**Índice**

[**Planeación de tablas 1**](#_q1slj98joesx)

[**Migraciones 7**](#_oxfl6geu175)

[Rol (se llama rols) 7](#_c0nh1hs10ai0)

[Tipo\_documento 8](#_1f2n91badvja)

[Usuarios (Users ya que conserve la que traia laravel) 9](#_8f5l15df1mm8)

[Tipo\_cita 10](#_ujozw4hnftvv)

[Cita 11](#_9ktgia93bri1)

[Compra 12](#_ldlmsqm3pib2)

[Compra\_producto 13](#_dpz68jdgwbcf)

[**Modelos 14**](#_e33avvcavpv2)

[Rol 14](#_okcbd2kiyrv)

[Tipo\_documento 14](#_gwfabcsqsgpk)

[Usuarios (User ya que conserve el que trae laravel) 15](#_l7c5jutg2f10)

[Tipo\_cita 17](#_d522bneya0zi)

[Cita 17](#_tgv1nmr3irw3)

[Compra 18](#_92my5egmmewm)

[Compra\_producto 19](#_cdvhaz4l556r)

[**Controladores 20**](#_qu7v0y4vkzkj)

[Rol 20](#_1siwt9c74kdh)

[Tipo\_documento 21](#_j39bdyu9875c)

[Usuarios (User ya que conserve el que trae laravel) 23](#_2tkgvnytoxiz)

[Tipo\_cita 25](#_pc1rvx79g12p)

[Cita 27](#_9mo2czbefgw9)

[Compra 28](#_9ec1nkworqwe)

[Compra\_producto 30](#_ctot6okpdiav)

[**Rutas Api.php 32**](#_nopv1lbwvpjd)

[**Rutas web.php 32**](#_f1wmfw8toazk)

## Planeación de tablas

Rol

* Id
* Descripcion (Admin, Vendedor, Cliente)

Tipo\_documento

* Id
* Descripcion (C.C, C.E, T.I, PSTE)

Usuario (Para esta solo sería editar la que ya tenemos de usuarios que pone laravel automatizada)

* Id
* Rol\_id (El id del rol) así a la ahora de inciar sesion es automatizado como dijo la profe entonces sería con una consulta y un if que creo que también se usan en php algo como if id rol 1 ir a perfil cliente y asi, else if id rol 2 ir a vendedor y eso
* Nombre
* Apellidos
* Correo
* Contraseña
* Tipo\_documento
* Documento
* Teléfono
* Ciudad
* Dirección
* (Creo que ya no seria mas tu que opinas???)

Compra (uff no se como hacer este)

* Id\_Factura
* Id\_vendedor
* Id\_usuario
* Fecha
* Valor
* Îva?

Compra (Producto)

* idFactura
* idProductos
* NomProducto
* CantidadProductos
* ValorTotalProd

Tipo\_cita

* Id
* Descripcion (Mantenimiento preventivo, Mantenimiento correctivo, limpieza, Personalización)

Cita

* Id
* Id\_tipo-cita
* Id\_cliente
* Id\_vendedor
* Hora
* Fecha
* Fecha\_finalización
* Finalizada (Si/No)

Comentarios

* Usuario
* Comentario
* Fecha de publicación
* Actualizado? (Si/No)

Perfil cliente

* Foto / nombre de usuario (Perfil\_usuario)
* Datos del usuario (Usuarios)
* Historial de comprar (todo .lo de productos)
* carrito (si)
* Citas (cita, tipo\_cita)
* Comentarios ()comentario - usuario

Perfil vendedor

* Foto / nombre de usuario (Perfil\_usuario)
* Datos del usuario (Usuarios)
* Historial de ventas
* Historial de productos
* Citas

Perfil administrador

* Foto / nombre de usuario (Perfil\_usuario)
* Datos del usuario (Usuarios)
* Permiso de cambiar el rol de vendedor (tabla usuarios)
* Productos más vendidos
* Actividad reciente
* Registro de datos de vendedores
* Control de quejas?

**Vistas**

General

* Barra de navegación
* Bienvenidos

Inicio

* Cookies
* Servicios -> Agendar la cita -> Iniciar sesión
* Galería
* Descuentos
* Categorías
* Opinión de nuestros clientes
* Footer

Colecciones

* Filtros / Menu
* Productos (Descripción de los productos)
* Carrito (Todo (Proceso de pago))
* No añadir a carrito hasta no iniciar sesión

Galería

* Barra de navegación
* Imágenes
* Información de imágenes

Cuidados (Opcional)

* Cuidados

Nosotros

* Servicios
* Historia (linea de tiempo)

Iniciar sesión

* Todo
* Entrar con un rol específico
* Poner el vendedor
* Conectar la base

Registrarse

* Todo
* Roles de usuario
* Conectar la base

Recuperar contraseña

* Arreglar todo el código

Contactanos

* Barra de navegación
* Bienvenidos

Perfiles

* Foto - Nombre Usuario
* Datos del cliente (Documento, Nombre, Ciudad, Dirección, Teléfono )
* Compras
* Citas
* Historial de comentarios (como en google maps)
* Logout

Registro

* Nombre
* Apellidos
* Correo
* Contraseña
* Teléfono

Iniciar sesión

* Seleccionar rol (Admin, vend, client)
* Correo - Usuario
* Contraseña

Contactanos

* Barra de navegación
* Bienvenidos

## Migraciones

### Rol (se llama rols)

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateRolsTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('rols', function (Blueprint $table) {

$table->id();

$table->string('descripcion')->unique(); *// Agregamos la columna 'descripcion'*

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('rols');

}

}

### 

### Tipo\_documento

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateTipoDocumentosTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up()

{

Schema::create('tipo\_documentos', function (Blueprint $table) {

$table->id();

$table->string('descripcion'); *// Agregamos la columna 'descripcion'*

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('tipo\_documentos');

}

}

### 

### Usuarios (Users ya que conserve la que traia laravel)

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration

{

/\*\*

\* Run the migrations.

\*

\*

\* @return void

\*/

public function up()

{

Schema::create('users', function (Blueprint $table) {

$table->id();

$table->foreignId('rol\_id')->constrained('rols'); *// Agregamos el nombre de la tabla referenciada*

$table->string('name');

$table->string('apellidos');

$table->string('email')->unique();

$table->timestamp('email\_verified\_at')->nullable();

$table->string('password');

$table->foreignId('tipo\_documento\_id')->constrained('tipo\_documentos'); *// Agregamos la relación con el tipo de documento*

$table->string('numero\_documento');

$table->string('telefono');

$table->string('ciudad');

$table->string('direccion');

$table->rememberToken();

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down()

{

Schema::dropIfExists('users');

}

}

### Tipo\_cita

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateTipoCitasTable extends Migration

{

public function up()

{

Schema::create('tipo\_citas', function (Blueprint $table) {

$table->id();

$table->string('descripcion');

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('tipo\_citas');

}

}

### Cita

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateCitasTable extends Migration

{

public function up()

{

Schema::create('citas', function (Blueprint $table) {

$table->id();

$table->foreignId('id\_tipo\_cita')->constrained('tipo\_citas');

$table->unsignedBigInteger('id\_cliente');

$table->unsignedBigInteger('id\_vendedor');

$table->string('hora');

$table->date('fecha');

$table->date('fecha\_finalizacion')->nullable();

$table->enum('finalizada', ['Si', 'No']);

$table->timestamps();

*// Definir las claves foráneas*

$table->foreign('id\_tipo\_cita')->references('id')->on('tipo\_citas');

$table->foreign('id\_cliente')->references('id')->on('users')->where('rol\_id', 1); *// Asume que el rol\_id 1 es para clientes*

$table->foreign('id\_vendedor')->references('id')->on('users')->where('rol\_id', 2); *// Asume que el rol\_id 2 es para vendedores*

});

}

public function down()

{

Schema::dropIfExists('citas');

}

}

### Compra

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateComprasTable extends Migration

{

public function up()

{

Schema::create('compras', function (Blueprint $table) {

$table->id();

$table->unsignedBigInteger('id\_factura');

$table->unsignedBigInteger('id\_vendedor');

$table->unsignedBigInteger('id\_usuario');

$table->date('fecha');

$table->decimal('valor', 10, 2);

$table->timestamps();

*// Definir las claves foráneas*

$table->foreign('id\_vendedor')->references('id')->on('users')->where('rol\_id', 2); *// Asumo que el rol\_id 2 es para vendedores*

$table->foreign('id\_usuario')->references('id')->on('users');

});

}

public function down()

{

Schema::dropIfExists('compras');

}

}

### Compra\_producto

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateCompraProductosTable extends Migration

{

public function up()

{

Schema::create('compra\_productos', function (Blueprint $table) {

$table->id();

$table->unsignedBigInteger('id\_factura');

$table->unsignedBigInteger('id\_producto');

$table->string('nom\_producto');

$table->integer('cantidad\_productos');

$table->decimal('valor\_total\_prod', 10, 2);

$table->timestamps();

*// Definir las claves foráneas*

$table->foreign('id\_factura')->references('id')->on('compras');

$table->foreign('id\_producto')->references('id')->on('productos'); *// Ajusta el nombre de la tabla de productos según tu modelo*

});

}

public function down()

{

Schema::dropIfExists('compra\_productos');

}

}

## Modelos

### Rol

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Rol extends Model

{

use HasFactory;

protected $fillable = ['descripcion']; *// Agregamos 'descripcion' a los campos llenables*

public function user()

{

return $this->hasMany(User::class); *// Cambiar 'Usuario' a 'User'*

}

}

### Tipo\_documento

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class TipoDocumento extends Model

{

use HasFactory;

protected $fillable = ['descripcion']; *// Agregamos 'descripcion' a los campos llenables*

public function user()

{

return $this->hasMany(User::class); *// Cambiar 'Usuario' a 'User'*

}

}

### Usuarios (User ya que conserve el que trae laravel)

<?php

namespace App\Models;

use App\Models\Rol;

use App\Models\TipoDocumento;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Foundation\Auth\User as Authenticatable;

use Illuminate\Notifications\Notifiable;

use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable

{

use HasApiTokens, HasFactory, Notifiable;

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array<int, string>

\*/

protected $fillable = [

'rol\_id',

'name',

'apellidos',

'email',

'password',

'tipo\_documento\_id',

'numero\_documento',

'telefono',

'ciudad',

'direccion',

];

/\*\*

\* The attributes that should be hidden for serialization.

\*

\* @var array<int, string>

\*/

protected $hidden = [

'password',

'remember\_token',

];

/\*\*

\* The attributes that should be cast.

\*

\* @var array<string, string>

\*/

protected $casts = [

'email\_verified\_at' => 'datetime',

];

*// Relación con Rol (uno a uno)*

public function rol()

{

return $this->belongsTo(Rol::class);

}

*// Relación con TipoDocumento (uno a uno)*

public function tipoDocumento()

{

return $this->belongsTo(TipoDocumento::class);

}

}

### 

### Tipo\_cita

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class TipoCita extends Model

{

use HasFactory;

protected $fillable = [

'descripcion',

];

*// Relación con Cita (Uno a muchos)*

public function citas()

{

return *$this*->hasMany(Cita::class, 'id\_tipo\_cita');

}

}

### Cita

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Cita extends Model

{

use HasFactory;

protected $fillable = [

'id\_tipo\_cita',

'id\_cliente',

'id\_vendedor',

'hora',

'fecha',

'fecha\_finalizacion',

'finalizada',

];

*// Relación con TipoCita (Uno a uno)*

public function tipoCita()

{

return *$this*->belongsTo(TipoCita::class, 'id\_tipo\_cita');

}

*// Relación con User (Cliente)*

public function cliente()

{

return *$this*->belongsTo(User::class, 'id\_cliente');

}

*// Relación con User (Vendedor)*

public function vendedor()

{

return *$this*->belongsTo(User::class, 'id\_vendedor');

}

}

### Compra

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Compra extends Model

{

use HasFactory;

protected $fillable = [

'id\_factura',

'id\_vendedor',

'id\_usuario',

'fecha',

'valor',

];

*// Relación con User (Vendedor)*

public function vendedor()

{

return *$this*->belongsTo(User::class, 'id\_vendedor');

}

*// Relación con User (Usuario)*

public function usuario()

{

return *$this*->belongsTo(User::class, 'id\_usuario');

}

*// Relación con CompraProducto (Uno a muchos)*

public function productos()

{

return *$this*->hasMany(CompraProducto::class, 'id\_factura');

}

}

### Compra\_producto

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class CompraProducto extends Model

{

use HasFactory;

protected $fillable = [

'id\_factura',

'id\_producto',

'nom\_producto',

'cantidad\_productos',

'valor\_total\_prod',

];

*// Relación con Compra*

public function compra()

{

return *$this*->belongsTo(Compra::class, 'id\_factura');

}

*// Relación con Producto*

public function producto()

{

return *$this*->belongsTo(Producto::class, 'id\_producto');

}

}

## Controladores

### Rol

<?php

namespace App\Http\Controllers;

use App\Models\Rol;

use Illuminate\Http\Request;

class RolController extends Controller

{

public function index()

{

$roles = Rol::all();

return response()->json($roles);

}

public function show($id)

{

$rol = Rol::findOrFail($id);

return response()->json($rol);

}

public function store(Request $request)

{

$request->validate([

'descripcion' => 'required|string|unique:rols',

]);

$rol = Rol::create([

'descripcion' => $request->input('descripcion'),

]);

return response()->json($rol, 201);

}

public function update(Request $request, $id)

{

$request->validate([

'descripcion' => 'required|string|unique:rols,descripcion,' . $id,

]);

$rol = Rol::findOrFail($id);

$rol->update([

'descripcion' => $request->input('descripcion'),

]);

return response()->json($rol);

}

public function destroy($id)

{

$rol = Rol::findOrFail($id);

$rol->delete();

return response()->json(null, 204);

}

}

### Tipo\_documento

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class TipoDocumentoController extends Controller

{

public function index()

{

$tiposDocumento = TipoDocumento::all();

return response()->json($tiposDocumento);

}

public function show($id)

{

$tipoDocumento = TipoDocumento::findOrFail($id);

return response()->json($tipoDocumento);

}

public function store(Request $request)

{

$request->validate([

'descripcion' => 'required|string|unique:tipo\_documentos',

]);

$tipoDocumento = TipoDocumento::create([

'descripcion' => $request->input('descripcion'),

]);

return response()->json($tipoDocumento, 201);

}

public function update(Request $request, $id)

{

$request->validate([

'descripcion' => 'required|string|unique:tipo\_documentos,descripcion,' . $id,

]);

$tipoDocumento = TipoDocumento::findOrFail($id);

$tipoDocumento->update([

'descripcion' => $request->input('descripcion'),

]);

return response()->json($tipoDocumento);

}

public function destroy($id)

{

$tipoDocumento = TipoDocumento::findOrFail($id);

$tipoDocumento->delete();

return response()->json(null, 204);

}

}

### Usuarios (User ya que conserve el que trae laravel)

<?php

namespace App\Http\Controllers;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Hash;

class UserController extends Controller

{

public function index()

{

$users = User::all();

return response()->json($users);

}

public function show($id)

{

$user = User::findOrFail($id);

return response()->json($user);

}

public function store(Request $request)

{

$request->validate([

'name' => 'required|string',

'email' => 'required|email|unique:users',

'password' => 'required|string|min:8',

'rol\_id' => 'required|exists:rols,id',

'tipo\_documento\_id' => 'required|exists:tipo\_documentos,id',

'numero\_documento' => 'required|string|unique:users',

'telefono' => 'required|string',

'ciudad' => 'required|string',

'direccion' => 'required|string',

]);

$user = User::create([

'name' => $request->input('name'),

'email' => $request->input('email'),

'password' => Hash::make($request->input('password')),

'rol\_id' => $request->input('rol\_id'),

'tipo\_documento\_id' => $request->input('tipo\_documento\_id'),

'numero\_documento' => $request->input('numero\_documento'),

'telefono' => $request->input('telefono'),

'ciudad' => $request->input('ciudad'),

'direccion' => $request->input('direccion'),

]);

return response()->json($user, 201);

}

public function update(Request $request, $id)

{

$request->validate([

'name' => 'required|string',

'email' => 'required|email|unique:users,email,' . $id,

'password' => 'nullable|string|min:8',

'rol\_id' => 'required|exists:rols,id',

'tipo\_documento\_id' => 'required|exists:tipo\_documentos,id',

'numero\_documento' => 'required|string|unique:users,numero\_documento,' . $id,

'telefono' => 'required|string',

'ciudad' => 'required|string',

'direccion' => 'required|string',

]);

$userData = [

'name' => $request->input('name'),

'email' => $request->input('email'),

'rol\_id' => $request->input('rol\_id'),

'tipo\_documento\_id' => $request->input('tipo\_documento\_id'),

'numero\_documento' => $request->input('numero\_documento'),

'telefono' => $request->input('telefono'),

'ciudad' => $request->input('ciudad'),

'direccion' => $request->input('direccion'),

];

if ($request->has('password')) {

$userData['password'] = Hash::make($request->input('password'));

}

$user = User::findOrFail($id);

$user->update($userData);

return response()->json($user);

}

public function destroy($id)

{

$user = User::findOrFail($id);

$user->delete();

return response()->json(null, 204);

}

}

### Tipo\_cita

<?php

namespace App\Http\Controllers;

use App\Models\TipoCita;

use Illuminate\Http\Request;

class TipoCitaController extends Controller

{

public function index()

{

$tipoCitas = TipoCita::all();

return response()->json($tipoCitas);

}

public function show($id)

{

$tipoCita = TipoCita::findOrFail($id);

return response()->json($tipoCita);

}

public function store(Request $request)

{

$request->validate([

'descripcion' => 'required|string',

]);

$tipoCita = TipoCita::create([

'descripcion' => $request->input('descripcion'),

]);

return response()->json($tipoCita, 201);

}

public function update(Request $request, $id)

{

$request->validate([

'descripcion' => 'required|string',

]);

$tipoCita = TipoCita::findOrFail($id);

$tipoCita->update([

'descripcion' => $request->input('descripcion'),

]);

return response()->json($tipoCita);

}

public function destroy($id)

{

$tipoCita = TipoCita::findOrFail($id);

$tipoCita->delete();

return response()->json(null, 204);

}

}

### Cita

<?php

namespace App\Http\Controllers;

use App\Models\Cita;

use Illuminate\Http\Request;

class CitaController extends Controller

{

public function index()

{

$citas = Cita::all();

return response()->json($citas);

}

public function show($id)

{

$cita = Cita::findOrFail($id);

return response()->json($cita);

}

public function store(Request $request)

{

$request->validate([

'id\_tipo\_cita' => 'required|exists:tipo\_citas,id',

'id\_cliente' => 'required|exists:users,id',

'id\_vendedor' => 'required|exists:users,id',

'hora' => 'required|string',

'fecha' => 'required|date',

'fecha\_finalizacion' => 'nullable|date',

'finalizada' => 'required|in:Si,No',

]);

$cita = Cita::create($request->all());

return response()->json($cita, 201);

}

public function update(Request $request, $id)

{

$request->validate([

'id\_tipo\_cita' => 'required|exists:tipo\_citas,id',

'id\_cliente' => 'required|exists:users,id',

'id\_vendedor' => 'required|exists:users,id',

'hora' => 'required|string',

'fecha' => 'required|date',

'fecha\_finalizacion' => 'nullable|date',

'finalizada' => 'required|in:Si,No',

]);

$cita = Cita::findOrFail($id);

$cita->update($request->all());

return response()->json($cita);

}

public function destroy($id)

{

$cita = Cita::findOrFail($id);

$cita->delete();

return response()->json(null, 204);

}

}

### Compra

<?php

namespace App\Http\Controllers;

use App\Models\Compra;

use Illuminate\Http\Request;

class CompraController extends Controller

{

public function index()

{

$compras = Compra::all();

return response()->json($compras);

}

public function show($id)

{

$compra = Compra::findOrFail($id);

return response()->json($compra);

}

public function store(Request $request)

{

$request->validate([

'id\_factura' => 'required|unique:compras,id\_factura',

'id\_vendedor' => 'required|exists:users,id',

'id\_usuario' => 'required|exists:users,id',

'fecha' => 'required|date',

'valor' => 'required|numeric',

]);

$compra = Compra::create($request->all());

return response()->json($compra, 201);

}

public function update(Request $request, $id)

{

$request->validate([

'id\_factura' => 'required|unique:compras,id\_factura,' . $id,

'id\_vendedor' => 'required|exists:users,id',

'id\_usuario' => 'required|exists:users,id',

'fecha' => 'required|date',

'valor' => 'required|numeric',

]);

$compra = Compra::findOrFail($id);

$compra->update($request->all());

return response()->json($compra);

}

public function destroy($id)

{

$compra = Compra::findOrFail($id);

$compra->delete();

return response()->json(null, 204);

}

}

### Compra\_producto

<?php

namespace App\Http\Controllers;

use App\Models\CompraProducto;

use Illuminate\Http\Request;

class CompraProductoController extends Controller

{

public function index()

{

$compraProductos = CompraProducto::all();

return response()->json($compraProductos);

}

public function show($id)

{

$compraProducto = CompraProducto::findOrFail($id);

return response()->json($compraProducto);

}

public function store(Request $request)

{

$request->validate([

'id\_factura' => 'required|exists:compras,id',

'id\_producto' => 'required|exists:productos,id',

'nom\_producto' => 'required|string',

'cantidad\_productos' => 'required|integer',

'valor\_total\_prod' => 'required|numeric',

]);

$compraProducto = CompraProducto::create($request->all());

return response()->json($compraProducto, 201);

}

public function update(Request $request, $id)

{

$request->validate([

'id\_factura' => 'required|exists:compras,id',

'id\_producto' => 'required|exists:productos,id',

'nom\_producto' => 'required|string',

'cantidad\_productos' => 'required|integer',

'valor\_total\_prod' => 'required|numeric',

]);

$compraProducto = CompraProducto::findOrFail($id);

$compraProducto->update($request->all());

return response()->json($compraProducto);

}

public function destroy($id)

{

$compraProducto = CompraProducto::findOrFail($id);

$compraProducto->delete();

return response()->json(null, 204);

}

}

## Rutas Api.php

use App\Http\Controllers\RolController;

use App\Http\Controllers\TipoDocumentoController;

use App\Http\Controllers\UserController;

Route::resource('/rols', RolController::class);

Route::resource('/tipo\_documentos', TipoDocumentoController::class);

Route::resource('/users', UserController::class);

<?php

use App\Http\Controllers\RolController;

use App\Http\Controllers\TipoDocumentoController;

use App\Http\Controllers\UserController;

use App\Http\Controllers\TipoCitaController;

use App\Http\Controllers\CitaController;

use App\Http\Controllers\CompraController;

use App\Http\Controllers\CompraProductoController;

use Illuminate\Support\Facades\Route;

Route::resource('rols', RolController::class);

Route::resource('tipo\_documentos', TipoDocumentoController::class);

Route::resource('users', UserController::class);

Route::resource('tipo\_citas', TipoCitaController::class);

Route::resource('citas', CitaController::class);

Route::resource('compras', CompraController::class);

Route::resource('compra\_productos', CompraProductoController::class);

## Rutas web.php

use App\Http\Controllers\RolController;

use App\Http\Controllers\TipoDocumentoController;

use App\Http\Controllers\UserController;

*// Rutas para el controlador de roles*

Route::resource('rols', RolController::class);

*// Rutas para el controlador de tipos de documentos*

Route::resource('tipo\_documentos', TipoDocumentoController::class);

*// Rutas para el controlador de usuarios*

Route::resource('users', UserController::class);

<?php

use App\Http\Controllers\RolController;

use App\Http\Controllers\TipoDocumentoController;

use App\Http\Controllers\UserController;

use App\Http\Controllers\TipoCitaController;

use App\Http\Controllers\CitaController;

use App\Http\Controllers\CompraController;

use App\Http\Controllers\CompraProductoController;

use Illuminate\Support\Facades\Route;

*// Rutas para el controlador de roles*

Route::resource('rols', RolController::class);

*// Rutas para el controlador de tipos de documentos*

Route::resource('tipo\_documentos', TipoDocumentoController::class);

*// Rutas para el controlador de usuarios*

Route::resource('users', UserController::class);

*// Rutas para el controlador de tipos de citas*

Route::resource('tipo\_citas', TipoCitaController::class);

*// Rutas para el controlador de citas*

Route::resource('citas', CitaController::class);

*// Rutas para el controlador de compras*

Route::resource('compras', CompraController::class);

*// Rutas para el controlador de productos comprados en una compra*

Route::resource('compra\_productos', CompraProductoController::class);