

08

Desenvolupament Web en entorn Client

Cicle: Desenvolupament d'aplicacions web

Curs 2024-2025

©José Masip Alonso

1. Introducció

En aquest document introduïrem Angular.

2. Contingut

2.1. Typescript.

<https://www.typescriptlang.org/es/>

<https://es.wikipedia.org/wiki/TypeScript>

TypeScript es un lenguaje de programación libre y de [código abierto](#) desarrollado y mantenido por [Microsoft](#). Es un superconjunto de [JavaScript](#), que esencialmente añade tipos estáticos y objetos basados en clases.

Exemples

```
let a: any; //variable pot contindre qualsevol valor com a javascript
let b: number;
b = "asdf"; // Error: 'string' not assignable to 'number'
let a: Array<string>;
let b: string[];

class Person {
  private name: string;
  constructor(name: string) { // Recibe un parámetro de tipo string
    this.name = name;
  }

  getName() : string { // Devuelve un valor string
    return this.name;
  }
}
```

Al ser typescript un superconjunt de javascript, tot el codi escrit amb javascript es compatible amb aquest. Typescript es util però no es obligatori coneixer-lo.

2.2. Angular.

[https://es.wikipedia.org/wiki/Angular_\(framework\)](https://es.wikipedia.org/wiki/Angular_(framework))

Angular es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript.

Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos.

Característiques:

- *Velocitat i rendiment.*
 - *Generació de codi: converteix les teues plantilles en codi optimmizat.*
 - *Universal: Genera HTML i CSS.*
 - *Divideix el codi: Sols carrega el codi de la vista sol·licitada*
- *Productivitat:*
 - *Plantilles: Crear rapidament vistes a partir de plantilles simples i potents.*
 - *Angular CLI: Ferramentes en linea de comandaments que permetens afegir components,realitzar test, previsualitzar, etc.*
 - *IDE's: Sugeriments de codi intel·ligent, detecció d'errors,...*

- *Testing:*
 - *karma per a proves unitaries*
 - *Protractor per a proves end-to-end*
- *Animacions.*
- *Integració amb frameworks de disseny com bootstrap, etc.*
- *Accesibilitat.*

Requisits previs:

- *nodejs.* <https://nodejs.org/es/>
- *npm.* <https://www.npmjs.com/>

Tutorial:

<https://docs.angular.lat/tutorial/>

2.3. Angular CLI

Angular CLI es una ferramenta per a crear i gestionar projectes Angular. Permet automatitzar tasques com la compilació, les proves, generació de codi, etc. Ens permet centrar-nos a la programació.

<https://angular.io/cli>

Alguns exemples:

Crear un component	ng g component my-new-component
Crear una directiva	ng g directive my-new-directive
Crear una classe	ng g class my-new-class
Crear un projecte	Ng new elmeuprojecte --style css

2.4. Crear el nostre projecte

Una volta tenim en producció el docker d'angular. Cal crear el nostre projecte. Per a tal fi utilitzem la ferramenta ng.

```
ng new elmeuprojecte --style css

root@0b02317ee065:/opt/app# ng new elmeuprojecte --style css
? Would you like to share pseudonymous usage data about this project with the Angular Team
  at Google under Google's Privacy Policy at https://policies.google.com/privacy. For more
  details and how to change this setting, see https://angular.io/analytics. No
Global setting: disabled
Local setting: No local workspace configuration file.
Effective status: disabled
? Would you like to add Angular routing? No
CREATE elmeuprojecte/README.md (1067 bytes)
CREATE elmeuprojecte/.editorconfig (274 bytes)
CREATE elmeuprojecte/.gitignore (548 bytes)
CREATE elmeuprojecte/angular.json (2735 bytes)
CREATE elmeuprojecte/package.json (1044 bytes)
CREATE elmeuprojecte/tsconfig.json (901 bytes)
CREATE elmeuprojecte/tsconfig.app.json (263 bytes)
CREATE elmeuprojecte/tsconfig.spec.json (273 bytes)
CREATE elmeuprojecte/.vscode/extensions.json (130 bytes)
CREATE elmeuprojecte/.vscode/launch.json (474 bytes)
CREATE elmeuprojecte/.vscode/tasks.json (938 bytes)
CREATE elmeuprojecte/src/favicon.ico (948 bytes)
CREATE elmeuprojecte/src/index.html (299 bytes)
CREATE elmeuprojecte/src/main.ts (214 bytes)
CREATE elmeuprojecte/src/styles.css (80 bytes)
CREATE elmeuprojecte/src/assets/.gitkeep (0 bytes)
CREATE elmeuprojecte/src/app/app.module.ts (314 bytes)
CREATE elmeuprojecte/src/app/app.component.css (0 bytes)
CREATE elmeuprojecte/src/app/app.component.html (23083 bytes)
CREATE elmeuprojecte/src/app/app.component.spec.ts (977 bytes)
CREATE elmeuprojecte/src/app/app.component.ts (217 bytes)
Installing packages (npm)...
```

L'esctructura d'arxius generats és:

- package.json: informació/ configuració npm
- angular.json: configuració general del projecte.
- Tsconfig.*: configuració del compilador de typescript.
- node_modules: carpeta amb les dependències de les aplicacions instal·lades amb npm.
- Src: carpeta amb la nostra aplicació.

2.5. Conceptes fonamentals.

L'arquitectura d'una aplicació Angular es basa en uns conceptes bàsics. Els blocs de construcció bàsics són els NgModules, que proporcionen un context de compilació per als components. Es a dir, recopilen codi relacionants en conjunts funcionals (una aplicació és defineix per un conjunt de NgModules). Una aplicació té una arrel que permet arrancar-la i generalment mes d'un mòdul de funcionalitat.

Els components defineixen vistes, que són conjunts d'elements de la pantalla que Angular pot elegir i modificar d'acord amb la lògica i les dades del programa.

Aquests components utilitzen serveis, els quals proporcionen funcionalitat específica que no està directament relacionada amb les vistes.

Els mòduls, components i serveis són classes que utilitzen decoradors. Aquests indiquen el tipus i proporcionen metadades que indiquen com utilitzar-los.

Les metadades per a un component són associades amb una plantilla que defineix la vista. Aquesta plantilla combina HTML amb directives Angular i markup que permeten modificar el HTML abans de mostrar-lo.

Els components d'una aplicació solen definir moltes vistes, ordenades jeràrquicament. El servei Router t'ajuda a definir rutes de navegació entre les vistes.

2.6. Canvis a l'aplicació d'exemple.

Obrim la carpeta del nostre projecte i entrem a `./src/app`. Observem que tenim un component anomenat `appcomponent` (dividit amb 3 arxius).

- `app.component.ts`— codi per a la classe de component escrita amb TypeScript.
- `app.component.html`— component plantilla escrit amb HTML.
- `app.component.css`— CSS d'aquest component.

A l'arxiu app.component.ts modificar el title i guardar el document.

```
title = 'elmeuprojecte -ieeeee';
```

Si observem el terminal del nostre docker observem que ha detectat la modificació al projecte i ha compilat aquest. Al mateix temps vorem que aquest canvi apareix al navegador.

```
✓ Browser application bundle generation complete.

Initial Chunk Files | Names | Raw Size
main.js             | main  | 46.01 kB |
runtime.js          | runtime | 6.52 kB |

3 unchanged chunks

Build at: 2023-01-08T19:04:38.748Z - Hash: 46db888dc5eba724 - Time: 1231ms

✓ Compiled successfully.
```

A l'arxiu app.component.html modifica tot el contingut o afegeix al final

```
<h1>{{title}}</h1>
```

Les claus dobles permeten la interpolació vinculant d'Angular. Es a dir, passa el valor de la propietat title del component a l'etiqueta h1.

Si generem un error al programar, com ng el compila automàticament ens apareixerà en el terminal. Per exemple, intente accedir a la variable «title» i al definir-la he escrit «titwle»

```
Initial Chunk Files | Names | Raw Size
runtime.js         | runtime | 6.52 kB |
main.js           | main   | 4.47 kB |

3 unchanged chunks

Build at: 2023-01-09T20:15:14.379Z - Hash: 91af671a8d41a84d - Time: 247ms

Error: src/app/app.component.html:1:7 - error TS2551: Property 'title' does not
exist on type 'AppComponent'. Did you mean 'titwle'?

1 <h1>{{title}}</h1>

src/app/app.component.ts:5:16
   templateUrl: './app.component.html',
   ~~~~~
Error occurs in the template of component AppComponent.

* Failed to compile.
```

2.7. Exemple 1 amb angular .

L'interpolació vinculant d'Angular passa el valor de la propietat definida al component a la vista.

```
elmeuprojecte/src/app/app.component.html

<h1> {{title}} </h1>
<h2> {{vector[2]}} </h2>

elmeuprojecte/src/app/app.component.ts

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'hola mon';
  vector = ['aeeee', 'eeeeee', 'ieeeeeee', 'oooooooo', 'ueeeeeee'];
}
```




2.8. Exemple 2 amb angular.

Per a mostrar tots els valors del vector, utilitzem la directiva `*ngfor`. `*ngfor` és l'estructura repetitiva d'angular. Agafa cadascun dels valors i realitza l'acció. En aquest exemple he afegit un estil sols per a que pugau vore on farien la part de disseny (aquesta part correspon a altre mòdul).

elmeuprojecte/src/app/app.component.html
<pre><h1>valors del vector</h1> <li *ngFor="let v of vector">{{v}} </pre>
elmeuprojecte/src/app/app.component.ts
<pre>import { Component } from '@angular/core'; @Component({ selector: 'app-root', templateUrl: './app.component.html', styleUrls: ['./app.component.css'] }) export class AppComponent { title = 'hola mon'; vector = ['aaaaa', 'aaaaaa', 'ieeeeeee', 'oooooooo', 'ueeeeeee']; }</pre>
elmeuprojecte/src/app/app.component.css
<pre>li{ background-color: aqua; }</pre>

valors del vector

- aeeee
- eeeeeee
- ieeeeeeee
- ooooooooo
- ueeeeeeee

2.9. Exemple 3 amb angular.

La directiva *ngIf permet incloure condicionalment blocs de codi HTML. Existeix també el else.

elmeuprojecte/src/app/app.component.html	
<pre><h1>if</h1> <h1 *ngIf="a>=0">positiu</h1></pre>	
elmeuprojecte/src/app/app.component.ts	
<pre>export class AppComponent { title = 'hola mon'; vector = ['aeeee', 'eeeeeee', 'ieeeeeee', 'oooooooo', 'ueeeeeee']; a = 33; }</pre>	
<p>if</p> <p>positiu</p> <p>Amb valor a=33</p>	<p>if</p> <p>Amb valor a=-3</p>

2.10. Exemple 4 amb angular.

A aquest exemple anem a crear un component nou. Per a generar aquest utilitzarem angular CLI.

Cal executar al docker:

ng generate component frutes

```
root@0b02317ee065:/opt/app/elmeuprojecte# ng generate component frutes
CREATE src/app/frutes/frutes.component.css (0 bytes)
CREATE src/app/frutes/frutes.component.html (21 bytes)
CREATE src/app/frutes/frutes.component.spec.ts (599 bytes)
CREATE src/app/frutes/frutes.component.ts (202 bytes)
UPDATE src/app/app.module.ts (396 bytes)
```

Aquesta ordre ens genera els arxius mostrats amb el següent contingut:

elmeuprojecte/src/app/frutes/frutes.component.html
<p>frutes works!</p>
elmeuprojecte/src/app/frutes/frutes.component.ts
import { Component } from '@angular/core'; @Component({ selector: 'app-frutes', templateUrl: './frutes.component.html', styleUrls: ['./frutes.component.css'] }) export class FrutesComponent { }

selector: 'app-frutes', //etiqueta per a utilitzar el component creat

templateUrl: './frutes.component.html', //plantilla per a la vista del component

styleUrls: ['./frutes.component.css'] //disseny del component

Per a utilitzar aquest component simplement l'afegim a alguna plantilla. Per exemple, el podem afegir a la plantilla del component principal de l'app.

```
elmeuprojecte/src/app/app.component.html
```

```
<h1>if</h1>
<h1 *ngIf="a>=0">positiu</h1>

<app-frutes></app-frutes>
```

if

positiu

frutes works!

Aquesta es la configuració o el codi que és genera per defecte. Ara adaptariem aquest component a les nostres necessitats. Primer crearem una interface (declara una serie de metodes i propietats que caldrà implementar en una o mes classes) que represente a cada fruta. Amb açò definirem propietats posseeix les nostres frutes.

ng generate interface fruta

```
root@0b02317ee065:/opt/app/elmeuprojecte# ng generate interface fruta
CREATE src/app/fruta.ts (27 bytes)
```

Aquest arxiu el modificarem així:

```
export interface Fruta {
  identificador :number;
  nom :string;
}
```

Resultat final del component frutes després d'importar la interface i de crear un vector de tipus fruta:

```
elmeuprojecte/src/app/frutes/frutes.component.html
```

```
<p *ngFor="let f of llistat">{{f.identificador + " - " + f.nom}}</p>
```

```
elmeuprojecte/src/app/frutes/frutes.component.ts
```

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

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

```
export class FrutesComponent {  
  llistat: Fruta[] = [  
    {identificador: 1, nom: "taronja"},  
    {identificador: 2, nom: "llima"},  
    {identificador: 3, nom: "melo"}  
  ];  
}
```

```
elmeuprojecte/src/app/app.component.html
```

```
<h1>mostrar les frutes</h1>
```

```
<app-frutes></app-frutes>
```

mostrar les frutes

1 - taronja

2 - llima

3 - melo

3. Activitats.

NOTA:Crea un projecte amb el teu nom i dins realitza els exercicis següents.

- E01. Modifica la pantalla d'inici de l'aplicació per a mostrar el valor de tres propietats definides (elteunom, curs i modul). Mostra aquests valors amb un titol 1, titol 2 i un paragraf. Defineix un fons gris clar per al titol 2.
- E02. Afegeix una propietat on guardar un número si aquest es positiu mostra el caracter an color blau i si es negatiu mostra'l en color roig.
- E03. Volem saber els horàries disponibles de tren. Per a tal fi realitza les següents tasques:
 1. Defineix una interface amb les següents propietats: ident, estacioorige, estaciodesti, horaeixida i horaarribada.
 2. Defineix un vector d'aquesta interface amb un mínim de 5 trens.
 3. Crea un select amb els orogens distints (com el que ferem als exercicis de les unitats anteriors).
 4. Independentment del select, mostra tots els horaris dels trens dins d'una taula. Aquesta taula tindrà una capçalera indicant el valor del camp i tantes files com trens hi ha al vector.

Origen - Desti estacio de tren



origen selecciona una opcio ▼ desti selecciona una opcio ▼

selecciona una opcio

valencia

xativa

4. **Activitats entregables.**

Entrega la carpeta del projecte comprimida en zip o tar.gz:

5. **Bibliografia.**

- <https://docs.angular.lat/tutorial/>
- <https://www.typescriptlang.org/es/>
- <https://angular.io/>

6. **Aclariments**

No intentes copiar exercicis d'altres companys o internet. ni tans sols pegar una miradeta. Si que pots consultar la sintaxis d'una funció, etc. Però no et serveix per a res intentar buscar la solució.

És el major error que pot realitzar qui comença a programar. Sols et deixarà una falsa sensació d'aprenentatge i després no sabràs solucionar els problemes per tu mateix.

Cal intentar que el resultat que genere el vostre codi siga el mes paregut possible a l'exercici (si hi ha imatge d'exemple).

Com els Navegadors no son compatibles entre sí al 100% jo corregire els examens amb FIREFOX. Per tant, es recomanable utilitzar aquest per als nostres exercicis.

Crea uns arxius per a cada exercici o apartat. Així un error a un no afectarà a la resta.

Per a entregar els exercicis crear una carpeta amb el vostre nom i comprimi aquesta (el nom de l'arxiu comprimit serà elteunom.zip o elteunom.tar.gz). Si

entregueu arxius solts o arxius genèrics sense identificar el vostre nom puc confondre'm i no puntuar-vos algun exercici al no saber de qui és.

Cal entregar tots els arxius necessaris per a l'exercici (fotos, llibreries, etc).