

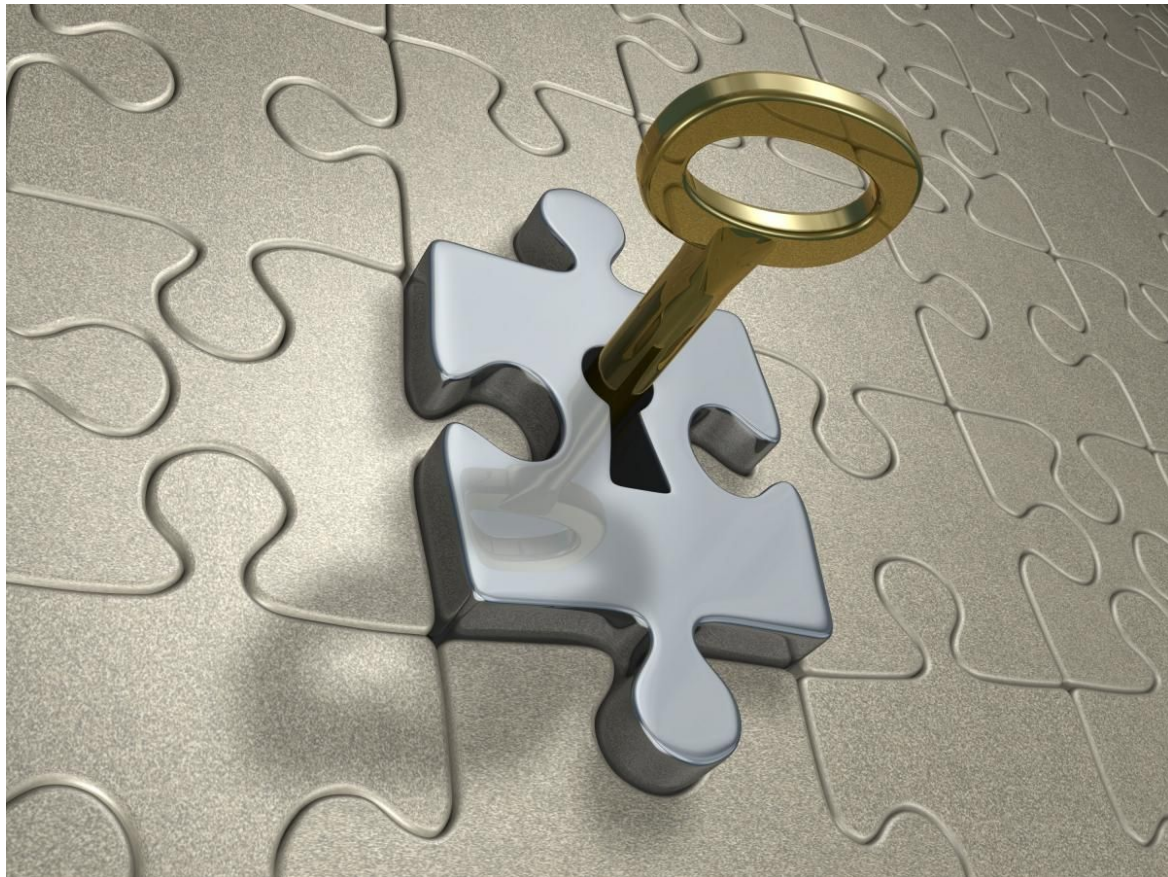


UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Lo que vimos la clase pasada

- ☐ Qué es Angular?
- ☐ Estructura de carpetas
- ☐ Qué son los componentes?
- ☐ Módulo raíz
- ☐ Vistas





Preguntas?



Lo que vamos a ver hoy!

- ☐ Componentes
- ☐ Servicios



Angular



Componentes

Componentes

Se puede pensar a un componente como un contenedor donde se soluciona una necesidad de la aplicación.

No sólo encapsulan un contenido sino también una funcionalidad.

Tienen relación padre/hijo y siempre habrá un componente raíz.

Componentes

Un componente está formado por 4 archivos:

- **CSS** para los estilos del componente
- **HTML** para el template
- **TypeScript** para el código
- Archivo de **testeo**

Podemos construir nuestros propios componentes para mantener más ordenado nuestro proyecto y conseguir que sea más fácil poder mantenerlo y reutilizar código.

Componentes

Para crear un componente

Desde la línea de comandos y posicionados en la carpeta del proyecto

```
$ ng generate component <nombre-componente>
```

Ejemplo: `ng g component calculadora`

Nota: generate = g

Componentes

Angular creará una nueva carpeta **src/app/nombre-componente** y dentro de ella los 4 archivos básicos.

```
D:\aprender\angular\my-app\src\app>ng g component calculadora
CREATE src/app/calculadora/calculadora.component.html (30 bytes)
CREATE src/app/calculadora/calculadora.component.spec.ts (663 bytes)
CREATE src/app/calculadora/calculadora.component.ts (289 bytes)
CREATE src/app/calculadora/calculadora.component.css (0 bytes)
UPDATE src/app/app.module.ts (540 bytes)
```

Componentes

Además también **agrega** el **nuevo componente** en el **@NgModule** del archivo **src/app/app.module.ts**

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

import { AppComponent } from './app.component';
import { PrimerComponenteComponent } from './primer-componente/primer-componente.component';
import { CalculadoraComponent } from './calculadora/calculadora.component';

@NgModule({
  declarations: [
    AppComponent,
    PrimerComponenteComponent,
    CalculadoraComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Componentes

Cómo usar un componente creado por nosotros

Los componentes pueden tener una relación padre/hijo. El padre controlará al hijo mediante el envío de información que el hijo necesite.

Componentes

Cómo usar un componente creado por nosotros

1. Crear el HTML para usar el componente

En el lugar de la aplicación que se vaya a usar el HTML

<nombre-componente></nombre-componente>

Angular

Componentes

Cómo usar un componente creado por nosotros

1. Crear el HTML para usar el componente



Lo que se pone acá tiene que ser igual a la definición del selector del archivo .ts del componente

```
app.component.css x calculadora.component.html x app.component.html x
1 <div style="text-align:center">
2   <calculadora></calculadora>
3 </div>
4 @Component({
   selector: 'calculadora',
   templateUrl: './calculadora.component.html',
   styleUrls: ['./calculadora.component.css']
})
```

Componentes

Cómo usar un componente creado por nosotros

2. Importar el código del componente

Desde el componente en el cual lo vamos a usar **agregar al ts**

```
import { nombre-componenteComponent } from './nombre-componente'
```

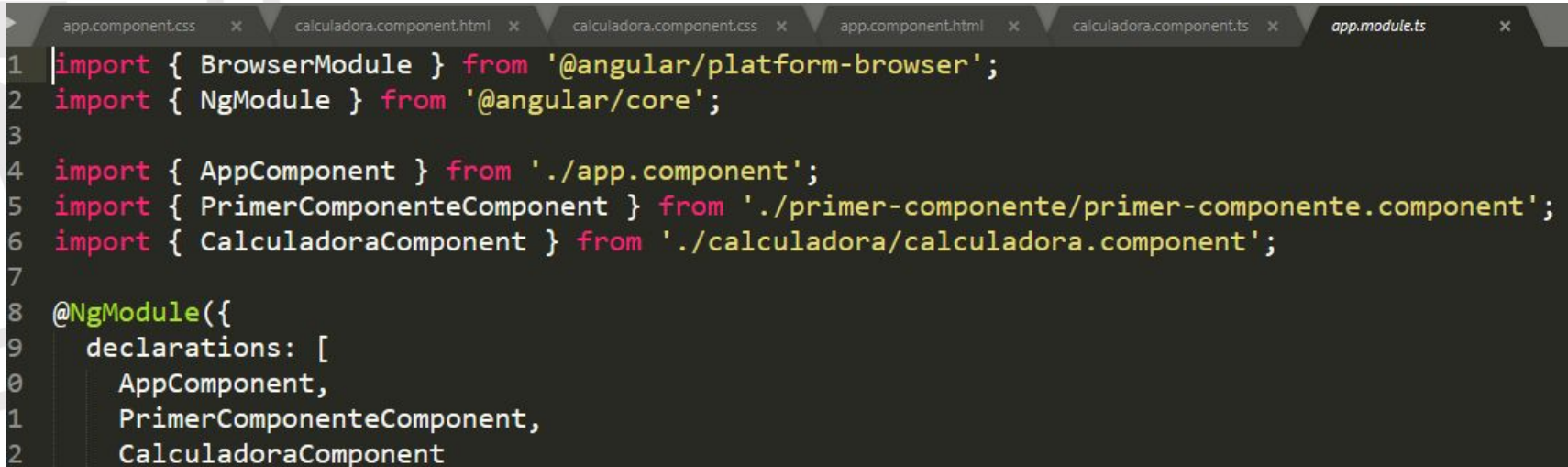
Este import permite conocer el código del componente pero todavía no es suficiente para poder usarlo.

Angular

Componentes

Cómo usar un componente creado por nosotros

2. Importar el código del componente



```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3
4 import { AppComponent } from './app.component';
5 import { PrimerComponenteComponent } from './primer-componente/primer-componente.component';
6 import { CalculadoraComponent } from './calculadora/calculadora.component';
7
8 @NgModule({
9   declarations: [
10     AppComponent,
11     PrimerComponenteComponent,
12     CalculadoraComponent
```


Componentes

Resumiendo...

- Se genera el componente, esto genera 4 archivos en una carpeta con el nombre del componente.
- Se realizan las modificaciones necesarias en el HTML, en el CSS y en el TS
- Se agrega el componente al app.module.ts
- Se importa el componente en el lugar donde se lo va a usar.



Servicios

Servicios

Los servicios son una forma de compartir información entre clases que no se conocen.

Cuando estamos creando nuestros componentes muchas veces debemos compartir información entre ellos, este es el caso del uso de los Servicios.

En las buenas prácticas de Angular se recomienda liberar a los componentes de cualquier lógica no relacionada con la vista. Hay que mover toda esa lógica a un servicio.

Servicios

Para crear un servicio

```
$ng g service nombre-servicio
```

Genera el esqueleto de la clase nombre-servicioService en **src/app/nombre-servicio.service.ts**.

Al consumo de los servicios inyectables se le conoce como DEPENDENCIA

Servicios

Cómo usar un Servicio?

Angular funciona a través de inyección de dependencias. Esto significa que se puede pasar una referencia a una instancia en diferentes componentes y permitirá utilizarla en diferentes partes de la App.

Servicios

Resumiendo...

- Se genera el servicio
- Se agrega al app.module.ts en 2 lugares:
 - se importan en la parte de import
 - se agrega al providers

Servicios

Cómo usar un Servicio?

Angular funciona a través de inyección de dependencias. Esto significa que se puede pasar una referencia a una instancia en diferentes componentes y permitirá utilizarla en diferentes partes de la App.

PRÁCTICA

PRÁCTICA ANGULAR

Ejercicio 1

Agregale a la calculadora las otras funciones disponibles (restar, multiplicar, dividir) y el resultado mostralo en otra página mediante servicios

FIN!!!!

