

Listar producto

The screenshot shows a web browser at the URL `localhost/producto/public/productos#`. The page has a navigation bar with links: **Tienda**, **Listar productos**, **Ingresar Producto**, and **Editar producto**. The main heading is **Listado de productos**. Below the heading, there is a green button labeled **Agregar Producto** and a search section with a text input labeled **Ingrese coincidencia** and a green **Buscar** button. A table displays the product list with columns: **#**, **Foto**, **Codigo**, **Descripcion**, **Marca**, and **Acciones**. The table contains five rows of product data. Each row has a yellow **Editar** button and a red **Eliminar** button. At the bottom of the table, there is a pagination bar with links for **1**, **2**, and **3**.

#	Foto	Codigo	Descripcion	Marca	Acciones
1		10001	Alicate punta larga	Truper	Editar Eliminar
2		123123	Alicate punta normal 6"	Truper	Editar Eliminar
3		132456	Clavo 3"	N/A	Editar Eliminar
4		1231234	Clavo 2"	N/A	Editar Eliminar
5		123123	Clavo 1"	N/A	Editar Eliminar

- 1 – Barra de navegación, 3 opciones, listar producto, ingresar y editar, esta última bloqueada ya que para editar un producto debe darle en el botón editar de cada ítem.
- 2 – Campo para realizar filtro por medio de nombre de producto. Para listar todos no ingresar ningún texto.
- 3 - Acciones, eliminar y editar por cada articulo
- 4- Barra de navegación
- 5 – Ir a formulario ingreso de datos
- 6 – Tabla con información de productos

Ingresar producto

1

Tienda Listar productos Ingresar Producto Editar producto

Crear un nuevo producto

Codigo

Descripcion

Marca

2

Foto

Examinar... No se ha seleccionado ningún archivo.

3

Agregar Producto Regresar :

1 – Barra de navegación, 3 opciones, listar producto, ingresar y editar, esta última bloqueada ya que para editar un producto debe darle en el botón editar de cada ítem.

2 – Campos a registrar

3 – Acciones, registrar y regresar

Editar producto

1 Tienda Listar productos Ingresar Producto Editar producto

Editar producto

Codigo

Descripcion

Marca

Foto

Examinar... No se ha seleccionado ningún archivo.

Modificar Producto Regresar

2

3

1 – Barra de navegación, 3 opciones, listar producto, ingresar y editar, esta última bloqueada ya que para editar un producto debe darle en el botón editar de cada ítem.

2 – Campos a registrar

3 – Acciones, registrar y regresar

Base de datos

phpMyAdmin

Reciente Favoritas

Nueva

bd_tienda

Nueva

failed_jobs

migrations

password_resets

productos

users

information_schema

mysql

performance_schema

phpmyadmin

test

_sistema

localhost/phpmyadmin/sql.php?db=bd_tienda&table=productos&pos=0

110%

Correo U Importante Cursos Mantenimiento Diseño Grafico </tools> Google Youtube GSoft-Machaquila Office 365 Curso_servidor Arquí 2 http://127.0.0.1:52546/... https://github.com/a...

Examinar Estructura SQL Buscar Insertar Exportar Importar Privilegios Operaciones Seguimiento Disparadores

Mostrando filas 0 - 11 (total de 12, La consulta tardó 0,0417 segundos.)

SELECT * FROM `productos`

☐ Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Sort by key: Ninguna

+ Opciones

				id	Codigo	Descripcion	Marca	Foto	created_at	updated_at
<input type="checkbox"/>	Editar	Copiar	Borrar	5	10001	Alicate punta larga	Truper	uploads/5hDvJxTiWiYjszy7PoCFpMxwqIWwNcZmVPV0q8m.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	6	123123	Alicate punta normal 6"	Truper	uploads/mvbwDPe8Wu8SAm7NGhB46dqRcJfmE5hEOMo5Q8tm.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	7	132456	Clavo 3"	N/A	uploads/d4DMuRNdWHka2dp7OPRHe0vdOiZ78vA7YH03W8j.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	8	1231234	Clavo 2"	N/A	uploads/srlqxOaWnjyqBUV7cKygg0NZ7KuoAXI3KjGZMKb8.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	9	123123	Clavo 1"	N/A	uploads/XIEQZqjgr7IBgzZEgK6d8Y47TW0VIYQOI6TapfoD.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	10	1231235	Clavo 4"	N/A	uploads/QqLyFpcywVDmy68CRHkflc31PCiuAiPfaaOIV51H.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	11	4545	Alambre de amarre	N/A	uploads/szxiwVUIAuNuaLi9hQZHMCEyNKv06mlTAwrzshG.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	12	45643	Alambre de amarre inoxidable	N/A	uploads/UnasVZcFK1W72pTq6hb7vUEaTYvC7PauYVVJA2Cl.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	13	9696	Pesa de 40kg	Detecto	uploads/gtbjBjGm47pkRppF5JdB6Ik2mpkMmtPVpHTamhy.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	14	9696	Pesa 50kg	Detecto	uploads/NMGZI2WTH7uDdH34JlbthpTp5kmVJ5X6Xgv11Kp4.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	15	694321	Pesa de 30kg	Detecto	uploads/C6GVPr8Ltu5uAIndidobinYUUPhyMaGIRdMOynn.j...	NULL	NULL
<input type="checkbox"/>	Editar	Copiar	Borrar	16	6546546	Rueda para llanta 4"	Truper	uploads/f5xb7gSJAcXvN27Ai8Yk3117xJLvcw67bJEj97g7.j...	NULL	NULL

☐ Seleccionar todo | Para los elementos que están marcados: Editar Copiar Borrar Exportar

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Sort by key: Ninguna

Código Importante

```
<?php

namespace App\Http\Controllers;

use App\Producto;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;

class ProductoController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        $nombre = $request ->get('buscar') ;
        //dd($nombre);
        $productos['productos']=Producto::where('Descripcion','like',"%$nombre%")->paginate(5);
        // $productos['productos']=Producto::paginate(4);

        return view('productos.index',$productos);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('productos.create');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
```

```
{

    $campos=[
        'Codigo'=> 'required|string|max:100',
        'Descripcion'=> 'required|string|max:100',
        'Marca'=> 'required|string|max:100',
        'Foto'=> 'required|max:1000|mimes:jpeg,png,jpg',

    ];

    $mensaje=[
        "required"=>'El campo :attribute es requerido'
    ];

    $this->validate($request,$campos,$mensaje);

    $datosProducto = request()->except('_token');
    //return response()->json($datosProducto) ;

    if($request->hasfile('Foto')){
        $datosProducto['Foto']=$request->file('Foto')->store('uploads','public');
    }

    Producto::insert($datosProducto);

    return redirect('productos')->with('Mensaje','Producto agregado con éxito');
}

/**
 * Display the specified resource.
 *
 * @param \App\Producto $producto
 * @return \Illuminate\Http\Response
 */
public function show(Producto $producto)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param \App\Producto $producto
 * @return \Illuminate\Http\Response
 */
public function edit($id)
```

```
{
    $producto = Producto::findOrFail($id);
    return view('productos.edit',compact('producto')); //producto es la misma variable de
ar arriba
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Producto $producto
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $campos=[
        'Codigo'=> 'required|string|max:100',
        'Descripcion'=> 'required|string|max:100',
        'Marca'=> 'required|string|max:100',

    ];

    if($request->hasfile('Foto')){
        $campos+=[ 'Foto'=> 'required|max:1000|mimes:jpeg,png,jpg' ];
    }

    $mensaje=[
        "required"=>'El campo :attribute es requerido'
    ];

    $this->validate($request,$campos,$mensaje);
    $datosProducto = request()->except(['_token','_method']);

    if($request->hasfile('Foto')){
        $producto = Producto::findOrFail($id);
        Storage::delete('public/'.$producto->Foto);
        $datosProducto['Foto']=$request->file('Foto')->store('uploads','public');
    }
}
```

```
        Producto::where('id','=',$id)->update($datosProducto);

        $producto = Producto::findOrFail($id);
        //return view('productos.edit',compact('producto'));
        return redirect('productos')->with('Mensaje','Producto modificado con éxito');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param \App\Producto $producto
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $producto = Producto::findOrFail($id);

        if(Storage::delete('public/'.$producto->Foto)){
            Producto::destroy($id);
        }

        return redirect('productos')->with('Mensaje','Producto Eliminado con éxito');
    }
}
```

Este es el código más importante (Controlador), ya que este es el controlador de producto, aquí se agrega, se edita y elimina. Lo demás es solo vista CSS + HTML.

```
<?php

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

class CreateProductosTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
}
```



```
*/  
public function up()  
{  
    Schema::create('productos', function (Blueprint $table) {  
        $table->increments('id');  
        $table->string('Codigo');  
        $table->string('Descripcion');  
        $table->string('Marca');  
        $table->string('Foto');  
        $table->timestamps();  
    });  
}  
  
/**  
 * Reverse the migrations.  
 *  
 * @return void  
 */  
public function down()  
{  
    Schema::dropIfExists('productos');  
}  
}
```

Configuración de base de datos

Consola

1. **Crear proyecto**
Composer create-project --prefer-dist laravel/laravel product
2. **Migraciones**
Php artisan migrate
3. **Rollback para migraciones**
artisan migrate:rollback
4. **Crear el modelo product**
php artisan make:model Producto -mcr
5. **Ver lista de rutas disponibles**
php artisan route:list
6. **Habilitar enlaces para fotografías almacenadas**
php artisan storage:link
7. **Habilitar bootstrap 4**
composer require laravel/ui
php artisan ui bootstrap
npm install
npm run dev