

Para saber mais: Validando com FormBuilder

Validando mais de um campo com `formBuilder.group()`

Durante o curso utilizei uma outra forma de validação de formulário adicionando um novo valor à função `group()` do `formBuilder`. Antes fizemos um arquivo novo chamado `minusculo.validador.ts` e dentro fizemos a validação do nome de usuário para que apenas letras minúsculas sejam utilizadas. Só que essa função é chamada apenas no campo de usuário. A função `group()` pode receber um parâmetro extra que é para validações e essas validações devem estar dentro de um objeto da mesma forma que os `inputs` do formulário são declarados.

```
{  
  validators: [funçãoDeValidação],  
}
```

[COPIAR CÓDIGO](#)

Criei uma outra validação para conferir se os campos de nome de usuário e senha são iguais, então para isso utilizei a mesma estratégia do arquivo `minusculo.validador.ts`. Foi criado um arquivo chamado `usuario-senha-iguais.validator.ts` e dentro há a seguinte estrutura:

```
: { FormGroup } from '@angular/forms';  
  
: function usuarioSenhaIguaisValidator(formGroup: FormGroup) {  
: it username = formGroup.get('userName')?.value ?? '';  
: it password = formGroup.get('password')?.value ?? '';
```

```
[username.trim() + password.trim()) {  
  return username !== password ? null : { senhaIgualUsuario: true };  
} else {  
  return null;  
}
```

[COPIAR CÓDIGO](#)

Esse arquivo precisa importar o `FormGroup` do Angular e depois podemos construir a função que fará a verificação do nome de usuário e senha. Precisamos do `FormGroup` para acessar esses dois parâmetros do nosso formulário e guardamos cada um desses parâmetros em variáveis e depois fazer um `if` ternário para validar esses dois campos.

No final basta adicionar a mensagem de erro no HTML:

```
<div class="form-group">  
  <input  
    type="password"  
    placeholder="Senha"  
    class="form-control"  
    formControlName="password"  
  />  
  <app-mensagem  
    *ngIf="novoUsuarioForm.errors?.senhaIgualUsuario"  
    mensagem="Senha deve ser diferente do usuário"  
  >  
  </app-mensagem>  
</div>
```

[COPIAR CÓDIGO](#)

Não podemos nos esquecer de importar essa função em nosso arquivo `novo-usuario.components.ts` :

```
UsuarioSenhaIguaisValidator } from './usuario-senha-iguais.validator';
```

COPIAR CÓDIGO

E também precisamos colocar esse objeto de `validators` depois do objeto de `inputs` do form:

```
this.novoUsuarioForm = this.formBuilder.group(  
  {  
    email: ['', [Validators.required, Validators.email]],  
    fullName: ['', [Validators.required, Validators.minLength]],  
    userName: ['', [UsuarioMinusculoValidator]],  
    password: ['', ],  
  },  
  {  
    validators: [usuarioSenhaIguaisValidator],  
  }  
);
```

COPIAR CÓDIGO