

Tech Up ***CHALLENGE***

Introduzindo Angular

Fluxo de Dados no Angular

O fluxo de dados no Angular é fundamental para o desenvolvimento de aplicativos web dinâmicos e interativos. O Angular segue o padrão de arquitetura MVVM (Model-View-ViewModel) **para gerenciar o fluxo de dados e a interação entre os componentes do aplicativo.**



Fluxo de Dados no Angular

Componentes: Os componentes são os blocos de construção fundamentais de um aplicativo Angular. Cada componente tem sua própria lógica e uma parte da interface do usuário. **Os componentes podem se comunicar entre si por meio do fluxo e vinculação de dados.**



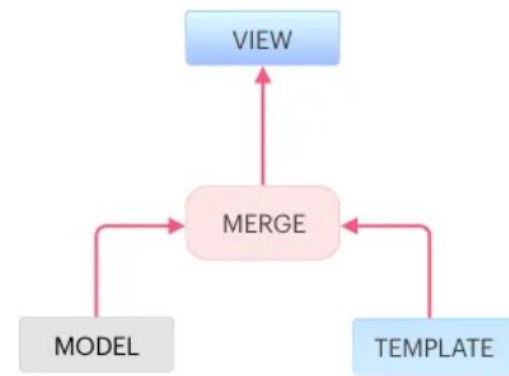
Fluxo de Dados no Angular

Data Binding (Vinculação de Dados): O Angular fornece várias formas de vincular dados entre a View e o ViewModel. Os tipos de vinculação de dados incluem vinculação unidirecional (**one-way binding**) e vinculação bidirecional (**two-way binding**).



Fluxo de Dados no Angular

Vinculação Unidirecional (One-Way Binding): Isso permite que os dados fluam da ViewModel para a View ou vice-versa, mas não ambos ao mesmo tempo. Isso é alcançado usando interpolação (`{{}}`), propriedades de diretivas ou a diretiva `ng-bind`.



One-Way Data Binding

Exemplo de Código One-Way Data Binding

The image shows a development environment with two main panels: a code editor on the left and a browser preview on the right.

Code Editor (Left Panel):

- app.component.ts:** Contains the following code:

```
1 import { Component } from '@angular/core';
2
3 You, 2 minutes ago | 1 author (You)
4 @Component({
5   selector: 'app-root',
6   templateUrl: './app.component.html',
7   styleUrls: ['./app.component.scss']
8 })
9
10 export class AppComponent {
11   title = 'Aula01';
12 }
```
- app.component.html:** Contains the following code:

```
1 <section class="fundo">
2   <h1>{{ title }}</h1>
3 </section>
```

Browser Preview (Right Panel):

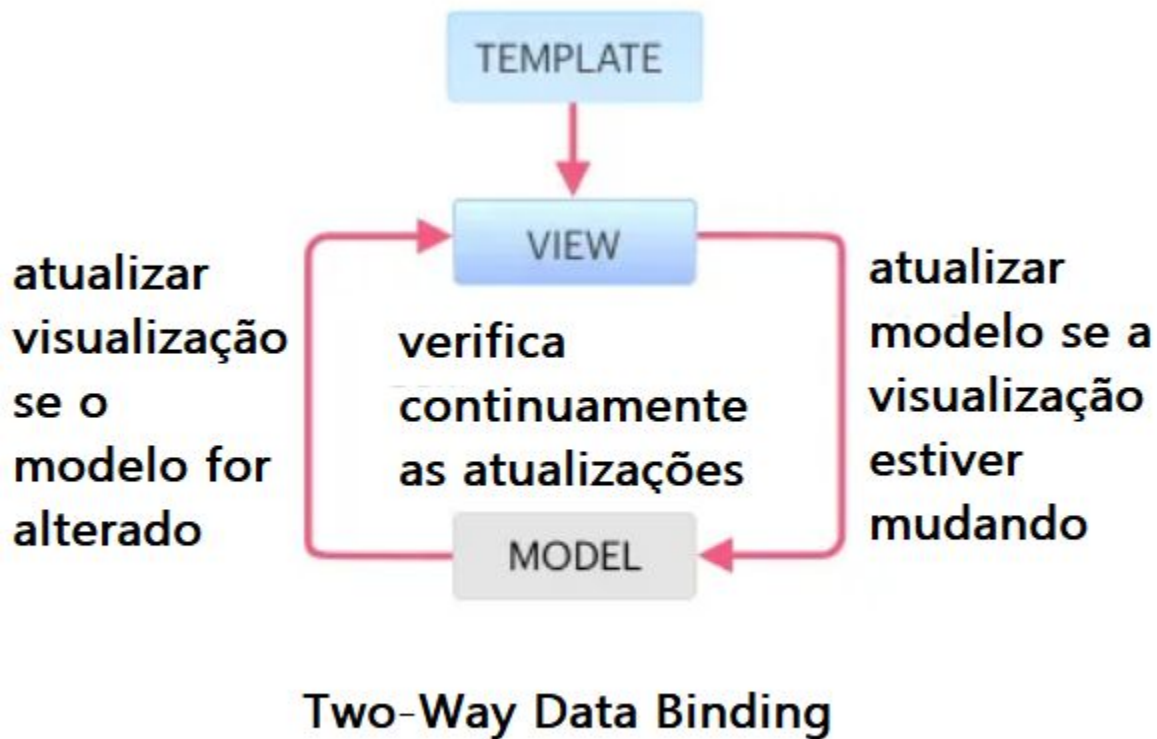
- The browser address bar shows `localhost:4200`.
- The page content displays `Aula01`, which is the value of the `title` property from the TypeScript file.

Diagrammatic Elements:

- A red box highlights the `title = 'Aula01';` line in the TypeScript file.
- A red box highlights the `<h1>{{ title }}</h1>` line in the HTML file.
- A red arrow points from the highlighted line in the TypeScript file to the highlighted line in the HTML file, illustrating the data binding.
- Another red arrow points from the highlighted line in the HTML file to the `Aula01` text in the browser preview, showing the rendered output.

Fluxo de Dados no Angular

Vinculação Bidirecional (Two-Way Binding): Isso permite que os dados fluam tanto da ViewModel para a View quanto da View para a ViewModel. É frequentemente usado com elementos de entrada de formulário, como caixas de texto, usando a diretiva ngModel.



Exemplo de Código Two-Way Data Binding

The image shows a development environment with VS Code and a web browser. In VS Code, the file `img-crateres.component.ts` contains the following TypeScript code:

```
14 dados: any;  
15 personagens = personagensData;  
16 selectedPersonagem: string = '';  
17  
18 constructor(private dataService: DataService) {}  
19 ngOnInit(): void {  
20   this.dataService.getData().subscribe((data) => {  
21     this.dados = data;  
22   });  
23 }  
24  
25 //esta opção é para trabalhar com two-way data binding  
26 getImageUrl(): string {  
27   const personagem =  
28   this.personagens.find(foto => foto.nome === this.selectedPersonagem);  
29   return personagem ? personagem.imageUrl : '';  
30 }  
31  
32
```

The file `img-crateres.component.html` contains the following HTML code:

```
19 </div>  
20  
21 <div class="coluna">  
22   <h3>{{ titled }}</h3>  
23   <!-- select para selecionar o nome do personagem -->  
24   <div class="envelope-img">  
25     <div class="linha">  
26       <!--two-way db-->  
27       <select [(ngModel)]="selectedPersonagem" class="form-control">  
28         <option value="">Clique para selecionar um personagem</option>  
29         <option  
30           *ngFor="let personagem of personagens"  
31           [value]="personagem.nome"  
32         >  
33           {{ personagem.nome }}  
34         </option>  
35       </select>  
36     </div>  
37     <div class="linha">  
38       <!-- Usando ngIf para exibir a imagem se o nome do personagem for selecionado -->  
39       <img  
40         *ngIf="getImageUrl()"  
41         [src]="getImageUrl()"  
42         alt="Imagem do personagem"
```

The web browser shows the application running. It has a heading "Selecione um personagem para ver a sua imagem" and a dropdown menu with "Lucy" selected. Below the dropdown is an image of a person in a costume. Red arrows indicate the flow of data: one from the `selectedPersonagem` property in the TypeScript code to the `[(ngModel)]` attribute in the HTML, and another from the selected value in the dropdown to the `ngIf` and `src` attributes in the image HTML element.

Tech Up ***CHALLENGE***