

Agendamiento de citas médicas desde Portal Web DOCMED

Cortés Navarrete, Sindy Andrea
Fundación Universitaria del Área Andina
scortes29@estudiantes.areandina.edu.co

Resumen – Por medio del presente proyecto se pretende implementar un proceso de agendamiento de citas médicas desde el portal web de la EPS, donde el usuario podrá realizar el ingreso mediante autenticación a todos los módulos y funcionalidades del portal del plan de salud, también estará habilitado el formulario de registro en caso de no tener acceso. Una vez ingrese con las credenciales asignadas, se podrá realizar el agendamiento de la cita requerida con medicina general, odontología o salud visual, quedando el registro en la base de datos de la entidad.

Palabras clave – Portal web, agendamiento, bases de datos, aplicación, python, conexión a mysql, flask, bcrypt, framework, jinja2.

Abstrac - This project pretend to implement a process for scheduling medical appointments from the EPS web portal, where the user will be able to enter through authentication to all the modules and functionalities of the health plan portal, the form will also be enabled registration in case of not having access. Once you enter with the assigned credentials, you can make the scheduling of the required appointment with general medicine, dentistry or visual health, saving the information in the entity's database.

Keywords – Web portal, scheduling, databases, app, python, connection to mysql, flask, bcrypt, framework, jinja2.

I. INTRODUCCIÓN

El presente proyecto tiene como objetivo mostrar algunas de las funcionalidades de python como lenguaje de programación y la consulta a bases de datos mysql aplicado a necesidades cotidianas. En esta oportunidad se llevará a cabo la inserción del módulo de citas en una aplicación web de una entidad de salud, permitiendo así que los pacientes tengan un nuevo canal de agendamiento para consulta con sus médicos tratantes, donde inicialmente estarán disponibles las citas con medicina general, odontología y salud visual.

II. FUNDAMENTACIÓN TEÓRICA

La puesta en marcha de la aplicación tiene múltiples opciones de desarrollo y las empresas en la actualidad buscan brindar mejores alternativas de servicio y funcionalidades a los usuarios, es por ello que un servicio de citas médicas es indispensable que tenga una recepción

oportuna. A continuación se listarán las aplicaciones utilizadas en el desarrollo del proyecto.

XAMPP

XAMPP fue creado con la intención de ser un programa de uso fácil, pensando en los nuevos desarrolladores que se encuentran aprendiendo del mundo de Apache como servidor web, en su distribución está incluido MariaDB/MySQL como sistema de gestión de bases de datos, PHP y Perl que son intérpretes para lenguajes de script. Se puede encontrar en una versión más ligera que permite realizar configuraciones desde interfaz web, adicionalmente mantiene actualizaciones regulares de cada uno de sus componentes.

MySQL

Es un sistema de gestión de bases de datos relacional, incluyendo herramientas de administración, escalable, de alta seguridad, promoviendo aplicaciones web, comercio electrónico, procesamiento de transacciones en línea, posee capacidad de reversión, recuperación de fallas entre otras funcionalidades. Para su más reciente versión 8.0, incluye mejoras en rendimiento y seguridad para aplicaciones web, móviles, integraciones y soluciones en la nube.

Python

Python es un lenguaje de programación que maneja una sintaxis sencilla permitiendo que la codificación sea legible mediante interpretación, multiplataforma y orientado a objetos. Es de código abierto, permitiendo el desarrollo para aplicaciones independientes, con gran contenido de librerías disponibles.

Flask

Flask es un micro Framework escrito en Python el cual permite crear de manera sencilla aplicaciones web, facilitando su desarrollo bajo el patrón MVC. Contiene las herramientas básicas para las aplicaciones web, sin embargo puede hacer uso de instalación de plugins

agregando nuevas funcionalidades. Permite encontrar librerías adaptadas al framework y proporciona estructura al proyecto, compartiendo los mismos ficheros y elementos a las aplicaciones creadas.

Bcrypt

Realiza la encriptación de contraseñas para software y servidores, basado en el cifrado de Blowfish, contiene un valor salt, el cual genera un fragmento aleatorio que se agrega al hash asociado a la clave, evitando la duplicidad o ataques por fuerza bruta. Opera en sistemas como OpenBSD, Linux y SUSE.

Apache 2.4

Servidor web de código abierto para sistemas operativos modernos, incluyendo UNIX y Windows. Proporciona seguridad con soporte SSL, reescritura de URL, es eficiente y extensible, incorporando los estándares actuales de HTTP. Realiza la configuración de servidores virtuales, manipulación sobre diferentes servidores, IP's, nombres de máquina o diversos puertos.

HTML5

Es un lenguaje de hipertexto, el cual se encarga de enlazar otros contenidos ya sea texto o archivos, utiliza el marcado, es decir, el contenido se elabora mediante etiquetas, este conjunto crea el estándar para definir la estructura y contenido de una página web. Su funcionamiento es mediante un navegador, el cual hace una petición al servidor, éste recupera el disco duro de la página y retorna su contenido.

CSS3

Hojas de estilo en cascada, proveniente del inglés, este lenguaje define la presentación de los documentos estructurados como HTML o XML, es guiado bajo las especificaciones de la W3C, define la estética de los sitios web como fuente, tamaño, colores, entre otros. Dentro de sus nuevas funcionalidades se encuentra incluido el borde redondeado, sombras, imágenes, entre otras múltiples opciones.

Java Script

Es un lenguaje basado en secuencias de comandos que permite emplear funciones complejas como actualizaciones de contenido, animación de imágenes, mapas, control de multimedia, operaciones sobre fragmentos de texto, además de integrarse con las API para emplear mayores funcionalidades.

jQuery

Es un complemento de JavaScript, dentro de su librería contiene diversas funciones y métodos para ampliar las funcionalidades de JavaScript. Tiene una gran ventaja ya que no genera conflicto con ningún navegador, maneja eventos, manipula el contenido de los documentos HTML. Incorpora utilidades de Ajax para realizar peticiones HTML dinámicamente, permitiendo trabajar con objetos y colecciones.

III. PROPUESTA DE SOLUCIÓN

Se plantea el desarrollo de la aplicación web con sus principales funcionalidades de gestión de agendamiento de control médico con sus datos y confirmación de inicio de sesión, adicionalmente contará con información general de los servicios del plan de salud.

Tareas que realiza el sistema de acuerdo al historial del usuario, describen su construcción para estimar tiempos en el desarrollo.

Se detallarán a continuación:

- Registro
- Logueo
- Asignación de citas

TABLA I

Número: 1	Nombre del requisito: Registro	
Programador: Cortés Navarrete Sindy Andrea	Iteración Asignada: 1	
Prioridad: Alta	Tiempo	Estimado: 6 horas
Riesgo en desarrollo: Alto		
Descripción: Registro del usuario en el portal de salud. Esta funcionalidad permite al usuario realizar su registro en las bases de datos para posteriormente ingresar a las funcionalidades del sistema. Los campos del formulario de registro son: <ul style="list-style-type: none"> - Nombre: información del usuario. - Apellido: información del usuario. - Documento: información del usuario. - Fecha de nacimiento: información del usuario. - Celular: contacto del usuario. - Email: correo del usuario que hará la vez de usuario de ingreso al sistema. - Contraseña: contraseña asignada por el usuario, encriptada por seguridad. 		
Observaciones:		
Prototipo:		

TABLA II

Número: 2	Nombre del requisito: Logueo	
Programador: Cortés Navarrete Sindy Andrea	Iteración Asignada: 1	
Prioridad: Alta	Tiempo	Estimado: 6 horas
Riesgo en desarrollo: Alto		
Descripción: Ingreso del usuario al portal de salud. Una vez se haya realizado el registro, el usuario podrá ingresar para acceder a más funcionalidades del portal, realizará las validaciones de logueo, en caso de no estar registrado, le solicitará diligenciar el formulario de registro. Los campos de ingreso son: <ul style="list-style-type: none"> - Email: correo registrado. - Contraseña: clave asignada. 		
Observaciones:		
Prototipo:		

TABLA III

Número: 3	Nombre del requisito: Asignación de citas	
Programador: Cortés Navarrete Sindy Andrea	Iteración Asignada: 1	
Prioridad: Alta	Tiempo	Estimado: 12 horas
Riesgo en desarrollo: Alto		
Descripción: Ingresando al portal, se encuentra el botón de agendamiento, al dar clic sobre el, desplegará el formulario de diligenciamiento para asignar la cita médica.		

Los campos de agendamiento son:

- Fecha cita: fecha del día seleccionado de la cita médica.
- Hora cita: hora del día seleccionado de la cita médica.
- Área: área de salud requerida por el usuario.
- Doctor: doctor seleccionado para atención de la consulta.
- Nombre: nombre del paciente agendado
- Apellido: apellido del paciente agendado
- Celular: contacto telefónico del paciente agendado.
- Email: correo del paciente agendado.

Observaciones:

Prototipo:

Botón de agendamiento

Direccionamiento

IV. MANUAL DE USUARIO

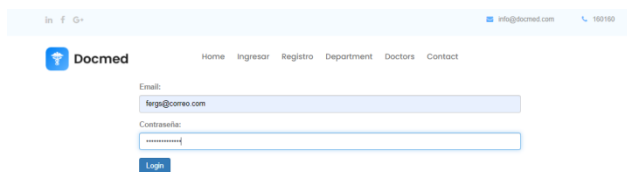
Por medio del siguiente manual, se brindará la guía al usuario para el proceso de agendamiento de citas médicas desde la aplicación web.

Home

Desde el Home el usuario tendrá acceso a la información de contacto, doctores y áreas de especialidad:



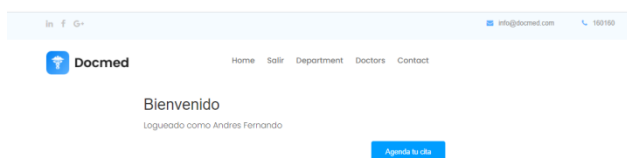
Para acceder a la funcionalidad de agendamiento y servicios médicos, deberá primero iniciar sesión con el correo y contraseña previamente registrados:



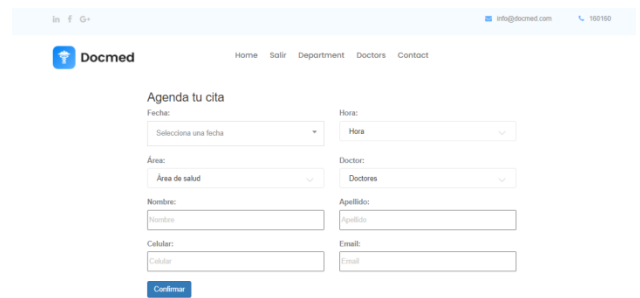
En caso que el usuario o contraseña sean incorrectos, será notificado por medio de un mensaje:

Usuario o contraseña incorrecta

Al realizar el logueo, el mensaje de bienvenida, indicará el nombre del usuario logueado y habilitará el botón de agendamiento de citas:



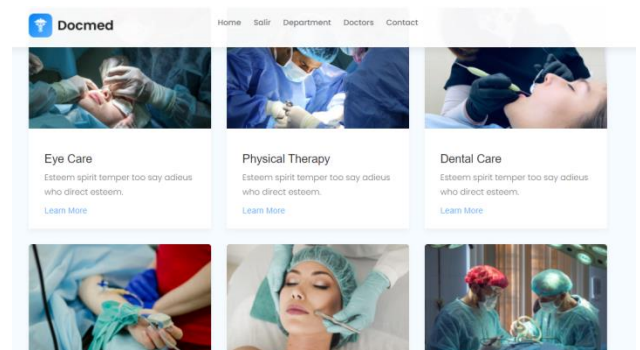
Para agendar una cita, dar clic en Agenda tu cita, se mostrará un formulario donde se seleccionará la fecha y hora deseada de la cita, además del área y doctor requerido, más los datos del paciente:



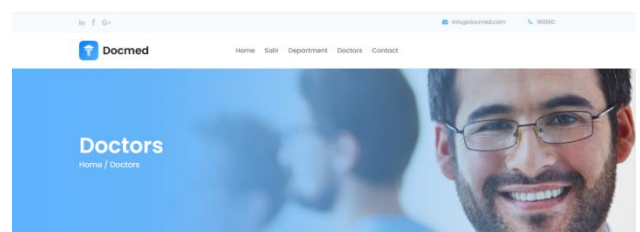
La cita quedará alojada en bases de datos, de acuerdo a las selecciones indicadas por el usuario.

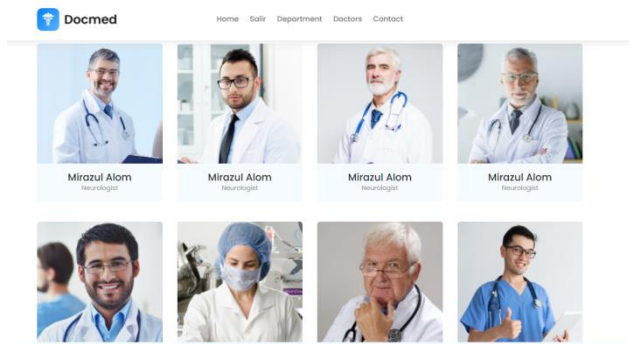
También se encontrará disponible la información de:

- Departamentos

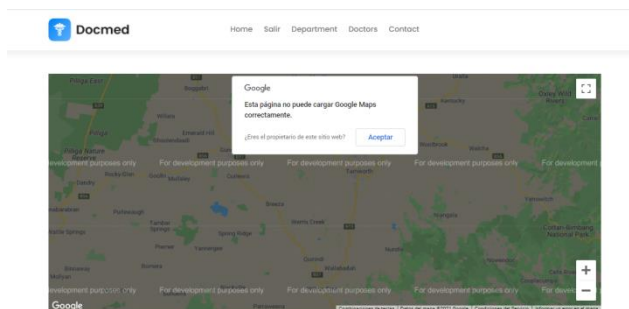


- Doctores





- Contacto



V. MANUAL DEL PROGRAMADOR

A continuación se brindarán las indicaciones sobre la codificación realizada por medio del lenguaje python para la realización de la aplicación.

Layout

En el layout.html quedará registrado como plantilla para las demás páginas que conecten con la principal, su bloqueo se realizará por medio de los comandos de jinja2.

Se bloqueará la parte del título para que cada una de las páginas adicionales pueda contener el suyo propio:

```

<!DOCTYPE html>
<html class="no-js" lang="zxx">
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>{% block title %}{% endblock %}</title>
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- <link rel="manifest" href="site.webmanifest"> -->
<link rel="shortcut icon" type="image/x-icon" href="static/img/favicon.png">
<!-- Place favicon.ico in the root directory -->

<!-- CSS here -->
<link rel="stylesheet" href="static/css/bootstrap.min.css">
<link rel="stylesheet" href="static/css/owl.carousel.min.css">
<link rel="stylesheet" href="static/css/magnific-popup.css">
<link rel="stylesheet" href="static/css/font-awesome.min.css">

```

En el body habrá un condicional que validará si el usuario se encuentra logueado para permitir la visualización de funcionalidades accesibles mediante sesión, de lo contrario lo redireccionará a las opciones de login o registro en caso de no tener datos en sistema:

```

<div class="col-xl-6 col-lg-7">
<div class="main-menu d-none d-lg-block">
<nav>
<ul id="navigation">
<!--<li><a class="active" href="index.html">Home</a></li>-->
<li>
<a class="active" href="#">Home</a>
</li>
{% if session['nombre'] %}<!--Validación del ingreso de sesión-->
<li>
<a href="/logout">Salir</a>
</li>
{% else %}
<li>
<a href="/login">Ingresar</a>
</li>
<li>
<a href="/register">Registro</a>
</li>
{% endif %}
<li><a href="/department">Department</a></li>
<li><a href="/doctors">Doctors</a></li>
<li><a href="/contact">Contact</a></li>
</ul>
</nav>
</div>

```

Javascript de la funcionalidad del datepicker que se desplegará a través de confirm.html:

```

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.timepicker/1.3.5/jquery.timepicker.min.js"></script>
<script src="jquery.datettimepicker.full.min.js"></script>
<script>
$( "#datepicker" ).datepicker({
iconsLibrary: 'fontawesome',
icons: {
rightIcon: 'span class="fa fa-caret-down"></span>'
}
});
$( "#datepicker2" ).datepicker({
iconsLibrary: 'fontawesome',
icons: {
rightIcon: 'span class="fa fa-caret-down"></span>'
}
});
$(document).ready(function () {
$( '.js-example-basic-multiple' ).select2();
});
</script>

```

Agenda tu cita

Fecha:

Selecciona una fecha

<

July 2021

>

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Home

Hereda de layout la plantilla de scripts, continuará validando el inicio de sesión de parte del usuario y se habilitará el botón de agendamiento:

[illegible]

Login

Hereda de layout, registra formulario para realizar el ingreso a la aplicación:

```

1 [% extends 'layout.html' %]{!--Herencia de layout.html--}
2
3 {% block title %} Login {% endblock %}
4
5 {% block content %}
6
7 <!--Login-->
8 <div class="row">
9     <div class="col-md-6 col-md-offset-3">
10         <form action="/login" method="POST">
11             <div class="form-group">
12                 <label>Email:</label>
13                 <input type="text" class="form-control" name="email">
14             </div>
15             <div class="form-group">
16                 <label>Contraseña:</label>
17                 <input type="password" class="form-control" name="password">
18             </div>
19             <div class="form-group">
20                 <button type="submit" class="btn btn-primary">Login</button>
21             </div>
22         </form>
23     </div>
24 </div>
25
26 {% endblock %}

```

Register

Hereda de layout, registra formulario para realizar el registro de los datos del usuario en la base de datos de la aplicación:

```

1 Users > SINDY > Desktop > proyecto_task > templates > register.html > ...
2 1 {{% extends 'layout.html' %}}<!--Herencia de layout.html-->
3 2
4 3 {{% block title %}} Registro {{% endblock %}}
5 4
6 5 {{% block content %}}
7 6
8 7 <!--Formulario de registro-->
9 8 <div class="row">
10 9   <div class="col-md-6 col-md-offset-3">
11 10     <form action="/register" method="POST">
12 11       <div class="form-group">
13 12         <label>Nombre:</label>
14 13         <input type="text" class="form-control" name="nombre">
15 14       </div>
16 15       <div class="form-group">
17 16         <label>Apellido:</label>
18 17         <input type="text" class="form-control" name="apellido">
19 18       </div>
20 19       <div class="form-group">
21 20         <label>Documento:</label>
22 21         <input type="number" class="form-control" name="documento">
23 22       </div>
24 23       <div class="form-group">
25 24         <label>Fecha de nacimiento:</label>
26 25         <input type="datetime" class="form-control" name="fecha_nac" placeholder="yy-mm-dd">
27 26       </div>
28 27       <div class="form-group">
29 28         <label>Celular:</label>
30 29         <input type="text" class="form-control" name="celular">
31 30       </div>
32 31       <div class="form-group">
33 32         <label>Email:</label>
34 33         <input type="text" class="form-control" name="email">
35 34       </div>
36 35       <div class="form-group">
37 36         <label>Contraseña:</label>
38 37         <input type="password" class="form-control" name="password">
39 38       </div>
40 39       <div class="form-group">
41 40         <button type="submit" class="btn btn-primary">Registrar</button>
42 41       </div>
43 42     </form>
44 43   </div>
45 44 </div>
46 45 {{% endblock %}}

```

Confirm

Hereda de layout, registra formulario para confirmar el registro del agendamiento de la cita seleccionada por el usuario en la base de datos de la aplicación:

```

1  % extends 'layout.html' %|<!--Herencia de layout.html-->
2
3  {% block title %} Confirmar {% endblock %}
4
5  {% block content %}
6
7  <!--Formulario de agendamiento-->
8  <div class="row">
9      <div class="col-md-6 col-md-offset-3">
10         <h3>Agenda tu cita</h3>
11         <form action="#"><confirm method="POST">
12             <div class="row">
13                 <div class="col-xl-6">
14                     <label>
15                         <input type="datetime" id="datepicker" placeholder="Selecciona una fecha" name="fecha_cita">
16                     </div>
17                     <div class="col-xl-6">
18                         <label>Hora: <label>
19                             <select type="time" class="form-select wde" id="default-select" class="" name="hora_cita">
20                                 <option data-display="Hora"--- : --</option>
21                                 <option value="1">07:00:00</option>
22                                 <option value="2">7:20 a.m.</option>
23                                 <option value="3">7:40 a.m.</option>
24                                 <option value="4">8:00 a.m.</option>
25                                 <option value="5">8:20 a.m.</option>
26                                 <option value="6">8:40 a.m.</option>
27                                 <option value="7">9:00 a.m.</option>
28                                 <option value="8">9:20 a.m.</option>
29                                 <option value="9">9:40 a.m.</option>
30                                 <option value="10">10:00 a.m.</option>
31                                 <option value="11">10:20 a.m.</option>
32                                 <option value="12">10:40 a.m.</option>
33                             </select>
34                         </div>
35                     </div>
36                 </div>
37             </div>
38         </form>
39     </div>
40 </div>
41
42 </div>
43
44 </div>
45
46 </div>
47
48 </div>
49
50 </div>
51
52 </div>
53
54 </div>
55
56 </div>
57
58 </div>
59
60 </div>
61
62 </div>
63
64 </div>
65
66 </div>
67
68 </div>
69
70 </div>
71
72 </div>
73
74 </div>
75
76 </div>
77
78 </div>
79
80 </div>
81
82 </div>
83
84 </div>
85
86 </div>
87
88 </div>
89
90 </div>
91
92 </div>
93
94 </div>
95
96 </div>
97
98 </div>
99
100 </div>
101
102 </div>
103
104 </div>
105
106 </div>
107
108 </div>
109
110 </div>
111
112 </div>
113
114 </div>
115
116 </div>
117
118 </div>
119
120 </div>
121
122 </div>
123
124 </div>
125
126 </div>
127
128 </div>
129
130 </div>
131
132 </div>
133
134 </div>
135
136 </div>
137
138 </div>
139
140 </div>
141
142 </div>
143
144 </div>
145
146 </div>
147
148 </div>
149
150 </div>
151
152 </div>
153
154 </div>
155
156 </div>
157
158 </div>
159
160 </div>
161
162 </div>
163
164 </div>
165
166 </div>
167
168 </div>
169
170 </div>
171
172 </div>
173
174 </div>
175
176 </div>
177
178 </div>
179
180 </div>
181
182 </div>
183
184 </div>
185
186 </div>
187
188 </div>
189
190 </div>
191
192 </div>
193
194 </div>
195
196 </div>
197
198 </div>
199
200 </div>
201
202 </div>
203
204 </div>
205
206 </div>
207
208 </div>
209
210 </div>
211
212 </div>
213
214 </div>
215
216 </div>
217
218 </div>
219
220 </div>
221
222 </div>
223
224 </div>
225
226 </div>
227
228 </div>
229
230 </div>
231
232 </div>
233
234 </div>
235
236 </div>
237
238 </div>
239
240 </div>
241
242 </div>
243
244 </div>
245
246 </div>
247
248 </div>
249
250 </div>
251
252 </div>
253
254 </div>
255
256 </div>
257
258 </div>
259
260 </div>
261
262 </div>
263
264 </div>
265
266 </div>
267
268 </div>
269
270 </div>
271
272 </div>
273
274 </div>
275
276 </div>
277
278 </div>
279
280 </div>
281
282 </div>
283
284 </div>
285
286 </div>
287
288 </div>
289
290 </div>
291
292 </div>
293
294 </div>
295
296 </div>
297
298 </div>
299
300 </div>
301
302 </div>
303
304 </div>
305
306 </div>
307
308 </div>
309
310 </div>
311
312 </div>
313
314 </div>
315
316 </div>
317
318 </div>
319
320 </div>
321
322 </div>
323
324 </div>
325
326 </div>
327
328 </div>
329
330 </div>
331
332 </div>
333
334 </div>
335
336 </div>
337
338 </div>
339
340 </div>
341
342 </div>
343
344 </div>
345
346 </div>
347
348 </div>
349
350 </div>
351
352 </div>
353
354 </div>
355
356 </div>
357
358 </div>
359
360 </div>
361
362 </div>
363
364 </div>
365
366 </div>
367
368 </div>
369
370 </div>
371
372 </div>
373
374 </div>
375
376 </div>
377
378 </div>
379
380 </div>
381
382 </div>
383
384 </div>
385
386 </div>
387
388 </div>
389
390 </div>
391
392 </div>
393
394 </div>
395
396 </div>
397
398 </div>
399
400 </div>
401
402 </div>
403
404 </div>
405
406 </div>
407
408 </div>
409
410 </div>
411
412 </div>
413
414 </div>
415
416 </div>
417
418 </div>
419
420 </div>
421
422 </div>
423
424 </div>
425
426 </div>
427
428 </div>
429
430 </div>
431
432 </div>
433
434 </div>
435
436 </div>
437
438 </div>
439
440 </div>
441
442 </div>
443
444 </div>
445
446 </div>
447
448 </div>
449
450 </div>
451
452 </div>
453
454 </div>
455
456 </div>
457
458 </div>
459
460 </div>
461
462 </div>
463
464 </div>
465
466 </div>
467
468 </div>
469
470 </div>
471
472 </div>
473
474 </div>
475
476 </div>
477
478 </div>
479
480 </div>
481
482 </div>
483
484 </div>
485
486 </div>
487
488 </div>
489
490 </div>
491
492 </div>
493
494 </div>
495
496 </div>
497
498 </div>
499
500 </div>
501
502 </div>
503
504 </div>
505
506 </div>
507
508 </div>
509
510 </div>
511
512 </div>
513
514 </div>
515
516 </div>
517
518 </div>
519
520 </div>
521
522 </div>
523
524 </div>
525
526 </div>
527
528 </div>
529
530 </div>
531
```

```

33         <option value="13">11:00 a.m.</option>
34         <option value="14">11:20 a.m.</option>
35         <option value="15">11:40 a.m.</option>
36         <option value="16">12:00 a.m.</option>
37         <option value="17">12:20 a.m.</option>
38         <option value="18">12:40 a.m.</option>
39         <option value="19">01:00 p.m.</option>
40         <option value="20">01:20 p.m.</option>
41         <option value="21">01:40 p.m.</option>
42         <option value="22">02:00 p.m.</option>
43         <option value="23">02:20 p.m.</option>
44         <option value="24">02:40 p.m.</option>
45         <option value="25">03:00 p.m.</option>
46         <option value="26">03:20 p.m.</option>
47         <option value="27">03:40 p.m.</option>
48         <option value="28">04:00 p.m.</option>
49         <option value="29">04:20 p.m.</option>
50         <option value="30">04:40 p.m.</option>
51         <option value="31">05:00 p.m.</option>
52         <option value="32">05:20 p.m.</option>
53         <option value="33">05:40 p.m.</option>
54         <option value="34">06:00 p.m.</option>
55         <option value="35">06:20 p.m.</option>
56         <option value="36">06:40 p.m.</option>
57     </select>
58 </div>

```

```

70 <select type="text" class="form-select wide" id="default-select" class="" name="doctor">
71     <option data-display="Doctores" Doctores</option>
72     <option value="1">Arizuel Alon</option>
73     <option value="2">Monzel Alon</option>
74     <option value="3">Arizuel Isalm</option>
75 </select>
76 </div>
77 <div class="col-md-6"><br>
78     <label Nombre:</label>
79     <input type="text" class="form-select wide" placeholder="Nombre" name="nombre">
80 </div>
81 <div class="col-md-6"><br>
82     <label Apellido:</label>
83     <input type="text" class="form-select wide" placeholder="Apellido" name="apellido">
84 </div>
85 <div class="col-xi-6"><br>
86     <label Celular:</label>
87     <input type="text" class="form-select wide" placeholder="Celular" name="celular">
88 </div>
89 <div class="col-xi-6"><br>
90     <label Email:</label>
91     <input type="email" class="form-select wide" placeholder="Email" name="email">
92 </div>
93 <div class="col-xi-12"><br>
94     <button type="submit" class="btn btn-primary" name="confirmar">Confirmar</button>
95 </div>
96 </div>
97 </div>
98 </form>
99 </div>
100 </div>
101 {% endblock %}

```

App

Aplicación realizada en python que realiza la conexión a base de datos mysql, se realiza la importación de librerías que se implementarán en cada una de las etapas del proyecto:

```

app.py  A X
C:\Users\SINDY\Desktop> proyecto_flask > app.py
1 from flask import Flask, render_template, request, redirect, url_for, flash, session
2 from flask_mysql import MySQL, MySQLdb
3 import bcrypt
4
5 app = Flask(__name__)
6

```

Conexión a MySQL con las respectivas credenciales y creación del cursor que realizará el recorrido:

```

# Conexión Mysql
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = '1q2w3e4r'
app.config['MYSQL_DB'] = 'agendamiento'
app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
mysql = MySQL(app)

```

Ruta al home:

```

# Direcccionamiento al home
@app.route('/')
def home():
    return render_template('home.html')

```

Validación del formulario de registro, realizando inserción en la base de datos, generando encriptación para la contraseña que se va a ingresar, retorno al home al final del proceso:

```

# Registro de nuevo usuario en base de datos
@app.route('/register', methods=["GET", "POST"])
def register():
    if request.method == 'GET':
        return render_template("register.html")
    else:
        nombre = request.form['nombre']
        apellido = request.form['apellido']
        documento = request.form['documento']
        fecha_nac = request.form['fecha_nac']
        celular = request.form['celular']
        email = request.form['email']
        password = request.form['password'].encode('utf-8')
        hash_password = bcrypt.hashpw(password, bcrypt.gensalt())

        cur = mysql.connection.cursor()
        cur.execute('INSERT INTO usuarios (documento, nombre, apellido, fecha_nac, email, celular, password) VALUES (%s, %s, %s, %s, %s, %s, %s)')
        mysql.connection.commit()
        session['nombre'] = nombre
        session['email'] = email
        return redirect(url_for("home"))

```

Validación de los datos ingresados, recorridos mediante un cursor en bases de datos, permitiendo el ingreso o retornando error de credenciales:

```

# Logueo
@app.route('/login', methods=["GET", "POST"])
def login():
    if request.method == "POST":
        email = request.form['email']
        password = request.form['password'].encode('utf-8')

        cur = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cur.execute('SELECT * FROM usuarios WHERE email=%s',(email,))
        user = cur.fetchone()
        cur.close()

        if len(user) > 0:
            if bcrypt.hashpw(password, user['password'].encode('utf-8')) == user['password'].encode('utf-8'):
                session['nombre'] = user['nombre']
                session['email'] = user['email']
                return render_template("home.html")
            else:
                return "Usuario o contraseña incorrecta"
        else:
            return "Usuario o contraseña incorrecta"
    else:
        return render_template("login.html")

```

Confirmación de las citas médicas, mediante la inserción de los datos seleccionados en la tabla de citas, con retorno al home:

```

# Confirmación de citas médicas
@app.route('/confirm', methods=["GET", "POST"])
def confirm():
    if request.method == 'GET':
        return render_template("confirm.html")
    else:
        nombre = request.form['nombre']
        apellido = request.form['apellido']
        celular = request.form['celular']
        email = request.form['email']
        fecha_cita = request.form['fecha_cita']
        hora_cita = request.form['hora_cita']
        area = request.form['area']
        doctor = request.form['doctor']

        cur = mysql.connection.cursor()
        cur.execute('INSERT INTO citas (nombre, apellido, celular, email, fecha_cita, hora_cita, area, doctor) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)')
        mysql.connection.commit()
        session['nombre'] = nombre
        session['email'] = email
        return redirect(url_for("home"))

```

Redireccionamiento a otros enlaces de la aplicación:


```
# Redireccionamiento a otras secciones
@app.route('/department')
def department():
    return render_template('department.html')

@app.route('/doctors')
def doctors():
    return render_template('doctors.html')

@app.route('/contact')
def contact():
    return render_template('contact.html')
```

Salida del portal, limpia la sesión iniciada:

```
# Salida del portal
@app.route('/logout')
def logout():
    session.clear()
    return render_template("home.html")
```

Función principal, contiene clave secreta para establecer sesión segura, modo debug encendido con operación en el localhost desde el puerto 3000:

```
if __name__ == '__main__':
    app.secret_key = 'Ao!UQf)2qK3h#04' #mysecretkey
    app.run(port=3000, debug=True)
```

CONCLUSIONES

El desarrollo del presente proyecto, permitió conocer, practicar y aplicar diversas funcionalidades de python para ofrecer soluciones a necesidades cotidianas, gracias a su versatilidad, aplicabilidad en diferentes campos y su facilidad de aprendizaje, considerado como lenguaje de sintaxis sencilla.

En la ejecución del proyecto se llevaron a cabo los objetivos propuestos, sin embargo queda el espacio como oportunidad de mejora, para que la funcionalidad de agendamiento contenga un asistente virtual o chatbot, que podrá ser entrenado para dar claridad a las inquietudes del usuario frente al agendamiento de citas médicas y su extensión a todos los servicios del plan de salud.

REFERENCIAS

APACHEFRIENDS, 2008. XAMPP. [Versión 3.3] Recuperado de <https://www.apachefriends.org/es/index.html>

ORACLE, 2015. MYSQL. [Versión 7.3] Recuperado de <https://www.mysql.com/products/enterprise/database/>

ROSSUM, GUIDO. V. 2019. PYTHON. [Versión 3.8] Recuperado de <https://www.python.org/>

MUÑOZ, JOSE. D. 2017. Qués es Flask. Blog de programación y sistemas. OpenWebinars. Recuperado de <https://openwebinars.net/blog/que-es-flask/>

The Python Cryptographic Authority developers. 2020. bcrypt. [Versión 3.2.0] Recuperado de <https://pypi.org/project/bcrypt/>

IZERTIS. 2017. Encriptación de password en NodeJS y MongoDB: bcrypt. Recuperado de <https://www.izertis.com/es/-/blog/encriptacion-de-password-en-nodejs-y-mongodb-bcrypt>

BARROSO GÓMEZ, BÁRBARO, G. (s.f.) Plataforma para la creación de boletines informativos para el centro Telemática de la UCI. Recuperado de http://www.laccei.org/LACCEI2017-BocaRaton/student_Papers/SP221.pdf

The Apache Software Foundation. 2018. APACHE. HTTP SERVER PROJECT. Recuperado de <https://httpd.apache.org/>

Pérez Jimenez, Juan. D. 2019. Qué es HTML5. Blog de programación y sistemas. OpenWebinars. Recuperado de <https://openwebinars.net/blog/que-es-html5/>

ECURED. 1996. CSS3. Introducción a CSS3. Recuperado de <https://www.ecured.cu/CSS3>

MDN contributors. 2021. ¿Qué es JavaScript? Recuperado de https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript

DESARROLLO WEB. (s.f.) Manual de jQuery. Recuperado de <https://desarrolloweb.com/manuales/manual-jquery.html>

COLORLIB. 2020. DOCMED. Bootstrap Templates. Recuperado de <https://colorlib.com/wp/template/docmed/>