



**TECNOLÓGICO
NACIONAL DE MÉXICO**



Tecnológico Nacional de México

Instituto Tecnológico de Tijuana

Subdirección Académica

Departamento de Sistemas y Computación

Ing. Informática

Inteligencia de Negocios

Documentación Examen Ionic

Docente: Juan Manuel Hernandez Martinez

Alumno: Aldo Efren Cuellar Ochoa – 16212347

Fecha: 1 de mayo de 2020

Desarrollo de la aplicación Buscador de películas OMDb

1. Se procede a crear el proyecto con el siguiente comando:

```
ionic start BuscadorApp blank --type-angular
```

2. Creamos las siguientes páginas extra y el servicio movie:

```
ionic g page pages/movies
```

```
ionic g page page/movieDetails
```

```
ionic g service services/movie
```

3. Iniciamos la aplicación:

```
ionic serve
```

4. En el archivo “app-routing.module.ts” modificamos la ruta para que se abra directamente la página “movies”

5. Ahora, en el archivo app.module.ts procedemos a importar los módulos de Angular HttpClient de la siguiente manera:

```
import { HttpClientModule } from '@angular/common/http';
```

Y en el @NgModule del mismo archivo lo siguiente:

```
HttpClientModule
```

6. Después, en el archivo de servicio “movie.service.ts” agregamos también el módulo HttpClient y debe quedar de esta manera el resto del código

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { map } from 'rxjs/operators';
```

```
export enum SearchType {
  all = '',
  movie = 'movie',
  series = 'series',
  episode = 'episode'
}
```

```
@Injectable({
  providedIn: 'root'
})
```

```
export class MovieService {
  url = 'http://www.omdbapi.com/';
  apiKey = 'bbd7e13';
```

```
/**
 * @param http
```

```

    */
    constructor(private http: HttpClient) { }

    /**
     * @param {string} title
     * @param {SearchType} type
     * @returns
     */
    searchData(title: string, type: SearchType): Observable<any> {
        return
this.http.get(`${this.url}?s=${encodeURIComponent(title)}&type=${type}&apikey=${
this.apiKey}`)
        .pipe(
            map(results => results['Search'])
        );
    }

    /**
     * @param {string} id
     * @returns
     */
    getDetails(id) {
        return
this.http.get(`${this.url}?i=${id}&plot=full&apikey=${this.apiKey}`);
    }
}

```

7. Procedemos ahora en el archivo movies.page.ts a agregar el código siguiente

```

import { MovieService, SearchType } from
'../../services/movie.service';
import { Component, OnInit } from '@angular/core';
import { Observable } from 'rxjs';

@Component({
    selector: 'app-movies',
    templateUrl: './movies.page.html',
    styleUrls: ['./movies.page.scss'],
})
export class MoviesPage implements OnInit {

```

```

results: Observable<any>;
searchTerm: string = '';
type: SearchType = SearchType.all;

/**
 * @param movieService
 */
constructor(private movieService: MovieService) { }

ngOnInit() { }

searchChanged() {
    this.results = this.movieService.searchData(this.searchTerm,
this.type);
}
}

```

8. Para terminar la parte principal y de inicio de la aplicación agregamos el siguiente código al archivo “movies.page.html”

```

<ion-header>
  <ion-toolbar color="dark">
    <ion-title>Buscador de Peliculas</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>

  <ion-searchbar [(ngModel)]="searchTerm"
(ionChange)="searchChanged($event)"></ion-searchbar>

  <ion-item>
    <ion-label>Tipo de busqueda</ion-label>
    <ion-select [(ngModel)]="type" (ionChange)="searchChanged($event)">
      <ion-select-option value="">All</ion-select-option>
      <ion-select-option value="movie">Peliculas</ion-select-option>
      <ion-select-option value="series">Series</ion-select-option>
      <ion-select-option value="episode">Episodios</ion-select-option>
    </ion-select>
  </ion-item>
</ion-list>

```

```

    <ion-item button *ngFor="let item of (results | async)"
[routerLink]="['/', 'movies', item.imdbID]">
      <ion-avatar slot="start">
        <img [src]="item.Poster" *ngIf="item.Poster != 'N/A'">
      </ion-avatar>

      <ion-label text-wrap>
        <h3>{{ item.Title }}</h3>
        {{ item.Year }}
      </ion-label>

      <ion-icon slot="end" *ngIf="item.Type == 'movie'"
name="videocam"></ion-icon>
      <ion-icon slot="end" *ngIf="item.Type == 'series'"
name="tv"></ion-icon>
      <ion-icon slot="end" *ngIf="item.Type == 'game'"
name="logo-game-controller-b"></ion-icon>

    </ion-item>
  </ion-list>
</ion-content>

```

9. Y por último en el archivo `movies-details.page.html` agregamos el siguiente código que mostrará los detalles de lo que haya buscado el usuario

```

<ion-header>
  <ion-toolbar color="dark">
    <ion-buttons slot="start">
      <ion-back-button defaultHref="/"></ion-back-button>
    </ion-buttons>
    <ion-title>{{ information?.Genre }}</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content padding>
  <ion-card *ngIf="information">
    <ion-card-header>
      <ion-card-title>
        {{ information.Title }}

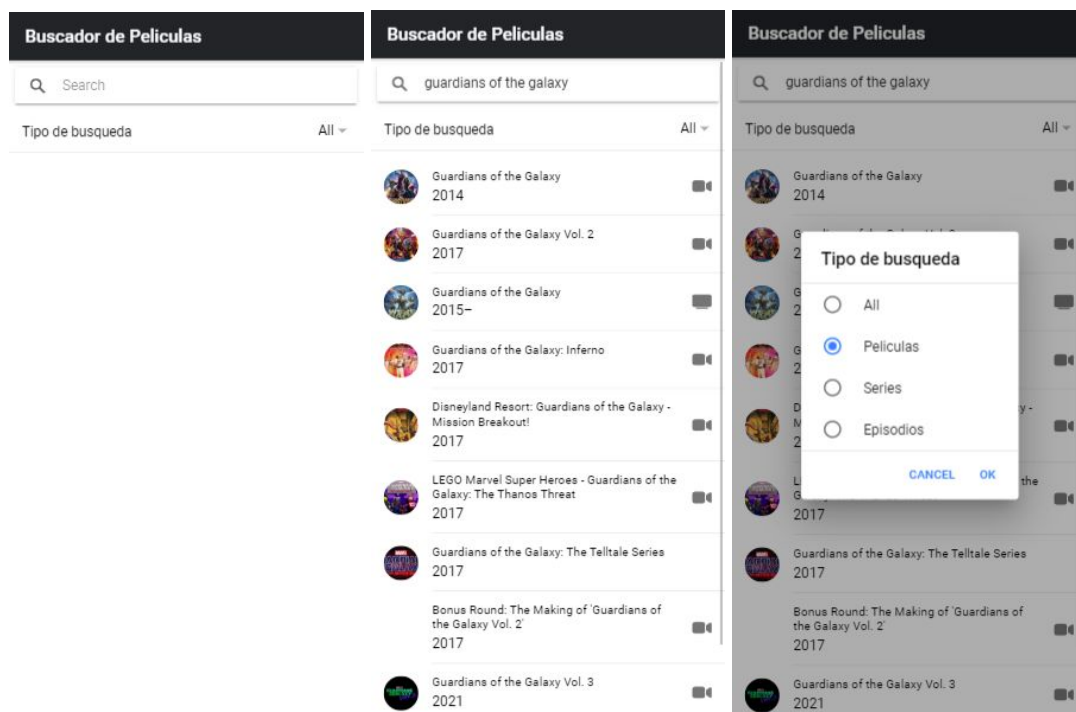
```

```

</ion-card-title>
<ion-card-subtitle>
  {{ information.Year }}
</ion-card-subtitle>
</ion-card-header>
<ion-card-content text-center>
  <img [src]="information.Poster" class="info-img">
  {{ information.Plot }}
  <ion-item lines="none">
    <ion-icon name="star-half" slot="start"></ion-icon>
    <ion-label>{{ information.imdbRating }}</ion-label>
  </ion-item>
  <ion-item lines="none">
    <ion-icon name="clipboard" slot="start"></ion-icon>
    <ion-label text-wrap>{{ information.Director }}</ion-label>
  </ion-item>
  <ion-item lines="none">
    <ion-icon name="contacts" slot="start"></ion-icon>
    <ion-label text-wrap>{{ information.Actors }}</ion-label>
  </ion-item>
  <ion-button expand="full" (click)="openWebsite()"
*ngIf="information.Website && information.Website != 'N/A'">
    <ion-icon name="open" slot="start"></ion-icon>
    Abrir Sitio
  </ion-button>
</ion-card-content>
</ion-card>
</ion-content>

```

Capturas de como debe quedar:



Conclusión

La elaboración de este examen fue bastante compleja y con varios obstáculos tanto de diseño como de programación pero finalmente se logró. gracias a esto también se adquiere el conocimiento de cómo implementar API's en el desarrollo de una aplicación móvil.