

Angular

3.1. Interpolación con Angular

1. Interpolación con Angular

- Sirve para agregar texto dinámico en nuestras plantillas HTML.
- Ya lo hemos utilizado anteriormente, son valores definidos en el TypeScript del componente y utilizados en la plantilla HTML.

1. Interpolación con Angular

```
export class AppComponent {  
  title = 'app2';  
  mensaje='hola tronco/a voy a lo loco con Angular 17';  
}
```

<> app.component.html X

src > app > <> app.component.html > ...

Go to component

```
1 <h1>Hello, {{ title }}</h1>  
2 <p> {{ mensaje }}</p>  
3
```

1. Interpolación con Angular



2. Atributos públicos y privados

- Recordamos que estamos trabajando con TypeScript, por defecto los atributos son **public**, si lo ponemos como privados no podremos acceder a ellos directamente desde la plantilla.

```
export class AppComponent {  
  title = 'app2';  
  mensaje='hola tronco/a voy a lo loco con Angular 17';  
    
  private juan="Hola este mensaje es privado";  
}
```

2. Atributos públicos y privados

- Si intentamos acceder a ese atributo desde la plantilla, nos llevaremos una sorpresa.

```
<p>{{juan}}</p>
```

Property 'juan' is private and only accessible within class 'AppComponent'. ngts(2341)

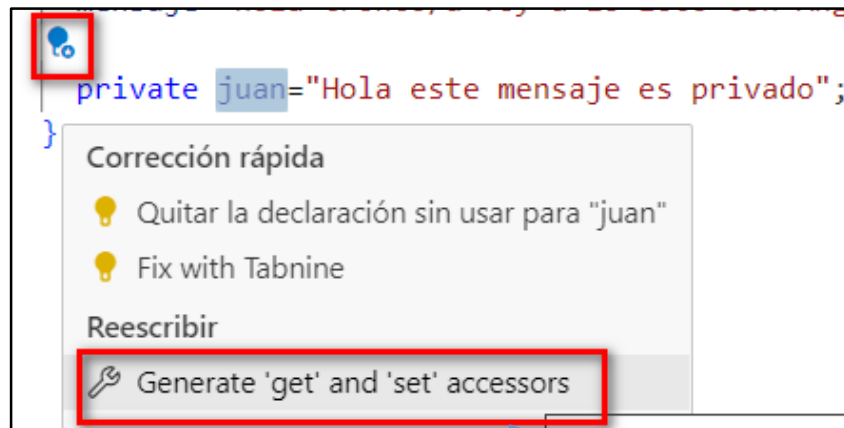
app.component.ts(4, 17): Error occurs in the template of component AppComponent.

```
private (property) AppComponent.juan: string
```

[Ver el problema \(Alt+F8\)](#) [Corrección Rápida \(Ctrl+.\)](#)

2. Atributos públicos y privados

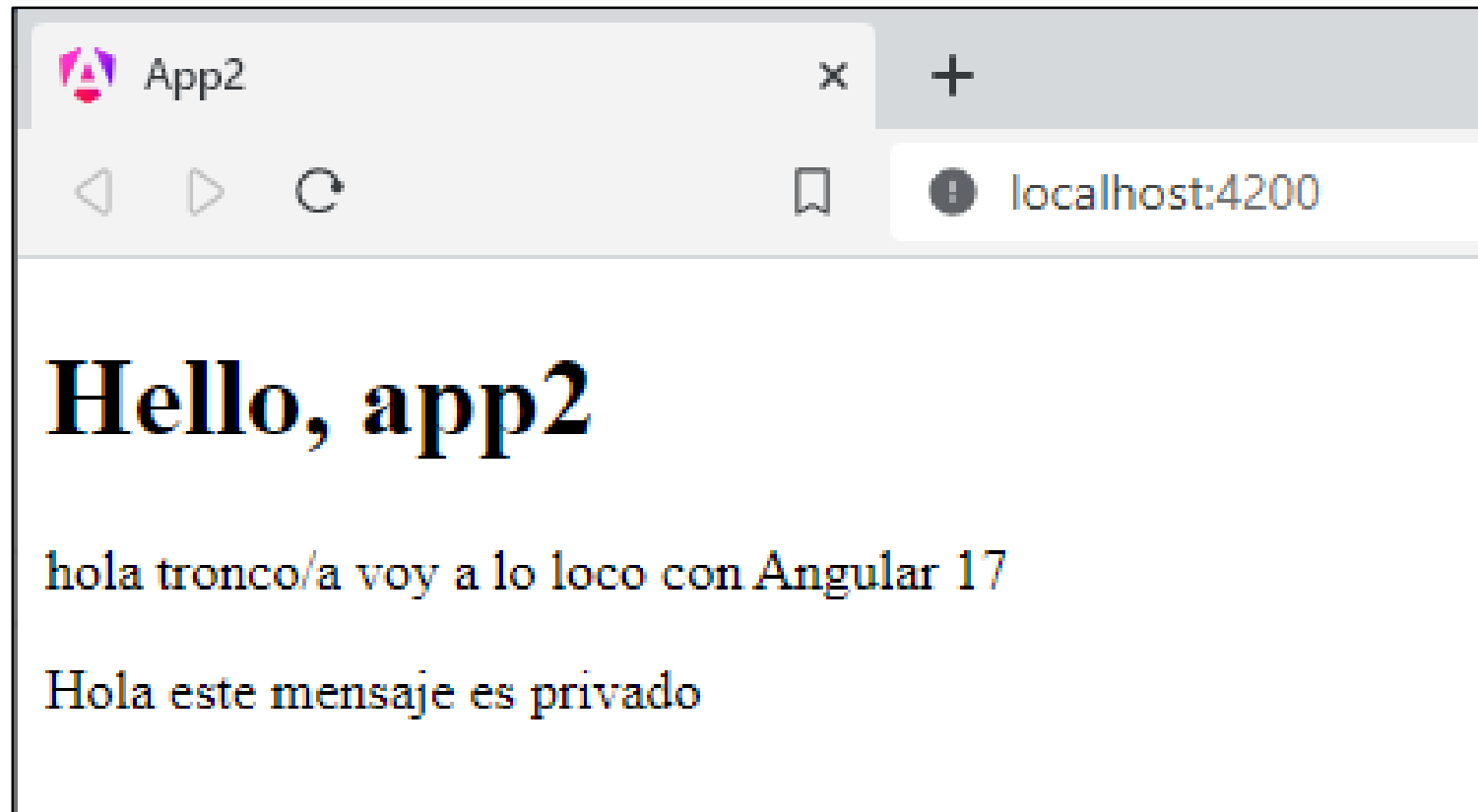
- Solución a esto: **Getters y Setters**



```
private _jua = "Hola este mensaje es privado";  
public get jua() {  
    return this._jua;  
}  
public set jua(value) {  
    this._jua = value;  
}
```

2. Atributos públicos y privados

- Ahora ya nos funcionaría la pagina:



3. Uso de métodos en las clases

- Podemos utilizar métodos también en las clases TypeScript, e interpolarlos en el código HTML.
- En fichero **app.component.ts** añadimos un método **muestraJuan()** que devuelve un **string** con el valor “muestro a Juan”

```
public muestraJuan(): string {  
    return "muestro a Juan";  
}
```

3. Uso de métodos en las clases

- Podemos interpolar desde la plantilla HTML (**app.component.html**) a este método (ya que es publico) y aparecerá en nuestra web.

Go to component

```
<h1>{{ title }} app is running!</h1>
<p> {{ mensaje }} </p>
<p> voy a mostrar a Juan: {{juan}} </p>

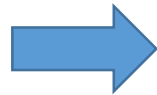
<p> ahora a muestra juan: {{muestraJuan()}} </p>
```



4. Expresiones

- Desde la plantilla HTML podemos utilizar expresiones, por ejemplo, con campos numéricos realizar operaciones aritméticas. Si son String realizará la concatenación.

```
export class AppComponent {  
  title = 'Primer ejemplo de Angular';  
  mensaje='Hola tronco voy a lo loco con Angular';  
  |  
  edad:number=20;  
}
```



```
Go to component  
<h1>{{ title }} app is running!</h1>  
<p> {{ mensaje }} </p>  
<h3> dentro de 5 años tendrás {{ edad+5 }} años</h3>
```

