

# Tópico Especiales

Erick Bolaños, Carlos Lucero, Diana Narváez, Toasa Lizeth

Escuela Politécnica Nacional (EPN)

Escuela de Formación de Tecnólogos(ESFOT)

## I. PROPÓSITO DE LA PRÁCTICA

Realizar una aplicación en IONIC que permita llevara el registro de sus calificaciones semestrales.

## II. OBJETIVO GENERAL

Realizar una aplicación móvil la cual nos permita insertar, eliminar o modificar las calificaciones ingresadas.

## III. OBJETIVOS ESPECÍFICOS

Conocer las herramientas a utilizarse.

Conocer la implementación del CRUD en Angular.

Analizar los resultados obtenidos.

## IV. INSTRUCCIONES

1. Instale el módulo Firebase en el proyecto Ionic 4 / Angular. Ejecute el siguiente comando en la terminal.
2. Para almacenar los datos en la base de datos de Firebase Realtime, debe agregar sus claves de Firebase environment.prod.ts

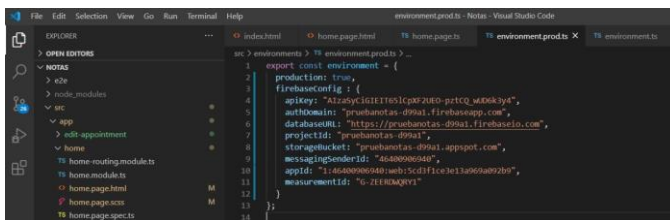


Imagen 1

3. Agrega el siguiente código dentro del environment.ts archivo.

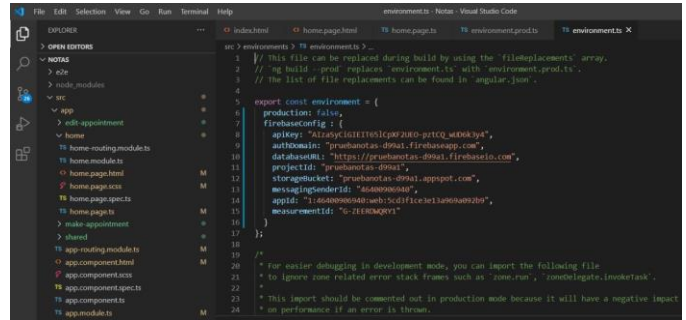


Imagen 2

4. Ahora importaremos y registraremos los servicios de Firebase y el objeto de entorno en el archivo app.module.ts .

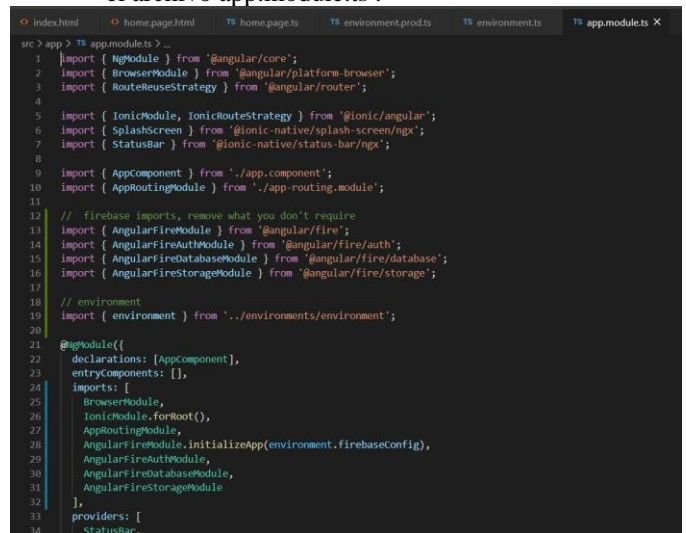


Imagen 3

5. Para navegar en nuestra aplicación Ionic Firebase necesitamos configurar rutas.

Abra el src/app/app-routing.ts archivo y agregue el siguiente código para configurar las rutas en la aplicación Ionic / Angular.

```

1 import { NgModule } from '@angular/core';
2 import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   { path: '', redirectTo: 'home', pathMatch: 'full' },
6   { path: 'home', loadChildren: () => import('./home/home.module').then( m => m.HomePageModule) },
7   {
8     path: 'make-appointment',
9     loadChildren: () => import('./make-appointment/make-appointment.module').then( m => m.MakeAppointmentPageModule)
10  },
11   {
12     path: 'edit-appointment/:id',
13     loadChildren: () => import('./edit-appointment/edit-appointment.module').then( m => m.EditAppointmentPageModule)
14   },
15 ];
16
17 @NgModule({
18   imports: [
19     RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
20   ],
21   exports: [RouterModule]
22 })
23 export class AppRoutingModule { }

```

Imagen 4

6. A continuación, abra el app.component.html archivo y agregue ion-router-outlet junto con las pestañas inferiores iónicas para activar los elementos de navegación.

```

1 <ion-app>
2   <ion-router-outlet> </ion-router-outlet>
3 </ion-app>
4
5 <ion-tabs>
6   <ion-tab-bar slot="bottom">
7     <ion-tab-button routerLinkActive="tab-selected" routerLink="/home" class="ion" tab="home">
8       <ion-icon name="school-outline"> </ion-icon>
9       <ion-label>Notas</ion-label>
10    </ion-tab-button>
11
12    <ion-tab-button routerLinkActive="tab-selected" routerLink="/make-appointment" class="ion" tab="make-appointment">
13      <ion-icon name="add-circle"> </ion-icon>
14      <ion-label>Agregar notas</ion-label>
15    </ion-tab-button>
16  </ion-tab-bar>
17 </ion-tabs>

```

Imagen 5

7. Cree una carpeta compartida y luego cree un Appointment.ts archivo dentro de ella.

```

1 export class Appointment {
2   $key: string;
3   nombre: string;
4   materia: string;
5   calificacion: number;
6 }

```

Imagen 6

8. Luego, incluya el siguiente código dentro de shared / quote.service.ts

```

1 @Injectable({
2   providedIn: 'root'
3 })
4
5 export class AppointmentService {
6   bookingListRef: AngularFirestore<any>;
7   bookingRef: AngularFirestore<any>;
8
9   constructor(private db: AngularFirestore) { }
10
11   // Create
12   createBooking(apt: Appointment) {
13     return this.bookingListRef.push({
14       nombre: apt.nombre,
15       materia: apt.materia,
16       calificacion: apt.calificacion
17     })
18   }
19
20   // Get Single
21   getBooking(id: string) {
22     this.bookingRef = this.db.object('/notas/' + id);
23     return this.bookingRef;
24   }
25
26   // Get List
27   getBookingList() {
28     this.bookingListRef = this.db.list('/notas');
29     return this.bookingListRef;
30   }
31
32   // Update
33   updateBooking(id, apt: Appointment) {
34     return this.bookingRef.update({
35

```

Imagen 7

```

36     // Get Single
37     getBooking(id: string) {
38       this.bookingRef = this.db.object('/notas/' + id);
39       return this.bookingRef;
40     }
41
42     // Get List
43     getBookingList() {
44       this.bookingListRef = this.db.list('/notas');
45       return this.bookingListRef;
46     }
47
48     // Update
49     updateBooking(id, apt: Appointment) {
50       return this.bookingRef.update({
51         nombre: apt.nombre,
52         materia: apt.materia,
53         calificacion: apt.calificacion
54       })
55     }
56
57     // Delete
58     deleteBooking(id: string) {
59       this.bookingRef = this.db.object('/notas/' + id);
60       this.bookingRef.remove();
61     }
62   }

```

Imagen 8

9. Para trabajar con formularios debe importar y registrar ReactiveFormsModule en el archivo make-quote.module.ts

```

src > app > make-appointment > TS make-appointment.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
4
5 import { IonicModule } from '@ionic/angular';
6
7 import { MakeAppointmentPageRoutingModule } from './make-appointment-routing.module';
8
9 import { MakeAppointmentPage } from './make-appointment.page';
10
11 @NgModule({
12   imports: [
13     CommonModule,
14     FormsModule,
15     IonicModule,
16     ReactiveFormsModule,
17     MakeAppointmentPageRoutingModule
18   ],
19   declarations: [MakeAppointmentPage]
20 })
21 export class MakeAppointmentPageModule {}
22

```

Imagen 9

10. Agregue el siguiente código en el archivo make-appointment.page.ts

```

src > app > make-appointment > TS make-appointment.page.ts > ...
8 templateUrl: './make-appointment.page.html',
9 styleUrls: ['./make-appointment.page.scss'],
10 })
11
12 export class MakeAppointmentPage implements OnInit {
13   bookingForm: FormGroup;
14
15   constructor(
16     private aptService: AppointmentService,
17     private router: Router,
18     public fb: FormBuilder
19   ) {}
20
21   ngOnInit() {
22     this.bookingForm = this.fb.group({
23       nombre: [''],
24       materia: [''],
25       calificacion: ['']
26     })
27   }
28
29   formSubmit() {
30     if (!this.bookingForm.valid) {
31       return false;
32     } else {
33       this.aptService.createBooking(this.bookingForm.value).then(res => {
34         console.log(res)
35         this.bookingForm.reset();
36         this.router.navigate(['/home']);
37       })
38       .catch(error => console.log(error));
39     }
40   }
41 }

```

Imagen 10

11. Abra el make-appointment.page.html archivo y agregue el siguiente código.

```

src > app > make-appointment > TS make-appointment.page.html > ...
1 <ion-header>
2   <ion-toolbar class="ios hydrated">
3     <h1>Calificaciones</h1>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <ion-list lines="full">
9     <form [formgroup]="bookingForm" (ngsubmit)="formSubmit()">
10      <ion-item>
11        <ion-label position="floating">Nombre</ion-label>
12        <ion-input formControlName="nombre" type="text" required</ion-input>
13      </ion-item>
14      <ion-item>
15        <ion-label position="floating">Materia</ion-label>
16        <ion-input formControlName="materia" type="text" required</ion-input>
17      </ion-item>
18      <ion-item>
19        <ion-label position="floating">Calificación</ion-label>
20        <ion-input formControlName="calificacion" type="text" required</ion-input>
21      </ion-item>
22      <ion-row>
23        <ion-col>
24          <ion-button type="submit" color="primary" shape="full" expand="block">Agregar Calificación</ion-button>
25        </ion-col>
26      </ion-row>
27    </form>
28  </ion-list>
29 </ion-content>

```

Imagen 11

12. Abra el home.page.ts archivo y agregue el siguiente código

```

src > app > home > TS home.page.ts > ...
11 export class HomePage implements OnInit {
12   bookings = [];
13
14   constructor(
15     private aptService: AppointmentService
16   ) {}
17
18   ngOnInit() {
19     this.fetchBookings();
20     let bookingRes = this.aptService.getBookingList();
21     bookingRes.snapshotChanges().subscribe(res => {
22       this.Bookings = [];
23       res.forEach(item => {
24         let a = item.payload.toJSON();
25         a['$key'] = item.key;
26         this.Bookings.push(a as Appointment);
27       })
28     })
29   }
30
31   fetchBookings() {
32     this.aptService.getBookingList().valueChanges().subscribe(res => {
33       console.log(res)
34     })
35   }
36
37   deleteBooking(id) {
38     console.log(id)
39     if (window.confirm('Do you really want to delete?')) {
40       this.aptService.deleteBooking(id)
41     }
42   }
43 }

```

Imagen 12

13. Para mostrar elementos, abra el archivo home.page.html e inserte el siguiente código en él.

```

src > app > home > home.page.html > ...
4 <h1>Calificaciones</h1>
5 </ion-title>
6 </ion-toolbar>
7 </ion-header>
8
9 <ion-content>
10   <ion-list class="ios list-ios hydrated">
11     <ion-list-header class="ios hydrated">
12       <h2>Estudiantes</h2>
13     </ion-list-header>
14
15     <ion-item *ngfor="let booking of Bookings" class="user-list">
16       <ion-label>
17         <div>
18           <ion-icon name="person"></ion-icon> Estudiante: {{booking.nombre}}
19         </div>
20         <div>
21           <ion-icon name="school-outline"></ion-icon> Materia: {{booking.materia}}
22         </div>
23         <div>
24           <ion-icon name="school-outline"></ion-icon> Calificación: {{booking.calificacion}} /10
25         </div>
26       </ion-label>
27
28       <div class="item-note" item-end>
29         <button ion-button clear [routerLink]="['/edit-appointment/', booking.$key]">
30           <ion-icon name="create" style="zoom:2.0;"></ion-icon>
31         </button>
32         <button ion-button clear (click)="deleteBooking(booking.$key)">
33           <ion-icon name="trash" style="zoom:2.0;"></ion-icon>
34         </button>
35       </div>
36     </ion-item>
37   </ion-list>

```

Imagen 13

14. Para editar el elemento, importe AppointmentService y luego inyecte en la clase constructora. Importe el ReactiveFormsModule en editappointment.module.ts archivo.

```

src > app > edit-appointment > TS edit-appointment.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
4
5 import { IonicModule } from '@ionic/angular';
6
7 import { EditAppointmentPageRoutingModule } from './edit-appointment-routing.module';
8
9 import { EditAppointmentPage } from './edit-appointment.page';
10
11 @NgModule({
12   imports: [
13     CommonModule,
14     FormsModule,
15     IonicModule,
16     ReactiveFormsModule,
17     EditAppointmentPageRoutingModule
18   ],
19   declarations: [EditAppointmentPage]
20 })
21 export class EditAppointmentPageModule {}
22

```

Imagen 14

15. A continuación, inserte el siguiente código en el edit-appointment.page.ts archivo.

```

src > app > edit-appointment > TS edit-appointment.page.ts > ...
11
12 export class EditAppointmentPage implements OnInit {
13   updateBookingForm: FormGroup;
14   id: any;
15
16   constructor(
17     private aptService: AppointmentService,
18     private actRoute: ActivatedRoute,
19     private router: Router,
20     public fb: FormBuilder
21   ) {
22     this.id = this.actRoute.snapshot.paramMap.get('id');
23     this.aptService.getBooking(this.id).valueChanges().subscribe(res => {
24       this.updateBookingForm.setValue(res);
25     });
26   }
27
28   ngOnInit() {
29     this.updateBookingForm = this.fb.group({
30       nombre: [''],
31       materia: [''],
32       calificacion: ['']
33     });
34     console.log(this.updateBookingForm.value)
35   }
36
37   updateForm() {
38     this.aptService.updateBooking(this.id, this.updateBookingForm.value)
39       .then(() => {
40         this.router.navigate(['/home']);
41       })
42       .catch(error => console.log(error));
43   }
44 }

```

Imagen 15

16. Para mostrar el formulario de edición, agregue el siguiente código HTML iónico en el edit-appointment.page.html archivo.

```

src > app > edit-appointment > edit-appointment.page.html > edit-appointment.page.html
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>
4       <h1>Editar Calificaciones</h1>
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content>
10   <ion-list lines="full">
11     <form [formGroup]="updateBookingForm" (ngSubmit)="updateForm()">
12       <ion-item>
13         <ion-label position="floating">Nombre</ion-label>
14         <ion-input formControlName="nombre" type="text" required</ion-input>
15       </ion-item>
16       <ion-item>
17         <ion-label position="floating">Materia</ion-label>
18         <ion-input formControlName="materia" type="text" required>
19       </ion-input>
20     </ion-item>
21     <ion-item>
22       <ion-label position="floating">Calificación</ion-label>
23       <ion-input formControlName="calificacion" type="text" required>
24     </ion-input>
25   </ion-item>
26   <ion-row>
27     <ion-col>
28       <ion-button type="submit" color="primary" shape="full" expand="block">Calificación Editada</ion-button>
29     </ion-col>
30   </ion-row>
31 </form>
32 </ion-content>
33 </ion-page>
34

```

Imagen 16

17. Iniciamos la aplicación y observamos los resultados

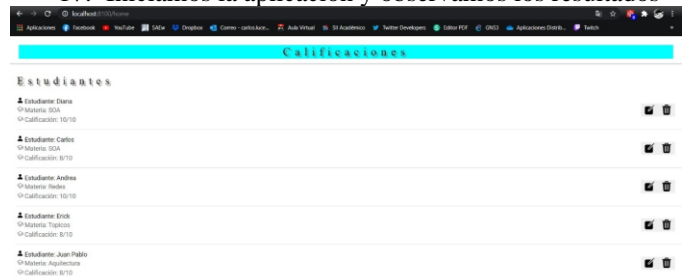


Imagen 17



Imagen 18



Imagen 19

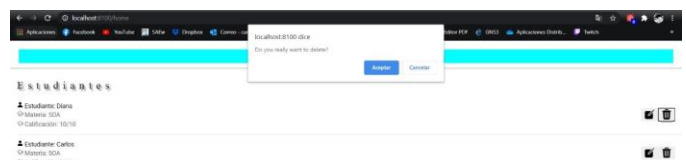
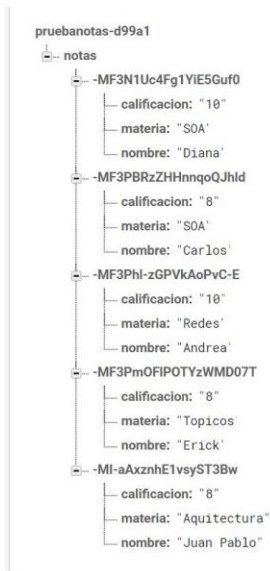


Imagen 20

18. Resultados guardados en nuestra base de datos de Firebase.





*Imagen 21*

[1]

## V. CONCLUSIÓN

Enos aprendido el cómo crear una aplicación móvil CRUD con Angular, a la vez nos enfocamos en como configurar en Ionic y angular, cómo definir operaciones CRUD en el servicio de Angular y finalmente verificamos cómo consumir las API REST de Firebase para realizar la operación CRUD

## VI. REFERENCIAS

- [1] Digamber, «positronX.io,» positronX.io, 08 Agosto 2020. [En línea]. Available: <https://www.positronx.io/build-ionic-firebase-crud-app-with-angular/>. [Último acceso: 24 Septiembre 2020].