

PLAN DE TRABAJO DEL ESTUDIANTE

1. INFORMACIÓN GENERAL

Apellidos y Nombres : **SAAVEDRA SUCA JERSON** ID : **001270288**
 Dirección Zonal/CFP : **CFP -Villa el Salvador / Escuela ETI**
 Carrera : **Desarrollo de Software** Semestre : **5**
 Curso/ Mód. Formativo : **Desarrollo de Aplicaciones Web**
 Tema del Trabajo : **Framework Laravel Parte 1**

2. PLANIFICACIÓN DEL TRABAJO

N°	ACTIVIDADES/ ENTREGABLES	CRONOGRAMA/ FECHA DE ENTREGA									
		26/10	27/10	28/10	30/10						
	Entregable 1										
1	Descargar las dependencias composer	X									
2	Ejecutar el comando y crear el proyecto laravel, configurar .env	X									
3	Crear y configurar las migraciones	X									
4	Crear base de datos		X								
5	Crear las rutas		X								
6	Crear los modelos y controladores		X								
7	Crear la vista para mostrar los datos			X							
8	Probar el funcionamiento del Proyecto			X							
9	Entrega de entregable 1				X						

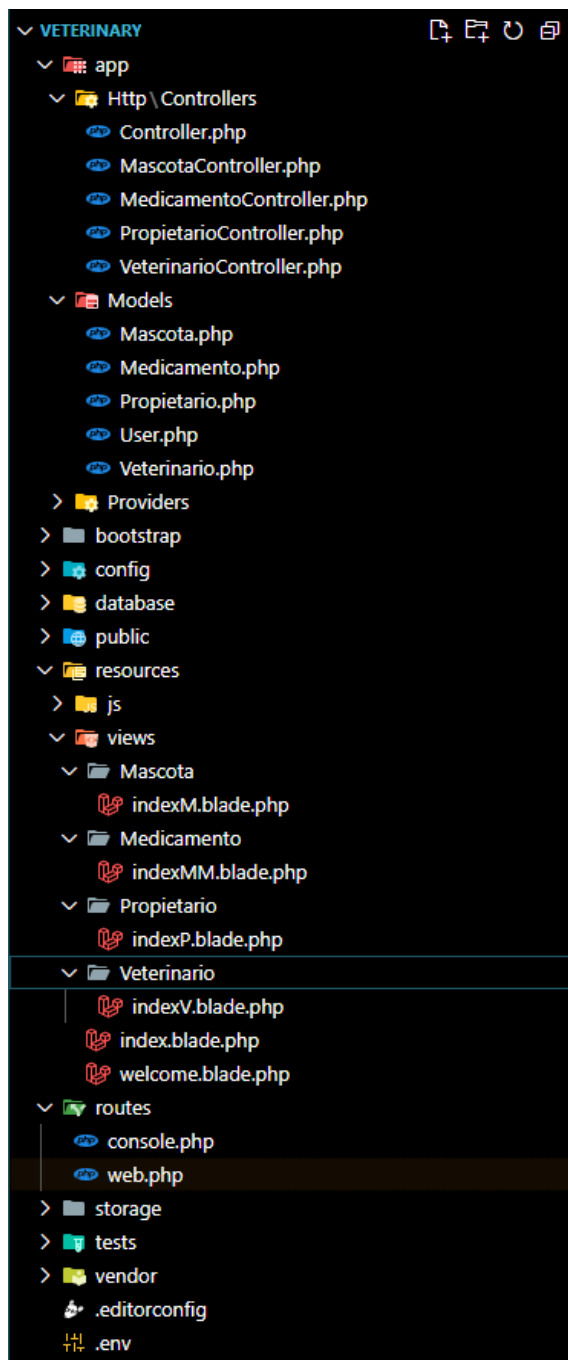
3. ENTREGABLES:

Durante la investigación de estudio, deberán de dar solución a los planteamientos de cada entregable:

Nº	ENTREGABLE 1
1	<p style="text-align: center;">Entregable 1: WebVeterinary - Laravel</p> <p>En la primera fase, se desarrolló un sistema utilizando Laravel, donde se realizaron migraciones para definir la estructura de las tablas en la base de datos. Se crearon modelos que representaban las entidades del sistema: Mascota, Propietario, Veterinario y Medicamento, facilitando la interacción con los datos.</p> <p>Se implementaron controladores, como MascotaController, PropietarioController, VeterinarioController y MedicamentoController, para gestionar las operaciones relacionadas con cada entidad. Las vistas se diseñaron utilizando Blade, lo que permitió crear interfaces de usuario dinámicas y fáciles de mantener.</p> <p>Las rutas se configuraron en el archivo de rutas de Laravel, permitiendo definir las distintas URL del sistema y su relación con los controladores. Además, se configuró el archivo .env para establecer la conexión con la base de datos, asegurando que el sistema pudiera almacenar y recuperar datos de manera efectiva.</p>

Desarrollo del Entregable N°1- DIBUJO / ESQUEMA / DIAGRAMA

Estructura del Proyecto: Se organizaron los paquetes en controladores, modelos y vistas, rutas lo que promovió una arquitectura clara y modular.



Modelos

1. Mascota

Representa a los animales que son atendidos en la veterinaria. Contiene atributos como nombre, especie, raza, edad, y color. Este modelo permite gestionar la información básica sobre cada mascota y sus respectivas citas.

```
Mascota.php X
app > Models > Mascota.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  2 references | 0 implementations
   class Mascota extends Model
9  {
10     use HasFactory;
11     0 references
   protected $table="mascotas";
12     0 references
   protected $primaryKey="mascota_id";
13     0 references
   public $timestamps=false;
14 }
15
```

2. Propietario

Representa a los dueños de las mascotas. Incluye información como nombre, apellido, DNI, correo y teléfono. Este modelo es crucial para vincular a cada mascota con su propietario, facilitando la gestión de la información de contacto y la programación de citas.



```
Propietario.php X
app > Models > Propietario.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  2 references | 0 implementations
9  class Propietario extends Model
10 {
11     use HasFactory;
12     0 references
13     protected $table="propietarios";
14     0 references
15     protected $primaryKey="propietario_id";
16     0 references
17     public $timestamps=false;
18 }
19
```

3. Veterinario

Representa a los profesionales que atienden a las mascotas. Contiene datos como nombre, apellido, especialidad, y teléfono. Este modelo permite gestionar la información del personal veterinario, facilitando la asignación de veterinarios a las citas de las mascotas.



```
Veterinario.php X
app > Models > Veterinario.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  2 references | 0 implementations
9  class Veterinario extends Model
10 {
11     use HasFactory;
12     0 references
13     protected $table="veterinarios";
14     0 references
15     protected $primaryKey="veterinario_id";
16     0 references
17     public $timestamps=false;
18 }
```

4. Medicamento

Representa los medicamentos que pueden ser prescritos a las mascotas. Incluye atributos como nombre, descripción, y precio. Este modelo es esencial para la gestión de tratamientos y recetas, permitiendo asociar medicamentos específicos con las mascotas y sus diagnósticos.



```
Medicamento.php X
app > Models > Medicamento.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  2 references | 0 implementations
9  class Medicamento extends Model
10 {
11     use HasFactory;
12     0 references
13     protected $table="medicamentos";
14     0 references
15     protected $primaryKey="medicamento_id";
16     0 references
17     public $timestamps=false;
18 }
```

Controladores

Los controladores son componentes clave en la arquitectura MVC (Modelo-Vista-Controlador) de Laravel. Su función es gestionar la lógica de la aplicación, procesar las solicitudes del usuario y retornar las respuestas adecuadas. En WebVeterinary, se implementaron los siguientes controladores:

- **MascotaController:** Encargado de manejar las operaciones relacionadas con las mascotas, como la creación, actualización y eliminación de registros. También se ocupa

de mostrar la lista de mascotas y los detalles de cada una.

```

MascotaController.php X
app > Http > Controllers > MascotaController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Mascota;
7
8  2 references | 0 implementations
9  class MascotaController extends Controller
10 {
11     1 reference | 0 overrides
12     public function index_GET(): Factory|View {
13         $listaMascotas = Mascota::all();
14         return view('Mascota.indexM',compact(var_name: 'listaMascotas'));
15     }
16 }

```

- **PropietarioController:** Gestiona las acciones vinculadas a los propietarios de las mascotas, permitiendo agregar, editar y eliminar información de contacto, así como relacionar propietarios con sus mascotas.

```

PropietarioController.php X
app > Http > Controllers > PropietarioController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Propietario;
6  use Illuminate\Http\Request;
7
8  2 references | 0 implementations
9  class PropietarioController extends Controller
10 {
11     1 reference | 0 overrides
12     public function index_GET(): Factory|View {
13         $listaPropietarios = Propietario::all();
14         return view('Propietario.indexP',compact(var_name: 'listaPropietarios'));
15     }
16 }

```

- **VeterinarioController:** Administra los registros de veterinarios, permitiendo crear, editar y eliminar información sobre el personal veterinario, así como asignar veterinarios a citas.

```

VeterinarioController.php X
app > Http > Controllers > VeterinarioController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Veterinario;
6  use Illuminate\Http\Request;
7
8  2 references | 0 implementations
9  class VeterinarioController extends Controller
10 {
11     1 reference | 0 overrides
12     public function index_GET(): Factory|View {
13         $listaVeterinarios = Veterinario::all();
14         return view('Veterinario.indexV',compact('listaVeterinarios'));
15     }
16 }
  
```

- **MedicamentoController:** Se ocupa de la gestión de los medicamentos, incluyendo su creación, edición y eliminación, además de asociarlos a tratamientos específicos de las mascotas.

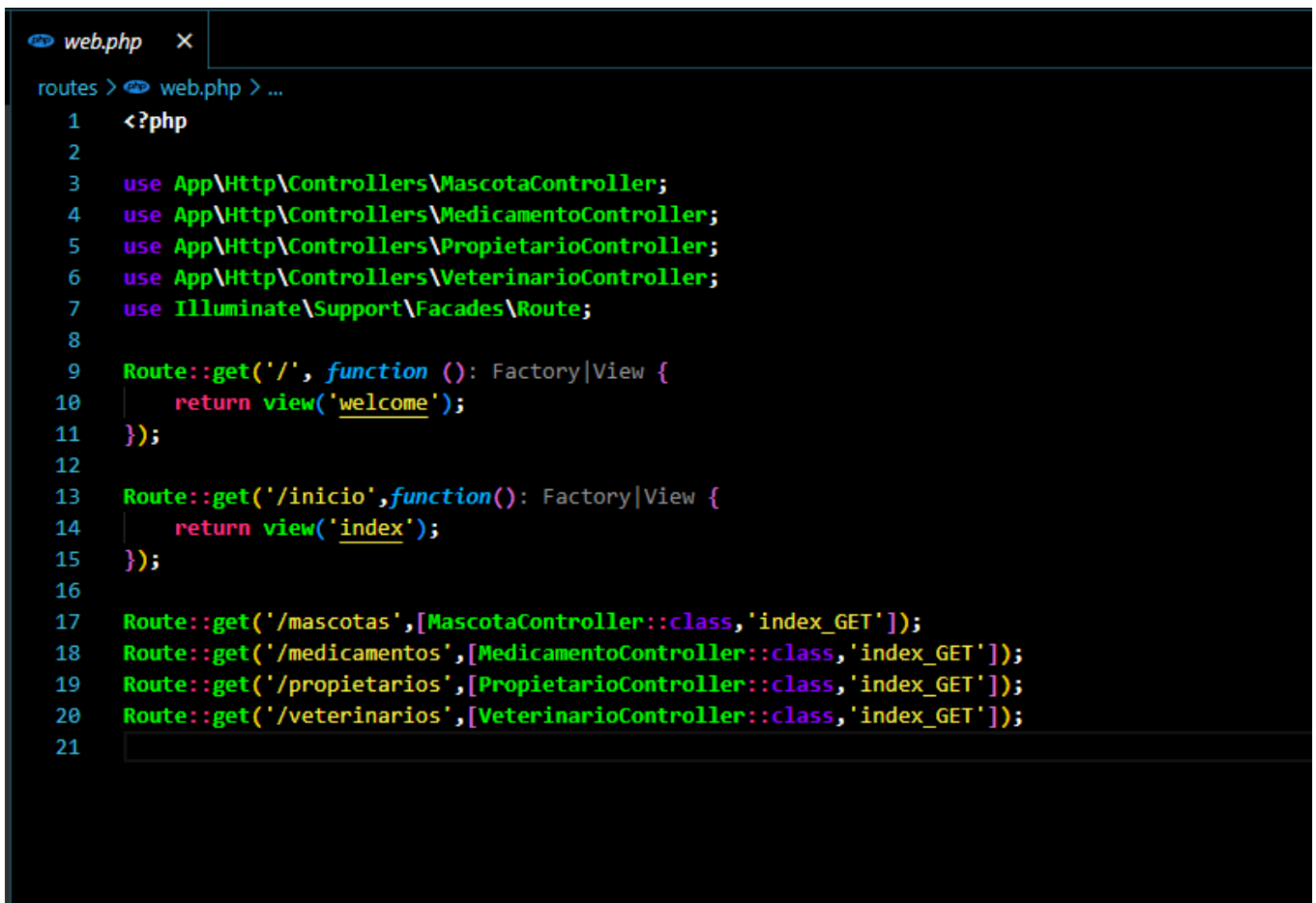
```

MedicamentoController.php X
app > Http > Controllers > MedicamentoController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Medicamento;
6  use Illuminate\Http\Request;
7
8  2 references | 0 implementations
9  class MedicamentoController extends Controller
10 {
11     1 reference | 0 overrides
12     public function index_GET(): Factory|View {
13         $listaMedicamentos = Medicamento::all();
14         return view('Medicamento.indexMM',compact('listaMedicamentos'));
15     }
16 }
  
```

Rutas

Las rutas en Laravel son responsables de definir la URL que se asigna a cada controlador. Permiten que el sistema responda a las solicitudes HTTP adecuadas. En WebVeterinary, se configuraron rutas para cada entidad y acción, tales como:

- **Rutas de Mascota:** Incluyen rutas para listar, crear, editar y eliminar mascotas, asignadas al MascotaController.
- **Rutas de Propietario:** Definen las acciones para gestionar los propietarios a través del PropietarioController.
- **Rutas de Veterinario:** Permiten acceder a las funcionalidades de gestión de veterinarios mediante el VeterinarioController.
- **Rutas de Medicamento:** Configuran las operaciones relacionadas con los medicamentos, conectando con el MedicamentoController.



```
web.php x
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\MascotaController;
4  use App\Http\Controllers\MedicamentoController;
5  use App\Http\Controllers\PropietarioController;
6  use App\Http\Controllers\VeterinarioController;
7  use Illuminate\Support\Facades\Route;
8
9  Route::get('/', function () { Factory|View {
10     return view('welcome');
11 });
12
13 Route::get('/inicio',function(): Factory|View {
14     return view('index');
15 });
16
17 Route::get('/mascotas',[MascotaController::class,'index_GET']);
18 Route::get('/medicamentos',[MedicamentoController::class,'index_GET']);
19 Route::get('/propietarios',[PropietarioController::class,'index_GET']);
20 Route::get('/veterinarios',[VeterinarioController::class,'index_GET']);
21
```

Vistas

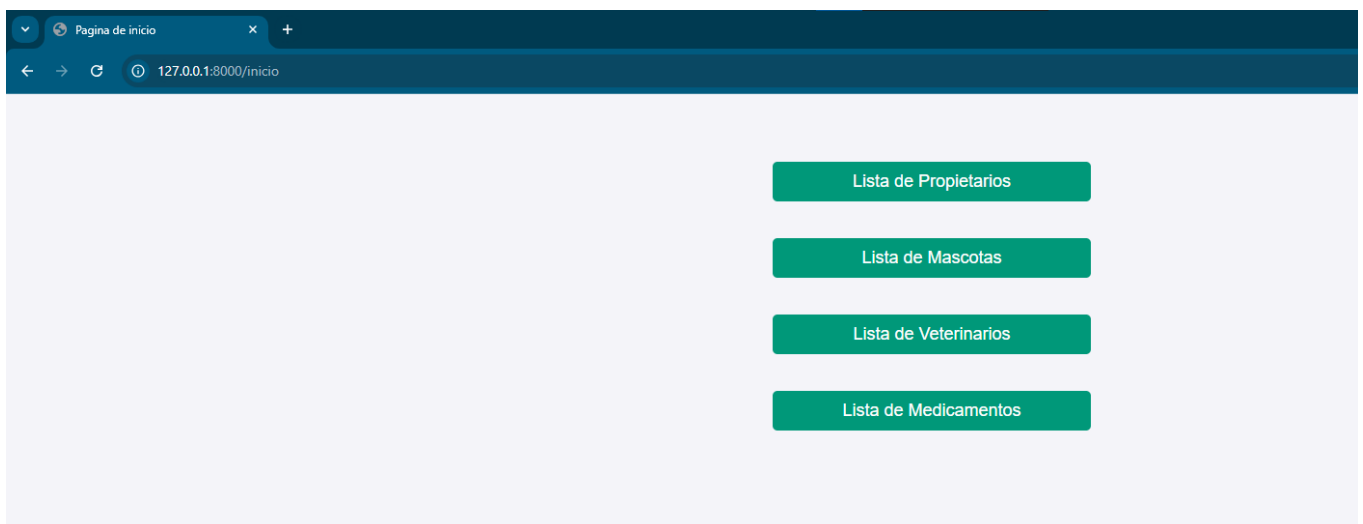
Las vistas son archivos que definen cómo se presentan los datos al usuario. En Laravel, se utilizan las plantillas Blade para crear vistas. En WebVeterinary, se diseñaron las siguientes vistas:

- **Vista de la Pagina principal:** Muestra una lista de los modulos.

```

index.blade.php x
resources > views > index.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <title>Pagina de inicio</title>
8
9     <link rel="stylesheet" href="{{ asset('css/indexStyle.css') }}">
10
11 </head>
12 <body>
13     <div class="link-container">
14         <a href="{{ url('/propietarios') }}">Lista de Propietarios</a><br>
15         <a href="{{ url('/mascotas') }}">Lista de Mascotas</a><br>
16         <a href="{{ url('/veterinarios') }}">Lista de Veterinarios</a><br>
17         <a href="{{ url('/medicamentos') }}">Lista de Medicamentos</a><br>
18     </div>
19 </body>
20 </html>
21

```



- **Vista de Mascotas:** Muestra una lista de todas las mascotas registradas y permite acceder a formularios.

```

indexM.blade.php ×
resources > views > Mascota > indexM.blade.php > ...
1  <!DOCTYPE html>
2  <html Lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Listado de Mascotas</title>
8
9      <link rel="stylesheet" href="{{ asset('css/listarStyle.css') }}">
10
11 </head>
12 <body>
13     <h3>¡Aplicación WEB - Mascotas!</h3>
14     <div>
15         <table>
16             <caption style="line-height: 30px">¡Mascotas - Listar!</caption>
17             <tr>
18                 <td><b>ID</b></td>
19                 <td><b>Nombre</b></td>
20                 <td><b>Especie</b></td>
21                 <td><b>Raza</b></td>
22                 <td><b>Género</b></td>
23                 <td><b>Edad</b></td>
24                 <td><b>Color</b></td>
25             </tr>
26             @foreach($listaMascotas as $mascota)
27                 <tr>
28                     <td>{{ $mascota->mascota_id }}</td>
29                     <td>{{ $mascota->nombre }}</td>
30                     <td>{{ $mascota->especie }}</td>
31                     <td>{{ $mascota->raza }}</td>
32                     <td>{{ $mascota->genero }}</td>
33                     <td>{{ $mascota->edad }}</td>
34                     <td>{{ $mascota->color }}</td>
35                 </tr>
36             @endforeach
37         </table>
38     </div>
39
40     <div class="link-container">
41         <a href="{{ url('/inicio') }}">Volver</a><br>
42     </div>
43 </body>
44 </html>
45

```

Lista de Mascotas

127.0.0.1:8000/mascotas

¡Aplicación WEB - Mascotas!

¡Mascotas - Listar!

ID	Nombre	Especie	Raza	Género	Edad	Color
1	Firulais	Perro	Bulldog Francés	H	3	Marrón
2	Michi	Gato	Siames	M	2	Gris
3	Rocky	Perro	Labrador	H	5	Negro
4	Pelusa	Gato	Persa	M	4	Bianco
5	Thor	Perro	Pastor Alemán	H	6	Dorado

Volver

- **Vista de Propietarios:** Presenta la información de los propietarios, permitiendo gestionar sus datos.

```

indexP.blade.php X
resources > views > Propietario > indexP.blade.php > Laraphense (Laravel PHP Intellisense) > html > body > div.link-container
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Listado de Propietarios</title>
8
9      <link rel="stylesheet" href="{{ asset('css/listarStyle.css') }}">
10
11 </head>
12 <body>
13     <h3>¡Aplicación WEB - Propietarios!</h3>
14     <div>
15         <table>
16             <caption style="line-height: 30px">¡Propietarios - Listar!</caption>
17             <tr>
18                 <td><b>ID</b></td>
19                 <td><b>DNI</b></td>
20                 <td><b>Nombre</b></td>
21                 <td><b>Apellido</b></td>
22                 <td><b>Teléfono</b></td>
23                 <td><b>Correo</b></td>
24             </tr>
25             @foreach($listaPropietarios as $propietario)
26                 <tr>
27                     <td>{{ $propietario->propietario_id }}</td>
28                     <td>{{ $propietario->dni }}</td>
29                     <td>{{ $propietario->nombre }}</td>
30                     <td>{{ $propietario->apellido }}</td>
31                     <td>{{ $propietario->telefono }}</td>
32                     <td>{{ $propietario->correo }}</td>
33                 </tr>
34             @endforeach
35         </table>
36     </div>
37     <div class="link-container">
38         <a href="{{ url('/inicio') }}">Volver</a><br>
39     </div>
40 </body>
41 </html>
42

```

¡Aplicación WEB - Propietarios!

¡Propietarios - Listar!

ID	DNI	Nombre	Apellido	Teléfono	Correo
1	75263412	Carlos	Quispe	987654321	carlos.quispe@hotmail.com
2	81234567	Ana	Pérez	912345678	ana.perez@hotmail.com
3	75612389	Luis	Gómez	965432198	luis.gomez@hotmail.com
4	78456932	Maria	Huamán	975312654	maria.huaman@hotmail.com
5	71928374	José	Rojas	954328176	jose.rojas@hotmail.com

Volver

- **Vista de Veterinarios:** Permite visualizar y gestionar la información del personal veterinario.

```

indexV.blade.php x
resources > views > Veterinario > indexV.blade.php > Laravel Blade Snippets > html > body > div.link-container
1  <!DOCTYPE html>
2  <html Lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Listado de Veterinarios</title>
8
9      <link rel="stylesheet" href="{{ asset('css/listarStyle.css') }}">
10
11 </head>
12 <body>
13     <h3>¡Aplicación WEB - Veterinarios!</h3>
14     <div>
15         <table>
16             <caption style="line-height: 30px">¡Veterinarios - Listar!</caption>
17             <tr>
18                 <td><b>ID</b></td>
19                 <td><b>DNI</b></td>
20                 <td><b>Nombre</b></td>
21                 <td><b>Apellido</b></td>
22                 <td><b>Teléfono</b></td>
23                 <td><b>Especialidad</b></td>
24                 <td><b>Correo</b></td>
25                 <td><b>Fecha de Nacimiento</b></td>
26             </tr>
27             @foreach($listaVeterinarios as $veterinario)
28                 <tr>
29                     <td>{{ $veterinario->veterinario_id }}</td>
30                     <td>{{ $veterinario->dni }}</td>
31                     <td>{{ $veterinario->nombre }}</td>
32                     <td>{{ $veterinario->apellido }}</td>
33                     <td>{{ $veterinario->telefono }}</td>
34                     <td>{{ $veterinario->especialidad }}</td>
35                     <td>{{ $veterinario->correo }}</td>
36                     <td>{{ $veterinario->fnacimiento }}</td>
37                 </tr>
38             @endforeach
39         </table>
40     </div>
41     <div class="link-container">
42         <a href="{{ url('/inicio') }}">Volver</a><br>
43     </div>
44 </body>
45 </html>
46

```

¡Aplicación WEB - Veterinarios!

¡Veterinarios - Listar!

ID	DNI	Nombre	Apellido	Teléfono	Especialidad	Correo	Fecha de Nacimiento
1	70123456	Fernando	Salazar	987123456	Medicina General	fernando.salazar@hotmail.com	1985-02-14
2	70234567	Gloria	Villanueva	987654321	Dermatología	gloria.villanueva@hotmail.com	1980-08-25
3	70345678	Andrés	Hernández	912345678	Cirugía	andres.hernandez@hotmail.com	1990-05-18
4	70456789	Patricia	Mendoza	987456123	Odontología	patricia.mendoza@hotmail.com	1983-11-02
5	70567890	Marco	Torres	978321654	Oftalmología	marco.torres@hotmail.com	1987-04-30

Volver

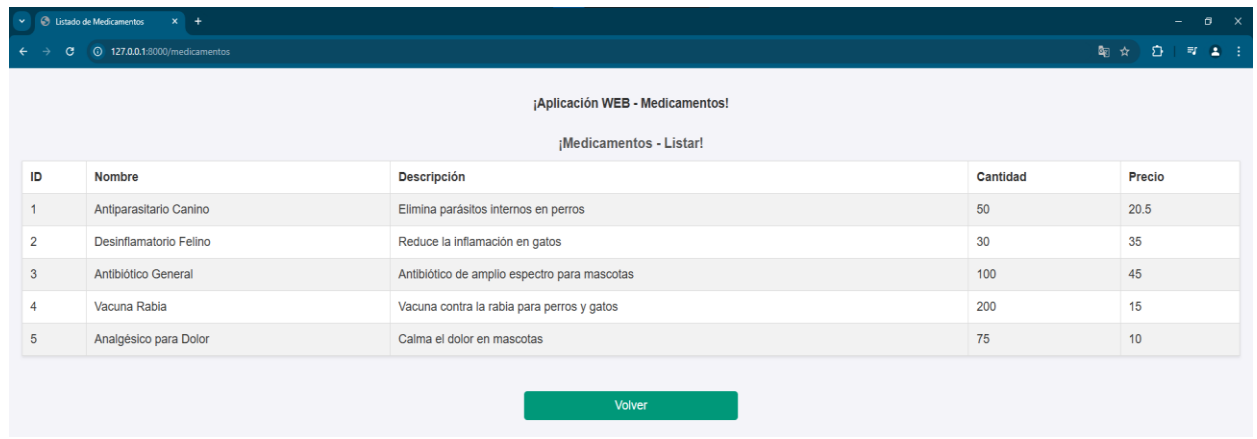
- **Vista de Medicamentos:** Muestra los medicamentos disponibles y permite gestionar su información.

```

indexMM.blade.php x
resources > views > Medicamento > indexMM.blade.php > Laraphense (Laravel PHP Intellisense) > html > body > div.link-container

1 <!DOCTYPE html>
2 <html Lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <title>Listado de Medicamentos</title>
8
9 <link rel="stylesheet" href="{{ asset('css/listarStyle.css') }}">
10
11 </head>
12 <body>
13 <h3>¡Aplicación WEB - Medicamentos!</h3>
14 <div>
15 <table>
16 <caption style="line-height: 30px">¡Medicamentos - Listar!</caption>
17 <tr>
18 <td><b>ID</b></td>
19 <td><b>Nombre</b></td>
20 <td><b>Descripción</b></td>
21 <td><b>Cantidad</b></td>
22 <td><b>Precio</b></td>
23 </tr>
24 @foreach($listaMedicamentos as $medicamento)
25 <tr>
26 <td>{{ $medicamento->medicamento_id }}</td>
27 <td>{{ $medicamento->nombre }}</td>
28 <td>{{ $medicamento->descripcion }}</td>
29 <td>{{ $medicamento->cantidad }}</td>
30 <td>{{ $medicamento->precio }}</td>
31 </tr>
32 @endforeach
33 </table>
34 </div>
35 <div class="link-container">
36 <a href="{{ url('/inicio') }}">Volver</a><br>
37 </div>
38 </body>
39 </html>
40

```



¡Aplicación WEB - Medicamentos!

¡Medicamentos - Listar!

ID	Nombre	Descripción	Cantidad	Precio
1	Antiparasitario Canino	Elimina parásitos internos en perros	50	20.5
2	Desinflamatorio Felino	Reduce la inflamación en gatos	30	35
3	Antibiótico General	Antibiótico de amplio espectro para mascotas	100	45
4	Vacuna Rabia	Vacuna contra la rabia para perros y gatos	200	15
5	Analgésico para Dolor	Calma el dolor en mascotas	75	10

Volver

Extras para el funcionamiento de los datos y poder mostrarlos
Migraciones de Veterinario, Medicamento, Propietario y Mascota.



2024_10_28_044123_create_propietarios_table.php X

database > migrations > 2024_10_28_044123_create_propietarios_table.php > class > up > Closure

```
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('propietarios', function (Blueprint $table): void {
15             $table->id('propietario_id');
16
17             $table->integer('dni');
18             $table->string('nombre',100);
19             $table->string('apellido',150);
20             $table->integer('telefono');
21             $table->string('correo',250);
22
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('propietarios');
33     }
34 };
35
```



2024_10_28_044134_create_veterinarios_table.php X

database > migrations > 2024_10_28_044134_create_veterinarios_table.php > class > up > Closure

```
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('veterinarios', function (Blueprint $table) {
15             $table->id('veterinario_id');
16
17             $table->integer('dni');
18             $table->string('nombre',100);
19             $table->string('apellido',100);
20             $table->integer('telefono');
21             $table->string('especialidad',150);
22             $table->string('correo',250);
23             $table->date('fnacimiento')->nullable();
24
25             $table->timestamps();
26         });
27     }
28
29     /**
30      * Reverse the migrations.
31      */
32     public function down(): void
33     {
34         Schema::dropIfExists('veterinarios');
35     }
36 };
37
38
39
```



2024_10_28_051057_create_medicamentos_table.php X

database > migrations > 2024_10_28_051057_create_medicamentos_table.php > class > up

```
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('medicamentos', function (Blueprint $table): void {
15             $table->id('medicamento_id');
16
17             $table->string('nombre',100);
18             $table->string('descripcion')->nullable();
19             $table->integer('cantidad');
20             $table->double('precio');
21
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('medicamentos');
32     }
33 };
34
```

```

2024_10_29_044108_create_mascotas_table.php X
database > migrations > 2024_10_29_044108_create_mascotas_table.php > class > up
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('mascotas', function (Blueprint $table): void {
15             $table->id('mascota_id');
16
17             $table->string('nombre', 150);
18             $table->string('especie', 150);
19             $table->string('raza', 100);
20             $table->enum('genero', ['H', 'M']);
21             $table->integer('edad');
22             $table->string('color', 50);
23             $table->timestamps();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('mascotas');
33     }
34 };
35

```

HOJA DE PLANIFICACIÓN (Entregable 1)
PROCESO DE EJECUCIÓN

OPERACIONES / PASOS /SUBPASOS	SEGURIDAD / MEDIO AMBIENTE / NORMAS -ESTANDARES
1. Selección y Análisis del Caso Elegido	Se mantuvo un ambiente tranquilo y sin distracciones para la lectura detallada del caso.
1.1: Seleccionar el caso del Veterinario como proyecto a desarrollar.	Fomentar un ambiente de colaboración para la generación de preguntas claras y pertinentes.
1.2: Leer y comprender en detalle los requerimientos del sistema de veterinario	Proporcionar un espacio físico adecuado para reuniones productivas y participativas.
1.3: Identificar las posibles entidades.	Acceso a bases de datos en línea y otras fuentes de información confiables.
2.Diseño de la Arquitectura del Sistema	Utilizar herramientas de software adecuadas para la recopilación y organización eficiente de la información.
2.1: Definir las los modelos	
2.2: Crear las migraciones	
2.3: Definir las rutas.	
3.Implementación de Clases de Veterinario	
3.1: Crear las modelos que implementan la interfaz de las entidades de veterinario	
3.2: Hacer la lógica necesaria dentro de cada clase.	
4.Hacer los Controladores	
4.1: Configurar los controladores de cada entidad	
4.2: Asegurarse de que el controlador pueda utilizar diferentes servicios sin necesidad de cambiar su código.	
5.Pruebas y Verificación	
5.1: Realizar pruebas para verificar que cada entidad puede realizar los servicios en las páginas Blade.	
5.2: Revisar y ajustar el código según los <u>resultados</u> de las pruebas para garantizar el correcto funcionamiento.	

INSTRUCCIONES: debes ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. No olvides los aspectos de calidad, medio ambiente y SHI.

LISTA DE RECURSOS

INSTRUCCIONES: completa la lista de recursos necesarios para la ejecución del trabajo.

1. MÁQUINAS Y EQUIPOS

1 laptop



2. HERRAMIENTAS E INSTRUMENTOS

Laravel

Visual Studio Code

Word



3. MATERIALES E INSUMOS

