

Universidad Tecnológica de Xicotepec

Proyecto en Angular



Integrantes:

- *David García Olivares
- *Adolfo Meléndez Rodríguez
- *Claribel Trejo Ocampo
- *Sandra Garrido Garrido
- *Soledad Reyes Martínez

DOCENTE:

Iván García Quintero

MATERIA

**Seguridad en el
Desarrollo de
Aplicaciones**

Wireframes del proyecto

Iniciar sesion

Email

Password

Login

CitasApp Inicio

Registro de Citas

Nueva Cita

#	Tramite	Fecha	Alumno
991b17cf-873d-4d97-a317-cb8fdec00bb3	Kardex	March 30, 2022 at 11:24:00 PM GMT-6	Citas App Administrador
aa4c89d3-78d0-40bb-9068-9ab65199159c	Reposición de Credencial	March 18, 2022 at 11:22:00 PM GMT-6	Citas App Administrador

Username or password changed! Save it? Username: admin@citassapp.com

New

Update

☐ Don't ask again for this site

Dismiss

Nueva Cita

Tramite

Kardex

Fecha

mm/dd/yyyy, --:-- --

March 2022

↑

↓

S	M	T	W	T	F	S
27	28	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	1	2
3	4	5	6	7	8	9

Clear

Today

11

24

PM

12

25

AM

01

26

02

27

03

28

04

29

05

30



Username or password changed! Save it? Username: admin@citassapp.com

New

Update

☐ Don't ask again for this site

Dismiss

Nueva Cita

Tramite

Fecha

mm/dd/yyyy, --:-- --



Registrar

Username or password changed! Save it? Username: admin@citassapp.com

New

Update

☐ Don't ask again for this site

Dismiss

Registro de Citas

Nueva Cita

#	Tramite	Fecha	Alumno
aa4c89d3-78d0-40bb-9068-9ab65199159c	Reposición de Credencial	March 18, 2022 at 11:22:00 PM GMT-6	Citas App Administrador



Componentes principales del proyecto

Un componente incluye las propiedades y métodos disponibles para su template, pero hay que ser consciente de que no toda la lógica debe contenerla esta clase, deberíamos incluir en esta clase todo lo referente al controlador de la vista y abstraer todos los demás métodos en servicios que serán inyectados posteriormente.

Un archivo que será nuestro Template (app.component.html), el cual es nuestro HTML, que es el que se va a visualizar en la interfaz de usuario, la vista o en términos más simples lo que se visualizara en la página.

Un archivo de lógica, la cual es la que pondremos en un archivo .ts (app.component.ts), ese archivo debe incluir una clase y esta es la que va a contener las propiedades que se van a usar en la vista (HTML) y los métodos que serán las acciones que se ejecutarán en la vista. En este archivo de lógica también se incluye una metadata, que es definida con un decorador, que identifica a Angular como un componente.

Un archivo para el CSS (podemos usar un preprocesador como SASS o LESS), donde incluiremos los estilos, lo que nos ayuda a hacer bonita nuestra aplicación.

Código del app.Component, y add-booking-component es decir, nuestros componentes principales de nuestro proyecto denominado "Citas Admin".

```
TS app.component.ts X
src > app > TS app.component.ts > ...
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'cservicios';
10 }
11
```



```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-add-booking',
5   templateUrl: './add-booking.component.html',
6   styleUrls: ['./add-booking.component.css'],
7 })
8 export class AddBookingComponent implements OnInit {
9   public options: string[];
10  public locale: any;
11  constructor() {
12    this.options = ['title', 'credential', 'college-insurance', 'other']
13    if (navigator.language == 'es') {
14      this.locale = {
15        firstDayOfWeek: 1,
16        dayNames: ["domingo", "lunes", "martes", "miércoles", "jueves", "viernes", "sábado"],
17        dayNamesShort: ["dom", "lun", "mar", "mié", "jue", "vie", "sáb"],
18        dayNamesMin: ["D", "L", "M", "X", "J", "V", "S"],
19        monthNames: ["enero", "febrero", "marzo", "abril", "mayo", "junio", "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"],
20        monthNamesShort: ["ene", "feb", "mar", "abr", "may", "jun", "jul", "ago", "sep", "oct", "nov", "dic"],
21        today: 'Hoy',
22        clear: 'Borrar'
23      }
24    } else {
25      this.locale = {
26        firstDayOfWeek: 0,
27        dayNames: ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"],
28        dayNamesShort: ["Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"],
29        dayNamesMin: ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"],
30        monthNames: ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"],
31        monthNamesShort: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"],
32        today: 'Today',
33        clear: 'Clear',
34        dateFormat: 'mm/dd/yy',
35        weekHeader: 'Wk'
36      }
37    }
38  }
39 }
```

Estos ejemplos muestran algunos de los atributos más importante y usados:

selector: Se trata de un selector CSS que indica a Angular que debe crear una instancia de este componente cuando encuentre un tag con ese nombre en el HTML. En este caso, cuando Angular encuentre `<app-root></app-root>` insertará una instancia de este componente.

templateUrl: Es la url donde se encuentra el template que se vinculara a este componente.

styleUrls: Un array con las urls a los archivos de estilo (CSS) que aplicaremos a nuestro componente.



Template

El template es el encargado de definir la vista de nuestro componente de un HTML tradicional pero con una serie de expresiones que definen el comportamiento de este y nos hace la vida más fácil.

En el template de un componente podemos encontrar, además, elementos distintos que utiliza Angular.

*ngIf

*ngFor

(event)

[property]

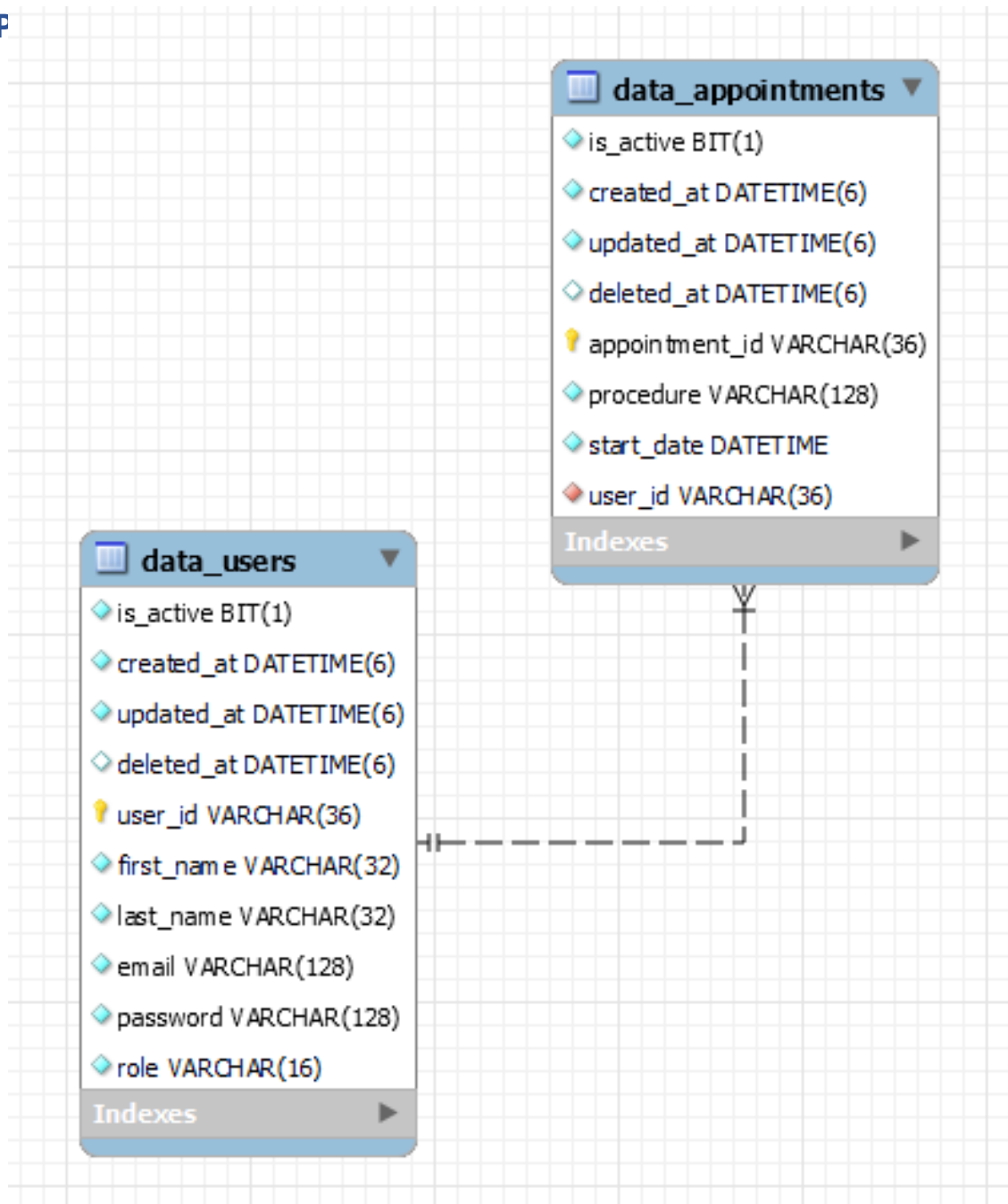
<component></component>

```
src > app > < app.component.html > div.container.h-100
1 <div class="container h-100">
2   <div class="row">
3     <div class="col-12">
4       <app-header></app-header>
5     </div>
6   </div>
7   <div class="row">
8     <div class="col-12">
9       {{'label.ejemplo' | translate}}
10      <router-outlet>
11    </router-outlet>
12  </div>
13 </div>
14 </div>
15 </div>
```

```
src > app > components > add-booking > < add-booking.component.html > div#add-booking.row > div.col-12 > form > div.row.form-group
1
2 <div class="row" id="add-booking">
3   <div class="col-12">
4
5     <form>
6       <div class="row form-group">
7         <div class="col-12 mt-2">
8           <label for="name">{{'label.name' | translate}} </label>
9           <input type="text" name="name" id="name" class="form-control">
10        </div>
11      </div>
12      <div class="row form-group">
13        <div class="col-12 mt-2">
14          <label for="date">{{'label.date' | translate}} </label>
15          <p-calendar [locale]="locale"
16            dateFormat="dd/mm/yy"
17            [stepMinute]="30"
18            id="date"
19            [showTime]="true"
20            [readonlyInput]="true"
21            inputStyleClass="form-control is-valid"
22          ></p-calendar>
23        </div>
24      </div>
25    </div>
26    <div class="row form-group">
27      <div class="col-12 mt-2">
28        <label for="services">{{'label.services' | translate}} </label>
29        <select class="form-select" name="services" id="services">
30          <option title="Titulo"></option>
31          <option credential="1">Credencial</option>
32          <option college-insurance="2">Seguro Facultativo</option>
33          <option other="3">Otro</option>
34          <option [selected]="first" [value]="option" *ngFor="let option of options; let first=first"></option>
35        </select>
36      </div>
37    </div>
38  </div>
39  <div class="row form-group">
40    <div class="col-12">
41
```



F



Propuesta del backend

A partir de la observación del contexto de la universidad de la Utxj y viendo las necesidades que se presentan se llevo a cabo la tarea de diseñar una aplicación dentro de la materia “Seguridad en el desarrollo de aplicaciones”.

La decisión se tomo de manera consensual dentro del equipo numero 4 llegando a la conclusión de que en muchas ocasiones los tramites administrativos dentro de la universidad son muy burocráticos y que debido a la necesidad de muchos estudiantes, principalmente los que trabajamos y radicamos fuera del municipio de Xicotepec no disponíamos de cualquier día u hora para poder realizarlos, es por ello que se tomo la decisión de realizar un sistema que permita agendar una o mas citas en la universidad para realizar el tramite correspondiente, dichas citas pueden ser agendadas por día y hora así como el tipo de servicio que se desea realizar.

La api desarrollada se realizo con la tecnología de node.js, la propuesta del backend radica en el hecho de que en primera instancia, cada alumno cuente con un usuario y contraseña proporcionado por la universidad, mismos que le permitirán acceder a la aplicación y poder generar las citas como punto inicial, como segundo punto el alumno podrá escoger el tipo de tramite que desea realizar, como tercero la misma aplicación le arrojará un mensaje al usuario indicándoles la advertencia si ya existe una cita registra a esa hora en que desea agendarla.

```
31
32     if (optional > 0)
33         throw new ConflictException(
34             `Ya existe una cita registrada para ${createAppointmentDto.procedure} a la hora indicada`,
35         );
36
37     const user = await this.usersService.findOne(createAppointmentDto.userId);
38
39     const appointment = this.appointmentRepository.create({
40         ...createAppointmentDto,
41         user: user,
42     });
43
44     return this.appointmentRepository.save(appointment);
45 }
46
47 findAll() {
48     return this.appointmentRepository.find();
49 }
50
51 async findOne(id: string) {
52     const appointment = await this.appointmentRepository.findOne(id);
53
54     if (!appointment)
55         throw new NotFoundException(`La cita con ID ${id} no existe`);
56
57     return appointment;
58 }
59
60 async update(id: string, updateAppointmentDto: UpdateAppointmentDto) {
61     const appointment = await this.appointmentRepository.preload({
62         ...updateAppointmentDto,
63         appointmentId: id,
64     });
65
66     if (!appointment)
67         throw new NotFoundException(`La cita con ID ${id} no existe`);
68
69     return this.appointmentRepository.save(appointment);
70 }
71 }
```



Enlace del proyecto:

<https://github.com/admin27092001/ProjectSDA>

