 <input type="submit" value="Cadastran">
17 </form>
18 </div>
19 </html>

```

3 - Na pasta Model corrija as relações das models (venda, itensvenda, entrada, itensentrada, fornecedor, produto) conforme relação apresentada entre o model cliente e endereço

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the project structure. The file explorer is expanded to the 'app' directory, specifically the 'Models' folder, where 'Cliente.php' is selected. The main editor window shows the content of 'Cliente.php'. The code defines a 'Cliente' model that extends 'Model'. It includes a 'hasMany' relationship with 'Endereco' and a 'hasMany' relationship with 'Venda'. The 'Cliente' model also has a 'fillable' property that includes 'nome' and 'debito'.

```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Cliente extends Model
9 {
10     use HasFactory;
11
12     protected $table = "clientes";
13
14     protected $fillable = [
15         'nome',
16         'debito'
17     ];
18
19     function endereco(){
20         return $this->hasMany(Endereco::class, 'idcliente', 'id');
21     }
22     function venda(){
23         return $this->hasMany(Venda::class, 'idcliente', 'id');
24     }
25 }
26

```

This screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The file explorer shows a directory named 'LUANA' containing an 'app' directory, which in turn contains 'Models'. Under 'Models', several files are listed, including 'Endereco.php' which is currently selected. The main editor area displays the code for 'Endereco.php'. The code defines a namespace 'App\Models', uses 'Illuminate\Database\Eloquent\Factories\HasFactory' and 'Illuminate\Database\Eloquent\Model', and defines a class 'Endereco' that extends 'Model'. The class has a protected static property '\$table' set to 'endereco', a protected array '\$fillable' containing 'logradouro', 'bairro', 'cidade', 'uf', and 'idcliente', and a public function 'cliente()' that returns a query using 'belongsTo' with 'Cliente' and 'idcliente'.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Endereco extends Model
9 {
10     use HasFactory;
11
12     protected $table = "endereco";
13
14     protected $fillable = [
15         'logradouro',
16         'bairro',
17         'cidade',
18         'uf',
19         'idcliente'
20     ];
21
22     public function cliente(){
23         return $this->belongsTo(Cliente::class, 'idcliente', 'id');
24     }
25 }
26
```

This screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows the 'LUANA' directory structure, with 'app' > 'Models' > 'Fornecedor.php' selected. The main editor area displays the code for 'Fornecedor.php'. The code defines a namespace 'App\Models', uses 'Illuminate\Database\Eloquent\Factories\HasFactory' and 'Illuminate\Database\Eloquent\Model', and defines a class 'Fornecedor' that extends 'Model'. The class has a protected static property '\$table' set to 'fornecedor', a protected array '\$fillable' containing 'nome' and 'email', and a public function 'entrada()' that returns a query using 'hasMany' with 'Entrada' and 'idfornecedor'.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Fornecedor extends Model
9 {
10     use HasFactory;
11     protected $table = "fornecedor";
12
13     protected $fillable = [
14         'nome',
15         'email'
16     ];
17
18     function entrada(){
19         return $this->hasMany(Entrada::class, 'idfornecedor', 'id');
20     }
21 }
22
```

This screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows the 'LUANA' directory structure, with 'app' > 'Models' > 'Venda.php' selected. The main editor area displays the code for 'Venda.php'. The code defines a namespace 'App\Models', uses 'Illuminate\Database\Eloquent\Factories\HasFactory' and 'Illuminate\Database\Eloquent\Model', and defines a class 'Venda' that extends 'Model'. The class has a protected static property '\$table' set to 'venda', a protected array '\$fillable' containing 'valor' and 'idcliente', and two public functions: 'cliente()' which returns a query using 'belongsTo' with 'Cliente' and 'idcliente', and 'itensvenda()' which returns a query using 'hasMany' with 'Itensvenda' and 'idvenda'.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Venda extends Model
9 {
10     use HasFactory;
11
12     protected $table = "venda";
13
14     protected $fillable = [
15         'valor',
16         'idcliente'
17     ];
18
19     public function cliente(){
20         return $this->belongsTo(Cliente::class, 'idcliente', 'id');
21     }
22
23     function itensvenda(){
24         return $this->hasMany(Itensvenda::class, 'idvenda', 'id');
25     }
26 }
27
```