GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA

red.es PROFESIONALES DIGITALES | Formación Continua

El FSE invierte en tu futuro

UNIÓN EUROPEA

PROGRAMA DE FORMACIÓN DIRIGIDO A PERSONAS TRABAJADORAS PRIORITARIAMENTE OCUPADAS, PARA LA ADQUISICIÓN Y MEJORA DE COMPETENCIAS EN EL ÁMBITO DE LA TRANSFORMACIÓN Y LA ECONOMÍA DIGITAL

C058/17-ED

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA

red.es PROFESIONALES DIGITALES | Formación Continua

UNIÓN EUROPEA

El FSE invierte en tu futuro

Jonatan Lucas
Jonatan.lucas@centic.es
@Jon_Lucas_

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Resumen

1. Introducción y conceptos previos.
2. Arquitectura de Angular.
3. Rutas para moverse entre componentes.
4. Relaciones entre componentes.
5. Conceptos avanzados de componentes.
6. Conceptos avanzados de Angular.
7. Inyección de dependencias y servicios http.
8. Formularios.
9. Material Design y testing.
10. Angular 9, pequeños cambios con Angular 8.

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Introducción y requisitos previos

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Introducción

- Usar las directivas de Angular para mostrar y ocultar elementos.
- Crear componentes en Angular para trabajar con los datos.
- Usar *one-way data binding* para mostrar datos.
- Añadir campos editables para actualizar un modelo con *two-way data binding*.
- Enlazar métodos de los componentes para usar eventos.
- Mostrar datos con *pipes*.
- Crear servicios para recuperar datos del servidor.
- Usar *routing* para navegar por las diferentes vistas y sus componentes.
- Crear una aplicación con multimodulado.
- Crear formularios.
- Dar estilo a nuestra aplicación

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

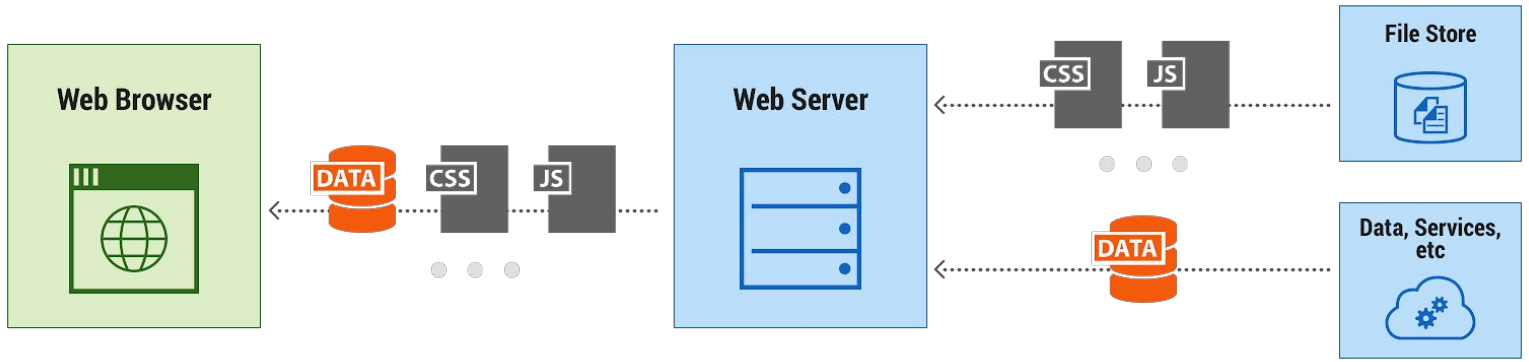
Single-page application

Consiste en una página web donde el contenido se carga en la primera petición.

- HTML
- CSS
- JavaScript

Conseguimos más fluidez en nuestra web. Ejemplos de SPA pueden ser:

- Youtube
- Netflix
- Gmail



Pero… ¿porqué Angular?

Angular es un *framework* desarrollado en *Typescript* de código abierto que construye aplicaciones en HTML y JavaScript. Fue desarrollado por Google. Este *framework* se utilizó para superar los obstáculos encontrados al trabajar con aplicaciones *Single Page*. El lanzamiento inicial de este *framework* fue en octubre de 2010.

Principales características de Angular:

- Velocidad y rendimiento.
  - Optimización de código a través de herramientas como Ivy.
  - Se puede ejecutar bajo cualquier servidor que renderice una web, el más común y usado es node.js
  - División del código para que la web solo cargue el código de la vista que se esté visualizando en ese momento.
- Productividad.
  - Creación de componentes, módulos o servicios de forma rápida.
  - Angular CLI. Herramienta de línea de comandos que permite crear proyectos nuevos, elementos y realizar test.
  - IDEs. Integración con la mayoría de editores populares.

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

TypeScript

- TypeScript nos permite escribir aplicaciones en JavaScript de una forma más sencilla.
- Es un lenguaje orientado a objetos.
- Pequeñas diferencias con JS.

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

Instalando Node.js y Angular 8

Para poder trabajar con Angular 8 debemos tener instalado:

- Node.js en versiones superiores a la 10.9.0. Se aconseja usar el gestor nvm, node version manager, para instalar la versión de node.js que necesitemos.
- Npm, node package manager, que se instala con la versión de Node.js

Instalar Angular CLI:

```
npm install -g @angular/cli@8
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Angular CLI

Comandos:

add	build	config	generate
help	new	serve	test
update			

Crear un nuevo proyecto:

```
ng new cursoAngular8
```

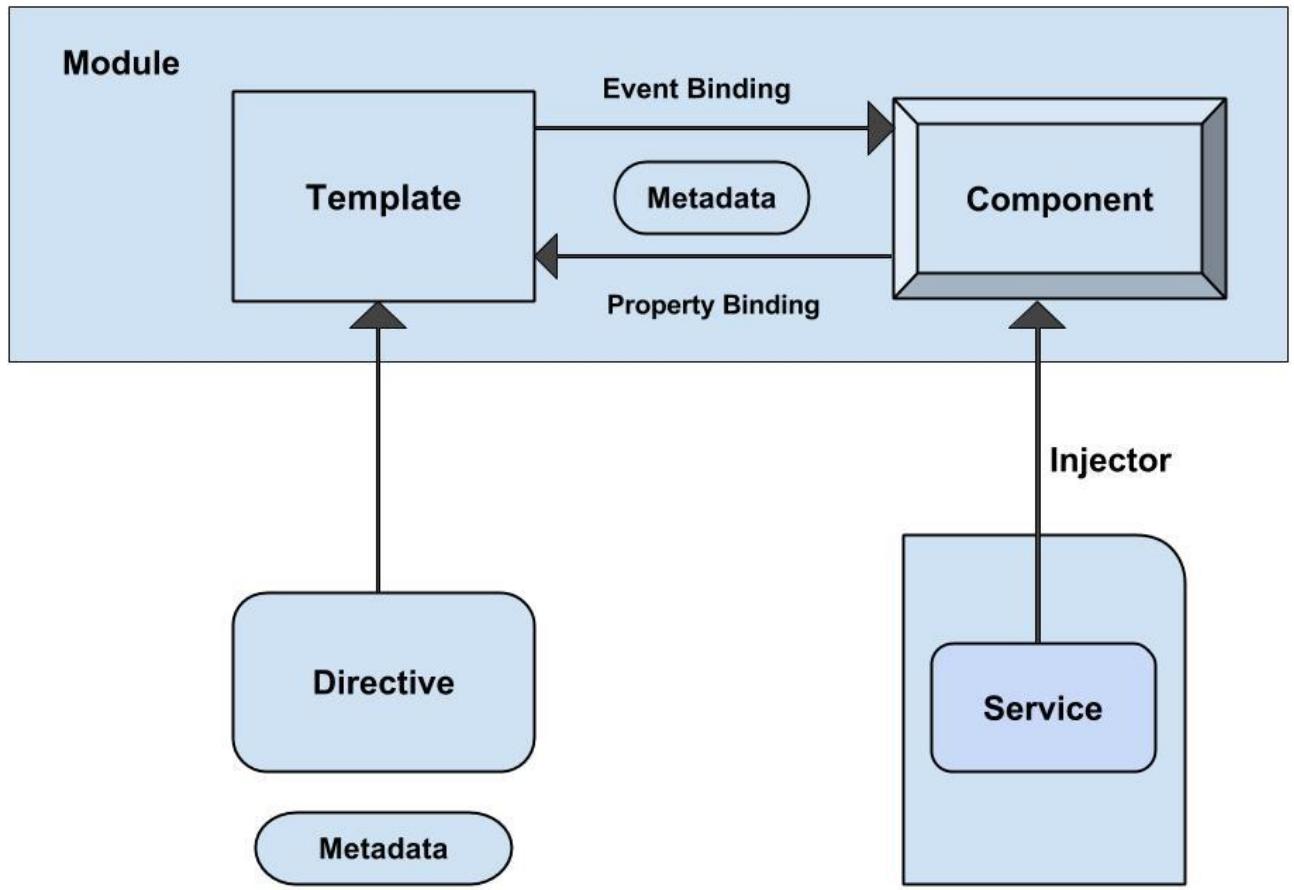
Ejecutar una aplicación:

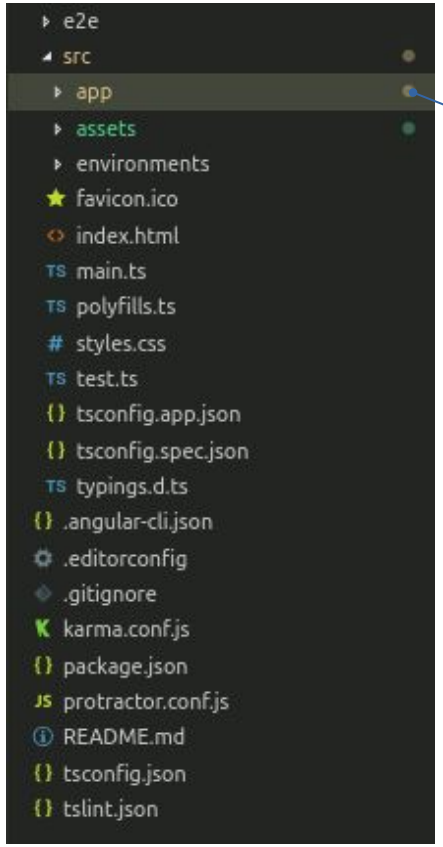
```
ng serve
```

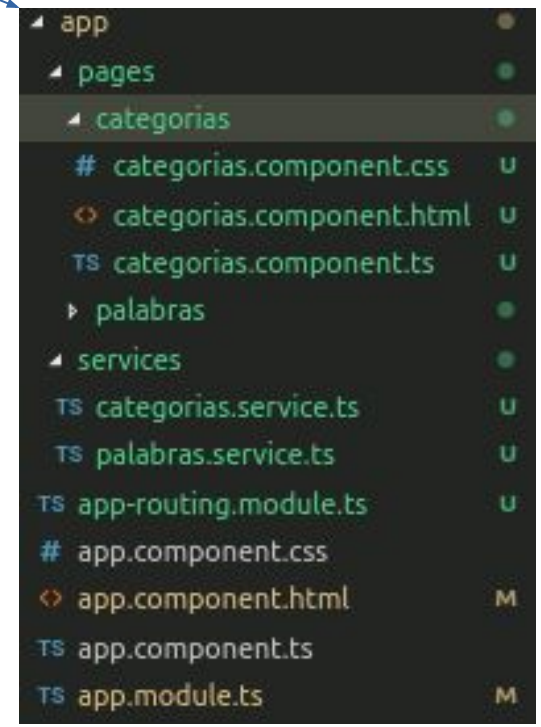
GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Arquitectura de Angular

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA




e2e
src
app
assets
environments
favicon.ico
index.html
main.ts
polyfills.ts
styles.css
test.ts
tsconfig.app.json
tsconfig.spec.json
typings.d.ts
.angular-cli.json
.editorconfig
.gitignore
karma.conf.js
package.json
protractor.conf.js
README.md
tsconfig.json
tslint.json


app
pages
categorias
categorias.component.css U
categorias.component.html U
categorias.component.ts U
palabras
services
categorias.service.ts U
palabras.service.ts U
app-routing.module.ts U
app.component.css
app.component.html M
app.component.ts
app.module.ts M

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Módulos

- Las aplicaciones en Angular son modulares.
- Posee su propio sistema de modulación, NgModules
- Cada módulo puede contener componentes, servicios y otros archivos de código necesario.
- Al menos se debe contener un módulo por app.
- Se encargará de lanzar todos los archivos necesarios para nuestra aplicación.

src/app/app.module.ts

```
import { NgModule }      from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
@NgModule({
  imports:      [ BrowserModule ],
  providers:    [ Logger ],
  declarations: [ AppComponent ],
  exports:      [ AppComponent ],
  bootstrap:    [ AppComponent ]
})
export class AppModule { }
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Componentes

- Definen la lógica de la aplicación.
- Angular crea, actualiza y destruye componentes a medida que el usuario se mueve a través de la aplicación.
- Cada vista de Angular esta compuesta por:
  - Un fichero HTML para la plantilla.
  - Un fichero TS para el componente.
  - Un fichero CSS para el diseño.

```
ng generate component name_component
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

src/app/hero-list.component.ts (metadata)

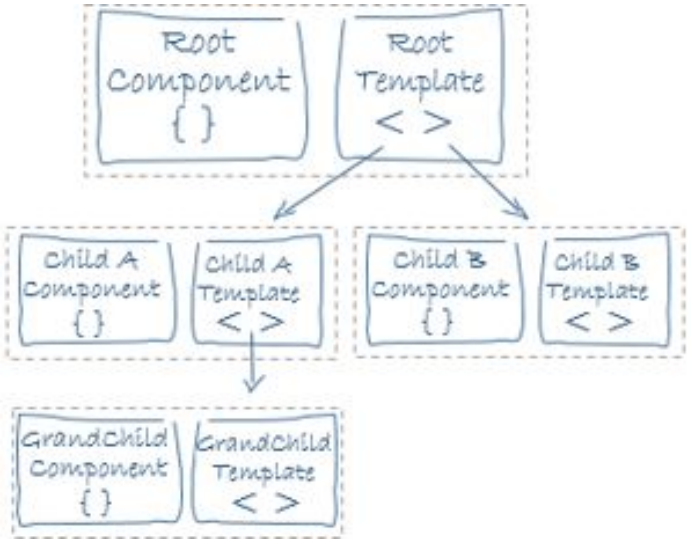
```
@Component({
  selector:    'hero-list',
  templateUrl: './hero-list.component.html',
  providers:  [ HeroService ]
})
export class HeroListComponent implements OnInit {
/* . . . */
}
```

src/app/hero-list.component.ts (class)

```
export class HeroListComponent implements OnInit {
  heroes: Hero[];
  selectedHero: Hero;

  constructor(private service: HeroService) { }

  ngOnInit() {
    this.heroes = this.service.getHeroes();
  }

  selectHero(hero: Hero) { this.selectedHero = hero; }
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

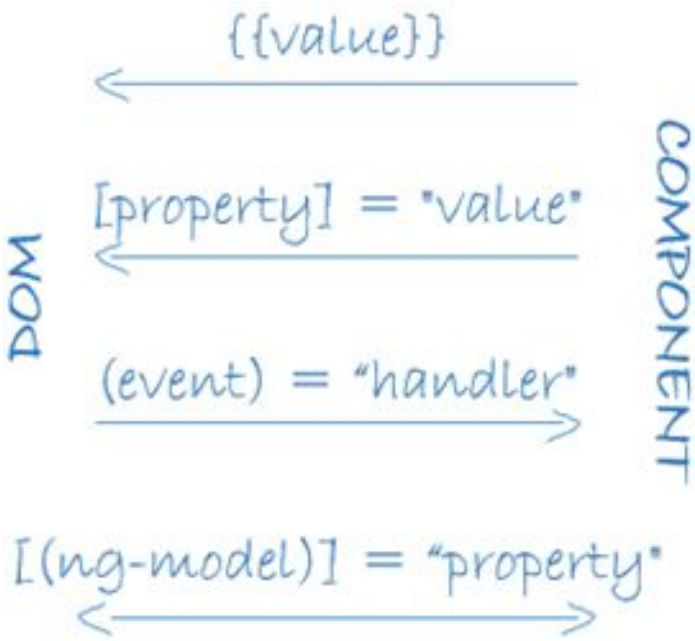
Plantillas y vistas




```
<h2>Hero List</h2>

<p><i>Pick a hero from the list</i></p>
<ul>
  <li *ngFor="let hero of heroes"
(click)="selectHero(hero)">
    {{hero.name}}
  </li>
</ul>

<app-hero-detail *ngIf="selectedHero"
[hero]="selectedHero"></app-hero-detail>
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Data Binding




```
<li>{{hero.name}}</li>

<app-hero-detail
[hero]="selectedHero"></app-hero-detail>

<li (click)="selectHero(hero)"></li>
```

```
<input [(ngModel)]="hero.name">
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Directivas

- Se usan para modificar el DOM y proporcional a la plantilla de HTML un aspecto más dinámico

```
<li *ngFor="let hero of heroes"></li>

<app-hero-detail *ngIf="selectedHero"></app-hero-detail>
```

- *ngFor: muestra una etiqueta <li> por cada objeto heroe del array de heroes.
- *ngIf: incluye el componente solo si existe el objeto *selectedHero*.

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Servicios e ID

- Es una clase amplia que abarca cualquier valor, función o característica que necesita una app.
- Angular distingue los componentes de los servicios para mejorar la modularidad y la reutilización.
- Los services se encarga sobre todo de recuperar datos del servidor, validar la entrada de usuarios o iniciar sesión directamente.
  - Para implementar esta lógica en los componentes Angular utiliza la inyección de dependencias

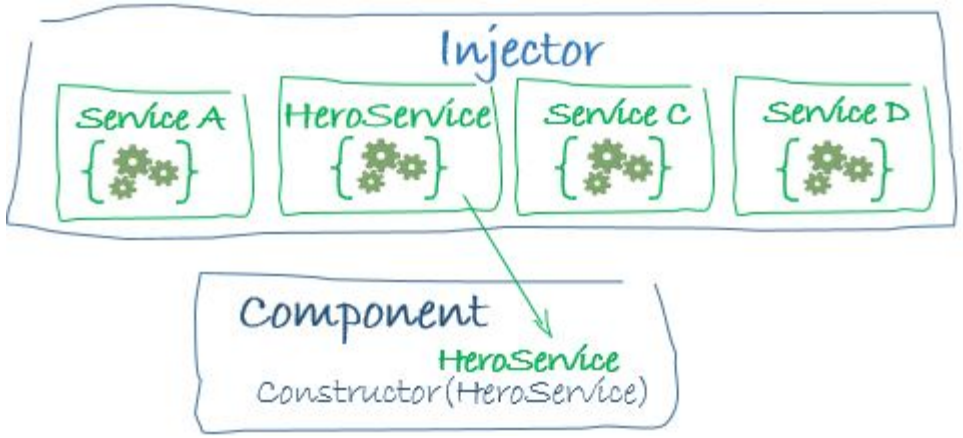
```
ng generate service name_service
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

```
@Injectable({})
```

```
export class HeroService {
  private heroes: Hero[] = [];

  constructor(
    private backend: BackendService,
    private logger: Logger) { }

  getHeroes() {
    this.backend.getAll(Hero).subscribe( (heroes: Hero[]) => {
      this.logger.log(`Fetched ${heroes.length} heroes.`);
      this.heroes.push(...heroes); // fill cache
    });
    return this.heroes;
  }
}
```

- Enlaza los componentes con los servicios.
- Cuando Angular crea una instancia de una clase componente, determina qué servicios u otras dependencias necesita ese componente al observar los parámetros del constructor.
- Cuando un componente depende de un servicio, el inyector crea una instancia de ese servicio.



GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Http Request

- Las aplicaciones de Angular se comunican con los servidores a través de los *services*.
- Para trabajar con las peticiones utilizaremos el objeto de *HttpClient* que se basa en la interfaz *XMLHttpRequest*.
- Un método de *HttpClient* no comienza su solicitud HTTP hasta que llamemos al método *.subcribe()*
- *HttpClient* admite solicitudes de GET, POST, PUT y DELETE.

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

const httpOptions = {
  headers: new HttpHeaders({
    'Content-Type':  'application/json',
    'Authorization': 'my-auth-token'
  })
};
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Rutas para navegar entre componentes

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Routing

- El archivo de rutas indica a Angular que componentes cargar según la ruta que visitemos.
- Podremos pasar parámetros sin problema.
- El objeto encargado de realizar las rutas para Angular es *Router*.
- Pero para poder trabajar con este objeto y rutas en Angular debemos colocar esta etiqueta en el index.html

```
<base href=”/”>
```

- En el AppComponent debemos sustituir el código HTML por la siguiente etiqueta:

```
<router-outlet></router-outlet>
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

```
ng generate module app-routing --flat --module=app
```

Angular te genera el siguiente código, pero no es el correcto a usar.

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';

@NgModule({
  imports: [
    CommonModule
  ],
  declarations: []
})

export class AppRoutingModule { }
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

```
import { NgModule }             from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

import { HeroesComponent } from ‘./heroes/heroes.component’;
import { HeroDetailComponent  } from ‘./heroDetail/heroDetail.component’;
import { PageNotFoundComponet } from ‘./page/page.component’;

const    routes:   Routes =  [
    {   path: ‘heroes’, component:  HeroesComponent  },
    {   path: ‘heroes/:id’, component:  HeroDetailComponent   },
    {   path: ‘’, redirecTo:  ‘/heroes’, pathMatch:  ‘full’  },
    {   path: ‘**’, component:  PageNotFoundComponent  }
];

@NgModule({
  exports: [ RouterModule ],
  imports: [ RouterModule.forRoot(routes) ]
})
export class AppRoutingModule {}
```

- Angular tiene dos maneras diferentes de navegar entre sus vistas:
  - Con el atributo routerLink en el HTML.

```
<a routerLink="/categorias">Categorias</a>
<a routerLink="/palabras">Palabras</a>
```

  - El objeto *router* desde el componente.

- Extraer parámetros de la url

```
{ path:  'heroes/:id', component:  HeroDetailComponent  }
```

```
import { ActivatedRoute } from '@angular/router';

…

constructor(…, private activatedRoute: ActivatedRoute ) {
   activatedRoute.params.subscribe( params => {
        this.id = +params['id'];
   });
   …
}
…
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Relación entre componentes

Opciones más importantes para comunicarse entre componentes:

1. Comunicación Padre-Hijo a través de la etiqueta *@Input.*
2. Comunicación entre componentes con *ngOnChanges().*
3. Comunicación Hijo-Padre a través de eventos.
4. Acceso componente Padre a variables del componente Hijo.
5. Comunicación entre componentes a través de *Services*.

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

Padre-Hijo @input

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

import { ArrayObjects } from './classObject';

@Component({
  selector: 'app-parent',
  template: `
    <h2>{{master}} controls {{arrayObjects.length}} numbers</h2>
    <app-child *ngFor="let element of arrayObjects"
      [objectChild]="element"
      [master]="master"  [master]="master">
    </app-child>
  `
})
export class ParentComponent {
  arrayObjects = ArrayObjects;
  master = 'Master';
}
```

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

import { Hero } from './classObject';

@Component({
  selector: 'app-child',
  template: `
    <h3>{{objectChild.name}} says:</h3>
                <p>I,  {{objectChild.name}},  am  at  your  service,
{{masterName}}.</p>
  `
})
export class ChildComponent {
  @Input() objectChild: Hero;
  @Input('master') masterName: string;
}
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

Padre-Hijo @input + ngOnChanges()

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

@Component({
  selector: 'app-version-parent',
  template: `
    <h2>Source code version</h2>
    <button (click)="newMinor()">New minor version</button>
    <button (click)="newMajor()">New major version</button>
    <app-version-child [major]="major" [minor]="minor"></app-version-child>
  `
})
export class VersionParentComponent {
  major = 1;
  minor = 23;

  newMinor() {
    this.minor++;
  }

  newMajor() {
    this.major++;
    this.minor = 0;
  }
}
```

```
import { Component, Input, OnChanges, SimpleChange } from '@angular/core';

@Component({
  selector: 'app-version-child',
  template: `
    <h3>Version {{major}}.{{minor}}</h3>
    <h4>Change log:</h4>
    <ul>
      <li *ngFor="let change of changeLog">{{change}}</li>
    </ul>
  `
})
export class VersionChildComponent implements OnChanges {
  @Input() major: number;
  @Input() minor: number;
  changeLog: string[] = [];

  ngOnChanges(changes: SimpleChanges) {
    let log: string[] = [];
    for (let propName in changes) {
      let changedProp = changes[propName];
      let to = JSON.stringify(changedProp.currentValue);
      if (changedProp.isFirstChange()) {
        log.push(`Initial value of ${propName} set to ${to}`);
      } else {
        let from = JSON.stringify(changedProp.previousValue);
        log.push(`${propName} changed from ${from} to ${to}`);
      }
    }
    this.changeLog.push(log.join(', '));
  }
}
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

Hijo-Padre a través de eventos

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

@Component({
  selector: 'app-vote-taker',
  template: `
    <h2>Should mankind colonize the Universe?</h2>
    <h3>Agree: {{agreed}}, Disagree: {{disagreed}}</h3>
    <app-voter *ngFor="let voter of voters"
      [name]="voter"
      (voted)="onVoted($event)">
    </app-voter>
  `
})
export class VoteTakerComponent {
  agreed = 0;
  disagreed = 0;
  voters = ['Narco', 'Celeritas', 'Bombasto'];

  onVoted(agreed: boolean) {
    agreed ? this.agreed++ : this.disagreed++;
  }
}
```

```
import { Component, EventEmitter, Input, Output } from '@angular/core';

@Component({
  selector: 'app-voter',
  template: `
    <h4>{{name}}</h4>
    <button (click)="vote(true)"  [disabled]="didVote">Agree</button>
    <button (click)="vote(false)" [disabled]="didVote">Disagree</button>
  `
})
export class VoterComponent {
  @Input()  name: string;
  @Output() voted = new EventEmitter<boolean>();
  didVote = false;

  vote(agreed: boolean) {
    this.voted.emit(agreed);
    this.didVote = true;
  }
}
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

Acceso Padre a variables Hijo

```
import { Component }              from '@angular/core';
import      {      CountdownTimerComponent      }      from
'./countdown-timer.component';

@Component({
  selector: 'app-countdown-parent-lv',
  template: `
  <h3>Countdown to Liftoff (via local variable)</h3>
  <button (click)="timer.start()">Start</button>
  <button (click)="timer.stop()">Stop</button>
  <div class="seconds">{{timer.seconds}}</div>
  <app-countdown-timer #timer></app-countdown-timer>
  `,
  styleUrls: ['../assets/demo.css']
})
export class CountdownLocalVarParentComponent { }
```

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

@Component({
  selector: 'app-countdown-timer',
  template: '<p>{{message}}</p>'
})
export class CountdownTimerComponent implements OnInit, OnDestroy {

  intervalId = 0;
  message = '';
  seconds = 11;

  clearTimer() { clearInterval(this.intervalId); }

  ngOnInit()    { this.start(); }
  ngOnDestroy() { this.clearTimer(); }

  start() { this.countDown(); }
  stop()  {
    this.clearTimer();
    this.message = `Holding at T-${this.seconds} seconds`;
  }

  private countDown() {
    this.clearTimer();
    this.intervalId = window.setInterval(() => {
      this.seconds -= 1;
      if (this.seconds === 0) {
        this.message = 'Blast off!';
      } else {
        if (this.seconds < 0) { this.seconds = 10; } // reset
        this.message = `T-${this.seconds} seconds and counting`;
      }
    }, 1000);
  }
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Comunicación entre *Services*

```
import { Injectable } from '@angular/core';
import { Subject }    from 'rxjs';

@Injectable()
export class MissionService {

  // Observable string sources
  private missionAnnouncedSource = new Subject<string>();
  private missionConfirmedSource = new Subject<string>();

  // Observable string streams
  missionAnnounced$ = this.missionAnnouncedSource.asObservable();
  missionConfirmed$ = this.missionConfirmedSource.asObservable();

  // Service message commands
  announceMission(mission: string) {
    this.missionAnnouncedSource.next(mission);
  }

  confirmMission(astronaut: string) {
    this.missionConfirmedSource.next(astronaut);
  }
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

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

import { MissionService } from './mission.service';
import { Subscription }   from 'rxjs';

@Component({
  selector: 'app-astronaut',
  template: `
    <p>
      {{astronaut}}: <strong>{{mission}}</strong>
      <button
        (click)="confirm()"
        [disabled]="!announced || confirmed">
        Confirm
      </button>
    </p>
  `
})
export class AstronautComponent implements OnDestroy {
  @Input() astronaut: string;
  mission = '<no mission announced>';
  confirmed = false;
  announced = false;
  subscription: Subscription;

  constructor(private missionService: MissionService) {
    this.subscription = missionService.missionAnnounced$.subscribe(
      mission => {
        this.mission = mission;
        this.announced = true;
        this.confirmed = false;
    });
  }

  confirm() {
    this.confirmed = true;
    this.missionService.confirmMission(this.astronaut);
  }

  ngOnDestroy() {
    // prevent memory leak when component destroyed
    this.subscription.unsubscribe(); }}
```

```
import { Injectable } from '@angular/core';
import { Subject }    from 'rxjs';

@Injectable()
export class MissionService {

  // Observable string sources
  private missionAnnouncedSource = new Subject<string>();
  private missionConfirmedSource = new Subject<string>();

  // Observable string streams
  missionAnnounced$ = this.missionAnnouncedSource.asObservable();
  missionConfirmed$ = this.missionConfirmedSource.asObservable();

  // Service message commands
  announceMission(mission: string) {
    this.missionAnnouncedSource.next(mission);
  }

  confirmMission(astronaut: string) {
    this.missionConfirmedSource.next(astronaut);
  }
}
```

```
import { Component }         from '@angular/core';
import { MissionService }     from './mission.service';

@Component({
  selector: 'app-mission-control',
  template: `
    <h2>Mission Control</h2>
    <button (click)="announce()">Announce mission</button>
    <app-astronaut *ngFor="let astronaut of astronauts"
      [astronaut]="astronaut">
    </app-astronaut>
    <h3>History</h3>
    <ul>
      <li *ngFor="let event of history">{{event}}</li>
    </ul>
  `,
  providers: [MissionService]
})
export class MissionControlComponent {
  astronauts = ['Lovell', 'Swigert', 'Haise'];
  history: string[] = [];
  missions = ['Fly to the moon!',
              'Fly to mars!',
              'Fly to Vegas!'];
  nextMission = 0;

  constructor(private missionService: MissionService) {
    missionService.missionConfirmed$.subscribe(
      astronaut => {
        this.history.push(`${astronaut} confirmed the mission`);
      });
  }

  announce() {
    let mission = this.missions[this.nextMission++];
    this.missionService.announceMission(mission);
    this.history.push(`Mission "${mission}" announced`);
    if (this.nextMission >= this.missions.length) { this.nextMission = 0; }
  }
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

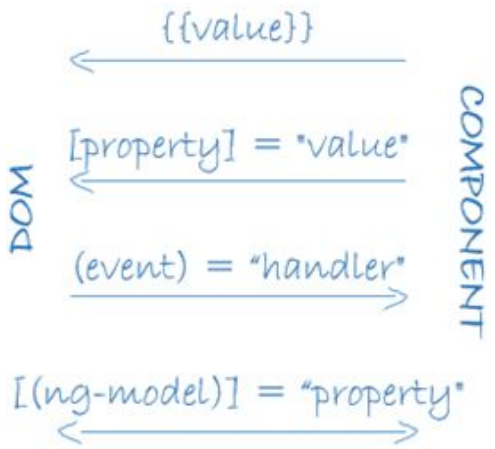
Conceptos avanzados de componentes

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Sintáxis binding

Angular posee diferentes tipos de data-binding, se podrían agrupar en tres categorías:

- *Source-to-view*. Desde el componente a la vista.
- *View-to-source*. Desde la vista al componente.
- *Two-way*. Doble binding



Categoría	Tipo	Sintáxis
Source-to-view	Interpolacion Propiedades Atributos Clases Estilos	{{expression}} [target]=”expression” bind-target=”expression”
View-to-source	Eventos	(target)=”statement” on-target=”statement”
Two-way		[(target)]=”expression” bindon-target=”expression”

Tipo	Target	Ejemplo
Propiedad	Elemento Componente Directiva	<img [src]="heroImageUrl"> <app-hero-detail [hero]="currentHero"></app-hero-detail> <div [ngClass]="{'special': isSpecial}">
Evento	Elemento Componente Directiva	<button (click)="onSave()">Save</button> <app-hero-detail (deleteRequest)="deleteHero()"></app-hero-detail> <div (myClick)="clicked=$event" clickable>click me
Two-way	Eventos Propiedades	<input [(ngModel)]="name">
Atributos	Atributos	<button [attr.aria-label]="help">help</button>
Clases	Clases	<div [class.special]="isSpecial">Special
Estilo	Estilo	<button [style.color]="isSpecial ? 'red' : 'green'">

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Two-way binding [(...)]

```
<app-sizer [(size)]="fontSizePx"></app-sizer>
<div [style.font-size.px]="fontSizePx">Resizable Text</div>
```

```
<app-sizer [size]="fontSizePx"
(sizeChange)="fontSizePx=$event"></app-sizer>
```

```
import { Component, Input, Output, EventEmitter } from '@angular/core';

@Component({
  selector: 'app-sizer',
  templateUrl: './sizer.component.html',
  styleUrls: ['./sizer.component.css']
})
export class SizerComponent {


  @Input()  size: number | string;
  @Output() sizeChange = new EventEmitter<number>();

  dec() { this.resize(-1); }
  inc() { this.resize(+1); }

  resize(delta: number) {
    this.size = Math.min(40, Math.max(8, +this.size + delta));
    this.sizeChange.emit(this.size);
  }

}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

[(ngModel)]

```
<label for="example-ngModel">[(ngModel)]:</label>
<input [(ngModel)]="currentItem.name" id="example-ngModel">
```

```
<label for="example-change">(ngModelChange)="...name=$event":</label>
<input [ngModel]="currentItem.name"
(ngModelChange)="currentItem.name=$event" id="example-change">
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

NgClass

```
<div [ngClass]="currentClasses">This div is initially saveable, unchanged, and special.</div>
```

```
currentClasses: {};
setCurrentClasses() {
  // CSS classes: added/removed per current state of component properties
  this.currentClasses =  {
    'saveable': this.canSave,
    'modified': !this.isUnchanged,
    'special':  this.isSpecial
  };
}
```

```
<!-- toggle the "special" class on/off with a property -->
<div [ngClass]="isSpecial ? 'special' : ''">This div is special</div>
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

NgStyle

```
<div [ngStyle]="currentStyles">
  This div is initially italic, normal weight, and extra large (24px).
</div>
```

```
currentStyles: {};
setCurrentStyles() {
   // CSS styles: set per current state of component properties
  this.currentStyles = {
    'font-style':  this.canSave      ? 'italic' : 'normal',
    'font-weight': !this.isUnchanged ? 'bold'   : 'normal',
    'font-size':   this.isSpecial    ? '24px'   : '12px'
  };
}
```

```
<div [style.font-size]="isSpecial ? 'x-large' : 'smaller'">
  This div is x-large or smaller.
</div>
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

NgSwitch

```
<div [ngSwitch]="currentItem.feature">
  <app-stout-item    *ngSwitchCase="'stout'"    [item]="currentItem"></app-stout-item>
  <app-device-item   *ngSwitchCase="'slim'"     [item]="currentItem"></app-device-item>
  <app-lost-item     *ngSwitchCase="'vintage'"  [item]="currentItem"></app-lost-item>
  <app-best-item     *ngSwitchCase="'bright'"   [item]="currentItem"></app-best-item>
<!-- . . . -->
  <app-unknown-item  *ngSwitchDefault           [item]="currentItem"></app-unknown-item>
</div>
```

Esta directiva trabaja tanto con componentes como elementos nativos:

```
<div *ngSwitchCase="'bright'"> Are you as bright as {{currentItem.name}}?</div>
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Variable #var para referenciar elementos

```
<input #phone placeholder="phone number" />

<!-- lots of other elements -->

<!-- phone refers to the input element; pass its `value` to an event handler -->
<button (click)="callPhone(phone.value)">Call</button>
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Pipes (|)

El valor de cualquier variable a veces necesita ser transformada para mostrarse en la plantilla. Los *Pipes* son simples funciones que aceptan un valor y lo devuelve transformado

```
<p>Title through uppercase pipe: {{title | uppercase}}</p>
```

```
<!-- convert title to uppercase, then to lowercase -->
<p>Title through a pipe chain: {{title | uppercase | lowercase}}</p>
```

```
<!-- pipe with configuration argument => "February 25, 1980" -->
<p>Manufacture date with date format pipe: {{item.manufactureDate | date:'longDate'}}</p>
```

```
<p>Item json pipe: {{item | json}}</p>
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

El operador (?)

```
<p>The item name is: {{item?.name}}</p>
```

Esta opción se utiliza para variables que son usadas en la plantilla pero que esperan respuesta de un servidor para ser inicializadas por lo que durante un breve momento son *null* o *undefined* y dan un error en el navegador. Dicho error no es grave y la aplicación continua su ejecución pero en algunos casos sí que hace que falle la plantilla y no se muestre la información.

Con este operador hacemos que la plantilla se renderice y una vez que se devuelva el valor del servidor mostrarlo.

Eventos

click	blur	change
focus	keypress	submit

En la siguiente web se pueden ver una lista completa de todos los eventos:

https://www.w3schools.com/jsref/dom_obj_event.asp

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Conceptos avanzados de Angular

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

NgModule API

```
@NgModule({
  // Static, that is compiler configuration
  declarations: [], // Configure the selectors
  entryComponents: [], // Generate the host factory

  // Runtime, or injector configuration
  providers: [], // Runtime injector configuration

  // Composability / Grouping
  imports: [], // composing NgModules together
  exports: [] // making NgModules available to other parts of the app
})
```

Propiedad	Descripción
declarations	Clases declarables (componentes, directivas y *pipes*). Todas estas clases se declaran en un solo módulo, según necesitemos, si no dará fallo el compilador.
providers	Clases inyectables (*Services* y algunas clases). Angular tiene asociado a cada módulo un injector, por lo que el módulo raíz tiene su injector raíz y cada submódulo tiene el suyo propio que parte del principal.
imports	Todos aquellos módulos o librerías de Angular que se vayan a utilizar en la aplicación.
exports	Clases declarables (componentes, directivas y *pipes*) que quieran ser importadas por otro módulo
bootstrap	Componente de arranque de la aplicación
entryComponents	Componentes que pueden ser cargados dinámicamente en la vista

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Multimodulado

Angular es modular, tiene la posibilidad de dividir sus aplicaciones en diferentes áreas. Esto aligera la primera petición al servidor para cargar la aplicación web además de organizar nuestro proyecto por diferentes funcionalidades.

```
ng generate module nameModule
```

Una vez creado el submódulo, para crear cualquier componente y añadirlo a ese submódulo:

```
ng generate component nameComponent
```

Para visualizar los componentes de esos submódulos podemos hacerlo de dos maneras:

- A través de los *exports* del submódulo.
- Añadiendo rutas al submódulo y a sus componentes.

Para esta segunda opción debemos de ir al *app-routing* e introducir la ruta del submódulo:

```
{ path: 'customer', loadChildren: '.. rutadelarchivo/customer-dashboard.module#CustomerDashboarModule' }
```

Una vez añadida la ruta al submódulo, dentro de este debemos crear las rutas pertinentes a sus componentes:

```
…
imports: [
  …
  RouterModule.forChild([
    { path: "", component: CustomerDashboardComponent }
  ]),
  …
]
...
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Providers

Providers es una instrucción del Inyector de Dependencias para obtener el valor de una dependencia. Esto quiere decir que todos los archivos que Angular detecte que se almacenan en el ID deben de ser importados en el *providers* para poder ser utilizados. La mayoría de estos archivos suelen ser *services*. Dependiendo del nivel donde importemos ese archivo podremos usarlo:

- Toda la aplicación
  - Providers del módulo principal
  - @Injectable({ providedIn: 'root' })
- Para un solo submódulo
  - Providers de ese submódulo
  - @Injectable({ providedIn: nombreModulo })
- Para un solo componente
  - Providers de ese componente

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Routing - Children

Hasta ahora se había visto como crear rutas sencillas entre componentes. Pero se pueden hacer varios niveles de enrutado para una aplicación web.

```
const crisisCenterRoutes: Routes = [
  {
    path: 'crisis-center', component: CrisisCenterComponent, children: [
      {
        path: '', component: CrisisListComponent,  children: [
          {
            path: ':id',  component: CrisisDetailComponent
          },
          {
            path: '', component: CrisisCenterHomeComponent
          }
        ]
      }
    ]
  }
];
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Routing - CanActivate

Se utiliza para limitar a los usuarios el acceso a las rutas que nosotros no queramos que accedan.

```
import { Injectable } from '@angular/core';
import { CanActivate, ActivatedRouteSnapshot, RouterStateSnapshot } from '@angular/router';

@Injectable({
  providedIn: 'root',
})
export class AuthGuard implements CanActivate {
    canActivate( next: ActivatedRouteSnapshot, state: RouterStateSnapshot): boolean {
      // Código para comprobar el token o clave para devolver true o false. Lo habitual justo antes de devolver false se reenvie con la clase router de Angular a otra vista, si no la app se quedará parada en este punto.
  }
}
```

```
{ path: 'categorias', canActivate: [AuthGuard], component: 'CategoriasComponent' }
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Eventos

Durante la navegación, el objeto *Router* emite una serie de eventos a través de la propiedad *Router.events*.

```
this.router.events.subcribe( (event) =>
{
   console.log(event);
});
```

NavigationStart	NavigationEnd	NavigationError
GuardsCheckEnd	ChildActivationStart	ResolveStart

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Inyector de dependencias

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Servicios que necesitan otros servicios

```
import { Injectable } from '@angular/core';
import { HEROES }     from './mock-heroes';
import { Logger }     from '../logger.service';

@Injectable({
  providedIn: 'root',
})
export class HeroService {

  constructor(private logger: Logger) {  }

  getHeroes() {
    this.logger.log('Getting heroes ...');
    return HEROES;
  }
}
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Dependencias opcionales

Cuando un componente o servicio declara una dependencia, el constructor de la clase toma esa dependencia como parámetro pero podemos decirle a Angular que la dependencia es opcional anotando el parámetro *@Optional()* en el constructor.

```
constructor(@Optional() private logger: Logger) { }
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

HttpClient

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Solicitud de respuesta

config.ts

```
export interface Config {
  heroesUrl: string;
  textfile: string;
}
```

config.service.ts

```
getConfig() {
  // now returns an Observable of Config
  return this.http.get<Config>(this.configUrl);
}
```

config.component.ts

```
config: Config;

showConfig() {
  this.configService.getConfig()
      // clone the data object, using its known Config
shape
    .subscribe((data: Config) => this.config = { ...data });
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Respuesta completa

```
getConfigResponse():
Observable<HttpResponse<Config>> {
  return this.http.get<Config>(
    this.configUrl, { observe: 'response' });
}
```

```
showConfigResponse() {
  this.configService.getConfigResponse()
    // resp is of type `HttpResponse<Config>`
    .subscribe(resp => {
      // display its headers
      const keys = resp.headers.keys();
      this.headers = keys.map(key =>
        `${key}: ${resp.headers.get(key)}`);

      // access the body directly, which is typed as `Config`.
      this.config = { ... resp.body };
    });
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Peticiones no JSON

```
getTextFile(filename: string) {
  // The Observable returned by get() is of type Observable<string>
  // because a text response was specified.
  // There's no need to pass a <string> type parameter to get().
  return this.http.get(filename, {responseType: 'text'})
    .pipe(
      tap( // Log the result or error
        data => this.log(filename, data),
        error => this.logError(filename, error)
      )
    );
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Formularios

En desarrollo web las aplicaciones usan formularios para loguear usuarios, actualizar un perfil, introducir información sensible y actualizar muchos otros datos.

Angular provee dos tipos de formularios:

- *Reactive forms*: son más robustos, son más escalables, reusables y testeables. Si la aplicación se base mayoritariamente en formularios, una zona de administración por ejemplo, se debe usar esta forma de generar formularios.
- *Template-driven forms*: son más útiles para añadir simples formularios.

	Reactive	***Template-driven***
Preparación	Se crea en el componente	Creado por directivas
Modelo	Estructural	No estructural
Previsibilidad	Síncrona	Asíncrona
Validación	Funciones	Directivas
Mutabilidad	Inmutable	Mutable
Escalabilidad	Acceso a API a bajo nivel	Abstracción de la API

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Reactive Forms

Los formularios *reactive* utilizan un enfoque explícito e inmutable para administrar el estado de un formulario. Cada cambio en el estado del formulario devuelve un nuevo estado, que mantiene la integridad del modelo entre cambios. También se debe de tener en cuenta a la hora de trabajar con ellos que se crean bajo la lógica de *Observable* de Angular

```
import { ReactiveFormsModule } from '@angular/forms';

@NgModule({
  imports: [
    // other imports ...
    ReactiveFormsModule
  ],
})
export class AppModule { }
```

```
import { Component } from '@angular/core';
import { FormControl } from '@angular/forms';

@Component({
  selector: 'app-name-editor',
  templatel: '<label>  Name:
      <input type="text" [formControl]="name">
    </label>',
  styleUrls: ['./name-editor.component.css']
})
export class NameEditorComponent {
  name = new FormControl('');
}
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

FormGroup

```
import { Component } from '@angular/core';
import { FormGroup, FormControl } from '@angular/forms';

@Component({
  selector: 'app-profile-editor',
  templateUrl: './profile-editor.component.html',
  styleUrls: ['./profile-editor.component.css']
})
export class ProfileEditorComponent {
  profileForm = new FormGroup({
    firstName: new FormControl(''),
    lastName: new FormControl(''),
  });
}
```

```
<form [formGroup]="profileForm">

  <label>
    First Name:
    <input type="text" formControlName="firstName">
  </label>

  <label>
    Last Name:
    <input type="text" formControlName="lastName">
  </label>

</form>
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

FormGroup anidados

```
import { Component } from '@angular/core';
import { FormGroup, FormControl } from '@angular/forms';

@Component({
  selector: 'app-profile-editor',
  templateUrl: './profile-editor.component.html',
  styleUrls: ['./profile-editor.component.css']
})
export class ProfileEditorComponent {
  profileForm = new FormGroup({
    firstName: new FormControl(''),
    lastName: new FormControl(''),
    address: new FormGroup({
      street: new FormControl(''),
      city: new FormControl(''),
      state: new FormControl(''),
      zip: new FormControl('')
    })
  });
}
```

```
<div formGroupName="address">
  <h3>Address</h3>
  <label> Street:
    <input type="text" formControlName="street">
  </label>
  <label> City:
    <input type="text" formControlName="city">
  </label>
  <label> State:
    <input type="text" formControlName="state">
  </label>
  <label> Zip Code:
    <input type="text" formControlName="zip">
  </label>
</div>
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

FormBuilder

FormBuilder es una forma diferente de realizar también un formulario *reactive*.

```
import { Component } from '@angular/core';
import { FormBuilder } from '@angular/forms';

@Component({
  selector: 'app-profile-editor',
  templateUrl: './profile-editor.component.html',
  styleUrls: ['./profile-editor.component.css']
})
export class ProfileEditorComponent {
  profileForm = this.fb.group({
    firstName: [''],
    lastName: [''],
    address: this.fb.group({
      street: [''],
      city: [''],
      state: [''],
      zip: ['']
    }),
  });

  constructor(private fb: FormBuilder) { }
}
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Template-driven forms

```
import { FormsModule } from '@angular/forms';

@NgModule({
  imports: [
    // other imports ...
    FormsModule
  ],
})
export class AppModule { }
```

```
<form (ngSubmit)="onSubmit()" #heroForm="ngForm">
    <div class="form-group">
      <label for="name">Name</label>
      <input type="text" class="form-control" id="name" required [(ngModel)]="model.name" name="name"
            #name="ngModel">
      <div [hidden]="name.valid || name.pristine" class="alert alert-danger"> Name is required </div>
    </div>

    <div class="form-group">
      <label for="alterEgo">Alter Ego</label>
     <input type="text" class="form-control" id="alterEgo" [(ngModel)]="model.alterEgo"
           name="alterEgo">
    </div>

    <div class="form-group">
      <label for="power">Hero Power</label>
      <select class="form-control" id="power" required  [(ngModel)]="model.power" name="power"
            #power="ngModel">
        <option *ngFor="let pow of powers" [value]="pow">{{pow}}</option>
      </select>
      <div [hidden]="power.valid || power.pristine" class="alert alert-danger"> Power is required </div>
    </div>

    <button type="submit" class="btn btn-success" [disabled]="!heroForm.form.valid">Submit</button>
    <button type="button" class="btn btn-default" (click)="newHero(); heroForm.reset()">
          New Hero</button>
</form>
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Validación - *Template-driven*

Para añadir validación a este tipo de formularios, se añadirá la misma forma de validación que cualquier formulario de HTML. Angular usa directivas para unir esos atributos con sus funciones de validación:

- Los atributos *required*, *minlenght* o *forbiddenName* en la etiqueta <input>
- *#var=”ngModel”* exporta la clase *NgModel* en una variable. Se usa para validar los estados de la etiqueta donde se coloque.

```
<input id="name" name="name" class="form-control" required minlength="4"
appForbiddenName="bob"
     [(ngModel)]="hero.name" #name="ngModel" >

<div *ngIf="name.invalid && (name.dirty || name.touched)" class="alert alert-danger">

  <div *ngIf="name.errors.required"> Name is required. </div>
  <div *ngIf="name.errors.minlength"> Name must be at least 4 characters long.  </div>
  <div *ngIf="name.errors.forbiddenName"> Name cannot be Bob. </div>

</div>
```

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Validación - *Reactive form*

Para formulario *reactive* la validación se puede implementar también en el HTML pero *FormControl* tiene una opción para pasarle los parámetros de la validación:

```
ngOnInit(): void {
  this.heroForm = new FormGroup({
    'name': new FormControl(this.hero.name, [ Validators.required, Validators.minLength(4),
      forbiddenNameValidator(/bob/i) // <-- Here's how you pass in the custom validator.
    ]),
    'alterEgo': new FormControl(this.hero.alterEgo),
    'power': new FormControl(this.hero.power, Validators.required)
  });

}

get name() { return this.heroForm.get('name'); }

get power() { return this.heroForm.get('power'); }
```

```
<input id="name" class="form-control" formControlName="name" >

<div *ngIf="heroForm.get(‘name’).invalid && (heroForm.get(‘name’).dirty ||
heroForm.get(‘name’).touched)" class="alert alert-danger">

  <div *ngIf="heroForm.get(‘name’).errors.required"> Name is required. </div>
  <div *ngIf="heroForm.get(‘name’).errors.minlength"> Name must be at least 4 characters
long. </div>
  <div *ngIf="heroForm.get(‘name’).errors.forbiddenName"> Name cannot be Bob. </div>
</div>
```

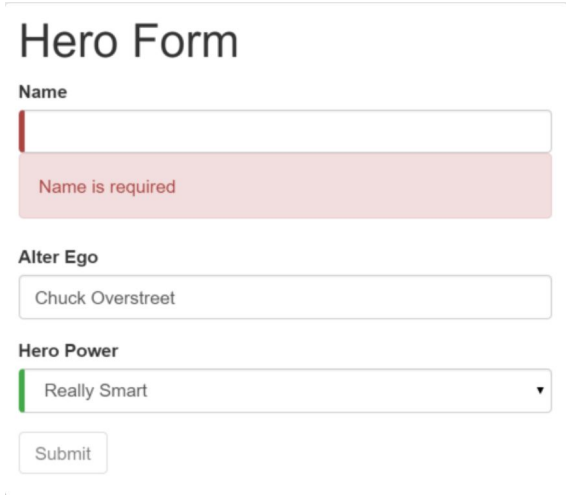
GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Validación - CSS

Angular provee una serie de clases que se añaden automáticamente a los elementos de control, con ellas se puede editar el estilo del elemento según su estado de validación.

.ng-valid	.ng-invalid	.ng-pending	.ng-dirty
.ng-pristine	.ng-untouched	.ng-touched	

```
.ng-valid[required], .ng-valid.required  {
  border-left: 5px solid #42A948; /* green */
}

.ng-invalid:not(form)  {
  border-left: 5px solid #a94442; /* red */
}
```



GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

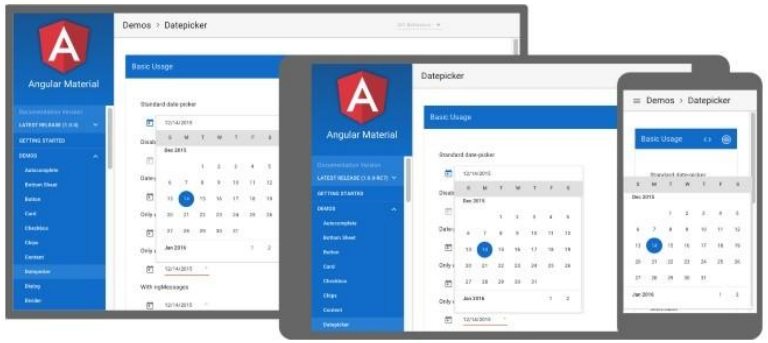
Material Design

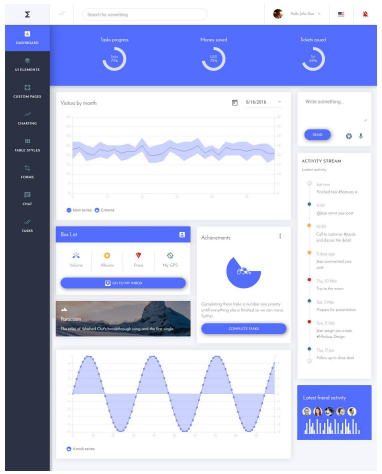
GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
UNIÓN EUROPEA
El FSE invierte en tu futuro

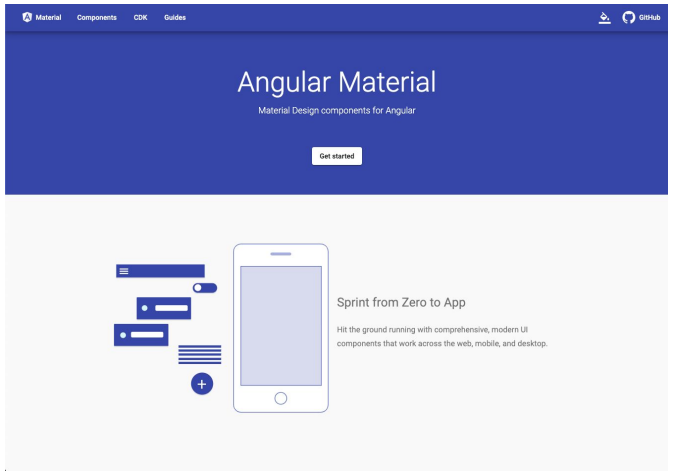
¿Qué es *Material Design?*

Material es un lenguaje de diseño que define un conjunto de pautas que muestran cómo diseñar mejor un sitio web. Le indica qué botones se debe usar y cuáles, cómo animarlos o moverlos, así como dónde y cómo deben colocarse, etc.

Es un lenguaje diseñado principalmente para dispositivos móviles, pero que se puede usar en otras plataformas y aplicaciones.







GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Diferencias entre *Material* y *Bootstrap*

Propósito

Si bien ambos se utilizan para el desarrollo web, *Material* está más orientado hacia la apariencia de un sitio web o aplicación. *Bootstrap*, por otro lado, se centra principalmente en crear fácilmente sitios webs *responsive* y aplicaciones web, que sean funcionales y de alta calidad en lo que respecta a la experiencia del usuario.

Compatibilidad del navegador y marcos de trabajo

Ambos son compatibles con todos los navegadores de hoy día.

En cuanto al framework, *Material* es compatible con *material angular*, *React Material* y utiliza el preprocesador de SASS. *Bootstrap* admite marcos de *React Bootstrap*, *Angular UI Bootstrap* y puede usar los idiomas de SASS y LESS.

Proceso de diseño

Material viene con numerosos componentes que proporcionan un diseño base, que luego puede ser modificados por los propios desarrolladores.

Bootstrap es más una biblioteca UI. Presenta una serie de componentes, como su sistema de cuadrícula avanzado.

En cuanto a componentes de terceros, hay varios compatibles con *Bootstrap*, mientras que con *Material* no hay ninguno.

De los dos, *Material* tiene mucho más atractivo en cuanto a apariencia por sus llamativos colores y sus animaciones, mientras que *Bootstrap* tiene un diseño más estándar.

Documentación y soporte

Bootstrap tiene una documentación y una comunidad bastante más amplia que *Material* debido sobre todo a que este segundo es más nuevo pero para ambos podemos encontrar muchas soluciones a problemas y muchas opciones para una web.

Elección

Lo más adecuado sería usar *Bootstrap* para sitios web *responsive* y más profesionales. *Material* es ideal para crear sitios webs centrados en la apariencia con gran detalle.

Instalación

```
ng add @angular/material
```

Este comando instalará el CDK (*Component Deb Kit*), las librerías de animación de Angular y mientras se instalan todas las demás librerías nos preguntará que deseamos instalar además:

- Tema de *Material*. Nos pedirá que elijamos un estilo de diseño o una plantilla en blanco.
- HammerJS. Nos preguntará si instalar o no esta librería. HammerJS se usa para poder reconocer y usar algunos gestos en la pantalla.
- Importar *BrowserAnimationsModule*.

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Cómo usar *Material*

Para usar material se seguirán estos pasos normalmente:

1. Importar la clase que queremos usar en el módulo.
2. Añadir la etiqueta al html.

```
import { MatSliderModule } from '@angular/material/slider';
…
@NgModule ({....
  imports: [...,
  MatSliderModule,
…]
})
```

```
<mat-slider min="1" max="100" step="1" value="1"></mat-slider>
```

Para más información material.angular.io

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Testing

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Testing en Angular

Realizar test en proyectos cuando superar cierto tamaño es muy importante para probar la aplicación sin tener que hacerlo manualmente. Además si se combina con la integración continua se puede minimizar el riesgo de futuros bugs

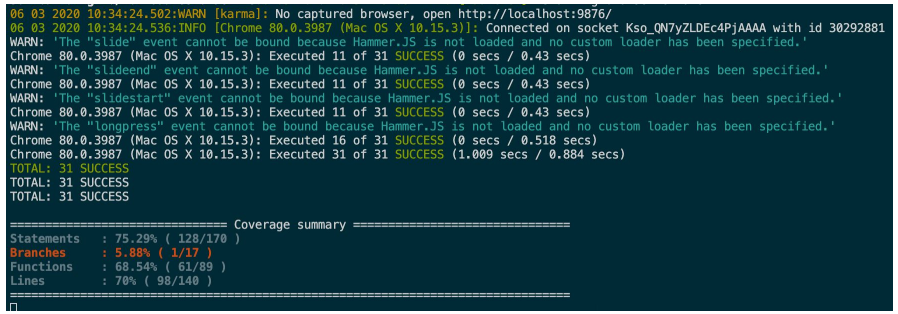
Tipos de test que existen:

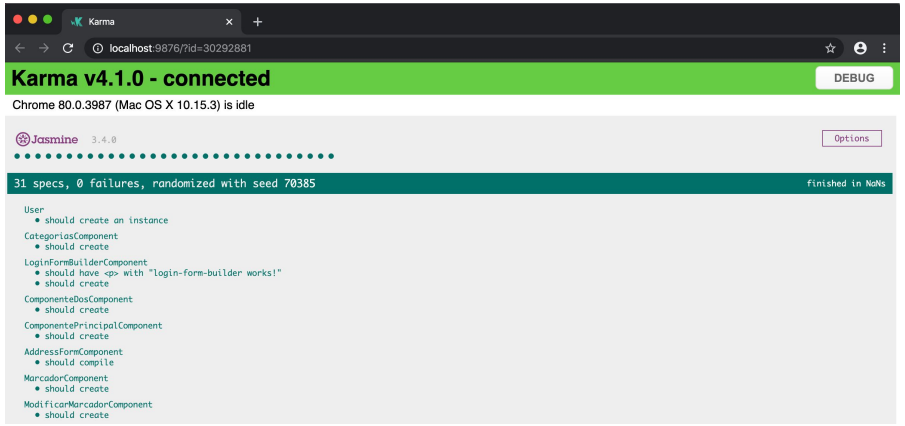
- **Tests Unitarios**: Consiste en probar unidades pequeñas (componentes por ejemplo).
- **Tests End to End (E2E)**: Consiste en probar toda la aplicación simulando la acción de un usuario, es decir, por ejemplo para desarrollo web, mediante herramientas automáticas, se abrirá el navegador y navegará y usará la página como lo haría un usuario normal.
- **Tests de Integración**: Consiste en probar el conjunto de la aplicación asegurando la correcta comunicación entre los distintos elementos de la aplicación. Por ejemplo, en Angular observando cómo se comunican los servicios con la API y con los componentes.

Angular CLI descarga e instala todo lo necesario para testear la aplicación con el framework de *Jasmine*. Para realizar un test de nuestra aplicación:

```
ng test
```

Al introducir este comando Angular realizará un testeo de toda la aplicación e irá recorriendo los diferentes *.spec.ts* de cada clase.





Integración continua

Los servicios de integración continua nos ayudan a mantener nuestro proyecto libre de bugs. Enlazando el repositorio del proyecto con alguna de las herramientas de integración continua se realizarán los test cada vez que se haga un *commit* y un *pull request*.

Las herramientas recomendadas por Angular son Circle CI, Travis CI o Jenkins

Informe de cobertura

Angular CLI puede ejecutar pruebas unitarias y crear informes de cobertura. Estos informes muestran cualquier parte de nuestro código que no pueda ser probada adecuadamente por las pruebas unitarias. Para generar el informe:

```
ng test --code-coverage
```

Este comando creará una carpeta llamada *coverage* en la raíz del proyecto. Al abrir el *index.html* que está dentro de esta carpeta se podrán ver los resultados. Si queremos que se genere automáticamente un informe en cada test debemos de activar esta opción en el *angular.json*:

```
"test": {
  "options": {
    "codeCoverage": true
  }
}
```

Jasmine

Para hacer los test en Angular se suele usar Jasmine. Jasmine es un framework Javascript, para la definición de test usando un lenguaje natural entendible por todo tipo de personas.

Un ejemplo de test sería:

```
describe("A suite name", function() {
  it("contains spec with an expectation", function() {
    expect(true).toBe(true);
  });
});
```

- **describe**: Define una colección de tests. Ésta función recibe dos parámetros, un string con el nombre de la colección y una función donde definiremos los tests.
- **it**: Define un test. Recibe como parámetro el nombre del test y una función a ejecutar por el test.
- **expect**: Lo que espera recibir el test. Es decir, con expect hacemos la comprobación del test. Si la comprobación no es cierta el test falla. En el ejemplo anterior se comprueba si *true* es *true*, por lo que el test pasa.

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

```
describe('Hello world', () => {

  let expected = "";

  beforeEach(() => {
    expected = "Hello World";
  });

  afterEach(() => {
    expected = "";
  });

  it('says hello', () => {
    expect(helloWorld())
        .toEqual(expected);
  });
});
```

```
import { async, ComponentFixture, TestBed } from '@angular/core/testing';

import { NotesComponent } from './notes.component';

describe('NotesComponent', () => {
  let component: NotesComponent;
  // Esta variable nos añadirá más información para que sea más fácil el testeo.
  let fixture: ComponentFixture<NotesComponent>;

  beforeEach(async(() => {
    TestBed.configureTestingModule({
      declarations: [ NotesComponent ]
      // Si tuviéramos algún servicio o dependencia debemos colocarla aquí también como si de un módulo se tratara.
    })
    .compileComponents();
  }));

  beforeEach(() => {
    fixture = TestBed.createComponent(NotesComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

```
import {LoginComponent} from './login.component';
 import {AuthService} from "./auth.service";

 describe('Login component', () => {

   let component: LoginComponent;
   let service: AuthService;

   beforeEach(() => {
     service = new AuthService();
     component = new LoginComponent(service);
   });

   afterEach(() => {
     localStorage.removeItem('token');
     service = null;
     component = null;
   });


   it('canLogin returns true when the user is authenticated', () => {
     localStorage.setItem('token', '12345');
     expect(component.isLogged()).toBeTruthy();
   });


 });
```

Spy de Jasmine

Jasmine ofrece también la posibilidad de coger una clase y devolver directamente lo que nos interese sin tener que ejecutar internamente sus métodos.

Con la función *spyOn* se puede hacer que el servicio devuelva directamente *true* en la llamada al nombre de función que le pasemos como parámetro al spy.

```
import {LoginComponent} from './login.component';
 import {AuthService} from "./auth.service";

 describe('Component: Login', () => {

   let component: LoginComponent;
   let service: AuthService;
   let spy: any;

   beforeEach(() => {
     service = new AuthService();
     component = new LoginComponent(service);
   });

   afterEach(() => {
     service = null;
     component = null;
   });


   it('canLogin returns true when the user is authenticated', () => {
     spy = spyOn(service, 'isAuthenticated').and.returnValue(true);
     expect(component.isLogged()).toBeTruthy();
   });
 });
```

Accediendo a la vista

Para acceder a los elementos html de la vista de un componente, podemos usar su *fixture*.

```
describe('Component: Login', () => {

  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;
  let submitButton: DebugElement;


  beforeEach(() => {

    TestBed.configureTestingModule({
      declarations: [LoginComponent]
    });

    // create component and test fixture
    fixture = TestBed.createComponent(LoginComponent);

    // get test component from the fixture
    component = fixture.componentInstance;

    submitButton = fixture.debugElement.query(By.css('button_submit'));

  });
});
```

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Angular 9, pequeños cambios con Angular 8

La última versión de Angular trae cambios eficientes y adiciones al framework. Esto significa que el estilo del código no tiene nada que cambiar, pero la versión actualizada ofrece correcciones de errores. El lanzamiento de la actualización de Angular 9 que abarcan toda la plataforma, incluyendo el framework, el material angular y la CLI.

Angular 9 es una versión importante porque contiene un nuevo procesador para Angular, llamado "Ivy" . Ivy hace que las aplicaciones sean más pequeñas y rápidas. Las características más importantes aparte de *Ivy* serían:

- Las aplicaciones CLI angulares se compilan en modo AOT de forma predeterminada tanto para la versión developer como producción.
- La última versión incluye comprobación de tipo de plantilla más estrictas que hasta ahora.
- No hay soporte para Typecript 3.1 y 3.5. Tienes que actualizar a Typecript 3.7

Pero para saber bien como actualizar de Angular 8 a Angular 9 debemos de ir a la web oficial.

GOBIERNO DE ESPAÑA
MINISTERIO DE ECONOMÍA Y EMPRESA
red.es
PROFESIONALES DIGITALES | Formación Continua
El FSE invierte en tu futuro
UNIÓN EUROPEA

Ejercicio Final

Para el ejercicio final vamos a hacer:

1. Una vista Login, con su formulario tipo *Template-driven* y una pequeña lógica de *fake* de enviar las credenciales al servidor y que devuelve un *token* que almacenaremos en *localStorage* para comprobar en *canActivate*.
2. Tras el logueo inicial entraremos a un Dashboard donde habrá un menú lateral con al menos 2 vistas de datos. Estos datos *fake* los sacaremos de la web https://jsonplaceholder.typicode.com/. Las tres vistas tendrán una tabla o tarjetas a modo de mostrar toda esa información pero al menos una de esas vistas tendrá:
   a. Una vista Editar, donde podremos editar los campos de ese objeto y hacer un envio *fake* a un servidor, el objeto form a utilizar será un formulario reactivo.
   b. En la misma vista mostrar un modal con la información del objeto.
3. Tras esas primeras 2 vistas crearemos dos más:
   a. Una vista con un ejemplo de libre elección de comunicación Hijo-Padre.
   b. Otra vista con un ejemplo de comunicación entre componentes mediantes servicios.

Información extra:

- El login irá en el módulo principal de la app, pero toda la información del Dashboard irá en un submódulo.
- Realizar el testing del componente login.
- Toda la documentación y ejercicio de ejemplo en https://github.com/ServerJon/cursoAngular8

Bibliografía

- Documentación oficial de Angular 8. Recuperado de https://v8.angular.io/docs
- Wikipedia. Angular (framework). Recuperado de https://es.wikipedia.org/wiki/Angular_(framework)
- Wikipedia. Single page application. Recuperado de https://es.wikipedia.org/wiki/Single-page_application
- Documentación oficial de Typescript. Recuperado de https://www.typescriptlang.org/index.html
- Wikipedia. TypeScript. Recuperado de https://es.wikipedia.org/wiki/TypeScript
- W3schools. HTML DOM Events. Recuperado de https://www.w3schools.com/jsref/dom_obj_event.asp
- Sharing data between Angular components - Four methods (20 Abr 2017) by Jeff Delaney. Fireship. Recuperado de https://fireship.io/lessons/sharing-data-between-angular-components-four-methods/
- Best Practices for Writing Angular Apps | Angular Guidelines (31 Ene 2020). Jsmount. Recuperado de https://www.jsmount.com/best-practices-for-writing-angular-apps/
- Material Design vs Bootstrap: Which One is Better? (6 May 2019) by Anli. Azmind. Recuperado de https://azmind.com/material-design-vs-bootstrap/
- Ractive FormGroup validation with AbstractControl in Angular 2 (26 Oct 2016) by Todd Motto. Ultimatecourses. Recuperado de https://ultimatecourses.com/blog/reactive-formgroup-validation-angular-2
- Documentación oficial de Angular Material 8. Recuperado de https://v8.material.angular.io
- Sitio web: https://jsonplaceholder.typicode.com/
- Angular - Cómo hacer testing unitario con Jasmine (14 Dic 2018). Codingpotions. Recuperado de https://codingpotions.com/angular-testing
- What's new in Angular 9? Top New Features and Ivy (10 Feb 2020) by Manigandan. Agiratech. Recuperado de https://www.agiratech.com/top-new-features-angular-9/
- Angular 9: What's new? (12 Feb 2020) by John Papa and Kapehe Jorgenson. Auth0. Recuperado de https://auth0.com/blog/angular-9-whats-new/

GOBIERNO DE ESPAÑA MINISTERIO DE ECONOMÍA Y EMPRESA

red.es PROFESIONALES DIGITALES | Formación Continua

El FSE invierte en tu futuro

UNIÓN EUROPEA

Muchas gracias por vuestra atención

Jonatan Lucas
Jonatan.lucas@centic.es
@Jon_Lucas_
www.linkedin.com/in/jonatan-lucas-molina-71a23a83