**LABORATORIO 2.2**

**Integrantes:**

* Jairo González
* Christian Jimbo
* Jorge Sarmiento

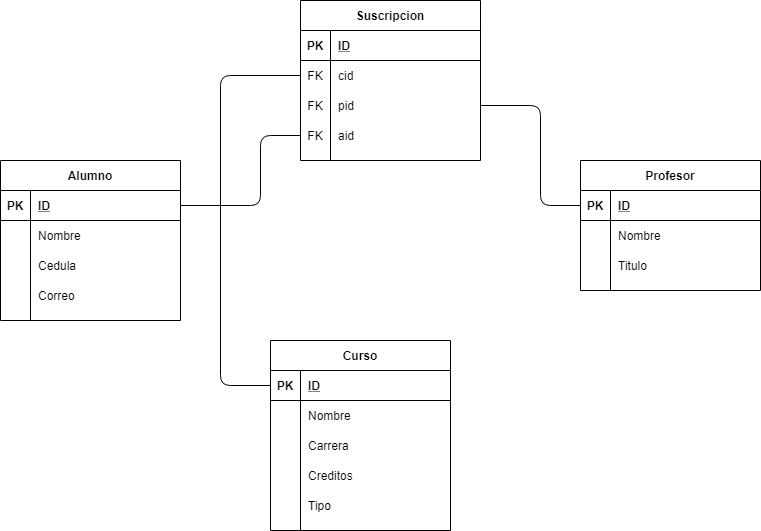
**Enunciado:** Desarrollar una aplicación que conecte a base de datos y permita su descarga en distintos formatos. Pueden utilizar la base de datos creada en el primer bimestre y sobre ella desarrollar la aplicación.

**Entorno de desarrollo:**

* Angular
* Firebase (Cloud Firestore)
* Typescript

**Desarrollo**

ESQUEMA BASE DE DATOS



En este esquema, se representa un sistema de suscripciones a cursos, donde se considera a un alumno inscrito a un curso y un profesor que dará la materia.

CÓDIGO

**Servicio de conexión:**

Se crea un servicio conexion.ts, el cuál inicializa las listas auxiliares llamando la información desde Firestore. Se utiliza el tipo de dato Observable<##>. En el espacio designado a ## se inyecta la interfaz a la que pertenecerá la información.

Ej:

alumnos: Observable<Alumno[]>;

Dado que la información no es de archivos individuales, se debe cargar en un array.

Se hace una referencia a la Collection de Firestore a través de alumnosCollection: AngularFirestoreCollection<Alumno>;

Se hace referencia a la interfaz para la recolección de los documentos

Se llama a la Collection desde la base de datos a través de AngularFirestore y la referencia de la Collection previamente declarada:

    this.alumnosCollection = db.collection<Alumno>('alumnos');

Se carga la información desde la collection a través de SnapshotChanges para tener control a tiempo real de los cambios en la base de datos. Posterior a eso se mapea la lista convirtiendo la información obtenida a un formato JSON, incluyendo el id del documento traido.

this.alumnos = this.alumnosCollection.snapshotChanges().pipe(

      map(actions => actions.map(a => {

        const data = a.payload.doc.data() as any;

        const id = a.payload.doc.id;

        return { id, ...data };

      })));

Se devuelve las listas a través de un método que será captado en app.component.ts

En el documento del componente se crea listas únicas de tipo any para captar la información en bruto traída del servicio de conexión.

 alumnos: any;

Al traer la información bruta se utiliza el método suscribe para convertir esa información al esquema de la clase correspondiente:

this.alumnos = conexion.listarAlumnos().subscribe(alumno => {

      this.alumnos = alumno;

      console.log(this.alumnos);

    });

Esto mapea la información y normaliza la información a un array de objetos.

Se presenta en pantalla

<div>

    <h1>Alumnos</h1>

    <table class="table">

        <thead class="thead-dark">

            <tr>

                <th scope="col">ID</th>

                <th scope="col">Nombre</th>

                <th scope="col">Cedula</th>

                <th scope="col">Correo</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let alumno of alumnos">

                <th scope="row">{{alumno.id}}</th>

                <td>{{alumno.nombre}}</td>

                <td>{{alumno.cedula}}</td>

                <td>{{alumno.correo}}</td>

            </tr>

        </tbody>

    </table>

    <div class="btn\_container">

        <h3>Descargar registro de alumnos</h3>

        <button type="button" class="btn btn-primary" (click)="AlumnoJSON()">

            JSON

        </button>

        <button type="button" class="btn btn-success" (click)="AlumnoTXT()">

            TXT

        </button>

        <button type="button" class="btn btn-danger" (click)="AlumnoCSV()">

            CSV

        </button>

        <button type="button" class="btn btn-warning" (click)="AlumnoXLS()">

            XLS

        </button>

    </div>

    <div>

    </div>

</div>

DESCARGA DE ARCHIVOS

Se utiliza la librería

import exportFromJSON from 'export-From-json';

la cuál a través de tres parámetros permite descargar la data a diferentes formatos.

Ej: descargar a JSON:

AlumnoJSON() {

    const data = this.alumnos;

    const fileName = 'Alumnos';

    const exportType = 'json';

    exportFromJSON({ data, fileName, exportType });

  }

A considerar, se debe tener una estructura JSON compatible para los 4 formatos soportados, ya que para csv y xls, un json dinámico no es compatible.

ANEXOS

conexion.service.ts

import { Injectable } from '@angular/core';

import { AngularFirestore, AngularFirestoreCollection, AngularFirestoreModule } from '@angular/fire/firestore';

import { map, switchMap } from 'rxjs/operators';

import { Observable, combineLatest, of } from 'rxjs';

import { Alumno } from '../../Interfaces/Alumno';

import { Curso } from '../../Interfaces/Curso';

import { Profesor } from '../../Interfaces/Profesor';

import { Subscripcion } from '../../Interfaces/Subscripcion';

import { uniq, flatten } from 'lodash';

@Injectable({

  providedIn: 'root'

})

export class ConexionService {

  private alumnosCollection: AngularFirestoreCollection<Alumno>;

  alumnos: Observable<Alumno[]>;

  private aux: AngularFirestoreCollection<Subscripcion>;

  auxAux: Observable<Subscripcion[]>;

  private cursoCollection: AngularFirestoreCollection<Curso>;

  cursos: Observable<Curso[]>

  private profesorCollection: AngularFirestoreCollection<Profesor>;

  profesores: Observable<Profesor[]>;

  subs: Observable<any>;

  constructor(db: AngularFirestore) {

    this.aux = db.collection<Subscripcion>('suscripciones');

    this.auxAux = this.aux.snapshotChanges().pipe(

      map(actions => actions.map(a => {

        const data = a.payload.doc.data() as Subscripcion;

        const id = a.payload.doc.id;

        return { id, ...data };

      })));

    this.alumnosCollection = db.collection<Alumno>('alumnos');

    this.alumnos = this.alumnosCollection.snapshotChanges().pipe(

      map(actions => actions.map(a => {

        const data = a.payload.doc.data() as any;

        const id = a.payload.doc.id;

        return { id, ...data };

      })));

    this.cursoCollection = db.collection<Curso>('cursos');

    this.cursos = this.cursoCollection.snapshotChanges().pipe(

      map(actions => actions.map(a => {

        const data = a.payload.doc.data() as Curso;

        const id = a.payload.doc.id;

        return { id, ...data };

      })));

    this.profesorCollection = db.collection<Profesor>('profesores');

    this.profesores = this.profesorCollection.snapshotChanges().pipe(

      map(actions => actions.map(a => {

        const data = a.payload.doc.data() as Profesor;

        const id = a.payload.doc.id;

        return { id, ...data };

      })));

    this.subs = combineLatest(

      db.collection<Subscripcion>('suscripciones').valueChanges(),

      db.collection<Alumno>('alumnos').valueChanges(),

      db.collection<Curso>('cursos').valueChanges(),

      db.collection<Profesor>('profesores').valueChanges()

    ).pipe(

      map(([suscripciones, alumnos, cursos, profesores]) => {

        return { suscripciones, alumnos, cursos, profesores };

      })

    )

  }

  listarAuxSub() {

    return this.subs;

  }

  listarProfesores() {

    return this.profesores;

  }

  listarCursos() {

    return this.cursos;

  }

  listarSubs() {

    return this.auxAux;

  }

  listarAlumnos() {

    return this.alumnos;

  }

}

app.component.ts

import { Component, OnInit } from '@angular/core';

import { ConexionService } from './services/conexion.service';

import exportFromJSON from 'export-From-json';

export interface Alumno { nombre: String; cedula: String; correo: String; };

export interface AlumnoId extends Alumno { id: String };

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit {

  header = ['Id', 'Nombre', 'Cedula', 'Correo'];

  title = 'Data Downloader';

  alumnos: any;

  subs: any;

  cursos: any;

  profesores: any;

  auxSub: any;

  constructor(private conexion: ConexionService) {

    this.alumnos = conexion.listarAlumnos().subscribe(alumno => {

      this.alumnos = alumno;

      console.log(this.alumnos);

    });

    this.subs = conexion.listarSubs().subscribe(sub => {

      this.subs = sub;

      console.log(this.subs);

    })

    this.cursos = conexion.listarCursos().subscribe(

      curso => {

        this.cursos = curso;

        console.log(this.cursos);

      }

    )

    this.profesores = conexion.listarProfesores().subscribe(

      profesor => {

        this.profesores = profesor;

        console.log(this.profesores);

      }

    )

    this.auxSub = conexion.listarAuxSub().subscribe(

      subscripcion => {

        this.auxSub = subscripcion;

        console.log(this.auxSub);

      })

  }

  ngOnInit(): void {

    console.log('Funcionando');

  }

  AlumnoJSON() {

    const data = this.alumnos;

    const fileName = 'Alumnos';

    const exportType = 'json';

    exportFromJSON({ data, fileName, exportType });

  }

  AlumnoTXT() {

    const data = this.alumnos;

    const fileName = 'Alumnos';

    const exportType = 'txt';

    exportFromJSON({ data, fileName, exportType });

  }

  AlumnoCSV() {

    const data = this.alumnos;

    const fileName = 'Alumnos';

    const exportType = 'csv';

    exportFromJSON({ data, fileName, exportType });

  }

  AlumnoXLS() {

    const data = this.alumnos;

    const fileName = 'Alumnos';

    const exportType = 'xls';

    exportFromJSON({ data, fileName, exportType });

  }

  CursoJSON() {

    const data = this.cursos;

    const fileName = 'Cursos';

    const exportType = 'json';

    exportFromJSON({ data, fileName, exportType });

  }

  CursoTXT() {

    const data = this.cursos;

    const fileName = 'Cursos';

    const exportType = 'txt';

    exportFromJSON({ data, fileName, exportType });

  }

  CursoCSV() {

    const data = this.cursos;

    const fileName = 'Cursos';

    const exportType = 'csv';

    exportFromJSON({ data, fileName, exportType });

  }

  CursoXLS() {

    const data = this.cursos;

    const fileName = 'Cursos';

    const exportType = 'xls';

    exportFromJSON({ data, fileName, exportType });

  }

  ProfesorJSON() {

    const data = this.profesores;

    const fileName = 'Profesores';

    const exportType = 'json';

    exportFromJSON({ data, fileName, exportType });

  }

  ProfesorTXT() {

    const data = this.profesores;

    const fileName = 'Profesores';

    const exportType = 'txt';

    exportFromJSON({ data, fileName, exportType });

  }

  ProfesorCSV() {

    const data = this.profesores;

    const fileName = 'Profesores';

    const exportType = 'csv';

    exportFromJSON({ data, fileName, exportType });

  }

  ProfesoresXLS() {

    const data = this.profesores;

    const fileName = 'Profesores';

    const exportType = 'xls';

    exportFromJSON({ data, fileName, exportType });

  }

  SubJSON() {

    const data = this.auxSub;

    const fileName = 'Suscripciones';

    const exportType = 'json';

    exportFromJSON({ data, fileName, exportType });

  }

  SubTXT() {

    const data = this.auxSub;

    const fileName = 'Suscripciones';

    const exportType = 'txt';

    exportFromJSON({ data, fileName, exportType });

  }

}

app.component.html

<h1>{{title}}</h1>

<div>

    <h1>Alumnos</h1>

    <table class="table">

        <thead class="thead-dark">

            <tr>

                <th scope="col">ID</th>

                <th scope="col">Nombre</th>

                <th scope="col">Cedula</th>

                <th scope="col">Correo</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let alumno of alumnos">

                <th scope="row">{{alumno.id}}</th>

                <td>{{alumno.nombre}}</td>

                <td>{{alumno.cedula}}</td>

                <td>{{alumno.correo}}</td>

            </tr>

        </tbody>

    </table>

    <div class="btn\_container">

        <h3>Descargar registro de alumnos</h3>

        <button type="button" class="btn btn-primary" (click)="AlumnoJSON()">

            JSON

        </button>

        <button type="button" class="btn btn-success" (click)="AlumnoTXT()">

            TXT

        </button>

        <button type="button" class="btn btn-danger" (click)="AlumnoCSV()">

            CSV

        </button>

        <button type="button" class="btn btn-warning" (click)="AlumnoXLS()">

            XLS

        </button>

    </div>

    <div>

    </div>

</div>

<div>

    <h1>Cursos</h1>

    <table class="table">

        <thead class="thead-dark">

            <tr>

                <th scope="col">ID</th>

                <th scope="col">Nombre</th>

                <th scope="col">Carrera</th>

                <th scope="col">Creditos</th>

                <th scope="col">Tipo</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let curso of cursos">

                <th scope="row">{{curso.id}}</th>

                <td>{{curso.nombre}}</td>

                <td>{{curso.carrera}}</td>

                <td>{{curso.creditos}}</td>

                <td>{{curso.tipo}}</td>

            </tr>

        </tbody>

    </table>

    <div class="btn\_container">

        <h3>Descargar registro de cursos</h3>

        <button type="button" class="btn btn-primary" (click)="CursoJSON()">

            JSON

        </button>

        <button type="button" class="btn btn-success" (click)="CursoTXT()">

            TXT

        </button>

        <button type="button" class="btn btn-danger" (click)="CursoCSV()">

            CSV

        </button>

        <button type="button" class="btn btn-warning" (click)="CursoXLS()">

            XLS

        </button>

    </div>

    <div>

    </div>

</div>

<div>

    <h1>Profesores</h1>

    <table class="table">

        <thead class="thead-dark">

            <tr>

                <th scope="col">ID</th>

                <th scope="col">Nombre</th>

                <th scope="col">Título</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let profesor of profesores">

                <th scope="row">{{profesor.id}}</th>

                <td>{{profesor.nombre}}</td>

                <td>{{profesor.titulo}}</td>

            </tr>

        </tbody>

    </table>

    <div class="btn\_container">

        <h3>Descargar registro de profesores</h3>

        <button type="button" class="btn btn-primary" (click)="CursoJSON()">

            JSON

        </button>

        <button type="button" class="btn btn-success" (click)="CursoTXT()">

            TXT

        </button>

        <button type="button" class="btn btn-danger" (click)="CursoCSV()">

            CSV

        </button>

        <button type="button" class="btn btn-warning" (click)="CursoXLS()">

            XLS

        </button>

    </div>

    <div>

    </div>

</div>

<div>

    <h1>Sucripciones</h1>

    <table class="table">

        <thead class="thead-dark">

            <tr>

                <th scope="col">ID Curso</th>

                <th scope="col">ID Profesor</th>

                <th scope="col">ID Alumno</th>

            </tr>

        </thead>

        <tbody>

            <tr \*ngFor="let sub of subs">

                <th scope="row">{{sub.cid}}</th>

                <td>{{sub.pid}}</td>

                <td>{{sub.aid}}</td>

            </tr>

        </tbody>

    </table>

    <div class="btn\_container">

        <h3>Descargar todos los registros</h3>

        <button type="button" class="btn btn-primary" (click)="SubJSON()">

            JSON

        </button>

        <button type="button" class="btn btn-success" (click)="SubTXT()">

            TXT

        </button>

    </div>

    <div>

    </div>

</div>

Interfaces

export interface Subscripcion{

     id:any;

     aid: any;

     cid: any;

     pid: any;

}

export interface Profesor {

     id: String;

     nombre: String;

     titulo: String;

}

export interface Curso {

id: String;

     nombre: String;

     carrera: String;

     creditos: Number;

     tipo:String;

}

export interface Alumno {

     id: String;

     nombre: String;

     cedula: String;

     correo: String;

}

index.html

<!doctype html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>DataDownloader</title>

  <base href="/">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="icon" type="image/x-icon" href="favicon.ico">

  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

    integrity="sha384-9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk" crossorigin="anonymous">

</head>

<body>

  <app-root></app-root>

  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"

    integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"

    crossorigin="anonymous"></script>

  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

    integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"

    crossorigin="anonymous"></script>

  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"

    integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"

    crossorigin="anonymous"></script>

</body>

</html>

environment.ts (Credenciales de Firebase)

export const environment = {

  production: false,

  firebase: {

    apiKey: "AIzaSyCYsQ6enhajFHdxm\_UJ4p3dgKdXVq4Z7Os",

    authDomain: "file-downloader-ca4db.firebaseapp.com",

    databaseURL: "https://file-downloader-ca4db.firebaseio.com",

    projectId: "file-downloader-ca4db",

    storageBucket: "file-downloader-ca4db.appspot.com",

    messagingSenderId: "166017984627",

    appId: "1:166017984627:web:85e7b06fc2287528106206",

    measurementId: "G-GY12G4T6RS"

  }

};

app.module.ts

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { AngularFireModule } from '@angular/fire';

import { environment } from '../environments/environment';

import { AngularFirestoreModule } from '@angular/fire/firestore';

import { ConexionService } from './services/conexion.service';

@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    AngularFireModule.initializeApp(environment.firebase),

    AppRoutingModule,

    AngularFirestoreModule

  ],

  providers: [

    ConexionService

  ],

  bootstrap: [AppComponent]

})

export class AppModule { }

MUESTRA

