

API con Formulario:

interfaces/response.ts

```
export interface ApiNombre{
  name: string;
  episode: string[];
}

export interface Response {
  results: ApiNombre[];
}

export interface ApiEpisode {
  name: string;
  characters: string[];
}

export interface ResponseEpisode {
  results: ApiEpisode[];
}
```

Request.ts

```
import { inject, Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { ApiEpisode, ApiNombre, Response } from '../interfaces/response';
import { ResponseEpisode } from '../interfaces/response';

@Injectable({
  providedIn: 'root',
})
export class Request {
  private http = inject(HttpClient);

  getCharacter(name: string): Observable<Response> {
    return this.http.get<Response>({
      url: `https://rickandmortyapi.com/api/character/?name=${name}`
    });
  }

  getEpisodeCharacter(ep: string): Observable<ApiEpisode> {
    return this.http.get<ApiEpisode>({
      url: `${ep}`
    });
  }

  getEpisode(name: string): Observable<ResponseEpisode> {
    return this.http.get<ResponseEpisode>({
      url: `https://rickandmortyapi.com/api/episode/?name=${name}`
    });
  }

  getCharacterEpisode(ch: string): Observable<ApiNombre> {
    return this.http.get<ApiNombre>({
      url: `${ch}`
    });
  }
}
```

Ejercicio1.ts

```
import { Component, inject } from '@angular/core';
import { FormControl, FormGroup, ReactiveFormsModule } from '@angular/forms';
import { ApiEpisode, Response } from '../../interfaces/response';
import { Request } from '../../services/request';
import { ApiNombre } from '../../interfaces/response';

@Component({
  selector: 'app-ejercicio1',
  imports: [ReactiveFormsModule],
  templateUrl: './ejercicio1.html',
  styleUrls: ['./ejercicio1.css'],
})
export class Ejercicio1 {
  public request = inject(Request)

  public episodeCh: ApiEpisode[] = [];
  public characters: ApiNombre[] = [];
  public episodes: ApiEpisode[] = [];
  public characterEpi: ApiNombre[] = []

  public reactiveForm = new FormGroup({
    selection: new FormControl(''),
    name: new FormControl(''),
  });

  indice: number = 0;

  public onSubmit(): void {
    const selection = this.reactiveForm.value.selection;
    const name = this.reactiveForm.value.name;

    if (selection === "personaje") {
      if (name) {
        this.request.getCharacter(name).subscribe((res) => {
          this.characters = res.results;
          const character = this.characters[this.indice];
          character.episode.map((url) => {
            this.request.getEpisodeCharacter(url).subscribe((epi) => {
              this.episodeCh.push(epi);
            });
          });
        });
      }
    }
  } else {
    if (name) {
      this.request.getEpisode(name).subscribe({
        next: (epi) => {
          this.episodes = epi.results;
          const episode = this.episodes[this.indice]
          episode.characters.map((n) => {
            this.request.getCharacterEpisode(n).subscribe({
              next: (ch) => {
                this.characterEpi.push(ch);
              }
            })
          })
        }
      });
    }
  }
}
```

Ejercicio1.html

```
<div class="question" id="question-1">
  <h4>Ejercicio 1</h4>
  <form method="post" (ngSubmit)="onSubmit()" [formGroup]="reactiveForm">
    <div class="answer-bank">
      <div class="answer">
        <select name="finish" id="finish" formControlName="selection">
          <option value="">- Búsqueda por: -</option>
          <option value="capitulo">Capítulo</option>
          <option value="personaje">Personaje</option>
        </select>
      </div>
      <div class="answer">
        <input type="text" name="name" id="name" value="nombre" placeholder="nombre" formControlName="name" />
      </div>
      <div class="answer">
        <input type="submit" value="Enviar" class="primary" />
      </div>
    </div>
  </form>
  @if (characters[indice] && reactiveForm.value.selection === 'personaje') {
  <p>Personaje: {{ characters[indice].name }}</p>
  @for (epi of episodeCh; track $index) {
  <p>{{ epi.name}}</p>
  }
  } @if (episodes[indice] && reactiveForm.value.selection === 'capitulo') {
  <p>Episodios: {{ episodes[indice].name }}</p>
  @for (ch of characterEpi; track $index) {
  <p>{{ ch.name }}</p>
  }
  }
</div>
```

API:

Response Github Developers:

```
export interface GithubUser {  
  login: string;  
  avatar_url: string;  
}  
  
export interface Response {  
  items: GithubUser[];  
}
```

Request Github Developers:

```
import { inject, Injectable } from '@angular/core';  
import { HttpClient } from '@angular/common/http';  
import { Observable } from 'rxjs';  
import { Response } from '../interfaces/response';  
  
@Injectable({  
  providedIn: 'root',  
})  
export class Request {  
  private http = inject(HttpClient);  
  
  getDevelopers(): Observable<Response> {  
    return this.http.get<Response>(  
      'https://api.github.com/search/users?q=developer'  
    );  
  }  
}
```

Github.ts

```
import { CommonModule } from '@angular/common';  
import { Component, inject } from '@angular/core';  
import { Request } from '../../services/request';  
  
@Component({  
  selector: 'app-github',  
  imports: [CommonModule],  
  templateUrl: './github.html',  
  styleUrls: ['./github.css'],  
})  
export class Github {  
  private request = inject(Request);  
  
  usuarios: any[] = [];  
  
  verPerfiles() {  
    this.request.getDevelopers()  
      .subscribe(res => {  
        this.usuarios = res.items;  
      });  
  }  
}
```

Response AngularApi:

```
export interface AngularApi{
  login: string;
  avatar_url: string;
}

export interface Response{
  items: AngularApi[];
}
```

Request Angular Api:

```
getAngular(): Observable<Response> {
  return this.http.get<Response>(
    'https://api.github.com/search/users?q=angular'
  );
}
```

Github2.ts

```
C:\> Users > Marcos Rodríguez > Documents > srcApis > app > views > github2 > TS github2.ts > ...
1  import { CommonModule } from '@angular/common';
2  import { Component, inject } from '@angular/core';
3  import { Request } from '../../services/request';
4  import { AngularApi } from '../../interfaces/response';
5
6  @Component({
7    selector: 'app-github2',
8    imports: [CommonModule],
9    templateUrl: './github2.html',
10    styleUrls: ['./github2.css'],
11  })
12  export class Github2 {
13    private request = inject(Request);
14
15    angular: AngularApi[] = [];
16
17    selectedTitle: string = '';
18    selectedImg: string = '';
19
20    verPerfiles() {
21      this.request.getAngular().subscribe((res) => {
22        this.angular = res.items;
23      });
24    }
25
26    modal: boolean = false;
27    modalOpen: string = 'modal';
28
29    openModal(img: string, title: string) {
30      this.selectedImg = img;
31      this.selectedTitle = title;
32      this.modal = true;
33      this.modalOpen = 'modal show-modal';
34      console.log(this.selectedImg, this.modal);
35    }
36
37    closeModal() {
38      this.modal = false;
39      this.selectedImg = '';
40      this.selectedTitle = '';
41      this.modalOpen = 'modal';
42    }
43  }
44
```

Github.html

```
<!DOCTYPE html>
<html lang="ca">

<head>
  <meta charset="UTF-8">
  <title>Tarea 5</title>

  <link rel="stylesheet" href="../../styles.css">
</head>

<body>

  <!-- Vista GitHub -->
  <div class="github">
    <div class="menu">
      <h2>Menu</h2>
      <nav>
        <ul>
          <li><a (click)="verPerfiles()">Ver perfiles</a></li>
        </ul>
      </nav>
    </div>
    <article>
      @for (user of angular; track $index) {
        <div>
          
          <h1>{{ user.login }}</h1>
        </div>
      }
    </article>
  </div>

  @if (modal) {
    <div class="{{ modalOpen }}">
      <div class="modal-content">
        <span class="close-button" (click)="closeModal()">✕</span>
        <img [src]="selectedImg" width="300">
        <h1>{{ selectedTitle }}</h1>
      </div>
    </div>
  }
  <footer></footer>

</body>

</html>
```

MemeApiModal y Carrusel:

```
import { Component, inject } from '@angular/core';
import { Request } from '../../../services/request';
import { CommonModule } from '@angular/common';
import { MemeApiModal } from '../../../interfaces/response';

@Component({
  selector: 'app-meme-three',
  imports: [CommonModule],
  templateUrl: './meme-three.html',
  styleUrls: ['./meme-three.css'],
})
export class MemeThree {
  private request = inject(Request);

  memes: MemeApiModal[] = [];

  selectedTitle: string = '';
  selectedImg: string = '';

  indice: number = 0;
  modal: boolean = false;
  modalOpen: string = 'modal';

  verMemes() {
    this.request.getMemes().subscribe((res) => {
      this.memes = res.data.memes;
      this.indice = 0;
    });
  }

  siguiente() {
    if (this.indice < this.memes.length - 1) {
      this.indice++;
    }
  }

  anterior() {
    if (this.indice > 0) {
      this.indice--;
    }
  }

  openModal(img: string, title: string) {
    this.selectedImg = img;
    this.selectedTitle = title;
    this.modal = true;
    this.modalOpen = 'modal show-modal';
  }
}
```

HTML:

```
<!-- Vista Meme3 -->
<div class="meme">
  <div class="menu">
    <h2>Menu</h2>
    <nav>
      <ul>
        <li><a (click)="verMemes()">Ver perfiles</a></li>
      </ul>
    </nav>
  </div>
  <section>
    <div class="grid">
      <!-- Componente Card -->
      @if (memes.length > 0) {
        <div class="card">
          <div class="photo">
            
            <div class="title">{{ memes[indice].name }}</div>
          </div>
        </div>
      }
    </div>
    <button (click)="anterior()">Anterior</button>
    <button (click)="siguiente()">Siguiente</button>
  </section>
</div>

@if (modal) {
  <div class="{{ modalOpen }}">
    <div class="modal-content">
      <span class="close-button" (click)="closeModal()">✕</span>
      <img [src]="selectedImg" width="300">
      <h1>{{ selectedTitle }}</h1>
    </div>
  </div>
}
</footer></footer>
```