

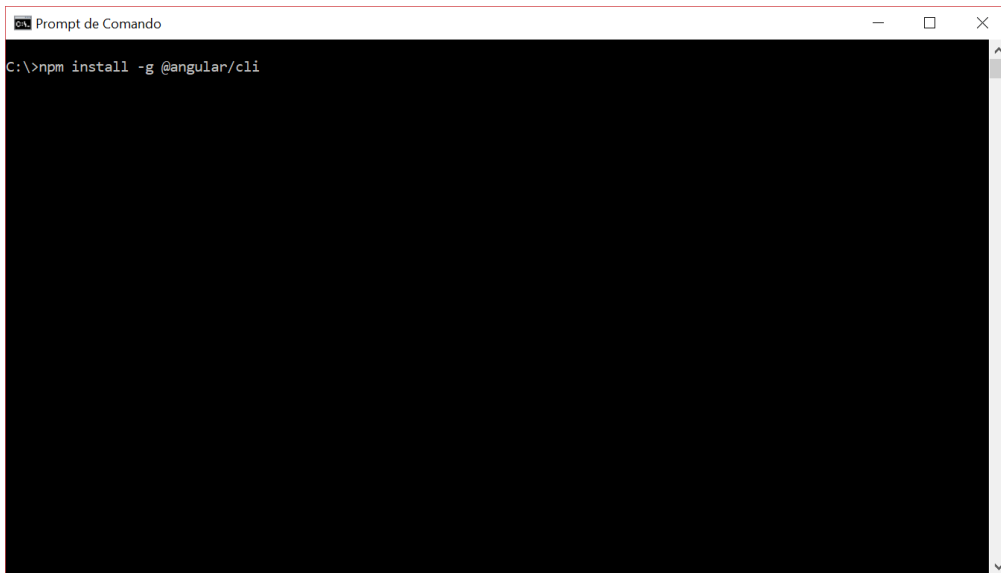
## Laboratório 7 – Introdução ao Angular

Este laboratório mostra como configurar um projeto inicial com o Angular CLI.

### 1 Ambiente

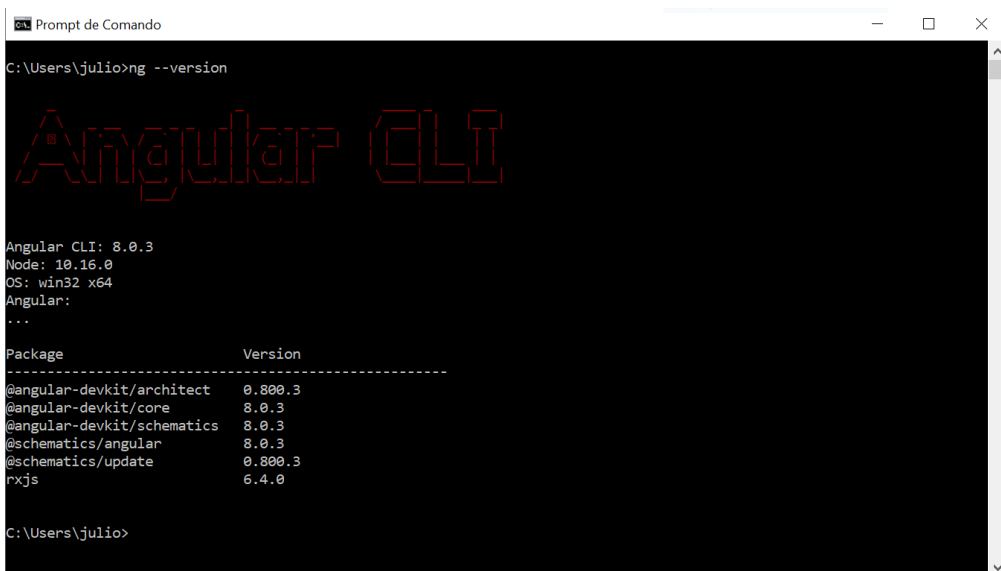
1. Abra um prompt da linha de comando e digite:

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



```
Prompt de Comando
C:\>npm install -g @angular/cli
```

```
ng --version
```



```
Prompt de Comando
C:\Users\julio>ng --version

Angular CLI
Angular CLI: 8.0.3
Node: 10.16.0
OS: win32 x64
Angular:
...
Package      Version
-----
@angular-devkit/architect    0.800.3
@angular-devkit/core        8.0.3
@angular-devkit/schematics   8.0.3
@schematics/angular         8.0.3
@schematics/update           0.800.3
rxjs                      6.4.0

C:\Users\julio>
```

2. Gere um novo projeto (aceite todos os valores padrão para a opções de criação do projeto pressionando “enter” a cada passo):

```
ng new my-app
```

3. Caso o NPM apresente mensagens de erro, utilize o seguinte comando e tente novamente:

```
npm cache verify
```

## **2 Estrutura da Aplicação**

1. Abra o subdiretório “my-app” no Visual Studio Code.
2. Observe as configurações do arquivo “tsconfig.json”. Procure entender as configurações conforme a documentação em <https://www.typescriptlang.org/docs/handbook/tsconfig-json.html>
3. Observe as configurações do arquivo “package.json”. Procure entender quais módulos foram adicionados por padrão pelo Angular CLI. Acesse <https://www.npmjs.com/> para procurar informações sobre os pacotes: zone.js, rxjs, codelyzer, jasmine-core, karma, protractor.
4. Observe as configurações do arquivo “angular.json”. Esse é o arquivo com as configurações a serem utilizadas na construção da aplicação Angular. Acesse <https://angular.io/guide/workspace-config> para entender as configurações padrão.
5. O código-fonte da aplicação Angular reside no subdiretório “src”. Os arquivos fora desse diretório suportam as configurações utilizadas pela ferramenta Angular CLI.
6. Abra o diretório “src/environments” e observe os dois arquivos. Estes arquivos devem ser utilizados para configurar valores (como strings de conexão a banco de dados, links para serviços web, etc) que são diferentes quando usamos a aplicação Angular em um ambiente de desenvolvimento, teste ou produção.
7. O diretório “src/assets” irá conter qualquer arquivo (como imagens, áudio, vídeo, etc) que deve ser copiado integralmente para o diretório de distribuição da aplicação.
8. Arquivo “src/index.html” é a página inicial da aplicação SPA no Angular. Apenas edite esse arquivo se for extremamente necessário. Em especial, evite adicionar referências a arquivos javascript ou css, pois as ferramentas do Angular CLI são responsáveis por isso.
9. Arquivo “src/styles.css” irá conter as definições de estilos globais da aplicação.
10. Arquivo “src/main.ts” é o ponto inicial da aplicação Angular.
11. Arquivo “src/polyfills.ts” possui as configurações necessárias para suportar diferentes navegadores. Observe que as opções para navegadores antigos estão todas desabilitadas no projeto padrão do Angular CLI. As opções de configuração estão em <https://angular.io/guide/browser-support>.

## **3 Execução da Aplicação**

1. Para executar o projeto em modo de desenvolvimento:

```
ng serve
```

2. Abra um navegador no endereço “http://localhost:4200”.

#### 4 Editando um Componente

1. Abra o arquivo “./src/app/app.component.ts” que corresponde ao elemento “<app-root>” no arquivo “./src/index.html”. Observe que ele é composto por um objeto definido em uma classe, um template para a view HTML e arquivos de definição do CSS.
2. Edite a propriedade “title” da classe “AppComponent” para “Minha Primeira Aplicação Angular”. Observe no arquivo “./src/app/app.component.html” como o template referenciou o valor da propriedade.
3. Edite o arquivo “./src/app/app.component.css” com o seguinte estilo CSS:

```
h1 {  
  color: #369;  
  font-family: Arial, Helvetica, sans-serif;  
  font-size: 250%;  
}
```

4. Observe o resultado no navegador.

#### 5 Empacotando a Aplicação

1. Para empacotar a aplicação para distribuição em modo de desenvolvimento:

```
ng build
```

2. Observe o conteúdo do diretório “./dist”. A ferramenta utilizada para o empacotamento é o <https://webpack.js.org/>. Em especial abra o arquivo “./dist/index.html” e observe como os arquivos javascript que implementam a aplicação Angular foram referenciados.
3. Para empacotar a aplicação para distribuição em modo de produção:  

```
ng build --prod
```
4. Observe o conteúdo do diretório “./dist”. Qual diferença em relação à distribuição de desenvolvimento?

#### 6 Implantando a Aplicação

1. Para implantar a aplicação no Azure da Microsoft como um site estático, siga os passos indicados em <https://code.visualstudio.com/tutorials/static-website/getting-started>. Ao seguir o tutorial, ignore os passos de criação de uma aplicação, já que a aplicação Angular já foi construída. No passo de *deploy*, escolha o diretório da sua aplicação dentro do diretório “dist” para ser implantado no Azure.
2. Para implantar uma aplicação no Azure da Microsoft como um site dinâmico baseado em Node.js, siga os passos indicados em <https://code.visualstudio.com/tutorials/app-service-extension/getting-started>. Não é o caso para a nossa aplicação Angular, mas seria o caso para a implementação do backend de serviços que nossa aplicação iria utilizar.
3. Para implantar uma aplicação no Firebase da Google, siga os passos indicados em <https://angular.io/start/deployment>.