6.1. Servicios en Angular

1. ¿Qué es un servicio?

- Un servicio es la capa encargada de traer/proporcionarnos los datos a nuestra aplicación de Angular.
- Normalmente, nuestro servicio para acceder a estos datos suele conectarse al servidor donde están almacenada dicha información, por ejemplo, una BBDD, una API Rest, etc.

2. Cómo crear un servicio

- Para crear un servicio usamos el comando "**generate service**“ (g s), indicando a continuación el nombre del servicio que queremos generar.

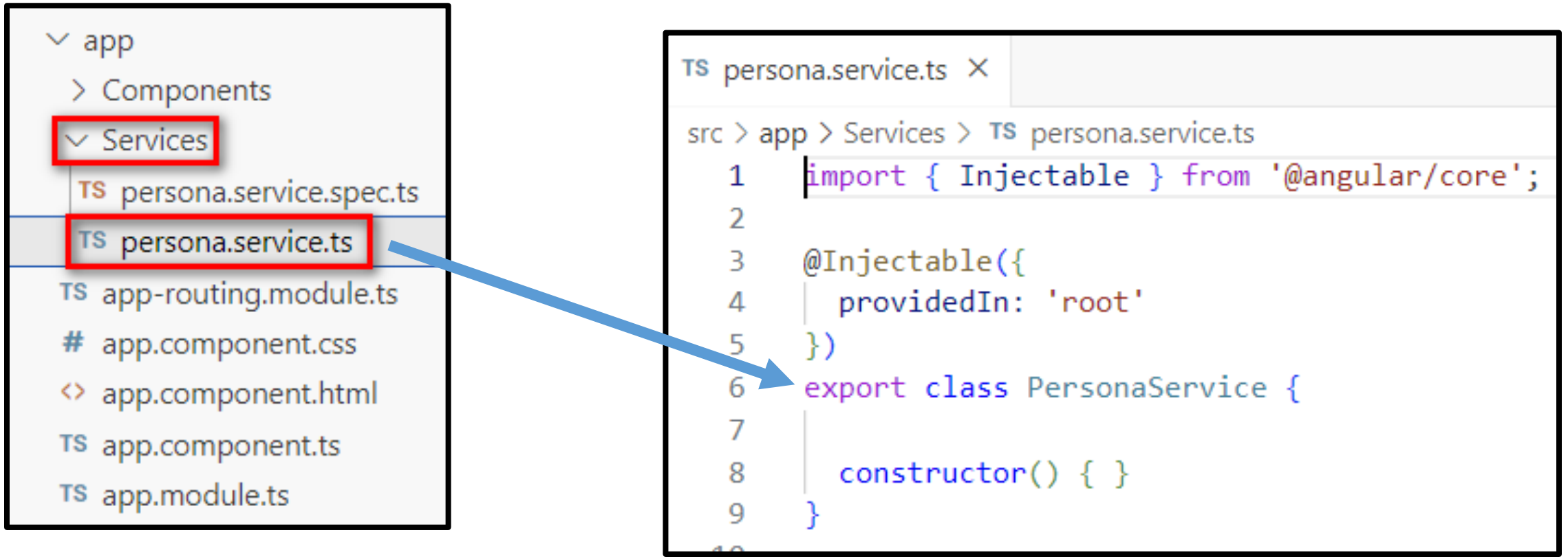
```
c:\angular\servicio>ng g s Services\persona
CREATE src/app/Services/persona.service.spec.ts (378 bytes)
CREATE src/app/Services/persona.service.ts (145 bytes)
```

3. Agregar la declaración del servicio a un módulo

- Para poder usar este servicio es necesario que lo agregues a un módulo.
- Inmediatamente lo podrás usar en cualquiera de los componentes que pertenecen a este módulo.

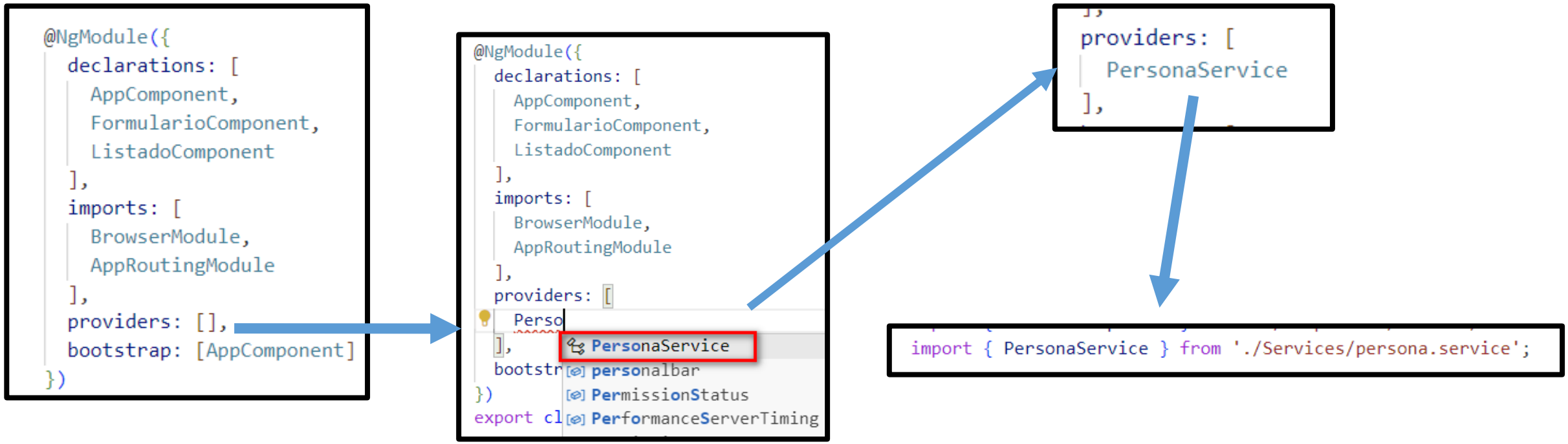
2. Cómo crear un servicio

- Esto nos genera el servicio llamado "*PersonaService*"



3. Agregar la declaración del servicio a un módulo

- Nos dirigimos a **app.module.ts** (nuestro módulo) y debemos añadir este servicio en los **providers.**



4. Código de un service

- El código de nuestro servicio es:

TS persona.service.ts

src > app > Services > TS persona.service.ts > ...

```
1   import { Injectable } from '@angular/core';
2
3   @Injectable({
4     providedIn: 'root'
5   })
6   export class PersonaService {
7
8     constructor() { }
9   }
10
```

4. Código de un service

- El decorador @injectable indica a Angular que la clase que se decora, en este caso la clase *PersonaService*, puede necesitar dependencias que puedan ser entregadas por inyección de dependencias.
- Los servicios necesitan de este decorador, aunque realmente disponer de él no es condición indispensable.

4. Código de un service

```
export class PersonaService {
  unaweb="https://www.google.es";
  constructor() { }
}
```

- En nuestra clase *PersonaService* hemos creado una propiedad llamada "*unaweb*", en la que hemos asignado un valor que sería común para todos los clientes.

4. Código de un service

- El dato es lo de menos, lo interesante es ver que la declaración no dista de otras declaraciones en clases que hayamos visto ya.

5. Cómo inyectar dependencias de servicios

- Ahora nos toca ver la magia de Angular y su inyección de dependencias, que vamos a usar para poder disponer del servicio en un componente.
- En Angular la inyección de dependencias se realiza por medio del constructor.
- En el constructor de un componente, que hasta ahora habíamos dejado siempre vacío, podemos declarar cualquiera de los servicios que vamos a usar y el framework se encargará de proporcionarlo, sin que tengamos que realizar nosotros ningún trabajo adicional.

5. Cómo inyectar dependencias de servicios

- Esto es tan sencillo como declarar como parámetro la dependencia en el constructor del componente.

5. Cómo inyectar dependencias de servicios

- De esta manera estamos indicando a TypeScript y Angular que vamos a usar un objeto *“ServicioPersona*" que es de la clase *“PersonaService*".
- A partir de entonces, dentro del componente existirá ese objeto, proporcionando todos los datos y funcionalidad definida en el servicio.

6. Usando el servicio en el componente

- Dentro de la clase de nuestro componente, tendremos el servicio a partir de la propiedad usada en su declaración. En el constructor dijimos que el servicio se llamaba “*PersonaServicio*“.
- Pues como cualquier otra propiedad, accederemos a ella mediante la variable "*this*".

6. Usando el servicio en el componente

```
export class AppComponent {
  constructor(public ServicioPersona:PersonaService) {
    console.log(this.ServicioPersona)
  }
}
```

```
▼ _PersonaService i
    unaweb: "https://www.google.es"
  ▶ [[Prototype]]: Object
```

6. Usando el servicio en el componente

app.component.html

src > app > app.component.html > p

Go to component

```
1   <p>
2       URL De acceso: {{clientesService.accesoFacturacion}}
3     </p>
```