

UNIVERSIDAD INTERNACIONAL DE VALENCIA (VIU)

MASTER UNIVERSITARIO DE DESARROLLO DE APLICACIONES Y SERVICIOS WEB

ACTIVIDAD 2: ANGULAR

Asignatura:

Desarrollo de aplicaciones Web II: front-end

Docente:

Ing. Ismael de Fez Lava

Integrantes:

Flores Quispe José Lolo

Guarquila Salto Mauro Geovanny

Landa Manotoa Ángel Patricio

Rodríguez Padilla Randy Jimmy

ÍNDICE

1.	INTRODUCCIÓN	1
2.	OBJETIVO	1
3.	JUSTIFICACIÓN	1
4.	DEFINICIÓN DE LA ACTIVIDAD.....	2
5.	DEFINICIÓN DEL PROYECTO	2
6.	DESARROLLO	4
	CONCLUSIONES	12
	REFERENCIAS	12

ÍNDICE DE FIGURAS

Figura 1	Arquitectura de la aplicación.....	2
Figura 2	Diagrama conceptual de la implementación.....	3
Figura 3	Diagrama de casos de uso: gestión de agenda.....	4
Figura 4	Estructura del proyecto.....	6
Figura 5	<i>Archivos de componentes de consultorios</i>	7
Figura 6	<i>Estructura del componente consultorios</i>	7
Figura 7	Archivo consultorios.component.html.....	8
Figura 8	Archivo app.routes.ts	9
Figura 9	Archivo api.service.ts.....	10
Figura 10	Archivo consultorios.component.ts	10
Figura 11	Aplicación SGCMedic - Lista de consultorios	11
Figura 12	Aplicación SGCMedic - Visualizar datos de consultorio	11

1. INTRODUCCIÓN

Las aplicaciones web han tenido una evolución desde su primera implementación, de ser páginas estáticas a aplicaciones con una gran riqueza de contenido e interactividad. Al surgir las aplicaciones de una sola página (SPA), se permite que la página no se recarga nuevamente, sino que se actualiza la parte necesaria; y más recientemente aparecieron las aplicaciones web progresivas (PWA), que permiten acceder a recursos de hardware y software de la máquina como si fuera una aplicación nativa (Quinteros, 2018).

Estos surgimientos dan lugar a nuevos Frameworks que faciliten la creación de aplicaciones en la web. Uno de estos Framework es Angular, que sirve para generar aplicaciones web de Front-end, en particular aplicaciones de una sola página y aplicaciones web progresivas. Este Framework se especializa en mejorar la velocidad de dibujo de la página web en el browser, es decir la renderización, por lo que lo hace ideal para aplicaciones de Front-end orientadas al usuario final.

Para la creación del Front-end de la aplicación web se empleó el Framework Angular; esta aplicación busca gestionar las consultas médicas, su dominio es el servicio de atención ambulatoria a pacientes. Para esta actividad, y debido a la dimensión del proyecto, se consideró solo la gestión de citas médicas, que incluyen paciente, medico, usuarios, horario de atención, agenda y cita.

2. OBJETIVO

- Demostrar el conocimiento adquirido sobre el Framework Angular.

3. JUSTIFICACIÓN

Al desarrollar software, se toma la decisión de programar mediante la ayuda de Framework, el cual les aporta grandes beneficios en la aplicación a desarrollar. Para desarrollar aplicaciones web; crear una interfaz atractiva, fácil de usar y rápida, se opta por el Framework Angular, una excelente opción para

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

aplicaciones web del SPA, que proporciona al usuario una experiencia más fluida.

4. DEFINICIÓN DE LA ACTIVIDAD

En base al proyecto que se está desarrollando, el cual le hemos puesto del nombre de SGCMedic (Sistema de Gestión de Consultorios Médicos), se creó una aplicación web utilizando el Framework Angular. En la realización de dicha aplicación web se utiliza los siguientes conceptos estudiados en clase:

- Componentes.
- Routing y navegación.
- Directivas.
- Modelos.
- Paso de parámetros.
- Servicios y llamadas a API'S.
- Formularios dinámicos.

El sistema de gestión de consultorios médicos integra dos Framework (Laravel para el Back-End y Angular para el Front-End), en la siguiente imagen se visualiza dicha arquitectura donde se detalla con las operaciones necesarias para cada uno.

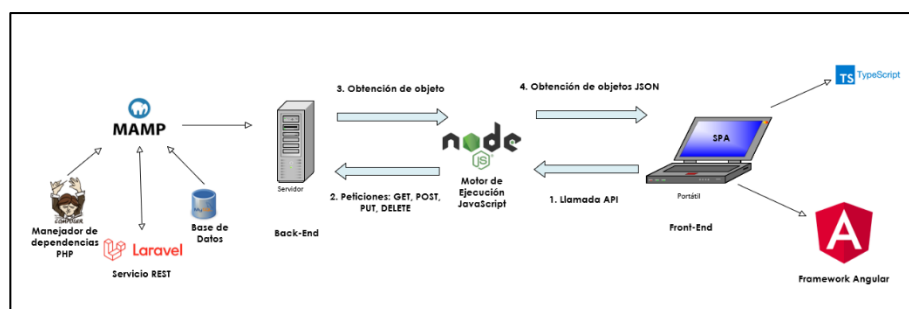


Figura 1 Arquitectura de la aplicación
Fuente: Elaboración propia en base a (Quinteros, 2018)

5. DEFINICIÓN DEL PROYECTO

El proyecto que se desarrolla es una aplicación web, para la gestión de consultorios médicos; su dominio es el servicio de atención ambulatoria a

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

pacientes. Se establece mediante módulos para amigable con el usuario. Los módulos que se consideraron son: médico (agenda, configuración de receta), paciente (historias clínicas, diagnóstico, documentos, medicamentos, citas), finanzas (cuentas por cobrar, cuentas por pagar, flujo de caja), centro médico (configuraciones, formatos, perfiles y más).

La aplicación web tiene tres tipos de usuarios: el usuario administrativo que gestiona accesos a los usuarios; los médicos que gestiona citas, historiales clínicos, cuentas, entre otros; y los pacientes que tendrán el papel de usuario para visualizar y gestionar citas médicas de acuerdo con la disponibilidad del médico.

Como se mencionó en la introducción, para esta actividad, se consideró solo la gestión de citas médicas, que incluyen paciente, medico, usuarios, horario de atención, agenda y cita. A continuación, se presenta el diagrama conceptual de la implementación en la API.

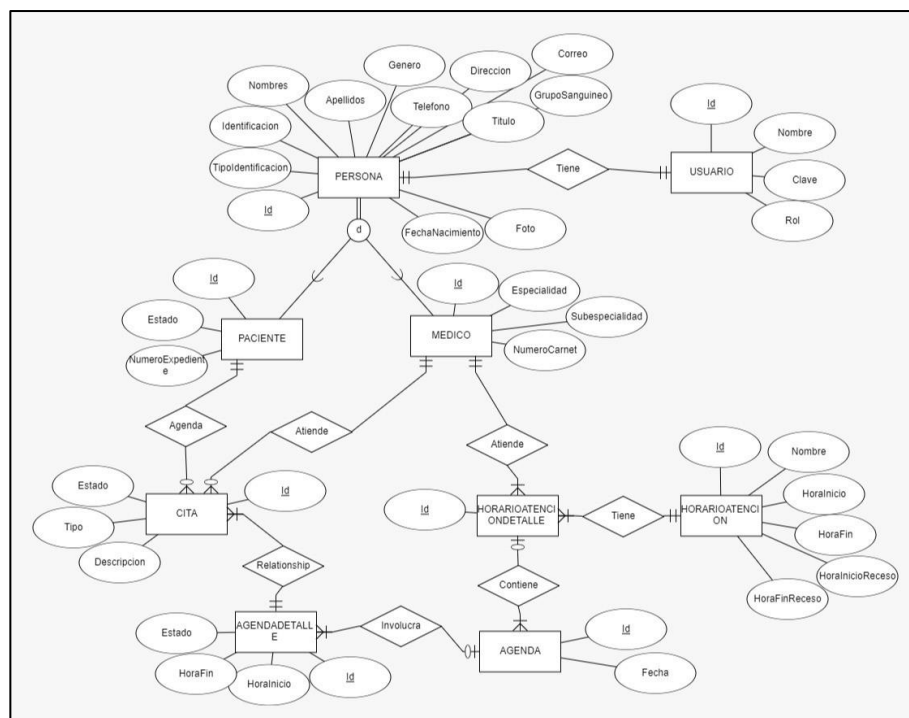


Figura 2 Diagrama conceptual de la implementación

Fuente: Elaboración propia en ERD Plus

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

La base de datos se desarrolló partiendo de los requisitos (funcionales o no funcionales) y características de calidad, a continuación, se redacta algunos de los requisitos de la aplicación web para la gestión de consultorios médicos:

- El usuario podrá registrar un nuevo paciente.
- El usuario podrá registrar una cita médica al paciente.
- El usuario (paciente) podrá visualizar la información de citas médicas.

Se realizaron modelos conceptuales que nos permitieron realizar razonamientos sobre el comportamiento del sistema con el usuario, mediante la representación de los procesos, como se indica en la siguiente figura.

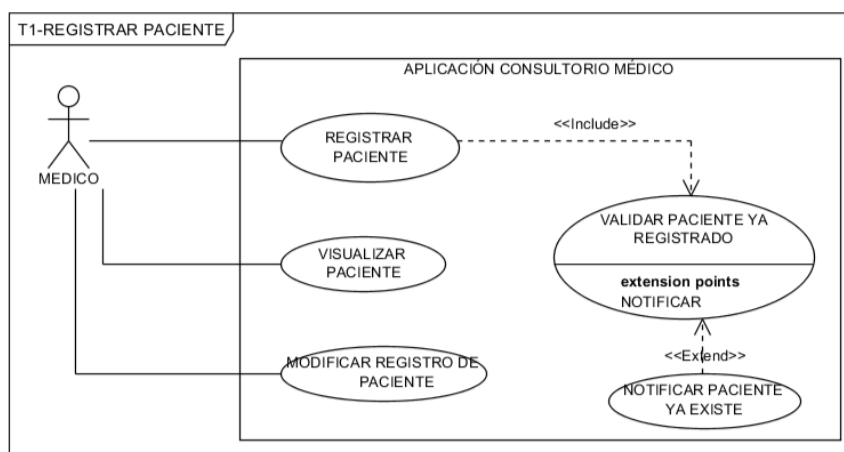


Figura 3 Diagrama de casos de uso: gestión de agenda
Fuente: Elaboración propia en Visual Paradigm

Después, analizando los requerimientos e identificado el propósito de la base de datos, se realizaron los diseños: El primer diseño es el conceptual, donde con entidades, relaciones; dan una visión detallada del negocio; como segundo diseño tenemos el desarrollo del esquema lógico, donde se normaliza, se definen las reglas de actualización y borrado; además de definir con claridad las claves primarias y ajenas; al final, se especifican las características físicas de la base de datos con el diseño físico.

6. DESARROLLO

Un paso muy importante para trabajar en Angular, así como en cualquier otro Framework orientado al desarrollo de aplicaciones, es la instalación de dicha

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

herramienta junto con sus requisitos previos, este paso es imprescindible para comenzar a trabajar en Angular.

Angular es una herramienta de JavaScript, por ellos requiere de Node.js, entorno en tiempo de ejecución, para la capa del servidor, el cual permite ejecutar JavaScript, lenguaje que utiliza Angular y es necesario para compilar y ejecutar los proyectos allí desarrollados, es por ello que se requiere la instalación previa para poder utilizar Angular (Boada & Gómez, 2019).

Ya instalado Node.js se puede hacer uso del instalador de paquetes npm, con ella se puede instalar Angular, una herramienta de línea de comando para generar proyectos listos para empezar, con ventajas como: desarrollar más rápido, añadir componentes y visualizar de forma instantánea la aplicación. Angular se instala con el siguiente comando.

```
npm install -g @angular/cli
```

Terminada esta instalación el equipo está configurado para comenzar a generar un nuevo proyecto y comenzar a desarrollar en él.

Para la creación de un proyecto en Angular primero es necesario iniciar Visual Studio Code y presionando Ctrl + ñ, se va a abrir la consola, para situarse en el directorio en el cual va a quedar guardado el proyecto, estando en dicha ruta se escribirá el comando:

```
ng new frontend
```

Seguido es procederá a situarse en la carpeta raíz del proyecto.

```
cd frontend
```

Allí se va a visualizar la estructura del proyecto en Angular, a continuación, una breve explicación de los archivos y directorios más importantes.

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

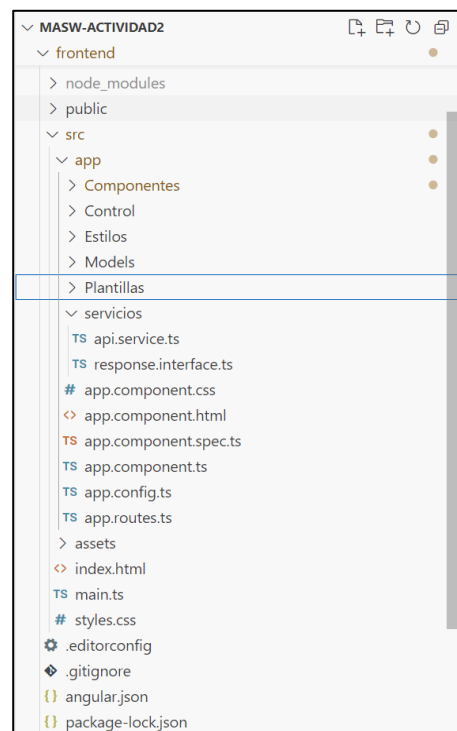


Figura 4 Estructura del proyecto

Fuente: Elaboración propia en Visual Studio Code

Se iniciará el servidor Node JS de desarrollo con el cual se va a ejecutar la aplicación en el navegador, con él se visualizarán los cambios de manera inmediata, estando situados en la carpeta raíz del proyecto se escribirá el siguiente comando:

```
ng serve
```

Se procederá a acceder a la siguiente dirección para visualizar la aplicación: <http://localhost:4200/>

Para la creación de los componentes se va a utilizar Angular, generando estos componentes de manera automática por medio del comando:

```
ng g component Componentes/consultorios
```

Generalmente, un componente suele llevar asociados diversos archivos con el siguiente cometido:

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

- consultorios.component.html: contiene el código HTML mediante el que fabricamos la pantalla (o Vista).
- consultorios.component.css: define los estilos que usaremos en la vista del componente.
- consultorios.component.ts: archivo que contiene la lógica del componente (Controlador) y que se escribe en TypeScript para ser traducido a JavaScript posteriormente (Boada & Gómez, 2019).

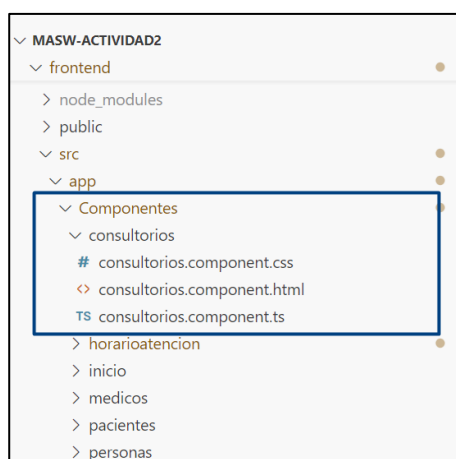


Figura 5 Archivos de componentes de consultorios

Fuente: Elaboración propia en Visual Studio Code

Gracias a Angular se pueden generar los componentes de forma automática el cual se encuentra conformado por bloques de código, estructura de un componente:

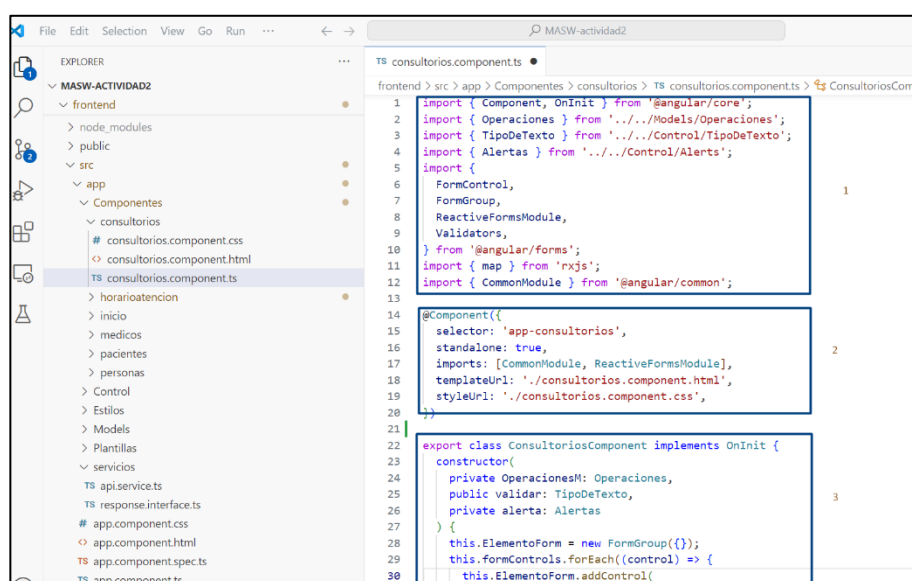


Figura 6 Estructura del componente consultorios

Fuente: Elaboración propia en Visual Studio Code

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

A continuación, se describe cada parte de la estructura de un componente:

1. Se describe el import del componente, para este ejemplo el núcleo (Core) de Angular.
2. Es un objeto de metadatos el cual consta de 3 definiciones: Selector: Nombre por el cual se invoca para llamar el componente, define en que etiqueta html es en la cual renderizara el componente. Template: define el código html, allí se indica la ruta del archivo template o plantilla. Referencia de estilo: Se define los estilos CSS, y aquí se indica la ruta del archivo.
3. La lógica del componente se definirá, por defecto se define solo el nombre.

Luego de trabajar en los componentes es importante conocer la manera de llamarlos y poder visualizarlos en el aplicativo web. Para que se incluya un componente en la página principal, sin necesidad de asignarle una ruta y que se va a visualizar siempre en las páginas, es necesario incluir estos componentes en el archivo: `consultorios.component.html`

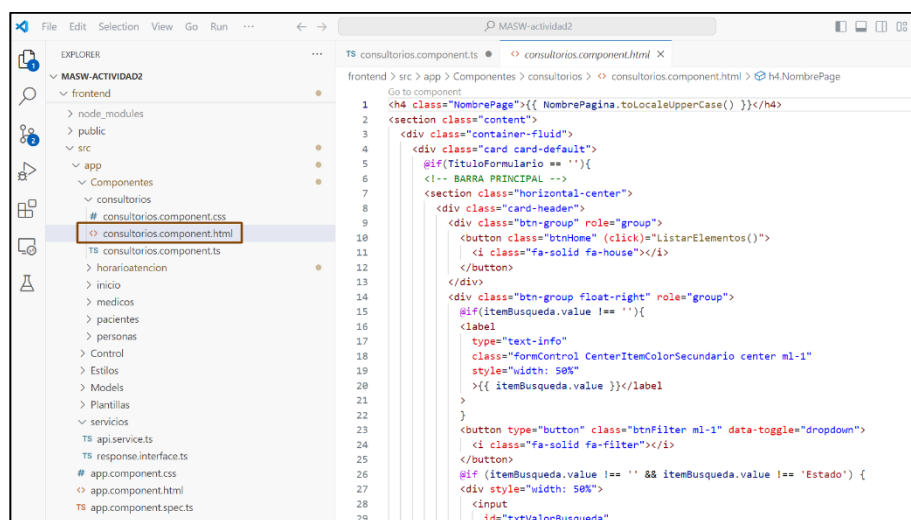


Figura 7 Archivo `consultorios.component.html`

Fuente: Elaboración propia en Visual Studio Code

Para la declaración de las rutas, se va a necesitar declararlas en el array: Routes, en el cual se le asigna un nombre de la ruta y el componente a mostrar en dicha ruta, de la siguiente manera.

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

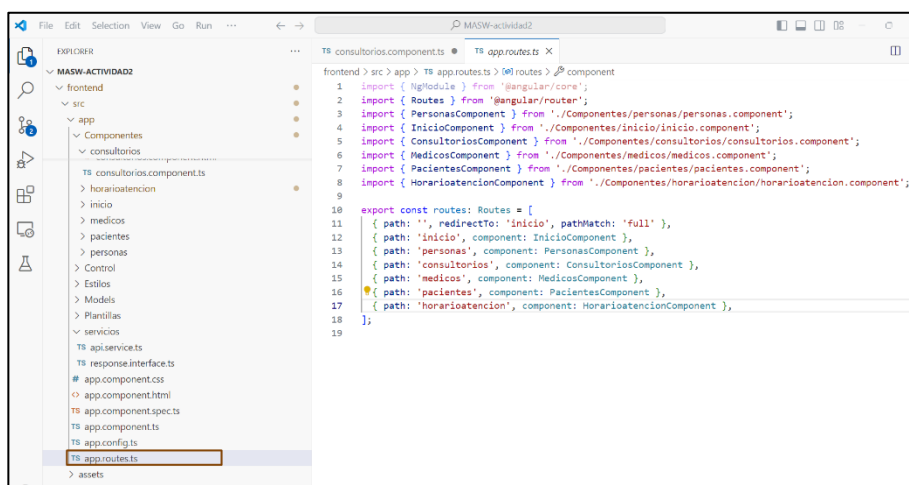


Figura 8 Archivo app.routes.ts

Fuente: Elaboración propia en Visual Studio Code

Si no se genera la importación por sí sola de los componentes es necesario realizar la importación de los componentes, en los imports de manera manual. Con ello ya se podrá visualizar el contenido de estos dos componentes en la dirección:

<http://localhost:4200/consultorios>

Un servicio en Angular es una clase con una funcionalidad específica que se utiliza para compartir datos, lógica de negocio y funcionalidades entre diferentes componentes de una aplicación. El propósito principal de un servicio es proporcionar una capa de abstracción y modularidad, permitiendo la reutilización del código y facilitando la separación de preocupaciones (Imagina, 2024).

Angular nos proporciona una forma sencilla de generar un nuevo servicio. En el terminal se ejecuta el siguiente comando:

`ng generate service servicios/api`

Esto creará un nuevo archivo llamado `api.service.ts` en el directorio `src/app/servicios` de tu proyecto Angular, con la estructura básica de un servicio.

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

El archivo `api.service.ts` contendrá una clase TypeScript que representa el servicio. A continuación, se muestra la estructura y funcionalidad de una petición GET a la API para obtener la lista de consultorios:

```

1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { Consultorio, Response, ResponseInterface } from './response.interface';
4 import { catchError, map, Observable } from 'rxjs';
5 import { Alertas } from '../Control/Alertas';
6
7 @Injectable({
8   providedIn: 'root',
9 })
10 export class ApiService {
11   constructor(private alerta: Alertas, private http: HttpClient) {}
12
13   url: string = 'http://127.0.0.1:8080/api/';
14
15   // CONSULTORIO
16   get(entidad:string, codigo: number, rango: number): Observable<any> {
17     let direccion = this.url + entidad + codigo + ',' + rango;
18     return this.http.get<any>(direccion).pipe(
19       map((data) => {
20         console.log(data);
21         return data;
22       }),
23       catchError((error) => {
24         this.alerta.ErrorAlRecuperarElementosError(
25           'Error al Conectar con el servidor',
26           error
27         );
28         throw error;
29       })
30     );
31
32     getEntidad(entidad:string, tipo: number, valor: string): Observable<any> {
33       let direccion = this.url + entidad + tipo + ',' + valor;
34       return this.http.get<any>(direccion).pipe(
35         map((data) => {
36           console.log(data);
37           return data;
38         })
39       );
40     }
41   }
42 }

```

Figura 9 Archivo `api.service.ts`

Fuente: Elaboración propia en Visual Studio Code

En el archivo `consultorios.component.ts`, establecemos que al cargar la página haga la petición a la API para obtener la lista de consultorios como se muestra a continuación:

```

19 styleUrls: ['./consultorios.component.css'],
20 })
21 export class ConsultoriosComponent implements OnInit {
22   constructor(
23     private OperacionesM: Operaciones,
24     public validar: TipoDeTexto,
25     private alerta: Alertas
26   ) {}
27   this.ElementoForm = new FormGroup({});
28   this.formControls.forEach((control) => {
29     this.ElementoForm.addControl(
30       control.name,
31       new FormControl(control.value || '', control.validators)
32     );
33   });
34
35   ngOnInit(): void {
36     this.ListarElementos();
37   }
38
39   NombrePagina: string = 'Consultorio';
40   TituloFormulario: string = '';
41   ParametrosDeBusqueda: Array<string> = [
42     'Ruc',
43     'Nombre Completo',
44     'Nombre Comercial Completo',
45     'Nombre Incompleto',
46     'Nombre Comercial Incompleto',
47     'Estado',
48   ];
49
50   ParametrosEstado: any[] = [
51     { name: 'Activo', value: 'Activo' },
52     { name: 'Inactivo', value: 'Inactivo' },
53     // { name: 'Eliminados', value: 3 },
54   ];
55 }

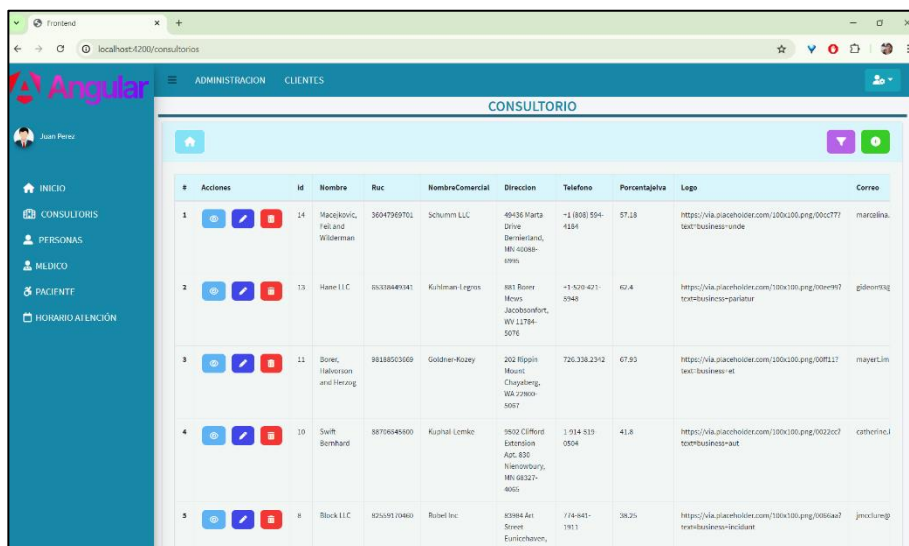
```

Figura 10 Archivo `consultorios.component.ts`

Fuente: Elaboración propia en Visual Studio Code

Máster Universitario en Desarrollo de Aplicaciones y Servicios Web

Si, vamos al navegador a la ruta: <http://localhost:4200/consultorios> se podrá visualizar la lista de consultorios obtenidas por medio de la petición a la API de Laravel:























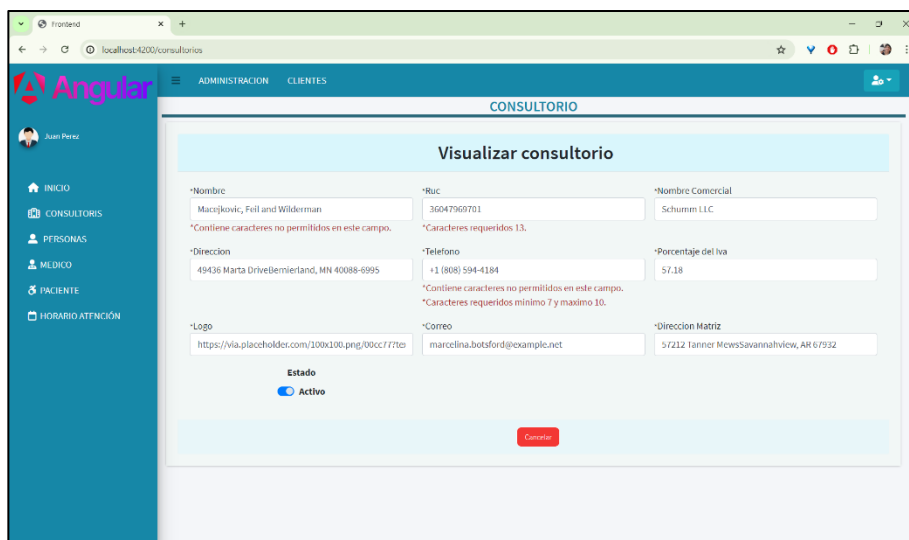
#	Acciones	Id	Nombre	Ruc	NombreComercial	Direccion	Telefono	PorcentajeIva	Logo	Correo
1	  	14	Mackovic, Feil and Wilderman	36047960701	Schumm LLC	49436 Marta Drive Denslerland, MN 40088-6995	+1 (808) 594-4184	57.18		marcelina.botsford@example.net
2	  	13	Rane LLC	65338449341	Kuhman Legros	881 Borer Mours Jacobsonfort, WV 11784-5076	+1 520-427-0240	62.4		gideonm@trembusbusinesspartners
3	  	11	Borer, Halverson and Herzog	98188503669	Goldner-Kozey	202 Hilpin Mount Chaydenberg, WA 22969-5067	720-338-2312	67.93		mayerlin@trembusbusinesspartners
4	  	10	Swift-Bernhard	88796045900	Kuphal Lemke	3829 Clifford Extension Apt. 830 Hienowbury, MN 68227-4055	1 914-819-0804	41.8		catherine@trembusbusinesspartners
5	  	8	Black LLC	82509510460	Buier Inc.	83984 Ari Sweet Eunithaven, BL 67436	714-845-1911	38.25		jmcclure@trembusbusinesspartners

Figura 11 Aplicación SGCMedic - Lista de consultorios
Fuente: Elaboración propia



Visualizar consultorio

*Nombre: Mackovic, Feil and Wilderman


*Ruc: 36047960701

*Nombre Comercial: Schumm LLC

*Direccion: 49436 Marta DriveDenslerland, MN 40088-6995

*Telefono: +1 (808) 594-4184

*Porcentaje del Iva: 57.18

*Logo: 

*Correo: marcelina.botsford@example.net

*Direccion Matriz: 57212 Tanner MewsSavannahview, AR 67932

Estado: ☒ Activo

Figura 12 Aplicación SGCMedic - Visualizar datos de consultorio
Fuente: Elaboración propia

CONCLUSIONES

Los resultados obtenidos en el desarrollo de esta actividad nos permitieron llegar a las siguientes conclusiones:

Angular es un marco de diseño que permite crear aplicaciones eficientes y sofisticadas de una sola página; durante el desarrollo los cambios que se realizaban se visualizaban en el proceso, lo que nos permitió ver los errores y corregirlos al momento.

El beneficio de integrar los dos Framework Angular y Laravel es que nos permite a los desarrolladores optar por nuevas herramientas de desarrollo, implementando diferentes formas de aplicar la lógica de negocio y consumo de servicios desde otras plataformas de multilenguaje de programación.

REFERENCIAS

- Boada, M., & Gómez, J. A. (01 de Enero de 2019). El gran libro de Angular. México D.F.
- Imagina. (15 de Julio de 2024). Cómo integrar servicios y llamadas a API en Angular. Recuperado el 21 de Julio de 2024, de <https://imaginaformacion.com/tutoriales/servicios-llamadas-api-angular>
- Quinteros, A. R. (Noviembre de 2018). Estudio de integración de los Frameworks Angular 4 y Yii2, orientado a servicios REST, que permitan la gestión y control de inventarios para mejorar la productividad en la empresa Induxion. Ibarra, Ecuador.