

## ANGULAR 2

### Desarrollo de diapositivas/PWEB2

Diego Aristides Cervantes Apaza

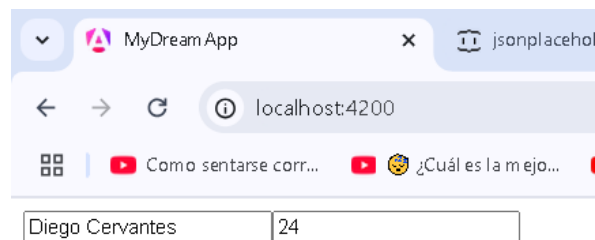
CUI:20213037

Desarrollo:

#### DIAPPOSITIVA 4: Haciendo DataBinding con datos del formulario

```
app.component.html X TS app.component.ts # styles.css TS hello-world.component
my-dream-app > src > app > app.component.html > div
1 <router-outlet></router-outlet>
2
3 <div>
4   <form>
5     <input type="text" name="name" [(ngModel)]="name" />
6     <input type="number" name="age" [(ngModel)]="age" />
7   </form>
8
9   <h2><strong>Name:</strong> {{ name }}</h2>
10  <h2><strong>Age:</strong> {{ age }}</h2>
11 </div>
```

```
app.component.html TS app.component.ts # styles.css TS hello-w
my-dream-app > src > app > TS app.component.ts > AppComponent
1 import { Component } from '@angular/core';
2 import { FormsModule } from '@angular/forms';
3 import { RouterModule } from '@angular/router';
4
5 @Component({
6   selector: 'app-root',
7   standalone: true,
8   imports: [FormsModule, RouterModule],
9   templateUrl: './app.component.html',
10  styleUrls: ['./app.component.css']
11 })
12 export class AppComponent {
13   name: string = 'Diego Cervantes';
14   age: number = 24;
15 }
```



**Name: Diego Cervantes**

**Age: 24**

Diego Cervantes	39
-----------------	----

Name: Diego Cervantes

Age: 39

Diego Cervantes	33
-----------------	----

Name: Diego Cervantes

Age: 33

Ejecutando el manejo de databinding y jugando con las edades en ejecución, usando standalone components, evitando problemas con las versiones antiguas de angular.

DIAPPOSITIVA 6:

```
jsonplaceholder.typicode.com/ X +
jsonplaceholder.typicode.com/posts
Dar formato al texto
{
  {
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
    "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"
  },
  {
    "userId": 1,
    "id": 2,
    "title": "qui est esse",
    "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"
  },
  {
    "userId": 1,
    "id": 3,
    "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
    "body": "et lusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut"
  },
  {
    "userId": 1,
    "id": 4,
    "title": "eum et est occaecati",
    "body": "ullam et ssepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"
  },
  {
    "userId": 1,
    "id": 5,
    "title": "nesciunt quas odio",
    "body": "repudiandae veniam quaerat sunt sed\nnihil aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dolor neque"
  },
  {
    "userId": 1,
    "id": 6,
    "title": "dolorem eum magni eos aperiam quia",
    "body": "ut aspernatur corporis harum nihil quis provident sequi\nmollitia nobis aliquid molestiae\nperspiciatis et ea nemo ab reprehenderit accusantium quas\nvoluptate dolores velit et doloremque molestiae"
  },
}
```

Usando web de JsonPlaceHolder como backend para Angular.

DIAPPOSITIVA 8:

```
PS C:\Users\USUARIO\Downloads\PWEB2-TEORIA\avanceGitAngular\proyectoAngular\my-dream-app> ng generate service data
CREATE src/app/data.ts (135 bytes)
PS C:\Users\USUARIO\Downloads\PWEB2-TEORIA\avanceGitAngular\proyectoAngular\my-dream-app>
```

Creando servicios “data” en angular más recientes.

## DIAPOSITIVA 9:

```
TS data.ts 1. U • < app.component.html TS app.component.ts TS hello-world.component.ts TS user.component.ts <
my-dream-app > src > app > TS data.ts > DataService
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Post } from '../Post';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class DataService {
9
10   constructor(private httpClient: HttpClient) {
11     console.log("Service working...");
12   }
13
14   getData() {
15     httpClient.get<Post[]>("https://jsonplaceholder.typicode.com/posts");
16   }
17 }
```

Estructura inicial de una API.

## DIAPOSITIVA 10: Insertando un Servicio para llamar a Json

```
TS app.module.ts 3 • TS data.ts TS Post.ts < app.component.html
my-dream-app > src > app > TS app.component.ts > AppComponent
1 import { Component } from '@angular/core';
2 import { DataService } from '../data';
3
4 @Component({ // Decorador @Component necesario
5   selector: 'app-root', // Selector que se usará en el HTML
6   templateUrl: './app.component.html',
7   styleUrls: ['./app.component.css']
8 })
9 export class AppComponent {
10   name: string = 'sdads';
11   age: number = 40;
12
13   constructor(private dataService: DataService) {
14     this.dataService.getData().subscribe(data => {
15       console.log(data);
16     });
17   }
18 }
```

```
TS app.module.ts 3 TS data.ts TS Post.ts app.component.html TS app.component
my-dream-app > src > app > TS app.module.ts > AppModule
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { AppRoutingModule } from './app-routing.module';
4 import { FormsModule } from '@angular/forms';
5 import { RouterModule, Route } from '@angular/router';
6
7 import { AppComponent } from './app.component';
8 import { HelloWorldComponent } from './hello-world/hello-world.component';
9 import { UserComponent } from './user/user.component';
10
11 import { DataService } from './data.service';
12 import { AboutComponent } from './about/about.component';
13
14 const routes: Route[] = [
15   { path: '', component: AppComponent },
16   { path: 'about', component: AboutComponent },
17   { path: 'hello', component: HelloWorldComponent },
18 ];
19
20 @NgModule({
21   declarations: [
22     AppComponent,
23     HelloWorldComponent,
24     UserComponent,
25     AboutComponent
26   ],
27   imports: [
28     BrowserModule,
29     AppRoutingModule,
30     FormsModule,
31     HttpClientModule,
32     RouterModule.forRoot(routes),
33   ],
34   providers: [DataService],
35   bootstrap: [AppComponent]
36 })
37 export class AppModule { }
```

```
world.component.html # user.component.css user.component.html # hello-world.component.css TS hello-world
my-dream-app > src > app > TS app.spect.ts > ...
1 import { TestBed } from '@angular/core/testing';
2 import { App } from './app';
3
4 describe('App', () => {
5   beforeEach(async () => {
6     await TestBed.configureTestingModule({
7       imports: [App],
8     }).compileComponents();
9   });
10
11   it('should create the app', () => {
12     const fixture = TestBed.createComponent(App);
13     const app = fixture.componentInstance;
14     expect(app).toBeTruthy();
15   });
16
17   it('should render title', () => {
18     const fixture = TestBed.createComponent(App);
19     fixture.detectChanges();
20     const compiled = fixture.nativeElement as HTMLElement;
21     expect(compiled.querySelector('h1')?.textContent).toContain('Hello, my-dream-app');
22   });
23 });
24
```

```
Help  < ->  proyectoAngular

TS data.service.ts X  TS data.spec.ts I  TS hello-world.component.ts  TS user.component.ts  <> hello-world

my-dream-app > src > app > TS data.service.ts > DataService
1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Post } from './Post';
4
5  @Injectable({
6    providedIn: 'root'
7  })
8  export class DataService {
9
10     constructor(private httpClient: HttpClient) {
11       console.log("Service working...");
12     }
13
14     getData() {
15       return this.httpClient.get<Post[]>("https://jsonplaceholder.typicode.com/posts");
16     }
17   }
```

```
TS data.ts  TS Post.ts  <> app.component.html  TS app.component.ts

my-dream-app > src > app > TS data.spec.ts > ...
1  import { TestBed } from '@angular/core/testing';
2
3  import { Data } from './data';
4
5  describe('Data', () => {
6    let service: Data;
7
8    beforeEach(() => {
9      TestBed.configureTestingModule({});
10     service = TestBed.inject(Data);
11   });
12
13   it('should be created', () => {
14     expect(service).toBeTruthy();
15   });
16 });
17 |
```

```
TS data.ts x TS Post.ts app.component.html TS app.component.ts TS data.service.ts TS
my-dream-app > src > app > TS data.ts > DataService
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { Post } from './Post';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class DataService {
9
10   constructor(private httpClient: HttpClient) {
11     console.log("Service working...");
12   }
13
14   getData() {
15     return this.httpClient.get<Post[]>("https://jsonplaceholder.typicode.com/posts");
16   }
17 }
```

```
elp
TS data.ts TS Post.ts x app.comp
my-dream-app > src > app > TS Post.ts > Post
1 export interface Post {
2   "userId" : number;
3   "id" : number;
4   "title" : string;
5   "body" : string;
6 }
```

```
elp
projectoAngular
onent.css user.component.html # hello-world.component.css TS hello-world.com
my-dream-app > src > TS main.ts > ...
1 import { bootstrapApplication } from '@angular/platform-browser';
2 import { AppComponent } from './app/app.component';
3
4 bootstrapApplication(AppComponent)
5 .catch(err => console.error(err));
```

```
elp  ← →  proyectoAngular

onent.css  user.component.html  # hello-world.component.css  TS hello-world.com

my-dream-app > src > TS main.ts > ...
1  import { bootstrapApplication } from '@angular/platform-browser';
2  import { AppComponent } from './app/app.component';
3
4  bootstrapApplication(AppComponent)
5  .catch(err => console.error(err));
```

git log --graph --pretty=oneline --abbrev-commit --all

```
MINGW64:/c:/Users/USUARIO/Downloads/PWEB2-TEORIA/avanceGitAngular/proyectoAngular
* 208e511 (HEAD -> angular2, origin/angular2) añadiendo mudile y dataservices al proyecto
* 2e48637 creando post.ts con atributos
* 95824ea insercion de servicios
* 150c07d actualizacion de versiones json
* f317eaf creando servicios: data.ts
* 1eb9c9f creando servicios: data.spec.ts
* ec16697 ingresando nombre y edad fijos en app.component.ts
* 13b5fda cambiando app.component.html por nuevos formularios de datos personales
* 98b51e3 (origin/main, origin/HEAD, main) añadiendo pdf de avance angular1
* 3953734 (origin/angular1, angular1) aceptando ingreso de datos en el formulario y listándolos en app.component.ts
* a8e7822 agregando formulario, que listará palabras ingresadas en app.component.html
* 0a19f43 iteración de eliminación en la lista desde app.component.ts
* 338c847 creando botón para borrar usuarios o viñetas desde app.component.html* 94f9659 saludando luego del click en los botones
* a07c763 boton hazme clickk solo a mi en component.html
* 64e875b app component funcion hola
* dfedf74 añadiendo boton final
* 5e8f301 boton en component.html
* 144ec95 funcion hola desde app.component.ts
:..skipping...
* 208e511 (HEAD -> angular2, origin/angular2) añadiendo mudile y dataservices al proyecto
* 2e48637 creando post.ts con atributos
* 95824ea insercion de servicios
* 150c07d actualizacion de versiones json
* f317eaf creando servicios: data.ts
* 1eb9c9f creando servicios: data.spec.ts
* ec16697 ingresando nombre y edad fijos en app.component.ts
* 13b5fda cambiando app.component.html por nuevos formularios de datos personales
```