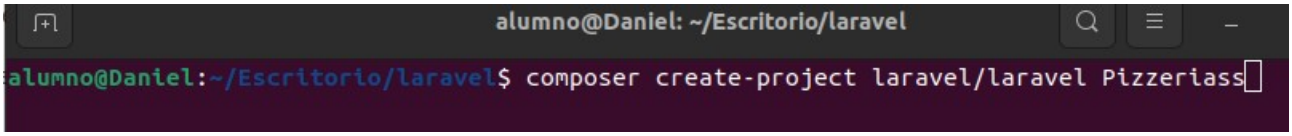
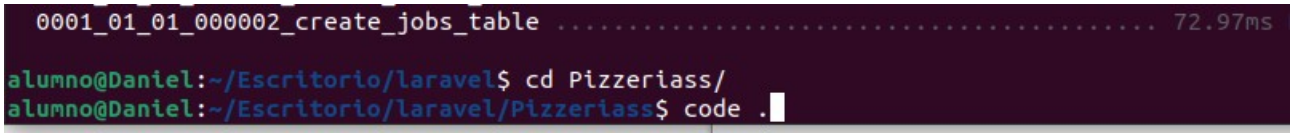
Crud con laravel

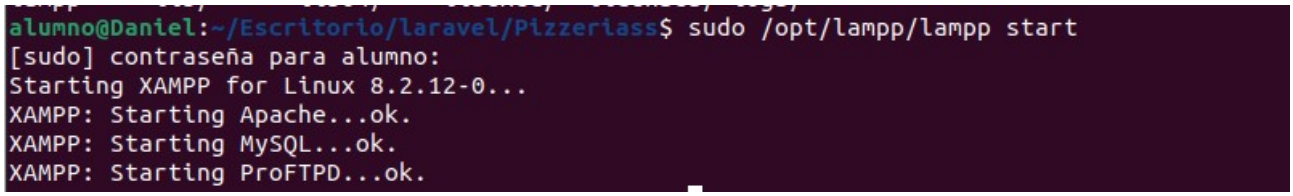
Creamos el projecto en un terminal



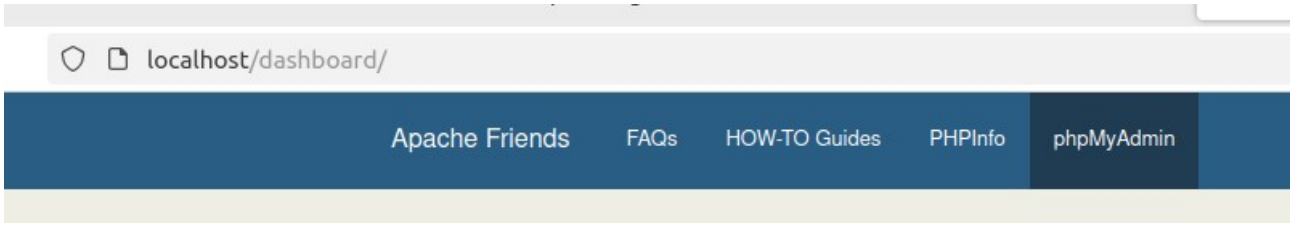
Nos vemos a la carpeta y abrimos visual



Arrancamos Xamp



Verficamos que se abrio correctamente



En visual

Fichero .env realizamos las siguientes modificaciones:

```
23
24  DB_CONNECTION=mysql
25  DB_HOST=127.0.0.1
26  DB_PORT=3306
27  DB_DATABASE=pizzeriass
28  DB_USERNAME=root
29  DB_PASSWORD=
30
```

DB_DATABASE: introducir el nombre de la BBDD sobre la cual vamos a trabajar, no hace falta que este creada, laravel la crea.

Abrimos terminal de visual para crear los modelos y/o tablas:

Para crear un modelo:

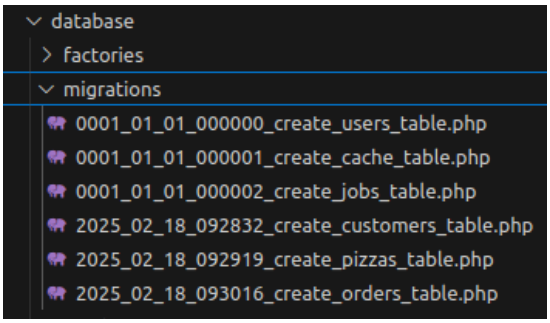
Hay que tener en cuenta las relaciones, se crean primero los modelos que no tienen FK.
El nombre debe ser con mayuscula , seguido de un -m para que te cree la migracion (“php artisan make:model Customer -m”);
No se crean modelos para tablas pivote.

```
alumno@Daniel:~/Escritorio/laravel/Pizzeriass$ php artisan make:model Customer -m

  INFO  Model [app/Models/Customer.php] created successfully.

  INFO  Migration [database/migrations/2025_02_18_092832_create_customers_table.php] created successfully.
```

```
alumno@Daniel:~/Escritorio/laravel/Pizzeriass$ php artisan make:model Pizza -m

  INFO  Model [app/Models/Pizza.php] created successfully.

  INFO  Migration [database/migrations/2025_02_18_092919_create_pizzas_table.php] created successfully.
```

```
alumno@Daniel:~/Escritorio/laravel/Pizzeriass$ php artisan make:model Order -m

  INFO  Model [app/Models/Order.php] created successfully.

  INFO  Migration [database/migrations/2025_02_18_093016_create_orders_table.php] created successfully.
```

Si todo salio bien, este seria el resultado



Creacion de Tabla pivote(en caso de ser necesaria):
create_orders_pizzas(siempre en orden alfabetico y en plural)
--create = order_pizza(siempre en orden alfabetico y en singular)

```
alumno@Daniel:~/Escritorio/laravel/Pizzeriass$ php artisan make:migration create_orders_pizzas_table --create=order_pizza

  INFO  Migration [database/migrations/2025_02_18_094234_create_orders_pizzas_table.php] created successfully.
```

Modificacion de las migraciones(segun enunciado):

Migracion de pizzas

Tener en cuenta que usa softDeletes→ hay que importarlos en su Modelo

```php
public function up(): void
{
    Schema::create('pizzas', function (Blueprint $table) {
        $table->id();
        $table->string('nombre');
        $table->text('descripcion');
        $table->double('precio');
        $table->timestamps();
        $table->softDeletes();
    });
}
```

Migracion customers

```php
public function up(): void
{
    Schema::create('customers', function (Blueprint $table) {
        $table->id();
        $table->string('nombre');
        $table->string('direccion');
        $table->string('telefono');
        $table->string('email');
        $table->timestamps();

    });
}
```

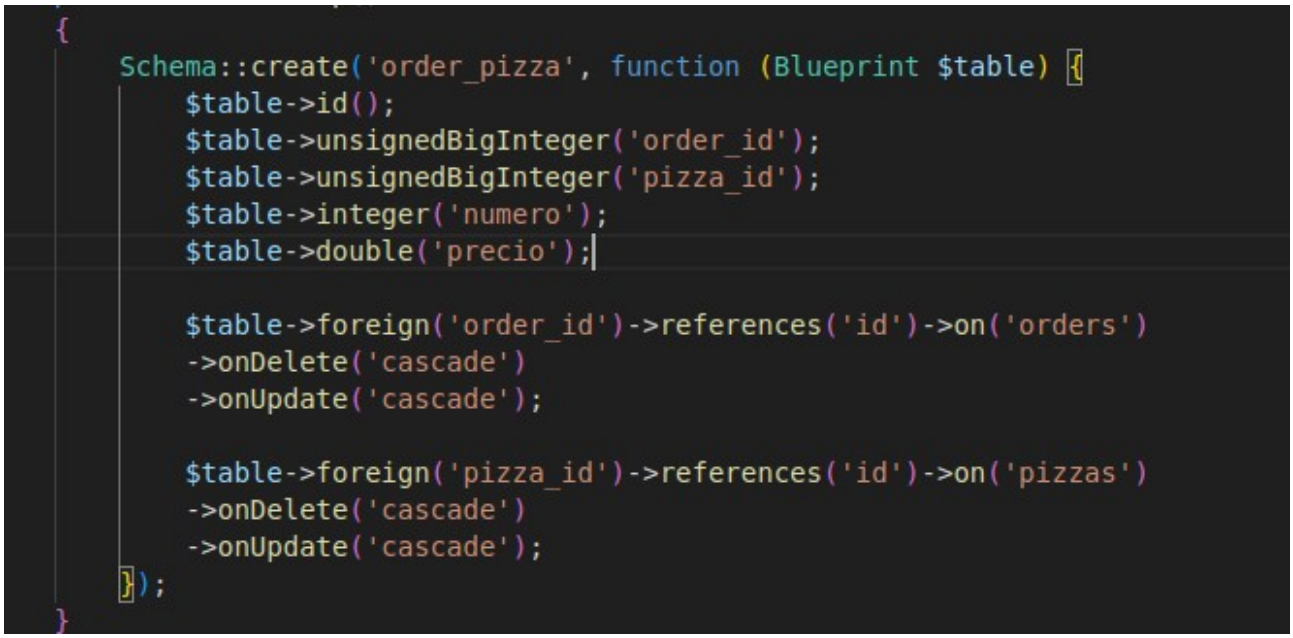
Migracion de orders

Tener en cuenta que esta tabla tiene una FK, asi que hay que declararlo dentro de sus propiedades

```
{
    Schema::create('orders', function (Blueprint $table) {
        $table->id();
        $table->date('fecha');
        $table->string('hora');
        $table->unsignedBigInteger('customer_id');
        $table->string('email');
        $table->timestamps();

        $table->foreign('customer_id')->references('id')->on('customers')
        ->onDelete('cascade')
        ->onUpdate('cascade');
    });
}
```

Migraciones de la tabla pivote

En el ejercicio pedia, eliminar el timestamp

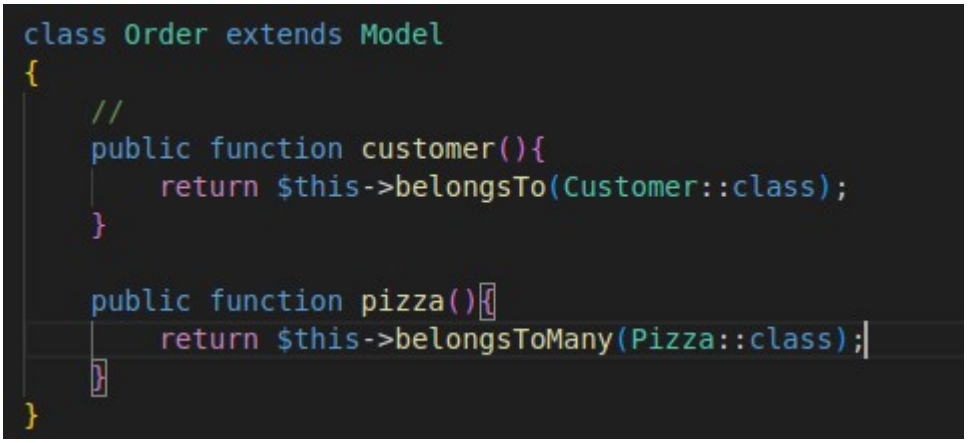


```
{
    Schema::create('order_pizza', function (Blueprint $table) {
        $table->id();
        $table->unsignedBigInteger('order_id');
        $table->unsignedBigInteger('pizza_id');
        $table->integer('numero');
        $table->double('precio');

        $table->foreign('order_id')->references('id')->on('orders')
        ->onDelete('cascade')
        ->onUpdate('cascade');

        $table->foreign('pizza_id')->references('id')->on('pizzas')
        ->onDelete('cascade')
        ->onUpdate('cascade');
    });
}
```

Modelos y Relaciones

Relación	Directa	Inversa
1:1	hasOne - Recupera hijo único	belongsTo - Recupera el padre único
1:n	hasMany - Recupera hijos	belongsTo - Recupera el padre único
n:n	belongsToMany - Recupera relacionados	belongsToMany - Recupera relacionados
Through (*)	hasManyThrough	

Modelo Order
ERRATA EN LA FOTO es customer y pizzas en plural!!!!!
Si es hasOne o belongTo (singular), sino plural
1 orden pertenece(belong to) a 1 usuario
tabla pivote → siempre belongToMany



Modelo Pizza

modelo pizza utiliza softdeletes, hay que importarlo

```php
class Pizza extends Model
{
    //
    use SoftDeletes;

    public function orders(){
        return $this->belongsToMany(Order::class);
    }
}
```

Modelo customer
1 customer tiene muchas orders

```php
class Customer extends Model
{
    //
    public function orders(){
        return $this-> hasMany(Order::class);
    }
}
```

REALIZAR LA MIGRACION!!!

```
alumno@Daniel:~/Escritorio/laravel/Pizzeriass$ php artisan migrate

  WARN The database 'pizzeriass' does not exist on the 'mysql' connection.

┌ Would you like to create it? ──────────────────────────────────────────────┐
  Yes
└──────────────────────────────────────────────────────────────────────────┘
```

Siempre verificar en PHPmyAdmin que se creo todo correctamente

Si existe algun error se puede corregir con
php artisan migrate:rollback, un paso hacia atrás
php artisan migrate:refresh → elimina todo y vuelve a hacer la migraciones

Cargar datos en la BBDD → realizar script en visual para no perder el insert

```sql
insert into customers(nombre,direccion,telefono,email)
values
("daniel", "finlandia 24", "62225588", "daniel@daniel"),
("pepe", "finlandia 24", "62225588", "daniel@daniel"),
("juan", "finlandia 24", "62225588", "daniel@daniel"),
("luis", "finlandia 24", "62225588", "daniel@daniel");
```

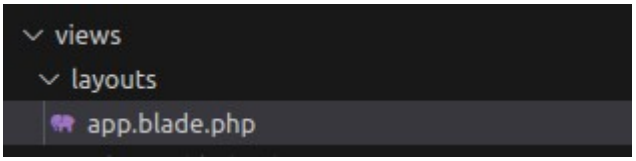
Creacion de Controladores y vistas

controlador customer –resource

```
alumno@Daniel:~/Escritorio/laravel/Pizzeriass$ php artisan make:controller CustomerController --resource

  INFO  Controller [app/Http/Controllers/CustomerController.php] created successfully.
```

Creacion de Vista.

Creamos la carpeta layouts, y la vista app.blade, esta servira de platilla



plantilla ejemplo

```blade
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>@yield('tile')</title>
</head>
<body>
    <header>
        <a href="{{route('home')}}">Home</a>
    </header>
    <h2>@yield('tile')</h2>

    <div class="contenedor">
        @yield('contenido')
    </div>

</body>
</html>
```

Rutas, Controlador , Vista HOME

En web.php definimos las rutas

Definimos la ruta HOME con get y el resto de rutas con resources(hay que importar el controlador)

```php
<?php
use App\Http\Controllers\CustomerController;
use Illuminate\Support\Facades\Route;

Route::get('/',[CustomerController::class,'home'])->name('home');
Route::resource('/customers',CustomerController::class);
```

comprobamos, y estas son con las rutas que vamos a trabajar

```
alumno@Daniel:~/Escritorio/laravel/Pizzeriass$ php artisan route:list

  GET|HEAD        / ..... home
  GET|HEAD        customers ..... customers.index
  POST            customers ..... customers.store
  GET|HEAD        customers/create ..... customers.create
  GET|HEAD        customers/{customer} ..... customers.show
  PUT|PATCH       customers/{customer} ..... customers.update
  DELETE          customers/{customer} ..... customers.destroy
  GET|HEAD        customers/{customer}/edit ..... customers.edit
```

Modificaciones en el controlador y en la vista home

Controlador

```php
//creacion de la funcion home. devulve todos los customer a la view home
public function home(){
    $customers = Customer::all();
    return view('home', compact('customers'));
}
```

Vista de HOME

```
es > views > home.blade.php > ...
@extends('layouts.app')

@section('titulo', 'Usuarios')
@section('contenido')
    @foreach ($customers as $customer)
    <div style="border = 1px solid black">
        <p>Nombre: {{$customer->nombre}}</p>
        <p>Direccion: {{$customer->direccion}}</p>
        <p>Telefono: {{$customer->telefono}}</p>
        <p>Email: {{$customer->email}}</p>
        <div>
            <a href="{{route('customers.edit',$customer->id)}}"><button>Modificar</button></a>

        <form method="post" action="{{route('customers.destroy',$customer->id)}}">
            @csrf
            @method('DELETE')
            <input type="submit" value="BORRAR">
        </form>
    </div>
    </div>

    @endforeach
@endsection
```

Creacion de la vista formulario de modificacion, sus rutas y controlador

controlador metodo edit y update

```php
    */
    public function update(Request $request, string $id)
    {
        $validated = $request->validate([
            'nombre' => 'required|max:255',
            'direccion' => 'required|max:255',
            'telefono' => 'required',
            'email' => 'required|max:255'
        ]);

        $customer = Customer::findOrFail($id);
        $customer->nombre = $request->nombre;
        $customer->direccion = $request->direccion;
        $customer->telefono = $request->telefono;
        $customer->email = $request->email;

        $customer->save();
        return back()->with('mensaje', 'Customer '. $customer->id . ' modificado con éxito');
    }
```

```php
    */
    public function edit(string $id)
    {
        $customer = Customer::findOrFail($id);
        return view('fModificacionCustomer', compact('customer'));

    }
```

Vista

```
es > views > formCustomer.blade.php > ...
@extends('layouts.app')

@section('titulo',"Modificar Usuario")
@section('contenido')
<form action="{{route('customers.update', $customer->id)}}" method="POST">
@csrf
@method('PUT')
    <p>
        <label for="nombre">Nombre</label>
        <input type="text" id="nombre" name="nombre" value="{{old('nombre')?old('nombre'):$customer->nombre}}">
    </p>
    <p>
        <label for="direccion">Direccion</label>
        <input type="text" id="direccion" name="direccion" value="{{old('direccion')?old('direccion'):$customer->direccion}}">
    </p>
    <p>
        <label for="telefono">Telefono</label>
        <input type="text" id="telefono" name="telefono" value="{{old('telefono')?old('telefono'):$customer->telefono}}">
    </p>
    <p>
        <label for="email">Email</label>
        <input type="text" id="email" name="email" value="{{old('email')?old('email'):$customer->email}}">
    </p>

    <input type="submit" value="Modificar">
</form>
```

```
</form>

@if(@session('mensaje'))
    {{session('mensaje')}}
@endif
@error('nombre')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@error('direccion')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@error('telefono')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@error('email')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@endsection
```

Eliminar sin confirmacion

```
    public function destroy(string $id)
    {
        //
        $customer = Customer::findOrFail($id);
        $customer->delete();
        return back()->with('mensaje', 'Customer '. $customer->id . ' Eliminado con éxito');

    }
}
```

Eliminar con confirmacion, hay que modificar algunas cosas:
HOME : el boton de eliminar pasa a ser un enlace

```
    <div>
        <div>
            <a href="{{route('customers.edit',$customer->id)}}"><button>Modificar</button></a>
        </div>
        <div>
            <a href="{{route('confirmar',$customer->id)}}"><button>Eliminar</button></a>
        </div>
        {{-- <div>
            <form method="post" action="{{route('customers.destroy',$customer->id)}}">
```

Creacion de una nueva ruta para que sirva el enlace de eliminacion

la ruta es la de confirmacion

```php
Route::get('/',[CustomerController::class,'home'])->name('home');
Route::get('/confirmar/{id}',[CustomerController::class,'confirmar'])->name('confirmar');
Route::resource('/customers',CustomerController::class);
```

Creacion de una nueva vista para la confirmacion

```
sources > views > confirmacionElimar.blade.php > div > form > input
@extends('layouts.app')

@section('titulo', 'Confirmar Borrado de usuario')
@section('contenido')
<div>
    <p>Nombre: {{$customer->nombre}}</p>
    <p>Direccion: {{$customer->direccion}}</p>
    <p>Telefono: {{$customer->telefono}}</p>
    <p>Email: {{$customer->email}}</p>
</div>

<div>
    <form method="post" action="{{route('customers.destroy',$customer->id)}}">
        @csrf
        @method('DELETE')
        <input type="submit" value="Confirmar!!!">
    </form>
</div>
<div>
    <a href="{{route('home')}}">Volver al home</a>
</div>

@endsection
```

Creacion del metodo confiracion en el Controlador y modificacion de delete para que redireccione al home

```php
     */
    public function destroy(string $id)
    {
        //
        $customer = Customer::findOrFail($id);
        $customer->delete();
        return redirect()->route('home')->with('mensaje', 'Customer '. $customer->id . ' Eliminado con éxito');
    }

    public function confirmar(string $id){
        $customer = Customer::findOrFail($id);
        return view ('confirmacionElimar', compact('customer'));
    }
}
```

ALTA

Inserto un boton de alta en el HOME

```
@section('titulo', 'Usuarios')
@section('contenido')
<div>
    <a href="{{route('customers.create')}}"><button>ALTA!!!</button></a>
</div>
```

Controlador

```
    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        //
            $validated = $request->validate([
            'nombre' => 'required|max:255',
            'direccion' => 'required|max:255',
            'telefono' => 'required',
            'email' => 'required|max:255'
        ]);

        $customer = new Cust   @param \Illuminate\Http\Request $request
        $customer->nombre = $request->nombre;
        $customer->direccion = $request->direccion;
        $customer->telefono = $request->telefono;
        $customer->email = $request->email;

        $customer->save();
        return back()->with('mensaje', 'Customer '. $customer->id . ' modificado con éxito');

    }
```

```
    */
    public function create()
    {
        //
        return view('formAlta');
    }
```

Vista

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

@section('titulo',"Modificar Usuario")
@section('contenido')
<form action="{{route('customers.store')}}" method="POST">
@csrf

    <p>
        <label for="nombre">Nombre</label>
        <input type="text" id="nombre" name="nombre" value="{{old('nombre')?old('nombre'):""}}">
    </p>
    <p>
        <label for="direccion">Direccion</label>
        <input type="text" id="direccion" name="direccion" value="{{old('direccion')?old('direccion'):""}}">
    </p>
    <p>
        <label for="telefono">Telefono</label>
        <input type="text" id="telefono" name="telefono" value="{{old('telefono')?old('telefono'):""}}">
    </p>
    <p>
        <label for="email">Email</label>
        <input type="text" id="email" name="email" value="{{old('email')?old('email'): ""}}">
    </p>

    <input type="submit" value="Modificar">
</form>
```

```blade
</form>

@if(@session('mensaje'))
    {{session('mensaje')}}
@endif
@error('nombre')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@error('direccion')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@error('telefono')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@error('email')
    <div class="error">
        {{ $message }}
    </div>
@enderror
@endsection
```

En caso de querer mostrar los customer en forma de tabla

```
<table style="border-collapse: collapse; width: 100%;">
    <thead>
        <tr>
            <th style="border: 1px solid black; padding: 8px;">Nombre</th>
            <th style="border: 1px solid black; padding: 8px;">Dirección</th>
            <th style="border: 1px solid black; padding: 8px;">Teléfono</th>
            <th style="border: 1px solid black; padding: 8px;">Email</th>
            <th style="border: 1px solid black; padding: 8px;">Modificar</th>
            <th style="border: 1px solid black; padding: 8px;">Eliminar</th>
            <th style="border: 1px solid black; padding: 8px;">Ver Pedidos</th>
        </tr>
    </thead>
    <tbody>
        @foreach ($customers as $customer)
        <tr>
            <td style="border: 1px solid black; padding: 8px;">{{$customer->nombre}}</td>
            <td style="border: 1px solid black; padding: 8px;">{{$customer->direccion}}</td>
            <td style="border: 1px solid black; padding: 8px;">{{$customer->telefono}}</td>
            <td style="border: 1px solid black; padding: 8px;">{{$customer->email}}</td>
            <td style="border: 1px solid black; padding: 8px;"> <a href="{{route('customers.edit', $customer->id)}}"><button>Modificar</bu
            <td style="border: 1px solid black; padding: 8px;">  <a href="{{route('confirmar', $customer->id)}}"><button>Eliminar</button>
            <td style="border: 1px solid black; padding: 8px;">  <a href="{{route('pedidos', $customer->id)}}"><button>Ver PEdidos</buttor
                |

        </tr>
        @endforeach
    </tbody>
</table>
```