



Formulários Template-driven e reativos



Formulários

- Lidam com a entrada do usuário e são a base de muitos aplicativos
- O angular provê duas diferentes abordagens para utilização de formulários
 - Template-driven
 - Reactive
- Ambos capturam eventos de entrada do usuário a partir da view, validam a mesma, criam um form model e um data model e fornecem uma maneira de detectar mudanças



Template Driven vs Reactive Forms

Template-driven	Reactive (Data-driven)
Fácil de usar	Mais flexível, mas requer mais prática
Cenários simples	Adequado para todos os cenários
Two-way data binding	Não utiliza binding
Formulário é criado e configurado no HTML, ou seja, pouco código no Component	Formulário é criado e configurado no Component, ou seja, menos código HTML
Detecção automática do form e de seus dados (pelo Angular)	Transformações reativas especificadas pelo programador
Validações são feitas no template HTML	Validações são feitas no Component
Valores do form são submetidos com <code>ngSubmit</code>	Form já está no Component e não precisa do <code>ngSubmit</code>



Criando um formulário template-driven

- Importar o FormsModule
- Adicionar controles
 - ngForm
 - ngSubmit
 - ngModel
- Caso necessário, utilizar o two-way data binding ([(ngModel)])



Validação no formulário template-driven

- Referência: <https://angular.io/api/forms/Validators>
- Validators:
 - Required
 - Email
 - Min
 - Max
- Utilizar classes adicionadas pelo Angular (Form State)
- HTML5 validation:
<https://www.the-art-of-web.com/html/html5-form-validation/>



Estado dos campos do Form

Estado	SIM	NÃO
Controle visitado	ng-touched	ng-untouched
Valor mudou	ng-dirty	ng-pristine
Controle válido	ng-valid	ng-invalid



Criando um formulário reativo

- Importar o módulo de forms reativos (*ReactiveFormsModule*)
- Criar uma nova instância FormControl no componente
- Registrar o FormControl no template



Registrando o módulo

```
import { ReactiveFormsModule } from '@angular/forms';

@NgModule({
  imports: [
    // other imports ...
    ReactiveFormsModule
  ],
})
export class AppModule { }
```




Adicionando um FormControl no componente

```
import { Component } from '@angular/core';
import { FormControl } from '@angular/forms';

@Component ({
  selector: 'app-name-editor',
  templateUrl:
    './name-editor.component.html',
  styleUrls: ['./name-editor.component.css']
})
export class NameEditorComponent {
  name = new FormControl ('');
}
```



Registrar o control no template

```
<label>  
  Name:  
  <input type="text" [formControl]="name">  
</label>
```



Demo: FormControl, FormGroup e Validation



Observações:

- Questionário: <https://forms.gle/TNvJQHnhjwStcyvh6>
- Código fonte: <https://github.com/GabriellBP/Angular-course>
- Dúvidas:
 - gabriel.pereira@edge.ufal.br
 - lucas.amorim@edge.ufal.br



Obrigado!