



Jean Junior Heldt

- Graduação: TADS - UNIPAR
- Especialização: Gestão de TI - FASUL
- Desenvolvedor na CWI Software



Aula 04

- Angular Material

<https://material.angular.io/>

- Projeto utilizado, cópia do *app-unipar2*

<https://github.com/jeanjunior/app-unipar4>



Angular Material

Alta qualidade

Componentes internacionalizados e acessíveis para todos. Bem testado para garantir desempenho e confiabilidade.

APIs simples com comportamento consistente entre plataformas.

Versátil

Fornece ferramentas que ajudam os desenvolvedores a construir seus próprios componentes personalizados com padrões de interação comuns.

Personalizável dentro dos limites da especificação do Material Design.

Sem atrito

Construído pela equipe do Angular para se integrar perfeitamente ao Angular.

Comece do zero ou entre em seus aplicativos existentes.

```

1 <!-- Campo de UserName -->
2 <div class="col-md-4 mt-1 form-group">
3   <label class="form-control-label">Login do usuário</label>
4   <input type="text" class="form-control" formControlName="username"
5     [ngClass]='{'is-invalid': hasErrors(username!)}">
6   <app-message-valid-control [control]="username!"> </app-message-valid-control>
7 </div>
8
9 <!-- Campo de Senha -->
10 <div class="col-md-4 mt-1 form-group">
11   <label class="form-control-label">Senha</label>
12   <input type="password" class="form-control" formControlName="password">
13   <app-message-valid-control [control]="password!"></app-message-valid-control>
14 </div>

```

<= Bootstrap

Material =>

```

1 import { MatFormFieldModule } from '@angular/material/form-field';
2 import { MatInputModule } from '@angular/material/input';

```

```

1 <!-- Campo de UserName -->
2 <mat-form-field class="col-md-4 mt-1">
3   <mat-label>Login do usuário</mat-label>
4   <input type="text" matInput formControlName="username">
5   <app-message-valid-control [control]="username!"> </app-message-valid-control>
6 </mat-form-field>
7
8 <!-- Campo de Senha -->
9 <mat-form-field class="col-md-4 mt-1">
10  <mat-label>Senha</mat-label>
11  <input type="password" matInput formControlName="password">
12  <app-message-valid-control [control]="password!"></app-message-valid-control>
13 </mat-form-field>

```

Animations

A animação fornece a ilusão de movimento: os elementos HTML mudam de estilo com o tempo. Animações bem projetadas podem tornar seu aplicativo mais divertido e fácil de usar, mas não são apenas cosméticas. As animações podem melhorar seu aplicativo e a experiência do usuário de várias maneiras:

- Sem animações, as transições de páginas da web podem parecer abruptas e chocantes.
- O Motion aprimora muito a experiência do usuário, portanto, as animações oferecem aos usuários a chance de detectar a resposta do aplicativo às suas ações.
- Boas animações chamam intuitivamente a atenção do usuário para onde é necessário.

Normalmente, as animações envolvem várias *transformações de* estilo ao longo do tempo. Um elemento HTML pode mover, mudar de cor, aumentar ou diminuir, esmaecer ou deslizar para fora da página. Essas mudanças podem ocorrer simultaneamente ou sequencialmente. Você pode controlar o tempo de cada transformação.

O sistema de animação do Angular é baseado na funcionalidade CSS, o que significa que você pode animar qualquer propriedade que o navegador considere animável. Isso inclui posições, tamanhos, transformações, cores, bordas e muito mais. O W3C mantém uma lista de propriedades animáveis em seu página.

<https://angular.io/guide/animations>

<https://github.com/jeanjunior/app-unipar-animation>

Deploy

Dockerfile X

Dockerfile > ...

```
1  ### STAGE 1: Build ###
2  FROM node:14.15-alpine AS build
3  WORKDIR /usr/src/app
4  COPY package.json package-lock.json ./
5  RUN npm install
6  COPY . .
7  RUN npm run build-prod
8
9  ### STAGE 2: Run ###
10 FROM nginx:1.21.0-alpine
11 COPY nginx.conf /etc/nginx/nginx.conf
12 COPY --from=build /usr/src/app/dist/app-unipar4 /usr/share/nginx/html
13
```

.dockerignore U X

.dockerignore

```
1  |git
2  .firebase
3  .editorconfig
4  /node_modules
5  /e2e
6  /docs
7  .gitignore
8  *.zip
9  *.md
```

nginx.conf X

nginx.conf

```
1  |events{}
2
3  http {
4
5      include /etc/nginx/mime.types;
6
7      server {
8          listen 80;
9          server_name localhost;
10         root /usr/share/nginx/html;
11         index index.html;
12
13         location / {
14             try_files $uri $uri/ /index.html;
15         }
16     }
17 }
18
```

```
1  // Faz o build e gera a imagem
2  docker build -t app-unipar4-image .
3
4  // Faz o deploy de um container com a imagem
5  docker run --name app-unipar4-container -d -p 8080:80 app-unipar4-image
```