

# Angular e Angular JS

Trabalho sobre Frameworks - DW2A4  
Lucas Hideki Kubo SP3046672





# Agenda

História da tecnologia

Versões

Características

Como utilizar





# Visão Geral, Contexto e Cronologia

- Antes do Angular, como era o desenvolvimento de aplicações web?
- Qual a primeira versão lançada do Angular? O que fazia?
- Quais os desafios que os desenvolvedores enfrentavam e porque eles justificavam a necessidade de se criar um conjunto de regras de negócio, bibliotecas, arquiteturas, metodologias e condensá-los em um único “Framework”?



# Versionamento do AngularJS

- AngularJS - A primeira versão de todas, direcionada a superar os desafios de desenvolver Single Page Applications (SPA).
- Lançado em 2009 por Hevery e Abrons como licença comercial
- Com baixa adesão da comunidade, logo decidiram torná-lo open-source. Hevery trabalhava na Google e continuou dentro da empresa com outros colegas a desenvolver o framework.
- Atualizações e manutenção conturbadas colocaram em cheque o futuro do projeto até 2016, com o lançamento do Angular2+, ou simplesmente “Angular”.
- No dia 31 de dezembro de 2021 será feita a última atualização de patch do AngularJS, e ele estará então, descontinuado.



# Versionamento do Angular

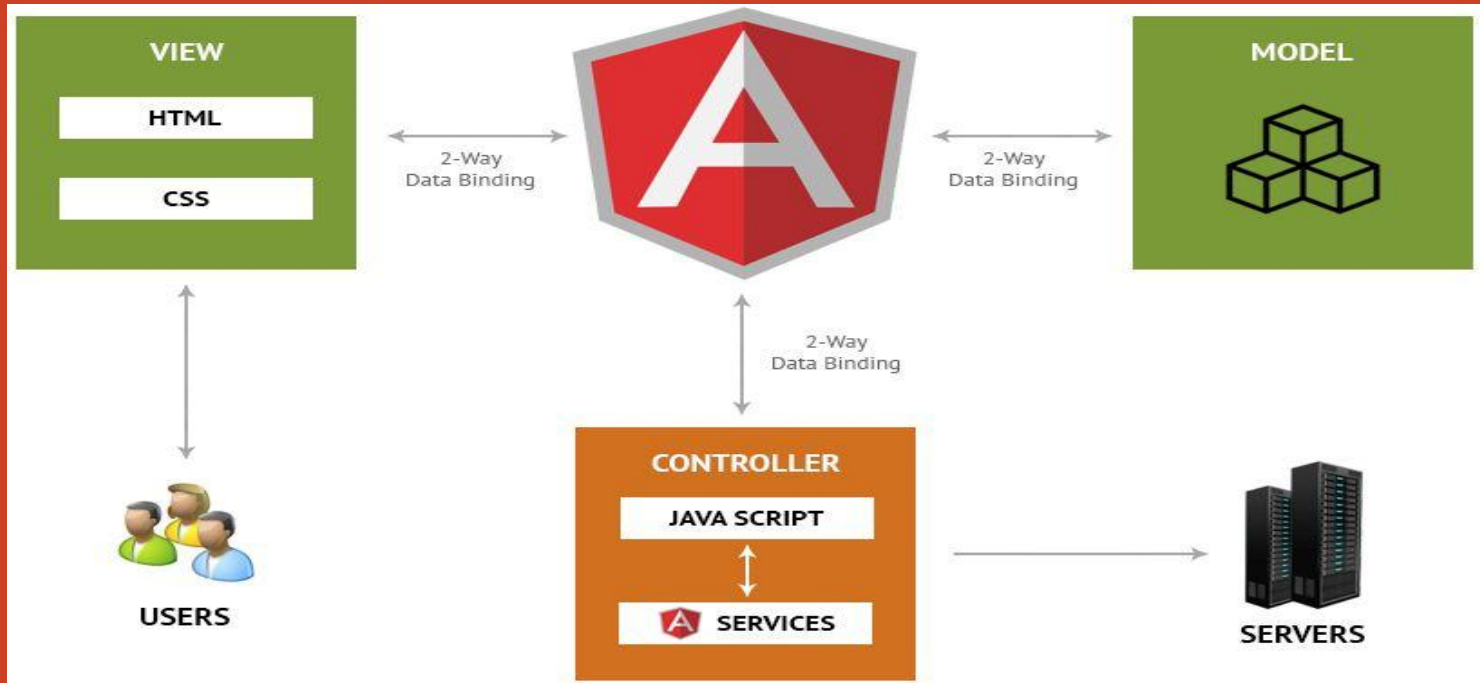
- Angular ou Angularv2 ou Angular2+ - Reescrita do Angular para a linguagem da Microsoft Typescript.
- Lançado sua primeira versão Angular 2.0 em 2016.
- Atualizado semestralmente pela Google, é um framework bem aceito na comunidade como um dos principais frameworks de desenvolvimento web do mercado.
- Atualmente se encontra na versão Angular 12 lançada em 12 de maio de 2021
- Diferente do Angular JS ele suporta as versões mais recentes de ferramentas web como HTML5, CSS3 e ECMAScript 6.



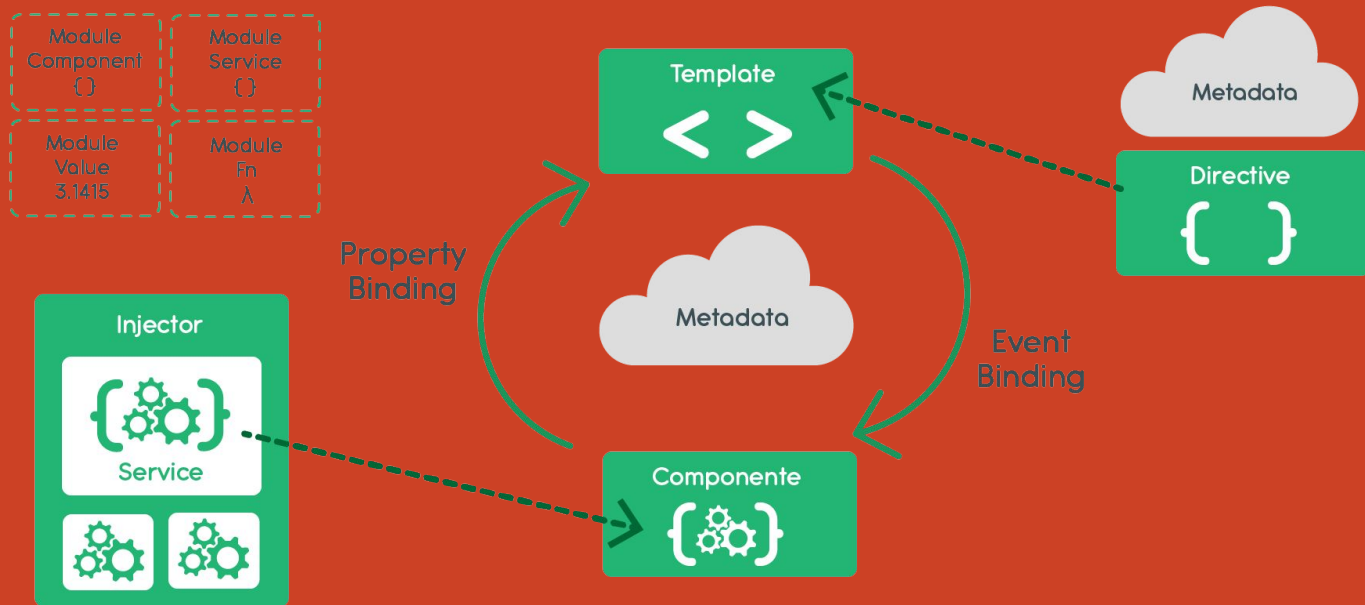
# Principais características Angular/AngularJS

- Arquitetura
- Linguagem/Sintaxe
- Suporte
- Performance

# Arquitetura - AngularJS - MVC



# Arquitetura - Angular - Componentes







## Linguagem/Sintaxe - AngularJS

- AngularJS como o nome diz, usa Javascript! =)
- É uma linguagem multiparadigma com tipagem dinâmica, o que pode ser difícil de lidar em aplicações de grande escala
- A sintaxe do AngularJS se utiliza de diretivas próprias para criar projetos, os comandos -ng.
- Com o conjunto de diretivas o AngularJS é capaz de criar e manter aplicações web.



## Linguagem/Sintaxe - Angular

- Angular usa a linguagem da Microsoft: Typescript.
- É um superset do Javascript que implementa tudo que o Javascript ES6 possui, e ainda mais funcionalidades.
- É orientada a objetos, permitindo encapsulamento de código-fonte, polimorfismo, etc,
- Possui tipagem estática, ou seja, é fortemente tipada.
- Estes fatores contribuem para uma escalabilidade maior das aplicações.



## Suporte - Angular/Angular JS

- O AngularJS não oferece suporte para as versões mais recentes de ferramentas de desenvolvimento web!!
- Também não fornece suporte para browsers mobile.
- O Angular por sua vez permite compatibilidade com as ferramentas mais recentes como HTML5, CSS3 e ES6.
- O Angular foi pensado com o ambiente mobile fazendo parte de seu escopo, portanto ele oferece suporte total a browsers web.



## Performance - AngularJS

- Possui arquitetura MVC, Model-View-Controller, considerada centralizada, com um modelo central e diversas visões satélites.
- Faz ligações bidirecionais entre seus módulos, o que pode prejudicar a escalabilidade de um sistema AngularJS.
- Apesar disso, estas práticas diminuem o esforço e o tempo necessário para realizá-lo, aliviando a equipe de desenvolvimento.



# Performance - Angular

- Maior modularização, com arquitetura de componentes associada à orientação a objetos e tipagem estática do Typescript.
- Injeção de dependência, permite prover as condições necessárias para que uma classe seja instanciada e utilizada.
- Serviços encapsulam regras de negócio, enquanto os componentes focam em manter e atualizar a UI.
- Menor acoplamento da aplicação.
- Maior performance e escalabilidade.



# Como utilizar

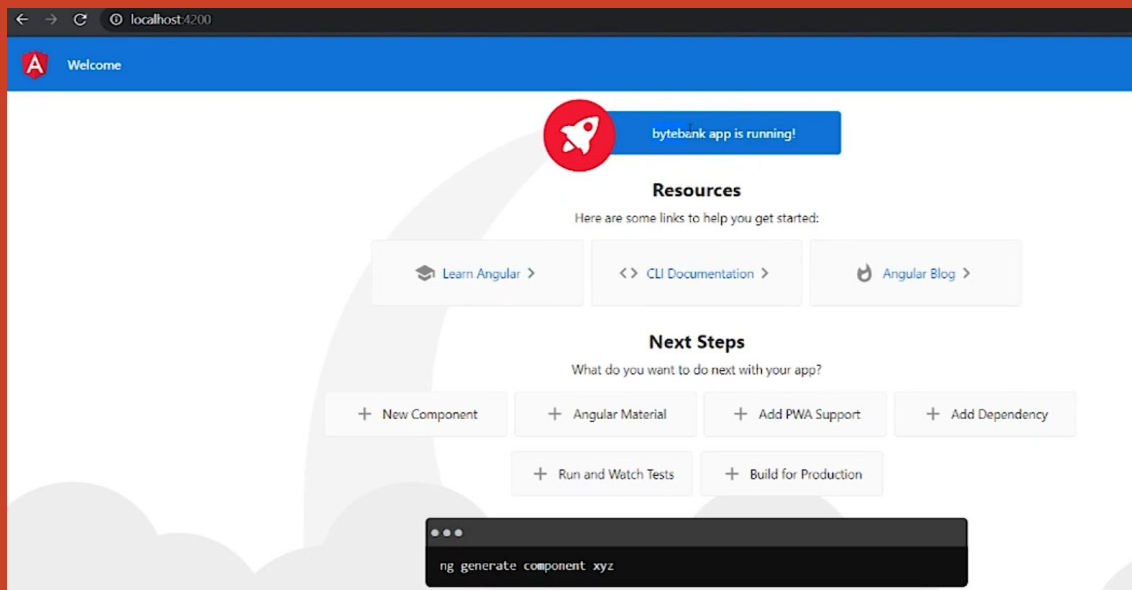
Instalando o Angular CLI e criando aplicação

```
> npm install -g @angular/cli  
  
> ng new my-dream-app  
  
> cd my-dream-app  
  
> ng serve
```

# Como utilizar

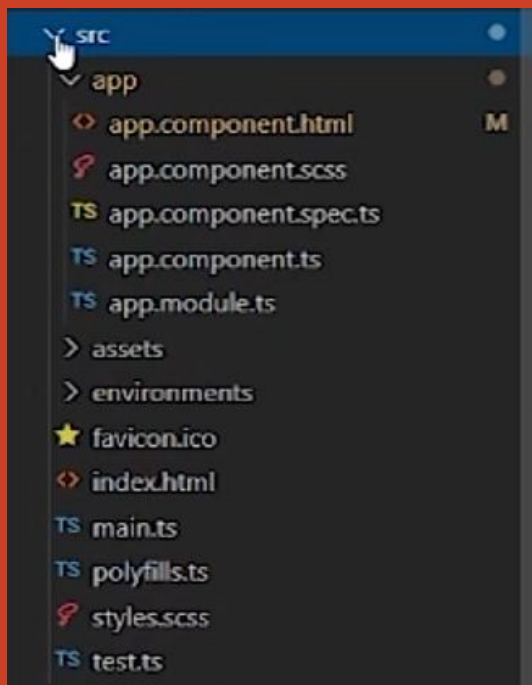
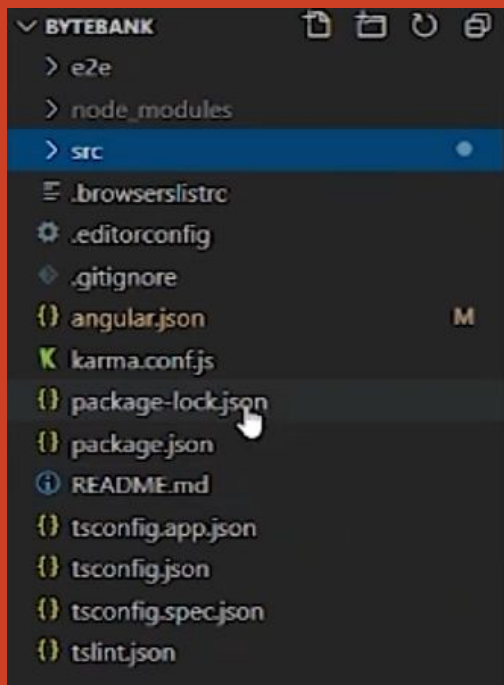
## Rodando a aplicação

```
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **  
: compiled successfully.
```



# Como utilizar

## Organização





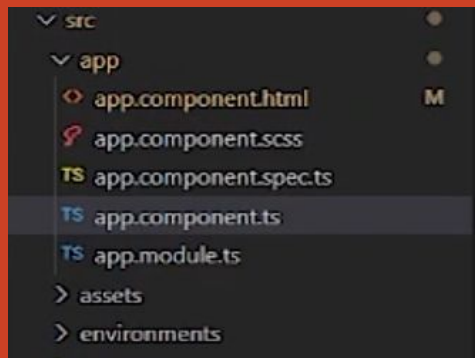
# Como utilizar

## Index

```
index.html X
src > index.html > html > body > app-root
1  <!doctype html>
2  <html lang="en">
3  <head>
4    <meta charset="utf-8">
5    <title>Bytebank</title>
6    <base href="/">
7    <meta name="viewport" content="width=device-width, initial-sca
8    <link rel="icon" type="image/x-icon" href="favicon.ico">
9  </head>
10 <body>
11   <app-root></app-root>
12 </body>
13 </html>
14
```

# Como utilizar

app.component.ts

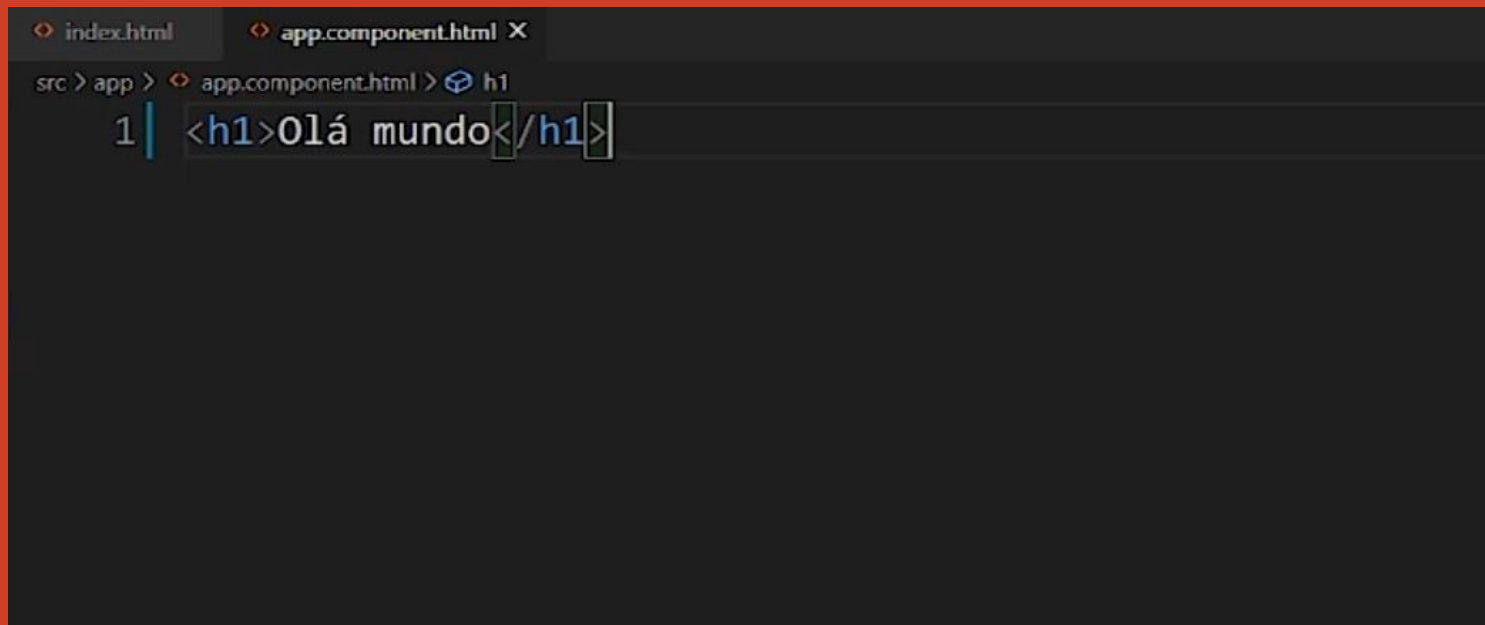


src > app > TS app.component.ts

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   templateUrl: './app.component.html',
6   styleUrls: ['./app.component.scss']
7 })
8 export class AppComponent {
9   title = 'bytebank';
10 }
11
```

# Como utilizar

app.component.html

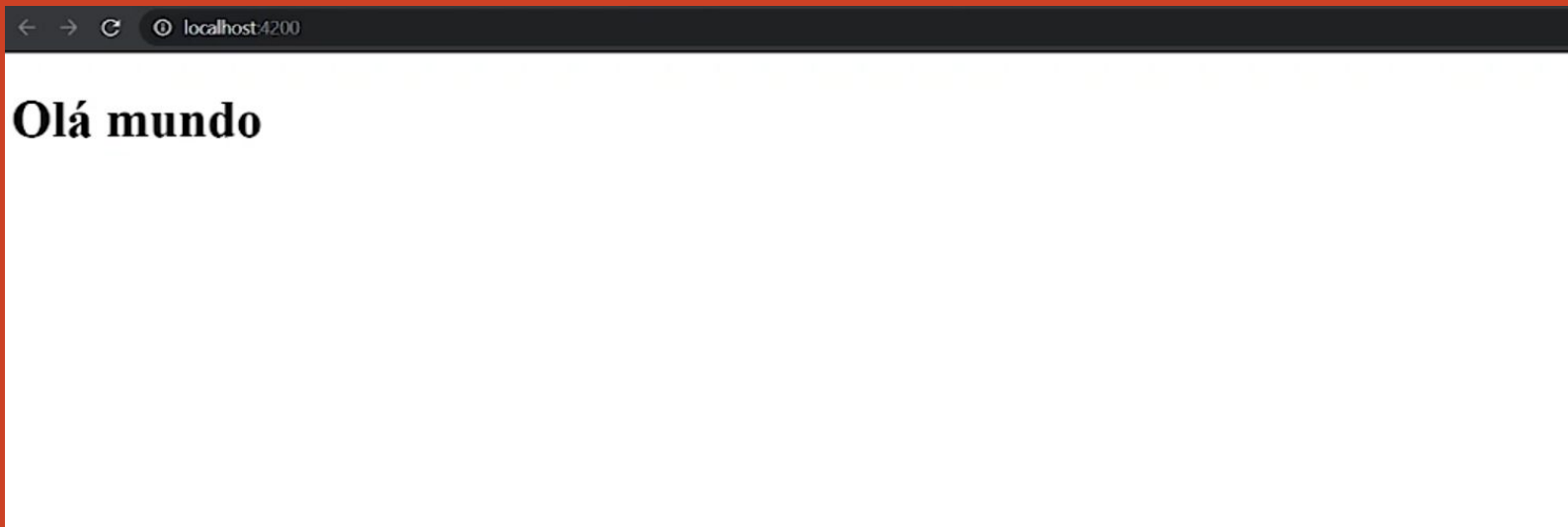


```
index.html  app.component.html X
src > app > app.component.html > h1
1 | <h1>Olá mundo</h1>
```



# Como utilizar

app.component.html



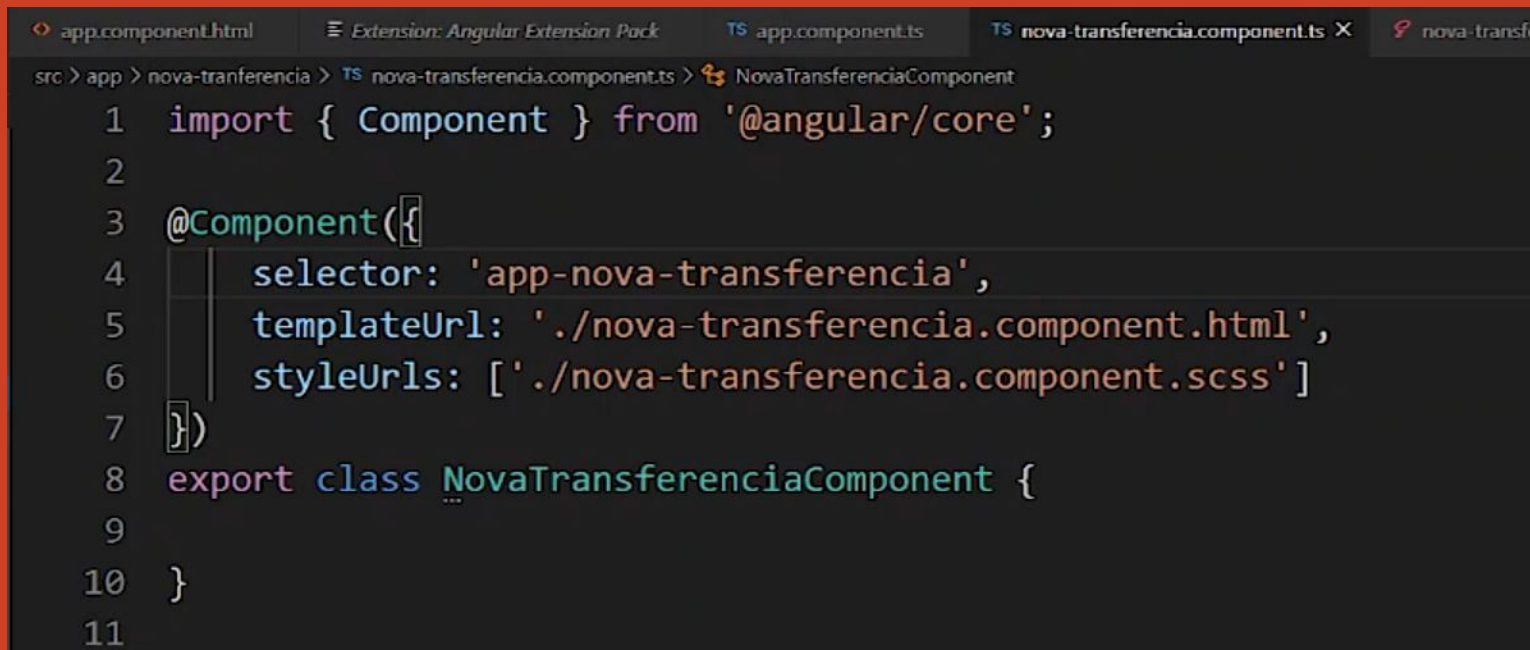
# Como utilizar

## Criando component - criando o html

```
index.html  app.component.html  nova-transferencia.component.html
src > app > nova-transferencia > nova-transferencia.component.html > form
1  <form>
2    <h2>Nova Transferência</h2>
3
4    <div>
5      <label for="valor">Valor</label>
6      <input id="valor" type="text">
7    </div>
8
9    <div>
10     <label for="destino">Destino</label>
11     <input id="destino" type="text">
12   </div>
13
14   <button>Transferir</button>
15 </form>
```

# Como utilizar

Criando component

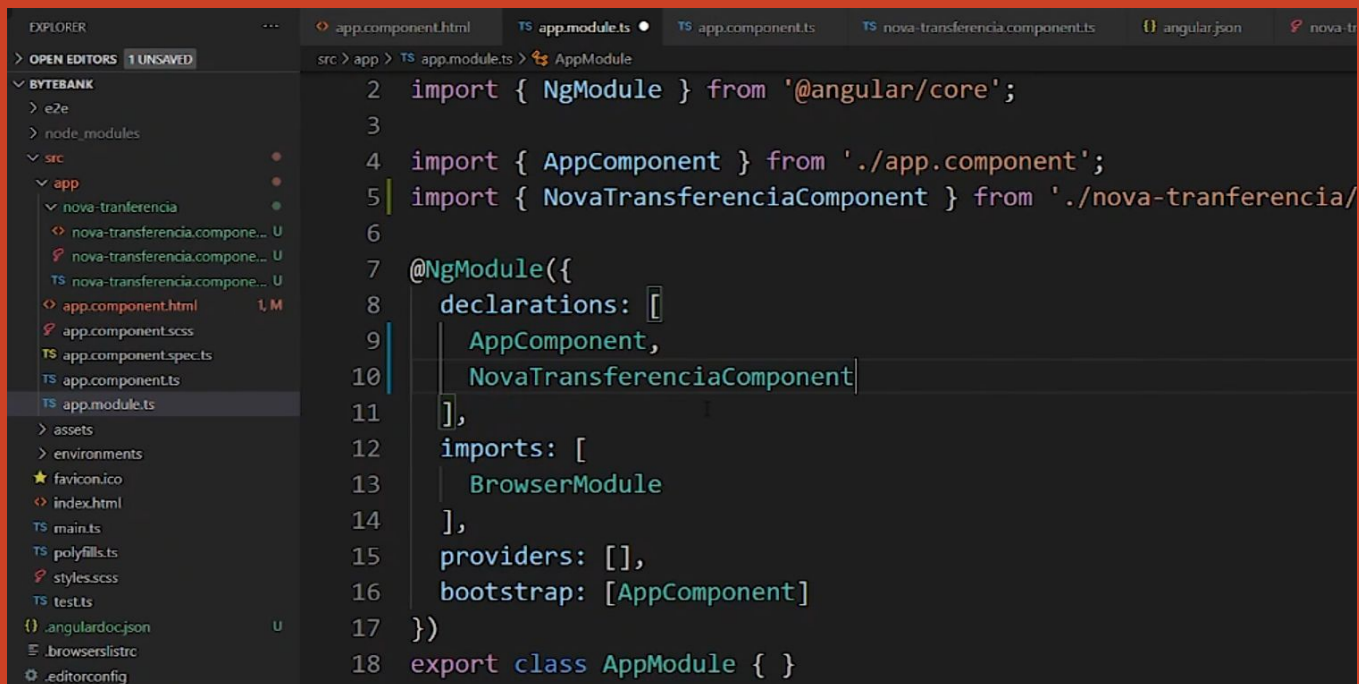


The screenshot shows the Visual Studio Code editor interface. The top bar displays the file explorer with the following structure: `src > app > nova-transferencia > TS nova-transferencia.component.ts`. The active file is `nova-transferencia.component.ts`, which contains the following TypeScript code:

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-nova-transferencia',
5   templateUrl: './nova-transferencia.component.html',
6   styleUrls: ['./nova-transferencia.component.scss']
7 })
8 export class NovaTransferenciaComponent {
9
10 }
11
```

# Como utilizar

Criando component - adicionando no app.module.ts

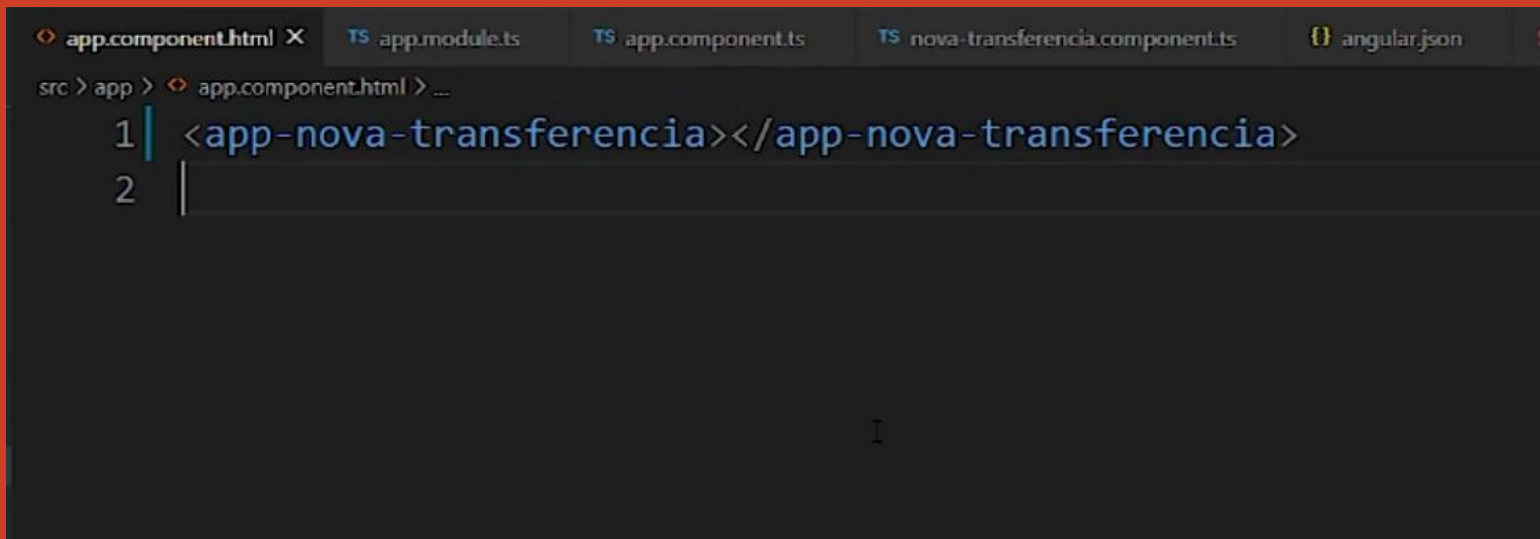


The screenshot shows a code editor with the Explorer sidebar on the left and the main editor on the right. The Explorer sidebar shows a project structure with a folder named 'app' containing a subfolder 'nova-transferencia'. The file 'app.module.ts' is selected. The main editor shows the code for 'app.module.ts' with the following content:

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

# Como utilizar

Adicionando no app.component.html



```
app.component.html X TS app.module.ts TS app.component.ts TS nova-transferencia.component.ts angular.json
src > app > app.component.html > ...
1 | <app-nova-transferencia></app-nova-transferencia>
2 |
```



# Como utilizar



A screenshot of a web browser window displaying a form titled "Nova Transferência". The browser's address bar shows "localhost:4200". The form contains two input fields: "Valor" and "Destino", both of which are empty. Below these fields is a button labeled "Transferir". A mouse cursor is positioned over the "Valor" input field.

← → ↻ ⓘ localhost:4200

## Nova Transferência

Valor

Destino

Transferir