05 - VALIDAÇÃO EM FORMULÁRIOS

1. Angular Validations

1.1. Abra o projeto **pw3-forms** e inclua as seguintes validações, Use como referência o guia disponibilizado em https://angular.io/api/forms/Validators.

Campo	Validação	Mensagem
Nome	Obrigatório	'Nome' é obrigatório
	Entre 3 e 35 caracteres	'Nome' deve ter entre 3 e 35 caracteres
E-Mail	Obrigatório	'E-Mail' é obrigatório
	Máximo de 40 caracteres	'E-Mail' deve ter até 40 caracteres
Nome de Usuário	Obrigatório	'Nome de Usuário' é obrigatório
	Máximo de 15 caracteres	'Nome de Usuário' deve ter até 15 caracteres
	Não pode conter espaço	'Nome de Usuário' não deve conter espaços
Senha	Obrigatória	'Senha' é obrigatória
	Deve ter entre 8 e 16 caracteres e conter 1 número, 1 letra maiúscula e 1 dígito	'Senha' não está no padrão correto
Confirmar Senha	Obrigatória	'Confirmar Senha' é obrigatória
	Deve ser igual à senha	'Confirmar Senha' é diferente da senha

Obs.: no caso da senha, utilize o pattern "**(?=.*[A-Z])(?=.*[!.@#$&*])(?=.*[0-9])(?!.*).{8,16}$**".

2. Validações Personalizadas

2.1. Crie uma nova classe para as validações personalizadas com o comando **ng g class validator/custom-validator**.

2.2. Crie uma função estática, por exemplo, que verifique se o campo contém espaços:

```
static contemEspacos(control: AbstractControl) {
  if (!control.value) {
    return null;
  }

  let resultado = control.value.indexOf(' ') !== -1;

  if (resultado) {
    control.setErrors({ contemEspacos: true });
    return { contemEspacos: true };
  } else {
    control.setErrors(null);
    return null;
  }
}
```

2.3. Utilize a função validadora dentro do arquivo **cadastro-component.ts**:

```
usuario: ['', CustomValidator.contemEspacos],
```

2.4. Renderize o erro **contemEspacos** no arquivo **cadastro-component.html**:

```
<mat-error *ngIf="cadastroForm.controls.usuario.hasError('contemEspacos')">'Nome de Usuário' não deve conter espaços</mat-error>
```

3. Validações com dois campos

3.1. Para verificar se a confirmação de senha é igual a senha, crie uma nova função estática no arquivo **custom-validator.ts**:

```
static mesmasSenhas(senha: string, confirmaSenha: string) {
  return (formGroup: AbstractControl): ValidationErrors | null => {
    const senhaControl = formGroup.get(senha);
    const confirmaSenhaControl = formGroup.get(confirmaSenha);

    if (!senhaControl || !confirmaSenhaControl) {
      return null;
    }

    if (senhaControl.value !== confirmaSenhaControl.value) {
      confirmaSenhaControl.setErrors({ senhasDiferentes: true });
      return { senhasDiferentes: true };
    } else {
      confirmaSenhaControl.setErrors(null);
      return null;
    }
  };
}
```

3.2. Utilize a função validadora dentro do arquivo **cadastro-component.ts**:

```
  confirmarSenha: ['']
},

{
  validators: [
    CustomValidator.mesmasSenhas("senha", "confirmarSenha")
  ],
}

);
```

3.3. Renderize o erro **sehasDiferentes** no arquivo **cadastro-component.html**.