

# MarketPlace com Angular

Diana Martine  
Dev. Front-end.

# Mais sobre mim

- Dev. Front-end com experiência em Back-end;
- SO's mobile e área de desenvolvimento web;
- Curiosidade e compartilhar conhecimento;
- Musicista profissional e artista;
- Bootcamp voltado para pessoas trans;
- Minhas redes:
  - [linkedin/in/dianamartine](#)
  - [github.com/dianamartine](#)
  - [Instagram.com/transdevoficial](#)

# Desafio

Vamos criar um marketplace usando o *framework* Angular. Você será capaz de compreender os conceitos de SPA's(*Single Page Applications*) e principais blocos do Angular.

O Angular é amplamente utilizado no desenvolvimento Frontend.

# Percurso

**Aula 1**

Overview

**Aula 2**

Iniciando o projeto

**Aula 3**

Homepage

**Aula 4**

Listando produtos

# Percurso

## Aula 5

Add produtos e Checkout

## Aula 6

Melhorias no projeto

# Requisitos

- Lógica de programação e POO;
- Conhecimentos sólidos(*TypeScript, JavaScript, HTML*);
- Computador com SO de sua preferência(*Windows, Linux, Mac OS*);
- Ferramentas de desenvolvimento(*IDE, Angular CLI, Node.js na versão 12.14+, TypeScript, Angular Material*).

# Dúvidas?

> Comunidade online (Discord)



SCAN ME

# Etapa 1: Overview

> MarketPlace com Angular



# Objetivos

- História do Angular;
- Vantagens;
- Conceitos importantes.

# História do Angular

Criado entre 2008/2009 pelos devs Misko Hevery e Adan Abrons, surgiu com a ideia de simplificar o desenvolvimento de apps web. Seu lançamento oficial foi em maio de 2011 chamado de AngularJS, as versões posteriores são chamadas somente por Angular. Muito utilizado no desenvolvimento Front-end e em SPA's.

# Vantagens

- *Open-Source*;
- Mantido pelo Google;
- Arquitetura da aplicação em camadas bem definidas;
- Permite a criação modular e de componentes reutilizáveis;

# Conceitos importantes

- ***Components:***

É o que vai aparecer para o usuário, é a view do Angular, que encapsula os conceitos de ***Template***, ***Metadata*** e ***Data Binding***.

Obs.: Tudo que temos na aplicação é um Component.

# Conceitos importantes

- ***Templates***: HTML do Component;
- ***Metadata***: Faz o processamento das Classes;
- ***Data Binding***: Associação dos dados com o HTML.

# Conceitos importantes

Obs.: Não é boa prática aplicar lógica de negócio em um Component. Para isso usamos:

- ***Services***: Implementa a lógica de negócio e pode ser injetado em outras Classes, por ***Dependency Injection***.

# Conceitos importantes

- ***Router***: Responsável pela navegação;
- ***Directives***: Modifica elementos DOM (*Document Object Model*) e/ou seu comportamento;

# Para saber mais

- Angular - [angular.io/](https://angular.io/)
- Angular Glossary - [angular.io/guide/glossary/](https://angular.io/guide/glossary/)
- Angular Docs - [angular.io/docs/](https://angular.io/docs/)
- O que é o Angular - [andrefelizardo.com.br/blog/o-que-e-angularjs/](https://andrefelizardo.com.br/blog/o-que-e-angularjs/)
- Node.js - [nodejs.org/](https://nodejs.org/)
- Angular CLI - [npmjs.com/package/@angular/cli/](https://npmjs.com/package/@angular/cli/)
- TypeScript - [npmjs.com/package/typescript/](https://npmjs.com/package/typescript/)
- Angular Material - [material.angular.io/](https://material.angular.io/)
- Material icons - [fonts.google.com/icons](https://fonts.google.com/icons)



# Dúvidas?

> Comunidade online (Discord)



SCAN ME

# Etapa 2: Iniciando o projeto

> MarketPlace com Angular

# Objetivos

- Criando nosso backend;
- Iniciando o projeto;
- Entendendo conteúdo gerado.

# Criando nosso backend

Para nossos dados iremos utilizar uma lib chamada json-server:

criar nossa base de dados;

**npm init;**

**npm i json-server;**

**npm start;**

# Iniciando o projeto

Para criar um novo projeto em Angular usamos o comando:

```
ng new {nome-do-projeto};
```

Ao finalizar serão criados alguns arquivos e diretórios padrões de projetos em Angular.

# Entendendo conteúdo gerado

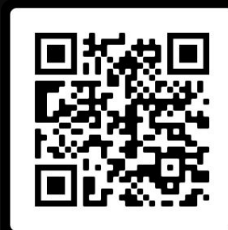
- **src:** pasta que contém os componentes da aplicação.
- **app:** componente criado por padrão que possui:
  - Módulo de rotas;**
  - Folha de estilo.**
  - Template.**
  - Arquivo de teste.**
  - Arquivo componente.**
  - Módulo componente.**

# Entendendo conteúdo gerado

- **assets:** pasta que contém ativos do projeto, como utilidades por exemplo.
- **environments:** pasta que contém arquivos de configuração de compilação.
- **index.html:** nosso componente principal, que recebe dentro de si o componente **app**.

# Dúvidas?

> Comunidade online (Discord)



SCAN ME



# Etapa 3: Homepage

> MarketPlace com Angular

# Objetivos

- Explorando poder do Angular material;
- Criando header, footer & nav component;
- Criando nossa home view.

# Explorando poder do Angular material

Para nossa estilização iremos utilizar uma lib chamada angular-material:

```
ng add @angular/material;
```

# Criando header, footer && nav component

Para criar um novo componente no Angular usamos:  
**ng generate component {nome-do-componente}** ou  
**ng g c {nome-do-componente}** ou  
**ng g c**  
**{diretorio-do-componente/nome-do-componente}.**

No angular podemos definir variáveis com um tipo obrigatório (!) ou opcional (?).

# Header

Para nosso header vamos usar dados da pasta **assets** e componentes do **angular material**.

# Footer

Para nosso footer vamos componentes do **angular material**.

# Nav

Para nossa nav vamos componentes do **angular material**.

Além de explorar o conceito de métodos em eventos e variável local no **template**.

# Criando nossa home view

Para nossa home view vamos usar dados da pasta **assets** e componentes do **angular material**.

E vamos explorar também o conceito de rotas para criar uma rota para nossa home view.



# Dúvidas?

> Comunidade online (Discord)



SCAN ME

# Etapa 4: Listando produtos

> MarketPlace com Angular

# Objetivos

- Criando nosso modelo de produto;
  - Criando nosso primeiro service;
  - Renderizando produtos na lista
- (Observable, Subscribe e \*ngFor).**

# Criando nosso modelo de produto

Iremos criar uma classe modelo de produto para padronizar os produtos que vamos receber nas requisições e renderizar na lista de produtos.

No angular podemos definir variáveis com um tipo obrigatório (!) ou opcional (?).

# Criando nosso primeiro service

Para criar um novo service no Angular usamos:

**ng generate service {nome-do-servico}** ou

**ng g s {nome-do-servico}** ou

**ng g s {diretorio-do-servico/nome-do-servico}**.

Vamos usar o **HttpClient** do próprio angular para fazer as requisições.

Além de modificadores de acesso, como **public**, **protected** e **private**.

# Renderizando produtos na lista

Ao realizar uma requisição o retorno dela será um **Observable**, que nada mais é que um observador de algo, nesse caso do nosso produto.

Podemos chamar o método e criar uma **Subscribe**, que seria a inscrição para o observador.

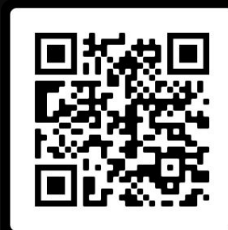
# Renderizando produtos na lista

Vamos criar dois novos componente e atualizar as rotas.

Para renderizar uma lista de dados no **template** do angular, podemos usar uma **directive**, ou diretiva, nesse caso usaremos a **\*ngFor** que cria uma variável local para percorrer uma matriz ou vetor de dados e renderizar em um componente html.

# Dúvidas?

> Comunidade online (Discord)



SCAN ME



# Etapa 5: Add produtos e Checkout

> MarketPlace com Angular

# Objetivos

- Métodos de add e remover produtos;
- Criando botão de select;
- Criando página de checkout;
- Explorando o poder do Angular Material;
- Criando botão de pagamento com mensagem em snackbar;

# Add e remover produtos

Para manipular os dados selecionados iremos criar um manipulador (**handler**) que permite buscar ou modificar o estado do preço e do produto selecionado, através de métodos assessores, como **getter** e **setter**.

# Criando botão de select

Iremos criar um botão separado que irá receber os métodos de selecionar e remover produtos para que a responsabilidade seja de alterar o estado de somente um produto por vez.

# Criando página de checkout

Para nosso checkout iremos renderizar os produtos selecionados.

Vamos criar mais uma rota e atualizar o módulo de rotas.

# Explorando o poder do Angular Material

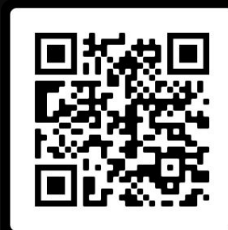
Vamos explorar formulários, campos, labels e inputs do Angular material.

# Criando botão de pagamento com mensagem em snackbar

Vamos explorar o componente **snack-bar** do angular material, e criar um botão de pagamento que mostra uma mensagem em pop-up na nossa tela.

# Dúvidas?

> Comunidade online (Discord)



SCAN ME



# Etapa 6: Melhorias no projeto

> MarketPlace com Angular

# Objetivos

- Contador de produtos selecionados;
- Explorando pipes do Angular;
- Explorando o data binding;
- Melhorando nosso checkout;

# Contador de produtos selecionados

Vamos criar um contador para os produtos selecionados.

# Explorando pipes do Angular

Pipes são transformadores de dados.

Exploraremos a **pipe currency** para transformar os dados de preço.

# Explorando o data binding

Vamos explorar o conceito de **two-way data binding** para criarmos um objeto com informações do cliente.



# Melhorando nosso checkout

Add regras no checkout.

# Desafio

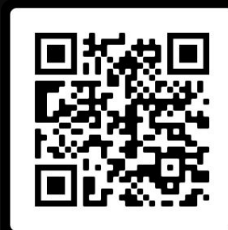
- Terminar a página de about, linkando as rotas e atualizando os componentes na nossa view;
- Subir o projeto em um repositório no GitHub.

Fiquem a vontade para usar o conhecimento aqui adquirido.

Tudo certo jovem!

# Dúvidas?

> Comunidade online (Discord)



SCAN ME