

# DESAFIO

## ANGULAR

---

Olá, estudante!

Chegou o momento de colocar em prática os conhecimentos aprendidos e, com isso, desenvolver competências. Para tanto, leia o detalhamento apresentado a seguir para compreender como realizar o desafio proposto. Prepare-se e bons estudos!

## ORIENTAÇÕES GERAIS

Para o desenvolvimento desse desafio, considere que:

- A atividade será realizada individualmente.
- A entrega ocorrerá no Ambiente Virtual de Aprendizagem (AVA).
- A data limite para entrega será informada no AVA.
- As fontes das referências bibliográficas deverão seguir as normas da Associação Brasileira de Normas Técnicas (ABNT), caso utilizadas.
- As etapas para a realização da tarefa devem ser lidas atentamente antes de iniciar.

## OBJETIVOS DE APRENDIZAGEM

- Explicar os recursos da linguagem *TypeScript*.
- Utilizar os recursos da linguagem *TypeScript*.
- Utilizar o Angular para o desenvolvimento ágil de aplicações robustas.
- Desenvolver componentes reativos para ter bons resultados de interatividade com o usuário.
- Otimizar aplicações com foco em performance.

## CONTEÚDOS FORMATIVOS

- Entendendo *TypeScript*
- Introdução *framework* Angular
- Angular: boas práticas em arquiteturas e formulários
- Angular: controle o fluxo de navegação
- Angular e RxJS: programação reativa

## EQUIPAMENTOS E MATERIAIS

Descrição do item	Especificação	Quantidade
Computador	Sistema operacional Windows 10 ou superior	1 por estudante

# DESCRÍÇÃO E PROCEDIMENTOS

## PROPOSTA

Nesta atividade, você irá desenvolver um projeto de *dashboard* para uma grande empresa, aplicando os conhecimentos adquiridos sobre a criação de componentes *Front-end* reativos com o *framework* Angular e a linguagem *TypeScript*. Para isso, considere a seguinte situação:

A empresa FORD está expandindo seu sistema com o objetivo de apoiar a equipe de gestão nas tomadas de decisões estratégicas. Para isso, foi solicitado o desenvolvimento de um *dashboard* que apresente informações relevantes sobre os veículos da empresa. Esses dados serão fornecidos por um *back-end* por meio de uma API REST.

Seu desafio é criar um sistema contendo três *interfaces* utilizando o *framework* Angular. O sistema deverá incluir, em sua estrutura, *modules*, *componentes* e *services* do Angular. Para isso, é importante considerar que:

- A API do *back-end*, que fornece os dados para a construção do *front-end*, foi disponibilizada.
  - Para a compilação e execução da API, devem ser executados no terminal os comandos “npm install” e, em seguida, “npm start”.
  - Os dados estarão disponíveis em <http://localhost:3000>.
- Foram disponibilizados as imagens e os arquivos dos modelos em *TypeScript* de usuário e veículo.

Agora, leia o detalhamento de cada ação a seguir para realizar a atividade proposta.

## AÇÃO 1

A primeira página do sistema corresponde à *interface* de *login*. Para estruturá-la, siga os passos abaixo:

Passo 1: desenvolver um formulário para aquisição de nome e senha.

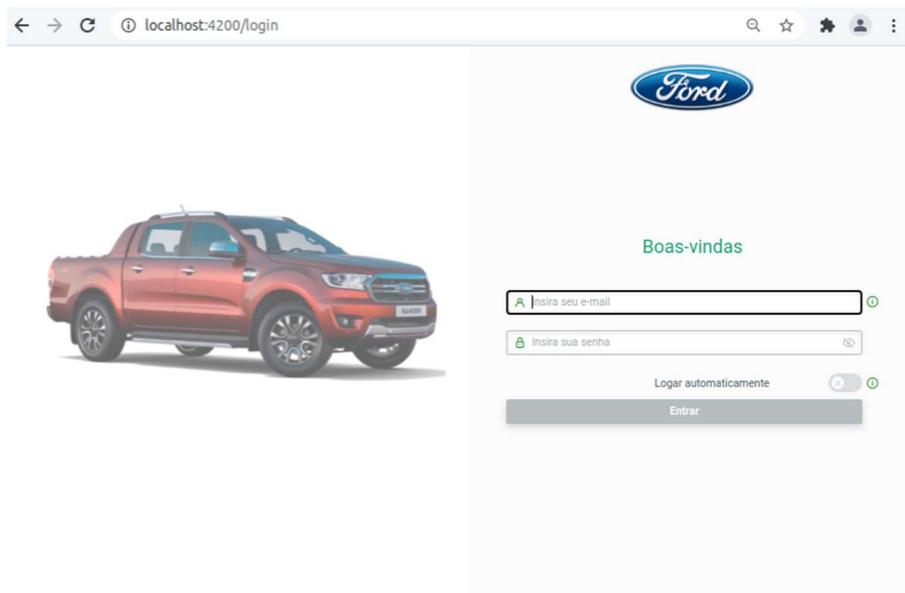
Passo 2: implementar um botão para confirmação do *login*.

Passo 3: desenvolver o módulo *login* com capacidade de busca dos dados de usuário cadastrados no *back-end*, autenticação no sistema e validação com base nos seguintes dados cadastrados:

- Usuário: admin
- Senha: 123456

Para visualizar essa estrutura, veja a seguir o protótipo da página:

Figura 1 - Protótipo da página de *login*



Fonte: Ford (2022).

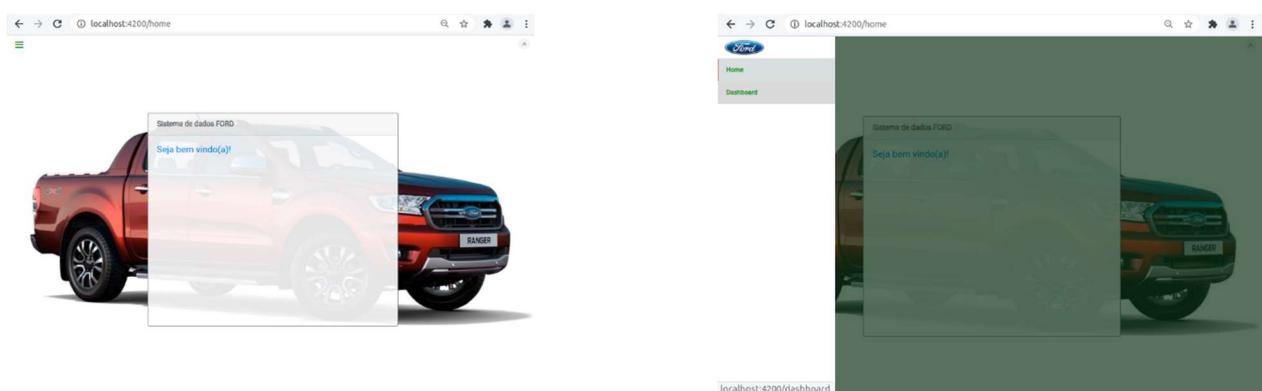
## AÇÃO 2

Após efetuar o *login*, o sistema fornecerá acesso à página de *home*. Para estruturá-la, siga os passos abaixo:

- Passo 4: adicionar um cartão com informação de boas-vindas ao sistema.
- Passo 5: incluir um item de *logout*.
- Passo 6: inserir uma imagem de fundo.
- Passo 7: disponibilizar um *link* para acesso à página *dashboard*.

Para visualizar essa estrutura, veja a seguir o protótipo da página:

Figura 2 - Protótipo da página de *home*



Fonte: Ford (2022).

## AÇÃO 3

A página denominada *Dashboard* é a principal do *site* e apresentará as informações disponibilizadas pelo *back-end*. Para estruturá-la, siga os passos abaixo:

Passo 8: adicionar um cartão com campo de busca, por nome do modelo do veículo.

- Buscar as opções de preenchimento no *back-end*.
- Utilizar o endereço: <http://localhost:3000/vehicle>

Passo 9: inserir três cartões apresentando os dados de total de vendas, número de veículos conectados ao sistema Ford e número de veículos com software atualizado.

- Atualizar essas informações conforme o modelo selecionado no primeiro cartão.

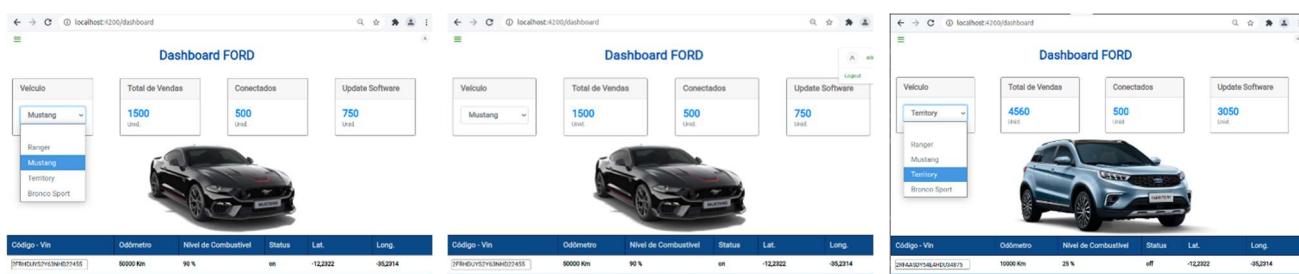
Passo 10: exibir, ao centro da página, a imagem do veículo correspondente ao modelo selecionado no primeiro cartão.

Passo 11: incluir uma tabela com campo de busca por código do veículo e cinco campos para apresentação de dados.

- Utilizar como exemplo de código do veículo: 2FRHDXYS2Y63NHD22454
- Utilizar o endereço: <http://localhost:3000/vehicleData>

Para visualizar essa estrutura, veja a seguir o protótipo da página:

Figura 3 - Protótipo das páginas de *Dashboards*



## ANEXOS

Para contribuir com a estruturação do desafio, foi organizado um arquivo ZIP contendo o arquivo de suporte Angular completo da proposta. Para acessá-la clique no *link* abaixo:

Arquivos: [https://cimatecead.senaicimatec.com.br/Ucs/extensao/programacao-front-end/desafio%207\\_Angular.zip](https://cimatecead.senaicimatec.com.br/Ucs/extensao/programacao-front-end/desafio%207_Angular.zip)

## CRTÉRIOS DE AVALIAÇÃO

Seu desafio será analisado pelo docente, que observará a entrega considerando os seguintes aspectos:

- Utilização apropriada de classes e funções da linguagem *TypeScript* para construção de componentes do *framework Angular*.
- Utilização de *modules*, *components* e *services* do Angular.
- Comunicação do *back-end* disponibilizado.

- Utilização de *bootstrap*.
- Utilização das diretivas do Angular, como *ngModel*, *ngIf* e *ngFor*.
- Uso da biblioteca *RxJS* por meio de seus operadores, como *map*, *pluck*, *debounceTime*, *filter* e *distinctUntilChanged*.

Sucesso!