

# Angular

## 4.4. Modelos en Angular

# 1. Modelos

- Un modelo en angular es una clase que representa un objeto con propiedades que permite la utilización de la misma desde clases externas, evitando la reiteración de código.

## 2. Construcción de un modelo

- Estas clases se colocan dentro de src/app en una carpeta llamada **models**.
- El nombre de estos ficheros se escribe en minúscula: **nombrefichero.model.ts**. También se puede utilizar otra nomenclatura y es terminar dicho fichero con la palabra **DTO** antes del ts.

## 2. Construcción de un modelo

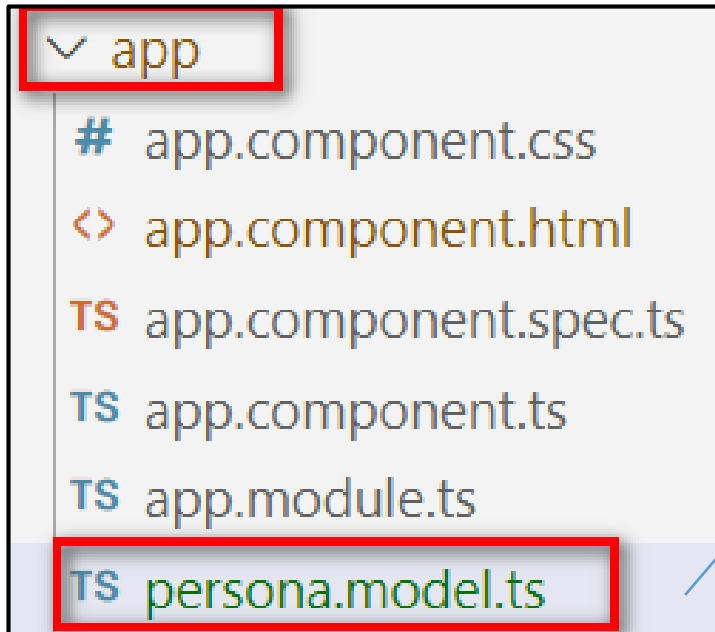
- Si queremos representar un modelo para una persona, los posibles nombres que podemos usar son:
  - **persona.model.ts**
  - **personaDTO.ts**
- Un modelo es una clase que define un objeto con sus respectivas características, su sintaxis es la siguiente:

## 2. Construcción de un modelo

```
export class NombreClase{  
    public propiedad : tipo;  
    public propiedad : tipo;  
    constructor(valor1, valor2){  
        this.propiedad : valor1;  
        this.propiedad: valor2;  
    }  
}
```

### 3. Clase persona

- Vamos a definir un fichero **persona.model.ts** con dos atributos y un constructor



A screenshot of a code editor showing the implementation of the 'Persona' class in 'persona.model.ts'. The breadcrumb navigation at the top reads 'src > app > TS persona.model.ts > Persona > constructor'. The code defines an 'export class Persona' with three attributes: 'nombre', 'apellido', and 'dni', all initialized as empty strings. A 'constructor' method is defined with parameters 'nombre: string', 'apellido: string', and 'dni: string'. Inside the constructor, 'this.nombre' is assigned to 'nombre', 'this.apellido' to 'apellido', and 'this.dni' to 'dni'. The code is as follows:

```
1 export class Persona {
2     nombre="";
3     apellido="";
4     dni="";
5
6
7     constructor(nombre:string,apellido:string,dni:string) {
8         this.nombre=nombre;
9         this.apellido=apellido;
10        this.dni=dni;
11    }
12
13 }
14
```

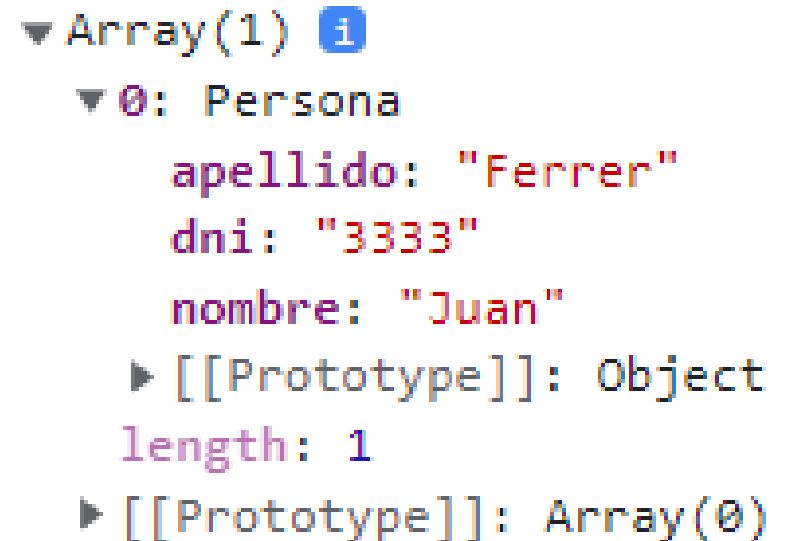
## 4. Usar la clase persona

- Primero debemos importar la clase persona del fichero en nuestro fichero **app.component.ts**

```
app.component.ts M X  
  
> app > TS app.component.ts > ...  
1  import { Component } from '@angular/core';  
2  import { Persona } from './persona.model';  
3
```

## 4. Usar la clase persona

```
export class AppComponent {  
  personas: Persona[]=[];  
  
  constructor() {  
    const p:Persona=new Persona("Juan","Ferrer","3333");  
    this.personas.push(p);  
    console.log(this.personas);  
  }  
}
```



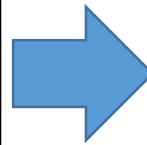
```
▼ Array(1) ⓘ  
  ▼ 0: Persona  
    apellido: "Ferrer"  
    dni: "3333"  
    nombre: "Juan"  
    ► [[Prototype]]: Object  
    length: 1  
    ► [[Prototype]]: Array(0)
```



## 4. Usar la clase persona

- Podemos añadir varios registros de tipo persona al array de Personas.

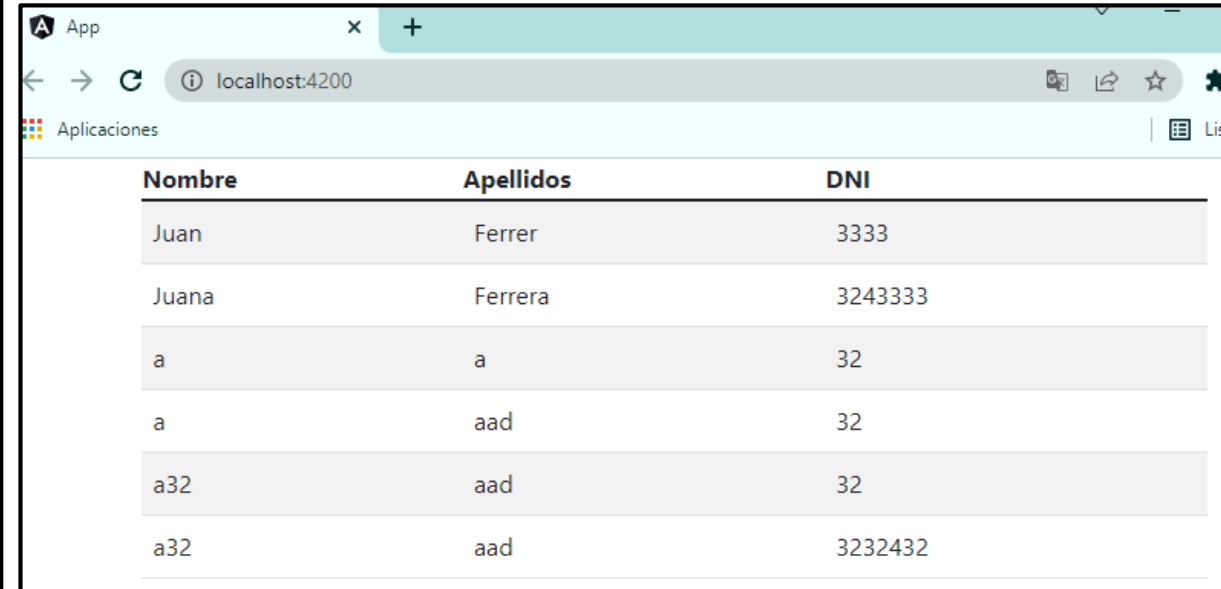
```
export class AppComponent {  
  personas: Persona[]=[];  
  
  constructor() {  
    const p:Persona=new Persona("Juan","Ferrer","3333");  
    this.personas.push(p);  
    const q:Persona=new Persona("Juana","Ferrera","3243333");  
    this.personas.push(q);  
  
    console.log(this.personas);  
  }  
}
```



```
app.component.t  
▼ Array(2) ⓘ  
  ► 0: Persona {nombre: 'Juan', apellido: 'Ferrer', dni: '3333'}  
  ► 1: Persona {nombre: 'Juana', apellido: 'Ferrera', dni: '3243333'}  
  length: 2
```

## 5. Mostrar la clase persona

```
<> app.component.html M X
src > app > <> app.component.html > div.container > hr
Go to component
1 <div class="container">
2   <table class="table table-striped table-hover">
3     <thead>
4       <th>Nombre</th>
5       <th>Apellidos</th>
6       <th>DNI</th>
7     </thead>
8     <tbody>
9       <tr *ngFor="let elemento of personas">
10        <td> {{ elemento.nombre }} </td>
11        <td> {{ elemento.apellido }} </td>
12        <td> {{ elemento.dni }} </td>
13      </tr>
14    </tbody>
15  </table>
```



A screenshot of a web browser window displaying a table of data. The browser's address bar shows 'localhost:4200'. The table has three columns: 'Nombre', 'Apellidos', and 'DNI'. It contains six rows of data, with alternating light and dark gray background colors for each row.

Nombre	Apellidos	DNI
Juan	Ferrer	3333
Juana	Ferrera	3243333
a	a	32
a	aad	32
a32	aad	32
a32	aad	3232432

## 6. Agregar elementos a persona

- Vamos a incluir un formulario que nos permita insertar registros al array.

```
export class AppComponent {  
  personas: Persona[]=[];  
  nombre="";  
  apellido="";  
  dni="";
```

```
<label class="control-label">Nombre</label>  
<input type="text" class="form-control" [(ngModel)]="nombre">  
<label class="control-label">Apellido</label>  
<input type="text" class="form-control" [(ngModel)]="apellido">  
<label class="control-label">DNI</label>  
<input type="text" class="form-control" [(ngModel)]="dni">  
<button class="btn btn-primary" (click)="insertar()">Insertar</button>
```

## 6. Agregar elementos a persona

```
insertar() {  
  const w=new Persona(this.nombre,this.apellido,this.dni);  
  this.personas.push(w);  
}
```

Nombre	Apellidos	DNI
Juan	Ferrer	3333
Juana	Ferrera	3243333

Nombre

Apellido

DNI

Insertar