

Aluno: Robério Fagundes dos Santos
Disciplina: Laboratório de Programação Web2
Professor: Fábio Lima Data: 02/11/2021

-----controllers-----

<?php

namespace App\Http\Controllers;

use App\Http\Requests\StoreCategoriaRequest;

use App\Models\Categoria;

class CategoriaController extends Controller

{

//

public function create(){

 \$Categoria = Categoria::get();

 return view('categoria.create',compact('Categoria'));

}

public function edit(\$id){

 \$Categoria = Categoria::findorFail(\$id);

 return view('categoria.edit',['Categoria'=>\$Categoria]);

}

public function update(Request \$request){

 Categoria::find(\$request->id)->update(\$request->except('_token'));

 return redirect('index/categoria')->with('msg', 'alteração realizado com sucesso');

```
}
```

```
public function destroy($id)
{
    Categoria::findorFail($id)->delete();
    return redirect('categoria.index')->with('msg', 'categoria apagada');
}
```

```
public function index(){
    $Categoria = Categoria::all();
    return view('categoria.index',compact('Categoria'));
}
```

```
public function store(StoreCategoriaRequest $request){

    $Categoria = new Categoria();
    $Categoria->nome=$request->nome;
    $Categoria->timestamps=$request->timestamps;
    $Categoria->save();

}
}
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Http\Requests\StoreClienteRequest;
```

```
use Illuminate\Http\Request;
```

```
use App\Models\Cliente;
```

```
class ClienteController extends Controller
```

```
{
```

```
    //
```

```
    public function create(){
```

```
        $Cliente = Cliente::get();
```

```
        return view('clientes.create',compact('Cliente'));
    }
```

```
    public function edit($id){
```

```
        $Cliente = Cliente::findorFail($id);
```

```
        return view('clientes.edit',['Cliente'=>$Cliente]);
    }
```

```
    public function update(Request $request){
```

```
        Cliente::find($request->id)->update($request->except('_token'));
    }
```

```
        return redirect('index/clientes')->with('msg', 'alteração realizado com sucesso');
```

```
    }
```

```
    public function destroy($id)
```

```
    {
```

```
        Cliente::findorFail($id)->delete();
```

```
        return redirect('clientes.index')->with('msg', 'cliente apagada');
```

```
    }
```

```
public function index(){  
    $Cliente = Cliente::all();  
    return view('clientes.index',compact('Cliente'));  
}
```

```
public function store(StoreClienteRequest $request){  
  
    $Cliente = new Cliente();  
    $Cliente->nome=$request->nome;  
    $Cliente->cpf=$request->cpf;  
    $Cliente->rg=$request->rg;  
    $Cliente->sexo=$request->sexo;  
    $Cliente->datanascimento=$request->datanascimento;  
    $Cliente->debito=$request->debito;  
    $Cliente->save();  
}  
}
```

<?php

namespace App\Http\Controllers;

```

use Illuminate\Http\Request;
use App\Http\Requests\StoreContatoClienteRequest;

class ContatoClienteController extends Controller
{
    //
    public function create(){
        $ContatoCliente = ContatoCliente::get();
        return view('contatocliente.create',compact('Cliente'));
    }

    public function edit($id){
        $ContatoCliente = ContatoCliente::findorFail($id);
        return view('contatocliente.edit',['Cliente'=>$Cliente]);
    }

    public function update(Request $request){
        ContatoCliente::find($request->id)->update($request->except('_token'));
        return redirect('index/contatocliente')->with('msg', 'alteração realizado com
sucesso');
    }

    public function destroy($id)
    {
        ContatoCliente::findorFail($id)->delete();
        return redirect('contatocliente.index')->with('msg', 'contato cliente apagada');
    }

    public function index(){

```

```
$ContatoCliente = ContatoCliente::all();  
return view('contatocliente.index',compact('Cliente'));  
}
```

```
public function store(StoreContatoClienteRequest $request){
```

```
    $ContatoCliente = new ContatoCliente();  
    $ContatoCliente->nome=$request->nome;  
    $ContatoCliente->cpf=$request->cpf;  
    $ContatoCliente->rg=$request->rg;  
    $ContatoCliente->sexo=$request->sexo;  
    $ContatoCliente->datanascimento=$request->datanascimento;  
    $ContatoCliente->debito=$request->debito;  
    $ContatoCliente->save();  
}  
}
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Http\Requests\StoreContatoFornecedorRequest;
```

```
use Illuminate\Http\Request;
```

```
class ContatoFornecedorController extends Controller
```

```
{
```

```

//
    public function create(){
        $ContatoFornecedor = ContatoFornecedor::get();
        return view('ContatoFornecedor.create',compact('Cliente'));
    }

    public function edit($id){
        $ContatoFornecedor = ContatoFornecedor::findorFail($id);
        return view('ContatoFornecedor.edit',['Cliente'=>$Cliente]);
    }

    public function update(Request $request){
        ContatoFornecedor::find($request->id)->update($request->except('_token'));
        return redirect('index/ContatoFornecedor')->with('msg', 'alteração realizado com
sucesso');
    }

    public function destroy($id)
    {
        ContatoFornecedor::findorFail($id)->delete();
        return redirect('ContatoFornecedor.index')->with('msg', 'contato Contato
Fornecedor apagada');
    }

    public function index(){
        $ContatoFornecedor = ContatoFornecedor::all();
        return view('ContatoFornecedor.index',compact('ContatoFornecedor'));
    }

```

```

public function store(StoreContatoFornecedorRequest $request){

    $ContatoFornecedor = new ContatoCliente();
    $ContatoFornecedor->telefone_fornecedor=$telefone_fornecedor;
    $ContatoFornecedor->whatsappfornecedor=$request->whatsappfornecedor;
    $ContatoFornecedor->email_fornecedor=$request->email_fornecedor;
    $ContatoFornecedor->fornecedor_id=$request->fornecedor_id;
    $ContatoFornecedor->save();
}

}

```

<?php

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests\StoreDevolucaoRequest;

class DevolucaoController extends Controller
{

    //
    public function create(){
        $Devolucao = Devolucao::get();
        return view('Devolucao.create',compact('Devolucao'));
    }
}

```



```
}
```

```
public function edit($id){  
    $Devolucao = Devolucao::findorFail($id);  
    return view('Devolucao.edit',['Devolucao'=>$Devolucao]);  
}
```

```
public function update(Request $request){  
    Devolucao::find($request->id)->update($request->except('_token'));  
    return redirect('index/Devolucao')->with('msg', 'alteração realizado com  
sucesso');  
}
```

```
public function destroy($id)  
{  
    Devolucao::findorFail($id)->delete();  
    return redirect('Devolucao.index')->with('msg', 'contato Devolucao apagada');  
}
```

```
public function index(){  
    $Devolucao = Devolucao::all();  
    return view('Devolucao.index',compact('Devolucao'));  
}
```

```
public function store(StoreDevolucaoRequest $request){  
  
    $Devolucao = new Devolucao();  
    $Devolucao->quantidade_devolvida=$quantidade_devolvida;
```

```
        $Devolucao->datadevolucao=$request->datadevolucao;
        $Devolucao->descricao=$request->descricao;
        $Devolucao->item_vendas_id=$request->item_vendas_id;
        $Devolucao->save();
    }
}
```

<?php

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests\StoreEnderecoClienteRequest;
```

```
class EnderecoClienteController extends Controller
```

```
{
```

```
    //
```

```
    //
```

```
    public function create(){
```

```
        $EnderecoCliente = EnderecoCliente::get();
```

```
        return view('EnderecoCliente.create',compact('EnderecoCliente'));
    }
```

```
    public function edit($id){
```

```
        $EnderecoCliente = EnderecoCliente::findorFail($id);
```

```
        return view('EnderecoCliente.edit',['EnderecoCliente'=>$EnderecoCliente]);
    }
```

```
public function update(Request $request){  
    EnderecoCliente::find($request->id)->update($request->except('_token'));  
    return redirect('index/EnderecoCliente')->with('msg', 'alteração realizado com  
sucesso');
```

```
}
```

```
public function destroy($id)  
{  
    EnderecoCliente::findorFail($id)->delete();  
    return redirect('EnderecoCliente.index')->with('msg', 'Endereco Cliente apagada');  
}
```

```
public function index(){  
    $EnderecoCliente = EnderecoCliente::all();  
    return view('EnderecoCliente.index',compact('EnderecoCliente'));  
}
```

```
public function store(StoreEnderecoClienteRequest $request){  
  
    $EnderecoCliente = new EnderecoCliente();  
    $EnderecoCliente->cidade=$cidade;  
    $EnderecoCliente->estado=$request->estado;  
    $EnderecoCliente->bairro=$request->bairro;  
    $EnderecoCliente->rua=$request->rua;  
    $EnderecoCliente->numero=$request->numero;  
    $EnderecoCliente->cep=$request->cep;  
    $EnderecoCliente->cliente_id=$request->cliente_id;  
    $EnderecoCliente->save();
```

```
}  
}
```

```
<?php
```

```
namespace App\Http\Controllers;  
use App\Http\Requests\StoreEntradaRequest;  
use Illuminate\Http\Request;  
use App\Models\Entrada;  
use App\Models\Fornecedor;
```

```
class EntradaController extends Controller  
{  
    //  
    //  
    public function create(){  
        $Entrada = Entrada::get();  
        $Fornecedor = Fornecedor::get();  
        return view('Entrada.create',compact('Entrada','Fornecedor'));  
    }
```

```
    public function edit($id){  
        $Entrada = Entrada::findorFail($id);  
        return view('Entrada.edit',['Entrada'=>$Entrada]);  
    }
```

```
    public function update(Request $request){  
        Entrada::find($request->id)->update($request->except('_token'));  
        return redirect('index/Entrada')->with('msg', 'alteração realizado com sucesso');
```

```
}
```

```
public function destroy($id)
{
    EnderecoCliente::findorFail($id)->delete();
    return redirect('Entrada.index')->with('msg', 'Endereco Entrada apagada');
}
```

```
public function index(){
    $Entrada = Entrada::all();
    return view('Entrada.index',compact('Entrada'));
}
```

```
public function store(StoreEntradaRequest $request){

    $Entrada=new Entrada();
    $Entrada->valortotaldanota=$request->valortotaldanota;
    $Entrada->dataentrada=$request->dataentrada;
    $Entrada->fornecedor_id=$request->fornecedor_id;
    $Entrada->save();
}
}
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Http\Requests\StoreFormaPagamentoRequest;
```

```
use Illuminate\Http\Request;
```

```
class FormaPagamentoController extends Controller
```

```
{
```

```
    public function create(){
```

```
        $FormaPagamento = FormaPagamento::get();
```

```
        return view('FormaPagamento.create',compact('FormaPagamento'));
    }
```

```
    public function edit($id){
```

```
        $FormaPagamento=FormaPagamento::findorFail($id);
```

```
        return view('FormaPagamento.edit',['FormaPagamento'=>$FormaPagamento]);
    }
```

```
    public function update(Request $request){
```

```
        FormaPagamento::find($request->id)->update($request->except('_token'));
    }
```

```
        return redirect('index/FormaPagamento')->with('msg', 'alteração realizado com sucesso');
    }
```

```
}
```

```
    public function destroy($id)
```

```
{
```

```
        FormaPagamento::findorFail($id)->delete();
```

```
        return redirect('FormaPagamento.index')->with('msg', 'Forma Pagamento apagada');
    }
```

```
}
```

```
    public function index(){
```

```
$FormaPagamento = StoreFormaPagamento::all();  
return view('FormaPagamento.index',compact('FormaPagamento'));  
}
```

```
public function store(FormaPagamentoRequest $request){  
  
    $FormaPagamento=new FormaPagamento();  
    $FormaPagamento->tipo_pagamento=$tipo_pagamento;  
    $FormaPagamento->venda_id=$request->venda_id;  
    $FormaPagamento->save();  
}  
}
```

<?php

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests\StoreFornecedorRequest;
```

```
class FornecedorController extends Controller
```

```
{  
    public function create(){  
        $Fornecedor = Fornecedor::get();  
        return view('FormaPagamento.create',compact('FormaPagamento'));  
    }  
}
```

```
public function edit($id){
```

```
$FormaPagamento=Fornecedor::findorFail($id);  
return view('FormaPagamento.edit',['FormaPagamento'=>$FormaPagamento]);  
}
```

```
public function update(Request $request){  
    Fornecedor::find($request->id)->update($request->except('_token'));  
    return redirect('index/FormaPagamento')->with('msg', 'alteração realizado com  
sucesso');
```

```
}
```

```
public function destroy($id)  
{  
    Fornecedor::findorFail($id)->delete();  
    return redirect('FormaPagamento.index')->with('msg', 'Forma Pagamento  
apagada');
```

```
}
```

```
public function index(){  
    $Fornecedor = Fornecedor::all();  
    return view('FormaPagamento.index',compact('FormaPagamento'));  
}
```

```
public function store(StoreFornecedorRequest $request){
```

```
    $Fornecedor=new Fornecedor();  
    $Fornecedor->razaosocial=$razaosocial;  
    $Fornecedor->cnpj=$request->cnpj;  
    $Fornecedor->save();  
}
```



```
}
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\ItensEntrada;
```

```
use App\Http\Requests\StoreItensEntradaRequest;
```

```
use App\Models\Entrada;
```

```
use App\Models\Produto;
```

```
use Illuminate\Http\Request;
```

```
class ItensEntradaController extends Controller
```

```
{
```

```
    public function create(){
```

```
        $ItensEntrada = ItensEntrada::get();
```

```
        $Entrada = Entrada::get();
```

```
        $Produto = Produto::get();
```

```
        return view('ItensEntrada.create',compact('ItensEntrada','Entrada','Produto'));
```

```
    }
```

```
    public function edit($id){
```

```
        $ItensEntrada=ItensEntrada::findorFail($id);
```

```
        return view('ItensEntrada.edit',['ItensEntrada'=>$ItensEntrada]);
```

```
    }
```

```
    public function update(Request $request){
```

```
        ItensEntrada::find($request->id)->update($request->except('_token'));
```

```
        return redirect('index/ItensEntrada')->with('msg', 'alteração realdizado com  
sucesso');
```

```
}
```

```
public function destroy($id)  
{  
    ItensEntrada::findorFail($id)->delete();  
    return redirect('ItensEntrada.index')->with('msg', 'Forma Pagamento apagada');  
}
```

```
public function index(){  
    $ItensEntrada = ItensEntrada::all();  
    return view('ItensEntrada.index',compact('ItensEntrada'));  
}
```

```
public function store(StoreItensEntradaRequest $request){  
  
    $ItensEntrada=new ItensEntrada();  
    $ItensEntrada->precompra=$request->precompra;  
    $ItensEntrada->quantidade=$request->quantidade;  
    $ItensEntrada->unidade=$request->unidade;  
    $ItensEntrada->ipi=$request->ipi;  
    $ItensEntrada->frete=$request->frete;  
    $ItensEntrada->icms=$request->icms;  
    $ItensEntrada->entrada_id=$request->entrada_id;  
    $ItensEntrada->produto_id=$request->produto_id;  
    $ItensEntrada->save();  
}
```

```
}
```

```
<?php
```

```
namespace App\Http\Controllers;  
use App\Models\ItensVendas;  
use App\Models\Produto;  
use App\Models\Venda;  
use App\Http\Requests\StoreItensVendaRequest;
```

```
use Illuminate\Http\Request;
```

```
class ItensVendasController extends Controller  
{  
    public function create(){  
        $ItensVendas = ItensVendas::get();  
        $Produto = Produto::get();  
        $Venda = Venda::get();  
        return view('ItensVendas.create',compact('Produto','Venda'));  
    }
```

```
    public function edit($id){  
        $ItensVendas=ItensVendas::findorFail($id);  
        return view('ItensVendas.edit',['ItensVendas'=>$ItensVendas]);  
    }
```

```
    public function update(Request $request){  
        ItensVendas::find($request->id)->update($request->except('_token'));  
    }
```

```
        return redirect('index/ItensVendas')->with('msg', 'alteração realizado com  
sucesso');
```

```
    }
```

```
    public function destroy($id)  
{  
        ItensVendas::findorFail($id)->delete();  
        return redirect('ItensVendas.index')->with('msg', 'ItensVendas');  
    }
```

```
    public function index(){  
        $ItensVendas = ItensVendas::all();  
        return view('ItensVendas.index',compact('ItensVendas'));  
    }
```

```
    public function store(StoreItensVendaRequest $request){  
  
        $ItensVendas=new ItensVendas();  
        $ItensVendas->quantidade_itens=$request->quantidade_itens;  
        $ItensVendas->desconto=$request->desconto;  
        $ItensVendas->valor=$request->valor;  
        $ItensVendas->venda_id=$request->venda_id;  
        $ItensVendas->produto_id=$request->produto_id;  
        $ItensVendas->save();  
    }  
}
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use App\Models\Marca;
```

```
use App\Http\Requests\StoreMarcaRequest;
```

```
use Illuminate\Http\Request;
```

```
class MarcaController extends Controller
```

```
{
```

```
    public function create(){
```

```
        $Marca = Marca::get();
```

```
        return view('Marca.create',compact('Marca'));
```

```
    }
```

```
    public function edit($id){
```

```
        $Marca=Marca::findorFail($id);
```

```
        return view('Marca.edit',['Marca'=>$Marca]);
```

```
    }
```

```
    public function update(Request $request){
```

```
        Marca::find($request->id)->update($request->except('_token'));
```

```
        return redirect('index/Marca')->with('msg', 'alteração realizado com sucesso');
```

```
    }
```

```
    public function destroy($id)
```

```
    {
```

```
        Marca::findorFail($id)->delete();
```

```
        return redirect('Marca.index')->with('msg', 'Marca');
```

```
}
```

```
public function index(){  
    $Marca = Marca::all();  
    return view('Marca.index',compact('Marca'));  
}
```

```
public function store(StoreMarcaRequest $request){  
  
    $Marca=new Marca();  
    $Marca->nome=$request->nome;  
    $Marca->save();  
}  
}
```

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests\StorePagamentoRequest;
```

```
class PagamentoController extends Controller
```

```
{  
    public function create(){  
        $Pagamento = Pagamento::get();  
        return view('Pagamento.create',compact('Pagamento'));  
}
```

```
public function edit($id){  
    $Pagamento=Pagamento::findorFail($id);  
    return view('Pagamento.edit',['Pagamento'=>$Pagamento]);  
}
```

```
public function update(Request $request){  
    Pagamento::find($request->id)->update($request->except('_token'));  
    return redirect('index/Pagamento')->with('msg', 'alteração realizado com  
sucesso');  
}
```

```
public function destroy($id)  
{  
    Pagamento::findorFail($id)->delete();  
    return redirect('Pagamento.index')->with('msg', 'Pagamento');  
}
```

```
public function index(){  
    $Pagamento = Pagamento::all();  
    return view('Pagamento.index',compact('Pagamento'));  
}
```

```
public function store(StorePagamentoRequest $request){  
  
    $Pagamento=new Pagamento();  
    $Pagamento->valor_pagamento=$request->valor_pagamento;  
    $Pagamento->data_pagamento=$request->data_pagamento;
```

```
        $Pagamento->save();
    }
}
```

<?php

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests\StoreParcelaRequest;
```

```
class ParcelasController extends Controller
```

```
{
    public function create(){
        $Parcelas = Parcelas::get();
        return view('Parcelas.create',compact('Parcelas'));
    }
```

```
    public function edit($id){
        $Parcelas=Parcelas::findorFail($id);
        return view('Parcelas.edit',['Parcelas'=>$Parcelas]);
    }
```

```
    public function update(Request $request){
        Parcelas::find($request->id)->update($request->except('_token'));
        return redirect('index/Parcelas')->with('msg', 'alteração realizado com sucesso');
    }
```



```
public function destroy($id)
{
    Parcelas::findorFail($id)->delete();
    return redirect('Parcelas.index')->with('msg', 'Parcelas');
}
```

```
public function index(){
    $Parcelas = Parcelas::all();
    return view('Parcelas.index',compact('Parcelas'));
}
```

```
public function store(StoreParcelaRequest $request){

    $Parcelas=new Parcelas();
    $Parcelas->numeros_parcelas=$request->numeros_parcelas;
    $Parcelas->valor_parcelas=$request->valor_parcelas;

    $Parcelas->status=$request->status;
    $Parcelas->vencimento=$request->vencimento;
    $Parcelas->valortotalparcela=$request->valortotalparcela;
    $Parcelas->pagamento_id=$request->pagamento_id;
    $Parcelas->save();
}
}
```

<?php

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Requests\StorePrateleiraRequest;
```

```
class PrateleiraController extends Controller
```

```
{
```

```
    public function create(){
```

```
        $Prateleira = Prateleira::get();
```

```
        return view('Prateleira.create',compact('Prateleira'));
    }
```

```
    public function edit($id){
```

```
        $Prateleira=Prateleira::findorFail($id);
```

```
        return view('Prateleira.edit',['Prateleira'=>$Prateleira]);
    }
```

```
    public function update(Request $request){
```

```
        Prateleira::find($request->id)->update($request->except('_token'));
    }
```

```
        return redirect('index/Prateleira')->with('msg', 'alteração realdizado com sucesso');
    }
```

```
}
```

```
    public function destroy($id)
```

```
{
```

```
    Prateleira::findorFail($id)->delete();
```

```
    return redirect('Prateleira.index')->with('msg', 'Prateleira');
```

```
}
```

```
public function index(){  
    $Prateleira = Prateleira::all();  
    return view('Prateleira.index',compact('Prateleira'));  
}
```

```
public function store(StorePrateleiraRequest $request){  
  
    $Prateleira=new Prateleira();  
    $Prateleira->descricao=$request->descricao;  
    $Prateleira->quantidadedefrascos=$request->quantidadedefrascos;  
    $Prateleira->save();  
}  
}
```

<?php

```
namespace App\Http\Controllers;  
use App\Models\Produto;  
use App\Models\Marca;  
use App\Models\Categoria;  
use App\Models\Prateleira;  
use App\Http\Requests\StoreProdutoRequest;
```

```
use Illuminate\Http\Request;
```

```
class ProdutoController extends Controller  
{
```

```
public function create(){
    $Produto = Produto::get();
    $Marca = Marca::get();
    $Categoria = Categoria::get();
    $Prateleira = Prateleira::get();
    return view('Produto.create',compact('Produto','Marca','Categoria','Prateleira'));
}
```

```
public function edit($id){
    $Produto=Produto::findorFail($id);
    return view('Produto.edit',['Produto'=>$Produto]);
}
```

```
public function update(Request $request){
    Produto::find($request->id)->update($request->except('_token'));
    return redirect('index/Produto')->with('msg', 'alteração realdizado com sucesso');
}
```

```
public function destroy($id)
{
    Produto::findorFail($id)->delete();
    return redirect('Produto.index')->with('msg', 'Produto');
}
```

```
public function index(){
    $Produto = Produto::all();
    return view('Produto.index',compact('Produto'));
}
```

```
public function store(StoreProdutoRequest $request){

    $Produto=new Produto();

    $Produto->nome=$request->nome;
    $Produto->icms=$request->icms;

    $Produto->ipi=$request->ipi;
    $Produto->frete=$request->frete;
    $Produto->valornafabrica=$request->valornafabrica;
    $Produto->valordecompra=$request->valordecompra;
    $Produto->lucro=$request->lucro;
    $Produto->valorvenda=$request->valorvenda;
    $Produto->desconto=$request->desconto;
    $Produto->quantidade=$request->quantidade;
    $Produto->datavencimento=$request->datavencimento;
    $Produto->marca_id=$request->marca_id;
    $Produto->categoria_id=$request->categoria_id;
    $Produto->prateleira_id=$request->prateleira_id;
    $Produto->save();
}
}
```

<?php

```
namespace App\Http\Controllers;
use App\Models\Venda;
use App\Models\Cliente;
use App\Http\Requests\StoreVendaRequest;
```

```
use Illuminate\Http\Request;
```

```
class VendaController extends Controller
```

```
{
```

```
    public function create(){
```

```
        $Venda = Venda::get();
```

```
        $Cliente = Cliente::get();
```

```
        return view('Venda.create',compact('Venda','Cliente'));
    }
```

```
    public function edit($id){
```

```
        $Venda=Venda::findorFail($id);
```

```
        return view('Venda.edit',['Venda'=>$Venda]);
    }
```

```
    public function update(Request $request){
```

```
        Venda::find($request->id)->update($request->except('_token'));
    }
```

```
        return redirect('index/Venda')->with('msg', 'alteração realizado com sucesso');
```

```
}
```

```
    public function destroy($id)
```

```
{
```

```
        Venda::findorFail($id)->delete();
```

```
        return redirect('Venda.index')->with('msg', 'Venda');
```

```
}
```

```
    public function index(){
```

```
        $Venda = Venda::all();
```

```
return view('Venda.index',compact('Venda'));  
}
```

```
public function store(StoreVendaRequest $request){  
  
    $Venda=new Venda();  
    $Venda->datavenda=$request->datavenda;  
    $Venda->descontototal=$request->descontototal;  
  
    $Venda->descontoacerto=$request->descontoacerto;  
    $Venda->valortotal=$request->valortotal;  
    $Venda->cliente_id=$request->cliente_id;  
    $Venda->save();  
}  
}
```

-----Requets-----

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreCategoriaRequest 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',
    ];
    if (empty('nome.required')) {
        return redirect()->route('categoria.validacao');
    }
}

public function messages(){
    return [
        'nome.required' => 'Campo nome obrigatorio',
    ];
}

```



```
}
```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreClienteRequest extends FormRequest
```

```
{
```

```
    public function authorize()
```

```
    {
```

```
        return TRUE;
```

```
    }
```

```
    /**
```

```
     * Get the validation rules that apply to the request.
```

```
     *
```

```
     * @return array
```

```
     */
```

```
    public function rules()
```

```
    {
```

```
        return [
```

```
            //
```

```
            'nome'=>'required',
```

```
            'debito'=>'required|numeric',
```

```
            'cpf'=>'required|min:11',
```

```
            'rg'=>'required',
```

```
            'datanascimento'=>'required',
```

```

        'sexo'=>'required',
    ];
}

public function messages(){
    return [
        'nome.required' => 'Campo nome obrigatorio',
        'debito.required' => 'Campo debito obrigatorio',
        'cpf.required' => 'Campo cpf obrigatorio',
        'min.required' => 'Campo e obrigatorio ter 11 digito ',
        'rg.required' => 'Campo rg obrigatorio',
        'datanascimento.required' => 'Campo data de nascimento obrigatorio',
        'sexo.required'=>'Campo sexo e obrigatorio'
    ];
}
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreContatoClienteRequest extends FormRequest
```

```
{
```

```
/**
```

```
 * Determine if the user is authorized to make this request.
```

```

*
* @return bool
*/
public function authorize()
{
    return false;
}

/**
 * Get the validation rules that apply to the request.
 *
 * @return array
 */
public function rules()
{
    return [
        //
    ];
}
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreContatoFornecedorRequest extends FormRequest
```

```

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

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            //
        ];
    }
}

```

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

```

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

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            //
        ];
    }
}

```

<?php

namespace App\Http\Requests;

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreEnderecoClienteRequest extends FormRequest
```

```
{
```

```
    public function rules()
```

```
    {
```

```
        return [
```

```
            //
```

```
            'nome'=>'required',
```

```
            'debito'=>'required|numeric',
```

```
            'cpf'=>'required|min:11',
```

```
            'rg'=>'required',
```

```
            'datanascimento'=>'required',
```

```
            'sexo'=>'required',
```

```
        ];
```

```
    }
```

```
    public function messages(){
```

```
        return [
```

```
            'nome.required' => 'Campo nome obrigatorio',
```

```
            'debito.required' => 'Campo debito obrigatorio',
```

```
            'cpf.required' => 'Campo cpf obrigatorio',
```

```
            'min.required' => 'Campo e obrigatorio ter 11 digito ',
```

```
            'rg.required' => 'Campo rg obrigatorio',
```

```
            'datanascimento.required' => 'Campo data de nascimento obrigatorio',
```

```
            'sexo.required'=>'Campo sexo e obrigatorio'
```

```
        ];
```

```
    }
```

```
}
```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreEntradaRequest extends FormRequest
```

```
{
```

```
    public function authorize()
```

```
    {
```

```
        return TRUE;
```

```
    }
```

```
    /**
```

```
     * Get the validation rules that apply to the request.
```

```
     *
```

```
     * @return array
```

```
     */
```

```
    public function rules()
```

```
    {
```

```
        return [
```

```
            //
```

```
            'valortotaldanota'=>'required',
```

```
            'dataentrada'=>'required',
```

```
            'fornecedor_id'=>'required',
```

```
        ];
```

```
    }
```

```

public function messages(){
    return [
        'valortotaldanota.required' => 'Campo valor total da nota e obrigatorio',
        'dataentrada.required' => 'Campo data da entrada obrigatorio',
        'fornecedor_id.required' => 'Campo fornecedor obrigatorio',
    ];
}
}

```

<?php

```

namespace App\Http\Requests;

```

```

use Illuminate\Foundation\Http\FormRequest;

```

```

class StoreFormaPagamentoRequest extends FormRequest

```

```

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

    /**
     * Get the validation rules that apply to the request.

```



```

*
* @return array
*/
public function rules()
{
    return [
        //
    ];
}
}

```

```

<?php

```

```

namespace App\Http\Requests;

```

```

use Illuminate\Foundation\Http\FormRequest;

```

```

class StoreFornecedorRequest extends FormRequest
{
    public function authorize()
    {
        return TRUE;
    }
}

```

```

/**
 * Get the validation rules that apply to the request.
 *
 * @return array
 */

```

```
public function rules()
{
    return [
        //
        'razaosocial'=>'required',
        'cnpj'=>'required',

    ];
}
```

```
public function messages(){
    return [
        'razaosocial.required' => 'Campo razao social e obrigatorio',
        'cnpj.required' => 'Campo cnpj obrigatorio',

    ];
}
}
```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreItensEntradaRequest 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 [
        //
        'precompra'=>'required',
        'quantidade'=>'required',
        'unidade'=>'required',
        'ipi'=>'required',
        'frete'=>'required',
        'icms'=>'required',
        'entrada_id'=>'required',
        'produto_id'=>'required',

    ];
}

public function messages(){

```

```

return [
    'precompra.required'=>'Campo preco de compra e obrigatorio',
    'quantidade.required'=>'Campo quantidade obrigatorio',
    'unidade.required'=>'Campo unidade e obrigatorio',
    'ipi.required'=>'Campo ipi obrigatorio',
    'frete.required'=>'Campo frete e obrigatorio',
    'icms.required'=>'Campo icms obrigatorio',
    'entrada_id.required'=>'a entrada e obrigatorio',
    'produto_id.required'=>'Campo produto obrigatorio',
];
}
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreItensVendaRequest 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 [
        //
        'quantidade_itens'=>'required',
        'desconto'=>'required',
        'valor'=>'required',
        'venda_id'=>'required',
        'produto_id'=>'required',

    ];
}

public function messages(){
    return [
        'quantidade_itens.required'=>'Campo quantidade de itens e obrigatorio',
        'desconto.required'=>'Campo desconto obrigatorio',
        'valor.required'=>'Campo valor e obrigatorio',
        'venda_id.required'=>'Campo venda obrigatorio',
        'produto_id.required'=>'Campo produto e obrigatorio',
    ];
}
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreMarcaRequest 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',
```

```

    ];
}

public function messages(){
    return [
        'nome.required'=>'Campo nome da marca e obrigatorio',
    ];
}
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```

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

    /**

```

```

    * Get the validation rules that apply to the request.
    *
    * @return array
    */
    public function rules()
    {
        return [
            //
        ];
    }
}

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

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

```



```

    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            //
        ];
    }
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StorePrateleiraRequest extends FormRequest
```

```

{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
}

```

```

public function authorize()
{
    return false;
}

/**
 * Get the validation rules that apply to the request.
 *
 * @return array
 */
public function rules()
{
    return [
        //
    ];
}
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreProdutoRequest 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',
        'icms'=>'required',
        'ipi'=>'required',
        'frete'=>'required',
        'valornafabrica'=>'required',
        'valordecompra'=>'required',
        'lucro'=>'required',
        'valorvenda'=>'required',
        'desconto'=>'required',
        'quantidade'=>'required',
        'datavencimento'=>'required',
        'marca_id'=>'required',
        'categoria_id'=>'required',
    ];
}

```

```

];
}

public function messages(){
    return [
        'nome.required' => 'Campo nome e obrigatorio',
        'icms.required' => 'Campo icms obrigatorio',
        'ipi.required' => 'Campo ipi e obrigatorio',
        'frete.required' => 'Campo frete obrigatorio',
        'valornafabrica.required' => 'Campo valor na fabrica e obrigatorio',
        'valordecompra.required' => 'Campo valor de compra e obrigatorio',
        'lucro.required' => 'Campo lucro obrigatorio',
        'valorvenda.required' => 'Campo valor de venda e obrigatorio',
        'desconto.required' => 'Campo desconto obrigatorio',
        'quantidade.required' => 'Campo quantidade e obrigatorio',
        'datavencimento.required' => 'Campo datavencimento e obrigatorio',
        'marca_id.required' => 'Campo marca obrigatorio',
        'categoria_id.required' => 'Campo categoria e obrigatorio',
    ];
}
}

```

```
<?php
```

```
namespace App\Http\Requests;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class StoreVendaRequest 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 [  
            //  
            'descontototal'=>'required',  
            'descontoacerto'=>'required',  
            'data_venda'=>'required',  
            'cliente_id'=>'required',  
        ];  
        return redirect()->route('venda.index');  
    }  
}
```

```
public function messages(){  
    return [  
        
```

```

        'descontototal.required' => 'Campo desconto total obrigatorio',
        'descontoacerto.required' => 'Campo desconto acerto obrigatorio',
        'data_venda.required' => 'Campo data venda obrigatorio',
        'cliente_id.required' => 'Campo nome do cliente obrigatorio',
    ];
}
}

```

-----comando-----

```

roberiofagundes@roberiofagundes-Lenovo-IdeaPad-S145-
15IGM:~/Documentos/projetos/perfume-shop$ su

```

Senha:

```

root@roberiofagundes-Lenovo-IdeaPad-S145-
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan
make:request StoreCategoriaRequest

```

Request created successfully.

```

root@roberiofagundes-Lenovo-IdeaPad-S145-
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan
make:request StoreClienteRequest

```

Request created successfully.

```

root@roberiofagundes-Lenovo-IdeaPad-S145-
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan
make:request StoreContatoClienteRequest

```

Request created successfully.

```

root@roberiofagundes-Lenovo-IdeaPad-S145-
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan
make:request StoreContatoFornecedorRequest

```

Request created successfully.

```

root@roberiofagundes-Lenovo-IdeaPad-S145-
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan
make:request StoreDevolucaoRequest

```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreEnderecoClienteRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreEntradaRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreFormaPagamentoRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreFornecedorRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreItensEntradaRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreItensVendaRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreMarcaRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StorePagamentoRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreParcelaRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StorePrateleiraRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreProdutoRequest
```

Request created successfully.

```
root@roberiofagundes-Lenovo-IdeaPad-S145-  
15IGM:/home/roberiofagundes/Documentos/projetos/perfume-shop# php artisan  
make:request StoreVendaRequest
```

Request created successfully.

```
-----middleware-----
```

-

```
<?php
```

```
namespace App\Http\Middleware;
```

```
use Illuminate\Auth\Middleware\Authenticate as Middleware;
```

```
class Authenticate extends Middleware
```

```
{
```

```
/**
```

```
 * Get the path the user should be redirected to when they are not authenticated.
```

```
 *
```

```
 * @param \Illuminate\Http\Request $request
```

```
 * @return string|null
```

```
 */
```

```
protected function redirectTo($request)
```

```
{
```

```
    if (!$request->expectsJson()) {
```

```
        return route('login');
```



```
    }  
  }  
}
```

```
<?php
```

```
namespace App\Http\Middleware;
```

```
use Closure;
```

```
use Illuminate\Http\Request;
```

```
class IsAdmin
```

```
{  
    /**  
     * Handle an incoming request.  
     *  
     * @param  \Illuminate\Http\Request $request  
     * @param  \Closure $next  
     * @return mixed  
     */  
    public function handle(Request $request, Closure $next)  
    {  
        if(auth()->user()->is_admin == 1){  
  
            return $next($request);  
        }  
    }  
}
```

-----model-----

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Categoria extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table="categorias";
```

```
    protected $guarded = [];
```

```
    public function produto(){
```

```
        return $this->benlogsTo(Produto::class, 'produto_id','id');
```

```
    }
```

```
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```

class Cliente extends Model
{
    use HasFactory;
    protected $table="clientes";
    protected $guarded = [];

    public function endereco(){
        return $this->hasOne(Cliente::class, 'cliente_id','id');
    }

    public function venda(){
        return $this->belongsTo(Venda::class, 'cliente_id','id');
    }
}

```

```
<?php
```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class ContatoCliente extends Model
{
    use HasFactory;

    protected $table="contato_clientes";

```

```
protected $guarded = [];
```

```
public function cliente(){  
    return $this->hasOne(Venda::class, 'cliente_id','id');  
}  
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class ContatoFornecedor extends Model
```

```
{  
    use HasFactory;  
    protected $table="contato_fornecedores";  
    protected $guarded = [];  
  
    public function fornecedor(){  
        return $this->belongsTo(fornecedor::class, 'fornecedor_id','id');  
    }  
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Devolucao extends Model
{
    use HasFactory;
    protected $table="devolucao";
    protected $guarded = [];

    public function produto(){
        return $this->belongsTo(Produto::class, 'produto_id','id');
    }
}
```

<?php

```
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class EnderecoCliente extends Model
{
    use HasFactory;
    protected $table="endereco_cliente";
    protected $guarded = [];

    public function cliente(){
```

```
        return $this->hasOne(Cliente::class, 'cliente_id','id');
    }
}
```

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Entrada extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table="entradas";
```

```
    protected $guarded = [];
```

```
    public function fornecedor(){
```

```
        return $this->hasOne(fornecedor::class, 'fornecedor_id','id');
```

```
    }
```

```
}
```

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class FormaPagamento extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table="forma_pagamentos";
```

```
    protected $guarded = [];
```

```
    public function venda(){
```

```
        return $this->hasOne(Venda::class, 'venda_id','id');
```

```
    }
```

```
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Fornecedor extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table="fornecedores";
```

```
protected $guarded = [];
```

```
public function entrada(){  
    return $this->belongsTo(Entrada::class, 'entrada_id', 'id');  
}  
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class ItensEntrada extends Model
```

```
{  
    use HasFactory;  
    protected $table="item_entradas";  
    protected $guarded = [];
```

```
    public function entrada(){  
        return $this->hasOne(Entrada::class, 'entrada_id', 'id');  
    }
```

```
    public function produto(){  
        return $this->belongsTo(Produto::class, 'produto_id', 'id');  
    }  
}
```



```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class ItensVendas extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table="item_vendas";
```

```
    protected $guarded = [];
```

```
    public function venda(){
```

```
        return $this->hasOne(Venda::class, 'venda_id','id');
```

```
    }
```

```
    public function produto(){
```

```
        return $this->hasOne(Produto::class, 'produto_id','id');
```

```
    }
```

```
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Marca extends Model
{
    use HasFactory;
    protected $table="marcas";
    protected $guarded = [];

    public function produto(){
        return $this->belongsTo(Produto::class, 'produto_id','id');
    }
}
```

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class Pagamento extends Model
{
    use HasFactory;
    protected $table="pagamentos";
    protected $guarded = [];

    public function parcela(){
        return $this->belongsTo(Parcela::class, 'parcela_id','id');
    }
}
```

```
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Parcelas extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table="parcelas";
```

```
    protected $guarded = [];
```

```
    public function pagamento(){
```

```
        return $this->hasOne(pagamento::class, 'pagamento_id','id');
```

```
    }
```

```
}
```

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
class Prateleira extends Model
{
    use HasFactory;
    protected $table="prateleiras";
    protected $guarded = [];

    public function produto(){
        return $this->belongsTo(Produto::class, 'produto_id','id');
    }
}
```

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class Produto extends Model
{
    use HasFactory;
    protected $table="produtos";
    protected $guarded = [];

    public function marca(){
        return $this->belongsTo(Marca::class, 'marca_id','id');
    }
}
```

```

    public function categoria(){
        return $this->belongsTo(Categoria::class, 'categoria_id','id');
    }

    public function prateleira(){
        return $this->belongsTo(Prateleira::class, 'prateleira_id','id');
    }
}

```

<?php

```

namespace App\Models;

use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var string[]
     */
    protected $fillable = [

```

```

        'name',
        'email',
        'password',
        'is_admin'
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}

```

<?php

namespace App\Models;

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class Venda extends Model
{
    use HasFactory;
    protected $table="vendas";
    protected $guarded = [];

    public function cliente(){
        return $this->hasOne(Cliente::class, 'cliente_id','id');
    }
}
```

-----views-----

```
@extends('layouts.main')
@extends('layouts.app')
```

```
@section('titulo','Perfume-Shop')
@section('conteudo')
<div class="col">
    @if($errors->any())
        <div class="alert alert-danger">

            @foreach($errors->all() as $error)

                <label>{{$error}}</label><br>
```

@endforeach

</div>

@endif

</div>

<div class="col">

<form action="{{route('categoria.store')}}" method="post">

@csrf

<div id="row">

<h4>cadastro de categoria</h4>

<label for="nome" class="label">nome</label>

<input type="text" class="form-control" name="nome" value="{{ old('nome')
}}">

<input type="submit" class="btn btn-primary" value="cadastrar">

</div>

</div>


```
        </label>
    </form>
@endsection('conteudo')

@section('footer')

@endsection('footer')

@extends('layouts.main')

@section('titulo','Perfume-Shop')

@section('conteudo')

<body>
<div class="row g-3">
    <div class="col">
        <form action="{{route('categoria.update', ['id' => $categoria->id])}}"
method="post">
<h4>Edição de categoria</h4><br>
```

```
@csrf
```

```
@method('PUT')
```

```
<label for="nome">nome da categoria</label>
```

```
{{-- Form Section 1 --}}
```

```
<input type="text" name="nome" class="form-control" value="{{ $categoria-  
>nome }}"><br>
```

```
<input type="submit" class="btn btn-primary" value="editar">
```

```
</form>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
@endsection('conteudo')
```

```
@section('footer')
```

```
@endsection('footer')
```

```
@extends('layouts.main')
```

```
@extends('layouts.app-index')
```

```
@section('conteudo')
```

```
<table class="table table-sm">
```

```
    @section('titulo','Categoria')
```

```
    @foreach ($Categoria as $categoria)
```

```
        <thead>
```

```
            <tr>
```

```
                <th>nome</th>
```

```
                <th></th>
```

```
                <th></th>
```

```
                <th></th>
```

```
            </tr>
```

```
        </thead>
```

```
        <td>{{$categoria->nome}}</td>
```

```

        <td>
            <form action="{{route('categoria.delete', ['id' => $categoria->id])}}"
method="post">
                @csrf
                @method('DELETE')
                <input type="submit" class="btn btn-primary" value="deletar">
            </form>
        </td>
        <td>
            <form action="{{route('categoria.create')}}" method="post">
                @csrf
                <input type="submit" class="btn btn-primary" name="formulario"
value="novo">
            </form>
        </td>
        <td>
            <form action="{{route('categoria.edit', ['id' => $categoria->id])}}"
method="post">
                @csrf
                <input type="submit" class="btn btn-primary" name="formulario"
value="alterar">
            </form>
        </td>

    @endforeach

</table>

```

```
@endsection('conteudo')
```

```
@section('footer')
```

```
@endsection('footer')
```

```
@extends('layouts.main')
```

```
@section('conteudo')
```

```
<form action="{{route('cliente.store')}}" method="post">
```

```
<div class="mb-3">
```

```
    @if($errors->any())
```

```
        <div class="alert alert-danger">
```

```
            @foreach($errors->all() as $error)
```

```
                <label>{{$error}}</label><br>
```

```
            @endforeach
```

```
        @endif
```

```

<h4 align="center">Cadastro de cliente</h4><br>
@csrf
<label for="nome">nome</label>
<input type="text" class="form-control" name="nome" id="nome">
<label for="debito">debito</label>
<input type="text" class="form-control" name="debito" id="debito">
<label for="cpf">cpf</label>
<input type="text" class="form-control" name="cpf" id="cpf">
<label for="rg">rg</label>
<input type="text" class="form-control" name="rg" id="rg">
<label for="sexo">sexo</label>
<input type="text" class="form-control" name="sexo" id="sexo">
<label for="datanascimento">datanascimento</label>
<input type="date" class="form-control" name="datanascimento"
id="datanascimento">
<label for="debito">debito</label>
<input type="text" class="form-control" name="debito"
id="debito"><br><br>
<input type="submit" class="btn btn-primary" value="cadastrar">
</div>

</div>

</form>

```

```
@endsection('conteudo')
```

```
@section('footer')
```

```
@endsection('footer')
```

```
@extends('layouts.main')
```

```
@section('titulo','Perfume-Shop')
```

```
@section('conteudo')
```

```
<div class="row g-3">
```

```
<h4>edição de cliente</h4>
```

```
<div class="col">
```

```
<form action="{{route('cliente.update', ['id' => $cliente->id])}}" method="post">
```

```
@csrf
```

```
@method('PUT')
```

```
<label for="nome">nome do cliente</label>
```

```
<input type="text" name="nome" class="form-control" value="{{ $cliente->nome }}">
```

```
<label for="nome">debito</label>
```

```
<input type="text" name="debito" class="form-control" value="{{ $cliente->debito }}">
```

```
<label for="cpf">cpf</label>
```

```
<input type="text" name="cpf" class="form-control" value="{{ $cliente->cpf }}"><br>
```

```
<input type="submit" class="btn btn-primary" value="editar">
```

```
</div><br>
```

```
<div class="col">
```

```
  <label for="rg">rg</label>
```

```
  <input type="text" name="rg" class="form-control" value="{{ $cliente->rg }}">
```

```
  <label for="datanascimento">datanascimento</label>
```

```
  <input type="date" name="datanascimento" class="form-control"
value="{{ $cliente->datanascimento }}">
```

```
  <label for="sexo">sexo</label>
```

```
  <input type="text" name="sexo" class="form-control" value="{{ $cliente-
>sexo }}"><br><br>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
@endsection('conteudo')
```

```
@section('footer')
```

```
@endsection('footer')
```



```
@extends('layouts.main')
```

```
@extends('layouts.app-index')
```

```
@section('conteudo')
```

```
<table class="table table-sm">
```

```
  @section('titulo','Clientes')
```

```
  @foreach ($Cliente as $cliente)
```

```
    <thead>
```

```
      <tr>
```

```
        <th scope="col">nome do cliente </th>
```

```
        <th>debito</th>
```

```
        <th>cpf</th>
```

```
        <th>sexo</th>
```

```
        <th>datanascimento</th>
```

```

<th></th>
<th></th>
<th></th>
<th></th>

</thead>
<thead>
  <td scope="row">{{ $cliente->nome }}

  <td>{{ $cliente->debito }}

  <td>{{ $cliente->cpf }}</td>
  <td>{{ $cliente->sexo }}</td>
  <td>{{ $cliente->datanascimento }}</td>
  <td class="tabela">
    <td>
      <form action="{{ route('cliente.create') }}" method="post">
        @csrf
        <input type="submit" class="btn btn-primary" name="formulario"
value="novo">
      </form>
    </td>

    <td>
      <form action="{{ route('cliente.edit', ['id' => $cliente->id]) }}" method="post">
        @csrf
        <input type="submit" class="btn btn-primary" name="formulario"
value="alterar">
      </form>
    </td>
  <td>

```

```
<form action="..../cliente/{{ $cliente->id }}" method="post">
    @csrf
    @method('DELETE')
    <div id="button">
        <input type="submit" class="btn btn-primary" name="formulario"
value="deletar">
    </div>

</form>
</td>
</thead>
@endforeach

</table>
```

```
@endsection('conteudo')
```

```
@extends('layouts.main')
```

```
@section('titulo','Perfume-Shop')
```

```
@section('conteudo')
```

```

<form action="{{route('vendas.store')}}" method="post">
  <div class="row g-3">

    <h4>cadastro de venda</h4><br>

    @if($errors->any())
      <div class="alert alert-danger">

        @foreach($errors->all() as $error)

          <label>{{$error}}</label><br>
        @endforeach

      </div>

    @endif

    <div class="col">
      @csrf

      <label for="nome">descontototal</label>

      <input class="form-control" type="text" name="descontototal"
id="descontototal" style="width:150px">

      <label for="descontoacerto">descontoacerto</label>

      <input class="form-control" type="text" name="descontoacerto"
id="descontoacerto" style="width:150px">

      <label for=" data_venda"> data_venda</label>

      <input class="form-control" type="date" name=" data_venda" id=" data_venda"
style="width:150px">

```

```
<label for="clientes">cliente</label>
```

```
{{-- pega a variavel passada pelo metodo create do controller --}}
```

```
<select class="form-control" name="cliente_id" style="width:150px">
```

```
  @foreach($Clientes as $cliente)
```

```
    <option value="{{ $cliente->cliente_id }}">
```

```
      {{ $cliente->nome }}
```

```
    </option>
```

```
  @endforeach
```

```
</select>
```

```
<div class="col">
```

```
  @if($errors->any())
```

```
    <div class="alert alert-danger">
```

```
      @foreach($errors->all() as $error)
```

```
        <label>{{$error}}</label><br>
```

```
      @endforeach
```

```
</div>
```

```
@endif
```

```
</div><br>
```

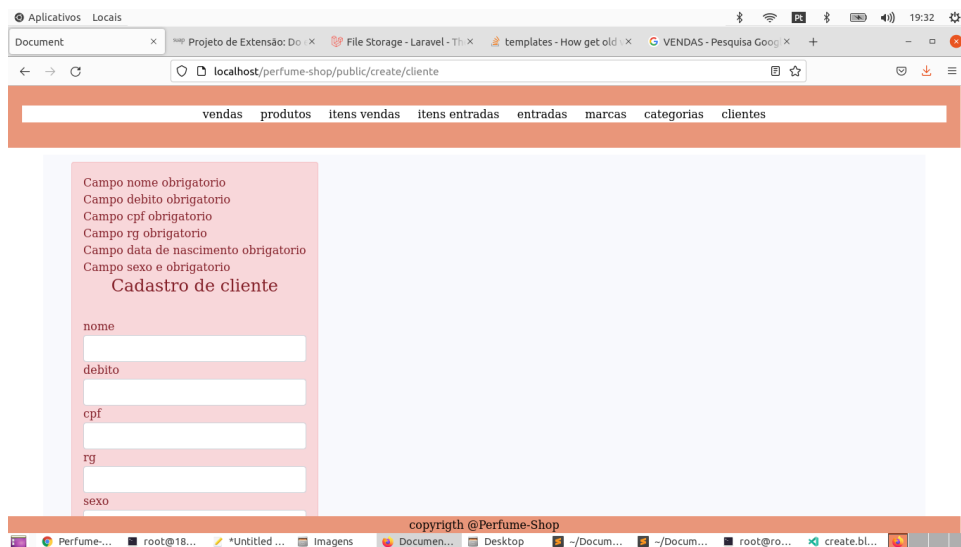
```
<input type="submit" class="btn btn-primary" value="cadastrar">
```

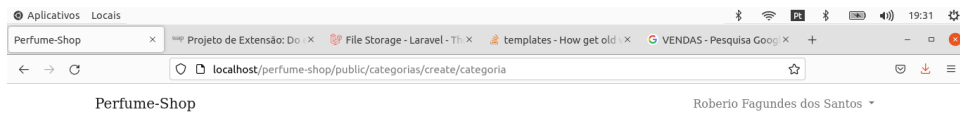
```
</form>
```

@endsection('conteudo')

@section('footer')

@endsection('footer')





Perfume-Shop

vendas produtos itens vendas itens entradas entradas marcas categorias clientes

cadastro de categoria

Campo nome obrigatorio

nome

[cadastrar](#)

pagina de cadastro de marca

nome

Campo nome da marca e obrigatorio

[cadastrar](#)

pagina de cadastro de item de entrada

Campo preco de compra e obrigatorio
Campo quantidade obrigatorio
Campo unidade e obrigatorio
Campo ipi obrigatorio
Campo frete e obrigatorio
Campo icms obrigatorio
a entrada e obrigatorio
Campo produto obrigatorio

| | | |
|----------------------|----------------------|----------------------|
| produto | quantidade | icms |
| kalak | <input type="text"/> | <input type="text"/> |
| preço de compra | unidade | entrada |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| | ipi | frete |
| | <input type="text"/> | <input type="text"/> |

[cadastrar](#)



AplicativosLocais

Perfume-ShopProjeto de Extensao: DoFile Storage - Laravel - Thtemplates - How get oldVENDAS - Pesquisa Google

localhost/perfume-shop/public/create/produto

Cadastro de produto

vendasprodutositens vendasitens entradasentradasmarcascategoriasclientes

Campo nome e obrigatorio
Campo icms obrigatorio
Campo ipi e obrigatorio
Campo frete obrigatorio
Campo valor na fabrica e obrigatorio
Campo valor de compra e obrigatorio
Campo lucro obrigatorio
Campo valor de venda e obrigatorio
Campo desconto obrigatorio
Campo quantidade e obrigatorio
Campo data vencimento e obrigatorio

nome do produto

nome do produto

icms

icms

lucro

lucro

cadastrar

ipi

ipi

frete

frete

valor de venda

valor de venda

valor na fabrica

valor na fabrica

valor de compra

valor de compra

desconto

desconto

quantidade

quantidade

data de vencimento

dd/mm/aaaa

marcas

seleccione

categoria

seleccione

prateleira

seleccione

copyright @Perfume-Shop

Perfume...root@18...*Untitled...ImagensPerfume...Desktop~/Docum...~/Docum...root@ro...home.bla...