Reactive Forms.

É a nova forma que o angular trás para implementar formulários. Ao invés de usar ngModel e configurar validadores nos campos, criamos instancias de formGroup ou formControl dentro de um componente.

Reactive forms também possuem os mesmo validadores do templates forms: min e maxlength, required e pattern

Também é possível criar um grupo, sintaxe:

  <form [formGroup]="orderForm" novalidate>

    <div class="col-sm-6 col-xs-12">

            <mt-input-container errorMessage="Campo obrigatório e com 5 caracteres" label="Endereço">

             <input type="text" class="form-control"  formControlName="name" minlength="5" required placeholder="Nome" autocomplete="off">

            </mt-input-container>

          </div>

<div formGroupName=”address”>

<input formControlName=”street”/>

</div>

No componente, já com as validações:

  orderForm: FormGroup;

 this.orderForm = this.fb.group({

      name: this.fb.control('', [Validators.required, Validators.minLength(5)]),

      email: this.fb.control('', [Validators.required, Validators.pattern(this.emailPattern)]),

      emailConfirmation: this.fb.control('', [Validators.required, Validators.pattern(this.emailPattern)]),

      address: this.fb.control('', [Validators.required, Validators.minLength(5)]),

      number: this.fb.control('', [Validators.required, Validators.pattern(this.numberParttern)]),

      optionalAddress: this.fb.control(''),

      paymentOption: this.fb.control('', [Validators.required])

    })

Criando a validação para email e número:

emailPattern = /^(([^<>()\[\]\.,;:\s@\"]+(\.[^<>()\[\]\.,;:\s@\"]+)\*)|(\".+\"))@(([^<>()[\]\.,;:\s@\"]+\.)+[^<>()[\]\.,;:\s@\"]{2,})$/i;

  numberParttern = /^[0-9]\*$/;