

Se incluyen un input y cuatro botones más, el input funciona como buscador de avalúos junto con el primer botón, el segundo botón proporciona la ubicación actual, el tercer botón reinicia cualquier búsqueda y el tercero redirecciona al histórico de avalúos.

- adminmap-ciduadmin.ts

```

1 import { Component, EventEmitter, OnDestroy, OnInit, Output } from '@angular/core'; 115k (gzipped: 36.2k)
2 import { FormControl, ReactiveFormsModule } from '@angular/forms'; 65.5k (gzipped: 14.4k)
3 import { FormGroup, FormBuilder, Validators } from '@angular/forms';
4 import { Router } from '@angular/router'; 97k (gzipped: 24.7k)
5 import { AngularFireAuth } from '@angular/fire/compat/auth'; 281.8k (gzipped: 78.8k)
6 import { AngularFirestore } from '@angular/fire/compat/firestore';
7
8 import { Avaluo } from '../models/Avaluo';
9 import { Mapa } from '../models/Mapa';
10 import { MarcadoresService } from '../servicio/marcadores.service';
11 import { CiduadminService } from '../servicio/ciduadmin.service';
12 import { ToastrService } from 'ngx-toastr'; 37.3k (gzipped: 8.1k)
13 import { Subscription } from 'rxjs'; 2.3k (gzipped: 968)
14
15 @Component({
16   selector: 'app-adminmap-ciduadmin',
17   templateUrl: './adminmap-ciduadmin.component.html',
18   styleUrls: ['./adminmap-ciduadmin.component.css']
19 })
20
21 export class AdminmapCiduadminComponent implements OnInit, OnDestroy {
22
23   lat: number;
24   lng: number;
25   zoom: number;
26   mapTypeId: string;
27   logo:string = '../assets/Pictures/LogoNombre.png';
28   ubi:string = '../assets/Pictures/ubi.png';
29   ubipro:string = '../assets/Pictures/ubipro.png';
30   his:string = '../assets/Pictures/His.png';
31   raisal:string = '../assets/Pictures/searchraisal.png';
32   iconMap = '../assets/Pictures/markerHouse.png';

```

```

33   located: boolean = false;
34   animation: string;
35   zoomControl: boolean;
36   scaleControl: boolean;
37   mapTypeControl: boolean;
38   fullscreenControl: boolean;
39   panControl: boolean;
40   Marcadores:any;
41
42   historico: Avaluo[] = [];
43   avaluo: Mapa[] = [];
44   tablas: any = [];
45   loading = false;
46
47   formMapa: FormGroup;
48   idavaluo: string = '';
49   latitud: string = '';
50   longitud: string = '';
51
52   suscriptionUser: Subscription = new Subscription();
53
54   constructor(private marcadoresService:MarcadoresService,
55               private fb: FormBuilder,
56               private router: Router,
57               private toastr: ToastrService,
58               private afAuth: AngularFireAuth,
59               private _firestore: AngularFirestore,
60               private _ciduadmin: CiduadminService,) {
61     this.lat = 4.656453;
62     this.lng = -74.122186;
63     this.zoom = 10.7;
64     this.mapTypeId = 'roadmap';

```

```

65     this.animation = "'BOUNCE'";
66     this.located = false;
67     this.zoomControl = true;
68     this.scaleControl = true;
69     this.mapTypeControl = true;
70     this.fullscreenControl = true;
71     this.panControl = true;
72     this.iconMap = '../assets/Pictures/markerHouse.png';
73     this.formMapa=this.fb.group({
74         idavaluo: ['', Validators.required],
75         latitud: ['', Validators.required],
76         longitud: ['', Validators.required],
77     });
78
79 }
80
81 ngOnInit(): void {
82
83     this.suscriptionUser = this.afAuth.user.subscribe(user =>{
84         if(user){
85
86             }else{
87                 this.router.navigate(['/']);
88             }
89     })
90     this.loading = true;
91     this._ciduadmin.historicoAvaluos().subscribe(data =>{
92         this.loading = false;
93         this.historico = [];
94         data.forEach((element: any) => {
95             this.historico.push({
96                 id: element.pavload.doc.id.

```

```

97                 ...element.payload.doc.data()
98             })
99         });
100     }
101 }
102
103 );
104 }
105
106 ngOnDestroy(): void {
107     this.suscriptionUser.unsubscribe();
108 }
109
110 getOriginalPosition(): void{
111     navigator.geolocation.getCurrentPosition(position =>{
112         this.lat = 4.656453;
113         this.lng = -74.122186;
114         this.zoom = 10.7;
115         this.mapTypeId = 'hybrid';
116         this.located = false;
117         this.animation = "'BOUNCE'";
118         this.iconMap = '../assets/Pictures/markerHouse.png';
119     })
120 }
121
122 getCurrentPosition(): void{
123     navigator.geolocation.getCurrentPosition(position => {
124         this.lat = position.coords.latitude;
125         this.lng = position.coords.longitude;
126         this.zoom = 17;
127         this.located = true;
128         this.iconMap = '../assets/Pictures/person.png';

```

```

133 SearchAppraisal() {
134   const mapa : Mapa = {
135     idavaluo: this.FormMapa.get('idavaluo')?.value,
136     latitud: 0,
137     longitud: 0
138   };
139   let resp:string[] = [
140
141   ];
142
143   this.ciduadmin.obtenerAvaluo().then(resp => {
144     this.tablas = resp;
145     //console.log(resp, 'resp');
146     for(let idavaluo of Object.keys(this.tablas)){
147       const codigos1 = this.tablas[idavaluo];
148       //console.log(codigos1);
149       for (let idavaluo of Object.keys(codigos1)){
150         const codigos2 = codigos1[idavaluo];
151         //console.log(idavaluo,":",codigos2);
152
153         if(mapa.idavaluo === codigos2 || mapa.latitud === codigos2 || mapa.longitud === codigos2 ){
154           console.log(codigos2, mapa.idavaluo);
155           this.avaluo = [];
156           this.avaluo.push({ ...codigos2 });
157           //console.log(this.avaluo);
158           navigator.geolocation.getCurrentPosition(position => {
159             this.lat = codigos2.latitud;
160             this.lng = codigos2.longitud;
161             //console.log(this.lat, this.lng);
162             this.zoom = 17;
163             this.located = false;
164           })

```

En el TS de la pagina Mapa de Avalúos, Se captan los datos guardados en la base de datos de Firestore y los almacena en un array para poder mostrarlos en el proceso anterior, los métodos para cada botón se pueden visualizar.

### Modulo Histórico de Avalúos.

- adminbackground-ciduadmin.html

```

1 <app-navbar-principal></app-navbar-principal>
2 <body class="fondo">
3   <section class="px-5" id="section1">
4     <br>
5     <div class="row align-items-center">
6       <div class="col" id="col1">
7         <img [src]="logo" id="logocidu" alt="...">
8       </div>
9     </div>
10    <br>
11  </section>
12  <section>
13    <div class="container" id="titulo">
14      <p id="h1">HISTORICO DE AVALÚOS</p>
15    </div>
16  </section>
17  <div *ngIf="loading" class="d-flex justify-content-center mt-5">
18    <app-spinner></app-spinner>
19  </div>
20  <br>
21  <div *ngIf="!loading" class="container-fluid horizontal-scrollable">
22    <table class="table text-white table-hover">
23      <thead>
24        <tr>
25          <th scope="col">
26            #
27          </th>
28          <th scope="col">
29            ESTADO
30          </th>
31          <th scope="col">
32            ESTADO

```

```

31 <th scope="col">
32 | CLIENTE
33 </th>
34 <th scope="col">
35 | MES
36 </th>
37 <th scope="col">
38 | CÓDIGO AVALÚO
39 </th>
40 <th scope="col">
41 | TIPO DE DOCUMENTO
42 </th>
43 <th scope="col">
44 | NUMERO DE DOCUMENTO
45 </th>
46 <th scope="col">
47 | NOMBRE CLIENTE
48 </th>
49 <th scope="col">
50 | DIRECCIÓN
51 </th>
52 <th scope="col">
53 | DETALLES
54 </th>
55 <th scope="col">
56 | CONJUNTO/BARRIO
57 </th>
58 <th scope="col">
59 | CIUDAD
60 </th>

```

```

61 <th scope="col">
62 | CONTACTO
63 </th>
64 <th scope="col">
65 | TEL CONTACTO
66 </th>
67 <th scope="col">
68 | FECHA DE ASIGNACIÓN
69 </th>
70 <th scope="col">
71 | FECHA DE AGENDAMIENTO
72 </th>
73 <th scope="col">
74 | FECHA DE VISITA
75 </th>
76 <th scope="col">
77 | HORA DE VISITA
78 </th>
79 <th scope="col">
80 | VISITADOR
81 </th>
82 <th scope="col">
83 | PAGO VISITADOR
84 </th>
85 <th scope="col">
86 | OBSERVACIONES DE GESTIÓN
87 </th>
88 <th scope="col">
89 | FUA
90 </th>
91 <th scope="col">
92 | REVISIÓN FUA MERCADO

```

```

94     <th scope="col">
95         FECHA ENVÍO INFORME
96     </th>
97     <th scope="col">
98         VALOR DE AVALÚO
99     </th>
100    <th scope="col">
101        URBANO/RURAL
102    </th>
103    <th scope="col">
104        TIPO DE AVALÚO
105    </th>
106 </tr>
107 </thead>
108 <tbody *ngFor ="let avaluo of historico; let i = index">
109     <tr>
110         <td>{{ i + 1 }}</td>
111         <td>Avalúo {{ avaluo.estado }}</td>
112         <td>{{ avaluo.cliente }}</td>
113         <td>{{ avaluo.mes }}</td>
114         <td>{{ avaluo.idavaluo }}</td>
115         <td>{{ avaluo.tipodocumento }}</td>
116         <td>{{ avaluo.identificacion }}</td>
117         <td>{{ avaluo.nombreCliente }}</td>
118         <td>{{ avaluo.ubicacion }}</td>
119         <td>{{ avaluo.detallesDir }}</td>
120         <td>{{ avaluo.barrío }}</td>
121         <td>{{ avaluo.ciudadAvaluo }}</td>
122         <td>{{ avaluo.contacto }}</td>
123         <td>{{ avaluo.telefonoContacto }}</td>
124         <td>{{ avaluo.fechaAsignacion }}</td>
125         <td>{{ avaluo.fechaAgendamiento }}</td>

```

```

126         <td>{{ avaluo.fechaVisita }}</td>
127         <td>{{ avaluo.horaVisita }}</td>
128         <td>{{ avaluo.visitador }}</td>
129         <td>{{ avaluo.pagoVisitador }}</td>
130         <td>{{ avaluo.observaciones }}</td>
131         <td>{{ avaluo.fua }}</td>
132         <td>{{ avaluo.revisionFua }}</td>
133         <td>{{ avaluo.fechaEnvio }}</td>
134         <td>{{ avaluo.valorAvaluo }}</td>
135         <td>{{ avaluo.sitio }}</td>
136         <td>{{ avaluo.tipoAvaluo }}</td>
137     </tr>
138 </tbody>
139 </table>
140 </div>
141
142 <div class="container-fluid">
143     <button class="fa fa-file-excel-o boton" type="button" (click)="exportExcel()" placement="top" ngbTooltip="Generar Reporte de Avalúos."></button>
144 </div>
145 </body>
146 <app-footer></app-footer>
147

```

Se visualiza el código fuente de la página Histórico de Avalúos de la aplicación web CIDUadmin, aquí se observa mediante una tabla todos los avalúos con cada dato que se ha agregado además de un botón que permite descargar un reporte en Excel.

## - adminbackground-ciduadmin.ts

```

1 import { Component, OnDestroy, OnInit } from '@angular/core'; 115k (gzipped: 36.2k)
2 import { AngularFireAuth } from '@angular/fire/compat/auth'; 281.8k (gzipped: 78.8k)
3 import { Router } from '@angular/router'; 97k (gzipped: 24.7k)
4 import { Avaluo } from '../models/Avaluo';
5 import { CiduadminService } from '../servicio/ciduadmin.service';
6 import * as FileSaver from 'file-saver'; 3.4k (gzipped: 1.5k)
7 import { Subscription } from 'rxjs'; 2.3k (gzipped: 968)
8
9 @Component({
10   selector: 'app-adminbackground-ciduadmin',
11   templateUrl: './adminbackground-ciduadmin.component.html',
12   styleUrls: ['./adminbackground-ciduadmin.component.css']
13 })
14 export class AdminbackgroundCiduadminComponent implements OnInit, OnDestroy {
15
16   historico: Avaluo[] = [];
17   loading = false;
18
19   suscriptionUser: Subscription = new Subscription();
20   logo:string = '../assets/Pictures/LogoNombre.png';
21
22   constructor(private _ciduadmin: CiduadminService
23               ,private afAuth: AngularFireAuth
24               ,private router: Router) {
25   }
26
27   ngOnInit(): void {
28     this.suscriptionUser = this.afAuth.user.subscribe(user =>{
29       if(user){
30
31       }else{
32         this.router.navigate(['/']);

```

```

34   })
35   this.loading = true;
36   this._ciduadmin.historicoAvaluos().subscribe(data =>{
37     this.loading = false;
38     this.historico = [];
39     data.forEach((element: any) => {
40       this.historico.push({
41         id: element.payload.doc.id,
42         ...element.payload.doc.data()
43       })
44     });
45     console.log(this.historico);
46   }
47   );
48 }
49
50
51
52 ngOnDestroy(): void {
53   this.suscriptionUser.unsubscribe();
54 }
55

```

```

56 exportExcel(){
57   if (this.historico.length > 0) {
58     import("xlsx").then(xlsx => { 420.8k (gzipped: 139k)
59       const myHeader = ["estado", "cliente", "mes", "idavaluo", "tipodocumento", "identificacion", "nombreCliente", "ubicacion", "detallesDir", "barrio",
        "ciudadAvaluo", "contacto", "telefonoContacto", "fechaAsignacion", "fechaAgendamiento", "fechaVisita", "horaVisita", "visitador", "pagoVisitador",
        "observaciones", "fua", "revisioFua", "fechaEnvio", "valorAvaluo", "sitio", "tipoAvaluo", "id", "uid", "direccion", "ciudad", "departamento", "latitud",
        "longitud"];
60       const worksheet = xlsx.utils.json_to_sheet(this.historico, {header: myHeader});
61       const workbook = { Sheets: { 'AVALUOS': worksheet }, SheetNames: ['AVALUOS'] };
62
63       const excelBuffer: any = xlsx.write(workbook, { bookType: 'xlsx', type: 'array' });
64
65       this.saveAsExcelFile(excelBuffer, "Reporte Avaluos");
66     });
67   }
68 }
69 saveAsExcelFile(buffer: any, fileName: string): void {
70   let EXCEL_TYPE = 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet;charset=UTF-8';
71   let EXCEL_EXTENSION = '.xlsx';
72   const data: Blob = new Blob([buffer], {
73     type: EXCEL_TYPE
74   });
75   FileSaver.saveAs(data, fileName + '_extraido_' + new Date() + EXCEL_EXTENSION);
76 }

```

En el TS de la página Histórico de Avalúos, Se captan los datos guardados en la base de datos de Firestore y los almacena en un array para poder mostrarlos en el proceso anterior. Además, de incluir el método de generar el reporte en Excel gracias a la biblioteca llamada fileSaver.