

Comenzamos creando el proyecto en el directorio htdocs mediante el comando "composer create-project laravel/laravel proyecto_laravel '10.*'".

A continuacion hubo que cambiar un par de cosas, el nombre de la base de datos (archivo .env) a la que accede el proyecto y una dependencia(archivo composer.json).

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=proyecto_laravel
DB_USERNAME=root
DB_PASSWORD=
```



```
"require-dev": {
    "fakerphp/faker": "^1.9.1" (v1.24.1),
    "laravel/pint": "^1.0" (v1.21.0),
    "laravel/sail": "^1.18" (v1.41.0),
    "mockery/mockery": "^1.4.4" (1.6.12),
    "nunomaduro/collision": "^7.0" (v7.11.0),
    "phpunit/phpunit": "^10.1" (10.5.45),
    "spatie/laravel-ignition": "^2.0" (2.9.1)
}
```

A continuación hubo que hacer las migraciones, primero con los comandos "php artisan make:migration create_supermercados_table" y "php artisan make:migration create_productos_table" cree las migraciones de supermercados y productos y las modifique para ajustarse a como queria las tablas.

```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::create(table: 'supermercados', callback: function (Blueprint $table): void {
15              $table->id();
16              $table->string(column: 'nombre');
17              $table->string(column: 'direccion');
18              $table->date(column: 'vigencia_contrato');
19              $table->decimal(column: 'margen_beneficio', total: 8, places: 2);
20              $table->string(column: 'telefono');
21              $table->string(column: 'correo');
22              $table->timestamps();
23          });
24      }
25
26      /**
27       * Reverse the migrations.
28       */
29      public function down(): void
30      {
31          Schema::dropIfExists(table: 'supermercados');
32      }
33  };
34
```

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create(table: 'productos', callback: function (Blueprint $table): void {
            $table->id();
            $table->string(column: 'nombre');
            $table->string(column: 'categoria');
            $table->integer(column: 'stock');
            $table->unsignedBigInteger(column: 'supermercado_id');
            $table->foreign(columns: 'supermercado_id')->references(columns: 'id')->on(table: 'supermercados')->onDelete(action: 'cascade');
            $table->date(column: 'fecha_caducidad');
            $table->decimal(column: 'precio', total: 8, places: 2);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists(table: 'productos');
    }
};

```

Ahora tocaba ejecutar un "php artisan migrate" para que apareciera eso reflejado en la base de datos.

Ahora como queria cambiar la tabla usuarios tuve que añadir columnas nuevas, para ello primero tuve que añadir una migracion que modificara users con "php artisan make:migration add columns to users table", añadi las siguientes columnas:

```

<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    public function up(): void
    {
        Schema::table(table: 'users', callback: function (Blueprint $table): void {
            $table->string(column: 'apellidos')->nullable();
            $table->boolean(column: 'admin')->default(value: false);
            $table->string(column: 'departamento')->nullable();
            $table->string(column: 'categoria')->nullable();
        });
    }

    public function down(): void
    {
        Schema::table(table: 'users', callback: function (Blueprint $table): void {
            $table->dropColumn(columns: ['nombre', 'apellidos', 'admin', 'departamento', 'categoria']);
        });
    }
};

```

y volví a ejecutar php artisan migrate. Con esto tenemos nuestra base de datos lista.

Ahora va la crecion del seeder para meter administradores con "php artisan make:seeder AdminSeeder", cambiamos el codigo para meter un usuario y luego lo llamamos desde DatabaseSeeder.

```
namespace Database\Seeders;

use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
use App\Models\User;
```

1 reference | 0 implementations

```
class AdminSeeder extends Seeder
```

```
{
    /**
     * Run the database seeds.
     */
    0 references | 0 overrides
    public function run(): void
    {
        User::create(attributes: [
            'name' => 'Fernando',
            'email' => 'fernando@gmail.com',
            'password' => bcrypt(value: 'admin123'),
            'admin' => true,
            'apellidos' => 'Rodriguez Tejedor',
            'departamento' => 'Contabilidad',
            'categoria' => 'Director'
        ]);
    }
}
```

```
// use Illuminate\Database\Console\Seeds\WithoutModelEvents;
use Illuminate\Database\Seeder;
```

0 references | 0 implementations

```
class DatabaseSeeder extends Seeder
```

```
{
    /**
     * Seed the application's database.
     */
    0 references | 0 overrides
    public function run(): void
    {
        $this->call(class: [
            AdminSeeder::class,
        ]);
        // \App\Models\User::factory(10)->create();

        // \App\Models\User::factory()->create([
        //     'name' => 'Test User',
        //     'email' => 'test@example.com',
        // ]);
    }
}
```

Ejecutamos php artisan [db:seed](#) y se aplicarán todos los seeders, en este caso solo uno, ya tenemos un usuario admin.

Ahora creamos los controladores con "php artisan make:controller SupermercadoController --resource" y "php artisan make:controller ProductoController --resource" y agregamos en web.php las rutas.

```
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\SupermercadoController;
use App\Http\Controllers\ProductoController;

/*
|-----
| Web Routes
|-----
|
| Here is where you can register web routes for your application. These
| routes are loaded by RouteServiceProvider within the group middleware
| that is set for the web group.
|
|*/
Route::get('/', function () {
    return view('welcome');
});

Route::middleware('auth')->group(function () {
    // Solo los admin pueden acceder a Supermercados
    Route::resource('supermercados', SupermercadoController::class)->middleware('admin');
    Route::resource('productos', ProductoController::class);
});

Auth::routes();

Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
```

Creamos el middleware que servirá verificar que tipo de usuario se está logueando con "php artisan make:middleware AdminMiddleware".

```
namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;

1 reference | 0 implementations
class AdminMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
     */
    0 references | 0 overrides
    public function handle($request, Closure $next): Redirector|RedirectResponse|Response
    {
        if (auth()->check() && auth()->user()->admin) {
            return $next($request);
        }

        return redirect(to: '/home');
    }
}
```

Metemos en el kernel el middleware:

```

/**
 * The application's middleware aliases.
 *
 * Aliases may be used instead of class names to conveniently assign middleware to routes and groups.
 *
 * @var array<string, class-string|string>
 */
0 references
protected $middlewareAliases = [
    'auth' => \App\Http\Middleware\Authenticate::class,
    'admin' => \App\Http\Middleware\AdminMiddleware::class,
    'auth.basic' => \Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'auth.session' => \Illuminate\Session\Middleware\AuthenticateSession::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'password.confirm' => \Illuminate\Auth\Middleware\RequirePassword::class,
    'precognitive' => \Illuminate\Foundation\Http\Middleware\HandlePrecognitiveRequests::class,
    'signed' => \App\Http\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'verified' => \Illuminate\Auth\Middleware\EnsureEmailIsVerified::class,
];
}

```

Ahora instalamos el sistema de autentiación con: composer require laravel/ui, php artisan ui bootstrap –auth, npm install, npm run dev

Ahora toca adaptar los modelos a nuestros datos quedando de la siguiente manera:

```

6 references | 0 implementations
class Producto extends Model
{
    use HasFactory;

    0 references
    protected $fillable = [
        'nombre',
        'categoria',
        'stock',
        'supermercado_id',
        'fecha_caducidad',
        'precio'
    ];

    // Definir la relación con el modelo Supermercado
    0 references | 0 overrides
    public function supermercado(): BelongsTo
    {
        return $this->belongsTo(related: Supermercado::class);
    }
}

```

```

10 references | 0 implementations
class Supermercado extends Model
{
    use HasFactory;

    0 references
    protected $fillable = [
        'nombre',
        'direccion',
        'vigencia_contrato',
        'margen_beneficio',
        'telefono',
        'correo'
    ];
}

```

Por ultimo quedan las vistas que serian en este orden: supermercado(create, index, edit), producto(create, edit, index), app.blade.php que se modifico y register que se modifico para adaptarse a las nuevas columnas de users:

```

resources > views > supermercados > create.blade.php > div.container > form > div.mb-3
3  @extends(view: 'layouts.app')
4
5  @section(section: 'content')
6      <div class="container">
7          <h1>Crear Supermercado</h1>
8
9          <form action="{{ route(name: 'supermercados.store') }}" method="POST">
10             @csrf
11             <div class="mb-3">
12                 <label for="nombre" class="form-label">Nombre</label>
13                 <input type="text" name="nombre" class="form-control" id="nombre" required>
14             </div>
15             <div class="mb-3">
16                 <label for="direccion" class="form-label">Dirección</label>
17                 <input type="text" name="direccion" class="form-control" id="direccion" required>
18             </div>
19             <div class="mb-3">
20                 <label for="vigencia_contrato" class="form-label">Vigencia del Contrato</label>
21                 <input type="date" name="vigencia_contrato" class="form-control" id="vigencia_contrato" required>
22             </div>
23             <div class="mb-3">
24                 <label for="margen_beneficio" class="form-label">Margen de Beneficio</label>
25                 <input type="number" name="margen_beneficio" class="form-control" id="margen_beneficio" required>
26             </div>
27             <div class="mb-3">
28                 <label for="telefono" class="form-label">Teléfono</label>
29                 <input type="text" name="telefono" class="form-control" id="telefono" required>
30             </div>
31             <div class="mb-3">
32                 <label for="correo" class="form-label">Correo</label>
33                 <input type="email" name="correo" class="form-control" id="correo" required>
34             </div>
35             <button type="submit" class="btn btn-primary">Crear Supermercado</button>
36         </form>
37     </div>
38 @endsection
39

```

```

@extends(view: 'layouts.app')

@section(section: 'content')
<div class="container">
    <h1>Lista de Supermercados</h1>

    <!-- Mostrar mensaje de éxito si hay -->
    @if(session(key: 'success'))
        <div class="alert alert-success">{{ session(key: 'success') }}</div>
    @endif

    <a href="{{ route(name: 'supermercados.create') }}" class="btn btn-primary mb-3">Crear Supermercado</a>

    @foreach($supermercados as $supermercado)
        <div class="mb-4">
            <h3>{{ $supermercado->nombre }}</h3>
            <p>Dirección: {{ $supermercado->direccion }}</p>
            <p>Vigencia del contrato: {{ $supermercado->vigencia_contrato }}</p>
            <p>Margen de beneficio: {{ $supermercado->margen_beneficio }}</p>
            <p>Teléfono: {{ $supermercado->telefono }}</p>
            <p>Correo: {{ $supermercado->correo }}</p>

            <a href="{{ route(name: 'supermercados.edit', parameters: $supermercado) }}" class="btn btn-warning">Editar</a>

            <form action="{{ route(name: 'supermercados.destroy', parameters: $supermercado) }}" method="POST" style="display:inline;">
                @csrf
                @method('DELETE')
                <button type="submit" class="btn btn-danger">Eliminar</button>
            </form>
        </div>
    @endforeach
</div>
@endsection

```

```

3  @extends(view: 'layouts.app')
4
5  @section(section: 'content')
6      <div class="container">
7          <h1>Editar Supermercado</h1>
8
9          <form action="{{ route(name: 'supermercados.update', parameters: $supermercado) }}" method="POST">
10              @csrf
11              @method('PUT')
12              <div class="mb-3">
13                  <label for="nombre" class="form-label">Nombre</label>
14                  <input type="text" name="nombre" class="form-control" id="nombre" value="{{ $supermercado->nombre }}" required>
15              </div>
16              <div class="mb-3">
17                  <label for="direccion" class="form-label">Dirección</label>
18                  <input type="text" name="direccion" class="form-control" id="direccion" value="{{ $supermercado->direccion }}" required>
19              </div>
20              <div class="mb-3">
21                  <label for="vigencia_contrato" class="form-label">Vigencia del Contrato</label>
22                  <input type="date" name="vigencia_contrato" class="form-control" id="vigencia_contrato" value="{{ $supermercado->vigencia_contrato }}" required>
23              </div>
24              <div class="mb-3">
25                  <label for="margen_beneficio" class="form-label">Margen de Beneficio</label>
26                  <input type="number" name="margen_beneficio" class="form-control" id="margen_beneficio" value="{{ $supermercado->margen_beneficio }}" required>
27              </div>
28              <div class="mb-3">
29                  <label for="telefono" class="form-label">Teléfono</label>
30                  <input type="text" name="telefono" class="form-control" id="telefono" value="{{ $supermercado->telefono }}" required>
31              </div>
32              <div class="mb-3">
33                  <label for="correo" class="form-label">Correo</label>
34                  <input type="email" name="correo" class="form-control" id="correo" value="{{ $supermercado->correo }}" required>
35              </div>
36              <button type="submit" class="btn btn-primary">Actualizar Supermercado</button>
37          </form>
38      </div>

```

```

resources > views > productos > create.blade.php > ...
6      <div class="container">
7          <h1>Crear Producto</h1>
8
9          <form action="{{ route('productos.store') }}" method="POST">
10             @csrf
11             <div class="mb-3">
12                 <label for="nombre" class="form-label">Nombre</label>
13                 <input type="text" name="nombre" class="form-control" id="nombre" required>
14             </div>
15             <div class="mb-3">
16                 <label for="categoria" class="form-label">Categoría</label>
17                 <input type="text" name="categoria" class="form-control" id="categoria" required>
18             </div>
19             <div class="mb-3">
20                 <label for="stock" class="form-label">Stock</label>
21                 <input type="number" name="stock" class="form-control" id="stock" required>
22             </div>
23             <div class="mb-3">
24                 <label for="supermercado_id" class="form-label">Supermercado</label>
25                 <select name="supermercado_id" class="form-control" required>
26                     @foreach($supermercados as $supermercado)
27                         <option value="{{ $supermercado->id }}">{{ $supermercado->nombre }}</option>
28                     @endforeach
29                 </select>
30             </div>
31             <div class="mb-3">
32                 <label for="fecha_caducidad" class="form-label">Fecha de caducidad</label>
33                 <input type="date" name="fecha_caducidad" class="form-control" id="fecha_caducidad" required>
34             </div>
35             <div class="mb-3">
36                 <label for="precio" class="form-label">Precio</label>
37                 <input type="number" step="0.01" name="precio" class="form-control" id="precio" required>
38             </div>
39             <button type="submit" class="btn btn-primary">Crear Producto</button>
40         </form>
41     </div>
42 @endsection

```

```

resources > views > productos > edit.blade.php > ...
6      <div class="container">
7
8          <form action="{{ route('productos.update', parameters: $producto) }}" method="POST">
9              @csrf
10              @method('PUT')
11              <div class="mb-3">
12                  <label for="nombre" class="form-label">Nombre</label>
13                  <input type="text" name="nombre" class="form-control" id="nombre" value="{{ $producto->nombre }}" required>
14              </div>
15              <div class="mb-3">
16                  <label for="categoria" class="form-label">Categoría</label>
17                  <input type="text" name="categoria" class="form-control" id="categoria" value="{{ $producto->categoria }}" required>
18              </div>
19              <div class="mb-3">
20                  <label for="stock" class="form-label">Stock</label>
21                  <input type="number" name="stock" class="form-control" id="stock" value="{{ $producto->stock }}" required>
22              </div>
23              <div class="mb-3">
24                  <label for="supermercado_id" class="form-label">Supermercado</label>
25                  <select name="supermercado_id" class="form-control" required>
26                      @foreach($supermercados as $supermercado)
27                          <option value="{{ $supermercado->id }}" {{ $supermercado->id == $producto->supermercado_id ? 'selected' : '' }}>
28                              {{ $supermercado->nombre }}
29                          </option>
30                      @endforeach
31                  </select>
32              </div>
33              <div class="mb-3">
34                  <label for="fecha_caducidad" class="form-label">Fecha de caducidad</label>
35                  <input type="date" name="fecha_caducidad" class="form-control" id="fecha_caducidad" value="{{ $producto->fecha_caducidad }}" required>
36              </div>
37              <div class="mb-3">
38                  <label for="precio" class="form-label">Precio</label>
39                  <input type="number" step="0.01" name="precio" class="form-control" id="precio" value="{{ $producto->precio }}" required>
40              </div>
41              <button type="submit" class="btn btn-primary">Actualizar Producto</button>
42          </form>
43      </div>
44

```



```

resources > views > productos > index.blade.php > div.container
5 @section(section: 'content')
6 <div class="container">
7     <h1>Lista de Productos</h1>
8
9     <!-- Mostrar mensaje de éxito si hay -->
10    @if(session(key: 'success'))
11        <div class="alert alert-success">{{ session(key: 'success') }}</div>
12    @endif
13
14    <a href="{{ route(name: 'productos.create') }}" class="btn btn-primary mb-3">Crear Producto</a>
15
16    @foreach($productos as $producto)
17        <div class="mb-4">
18            <h3>{{ $producto->nombre }}</h3>
19            <p>Categoría: {{ $producto->categoria }}</p>
20            <p>Stock: {{ $producto->stock }}</p>
21            <p>Supermercado: {{ $producto->supermercado->nombre }}</p>
22            <p>Precio: ${{ $producto->precio }}</p>
23            <p>Fecha de caducidad: {{ $producto->fecha_caducidad }}</p>
24
25            <a href="{{ route(name: 'productos.edit', parameters: $producto) }}" class="btn btn-warning">Editar</a>
26
27            <form action="{{ route(name: 'productos.destroy', parameters: $producto) }}" method="POST" style="display:inline;">
28                @csrf
29                @method('DELETE')
30                <button type="submit" class="btn btn-danger">Eliminar</button>
31            </form>
32        </div>
33    @endforeach
34 </div>
35 @endsection
36

```

```

69 </li>
70
71 <!-- Enlaces de administración -->
72 @if(auth()->check())
73     @if(auth()->user()->admin)
74         <li class="nav-item">
75             <a class="nav-link" href="{{ route(name: 'supermercados.index') }}">Gestionar Supermercados</a>
76         </li>
77     @endif
78     <li class="nav-item">
79         <a class="nav-link" href="{{ route(name: 'productos.index') }}">Gestionar Productos</a>
80     </li>
81 @endif

```

```

@section(section: 'content')
<div class="container">
    <h2>Registro de Usuario</h2>

    <form method="POST" action="{{ route(name: 'register') }}">
        @csrf

        <div class="row mb-3">
            <label for="name" class="col-md-4 col-form-label text-md-end">{{ __(key: 'Nombre') }}</label>

            <div class="col-md-6">
                <input id="name" type="text" class="form-control @error('name') is-invalid @enderror" name="name" value="{{ old(key: 'name') }}"

                @error('name')
                <span class="invalid-feedback" role="alert">
                    <strong>{{ $message }}</strong>
                </span>
                @enderror
            </div>
        </div>
    </div>

```

Resto de la clase register son los mismos divs pero con un dato diferente