

```

class Produto extends Model
{
    use HasFactory;

    protected $table = "produto";

    protected $primaryKey = "idproduto";

    public $timestamps = false;

    protected $fillable = [
        'idmotor',
        'idcarro',
        'idvalvulas',
        'idfabricacao',
        'idcategoria',
        'idmarca',
        'idlocalizacao',
        'icms',
        'ipi',
        'frete',
        'valornafabrica',
        'valordecompra',
        'lucro',
        'valorvenda',
        'desconto',
        'quantidade',
        'unidade',
        'referencia'
    ];

    function itensvenda()
    {
        return $this->hasMany(ItensVenda::class, 'idproduto', 'id');
    }

    function itensentrada()
    {
        return $this->hasMany(ItensEntrada::class, 'idproduto', 'id'
    );
    }

    function motor()
    {
        return $this->belongsTo(Motor::class, 'idmotor', 'idmotor');
    }

    function carro()
    {
        return $this->belongsTo(Carro::class, 'idcarro', 'id');
    }

    function valvulas()
    {
        return $this->belongsTo(Valvulas::class, 'idvalvulas',
'idvalvulas');
    }

    function fabricacao()
    {
        return $this->belongsTo(Fabricacao::class, 'idfabricacao',
'id');
    }

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

    function marca()
    {
        return $this->belongsTo(Marca::class, 'idmarca', 'idmarca');
    }

    function localizacao()
    {
        return $this->belongsTo(Localizacao::class, 'idlocalizacao'
, 'idlocalizacao');
    }
}

```

```
class Venda extends Model
{
    use HasFactory;

    protected $table = "venda";

    protected $primaryKey = "idvenda";

    public $timestamps = false;

    protected $fillable = [
        'idcliente',
        'idformapagamento',
        'data',
        'valortotal',
        'status'
    ];

    public function cliente()
    {
        return $this->belongsTo(Cliente::class, 'idcliente',
'idcliente');
    }

    public function formapagamento()
    {
        return $this->belongsTo(FormaPagamento::class,
'idformapagamento', 'idformapagamento');
    }

    public function itensvenda()
    {
        return $this->hasMany(ItensVenda::class, 'idvenda',
'idvenda');
    }

    public function countItensVenda()
    {
        return $this->itensvenda()->sum('quantidade');
    }
}
```

```
Arquivo Editor Seleção Ver Acesso Executar Terminal Ajuda web.php - Inspecionar | Visual Studio Code
web.php - Controller.php
routes > web.php > #functionfe0c558 > #functionf00cf037

36 Route::prefix('dashboard')->middleware('auth')->group(function () {
37     Route::get('/', function () {
38         return view('dashboard.index', [
39             'produtosnaivevidios' => DB::select("select * from produtosnaivevidios" . date("m" . " "));
40         ]);
41     }->name('dashboard.index');
42 }
43
44 Route::prefix('venda')->group(function () {
45     Route::get('/', [VendaController::class, 'index']->name('venda.index');
46     Route::get('{idvenda}/itensvenda/{idproduto}', [VendaController::class, 'itensvenda']->name('venda.itensvenda');
47     Route::get('{idvenda}/inserviten', [VendaController::class, 'inserviten']->name('venda.inserviten');
48     Route::post('{inserviten}/itensvenda', [VendaController::class, 'store']->name('itensvenda.store');
49     Route::post('{idvenda}/finalizarvenda', [VendaController::class, 'finalizarvenda']->name('venda.finalizarvenda');
50     Route::post('/store', [VendaController::class, 'store']->name('venda.store');
51     Route::post('/getAutocomplete', [VendaController::class, 'getAutocomplete']->name('venda.getAutocomplete');
52 }
53 }
54
55 Route::prefix('entrada')->group(function () {
56     Route::get('/', [EntradaController::class, 'index']->name('entrada.index');
57     Route::get('{identrada}/itensentrada/{idproduto}', [EntradaController::class, 'itensentrada']->name('entrada.itensentrada');
58     Route::get('{identrada}/inserviten', [EntradaController::class, 'inserviten']->name('entrada.inserviten');
59     Route::post('{inserviten}/itensentrada', [EntradaController::class, 'store']->name('itensentrada.store');
60     Route::post('{identrada}/finalizarentrada', [EntradaController::class, 'finalizarentrada']->name('entrada.finalizarentrada');
61     Route::post('/store', [EntradaController::class, 'store']->name('entrada.store');
62     Route::post('/getAutocomplete', [EntradaController::class, 'getAutocomplete']->name('entrada.getAutocomplete');
63 }
64 }
65
66 Route::prefix('consulta')->group(function () {
67     Route::get('/cliente', [ClienteController::class, 'index']->name('consulta.cliente');
68     Route::get('/fornecedor', [FornecedorController::class, 'index']->name('consulta.fornecedor');
69     Route::get('/produto', [ProdutoController::class, 'index']->name('consulta.produto');
70     Route::get('/carro', [CarroController::class, 'index']->name('consulta.carro');
71     Route::get('/localizacao', [LocalizacaoController::class, 'index']->name('consulta.localizacao');
72     Route::get('/valvula', [ValvulaController::class, 'index']->name('consulta.valvula');
73     Route::get('/categoria', [CategoriaController::class, 'index']->name('consulta.categoria');
74     Route::get('/motor', [MotorController::class, 'index']->name('consulta.motor');
75     Route::get('/fabricacao', [FabricacaoController::class, 'index']->name('consulta.fabricacao');
76     Route::get('/marca', [MarcaController::class, 'index']->name('consulta.marca');
77 }
78 }
79
80 Route::prefix('cadastro')->group(function () {
81     Route::get('/cliente', [ClienteController::class, 'create']->name('cadastro.cliente');
82     Route::post('/cliente', [ClienteController::class, 'store']->name('cliente.store');
83     Route::get('/fornecedor', [FornecedorController::class, 'create']->name('cadastro.fornecedor');
84     Route::post('/fornecedor', [FornecedorController::class, 'store']->name('fornecedor.store');
85     Route::get('/produto', [ProdutoController::class, 'create']->name('cadastro.produto');
86     Route::post('/produto', [ProdutoController::class, 'store']->name('produto.store');
87     Route::get('/carro', [CarroController::class, 'create']->name('cadastro.carro');
88     Route::post('/carro', [CarroController::class, 'store']->name('carro.store');
89     Route::get('/motor', [MotorController::class, 'create']->name('cadastro.motor');
90     Route::post('/motor', [MotorController::class, 'store']->name('motor.store');
91     Route::get('/fabricacao', [FabricacaoController::class, 'create']->name('cadastro.fabricacao');
92     Route::post('/fabricacao', [FabricacaoController::class, 'store']->name('fabricacao.store');
93     Route::get('/marca', [MarcaController::class, 'create']->name('cadastro.marca');
94     Route::post('/marca', [MarcaController::class, 'store']->name('marca.store');
95 }
96 }
97
98 }
99
100 }
101
102 }
103
104 }
105
106 }
107
108 }
109
110 }
111
112 }
113
114 }
115
116 }
117
118 }
119
120 }
121
122 }
123
124 }
125
126 }
127
128 }
129
130 }
131
132 }
133
134 }
135
136 }
137
138 }
139
140 }
141
142 }
143
144 }
145
146 }
147
148 }
149
150 }
151
152 }
153
154 }
155
156 }
157
158 }
159
160 }
161
162 }
163
164 }
165
166 }
167
168 }
169
170 }
171
172 }
173
174 }
175
176 }
177
178 }
179
180 }
181
182 }
183
184 }
185
186 }
187
188 }
189
190 }
191
192 }
193
194 }
195
196 }
197
198 }
199
200 }
201
202 }
203
204 }
205
206 }
207
208 }
209
210 }
211
212 }
213
214 }
215
216 }
217
218 }
219
220 }
221
222 }
223
224 }
225
226 }
227
228 }
229
230 }
231
232 }
233
234 }
235
236 }
237
238 }
239
240 }
241
242 }
243
244 }
245
246 }
247
248 }
249
250 }
251
252 }
253
254 }
255
256 }
257
258 }
259
260 }
261
262 }
263
264 }
265
266 }
267
268 }
269
270 }
271
272 }
273
274 }
275
276 }
277
278 }
279
280 }
281
282 }
283
284 }
285
286 }
287
288 }
289
290 }
291
292 }
293
294 }
295
296 }
297
298 }
299
300 }
301
302 }
303
304 }
305
306 }
307
308 }
309
310 }
311
312 }
313
314 }
315
316 }
317
318 }
319
320 }
321
322 }
323
324 }
325
326 }
327
328 }
329
330 }
331
332 }
333
334 }
335
336 }
337
338 }
339
340 }
341
342 }
343
344 }
345
346 }
347
348 }
349
350 }
351
352 }
353
354 }
355
356 }
357
358 }
359
360 }
361
362 }
363
364 }
365
366 }
367
368 }
369
370 }
371
372 }
373
374 }
375
376 }
377
378 }
379
380 }
381
382 }
383
384 }
385
386 }
387
388 }
389
390 }
391
392 }
393
394 }
395
396 }
397
398 }
399
400 }
401
402 }
403
404 }
405
406 }
407
408 }
409
410 }
411
412 }
413
414 }
415
416 }
417
418 }
419
420 }
421
422 }
423
424 }
425
426 }
427
428 }
429
430 }
431
432 }
433
434 }
435
436 }
437
438 }
439
440 }
441
442 }
443
444 }
445
446 }
447
448 }
449
450 }
451
452 }
453
454 }
455
456 }
457
458 }
459
460 }
461
462 }
463
464 }
465
466 }
467
468 }
469
470 }
471
472 }
473
474 }
475
476 }
477
478 }
479
480 }
481
482 }
483
484 }
485
486 }
487
488 }
489
490 }
491
492 }
493
494 }
495
496 }
497
498 }
499
500 }
501
502 }
503
504 }
505
506 }
507
508 }
509
510 }
511
512 }
513
514 }
515
516 }
517
518 }
519
520 }
521
522 }
523
524 }
525
526 }
527
528 }
529
530 }
531
532 }
533
534 }
535
536 }
537
538 }
539
540 }
541
542 }
543
544 }
545
546 }
547
548 }
549
550 }
551
552 }
553
554 }
555
556 }
557
558 }
559
560 }
561
562 }
563
564 }
565
566 }
567
568 }
569
570 }
571
572 }
573
574 }
575
576 }
577
578 }
579
580 }
581
582 }
583
584 }
585
586 }
587
588 }
589
590 }
591
592 }
593
594 }
595
596 }
597
598 }
599
600 }
601
602 }
603
604 }
605
606 }
607
608 }
609
610 }
611
612 }
613
614 }
615
616 }
617
618 }
619
620 }
621
622 }
623
624 }
625
626 }
627
628 }
629
630 }
631
632 }
633
634 }
635
636 }
637
638 }
639
640 }
641
642 }
643
644 }
645
646 }
647
648 }
649
650 }
651
652 }
653
654 }
655
656 }
657
658 }
659
660 }
661
662 }
663
664 }
665
666 }
667
668 }
669
670 }
671
672 }
673
674 }
675
676 }
677
678 }
679
680 }
6
```

```
class StoreItensVendaRequest extends FormRequest
{

    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            'idvenda' => 'required',
            'idproduto' => 'required',
            'valorvenda' => 'required',
            'quantidade' => 'required',
            'desconto' => 'required',
        ];
    }

    public function messages()
    {
        return [
            'idvenda.required' => 'O campo venda é obrigatório.',
            'idproduto.required' => 'O campo produto é obrigatório.',
            'valorvenda.required' =>
                'O campo valor de venda é obrigatório.',
            'quantidade.required' =>
                'O campo quantidade é obrigatório.',
            'desconto.required' => 'O campo desconto é obrigatório.'
        ];
    }
}
```

```
class StoreVendaRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            'idcliente' => 'required',
        ];
    }

    public function messages()
    {
        return [
            'idcliente.required' => 'O campo cliente é obrigatório.',
        ];
    }
}
```

```
public function store(StoreItensVendaRequest $request)
{
    $itemVenda = ItensVenda::create($request->all());

    return redirect()->route('venda.itensvenda', [
        'idvenda' => $itemVenda->idvenda
    ])->with('success', 'Item de venda cadastrado com sucesso!');
}
```

```
public function store(StoreVendaRequest $request)
{
    $venda = Venda::create($request->all());
    return redirect()->route('venda.itensvenda', ['idvenda' => $venda
->idvenda]);
}
```

SOCAPE

INÍCIO VENDER DAR ENTRADA CONSULTAR MINHA CONTA

PRODUTOS MAIS VENDIDOS

SILENCIOSO/TUPER
T9POTE-08SEDAN
R\$ 190,75
2 vendidos

VISUALIZAR

DESENVOLVIDO POR GABRIEL LOBO

SOCAPE

INÍCIO VENDER DAR ENTRADA CONSULTAR MINHA CONTA

CONSULTAR CLIENTE

NOVO CADASTRO

Pesquisar nome...



NOME	TELEFONE	TIPO/IDENTIFICAÇÃO	DÉBITO	AÇÕES
LUCAS AFONSO	31 231233321	Cliente PF / CPF: 055.605.653-05	0	VISUALIZAR APAGAR
FÁBIO LIMA	(77) 9-9999-9999	Cliente PJ / CNPJ: 81.236.532/1554-88	0	VISUALIZAR APAGAR
GEORGE MARTIN	(77) 9-9999-9999	Cliente PJ / CNPJ: 81.236.532/1554-88	0	VISUALIZAR APAGAR
REGIS TAWON	(77) 9-9999-9999	Cliente PJ / CNPJ: 81.236.532/1554-88	0	VISUALIZAR APAGAR
ARTHUR MAGALHÃES	(77) 9-9999-9999	Cliente PJ / CNPJ: 81.236.532/1554-88	0	VISUALIZAR APAGAR
MATHEUS COCA	(77) 9-9999-9999	Cliente PJ / CNPJ: 81.236.532/1554-88	0	VISUALIZAR APAGAR
JOÃO RIBEIRO	(77) 9-9999-9999	Cliente PF / CPF: 000.000.000-00	0	VISUALIZAR APAGAR
MARCOS COELHO	(77) 9-9999-9999	Cliente PF / CPF: 000.000.000-00	0	VISUALIZAR APAGAR
JEFFERSON BATISTA	(77) 9-9999-9999	Cliente PF / CPF: 000.000.000-00	0	VISUALIZAR APAGAR
DANIEL ROCHA	(77) 9-9999-9999	Cliente PF / CPF: 000.000.000-00	0	VISUALIZAR APAGAR





DESENVOLVIDO POR GABRIEL LOBO

REALIZAR VENDA

CLIENTE

REALIZAR VENDA

VENDAS

CLIENTE	FORMA DE PAGAMENTO	DATA	STATUS	ITENS	VALOR TOTAL	AÇÕES	
 GABRIEL LOBO	CARNÊ - A VISTA	03 Feb 2022	FINALIZADA	1	R\$ 343,36	VISUALIZAR	APAGAR
 JEFFERSON BATISTA	CARTÃO - A VISTA	11 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR
 MARCOS COLUNHO	CARTÃO - A PRAZO	11 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR
 JOÃO RIBEIRO	CARNÊ - A PRAZO	11 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR
 DANIEL ROCHA	CARNÊ - A VISTA	11 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR
 ARTHUR MAGALHÃES	CARNÊ - A VISTA	11 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR
 GEORGE MARTIN	CARTÃO - A PRAZO	11 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR
 FÁBIO LIMA	CARNÊ - A PRAZO	11 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR
 REGIS	CARTÃO - A VISTA	10 Mar 2021	ABERTA	0	R\$ 0	VISUALIZAR	APAGAR

DESENVOLVIDO POR GABRIEL LOBO