



iesperemariaorts

Desarrollo web en entorno cliente

Tema 04

Angular



iesperemariaorts

Angular - Introducción

Es una plataforma que permite desarrollar aplicaciones web en la parte cliente con HTML y JS (mayor velocidad de ejecución).

Permite la creación de aplicaciones web de una sola página (SPA).

Gracias al uso de componentes → encapsular información y mejor mantenimiento.

Está orientado a objetos, trabaja con clase y favorece uso de MVC.

Permite uso de TypeScript. Compilador traduce a JS



Angular – Introducción aplicaciones SPA

Estructura cliente-servidor:

- Cliente – Servidor estático
- Cliente estático – Servidor dinámico
- Cliente y Servidor dinámico

Patrones de diseño:

- Multi page applications (MPA)
- Single page application (SPA)



iesperemariaorts

Angular – Introducción

Nació como **AnguJarJS (2009)**, en la v2.0 se rediseño y "lo absorbió" Google y pasó a llamarse **Angular** (actualmente v10):

- Se rediseñó completamente (basada en componentes).
 - Se introdujo el uso de TypeScript.
- Google sigue dando soporte a versiones antiguas.
- Cambio laborioso, Google documentó proceso de migración (<https://angular.io/guide/upgrade>)



iesperemariaorts

Angular - Instalación

El desarrollo de una aplicación Angular se hace basándose en las herramientas de Node.js, principalmente npm.

The screenshot shows the Node.js website in Spanish. The navigation bar includes links for INICIO, ACERCA, DESCARGAS, DOCUMENTACIÓN, PARTICIPE, SEGURIDAD, NOTICIAS, and CERTIFICATION. The main content area features the Node.js logo and a description: "Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome." Below this is a dark grey button with the text "#BlackLivesMatter". A green banner announces "New security releases now available for 15.x, 14.x and 12.x release lines". The "Descargar para Windows (x64)" section offers two options: "14.15.1 LTS" (Recomendado para la mayoría) and "15.2.1 Actual" (Últimas características). At the bottom, there are links for "Otras Descargas", "Cambios", and "Documentación del API" for both LTS and Actual versions. A footer note mentions "O eche un vistazo a la Programa de soporte a largo plazo (LTS)."

nodejs.org/es/

node

INICIO | ACERCA | DESCARGAS | DOCUMENTACIÓN | PARTICIPE | SEGURIDAD | NOTICIAS | CERTIFICATION

Node.js® es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome.

#BlackLivesMatter

New security releases now available for 15.x, 14.x and 12.x release lines

Descargar para Windows (x64)

14.15.1 LTS
Recomendado para la mayoría

15.2.1 Actual
Últimas características

Otras Descargas | Cambios | Documentación del API | Otras Descargas | Cambios | Documentación del API

O eche un vistazo a la Programa de soporte a largo plazo (LTS).



iesperemariaorts

Angular - Instalación

Actualizamos NPM

```
C:\> npm
Microsoft Windows [Versión 10.0.19042.572]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>npm install -g npm@latest
[.....] | loadRequestedDeps: sill install loadAllDepsIntoIdealTree
```

Deshabilitamos auditoría NPM

```
C:\> Administrador: Símbolo del sistema
C:\Windows\system32>npm set audit false
C:\Windows\system32>
```



iesperemariaorts

Angular - Instalación

Borramos caché NPM

```
C:\Windows\system32>npm cache clean --force
npm WARN using --force I sure hope you know what you are doing.
C:\Windows\system32>
```

Instalamos última versión de angular CLI

```
C:\Windows\system32>npm install -g @angular/cli@latest
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
C:\Users\Usuario\AppData\Roaming\npm\ng -> C:\Users\Usuario\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng
> @angular/cli@11.0.2 postinstall C:\Users\Usuario\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/postinstall/script.js

? Would you like to share anonymous usage data 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 http://angular.io/analytics. Yes

Thank you for sharing anonymous usage data. If you change your mind, the following
command will disable this feature entirely:

  ng analytics off

+ @angular/cli@11.0.2
added 253 packages from 201 contributors in 67.253s
```



iesperemariaorts

Angular - Instalación

Instalamos TypeScript

```
C:\Users\Tomas\ejemplos\ts>npm install -g typescript
C:\Users\Tomas\AppData\Roaming\npm\tsc -> C:\Users\Tomas\AppData\Roaming\npm\node_modules\typescript\bin\tsc
C:\Users\Tomas\AppData\Roaming\npm\tsserver -> C:\Users\Tomas\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
+ typescript@4.1.3
added 1 package from 1 contributor in 2.058s
```




iesperemariaorts

Angular - Instalación

Comprobamos la versión

```
C:\Windows\system32>ng --version
```

Angular CLI

```
Angular CLI: 11.0.2  
Node: 14.15.1  
OS: win32 x64
```

```
Angular:
```

```
...
```

```
Ivy Workspace:
```

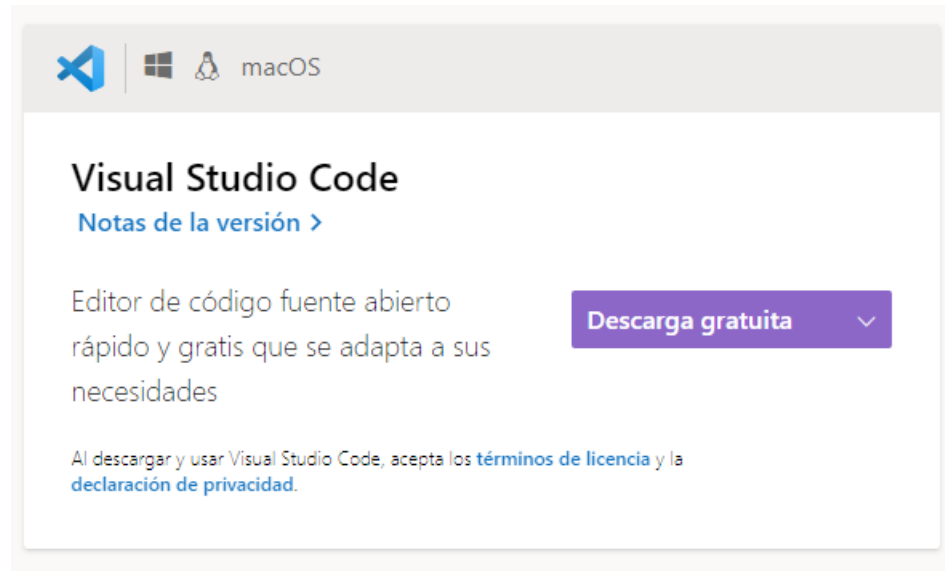
Package	Version
@angular-devkit/architect	0.1100.2 (cli-only)
@angular-devkit/core	11.0.2 (cli-only)
@angular-devkit/schematics	11.0.2 (cli-only)
@schematics/angular	11.0.2 (cli-only)
@schematics/update	0.1100.2 (cli-only)



iesperemariaorts

Angular - Instalación

Instalamos Visual Studio Code



Instalamos extensión "Angular Essentials"





iesperemariaorts

Angular - Proyecto

Generamos un nuevo proyecto

```
Ctrl npm
C:\Users\Tomas>ng new proyecto1
? Would you like to add Angular routing? (y/N) N
```

Opciones: Routing: No y CSS

```
Ctrl npm
C:\Users\Tomas>ng new proyecto1
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
Stylus [ http://stylus-lang.com ]
```



iesperemariaorts

Angular - Proyecto

Comprobación proyecto: Dentro del directorio lanzamos el servidor (ng serve)

```
ngcc (worker)
C:\Users\Tomas\proyecto1>ng serve
? Would you like to share anonymous 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 http://angular.io/analytics. No
Compiling @angular/core : es2015 as esm2015
Compiling @angular/common : es2015 as esm2015
Compiling @angular/platform-browser : es2015 as esm2015
Compiling @angular/platform-browser-dynamic : es2015 as esm2015

chunk {main} main.js, main.js.map (main) 56.9 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 141 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 12.5 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 2.38 MB [initial] [rendered]
Date: 2020-11-30T09:44:34.263Z - Hash: 7e23a1e978aa1f4aea89 - Time: 6939ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
: Compiled successfully.

Date: 2020-11-30T09:44:34.966Z - Hash: 7e23a1e978aa1f4aea89
5 unchanged chunks

Time: 352ms
: Compiled successfully.
```



iesperemariaorts

Angular - Proyecto

Abrimos `http://localhost:4200`

The screenshot shows a web browser window with the title 'Proyecto1' and the address bar displaying 'localhost:4200'. The page has a blue header with the Angular logo and the word 'Welcome'. Below the header, a red circular notification with a rocket icon states 'proyecto1 app is running!'. The main content area is divided into two sections: 'Resources' and 'Next Steps'. The 'Resources' section includes links to 'Learn Angular', 'CLI Documentation', and the 'Angular' website. The 'Next Steps' section includes buttons for 'New Component', 'Angular Material', and 'Add PWA Support'. A large, faint 'V' watermark is visible in the background of the page.

Resources

Here are some links to help you get started:

- [Learn Angular >](#)
- [CLI Documentation >](#)
- [Angular](#)

Next Steps

What do you want to do next with your app?

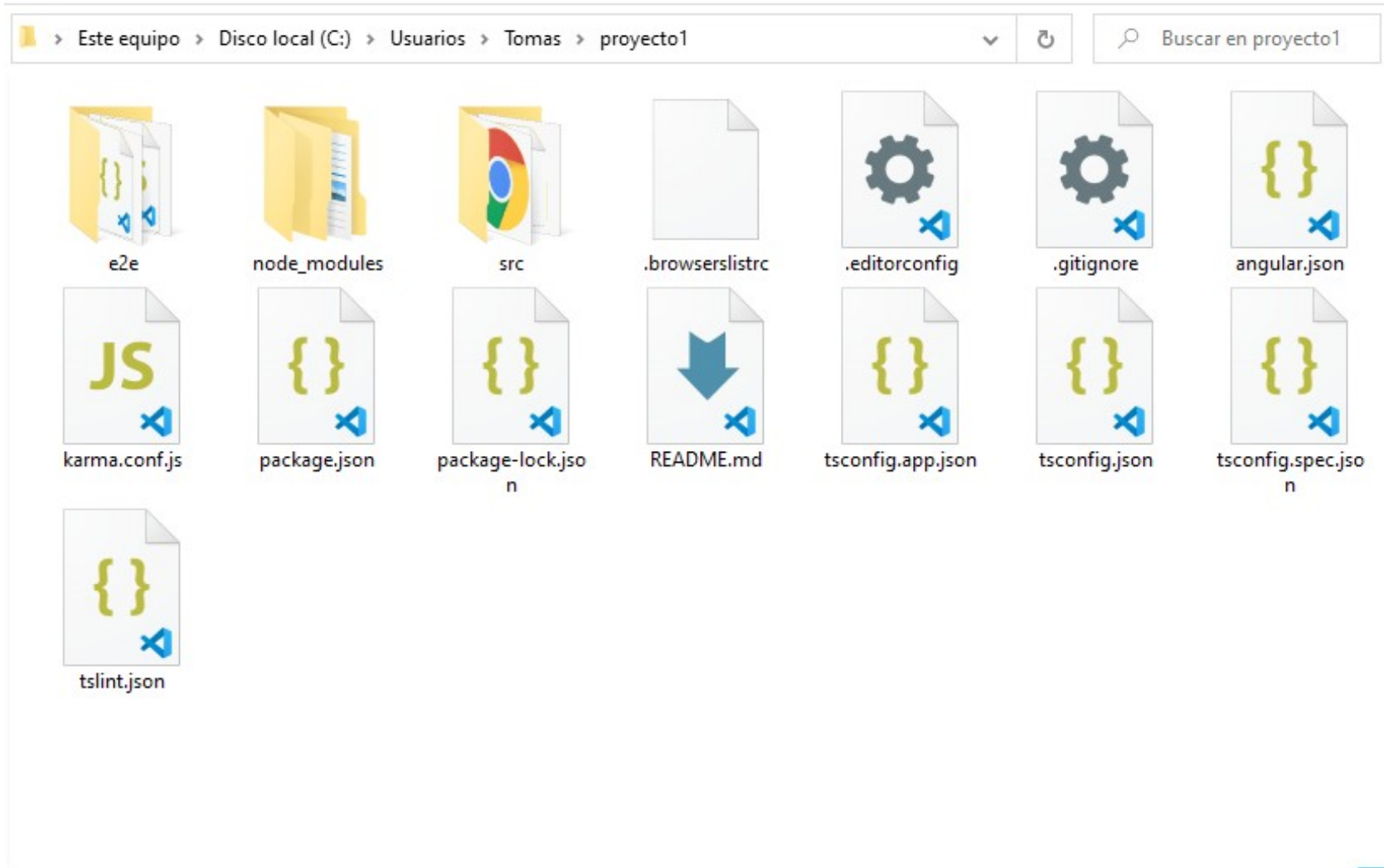
- [+ New Component](#)
- [+ Angular Material](#)
- [+ Add PWA Support](#)



iesperemariaorts

Angular - Proyecto

Estructura de directorios y archivos





iesperemariaorts

Angular - Proyecto

Estructura de directorios y archivos

- NPM (package.json). Entre otras cosas encontramos las dependencias del proyecto
- Angular CLI (angular.json). Configuración global del proyecto Angular.
- El compilador de TypeScript (tsconfig.json)
- [TSLint](#) (tslint.json): una herramienta para gestionar las reglas de estilo de código y sobre qué tipo de errores (y warnings) debe avisar el editor en tiempo de desarrollo. Estos archivos son todos configurables a gusto del desarrollador o equipo de desarrollo.
- [Karma](#) (karma.conf.js): Una herramienta para ejecutar tests del proyecto en el navegador y que usa frameworks como [Jasmine](#) (pruebas unitarias), o [Protractor](#) (e2e/protractor.conf.js): para ejecutar pruebas funcionales.

También encontramos estos subdirectorios:

- e2e: Aquí se definen los tests de integración ([Protractor](#)).
- node_modules: Dependencias de la aplicación instaladas con NPM.
- src: Donde está nuestra aplicación (también tests unitarios: [Jasmine](#))



iesperemariaorts

Angular - Proyecto

Carga inicial de la aplicación

- Se debe tener al menos un módulo principal (**@NgModule**) y tantos submódulos como se quiera.
- Se debe tener un componente raíz que contenga la aplicación, por defecto AppComponent
- Dividir la aplicación es muy útil para separar código y funcionalidad.



Angular – Proyecto – Módulo principal

iesperemariaorts

Representa un contenedor que agrupa al conjunto de elementos de la aplicación.

Se necesita un módulo raíz

Es una clase TypeScript decorada con NgModule.

Módulo principal **src/app/app.module.ts**

```
src > app > TS app.module.ts > ...
```

```
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import { AppComponent } from './app.component';
5
6  @NgModule({
7    declarations: [
8      AppComponent
9    ],
10   imports: [
11     BrowserModule
12   ],
13   providers: [],
14   bootstrap: [AppComponent]
15 })
16 export class AppModule { }
17
```



Angular - Proyecto – Módulo principal

iesperemariaorts

Imports `src/app/app.module.ts`

```
import { BrowserModule } from '@angular/platform-browser';  
import { NgModule } from '@angular/core';  
import { AppComponent } from './app.component';
```

- **BrowserModule:** Este módulo del framework contiene lo necesario para trabajar con la funcionalidad del navegador (manipulación del DOM, eventos, animaciones, etc.). Desde la versión 2, Angular no está fuertemente integrado con el navegador (es un módulo más), por ello puede funcionar en otros entornos, como por ejemplo junto a **NativeScript** para el desarrollo de aplicaciones móviles con componentes nativos, etc.
- **NgModule:** Esto importa el decorador `@NgModule`. Los decoradores son funciones (equivalentes a las anotaciones de Java, por ejemplo), que especifican metadatos que describen una clase o un método. Estos metadatos son interpretados por Angular en este caso, para saber qué tipo de componente del framework (componente, servicio, módulo, filtro, etc.) representa la clase implementada. En este caso hablamos de un módulo (además, es el módulo principal de la aplicación). Existen otros decoradores como `@Component`, `@Pipe`, `@Directive`, `@Injectable`, ...
- **AppComponent:** Es el componente principal de la aplicación. Toda la aplicación se ejecuta dentro de este componente como pronto veremos.



Angular - Proyecto – Módulo principal

iesperemariaorts

Módulo **src/app/app.module.ts**

Tipos de metadatos definidos en el módulo:

```
@NgModule({  
  declarations: [  
    AppComponent  
  ],  
  imports: [  
    BrowserModule  
  ],  
  providers: [],  
  bootstrap: [AppComponent]
```

- **declarations** → **Componentes, directivas y filtros** (pipes) que se utilizarán en este módulo. Como veremos, cuando creamos un componente (o directiva, o filtro) debemos añadirlo a este array para poder usarlo.
- **imports** → En este array añadimos módulos externos (de Angular o creados por nosotros). Los elementos exportados por esos módulos (componentes, etc.) serán accesibles por el módulo actual.
- **exports** → No está presente en el ejemplo actual. En este array añadiremos los componentes, directivas, etc. del módulo actual (declarations) que queramos exportar. Es decir, cuando otro módulo importe el actual, qué cosas podrá usar.
- **providers** → En este array añadimos servicios de Angular. Esta parte la veremos más adelante.
- **bootstrap** → Este array sólo suele encontrarse en el módulo de la aplicación (AppModule). Se define el componente principal o inicial de la aplicación. Lo primero que se carga y se muestra. El resto de componentes se situarán dentro del componente principal.



iesperemariaorts

Angular – Proyecto – Componente raíz

Componente raíz: está formado por tres archivos: el TypeScript, el HTML y la hoja de estilo:

app.component.html: archivo HTML (Vista)

app.component.ts: archivo TypeScript (Controlador)

app.component.css: hoja de estilo

```
▼ app
  # app.component.css
  <> app.component.html
  TS app.component.spec.ts
  TS app.component.ts
```



Angular - Proyecto – Componente raíz

iesperemariaorts

Archivo app.component.ts

```
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.css']
7  })
8  export class AppComponent {
9    title = 'proyecto1';
10 }
```

Es una clase TypeScript que se decora con `@Component`
Podemos definir el selector (`app-root`), es la etiqueta que se corresponde con el componente.

`app.component.html`: va a describir el renderizado HTML del componente.



iesperemariaorts

Angular - Proyecto – Componente raíz

Metadatos en app.component.ts

- **selector** → Nombre de la etiqueta que se utilizará en la vista (HTML) para cargar el componente dentro. Como HTML no es sensible a mayúsculas se utiliza una sintaxis del tipo **kebab-case**. Mira en **src/index.html** donde verás la etiqueta **<app-root>** ya creada (ahí es donde se carga la aplicación).
 - Por defecto, todos los componentes se crean con un selector que tiene el prefijo **app**. Este prefijo sería mejor que tuviera las iniciales de nuestro proyecto (o fuese más descriptivo). Esto se puede configurar al crear el proyecto → **ng new mi-proyecto --prefix mp**. Posteriormente también se puede en el archivo **angular.json** (busca el atributo “**prefix**”).
- **templateUrl** → Este archivo representa la vista asociada al componente. El HTML que se cargará dentro del selector cuando esté todo cargado.
- **styleUrls** → Angular permite asignar una o varias hojas de estilo CSS (o SASS o LESS si se ha creado así el proyecto) a un componente. Estos estilos se aplicarán **solamente** al componente asociado.



Angular - Proyecto – Ejecución

iesperemariaorts

- Tenemos el AppComponent y el AppModule, AppModule sabe que de arrancar con el AppComponent.
- ¿Cómo indicamos a la aplicación que utilice el AppModule?: Con el archivo src/main.ts

```
src > TS main.ts > ...
```

```
1  import { enableProdMode } from '@angular/core';
2  import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3
4  import { AppModule } from './app/app.module';
5  import { environment } from './environments/environment';
6
7  if (environment.production) {
8    | enableProdMode();
9  }
10
11  platformBrowserDynamic().bootstrapModule(AppModule)
12  | .catch(err => console.error(err));
```




iesperemariaorts

Angular - Plantillas

- Es una vista asociada a un componente.
- Podemos utilizar bootstrap para la parte de CSS.

```
C:\Users\Tomas\proyecto1>npm i bootstrap
npm WARN bootstrap@4.5.3 requires a peer of jquery@1.9.1 - 3 but none is
npm WARN bootstrap@4.5.3 requires a peer of popper.js@^1.16.1 but none is
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.1.3 (node_modu
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fs
"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_mod
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fs
y"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_mod
:
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fs
y"} (current: {"os":"win32","arch":"x64"})

+ bootstrap@4.5.3
added 1 package from 2 contributors, removed 1 package and audited 1497 p

76 packages are looking for funding
  run `npm fund` for details

found 4 vulnerabilities (3 low, 1 high)
  run `npm audit fix` to fix them, or `npm audit` for details
```




iesperemariaorts

Angular - Plantillas

- Inclumos bootstrap en src/styles.css

```
styles.css ×  
c > # styles.css  
1  /* You can add global styles to this file, and also import o  
2  @import "../node_modules/bootstrap/dist/css/bootstrap.css"  
3  |
```

- También podemos incluirlo en angular.json

```
"styles": [  
  "node_modules/bootstrap/dist/css/bootstrap.css",  
  "src/styles.css"  
],
```



iesperemariaorts

Desarrollo web en entorno cliente

Tema 04

Angular - TypeScript



iesperemariaorts

TypeScript

- Es un lenguaje compilado que genera JavaScript comprensible por todos los navegadores.
- Es fuertemente tipado: permite asociar un tipo a un elemento e impedir que cambie.
- Se valida sintácticamente el código en la fase de compilación.

```
var id: number;  
id = "1234";
```

Type "'1234'" is no assignable to type 'number'



TypeScript – Sintaxis - Variables

- Declaración:

```
var miVariableBooleana: boolean;
```

- Existen diferentes tipos: boolean, number, string, []

- Declaración con tipado dinámico JS:

```
var miVariableDinamica: any;
```



TypeScript – Sintaxis - Funciones

- Se declaran de manera clásica:

```
function miFuncion(): void {  
    ...  
}
```

- Si reciben argumentos se tipán como variables

```
function miFuncion(miArgumento: string, miOtroArgumento:  
boolean): void {  
    ...  
}
```



iesperemariaorts

TypeScript – Sintaxis - Clases

- Se declaran con la palabra **class**

```
class Person {  
    name: string;  
    age: number;  
  
    constructor(name: string, age: number) {  
        this.name = name;  
        this.age = age;  
  
    toString() {  
        return `Hi I'm ${this.name} and I'm ${this.age} years old!`;  
    }  
}
```



TypeScript – Sintaxis - Clases

- Podemos declarar herencia entre clases con la palabra **extends**.
- Accedemos al padre con método **super**.

```
class Developer extends Person {  
    constructor(name, age, language) {  
        super(name, age);  
        this.language = language;  
    }  
  
    toString() {  
        return super.toString() + `:: I'm a Developer who likes  
${this.language}`;  
    }  
}
```



TypeScript – Sintaxis - Clases

- Permite "visibilidad" de las propiedades:
 - **private**: accesible desde clase actual
- **protected**: accesible desde clase actual o que heredan clase actual.
- **public**: accesible desde exterior de la clase.

```
class Modificar {  
    public myPublicProperty: string;  
    protected myProtectedProperty: string;  
    private myPrivateProperty: string;  
}
```




iesperemariaorts

TypeScript – Sintaxis – Unión de tipos

- Permiten definir si un valor puede tener varios tipos
 - Dinamicidad de Javascript conservando tipado.

```
var pet: Fish | Bird;
```



iesperemariaorts

TypeScript – Sintaxis – Genérico

- Se puede utilizar genéricos (no existe en JS)

```
class Greeter<T> {  
    greeting: T;  
  
    constructor(message: T) {  
        this.greeting = message;  
    }  
    greet() {  
        return this.greeting;  
    }  
}
```

- Luego se instancia especificando el tipo deseado:

```
let greeter = new Greeter<string>("Hello, world");
```



iesperemariaorts

TypeScript – Typings

- Al utilizar librerías de terceros debemos de declarar los tipos de esas librerías (si no serán tipo **any**)
- Las declaraciones se hacen en los archivos de extensión .ts

```
declare module Math {  
    export interface Calculator {  
        addition(a: number, b: number): number;  
    }  
}
```

- Interfaz llamada Calculator en módulo Math que contiene método addition.



iesperemariaorts

TypeScript – Typings

- Al utilizar librerías de terceros debemos de declarar los tipos de esas librerías (si no serán tipo **any**)
- Las declaraciones se hacen en los archivos de extensión d.ts

```
declare module Math {  
    export interface Calculator {  
        addition(a: number, b: number): number;  
    }  
}
```

- Interfaz llamada Calculator en módulo Math que contiene método addition.



TypeScript – Typings

- Es un lenguaje y debe ser compilado, el compilador tsc transforma los archivos TypeScript en archivos JS (**transpilación**)
- El archivo de configuración tsconfig.json contiene los argumentos de compilación. Se encuentra en la raíz de los proyectos angular.



iesperemariaorts

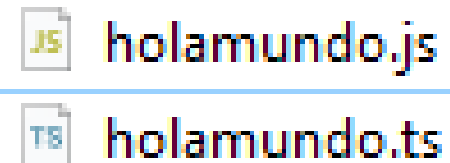
TypeScript – Ejemplos

Archivo holamundo.ts

```
var texto:string="Hola Mundo"  
console.log(texto)
```

Compilamos

```
C:\Users\Tomas\ejemplos>tsc holamundo.ts
```



Ejecutamos

```
C:\Users\Tomas\ejemplos>node holamundo  
Hola Mundo
```