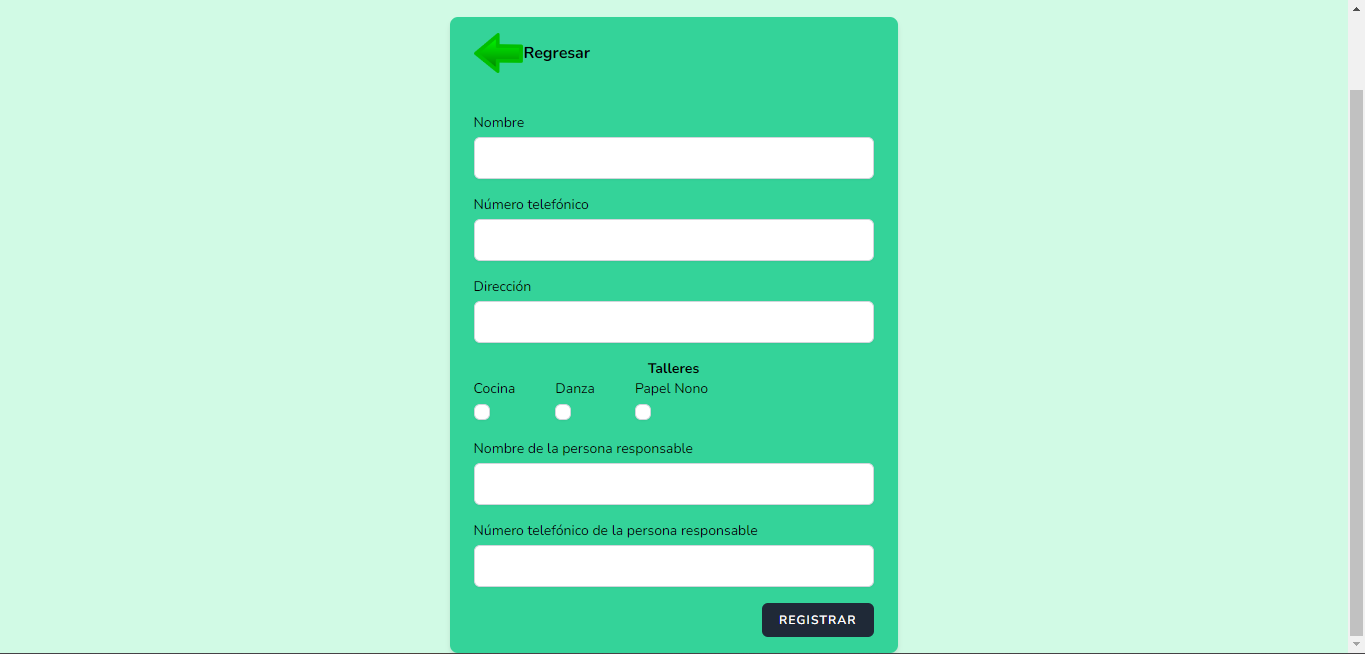
# Codificación de la interfaz de registro de usuarios

Para implementar el registro de adultos mayores en la aplicación web, se utiliza la biblioteca Jetstream para el proceso de registro. Durante el registro, se recopilan los siguientes datos:

* Número telefónico
* Dirección
* Talleres: Cocina, Danza y Papel Nono
* Nombre de la persona responsable
* Número telefónico de la persona responsable



Ahora bien, se creó con Fotify el archivo **CreateNewUser.php** en donde se implementa la validación de los campos.

Algunas de las reglas de validación incluyen:

* El campo name debe ser requerido, una cadena de texto con un máximo de 255 caracteres y solo debe contener letras y espacios.
* El campo email es opcional, pero si se proporciona, debe ser una dirección de correo electrónico válida con un máximo de 255 caracteres.
* Los campos number\_phoneViejito y number\_phoneTutor son requeridos y deben ser números con exactamente 10 dígitos.
* El campo name\_tutor debe ser requerido y solo debe contener letras y espacios.
* Varios campos relacionados con talleres y servicios son opcionales, y se utiliza una regla personalizada AtLeastOneTallerSelected para asegurarse de que al menos uno de los talleres esté seleccionado.

Después de definir las reglas de validación, se llama al método validate en el objeto $validator para asegurarse de que los datos cumplan con estas reglas. Si la validación falla, se generará una excepción.

Luego, se establece manualmente el valor del campo role\_id en 2, lo que sugiere que el usuario registrado tendrá un rol específico en tu aplicación.

Finalmente, se crea una nueva instancia de la clase User y se le asignan los valores de los campos del usuario a partir de los datos proporcionados en $input. Esto crea un nuevo usuario en la base de datos.

<?php

namespace App\Actions\Fortify;

use App\Models\User;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator;

use Laravel\Fortify\Contracts\CreatesNewUsers;

use Laravel\Jetstream\Jetstream;

use App\Rules\AtLeastOneTallerSelected;

use Illuminate\Validation\Rule;

class CreateNewUser implements CreatesNewUsers

{

    use PasswordValidationRules;

*/\*\**

*\* Validate and create a newly registered user.*

*\**

*\* @param  array  $input*

*\* @return \App\Models\User*

*\*/*

    public function create(array $input)

    {

        $validator = Validator::make($input, [

            'name' => ['required', 'string', 'max:255', 'regex:/^[A-Za-z\s]+$/'], *// Solo letras y espacios*

            'email' => ['nullable', 'string', 'email', 'max:255'],

            'number\_phoneViejito' => ['required', 'numeric', 'digits:10', Rule::unique('users', 'number\_phoneViejito')],

            'number\_phoneTutor' => ['required', 'numeric', 'digits:10'],

            'viejito\_address' => ['required'],

            'name\_tutor' => ['required', 'regex:/^[A-Za-z\s]+$/'], *// Solo letras y espacios*

            'taller1' => ['nullable'],  *// Se espera un valor booleano (true/false) para los talleres*

            'taller2' => ['nullable'],

            'taller3' => ['nullable'],

            'servicio'  => ['nullable'],

            'talleres' => [new AtLeastOneTallerSelected], *// Agrega la regla personalizada*

        ]);

        $validator->validate();

*// Establece manualmente el valor del rol*

        $input['role\_id'] = 2;

        return User::create([

            'name' => $input['name'],

            'email' => $input['email'] ?? null,

            'role\_id' => $input['role\_id'],

            'number\_phoneViejito' => $input['number\_phoneViejito'] ?? null,

            'viejito\_address' => $input['viejito\_address'] ?? null,

            'name\_tutor' => $input['name\_tutor'] ?? null,

            'number\_phoneTutor' => $input['number\_phoneTutor'] ?? null,

            'taller1' => $input['taller1'] ?? null,

            'taller2' => $input['taller2'] ?? null,

            'taller3' => $input['taller3'] ?? null,

            'servicio' => $input['servicio'] ?? null,

        ]);

    }

}

El archivo **CreateNewUser.php** es responsable de validar los datos del formulario de registro de usuarios y crear un nuevo usuario en la base de datos de acuerdo con las reglas de validación especificadas. Además, establece manualmente el rol del usuario y utiliza las reglas personalizadas para validar los talleres seleccionados.

# Codificación para el inicio de sesión del Personal Administrativo

Para la implementación del inicio de sesión para el administrador y los tres doctores en donde se piden los siguientes datos para validar los campos que ingresa con los que están en la base de datos:

* Correo electrónico
* Contraseña



Para esto se creó el controlador **LoginRedirectController.php** de manera invocable para validar que usuario está iniciando sesión. En donde verifica si el usuario que se va a autenticar tiene el rol\_id=1 que es el rol del Administrador lo va a redirigir a admin/users, pero si el usuario tiene el rol\_id=3 que es el rol de los Doctores lo va a redirigir a doctor/servicios

<?php

namespace App\Http\Controllers\Auth;

use Illuminate\Http\Request;

class LoginRedirectController

{

    public function \_\_invoke()

    {

        if (auth()->user()->role\_id == 1) {

            return redirect('admin/users');

        }

        if (auth()->user()->role\_id == 3) {

            return redirect('doctor/servicios');

        }

    }

}

Posteriormente se creó otro controlador, pero solo para el Administrador en donde lo va a redirigir al apartado de servicios que se encuentra en la vista de admin/users/index.blade.php.

<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;

class UserController extends Controller

{

    public function index()

    {

        return view('admin.users.index');

    }

}

Ahora se crea el otro controlador, pero solo para el rol de Doctores en donde lo va a redirigir al apartado de servicio asignados que se encuentra en la vista de doctor/servicios/index.blade.php

<?php

namespace App\Http\Controllers\Doctores;

use App\Http\Controllers\Controller;

use Illuminate\Support\Facades\Gate;

class ServiciosController extends Controller

*// CourseController.php*

{

    public function index()

    {

        if (Gate::denies('manage-servicios')) {

            abort(403);

        }

        return view('doctor.servicios.index');

    }

}

Ahora en el archivo **web.php** se definen las rutas que utilizaran cada rol para iniciar sesión tanto para el Administrador, Doctores y Adultos mayores.

Route::group(['middleware' => 'auth'], function () {

    Route::group(['middleware' => 'role:adulto\_mayor', 'prefix' => 'adulto\_mayor', 'as' => 'adulto\_mayor.'], function () {

        Route::resource('lessons', \App\Http\Controllers\Adulto\_mayor\LessonController::class);

    });

    Route::group(['middleware' => 'role:doctor', 'prefix' => 'doctor', 'as' => 'doctor.'], function () {

        Route::resource('servicios', \App\Http\Controllers\Doctores\ServiciosController::class);

    });

    Route::group(['middleware' => 'role:admin', 'prefix' => 'admin', 'as' => 'admin.'], function () {

        Route::resource('users', \App\Http\Controllers\Admin\UserController::class);

    });

});

# Codificación para el inicio de sesión de los Adultos mayores

Para implementar el inicio de sesión para los adultos mayores se requiere únicamente el Número telefónico para validar el número que ingresa con los que está en la base de datos.



Se creo el controlador para el Adulto mayor en donde lo va a redirigir al apartado de Actualizar mis datos que se encuentra en la vista de adulto\_mayor/lessons/index.blade.php.

<?php

namespace App\Http\Controllers\Adulto\_mayor;

use App\Models\User;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use Illuminate\Support\Facades\Auth;

use Laravel\Fortify\Http\Controllers\AuthenticatedSessionController;

class LessonController extends AuthenticatedSessionController

{

    public function index()

    {

        return view('adulto\_mayor.lessons.index');

    }

}

Posteriormente se crea otro controlador para el adulto mayor que se encarga de autenticar a los adultos mayores en la aplicación utilizando su número de teléfono como credencial. Si el número de teléfono se encuentra en la base de datos, el usuario se autentica y se redirige a la página de lecciones. En caso contrario, se muestra un mensaje de error y se redirige de nuevo al formulario de inicio de sesión.

<?php

namespace App\Http\Controllers\AdultoMayor;

use App\Http\Controllers\Controller;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use App\Models\User;

class LoginController extends Controller

{

    public function create()

    {

        return view('auth.login-adulto-mayor');

    }

    public function store(Request $request)

    {

        $request->validate([

            'number\_phoneViejito' => 'required', *// Asegúrate de que la validación esté configurada correctamente*

        ]);

        $numero = $request->input('number\_phoneViejito');

        $user = User::where('number\_phoneViejito', $numero)->first();

        if ($user) {

*// Autenticar al usuario*

            Auth::login($user);

            return redirect('/adulto\_mayor/lessons'); *// Redirige a /adulto\_mayor/lessons si la autenticación es exitosa*

        } else {

            return back()->withErrors(['number\_phoneViejito' => 'El número de teléfono no es válido.']); *// Maneja el caso de autenticación fallida*

        }

    }

}

Ahora en el archivo **web.php** se crea la ruta para el inicio de sesión de los adultos mayores

*// rutas para el adulto mayor*

Route::group(['middleware' => 'guest'], function () {

    Route::get('/login/adulto-mayor', [LoginController::class, 'create'])->name('login.adulto-mayor');

    Route::post('/login/adulto-mayor', [LoginController::class, 'store']);

});

# Codificar sección de talleres

Este apartado se encuentra en la pantalla principal donde se va a mostrar varios datos incluyendo la sección de talleres que ofrece el Centro Gerontológico, por lo cual se creó el archivo **welcome.blade.php** donde se presenta una imagen referente al taller, el título, una pequeña descripción de cada taller y los días de la semana en la que ofrecen el taller.



<h2 class="mt-4 text-center text-3xl font-bold text-gray-900 dark:text-white">Talleres</h2>

                <div class="mt-4 grid grid-cols-1 sm:grid-cols-1 md:grid-cols-3 lg:grid-cols-3 xl:grid-cols-3 gap-5">

*<!--Card 1-->*

                    <div class="rounded overflow-hidden shadow-lg">

                        <img class="w-full" src="{{ asset('storage/img/cocina.jpg') }}" alt="Cocina"/>

                        <div class="px-6 py-4">

                            <div class="font-bold text-xl mb-2">Cocina</div>

                            <p class="text-gray-700 text-base text-justify">

                                Este taller está pensado para que los residentes puedan explorar su creatividad culinaria y compartir recetas tradicionales.

                                Bajo la guía de un experto, aprenderán técnicas básicas de cocina y repostería para preparar deliciosas recetas.

                                La actividad promueve la motricidad, la paciencia y la socialización en un ambiente divertido.

                            </p>

                        </div>

                        <div class="px-6 pt-4 pb-2">

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Jueves</span>

                        </div>

                        </div>

*<!--Card 2-->*

                        <div class="rounded overflow-hidden shadow-lg">

                        <img class="w-full" src="{{ asset('storage/img/papel-nono2.png') }}" alt="Papel nono"/>

                        <div class="px-6 py-4">

                            <div class="font-bold text-xl mb-2">Papel nono</div>

                            <p class="text-gray-700 text-base text-justify">

                                Es un taller destinado a que los adultos mayores participen en la creación de figuras de papel.

                                Durante estas clases, se les brinda la oportunidad de expresar su creatividad y habilidades artísticas a

                                través de la elaboración de diversas figuras y formas de papel. Este taller fomenta la destreza manual,

                                la concentración y la satisfacción personal al completar proyectos de papel, proporcionando a los adultos

                                mayores una actividad enriquecedora y entretenida.

                            </p>

                        </div>

                        <div class="px-6 pt-4 pb-2">

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Lunes</span>

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Miércoles</span>

                        </div>

                    </div>

*<!--Card 3-->*

                    <div class="rounded overflow-hidden shadow-lg">

                        <img class="w-full" src="{{ asset('storage/img/danza.webp') }}" alt="Danza"/>

                        <div class="px-6 py-4">

                            <div class="font-bold text-xl mb-2">Danza</div>

                            <p class="text-gray-700 text-base text-justify">

                                En este innovador taller los residentes aprenderán a crear originales figuras y elementos decorativos utilizando papel maché.

                                Guiados por un instructor, conocerán las técnicas para moldear y dar forma al papel, pintarlo y crear piezas artísticas de forma manual.

                                La actividad promueve la motricidad fina, la paciencia, la imaginación y la satisfacción de concretar una obra.

                            </p>

                        </div>

                        <div class="px-6 pt-4 pb-2">

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Martes</span>

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Viernes</span>

                        </div>

                    </div>

                </div>

# Codificar sección de servicios

De igual forma el apartado de Servicios que ofrece el Centro Gerontológico se encuentra en el archivo **welcome.blade.php,** en donde se presenta una imagen referente al servicio, el título, una pequeña descripción y el responsable de cada servicio.



<h2 class="mt-4 text-center text-3xl font-bold text-gray-900 dark:text-white">Servicios</h2>

                <div class="mt-4 grid grid-cols-1 sm:grid-cols-1 md:grid-cols-3 lg:grid-cols-3 xl:grid-cols-3 gap-5">

*<!--Card 1-->*

                    <div class="rounded overflow-hidden shadow-lg">

                        <img class="w-full" src="{{ asset('storage/img/atencion\_medica.png') }}" alt="Atención Medica"/>

                        <div class="px-6 py-4">

                            <div class="font-bold text-xl mb-2">Atención medica</div>

                            <p class="text-gray-700 text-base text-justify">

                                Contamos con un equipo de médicos y enfermeros altamente capacitados para brindar atención integral a los adultos mayores.

                                Se realizan chequeos periódicos de salud, seguimiento de enfermedades crónicas, suministro de medicamentos y cualquier otro tratamiento

                                que requiera el residente. Nuestro objetivo es mantener el bienestar físico de cada persona.

                            </p>

                        </div>

                        <div class="px-6 pt-4 pb-2">

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Responsable: Ibeth Hernández</span>

                        </div>

                        </div>

*<!--Card 2-->*

                        <div class="rounded overflow-hidden shadow-lg">

                        <img class="w-full" src="{{ asset('storage/img/fisioterapia.jpeg') }}" alt="Fisioterapia"/>

                        <div class="px-6 py-4">

                            <div class="font-bold text-xl mb-2">Fisioterapia</div>

                            <p class="text-gray-700 text-base text-justify">

                                Ofrecemos servicios de fisioterapia y rehabilitación para mantener y mejorar la movilidad y calidad de vida de los residentes.

                                Contamos con fisioterapeutas especializados en geriatría que realizan ejercicios y terapias personalizadas,

                                acordes a las necesidades de cada adulto mayor. Buscamos que cada persona conserve su mayor grado de independencia posible.

                            </p>

                        </div>

                        <div class="px-6 pt-4 pb-2">

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Responsable: Ismael Valencia</span>

                        </div>

                    </div>

*<!--Card 3-->*

                    <div class="rounded overflow-hidden shadow-lg">

                        <img class="w-full" src="{{ asset('storage/img/psicologia.jpg') }}" alt="Psicología"/>

                        <div class="px-6 py-4">

                            <div class="font-bold text-xl mb-2">Psicología</div>

                            <p class="text-gray-700 text-base text-justify">

                                Entendemos la importancia del bienestar mental y emocional en la tercera edad.

                                Nuestro equipo de psicólogos brinda apoyo psicológico individual o grupal,

                                para ayudar a los residentes a tener una vejez tranquila y feliz. Ofrecemos terapias y actividades para estimular

                                las facultades cognitivas, prevenir la depresión y mejorar la autoestima.

                            </p>

                        </div>

                        <div class="px-6 pt-4 pb-2">

                            <span class="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">Responsable: Dulce María</span>

                        </div>

                    </div>

                </div>

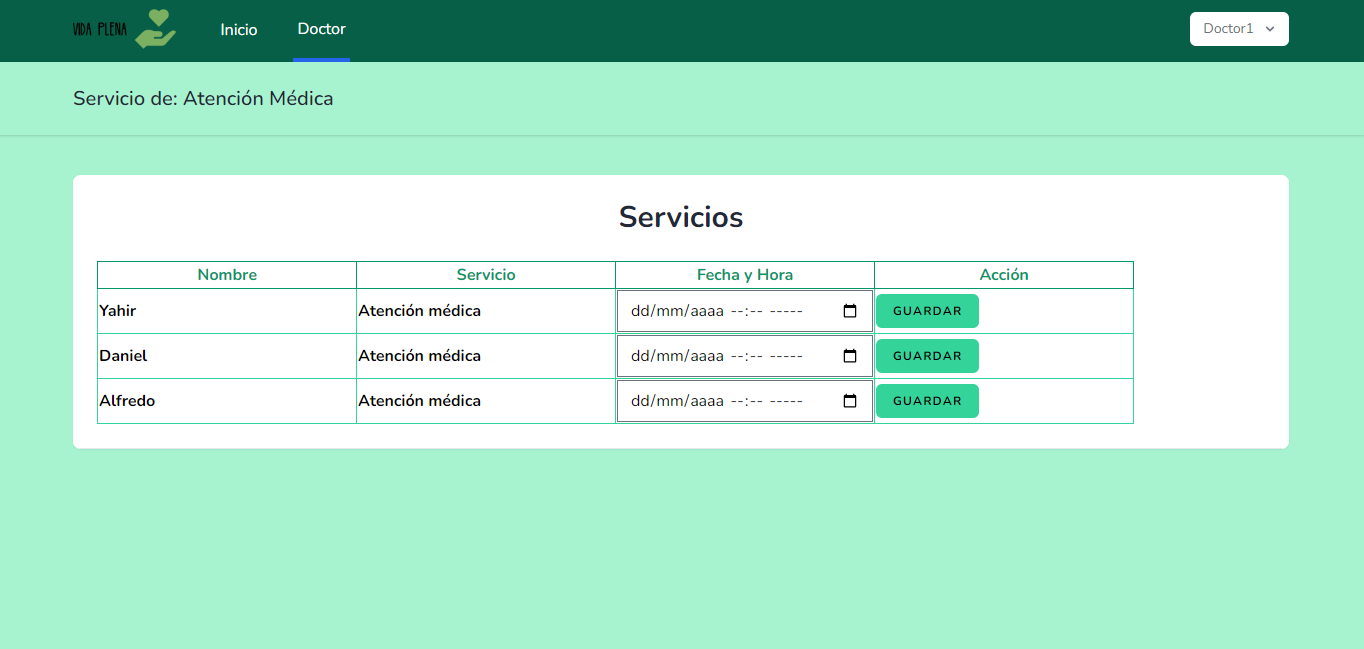
# Codificación de la interfaz para doctores

Para la interfaz de los doctores hay que tener en cuenta que son 3 doctores para cada servicio: Atención médica, Fisioterapia y Psicología. Por lo que utilice la librería de Livewire para cada interfaz de los doctores.

## Doctor a cargo de Atención médica

Para el primer doctor se creó los componentes tanto la vista como el controlador con Livewire, la vista es el archivo **doctor-servicios.blade.php** que muestra una tabla la cual muestra lo siguiente:

* **Nombre** (del adulto mayor)
* **Servicio** (en la que está el adulto mayor)
* **Fecha y Hora** (registrado por el doctor)
* **Acción** (botón para guardar los datos de cada fila)



Ahora en el controlador **DoctorServicios.php** es el componente Livewire que gestiona la programación del servicio “Atención Médica” en una aplicación. Realiza las siguientes acciones:

* Recopila y muestra la lista de doctores (usuarios con un rol específico) en la variable $doctores.
* Permite a los usuarios asignar y guardar fechas y horas para servicios médicos mediante el método guardarFechaHora. También gestiona mensajes de éxito y error.
* En el método render, filtra y obtiene una lista de doctores específicos (en este caso, el Doctor 1 con ID 10) y los servicios registrados relacionados con ellos, que luego se muestran en la vista correspondiente.

En resumen, este archivo permite la gestión de fechas y horas para servicios médicos proporcionados por el doctor de “Atención Médica”.

<?php

namespace App\Http\Livewire;

use Livewire\Component;

use App\Models\User;

use App\Models\ServicioRegistrado;

class DoctorServicios extends Component

{

    public $doctores;

    public $fechaHoraServicio = [];

    public function guardarFechaHora($servicioRegistradoId)

    {

        $servicioRegistrado = ServicioRegistrado::find($servicioRegistradoId);

        if ($servicioRegistrado) {

            if ($servicioRegistrado->fecha\_hora\_servicio) {

                session()->flash('messageError', 'Ya se ha guardado una fecha y hora para este servicio.');

                return;

            }

            $servicioRegistrado->fecha\_hora\_servicio = $this->fechaHoraServicio[$servicioRegistradoId];

            $servicioRegistrado->save();

            session()->flash('message', 'Fecha y hora guardadas correctamente.');

        }

    }

    public function render()

    {

*// Filtra los registros de servicios registrados relacionados con el Doctor 1 (ID 10)*

        $this->doctores = User::where('role\_id', 2)

            ->whereHas('serviciosRegistrados', function ($query) {

                $query->where('doctor\_id', 10); *// Ajusta el ID del Doctor 1 según tu estructura de base de datos*

            })

            ->get();

        return view('livewire.doctor-servicios');

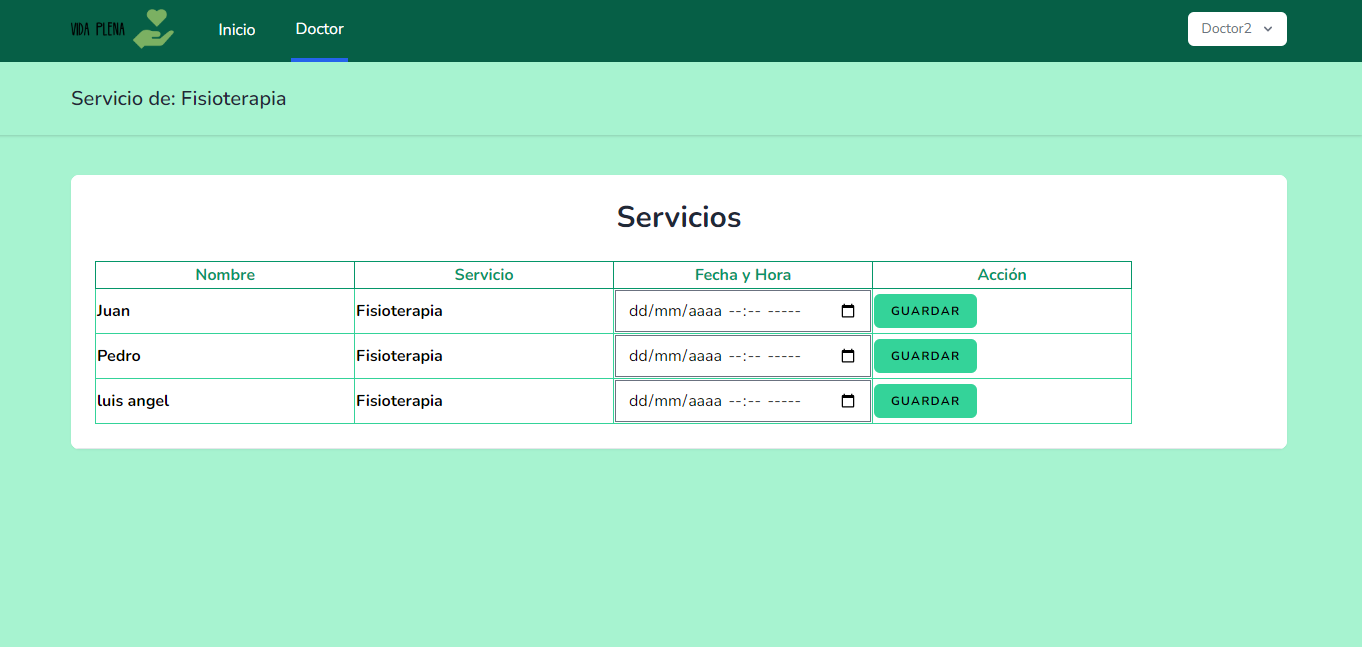
    }

}

## Doctor a cargo de Fisioterapia

Para el segundo doctor se creó la vista **doctor2-servicios.blade.php** que muestra una tabla la cual muestra lo siguiente:

* **Nombre** (del adulto mayor)
* **Servicio** (en la que está el adulto mayor)
* **Fecha y Hora** (registrado por el doctor)
* **Acción** (botón para guardar los datos de cada fila)



Ahora en el controlador **Doctor2Servicios.php** es el componente Livewire que gestiona la programación del servicio “Fisioterapia” en una aplicación. Realiza las siguientes acciones:

* Recopila y muestra la lista de doctores (usuarios con un rol específico) en la variable $doctores.
* Permite a los usuarios asignar y guardar fechas y horas para servicios médicos mediante el método guardarFechaHora. También gestiona mensajes de éxito y error.
* En el método render, filtra y obtiene una lista de doctores específicos (en este caso, el Doctor 2 con ID 5) y los servicios registrados relacionados con ellos, que luego se muestran en la vista correspondiente.

En resumen, este archivo permite la gestión de fechas y horas para servicios médicos proporcionados por el doctor de “Fisioterapia”.

<?php

namespace App\Http\Livewire;

use Livewire\Component;

use App\Models\User;

use App\Models\ServicioRegistrado;

class Doctor2Servicios extends Component

{

    public $doctores;

    public $fechaHoraServicio = [];

    public function guardarFechaHora($servicioRegistradoId)

    {

        $servicioRegistrado = ServicioRegistrado::find($servicioRegistradoId);

        if ($servicioRegistrado) {

            if ($servicioRegistrado->fecha\_hora\_servicio) {

                session()->flash('messageError', 'Ya se ha guardado una fecha y hora para este servicio.');

                return;

            }

            $servicioRegistrado->fecha\_hora\_servicio = $this->fechaHoraServicio[$servicioRegistradoId];

            $servicioRegistrado->save();

            session()->flash('message', 'Fecha y hora guardadas correctamente.');

        }

    }

    public function render()

    {

*// Filtra los registros de servicios registrados relacionados con el Doctor 1 (ID 10)*

        $this->doctores = User::where('role\_id', 2)

            ->whereHas('serviciosRegistrados', function ($query) {

                $query->where('doctor\_id', 5); *// Ajusta el ID del Doctor 1 según tu estructura de base de datos*

            })

            ->get();

        return view('livewire.doctor2-servicios');

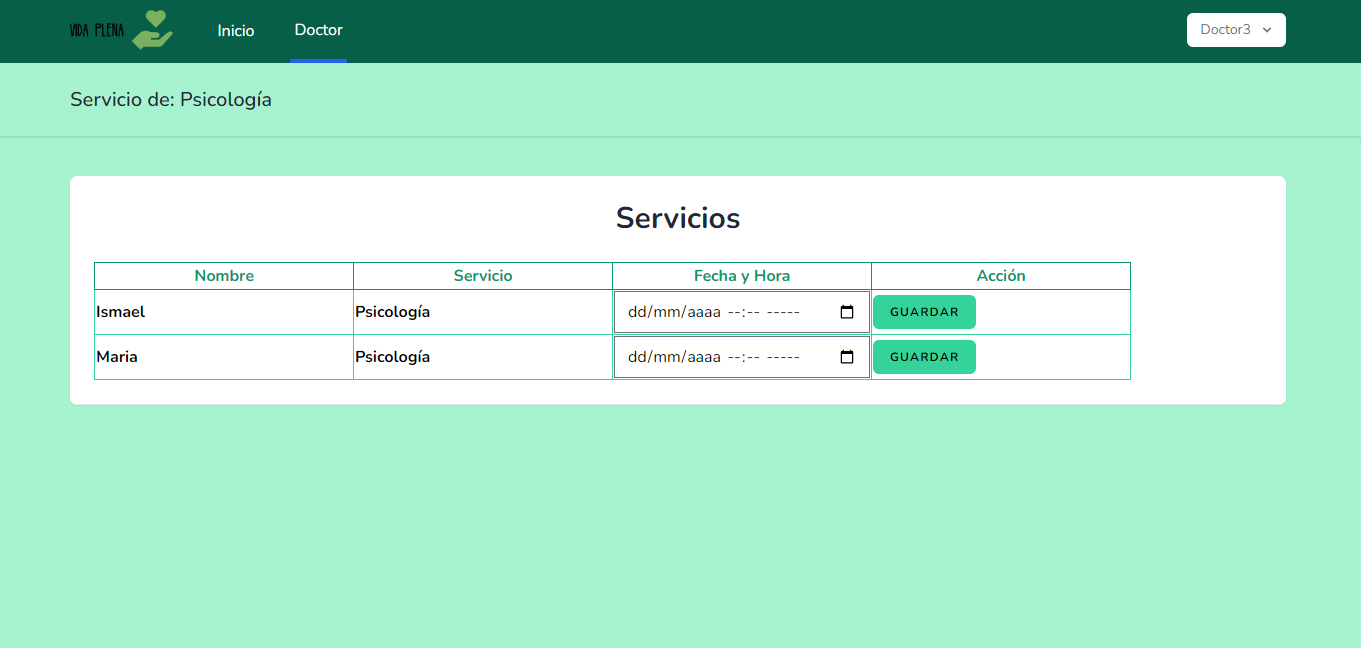
    }

}

## Doctor a cargo de Psicología

Para el tercer doctor se creó la vista **doctor3-servicios.blade.php** que muestra una tabla la cual muestra lo siguiente:

* **Nombre** (del adulto mayor)
* **Servicio** (en la que está el adulto mayor)
* **Fecha y Hora** (registrado por el doctor)
* **Acción** (botón para guardar los datos de cada fila)



En los 3 componentes para las vistas creadas por la librería de Livewire utilizan la misma vista para renderizar los elementos que puede visualizar cada doctor.

<div>

    <h2 class="font-bold text-3xl text-gray-800 leading-tight text-center mb-6">

        Servicios

    </h2>

    @if (session()->has('message'))

        <div class="uppercase border border-green-600 bg-green-100 text-green-600 font-bold p-2 my-3 text-sm">

            {{ session('message') }}

        </div>

    @endif

    @if (session()->has('messageError'))

        <div class="uppercase border border-red-600 bg-red-100 text-red-600 font-bold p-2 my-3 text-sm">

            {{ session('messageError') }}

        </div>

    @endif

    <div class="overflow-x-auto">

        <table class="table">

            <thead>

                <tr>

                    <th class="w-1/4 border border-green-600 text-green-600 font-bold">Nombre</th>

                    <th class="w-1/4 border border-green-600 text-green-600 font-bold">Servicio</th>

                    <th class="w-1/4 border border-green-600 text-green-600 font-bold">Fecha y Hora</th>

                    <th class="w-1/4 border border-green-600 text-green-600 font-bold">Acción</th>

                </tr>

            </thead>

            <tbody class="border border-green-400">

                @foreach($doctores as $doctor)

                    @foreach($doctor->serviciosRegistrados as $servicioRegistrado)

                        <tr>

                            <td class="font-bold border border-green-400">{{ $doctor->name }}</td>

                            <td class="font-bold border border-green-400">{{ $servicioRegistrado->servicio->nombreServicio }}</td>

                            <td class="border border-green-400">

                                <input type="datetime-local" wire:model="fechaHoraServicio.{{ $servicioRegistrado->id }}">

                            </td>

                            <td class="border border-green-400">

                                <button class="inline-flex items-center px-4 py-2 bg-green-400 border border-transparent rounded-md font-bold text-xs text-black uppercase tracking-widest hover:bg-green-400 active:bg-gray-900 focus:outline-none focus:border-green-500 focus:ring focus:ring-gray-900 disabled:opacity-25 transition"

                                wire:click="guardarFechaHora({{ $servicioRegistrado->id }})">Guardar</button>

                            </td>

                        </tr>

                    @endforeach

                @endforeach

            </tbody>

        </table>

    </div>

</div>

Ahora en el controlador **Doctor3Servicios.php** es el componente Livewire que gestiona la programación del servicio “Psicología” en una aplicación. Realiza las siguientes acciones:

* Recopila y muestra la lista de doctores (usuarios con un rol específico) en la variable $doctores.
* Permite a los usuarios asignar y guardar fechas y horas para servicios médicos mediante el método guardarFechaHora. También gestiona mensajes de éxito y error.
* En el método render, filtra y obtiene una lista de doctores específicos (en este caso, el Doctor 2 con ID 9) y los servicios registrados relacionados con ellos, que luego se muestran en la vista correspondiente.

En resumen, este archivo permite la gestión de fechas y horas para servicios médicos proporcionados por el doctor de “Psicología”.

<?php

namespace App\Http\Livewire;

use Livewire\Component;

use App\Models\User;

use App\Models\ServicioRegistrado;

class Doctor3Servicios extends Component

{

    public $doctores;

    public $fechaHoraServicio = [];

    public function guardarFechaHora($servicioRegistradoId)

    {

        $servicioRegistrado = ServicioRegistrado::find($servicioRegistradoId);

        if ($servicioRegistrado) {

            if ($servicioRegistrado->fecha\_hora\_servicio) {

                session()->flash('messageError', 'Ya se ha guardado una fecha y hora para este servicio.');

                return;

            }

            $servicioRegistrado->fecha\_hora\_servicio = $this->fechaHoraServicio[$servicioRegistradoId];

            $servicioRegistrado->save();

            session()->flash('message', 'Fecha y hora guardadas correctamente.');

        }

    }

    public function render()

    {

*// Filtra los registros de servicios registrados relacionados con el Doctor 1 (ID 10)*

        $this->doctores = User::where('role\_id', 2)

            ->whereHas('serviciosRegistrados', function ($query) {

                $query->where('doctor\_id', 9); *// Ajusta el ID del Doctor 1 según tu estructura de base de datos*

            })

            ->get();

        return view('livewire.doctor3-servicios');

    }

}