



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

FACULTAD DE INGENIERÍA

ÁREA DE CIENCIAS DE LA COMPUTACIÓN

NOMBRE DEL ESTUDIANTE:

RICO MARTÍNEZ EMANUEL

MATERIA: APLICACIONES WEB INTERACTIVAS

MANUAL DE PROGRAMADOR

Controladores

PrincipalController

```
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Ventas;
use App\Models/Productos;
use Barryvdh\DomPDF\Facade\PDF;
use Illuminate\Support\Facades\DB;

class PrincipalController extends Controller
{
```

Las bibliotecas que se importan son los modelos de las ventas y los productos, así como dompdf y DB que es para realizar queries.

```
public function agregar()
{
    $ventas = Ventas::all();
    return view('/ventas/agregar_venta')->with('ventas', $ventas);
}

public function ver()
{
    $ventas = Ventas::all();
    $contador = count(Ventas::all());
    return view('/ventas/ver_ventas')->with('ventas', $ventas)->with('contador', $contador);
}

public function imprimir($id)
{
    $venta = Ventas::find($id);
    $productos = Productos::all()->where('id_venta', '=', $id);

    $pdf = PDF::loadView('pdf.pdf_venta', compact('venta'), compact('productos'));
    $pdf->setPaper('Letter');
    return $pdf->stream();
}
```

```

public function imprimirglobal()
{
    $result = DB::select(DB::raw("SELECT DATE_FORMAT(fecha, '%d/%m/%y') AS fechas, SUM(preciot) AS
    ventasdiarias FROM ventas GROUP BY fecha"));

    $pdf = PDF::loadView('pdf.pdf_venta_global', compact('result'));
    $pdf->setPaper('Letter');
    return $pdf->stream();
}

public function grafica()
{
    $result = DB::select(DB::raw("SELECT DATE_FORMAT(fecha, '%d/%m') AS fechas, SUM(preciot) AS
    ventasdiarias FROM ventas GROUP BY fecha"));
    $data = "";
    $contador = count(Ventas::all());

    foreach ($result as $val) {
        $data .= "[" . $val->fechas . ", " . $val->ventasdiarias . "],";
    }

    return view('/ventas/reporte_ventas', compact('data'))->with('contador', $contador);
}

public function nosotros()
{
    return view('/nosotros');
}

```

```

public function nosotros()
{
    return view('/nosotros');
}

public function productos()
{
    return view('/productos');
}

public function eliminar($id)
{
    $venta = Ventas::find($id);
    $venta->delete();
    return redirect()->back();
}

public function eliminarventas()
{
    DB::table('ventas')->delete();
    return redirect()->back();
}

public function muestraeditar($id)
{
    $venta = Ventas::find($id);
    $productos = Productos::all()->where('id_venta', '=', $id);

    return view('/ventas/editar_venta')->with('venta', $venta)->with('productos', $productos);
}

```

```

public function guardar(Request $request)
{
    $venta = new Ventas();
    $venta->nombre_cliente = $request->nombre_cliente;
    $venta->celular_cliente = $request->celular_cliente;
    $venta->fecha = now()->format('y/m/d');
    $venta->save();

    return redirect()->action('App\Http\Controllers\PrincipalController@muestraeditar', ['id' => $venta->id]);
}

```

```

public function guardaredicion(Request $request)
{
    $venta = Ventas::find($request->id);

    $venta->nombre_cliente = $request->nombre_cliente;
    $venta->celular_cliente = $request->celular_cliente;
    $venta->adelanto = floatval($request->adelanto);
    $venta->preciot = floatval($request->preciot);
    $venta->save();

    return redirect('/ventas/ver_ventas');
}

```

El controlador de la pantalla principal cuenta con 11 funciones las cuales sirven para agregar ventas, editar ventas, eliminar ventas, mostrar ventas, imprimir pdf por venta, imprimir pdf de ventas globales y definir algunas rutas como productos y nosotros.

ProductoController

```
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Models\Productos;
```

En este controlador solo importamos el modelo del producto.

```
class ProductoController extends Controller
{

    public function eliminar($id)
    {
        $producto = Productos::find($id);
        $producto->delete();
        return redirect()->back();
    }

    public function muestraeditar($id)
    {
        $producto = Productos::find($id);

        return view('editar')->with('producto', $producto);
    }

    public function guardar(Request $request)
    {
        $producto = new Productos();
        $producto->cantidad = $request->cantidad;
        $producto->producto = $request->producto;
        $producto->descripcion = $request->descripcion;
        $producto->preciou = floatval($request->preciou);
        $producto->importe = floatval($request->importe);
        $producto->id_venta = $request->id_venta;
        $producto->save();

        return redirect()->back();
    }
}
```

```
public function guardaredicion_producto(Request $request)
{
    $producto = Productos::find($request->id);

    $producto->cantidad = $request->cantidad;
    $producto->producto = $request->producto;
    $producto->descripcion = $request->descripcion;
    $producto->preciou = floatval($request->preciou);
    $producto->importe = floatval($request->importe);

    $producto->save();

    return redirect()->back();
}
```

Y de igual manera las funciones definidas están hechas para crear, editar, leer y eliminar productos.

Migraciones

Ventas

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class Ventas extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('ventas', function (Blueprint $table) {
            $table->engine = 'InnoDB';
            $table->id();
            $table->string('nombre_cliente', 30)->nullable();
            $table->bigInteger('celular_cliente')->nullable();
            $table->float('adelanto')->nullable();
            $table->float('preciot')->nullable();
            $table->date('fecha')->nullable();
            $table->timestamps();
        });
    }
}

```

En la migración de ventas tenemos todos los campos que llevará la tabla, así como sus tipos.

Productos


```

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class Productos extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('productos', function (Blueprint $table) {
            $table->engine = 'InnoDB';
            $table->id();
            $table->unsignedBigInteger('id_venta');
            $table->integer('cantidad');
            $table->string('producto', 50);
            $table->string('descripcion', 100);
            $table->float('preciou');
            $table->float('importe');
            $table->timestamps();

            $table->foreign('id_venta')->references('id')->on('ventas')->onDelete('cascade');
        });
    }
}

```

En la migración de productos tenemos todos los campos que llevará la tabla, así como sus tipos.

```

return new class extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->foreignId('current_team_id')->nullable();
            $table->string('profile_photo_path', 2048)->nullable();
            $table->integer('type')->nullable();
            $table->timestamps();
        });
    }
}

```

En la migración de usuarios tenemos todos los campos que llevará la tabla, así como sus tipos.

Modelos

```
<?php

namespace App\Models;

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

class Productos extends Model
{
    use HasFactory;

    protected $table = 'productos';
    protected $fillable=[
        'cantidad',
        'producto',
        'descripcion',
        'preciou' => 'decimal:2',
        'importe' => 'decimal:2'
    ];
}
```

```
use HasApiTokens;
use HasFactory;
use HasProfilePhoto;
use Notifiable;
use TwoFactorAuthenticatable;

/**
 * The attributes that are mass assignable.
 *
 * @var string[]
 */
protected $fillable = [
    'name',
    'email',
    'password',
];

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

/**
```

```
<?php
```

```
namespace App\Models;
```

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

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

```
class Ventas extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table = 'ventas';
```

```
    protected $fillable=[
```

```
        'nombre_cliente',
```

```
        'celular_cliente',
```

```
        'adelanto' => 'decimal:2',
```

```
        'preciot' => 'decimal:2'
```

```
    ];
```

```
}
```