**Requisitos**

* PHP 8.1+ (ou XAMPP com PHP 8+)
* Composer
* MySQL (pode ser o do XAMPP)
* Node.js (para assets opcionais)

**Passo 1 – Criar Banco de dados**

Crie o Banco de dados

CREATE DATABASE pizzaria\_db;

USE pizzaria\_db;

CREATE TABLE usuarios (

  id *INT* AUTO\_INCREMENT PRIMARY KEY,

  nome *VARCHAR*(100) NOT NULL,

  senha *VARCHAR*(255) NOT NULL

);

CREATE TABLE pizzas (

  id *INT* AUTO\_INCREMENT PRIMARY KEY,

  nome *VARCHAR*(100) NOT NULL,

  ingredientes *TEXT* NOT NULL,

  preco *DECIMAL*(8,2) NOT NULL,

  tamanho ENUM('Pequena', 'Media', 'Grande') NOT NULL,

  categoria *VARCHAR*(50) NOT NULL,

  estoque\_minimo *INT* DEFAULT 5,

  ativo *BOOLEAN* DEFAULT TRUE

);

CREATE TABLE movimentacoes (

  id *INT* AUTO\_INCREMENT PRIMARY KEY,

  pizza\_id *INT* NOT NULL,

  usuario\_id *INT* NOT NULL,

  data\_hora DATETIME NOT NULL,

  tipo ENUM('entrada','saida') NOT NULL,

  quantidade *INT* NOT NULL,

  observacoes *TEXT*,

  FOREIGN KEY (pizza\_id) REFERENCES pizzas(id),

  FOREIGN KEY (usuario\_id) REFERENCES usuarios(id)

);

INSERT INTO usuarios (nome, senha) VALUES

('Admin', 'admin123'),

('Gerente', 'gerente123'),

('Atendente', 'atendente123');

INSERT INTO pizzas (nome, ingredientes, preco, tamanho, categoria) VALUES

('Margherita', 'Molho de tomate, mussarela, manjericão', 25.90, 'Media', 'Tradicional'),

('Calabresa', 'Molho de tomate, mussarela, calabresa, cebola', 28.90, 'Media', 'Tradicional'),

('Portuguesa', 'Molho de tomate, mussarela, presunto, ovos, ervilha, cebola', 32.90, 'Media', 'Especial'),

('Frango com Catupiry', 'Molho de tomate, mussarela, frango desfiado, catupiry', 35.90, 'Media', 'Especial'),

('Quatro Queijos', 'Molho de tomate, mussarela, gorgonzola, parmesão, provolone', 38.90, 'Media', 'Especial');

INSERT INTO movimentacoes (pizza\_id, usuario\_id, data\_hora, tipo, quantidade, observacoes) VALUES

(1, 1, '2025-01-15 10:30:00', 'entrada', 5, 'Estoque inicial'),

(2, 1, '2025-01-15 10:35:00', 'entrada', 3, 'Estoque inicial'),

(3, 2, '2025-01-15 11:00:00', 'saida', 2, 'Venda para cliente'),

(4, 2, '2025-01-15 11:15:00', 'entrada', 4, 'Reposição de estoque');

**Passo 2**

Crie uma pasta dentro do htdocs chamada pizzaria (para ser a pasta raiz do projeto)

Criar projeto laravel na raiz

composer create-project laravel/laravel .

**verificar se esta no caminho**

C:\xamppp\htdocs\pizzaria>

**arrumar arquivo .env**

APP\_NAME=Laravel

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=pizzaria\_db

DB\_USERNAME=root

DB\_PASSWORD=

### Instalar Breeze para autenticação

***Rodar um comando de cada vez***

composer require laravel/breeze --dev

php artisan breeze:install blade

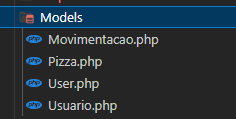
npm install

npm run build

**Passo 3 - Criar Models para suas tabelas existentes**

Dentro da pasta

**app>Models**

****

Crie os arquivos:

**Usuario.php**

<?php

namespace App\Models;

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

use Illuminate\Database\Eloquent\*Model*;

use Illuminate\Database\Eloquent\Relations\*HasMany*;

use Illuminate\Foundation\Auth\*User* as Authenticatable;

use Illuminate\Notifications\*Notifiable*;

*class* Usuario extends *Authenticatable*

{

    public $timestamps = false;

    use *HasFactory*, *Notifiable*;

    protected $table = 'usuarios';

    protected $fillable = [

        'nome',

        'senha',

    ];

    protected $hidden = [

        'senha',

    ];

    protected $casts = [

        'senha' => 'hashed',

    ];

    public *function* movimentacoes(): *HasMany*

    {

        return $this->hasMany(*Movimentacao*::class, 'usuario\_id');

    }

    // Método para autenticação personalizada

    public *function* getAuthPassword()

    {

        return $this->senha;

    }

}

**Pizza.php**

<?php

namespace App\Models;

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

use Illuminate\Database\Eloquent\*Model*;

use Illuminate\Database\Eloquent\Relations\*HasMany*;

*class* Pizza extends *Model*

{

    public $timestamps = false;

    use *HasFactory*;

    protected $fillable = [

        'nome',

        'ingredientes',

        'preco',

        'tamanho',

        'categoria',

        'estoque\_minimo',

        'ativo',

    ];

    protected $casts = [

        'preco' => 'decimal:2',

        'ativo' => 'boolean',

        'estoque\_minimo' => 'integer',

    ];

    public *function* movimentacoes(): *HasMany*

    {

        return $this->hasMany(*Movimentacao*::class);

    }

    // Calcula estoque atual (entradas - saídas)

    public *function* getEstoqueAtualAttribute(): int

    {

        $entradas = $this->movimentacoes()->where('tipo', 'entrada')->sum('quantidade');

        $saidas = $this->movimentacoes()->where('tipo', 'saida')->sum('quantidade');

        return (*int*) $entradas - (*int*) $saidas;

    }

    // Verifica se estoque está baixo

    public *function* getEstoqueBaixoAttribute(): bool

    {

        return $this->estoque\_atual <= $this->estoque\_minimo;

    }

    // Scope para pizzas ativas

    public *function* scopeAtivas($query)

    {

        return $query->where('ativo', true);

    }

}

**Movimentacao.php**

<?php

namespace App\Models;

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

use Illuminate\Database\Eloquent\*Model*;

use Illuminate\Database\Eloquent\Relations\*BelongsTo*;

*class* Movimentacao extends *Model*

{

    use *HasFactory*;

    public $timestamps = false;

    protected $table = 'movimentacoes';  // Corrigindo o nome da tabela

    protected $fillable = [

        'pizza\_id',

        'usuario\_id',

        'data\_hora',

        'tipo',

        'quantidade',

        'observacoes',

    ];

    protected $casts = [

        'data\_hora' => 'datetime',

        'quantidade' => 'integer',

    ];

    public *function* pizza(): *BelongsTo*

    {

        return $this->belongsTo(*Pizza*::class);

    }

    public *function* usuario(): *BelongsTo*

    {

        return $this->belongsTo(*Usuario*::class, 'usuario\_id');

    }

    // Scope para entradas

    public *function* scopeEntradas($query)

    {

        return $query->where('tipo', 'entrada');

    }

    // Scope para saídas

    public *function* scopeSaidas($query)

    {

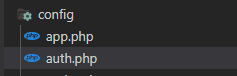
        return $query->where('tipo', 'saida');

    }

}

**Passo 4 - Configurar autenticação personalizada**

***Dentro de config>auth.php***

****

**Crie o auth.php**

<?php

return [

    'defaults' => [

        'guard' => 'usuarios',

        'passwords' => 'usuarios',

    ],

    'guards' => [

        'web' => [

            'driver' => 'session',

            'provider' => 'users',

        ],

        'usuarios' => [

            'driver' => 'session',

            'provider' => 'usuarios',

        ],

    ],

    'providers' => [

        'users' => [

            'driver' => 'eloquent',

            'model' => App\Models\*User*::class,

        ],

        'usuarios' => [

            'driver' => 'eloquent',

            'model' => App\Models\*Usuario*::class,

        ],

        // 'users' => [

        //     'driver' => 'database',

        //     'table' => 'users',

        // ],

    ],

    'passwords' => [

        'users' => [

            'provider' => 'users',

            'table' => 'password\_reset\_tokens',

            'expire' => 60,

            'throttle' => 60,

        ],

        'usuarios' => [

            'provider' => 'usuarios',

            'table' => 'password\_reset\_tokens',

            'expire' => 60,

            'throttle' => 60,

        ],

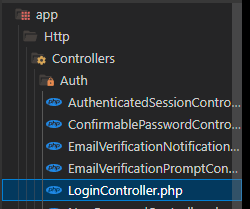
    ],

    'password\_timeout' => 10800,

];

**Passo 5 - Criar Controllers**

***Dentro de app>Http>Controllers>Auth***

****

**Crie o arquivo LoginController.php**

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\*Controller*;

use App\Models\*Usuario*;

use Illuminate\Http\*Request*;

use Illuminate\Support\Facades\*Auth*;

use Illuminate\Support\Facades\*Hash*;

use Illuminate\Validation\*ValidationException*;

*class* LoginController extends *Controller*

{

    public *function* showLoginForm()

    {

        return view('auth.login');

    }

public *function* login(*Request* $request)

{

    $request->validate([

        'nome' => 'required|string',

        'senha' => 'required|string',

    ]);

    $usuario = *Usuario*::where('nome', $request->nome)->first();

    if (!$usuario || $request->senha !== $usuario->senha) {

        throw *ValidationException*::withMessages([

            'nome' => ['As credenciais fornecidas estão incorretas.'],

        ]);

    }

*Auth*::guard('usuarios')->login($usuario);

    $request->session()->regenerate();

    return redirect()->intended('/');

}

public *function* logout(*Request* $request)

{

*Auth*::guard('usuarios')->logout(); // ou Auth::logout() dependendo do guard usado

    $request->session()->invalidate();

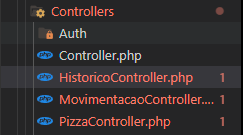
    $request->session()->regenerateToken();

    return redirect('/login');

}

}

**Dentro de app>Http>Controllers**

****

**Em Controller.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\*AuthorizesRequests*;

use Illuminate\Foundation\Bus\*DispatchesJobs*;

use Illuminate\Foundation\Validation\*ValidatesRequests*;

use Illuminate\Routing\*Controller* as BaseController;

*class* Controller extends *BaseController*

{

    use *AuthorizesRequests*, *DispatchesJobs*, *ValidatesRequests*;

}

**Crie o HistoricoController.php**

<?php

namespace App\Http\Controllers;

use App\Http\Controllers\*Controller*;

use App\Models\*Movimentacao*;

use App\Models\*Pizza*;

use Illuminate\Http\*Request*;

*class* HistoricoController extends *Controller*

{

    public *function* \_\_construct()

    {

        $this->middleware('auth:usuarios');

    }

    public *function* index(*Request* $request)

    {

        $query = *Movimentacao*::with(['pizza', 'usuario'])->orderByDesc('data\_hora');

        // Filtros

        if ($request->filled('pizza\_id')) {

            $query->where('pizza\_id', $request->pizza\_id);

        }

        if ($request->filled('tipo')) {

            $query->where('tipo', $request->tipo);

        }

        if ($request->filled('de')) {

            $query->where('data\_hora', '>=', $request->de . ' 00:00:00');

        }

        if ($request->filled('ate')) {

            $query->where('data\_hora', '<=', $request->ate . ' 23:59:59');

        }

        $movimentacoes = $query->paginate(50);

        $pizzas = *Pizza*::orderBy('nome')->get();

        return view('historico.index', compact('movimentacoes', 'pizzas'));

    }

}

**Crie o MovimentacaoController.php**

<?php

namespace App\Http\Controllers;

use App\Http\Controllers\*Controller*;

use App\Models\*Movimentacao*;

use App\Models\*Pizza*;

use Illuminate\Http\*Request*;

use Illuminate\Support\Facades\*Log*;

*class* MovimentacaoController extends *Controller*

{

    public *function* \_\_construct()

    {

        $this->middleware('auth:usuarios');

    }

    public *function* index(*Request* $request)

    {

        $pizzas = *Pizza*::orderBy('nome')->get();

        $movimentacoes = *Movimentacao*::with(['pizza', 'usuario'])

            ->latest('data\_hora')

            ->paginate(20);

        $debug = (*bool*) $request->query('debug');

        // Log para debug

        if ($debug) {

*Log*::info('Debug Movimentações', [

                'post' => $request->all(),

                'session' => session()->all()

            ]);

        }

        return view('movimentacoes.index', compact('pizzas', 'movimentacoes', 'debug'));

    }

    public *function* store(*Request* $request)

    {

        $data = $request->validate([

            'pizza\_id' => 'required|exists:pizzas,id',

            'tipo' => 'required|in:entrada,saida',

            'quantidade' => 'required|integer|min:1',

            'observacoes' => 'nullable|string',

        ]);

        $pizza = *Pizza*::findOrFail($data['pizza\_id']);

        // Validar se há estoque suficiente para saída

        if ($data['tipo'] === 'saida') {

            $estoqueAtual = $pizza->estoque\_atual;

            if ($data['quantidade'] > $estoqueAtual) {

                return back()

                    ->withErrors(['quantidade' => "Quantidade maior que o estoque atual ({$estoqueAtual})."])

                    ->withInput();

            }

        }

        try {

*Movimentacao*::create([

                'pizza\_id' => $pizza->id,

                'usuario\_id' => auth('usuarios')->id(),

                'data\_hora' => now(),

                'tipo' => $data['tipo'],

                'quantidade' => $data['quantidade'],

                'observacoes' => $data['observacoes'] ?? null,

            ]);

*Log*::info('Movimentação registrada', [

                'pizza' => $pizza->nome,

                'tipo' => $data['tipo'],

                'quantidade' => $data['quantidade']

            ]);

            return redirect()->route('movimentacoes.index')

                ->with('success', 'Movimentação registrada com sucesso!');

        } catch (\*Exception* $e) {

*Log*::error('Erro ao registrar movimentação', ['erro' => $e->getMessage()]);

            return redirect()->route('movimentacoes.index')

                ->with('error', 'Erro ao registrar movimentação: ' . $e->getMessage());

        }

    }

}

**Crie o PizzaController.php**

<?php

namespace App\Http\Controllers;

use App\Http\Controllers\*Controller*;

use App\Models\*Pizza*;

use Illuminate\Http\*Request*;

use Illuminate\Support\Facades\*Log*;

*class* PizzaController extends *Controller*

{

    public *function* \_\_construct()

    {

        $this->middleware('auth:usuarios');

    }

    public *function* index(*Request* $request)

    {

        $pizzas = *Pizza*::orderBy('nome')->paginate(15);

        $debug = (*bool*) $request->query('debug');

        // Log para debug

        if ($debug) {

*Log*::info('Debug Pizzas', [

                'post' => $request->all(),

                'session' => session()->all()

            ]);

        }

        return view('pizzas.index', compact('pizzas', 'debug'));

    }

    public *function* store(*Request* $request)

    {

        $data = $request->validate([

            'nome' => 'required|string|max:100',

            'ingredientes' => 'required|string',

            'preco' => 'required|numeric|min:0',

            'tamanho' => 'required|in:Pequena,Media,Grande',

            'categoria' => 'required|string|max:50',

            'estoque\_minimo' => 'required|integer|min:0',

            'ativo' => 'nullable|boolean',

        ]);

        // Normalizar preço (vírgula para ponto)

        $data['preco'] = str\_replace(',', '.', $data['preco']);

        $data['ativo'] = (*bool*)($data['ativo'] ?? true);

        try {

*Pizza*::create($data);

*Log*::info('Pizza cadastrada', ['pizza' => $data['nome']]);

            return redirect()->route('pizzas.index')

                ->with('success', 'Pizza cadastrada com sucesso!');

        } catch (\*Exception* $e) {

*Log*::error('Erro ao cadastrar pizza', ['erro' => $e->getMessage()]);

            return redirect()->route('pizzas.index')

                ->with('error', 'Erro ao cadastrar pizza: ' . $e->getMessage());

        }

    }

    public *function* edit(*Pizza* $pizza)

    {

        return view('pizzas.edit', compact('pizza'));

    }

    public *function* update(*Request* $request, *Pizza* $pizza)

    {

        $data = $request->validate([

            'nome' => 'required|string|max:100',

            'ingredientes' => 'required|string',

            'preco' => 'required|numeric|min:0',

            'tamanho' => 'required|in:Pequena,Media,Grande',

            'categoria' => 'required|string|max:50',

            'estoque\_minimo' => 'required|integer|min:0',

            'ativo' => 'nullable|boolean',

        ]);

        // Normalizar preço (vírgula para ponto)

        $data['preco'] = str\_replace(',', '.', $data['preco']);

        $data['ativo'] = (*bool*)($data['ativo'] ?? true);

        try {

            $pizza->update($data);

*Log*::info('Pizza atualizada', ['pizza' => $pizza->nome]);

            return redirect()->route('pizzas.index')

                ->with('success', 'Pizza atualizada com sucesso!');

        } catch (\*Exception* $e) {

*Log*::error('Erro ao atualizar pizza', ['erro' => $e->getMessage()]);

            return redirect()->route('pizzas.index')

                ->with('error', 'Erro ao atualizar pizza: ' . $e->getMessage());

        }

    }

    public *function* destroy(*Pizza* $pizza)

    {

        try {

            $nome = $pizza->nome;

            $pizza->delete();

*Log*::info('Pizza excluída', ['pizza' => $nome]);

            return redirect()->route('pizzas.index')

                ->with('success', 'Pizza excluída com sucesso!');

        } catch (\*Exception* $e) {

*Log*::error('Erro ao excluir pizza', ['erro' => $e->getMessage()]);

            return redirect()->route('pizzas.index')

                ->with('error', 'Erro ao excluir pizza: ' . $e->getMessage());

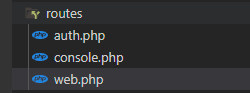
        }

    }

}

**Passo 5 - Configurar Rotas**

***Em routes***

****

**No arquivo web.php**

<?php

use App\Http\Controllers\Auth\*LoginController*;

use App\Http\Controllers\*HistoricoController*;

use App\Http\Controllers\*MovimentacaoController*;

use App\Http\Controllers\*PizzaController*;

use Illuminate\Support\Facades\*Route*;

// Rota principal redireciona para pizzas

*Route*::get('/', *function* () {

    return redirect()->route('pizzas.index');

});

// Rotas de autenticação

*Route*::get('/login', [*LoginController*::class, 'showLoginForm'])->name('login');

*Route*::post('/login', [*LoginController*::class, 'login']);

*Route*::post('/logout', [*LoginController*::class, 'logout'])->name('logout');

// Rotas protegidas

*Route*::middleware('auth:usuarios')->group(*function* () {

    // Pizzas

*Route*::get('/pizzas', [*PizzaController*::class, 'index'])->name('pizzas.index');

*Route*::post('/pizzas', [*PizzaController*::class, 'store'])->name('pizzas.store');

*Route*::get('/pizzas/{pizza}/edit', [*PizzaController*::class, 'edit'])->name('pizzas.edit');

*Route*::put('/pizzas/{pizza}', [*PizzaController*::class, 'update'])->name('pizzas.update');

*Route*::delete('/pizzas/{pizza}', [*PizzaController*::class, 'destroy'])->name('pizzas.destroy');

    // Movimentações

*Route*::get('/movimentacoes', [*MovimentacaoController*::class, 'index'])->name('movimentacoes.index');

*Route*::post('/movimentacoes', [*MovimentacaoController*::class, 'store'])->name('movimentacoes.store');

    // Histórico

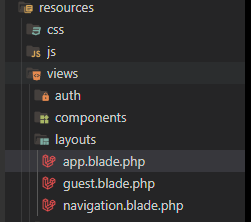
*Route*::get('/historico', [*HistoricoController*::class, 'index'])->name('historico.index');

*Route*::post('/logout', [*LoginController*::class, 'logout'])->name('logout');

});

**Passo 5 - Criar Views (Blade Templates)**

***Em resources>views>layouts***

****

**Arquivo app.blade.php**

<!DOCTYPE html>

<html lang="pt-BR">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>@yield('title', 'Pizzaria')</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">

    <style>

        .estoque-baixo { *color*: #dc3545; *font-weight*: bold; }

        .estoque-normal { *color*: #28a745; }

        .navbar-brand { *font-weight*: bold; }

        .card { *box-shadow*: 0 0.125rem 0.25rem rgba(0, 0, 0, 0.075); }

        .table th { *background-color*: #f8f9fa; }

    </style>

</head>

<body>

    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">

        <div class="container">

            <a class="navbar-brand" href="{{ route('pizzas.index') }}">

                <i class="fas fa-pizza-slice"></i> Pizzaria

            </a>

            <div class="navbar-nav ms-auto">

                @auth('usuarios')

                    <div class="nav-item dropdown">

                        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">

                            <i class="fas fa-user"></i> {{ auth('usuarios')->user()->nome }}

                        </a>

                        <ul class="dropdown-menu">

                            <li><a class="dropdown-item" href="{{ route('pizzas.index') }}">

                                <i class="fas fa-list"></i> Pizzas

                            </a></li>

                            <li><a class="dropdown-item" href="{{ route('movimentacoes.index') }}">

                                <i class="fas fa-exchange-alt"></i> Movimentações

                            </a></li>

                            <li><a class="dropdown-item" href="{{ route('historico.index') }}">

                                <i class="fas fa-history"></i> Histórico

                            </a></li>

                            <li><hr class="dropdown-divider"></li>

                            <li>

                                <form method="POST" action="{{ route('logout') }}" class="d-inline">

                                    @csrf

                                    <button type="submit" class="dropdown-item">

                                        <i class="fas fa-sign-out-alt"></i> Sair

                                    </button>

                                </form>

                            </li>

                        </ul>

                    </div>

                @endauth

            </div>

        </div>

    </nav>

    <div class="container mt-4">

        @if(session('success'))

            <div class="alert alert-success alert-dismissible fade show" role="alert">

                <i class="fas fa-check-circle"></i> {{ session('success') }}

                <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

            </div>

        @endif

        @if(session('error'))

            <div class="alert alert-danger alert-dismissible fade show" role="alert">

                <i class="fas fa-exclamation-circle"></i> {{ session('error') }}

                <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

            </div>

        @endif

        @if($errors->any())

            <div class="alert alert-danger alert-dismissible fade show" role="alert">

                <i class="fas fa-exclamation-triangle"></i>

                <ul class="mb-0">

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

                        <li>{{ $error }}</li>

                    @endforeach

                </ul>

                <button type="button" class="btn-close" data-bs-dismiss="alert"></button>

            </div>

        @endif

        @yield('content')

    </div>

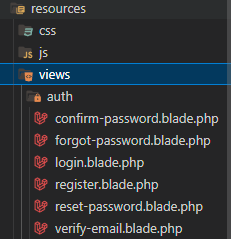
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

    @yield('scripts')

</body>

</html>

**Em resources>views>auth**

****

**Em *resources>views>auth***

**login.blade.php**

<!DOCTYPE html>

<html lang="pt-BR">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login - Pizzaria</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet">

    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">

    <style>

        body { *background-color*: #f8f9fa; }

        .login-card { *max-width*: 400px; *margin*: 100px auto; }

        .card { *box-shadow*: 0 0.5rem 1rem rgba(0, 0, 0, 0.15); }

    </style>

</head>

<body>

    <div class="container">

        <div class="login-card">

            <div class="card">

                <div class="card-body p-5">

                    <div class="text-center mb-4">

                        <i class="fas fa-pizza-slice fa-3x text-primary"></i>

                        <h3 class="mt-3">Pizzaria</h3>

                        <p class="text-muted">Sistema de Gestão</p>

                    </div>

                    @if($errors->any())

                        <div class="alert alert-danger">

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

                                <div>{{ $error }}</div>

                            @endforeach

                        </div>

                    @endif

                    <form method="POST" action="{{ route('login') }}">

                        @csrf

                        <div class="mb-3">

                            <label for="nome" class="form-label">Usuário</label>

                            <div class="input-group">

                                <span class="input-group-text"><i class="fas fa-user"></i></span>

                                <input type="text" class="form-control @error('nome') is-invalid @enderror"

                                       id="nome" name="nome" value="{{ old('nome') }}" required autofocus>

                            </div>

                        </div>

                        <div class="mb-3">

                            <label for="senha" class="form-label">Senha</label>

                            <div class="input-group">

                                <span class="input-group-text"><i class="fas fa-lock"></i></span>

                                <input type="password" class="form-control @error('senha') is-invalid @enderror"

                                       id="senha" name="senha" required>

                            </div>

                        </div>

                        <button type="submit" class="btn btn-primary w-100">

                            <i class="fas fa-sign-in-alt"></i> Entrar

                        </button>

                    </form>

                    <div class="mt-4 text-center">

                        <small class="text-muted">

                            Usuários de teste:<br>

                            Admin / admin123<br>

                            Gerente / gerente123<br>

                            Atendente / atendente123

                        </small>

                    </div>

                </div>

            </div>

        </div>

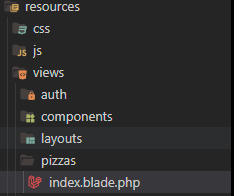
    </div>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

***Dentro de resources>views>pizzas (criar a pasta pizzas)***

****

**Crie o arquivo index.blade.php**

@extends('layouts.app')

@section('title', 'Pizzas')

@section('content')

<div class="row">

    <div class="col-12">

        <div class="d-flex justify-content-between align-items-center mb-4">

            <h2><i class="fas fa-pizza-slice"></i> Pizzas</h2>

            <button class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#cadastrarModal">

                <i class="fas fa-plus"></i> Nova Pizza

            </button>

        </div>

        @if(isset($debug) && $debug)

            <div class="alert alert-info">

                <h5>Debug Mode</h5>

                <strong>POST:</strong>

                <pre>{{ print\_r(request()->all(), true) }}</pre>

                <strong>SESSION:</strong>

                <pre>{{ print\_r(session()->all(), true) }}</pre>

            </div>

        @endif

        <div class="card">

            <div class="card-body">

                <div class="table-responsive">

                    <table class="table table-hover">

                        <thead>

                            <tr>

                                <th>Nome</th>

                                <th>Ingredientes</th>

                                <th>Preço</th>

                                <th>Tamanho</th>

                                <th>Categoria</th>

                                <th>Estoque Atual</th>

                                <th>Estoque Mínimo</th>

                                <th>Status</th>

                                <th>Ações</th>

                            </tr>

                        </thead>

                        <tbody>

                            @forelse($pizzas as $pizza)

                                <tr>

                                    <td><strong>{{ $pizza->nome }}</strong></td>

                                    <td>{{ Str::limit($pizza->ingredientes, 50) }}</td>

                                    <td>R$ {{ number\_format($pizza->preco, 2, ',', '.') }}</td>

                                    <td>

                                        <span class="badge bg-secondary">{{ $pizza->tamanho }}</span>

                                    </td>

                                    <td>{{ $pizza->categoria }}</td>

                                    <td>

                                        <span class="badge {{ $pizza->estoque\_baixo ? 'bg-danger' : 'bg-success' }}">

                                            {{ $pizza->estoque\_atual }}

                                        </span>

                                    </td>

                                    <td>{{ $pizza->estoque\_minimo }}</td>

                                    <td>

                                        @if($pizza->ativo)

                                            <span class="badge bg-success">Ativo</span>

                                        @else

                                            <span class="badge bg-secondary">Inativo</span>

                                        @endif

                                    </td>

                                    <td>

                                        <div class="btn-group btn-group-sm">

                                            <button class="btn btn-outline-primary"

                                                    data-bs-toggle="modal"

                                                    data-bs-target="#editarModal"

                                                    data-pizza='@json($pizza)'>

                                                <i class="fas fa-edit"></i>

                                            </button>

                                            <form method="POST" action="{{ route('pizzas.destroy', $pizza) }}"

                                                  class="d-inline"

                                                  onsubmit="return confirm('Tem certeza que deseja excluir esta pizza?')">

                                                @csrf

                                                @method('DELETE')

                                                <button type="submit" class="btn btn-outline-danger">

                                                    <i class="fas fa-trash"></i>

                                                </button>

                                            </form>

                                        </div>

                                    </td>

                                </tr>

                            @empty

                                <tr>

                                    <td colspan="9" class="text-center text-muted">

                                        <i class="fas fa-pizza-slice fa-2x mb-2"></i><br>

                                        Nenhuma pizza cadastrada

                                    </td>

                                </tr>

                            @endforelse

                        </tbody>

                    </table>

                </div>

                {{ $pizzas->links() }}

            </div>

        </div>

    </div>

</div>

<!-- Modal Cadastrar -->

<div class="modal fade" id="cadastrarModal" tabindex="-1">

    <div class="modal-dialog modal-lg">

        <div class="modal-content">

            <form method="POST" action="{{ route('pizzas.store') }}">

                @csrf

                <div class="modal-header">

                    <h5 class="modal-title"><i class="fas fa-plus"></i> Nova Pizza</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal"></button>

                </div>

                <div class="modal-body">

                    <div class="row">

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="nome" class="form-label">Nome \*</label>

                                <input type="text" class="form-control" id="nome" name="nome" required>

                            </div>

                        </div>

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="preco" class="form-label">Preço \*</label>

                                <input type="text" class="form-control" id="preco" name="preco"

                                       placeholder="25,90" required>

                            </div>

                        </div>

                    </div>

                    <div class="row">

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="tamanho" class="form-label">Tamanho \*</label>

                                <select class="form-select" id="tamanho" name="tamanho" required>

                                    <option value="">Selecione...</option>

                                    <option value="Pequena">Pequena</option>

                                    <option value="Media">Média</option>

                                    <option value="Grande">Grande</option>

                                </select>

                            </div>

                        </div>

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="categoria" class="form-label">Categoria \*</label>

                                <input type="text" class="form-control" id="categoria" name="categoria" required>

                            </div>

                        </div>

                    </div>

                    <div class="mb-3">

                        <label for="ingredientes" class="form-label">Ingredientes \*</label>

                        <textarea class="form-control" id="ingredientes" name="ingredientes" rows="3" required></textarea>

                    </div>

                    <div class="row">

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="estoque\_minimo" class="form-label">Estoque Mínimo \*</label>

                                <input type="number" class="form-control" id="estoque\_minimo" name="estoque\_minimo"

                                       min="0" value="5" required>

                            </div>

                        </div>

                        <div class="col-md-6">

                            <div class="mb-3">

                                <div class="form-check mt-4">

                                    <input class="form-check-input" type="checkbox" id="ativo" name="ativo" value="1" checked>

                                    <label class="form-check-label" for="ativo">

                                        Pizza ativa

                                    </label>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

                    <button type="submit" class="btn btn-primary">Cadastrar</button>

                </div>

            </form>

        </div>

    </div>

</div>

<!-- Modal Editar -->

<div class="modal fade" id="editarModal" tabindex="-1">

    <div class="modal-dialog modal-lg">

        <div class="modal-content">

            <form method="POST" id="editarForm">

                @csrf

                @method('PUT')

                <div class="modal-header">

                    <h5 class="modal-title"><i class="fas fa-edit"></i> Editar Pizza</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal"></button>

                </div>

                <div class="modal-body">

                    <div class="row">

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="edit\_nome" class="form-label">Nome \*</label>

                                <input type="text" class="form-control" id="edit\_nome" name="nome" required>

                            </div>

                        </div>

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="edit\_preco" class="form-label">Preço \*</label>

                                <input type="text" class="form-control" id="edit\_preco" name="preco" required>

                            </div>

                        </div>

                    </div>

                    <div class="row">

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="edit\_tamanho" class="form-label">Tamanho \*</label>

                                <select class="form-select" id="edit\_tamanho" name="tamanho" required>

                                    <option value="Pequena">Pequena</option>

                                    <option value="Media">Média</option>

                                    <option value="Grande">Grande</option>

                                </select>

                            </div>

                        </div>

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="edit\_categoria" class="form-label">Categoria \*</label>

                                <input type="text" class="form-control" id="edit\_categoria" name="categoria" required>

                            </div>

                        </div>

                    </div>

                    <div class="mb-3">

                        <label for="edit\_ingredientes" class="form-label">Ingredientes \*</label>

                        <textarea class="form-control" id="edit\_ingredientes" name="ingredientes" rows="3" required></textarea>

                    </div>

                    <div class="row">

                        <div class="col-md-6">

                            <div class="mb-3">

                                <label for="edit\_estoque\_minimo" class="form-label">Estoque Mínimo \*</label>

                                <input type="number" class="form-control" id="edit\_estoque\_minimo" name="estoque\_minimo"

                                       min="0" required>

                            </div>

                        </div>

                        <div class="col-md-6">

                            <div class="mb-3">

                                <div class="form-check mt-4">

                                    <input class="form-check-input" type="checkbox" id="edit\_ativo" name="ativo" value="1">

                                    <label class="form-check-label" for="edit\_ativo">

                                        Pizza ativa

                                    </label>

                                </div>

                            </div>

                        </div>

                    </div>

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancelar</button>

                    <button type="submit" class="btn btn-primary">Atualizar</button>

                </div>

            </form>

        </div>

    </div>

</div>

@endsection

@section('scripts')

<script>

document.addEventListener('DOMContentLoaded', *function*() {

    // Preencher modal de edição

    document.getElementById('editarModal').addEventListener('show.bs.modal', *function* (*event*) {

*const* button = event.relatedTarget;

*const* pizza = JSON.parse(button.getAttribute('data-pizza'));

        document.getElementById('editarForm').action = `/pizzas/${pizza.id}`;

        document.getElementById('edit\_nome').value = pizza.nome;

        document.getElementById('edit\_preco').value = pizza.preco;

        document.getElementById('edit\_tamanho').value = pizza.tamanho;

        document.getElementById('edit\_categoria').value = pizza.categoria;

        document.getElementById('edit\_ingredientes').value = pizza.ingredientes;

        document.getElementById('edit\_estoque\_minimo').value = pizza.estoque\_minimo;

        document.getElementById('edit\_ativo').checked = pizza.ativo;

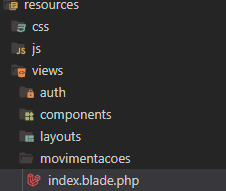
    });

});

</script>

@endsection

***Resources>views>movimentacoes (criar a pasta movimentacoes)***

****

**Criar o arquivo index.blade.php**

@extends('layouts.app')

@section('title', 'Movimentações')

@section('content')

<div class="row">

    <div class="col-12">

        <div class="d-flex justify-content-between align-items-center mb-4">

            <h2><i class="fas fa-exchange-alt"></i> Movimentações de Estoque</h2>

            <a href="{{ route('historico.index') }}" class="btn btn-outline-primary">

                <i class="fas fa-history"></i> Ver Histórico Completo

            </a>

        </div>

        @if(isset($debug) && $debug)

            <div class="alert alert-info">

                <h5>Debug Mode</h5>

                <strong>POST:</strong>

                <pre>{{ print\_r(request()->all(), true) }}</pre>

                <strong>SESSION:</strong>

                <pre>{{ print\_r(session()->all(), true) }}</pre>

            </div>

        @endif

        <!-- Formulário de Movimentação -->

        <div class="card mb-4">

            <div class="card-header">

                <h5 class="mb-0"><i class="fas fa-plus"></i> Nova Movimentação</h5>

            </div>

            <div class="card-body">

                <form method="POST" action="{{ route('movimentacoes.store') }}">

                    @csrf

                    <div class="row">

                        <div class="col-md-3">

                            <div class="mb-3">

                                <label for="pizza\_id" class="form-label">Pizza \*</label>

                                <select class="form-select @error('pizza\_id') is-invalid @enderror"

                                        id="pizza\_id" name="pizza\_id" required>

                                    <option value="">Selecione uma pizza...</option>

                                    @foreach($pizzas as $pizza)

                                        <option value="{{ $pizza->id }}"

                                                data-estoque="{{ $pizza->estoque\_atual }}"

                                                {{ old('pizza\_id') == $pizza->id ? 'selected' : '' }}>

                                            {{ $pizza->nome }} (Estoque: {{ $pizza->estoque\_atual }})

                                        </option>

                                    @endforeach

                                </select>

                                @error('pizza\_id')

                                    <div class="invalid-feedback">{{ $message }}</div>

                                @enderror

                            </div>

                        </div>

                        <div class="col-md-2">

                            <div class="mb-3">

                                <label for="tipo" class="form-label">Tipo \*</label>

                                <select class="form-select @error('tipo') is-invalid @enderror"

                                        id="tipo" name="tipo" required>

                                    <option value="">Selecione...</option>

                                    <option value="entrada" {{ old('tipo') == 'entrada' ? 'selected' : '' }}>

                                        Entrada

                                    </option>

                                    <option value="saida" {{ old('tipo') == 'saida' ? 'selected' : '' }}>

                                        Saída

                                    </option>

                                </select>

                                @error('tipo')

                                    <div class="invalid-feedback">{{ $message }}</div>

                                @enderror

                            </div>

                        </div>

                        <div class="col-md-2">

                            <div class="mb-3">

                                <label for="quantidade" class="form-label">Quantidade \*</label>

                                <input type="number" class="form-control @error('quantidade') is-invalid @enderror"

                                       id="quantidade" name="quantidade" min="1"

                                       value="{{ old('quantidade') }}" required>

                                @error('quantidade')

                                    <div class="invalid-feedback">{{ $message }}</div>

                                @enderror

                            </div>

                        </div>

                        <div class="col-md-3">

                            <div class="mb-3">

                                <label for="observacoes" class="form-label">Observações</label>

                                <input type="text" class="form-control" id="observacoes" name="observacoes"

                                       value="{{ old('observacoes') }}"

                                       placeholder="Ex: Reposição de estoque">

                            </div>

                        </div>

                        <div class="col-md-2">

                            <div class="mb-3">

                                <label class="form-label">&nbsp;</label>

                                <button type="submit" class="btn btn-primary w-100">

                                    <i class="fas fa-save"></i> Registrar

                                </button>

                            </div>

                        </div>

                    </div>

                </form>

            </div>

        </div>

        <!-- Lista de Movimentações Recentes -->

        <div class="card">

            <div class="card-header">

                <h5 class="mb-0"><i class="fas fa-list"></i> Movimentações Recentes</h5>

            </div>

            <div class="card-body">

                <div class="table-responsive">

                    <table class="table table-hover">

                        <thead>

                            <tr>

                                <th>Data/Hora</th>

                                <th>Pizza</th>

                                <th>Tipo</th>

                                <th>Quantidade</th>

                                <th>Usuário</th>

                                <th>Observações</th>

                            </tr>

                        </thead>

                        <tbody>

                            @forelse($movimentacoes as $movimentacao)

                                <tr>

                                    <td>{{ $movimentacao->data\_hora->format('d/m/Y H:i') }}</td>

                                    <td><strong>{{ $movimentacao->pizza->nome }}</strong></td>

                                    <td>

                                        @if($movimentacao->tipo === 'entrada')

                                            <span class="badge bg-success">

                                                <i class="fas fa-arrow-up"></i> Entrada

                                            </span>

                                        @else

                                            <span class="badge bg-danger">

                                                <i class="fas fa-arrow-down"></i> Saída

                                            </span>

                                        @endif

                                    </td>

                                    <td>

                                        <strong>{{ $movimentacao->quantidade }}</strong>

                                    </td>

                                    <td>{{ $movimentacao->usuario->nome }}</td>

                                    <td>{{ $movimentacao->observacoes ?? '-' }}</td>

                                </tr>

                            @empty

                                <tr>

                                    <td colspan="6" class="text-center text-muted">

                                        <i class="fas fa-exchange-alt fa-2x mb-2"></i><br>

                                        Nenhuma movimentação registrada

                                    </td>

                                </tr>

                            @endforelse

                        </tbody>

                    </table>

                </div>

                {{ $movimentacoes->links() }}

            </div>

        </div>

    </div>

</div>

@endsection

@section('scripts')

<script>

document.addEventListener('DOMContentLoaded', *function*() {

*const* pizzaSelect = document.getElementById('pizza\_id');

*const* tipoSelect = document.getElementById('tipo');

*const* quantidadeInput = document.getElementById('quantidade');

    // Atualizar validação quando pizza ou tipo mudarem

*function* updateValidation() {

*const* selectedOption = pizzaSelect.options[pizzaSelect.selectedIndex];

*const* estoqueAtual = selectedOption ? parseInt(selectedOption.dataset.estoque) : 0;

*const* tipo = tipoSelect.value;

*const* quantidade = parseInt(quantidadeInput.value) || 0;

        if (tipo === 'saida' && quantidade > estoqueAtual) {

            quantidadeInput.setCustomValidity(`Quantidade maior que o estoque atual (${estoqueAtual})`);

        } else {

            quantidadeInput.setCustomValidity('');

        }

    }

    pizzaSelect.addEventListener('change', updateValidation);

    tipoSelect.addEventListener('change', updateValidation);

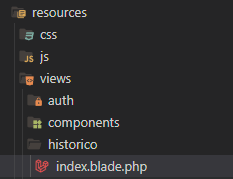
    quantidadeInput.addEventListener('input', updateValidation);

});

</script>

@endsection

***Resources>views>historico (criar a pasta historico)***

****

**Criar o arquivo index.blade.php**

@extends('layouts.app')

@section('title', 'Histórico de Movimentações')

@section('content')

<div class="row">

    <div class="col-12">

        <div class="d-flex justify-content-between align-items-center mb-4">

            <h2><i class="fas fa-history"></i> Histórico de Movimentações</h2>

            <a href="{{ route('movimentacoes.index') }}" class="btn btn-outline-primary">

                <i class="fas fa-exchange-alt"></i> Nova Movimentação

            </a>

        </div>

        <!-- Filtros -->

        <div class="card mb-4">

            <div class="card-header">

                <h5 class="mb-0"><i class="fas fa-filter"></i> Filtros</h5>

            </div>

            <div class="card-body">

                <form method="GET" action="{{ route('historico.index') }}">

                    <div class="row">

                        <div class="col-md-3">

                            <div class="mb-3">

                                <label for="pizza\_id" class="form-label">Pizza</label>

                                <select class="form-select" id="pizza\_id" name="pizza\_id">

                                    <option value="">Todas as pizzas</option>

                                    @foreach($pizzas as $pizza)

                                        <option value="{{ $pizza->id }}"

                                                {{ request('pizza\_id') == $pizza->id ? 'selected' : '' }}>

                                            {{ $pizza->nome }}

                                        </option>

                                    @endforeach

                                </select>

                            </div>

                        </div>

                        <div class="col-md-2">

                            <div class="mb-3">

                                <label for="tipo" class="form-label">Tipo</label>

                                <select class="form-select" id="tipo" name="tipo">

                                    <option value="">Todos</option>

                                    <option value="entrada" {{ request('tipo') == 'entrada' ? 'selected' : '' }}>

                                        Entrada

                                    </option>

                                    <option value="saida" {{ request('tipo') == 'saida' ? 'selected' : '' }}>

                                        Saída

                                    </option>

                                </select>

                            </div>

                        </div>

                        <div class="col-md-2">

                            <div class="mb-3">

                                <label for="de" class="form-label">De</label>

                                <input type="date" class="form-control" id="de" name="de"

                                       value="{{ request('de') }}">

                            </div>

                        </div>

                        <div class="col-md-2">

                            <div class="mb-3">

                                <label for="ate" class="form-label">Até</label>

                                <input type="date" class="form-control" id="ate" name="ate"

                                       value="{{ request('ate') }}">

                            </div>

                        </div>

                        <div class="col-md-3">

                            <div class="mb-3">

                                <label class="form-label">&nbsp;</label>

                                <div class="d-flex gap-2">

                                    <button type="submit" class="btn btn-primary">

                                        <i class="fas fa-search"></i> Filtrar

                                    </button>

                                    <a href="{{ route('historico.index') }}" class="btn btn-outline-secondary">

                                        <i class="fas fa-times"></i> Limpar

                                    </a>

                                </div>

                            </div>

                        </div>

                    </div>

                </form>

            </div>

        </div>

        <!-- Lista de Movimentações -->

        <div class="card">

            <div class="card-header">

                <h5 class="mb-0">

                    <i class="fas fa-list"></i> Movimentações

                    @if(request()->hasAny(['pizza\_id', 'tipo', 'de', 'ate']))

                        <small class="text-muted">(Filtradas)</small>

                    @endif

                </h5>

            </div>

            <div class="card-body">

                <div class="table-responsive">

                    <table class="table table-hover">

                        <thead>

                            <tr>

                                <th>Data/Hora</th>

                                <th>Pizza</th>

                                <th>Tipo</th>

                                <th>Quantidade</th>

                                <th>Usuário</th>

                                <th>Observações</th>

                            </tr>

                        </thead>

                        <tbody>

                            @forelse($movimentacoes as $movimentacao)

                                <tr>

                                    <td>

                                        <strong>{{ $movimentacao->data\_hora->format('d/m/Y') }}</strong><br>

                                        <small class="text-muted">{{ $movimentacao->data\_hora->format('H:i:s') }}</small>

                                    </td>

                                    <td>

                                        <strong>{{ $movimentacao->pizza->nome }}</strong><br>

                                        <small class="text-muted">{{ $movimentacao->pizza->categoria }}</small>

                                    </td>

                                    <td>

                                        @if($movimentacao->tipo === 'entrada')

                                            <span class="badge bg-success">

                                                <i class="fas fa-arrow-up"></i> Entrada

                                            </span>

                                        @else

                                            <span class="badge bg-danger">

                                                <i class="fas fa-arrow-down"></i> Saída

                                            </span>

                                        @endif

                                    </td>

                                    <td>

                                        <strong class="{{ $movimentacao->tipo === 'entrada' ? 'text-success' : 'text-danger' }}">

                                            {{ $movimentacao->tipo === 'entrada' ? '+' : '-' }}{{ $movimentacao->quantidade }}

                                        </strong>

                                    </td>

                                    <td>{{ $movimentacao->usuario->nome }}</td>

                                    <td>{{ $movimentacao->observacoes ?? '-' }}</td>

                                </tr>

                            @empty

                                <tr>

                                    <td colspan="6" class="text-center text-muted">

                                        <i class="fas fa-history fa-2x mb-2"></i><br>

                                        Nenhuma movimentação encontrada

                                    </td>

                                </tr>

                            @endforelse

                        </tbody>

                    </table>

                </div>

                {{ $movimentacoes->appends(request()->query())->links() }}

            </div>

        </div>

    </div>

</div>

@endsection

**Agora rode o comando**

**php artisan session:table**

**php artisan migrate**

**php artisan serve**