

Angular

Guía Inicial



Índice

Índice	2
¿Qué es Angular?	3
Características de Angular	3
Arquitectura de Angular	3
Funcionalidades de Angular	6

¿Qué es Angular?

Angular es un Framework de JavaScript de código abierto escrito en TypeScript. Su objetivo principal es desarrollar aplicaciones de una sola página. Fue desarrollado por Google (2009) y además de mantenerla y actualizarla, tiene un gran comunidad detrás dando soporte.

Características de Angular

Angular usa DOM, TypeScript y Data Binding

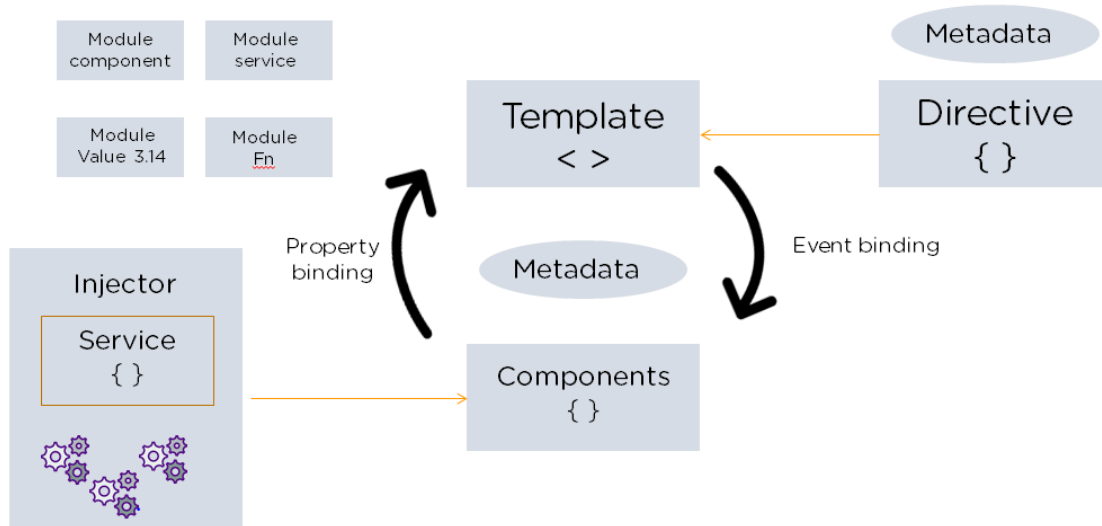
TypeScript no es obligatorio para desarrollar una aplicación Angular. Sin embargo, es muy recomendable ya que ofrece una mejor estructura sintáctica, al tiempo que hace que la base de código sea más fácil de entender y mantener.

El data binding es un proceso que permite a los usuarios manipular elementos de la página web a través de un navegador web.

Angular usa el enlace bidireccional. Reflejando los cambios realizados en los elementos de la interfaz de usuario correspondientes.

Arquitectura de Angular

Angular es un marco modelo-vista-controlador (MVC) completo.



En todo el conjunto de componentes que conforman la estructura de Angular, hay que destacar dos principales: Components (Componentes) y los Modules (Módulos). Hay que tener en cuenta que todo en Angular son Class (clases).

Components (Componentes)

Los componentes son piezas o fragmentos de código que define una clase que contiene la lógica y los datos de la aplicación. Un componente por lo general define una parte de la interfaz de usuario (UI). Los componentes son archivos TypeScript, a excepción de los html y css que se puedan usar.

La estructura básica de un componente es la siguiente:

1. Un import del componente , con el cual podemos usar ese componente en otros lugares del proyecto de Angular.

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

2. Un decorador @Component el cual tiene el nombre de como va a ser llamado el componente y las rutas de archivos html y css (aunque en pequeños programas pueda ser escrito directamente ahí, sin enrutar los archivos).

```
@Component({  
  selector: 'app-main-page',  
  templateUrl: './main-page.component.html',  
  styleUrls: ['./main-page.component.css']  
})
```

3. Y por último la clase en sí donde se podrán crear todas las variables y métodos que podrán ser usados en la página web.

```
export class MainPageComponent implements OnInit {  
  
  characters= [  
    {name:"Shelly",life:3600},  
    {name:"Nita",life:3800},  
    {name:"Colt",life:2800},  
    {name:"Pinp",life:840},  
    {name:"Mant",life:1800},  
  ]  
}
```

Modules (Módulos)

Los módulos tienen uno o varios componentes, pero un componente sólo puede tener un solo módulo.

Angular tiene un módulo raíz, llamado AppModule, que proporciona el mecanismo de arranque para iniciar la aplicación.

Todos los componentes tienen que estar declarados en algún módulo.

Si el componente tiene su propio módulo tiene que ser importado en el módulo AppModule.

Si no será declarado el módulo AppModule.

Si no está en el módulo principal el componente no funcionará.

Abajo una captura del módulo AppModule donde podemos observar algunos componentes declarados y otros módulos importados.

```
src > app > app.module.ts > AppModule
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3
4  import { AppComponent } from './app.component';
5  import { Contador } from './contador/contador.component';
6  import { lista } from './lista/lista.component';
7  import { AlumnosModule } from './alumnos/alumnos.module';
8  import { MainPageComponent } from './brawl-stars/main-page/main-page.component';
9  import { BrawlStarsModule } from './brawl-stars/brawl-stars.module';
10
11  @NgModule({
12    declarations: [
13      AppComponent,
14      Contador,
15      lista,
16      MainPageComponent,
17    ],
18    imports: [
19      BrowserModule,
20      AlumnosModule,
21      BrawlStarsModule
22    ],
23    providers: [],
24    bootstrap: [AppComponent]
25  })
26  export class AppModule { }
27
```

Funcionalidades de Angular

En un componente angular tenemos un archivo ts el cual es una clase. En él podemos escribir toda la lógica del programa que usaremos en los archivos HTML y CSS.

Al ser un archivo TypeScript tenemos todas las funcionalidades de TypeScript siendo un envoltorio de JavaScript y el cual podemos usar con HTML y CSS.

Es en HTML donde tenemos nuevas “etiquetas y funcionalidades” proporcionadas por Angular. Estas funcionalidades se basan en un lenguaje de programación. como bucles, condicionales, triggers (disparadores), llamadas a métodos y variables, entre otras.

En la imagen de abajo podemos ver un ejemplo simple de cómo podemos combinar etiquetas de HTML y código de Angular.

```
src > app > brawl-stars > main-page >  main-page.component.html > ...  
Go to component  
1 <h1>Brawl Stars</h1>  
2 <hr>  
3 <div class="row">  
4   <div class="col">  
5     <h3>Personajes</h3>  
6     <hr>  
7     <ul>  
8       <li *ngFor="let i of characters">  
9         {{i.name}} - {{i.life}}  
10      </li>  
11    </ul>  
12    <button (click)="colorLife()">Colorear</button>  
13  </div>  
14 </div>
```