



Introdução

Desenvolvimento WEB

UNISANTA

Índice

Aplicações WEB.....	3
Introdução.....	3
Exemplo de arquitetura de uma aplicação WEB.....	4
Preparação do Ambiente.....	5
Instalação do Visual Studio Code.....	5
Testando o Visual Studio Code.....	5
Instalação do git.....	6
Testando o git.....	6
Instalação do Node.js.....	6
Testando o Node.....	6
Instalação do Angular CLI.....	7
Testando o Angular CLI.....	7
Instalação do Postman.....	8
Testando o Postman.....	8
Projeto TODOapp.....	9
Criando o workspace.....	10
Criando o projeto TODOapp.....	11
Executando o servidor.....	13
Testando o TODOapp localmente a partir do browser.....	14
Alterando a View do AppComponent.....	15
Testando no Browser.....	15
Histórico de revisões.....	16



Aplicações WEB

Introdução

Segue um resumo (bem resumido) da evolução do conteúdo WEB ao longo do tempo:

- Páginas HTML Estáticas
- Java Script
- Pacotes reutilizáveis (Libraries)
- Frameworks Frontend e Backend

No início o conteúdo era disponibilizado basicamente através de **Páginas Estáticas**. De forma grosseira podemos comparar isso ao conteúdo de um livro. Alguém escrevia o conteúdo da página e disponibilizava para ser acessado a partir de um Browser de Internet (ex: Google Chrome, Microsoft Edge, Firefox etc). Para haver uma alteração na informação, alguém precisaria alterar o conteúdo da página.

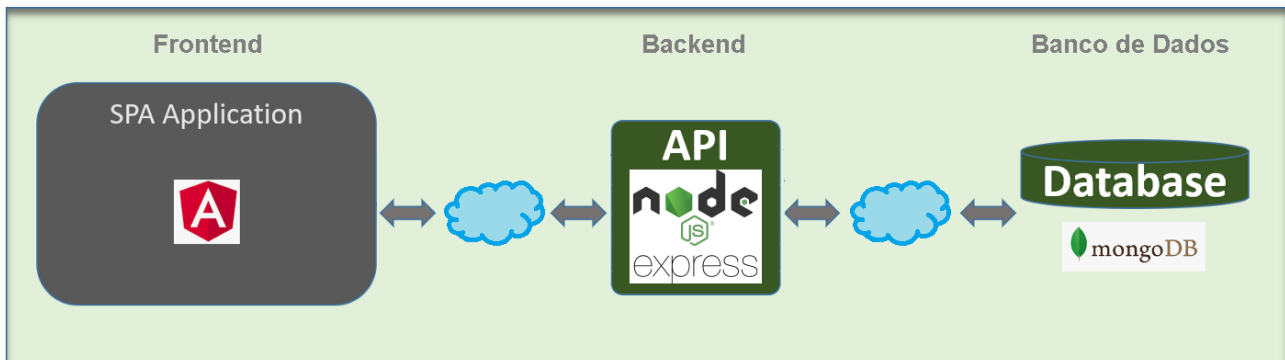
Com o passar do tempo, os browsers passaram a executar código escrito na linguagem **Java Script**. Isso permitiu que o conteúdo pudesse se comportar de modo mais dinâmico, pois era possível de ser alterado pelo programa que estava rodando dentro do browser.

Com o sucesso do **Java Script**, foram criadas bibliotecas (**Libraries**) que agrupavam código interessante para resolver vários problemas comuns.

Com a crescente disponibilidade da Internet e com o significativo aumento de possibilidade de aplicações, os sistemas se tornaram cada vez mais complexos e trabalhosos. Surgem então os **Frameworks** para facilitar a criação de aplicações através de formas padronizadas de desenvolvimento. Atualmente existem muitos Frameworks disponíveis, cada um com suas características específicas.

Exemplo de arquitetura de uma aplicação WEB

Atualmente existem muitas formas para a criação de aplicações WEB. Segue abaixo um diagrama simplificado da arquitetura escolhida para utilizarmos em nosso estudo:



Como podemos ver acima, nossa aplicação será composta por três partes principais:

- Frontend
- Backend
- Banco de Dados

Na esquerda temos o **Frontend**, que é a parte da aplicação acessada pelo usuário. Esta parte roda no browser do dispositivo do usuário (computador, smartphone, tablet etc). Para desenvolver esta parte, utilizaremos um **Framework** chamado de **Angular**. É comum também chamarmos a parte do Frontend de **Client-Side**.

Na parte do meio temos o **Backend**, que roda em um SERVIDOR na nuvem. Esta parte é que vai disponibilizar os serviços funcionando como uma API (Application Programming Interface). Para desenvolver esta parte, usaremos um **Framework** chamado de **Express**. É muito comum também chamarmos a parte do Backend de **Server-Side**.

Na direita temos o BANCO DE DADOS, que é a parte responsável por armazenar os dados manipulados pela aplicação. Em nossa aplicação, utilizaremos um Banco de Dados chamado de **MongoDB**.



Preparação do Ambiente

Para desenvolver nossa aplicação, vamos precisar instalar as seguintes ferramentas em um computador com o Sistema Operacional Windows:

- Visual Studio Code
- Git
- Node.js
- Angular CLI
- Postman

Instalação do Visual Studio Code

Este será o editor de texto utilizado para escrevermos os nossos programas. Considerando que todos os exemplos serão realizados com ele, é muito recomendado que você instale em seu computador.

Link para baixar o arquivo de instalação:

<https://code.visualstudio.com/download>

Testando o Visual Studio Code

Abrir o Prompt de Comando do Windows e executar o seguinte comando:

```
code .
```

O VsCode será executado

Instalação do git

O git é um software muito eficiente para gerenciar alterações em arquivos. Por mais que não façamos uso do mesmo em nossas aulas, é recomendado sua instalação para evitar mensagens de warning provenientes do Angular CLI.

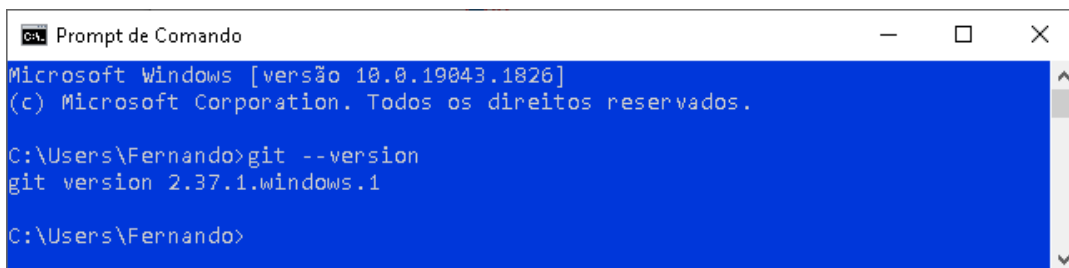
Link para baixar o arquivo de instalação:

<https://git-scm.com/downloads>

Testando o git

Abrir o Prompt de Comando do Windows e executar o seguinte comando:

```
git --version
```



```
Microsoft Windows [versão 10.0.19043.1826]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\Fernando>git --version  
git version 2.37.1.windows.1  
  
C:\Users\Fernando>
```

Instalação do Node.js

O Node é um ambiente para execução de JavaScript. Ele será utilizado em todas as aulas, seja para oferecer o ambiente de desenvolvimento do App Angular como também para disponibilizar o servidor feito Express no Backend.

Link para baixar o arquivo de instalação:

<https://nodejs.org/en/>

Testando o Node

Abrir o Prompt de Comando do Windows e executar o seguinte comando:

```
node --version
```

O programa apresentará a versão instalada.



Instalação do Angular CLI

O Angular CLI oferece o ambiente de desenvolvimento do App Angular. Para ser instalado ele faz uso do gerenciador de pacotes (npm) oferecido pelo Node.js.

Abrir o Prompt de Comando do Windows e executar o seguinte comando:

```
npm install -g @angular/cli
```

Testando o Angular CLI

Abrir o Prompt de Comando do Windows e executar o seguinte comando:

```
ng version
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 10.0.19043.1826]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Fernando>ng version

Angular CLI
Angular CLI: 14.0.6
Node: 16.16.0
Package Manager: npm 8.14.0
OS: win32 x64

Angular:
...

Package                                Version
-----
@angular-devkit/architect              0.1400.6 (cli-only)
@angular-devkit/core                   14.0.6 (cli-only)
@angular-devkit/schematics             14.0.6 (cli-only)
@schematics/angular                   14.0.6 (cli-only)

C:\Users\Fernando>
```

Instalação do Postman

Vamos utilizar o Postman para enviar requisições HTTP de teste de nosso servidor no Backend.

Link para baixar o arquivo de instalação:

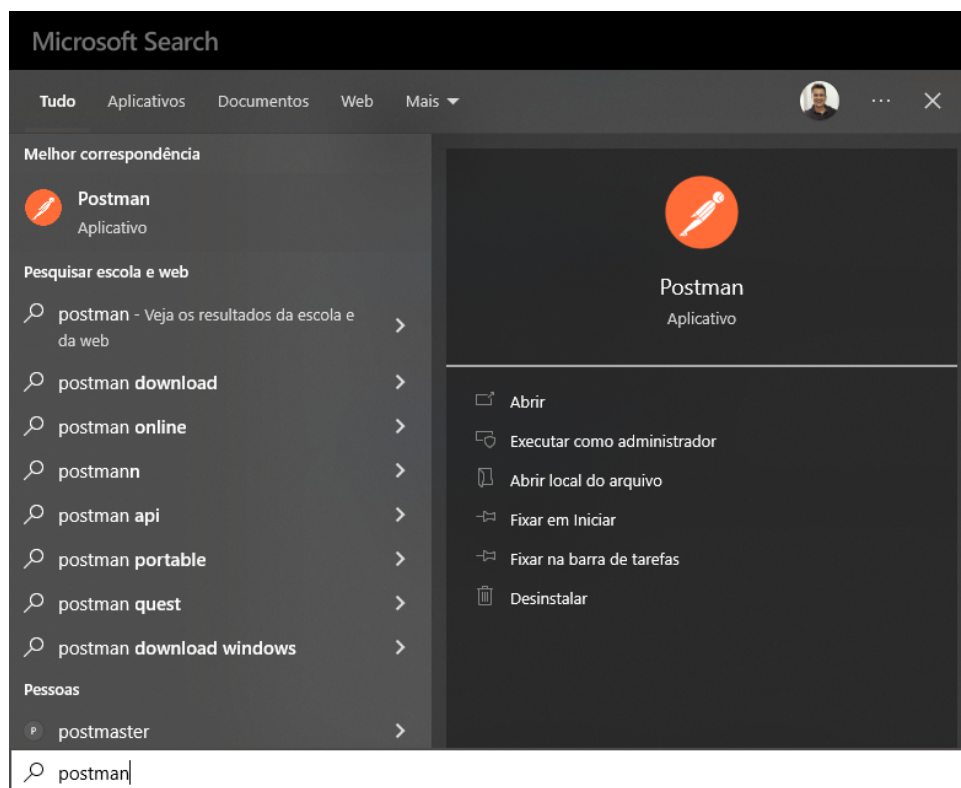
<https://www.postman.com/downloads/>

Testando o Postman

Pressione o botão iniciar do Windows e digite:

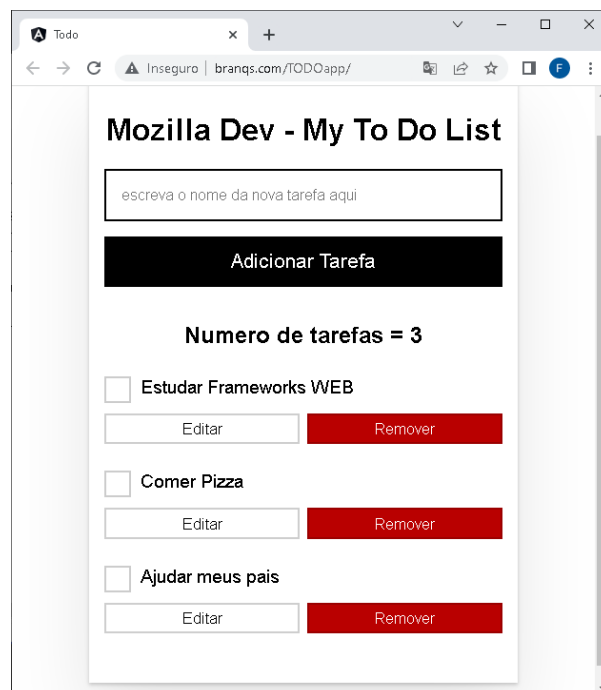
postman

Execute o programa a partir do ícone apresentado. Ex:



Projeto TODOapp

Vamos iniciar o desenvolvimento do nosso primeiro aplicativo Frontend, que será destinado para gerenciar uma lista de **Tarefas a Fazer**, permitindo Gravar, Consultar, Alterar e Excluir (CRUD) tarefas de uma lista. Segue o resultado que pretendemos atingir até o final da primeira parte do curso (Frontend) e também um link que permite experimentar a utilização do App:



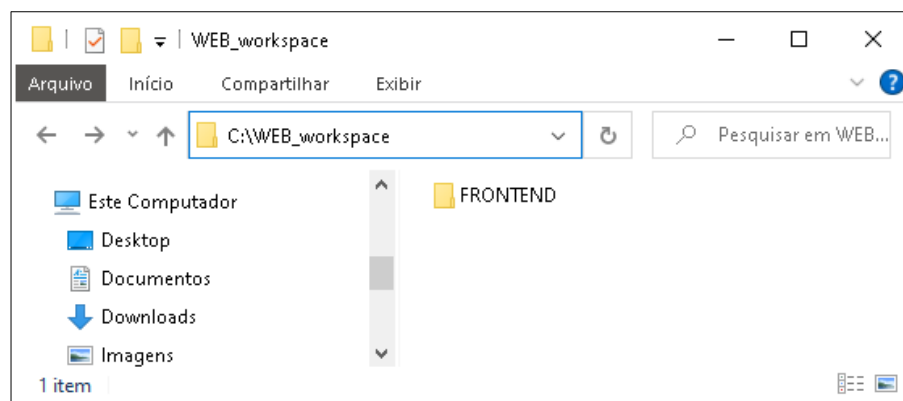
<http://brangs.com/TODOexemplo/>

Nosso App será desenvolvido com ajuda do **Framework Angular**. A principal ferramenta de desenvolvimento do Angular é chamada de **Angular CLI**. Ela permite realizar várias atividades como: Criar projetos; Implementar código estrutural; Efetuar testes; Efetuar Deploy e etc.

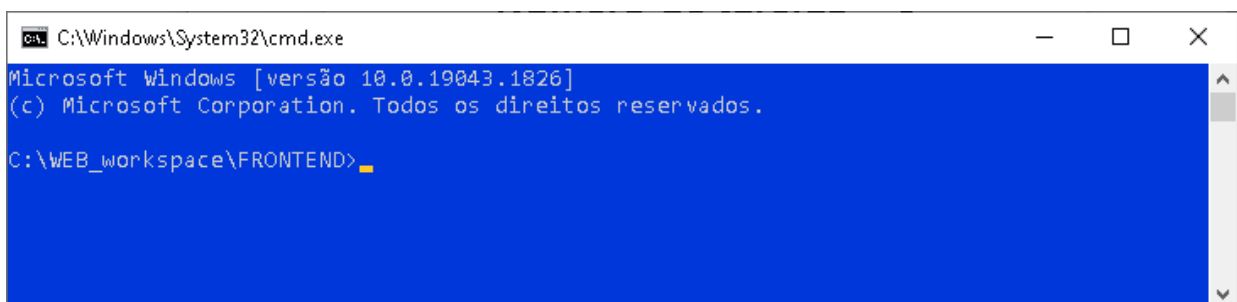
Criando o workspace

Para facilitar o acompanhamento das aulas, crie uma pasta chamada "WEB_workspace" no raiz da unidade C: de seu computador.

Dentro dessa pasta crie uma nova pasta chamada de FRONTEND. Ex:



Acesse o prompt de comando do Windows dentro da pasta FRONTEND. É aqui onde você irá executar os primeiros comandos do Angular CLI para criação e teste de seu App:





Criando o projeto TODOapp

Abra o prompt de comando dentro da pasta FRONTEND, e execute o seguinte comando para criar um App Angular com o nome "TODOapp":

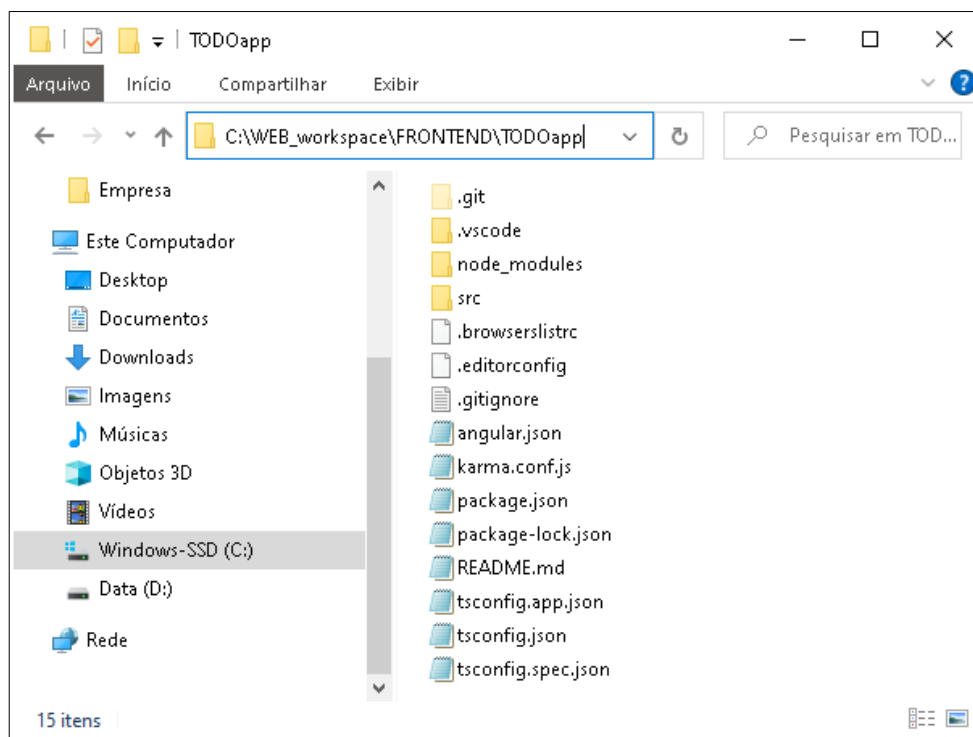
ng new TODOapp --routing=false --style=css

Ao ser executado o comando acima, a ferramenta inicia a criação da estrutura de pastas do seu novo projeto. Ex:

```
C:\WEB_workspace\FRONTEND>ng new TODOapp --routing=false --style=css
[INFO] TODOapp/angular.json (2927 bytes)
[INFO] TODOapp/package.json (1038 bytes)
[INFO] TODOapp/README.md (1061 bytes)
[INFO] TODOapp/tsconfig.json (863 bytes)
[INFO] TODOapp/.editorconfig (274 bytes)
[INFO] TODOapp/.gitignore (548 bytes)
[INFO] TODOapp/.browserslistrc (600 bytes)
[INFO] TODOapp/karma.conf.js (1424 bytes)
[INFO] TODOapp/tsconfig.app.json (287 bytes)
[INFO] TODOapp/tsconfig.spec.json (333 bytes)
[INFO] TODOapp/.vscode/extensions.json (130 bytes)
[INFO] TODOapp/.vscode/launch.json (474 bytes)
[INFO] TODOapp/.vscode/tasks.json (938 bytes)
[INFO] TODOapp/src/favicon.ico (948 bytes)
[INFO] TODOapp/src/index.html (293 bytes)
[INFO] TODOapp/src/main.ts (372 bytes)
[INFO] TODOapp/src/polyfills.ts (2338 bytes)
[INFO] TODOapp/src/styles.css (80 bytes)
[INFO] TODOapp/src/test.ts (749 bytes)
[INFO] TODOapp/src/assets/.gitkeep (0 bytes)
[INFO] TODOapp/src/environments/environment.prod.ts (51 bytes)
[INFO] TODOapp/src/environments/environment.ts (658 bytes)
[INFO] TODOapp/src/app/app.module.ts (314 bytes)
[INFO] TODOapp/src/app/app.component.html (23332 bytes)
[INFO] TODOapp/src/app/app.component.spec.ts (959 bytes)
[INFO] TODOapp/src/app/app.component.ts (211 bytes)
[INFO] TODOapp/src/app/app.component.css (0 bytes)
| Installing packages (npm)...
```

Não se preocupe se aparecerem vários warnings sobre "...LF will be replaced by CRLF the next time..." e também mensagens sobre a falta de identificação do usuário para o git".

O importante é perceber que todos os pacotes foram instalados e que foi criada a seguinte estrutura de pastas para o seu projeto:



A partir de agora, já podemos iniciar o ambiente de desenvolvimento do Angular, executando um servidor que permitirá apresentar o App em execução.

É importante lembrar que este servidor serve somente para efeito de testes de forma local em seu computador durante o desenvolvimento.

Ao final do desenvolvimento, o código final de seu App deverá ser transferido para o ambiente de um servidor HTTP, permitindo que possa ser carregado a partir de qualquer browser.

Executando o servidor

Para que possamos testar nosso App localmente, o Angular CLI oferece um servidor HTTP que disponibiliza o conteúdo de nosso App a partir da porta 4200. Entre na pasta criada para o projeto:

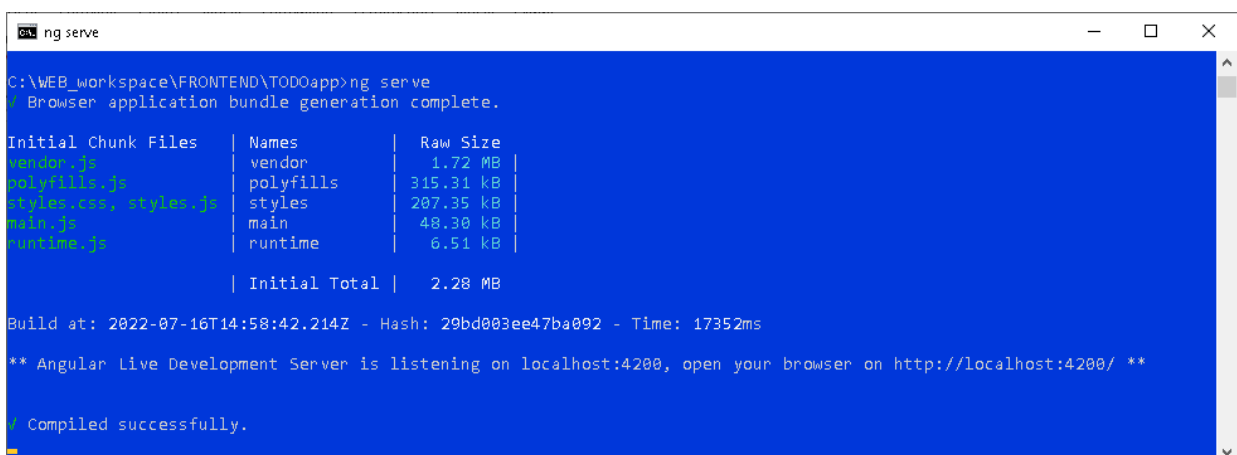
```
cd TODOapp
```

Execute o seguinte comando para iniciar o servidor local:

```
ng serve
```

O Angular então irá gerar um pacote da aplicação para poder ser executado no browser.

Ao final da compilação, a ferramenta vai informar que o App está disponível para uso a partir da porta 4200 de localhost:



```
ng serve
C:\WEB_workspace\FRONTEND\TODOapp>ng serve
✓ Browser application bundle generation complete.

Initial Chunk Files | Names          | Raw Size
---
vendor.js           | vendor         | 1.72 MB
polyfills.js       | polyfills      | 315.31 kB
styles.css, styles.js | styles         | 207.35 kB
main.js            | main           | 40.30 kB
runtime.js          | runtime        | 6.51 kB
---
Initial Total      |                | 2.28 MB

Build at: 2022-07-16T14:58:42.214Z - Hash: 29bd003ee47ba092 - Time: 17352ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
```

A janela acima mostra o resultado apresentado pelo comando "ng serve".

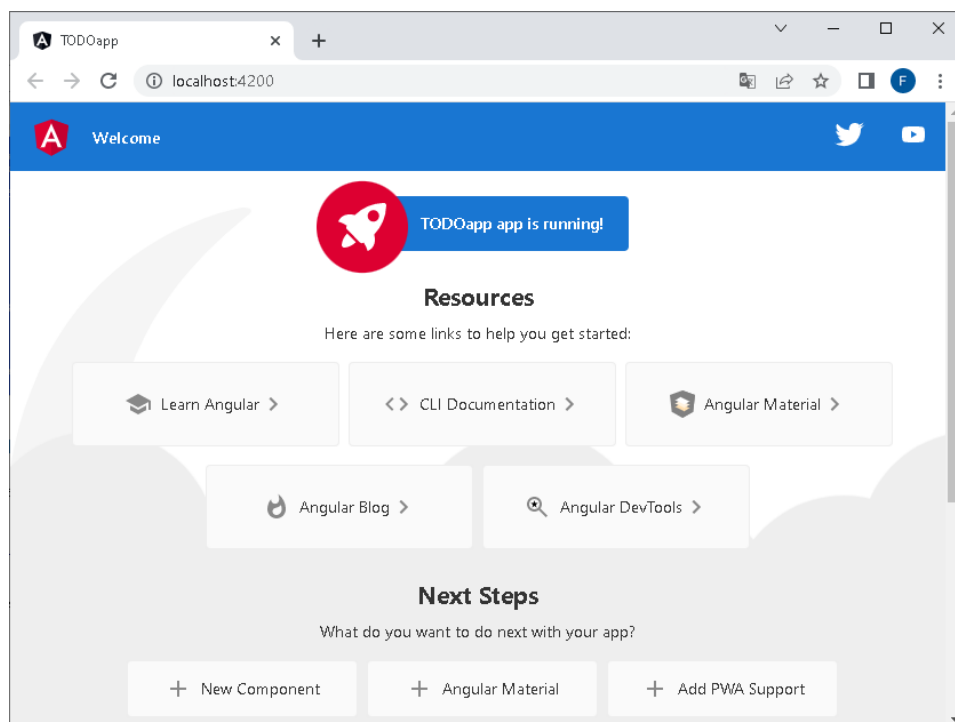
Deixe essa janela aberta enquanto estiver testando o seu App a partir do browser.

Testando o TODOapp localmente a partir do browser

Agora basta utilizar um browser de sua preferência para abrir o seguinte endereço:

<http://localhost:4200/>

O browser apresentará um App Angular **default** em execução. Ex:



O conteúdo acima corresponde a um esqueleto padrão de exemplo oferecido pelo Angular CLI durante a criação de um novo projeto.

Nosso objetivo agora é aproveitar a estrutura criada, e realizar as alterações necessárias para a construção de um App de acordo com as nossas necessidades.

Alterando a View do AppComponent

Agora vamos experimentar modificar o conteúdo apresentado pelo App. Abra um novo prompt de comando dentro da pasta TODOapp. Execute o VsCode utilizando o seguinte comando:

```
code .
```

Substitua todo o conteúdo do arquivo **app.component.html** pelas linhas abaixo:

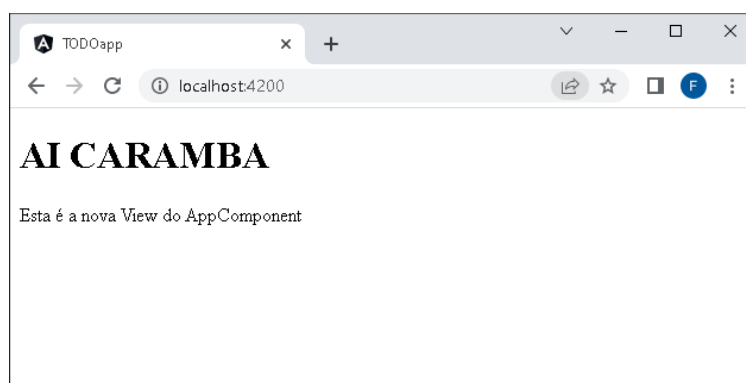
```
<h1>AI CARAMBA</h1>  
Esta é a nova View do AppComponent
```

Testando no Browser

Abra novamente o browser utilizando o mesmo endereço:

<http://localhost:4200/>

Resultado esperado:



Histórico de revisões

Revisão: 01
Data: 05/02/2023
Descrição das alterações:
 Adaptação para curso presencial

Revisão: 00
Data: 13/07/2022
Descrição das alterações:
 Documento original