


Formulários

Durante o desenvolvimento web, é claro que sempre vamos ter que trabalhar com formulários, seja para fazer uma simples pesquisa no seu site ou para fazer cadastro de um usuário ou produto. Pensando nisso, o Angular desenvolveu seus próprios métodos para trabalhar com formulários. O Angular nos oferece duas maneiras de trabalhar com formulários: A primeira delas é trabalhar com **Template Driven Forms** e com formulários reativos (**Reactive Forms**). Vamos ver agora como se trabalha com o Template Driven Forms

Template Driven Forms

Para iniciarmos os trabalhos com o Template Driven Forms, nós devemos importar o módulo **FormsModule** no módulo principal.



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

import { AppComponent } from './app.component';
import { FormsModule } from '@angular/forms';

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

Agora sim, nossa aplicação está apta para trabalhar com o Template Driven Forms. Para iniciar, vamos fazer um formulário simples para testarmos.

```
<form>
  <label for="nome">Nome</label>
  <input type="text" name="nome" id="nome"> <br><br>

  <label for="email">E-mail</label>
  <input type="email" name="email" id="email"> <br><br>

  <label for="senha">Senha</label>
  <input type="password" name="senha" id="senha"> <br><br>
</form>
```

Vamos manter assim, por enquanto. Em breve faremos algumas alterações. Vamos criar propriedades no nosso componente que usaremos dentro do nosso formulário.

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

@Component({
  selector: 'my-app',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  nome: string = ''
  email: string = ''
  senha: string = ''
}
```

Agora vamos ligar essas propriedades aos nossos inputs pelo conceito do Two Way Data Binding, visto anteriormente.

```

<form>
  <label for="nome">Nome</label>
  <input
    type="text"
    name="nome"
    id="nome"
    [(ngModel)]="nome"
  > <br><br>

  <label for="email">E-mail</label>
  <input
    type="email"
    name="email"
    id="email"
    [(ngModel)]="email"
  > <br><br>

  <label for="senha">Nome</label>
  <input
    type="password"
    name="senha"
    id="senha"
    [(ngModel)]="senha"
  > <br><br>
</form>

```

Agora nós devemos informar que esse formulário é controlado pelo Template Driven Form. Para isso, devemos criar uma template variable dentro da tag form e informar que essa variável vai receber um valor especial. Ela vai receber um valor chamado **ngForm**.

```

<form #meuForm="ngForm">
  <label for="nome">Nome</label>
  <input
    type="text"
    name="nome"
    id="nome"
    [(ngModel)]="nome"
  > <br><br>

  <label for="email">E-mail</label>
  <input
    type="email"
    name="email"
    id="email"
    [(ngModel)]="email"
  > <br><br>

  <label for="senha">Nome</label>
  <input
    type="password"
    name="senha"
    id="senha"
    [(ngModel)]="senha"
  > <br><br>
</form>

```

Além disso, quando trabalhamos com formulários dentro do Angular, ele nos fornece um evento especial para ouvir o envio dos formulários. Esse evento é o **ngSubmit**. Vamos implementá-lo e deixá-lo pronto.

```
<form #meuForm="ngForm" (ngSubmit)="doSomething()">
  <label for="nome">Nome</label>
  <input
    type="text"
    name="nome"
    id="nome"
    [(ngModel)]="nome"
  > <br><br>

  <label for="email">E-mail</label>
  <input
    type="email"
    name="email"
    id="email"
    [(ngModel)]="email"
  > <br><br>

  <label for="senha">Nome</label>
  <input
    type="password"
    name="senha"
    id="senha"
    [(ngModel)]="senha"
  > <br><br>
</form>
```

Agora vamos, em cada input do nosso formulário, colocar atributos de verificação padrão do Javascript (minlength, min, max, required, etc...).

```
<form #meuForm="ngForm" (ngSubmit)="doSomething()">
  <label for="nome">Nome </label>
  <input
    type="text"
    name="nome"
    id="nome"
    [(ngModel)]="nome"
    minlength="5"
    required
  > <br><br>

  <label for="email">E-mail </label>
  <input
    type="email"
    name="email"
    id="email"
    [(ngModel)]="email"
    required
    minlength="10"
  > <br><br>

  <label for="senha">Senha </label>
  <input
    type="password"
    name="senha"
    id="senha"
    [(ngModel)]="senha"
    minlength="6"
    required
  > <br><br>
</form>
```

Agora, para concluirmos, devemos informar ao Template Driven Form quais campos dos formulários vão ser controlados por ele. Para isso, nós devemos criar template variables nos inputs e informar que o valor que elas irão receber é o valor **ngModel**

```
<form #meuForm="ngForm" (ngSubmit)="doSomething()">
  <label for="nome">Nome </label>
  <input
    type="text"
    name="nome"
    id="nome"
    [(ngModel)]="nome"
    minlength="5"
    required
    #inputNome="ngModel"
  > <br><br>

  <label for="email">E-mail </label>
  <input
    type="email"
    name="email"
    id="email"
    [(ngModel)]="email"
    required
    minlength="10"
    #inputEmail="ngModel"
  > <br><br>

  <label for="senha">Senha </label>
  <input
    type="password"
    name="senha"
    id="senha"
    [(ngModel)]="senha"
    minlength="6"
    required
    #inputSenha="ngModel"
  > <br><br>
</form>
```

A partir de todas as template variables definidas, tanto a do formulário quanto as dos inputs, nos fornecem propriedades para testar se nossos inputs ou se o próprio formulário estão válidos. Vamos fazer uma validação simples para testar nosso formulário. Vamos ver se o input **nome** está válido. Caso não esteja, vamos mostrar um texto na tela informando que o campo está inválido.

```
<form #meuForm="ngForm" (ngSubmit)="doSomething()">
  <label for="nome">Nome </label>
  <input
    type="text"
    name="nome"
    id="nome"
    [(ngModel)]="nome"
    minlength="5"
    required
    #inputNome="ngModel"
  > <br><br>

  <p *ngIf="inputNome.invalid" style="color: red;">
    0 campo "nome" está inválido
  </p>
</form>
```

Uma das propriedades que nossas template variables possuem é a propriedade **invalid**. Ela informa se o input em que a variável está atrelada está inválida. Nesse caso, quando o input estiver inválido, o parágrafo abaixo irá aparecer. A propriedade **invalid** será verdadeira caso algum dos atributos de validação do HTML que estão dentro do input não estiver sendo atendido.

Por essas propriedades, você também pode saber qual dos atributos não estão sendo atendidos. Para isso, nós podemos utilizar a propriedade **errors**. A partir dela, nós podemos saber exatamente qual atributo não está sendo atendido.

```
<form #meuForm="ngForm" (ngSubmit)="doSomething()">
  <label for="nome">Nome </label>
  <input
    type="text"
    name="nome"
    id="nome"
    [(ngModel)]="nome"
    minlength="5"
    required
    #inputNome="ngModel"
  > <br><br>

  <p *ngIf="inputNome.touched && inputNome.invalid" style="color: red;">
    <span *ngIf="inputNome.errors?.minlength">
      Quantidade de caracteres insuficiente
    </span>
    <span *ngIf="inputNome.errors?.required">
      Este campo é obrigatório
    </span>
  </p>
</form>
```

A partir do atributo **errors**, nós podemos saber o nome da propriedade que não foi atendida e mostrar uma mensagem personalizada para esses atributos.