

DESENVOLVIMENTO DE
APLICAÇÕES WEB COM
ANGULAR

Cap. de Angular

Cap. de Amostra

WILLIAM PEREIRA ALVES

DESENVOLVIMENTO DE APLICAÇÕES WEB COM ANGULAR

Fundamentos de Angular • Programação em Type Script
Criação de Web Service PHP • Uso do Framework Bootstrap
Validação de Dados e uso de Filtros • Funções Assíncronas



ALTA BOOKS
EDITORA
Rio de Janeiro, 2019

Cap. de Amostra

Sumário

Sobre o Autor	viii
Apresentação	ix
Capítulo 1: Do AngularJS ao Angular 6	1
Capítulo 2: Instalação e Configuração do Apache	17
Capítulo 3: Introdução ao AngularJS	23
Capítulo 4: Diretiva ngOptions e Framework Bootstrap	41
Capítulo 5: Validação de Dados e Uso de Filtros	65
Capítulo 6: Modularização do Código	105
Capítulo 7: Introdução à Linguagem TypeScript	121
Capítulo 8: Funções	139
Capítulo 9: Orientação a Objetos com TypeScript	153
Capítulo 10: Decoradores, Classes Genéricas e Funções Assíncronas	183
Capítulo 11: Primeiros Passos com Angular 6	211
Capítulo 12: Criação de Projetos com IDE	227
Capítulo 13: Fundamentos de Angular 6	245
Capítulo 14: Criação de Exemplo Prático	255
Capítulo 15: Instalação do MySQL e do PHP	269
Capítulo 16: Criação de Web Service com PHP	287
Capítulo 17: AngularJS com Web Service	305
Capítulo 18: Angular 6 com Web Service	325
Capítulo 19: Conclusão	351
Apêndice	353
Respostas dos Exercícios	355

Aviso

Para melhor entendimento as figuras coloridas estão disponíveis no site da editora Alta Books. Acesse: www.altabooks.com.br e procure pelo nome do livro ou ISBN.

Cap. de Amostra

Dedicatória

Este livro é dedicado com muito amor a três pessoas de grande importância na minha vida: minha querida esposa Lucimara e meus filhos Brian e Liam.

Também quero dedicá-lo aos meus pais, meus irmãos e aos demais familiares meus e da minha esposa.

E, por fim, uma dedicação especial a meus avós e meus sogros, pessoas queridas que já partiram.

Cap. de Amostra

Agradecimento

Desejo expressar meus sinceros e enormes agradecimentos ao pessoal da Editora Alta Books pela oportunidade que me foi oferecida para a realização deste trabalho.

Aos meus amigos e professores do curso de Análise e Desenvolvimento de Sistemas do Centro Universitário Claretiano de São Paulo e aos meus novos colegas de trabalho.

Sobre o Autor

William Pereira Alves é formado em Análise e Desenvolvimento de Sistemas pelo Centro Universitário Claretiano de São Paulo. Autor de diversos livros sobre computação, desde 1992, contando com diversas obras já publicadas, que abrangem as áreas de linguagens de programação (Delphi, C/C++, Java, Visual Basic, PHP), bancos de dados (Access), computação gráfica (CorelDRAW, Illustrator e Blender), desenvolvimento de sites (Dreamweaver, Flash e Fireworks) e de aplicações para dispositivos móveis (Palm e smartphone/tablet Android).

Atuando na área de informática desde 1985, trabalhou na Cia. Energética de São Paulo (CESP) e na Eletricidade e Serviços S.A. (Elektro) no desenvolvimento de sistemas aplicativos para os departamentos comercial e de suprimento de materiais, inclusive com a utilização de coletores de dados eletrônicos e leitura de códigos de barras.

Também foi responsável por todo o projeto e desenvolvimento do sistema de gestão da Editora Érica, entre 2007 e 2015. Atualmente trabalha no departamento de TI da Leonard, empresa de engenharia civil especializada em concreto pré-fabricado.

Apresentação

Entre as inúmeras opções de frameworks JavaScript existentes atualmente no mercado, o Angular tem obtido grande aceitação, e também se destacou, em função dos recursos oferecidos para agilizar o desenvolvimento de aplicações para web.

Em meio a esse mar de opções, o desenvolvedor se vê, de alguma forma, perdido na hora de escolher a ferramenta que melhor atende suas necessidades.

Como forma de compartilhar os conhecimentos que adquiri a respeito dessa grande ferramenta, e assim poder subsidiar uma futura decisão por parte do desenvolvedor quanto ao framework a ser adotado em seus projetos, me dispus a escrever este livro, trabalho que foi um verdadeiro desafio, em função de diversos contratemplos que surgiram no meio do caminho. Mas, enfim, aqui está o resultado final.

Assim como em outros trabalhos que já desenvolvi, procurei demonstrar os conceitos e técnicas empregadas no desenvolvimento com Angular por meio de projetos de aplicações concretas em vez de simplesmente apresentar fragmentos de códigos isolados, que poderiam tornar o aprendizado mais difícil.

A obra também não se restringe apenas ao Angular JS, mas tem como foco principal o estudo do Angular 6. Em função disso, são dedicados alguns capítulos à introdução da linguagem de programação TypeScript, empregada pelo Angular desde a versão 2. Ao oferecer cobertura das duas versões, os leitores poderão conhecer os dois mundos. Mas certamente escolherão a versão mais recente, tendo em vista ser mais robusta e oferecer recursos mais avançados de desenvolvimento.

Espero que o amigo leitor esteja pronto para iniciar a jornada.

Bons estudos!
O autor

Cap. de Amostra

1

Do AngularJS ao Angular 6

Neste capítulo inicial temos uma breve visão do que é um framework e uma apresentação do AngularJS, com descrição dos passos necessários para efetuar o download do pacote. O capítulo também apresenta uma pequena comparação com o Angular 6, a mais recente versão desse framework.

Ainda é abordado o processo de instalação do Angular 6, tanto no Windows quanto no Ubuntu (uma distribuição Linux), e uma descrição do conceito por trás do padrão de projeto de software MVC.

E, para finalizar, é descrito o conceito de aplicações que fazem uso de uma técnica conhecida como página única (*Single-Page Applications*).

1.1 Frameworks e AngularJS

Podemos entender framework como uma sofisticada estrutura que abriga códigos de programação para oferecer ao programador inúmeros recursos capazes, por sua vez, de agilizar o desenvolvimento e a manutenção dos aplicativos. É uma evolução das antigas bibliotecas de funções e componentes disponíveis para uso em diversas linguagens de programação, como C, C++, Visual Basic ou Delphi.

Esses recursos compreendem funções, objetos e métodos que tornam possível executar determinadas operações de forma automatizada, sem a necessidade de escrita de longos códigos para obter o mesmo resultado.

Suponha, como exemplo, a criação de um sistema de menu hierárquico para uma aplicação. Utilizando o método tradicional, o programador precisa escrever todo o código responsável por essa criação, ou seja, deve utilizar as instruções da linguagem para definir a posição de cada opção na tela e as ações a serem executadas. Com o uso de um framework, tudo que o programador precisa fazer é utilizar o recurso oferecido por ele, como um objeto de construção de menus, passando alguns parâmetros.

Frameworks não são empregados apenas no desenvolvimento de aplicações web. Programadores Java, C#, C++ e de outras linguagens também utilizam, frequentemente, algum framework, em especial no desenvolvimento de aplicações gráficas e jogos.

Diferentemente das bibliotecas de funções ou de classes, o framework é o responsável por ditar as regras que controlam o fluxo de execução da aplicação.

AngularJS é um dos frameworks JavaScript mais utilizados no desenvolvimento de aplicações web. Seu projeto teve início em 2009, pelas mãos de um engenheiro do Google chamado Misko Hevery, que na época fazia parte de um grupo que estava envolvido em outro projeto denominado Google Feedback. Esse projeto consumiu muito tempo e resultou em uma enorme quantidade de código (cerca de 17.000 linhas), o que não agradou Misko e o levou a reescrever tudo por meio da criação de um framework para uso próprio, resultando no AngularJS.

O Google, ao perceber o potencial da nova ferramenta, ofereceu total apoio, tornando-a *open source* sob a licença MIT. Isso significa que podemos utilizá-la livremente, sem qualquer pagamento de royalties, além da possibilidade de contribuir com a sua evolução.

O AngularJS trabalha dentro do padrão de projeto MVC (*Model-View-Controller*), ou seja, uma aplicação web desenvolvida com esse framework possui uma arquitetura bem definida, com separação distinta das camadas de visualização, de modelagem dos dados e de controle. Ele é direcionado ao desenvolvimento de aplicações de página única (*single-page*), um conceito que veremos em maiores detalhes mais à frente.

As primeiras versões do AngularJS fizeram muito sucesso junto à comunidade de desenvolvedores de aplicações web. No entanto, em função de algumas de suas deficiências, foi criada a versão 2, denominada Angular2 (sem o sufixo JS). Essa versão trouxe grandes mudanças, até mesmo estruturais. Entre elas, as que mais se destacam são o suporte a ECMAScript6 (ou ECMAScript 2015) e TypeScript, que possibilitam obter um código JavaScript mais claro e bem escrito.

Nota

ECMAScript é uma linguagem de script para internet que foi a base para o desenvolvimento das linguagens JavaScript (Netscape), JScript (Microsoft) e ActionScript (Adobe).

TypeScript é uma linguagem de script open source criada pela Microsoft.

Ainda é possível desenvolver aplicações web com AngularJS (versões 1.x), mas pouco recomendável, uma vez que não há garantias de que elas possam ser mantidas por longo tempo. A última versão estável, disponível na época de produção deste livro, era a de número 1.7.2.

Para baixar e utilizar essa versão, acesse o endereço angularjs.org (conteúdo em inglês) e clique no botão **DOWNLOAD ANGULARJS** da página mostrada pela Figura 1.1. Uma janela é apresentada para seleção da versão desejada e o tipo de arquivo (Figura 1.2). Nesta tela também há instruções para se utilizar o framework a partir de um repositório CDN ou fazer a instalação manual por meio das ferramentas Bower e Npm.



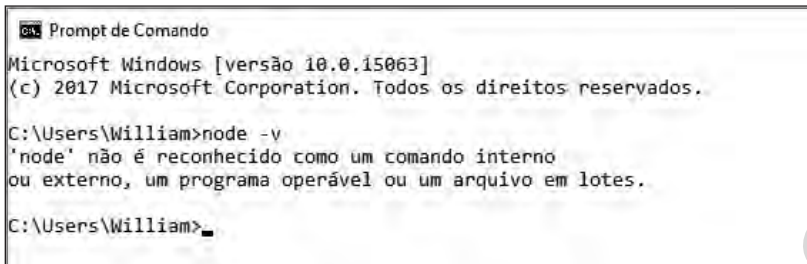
Figura 1.1 - Página inicial para baixar o framework AngularJS.



Figura 1.2 - Versões disponíveis para download do AngularJS.

É indicado baixar a versão em formato ZIP pelo fato de no arquivo virem, além do próprio framework, diversos módulos complementares, como biblioteca de animação, sanitização de URLs, gerenciamento de cookies, entre outros.

Outra opção é acessar a ferramenta **PowerShell** ou o prompt de comando do Windows e digitar o comando **node -v**. Se o Node.js estiver instalado, será exibida a versão, caso contrário, aparecerá uma mensagem similar à da Figura 1.4.



```

CA: Prompt de Comando
Microsoft Windows [versão 10.0.15063]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.

C:\Users\William>node -v
'node' não é reconhecido como um comando interno
ou externo, um programa operável ou um arquivo em lotes.

C:\Users\William>
  
```

Figura 1.4 - Mensagem exibida se o Node.js não estiver instalado.

Caso precise instalá-lo, acesse o endereço *nodejs.org* (conteúdo em inglês) e, na tela da Figura 1.5, clique na opção **Downloads**. Na tela apresentada em seguida (Figura 1.6), clique na opção **32-bit** ou **64-bit**, apresentadas ao lado do item **Windows Installer (.msi)**. Após ter sido finalizado o download do arquivo, execute-o para iniciar a instalação. A tela da Figura 1.7 é mostrada. Clique no botão **Next** e depois aceite os termos de uso e clique novamente em **Next** (Figura 1.8). Siga as orientações dadas nas próximas telas (Figuras 1.9 a 1.10) para concluir a instalação.

É preciso deixar claro que, para rodar o Angular 6, os requisitos mínimos são Node.js versão 6.11 e NPM versão 3.10.

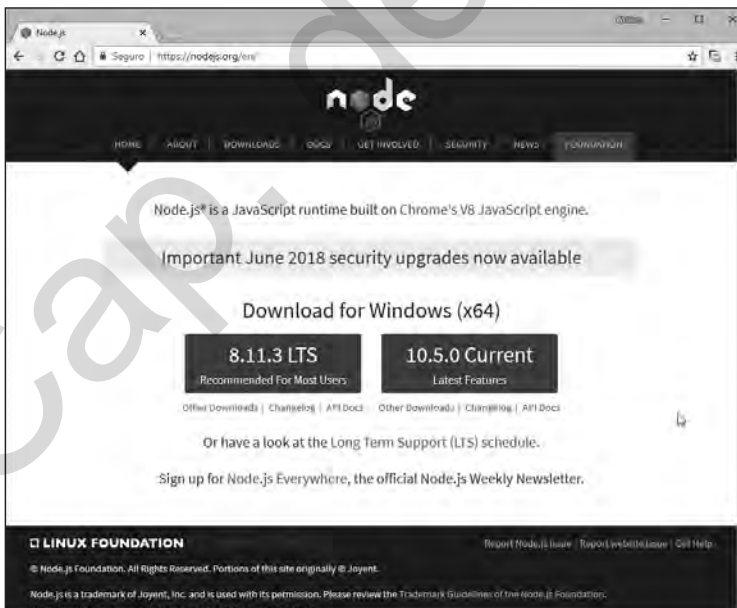


Figura 1.5 - Página de abertura do site do Node.js.

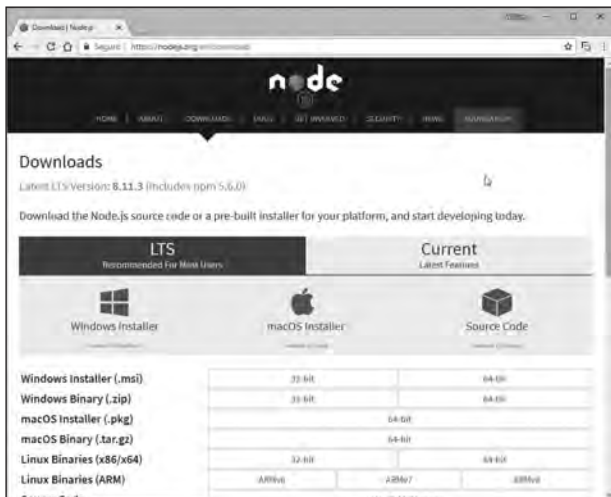


Figura 1.6 - Opções de instalação do Node.js.

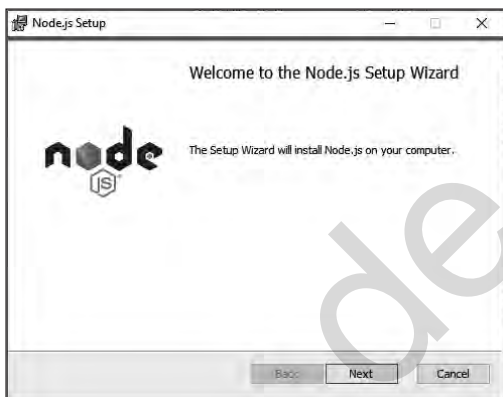


Figura 1.7 - Tela inicial do instalador do Node.js.

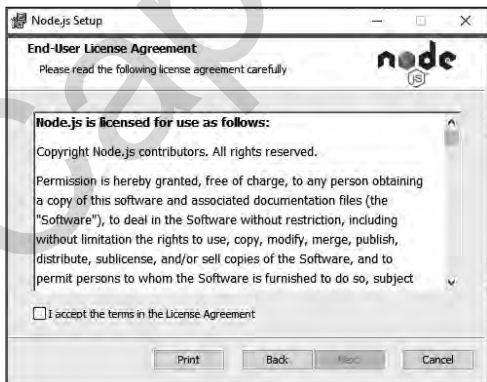


Figura 1.8 - Tela de aceitação dos termos de licença.

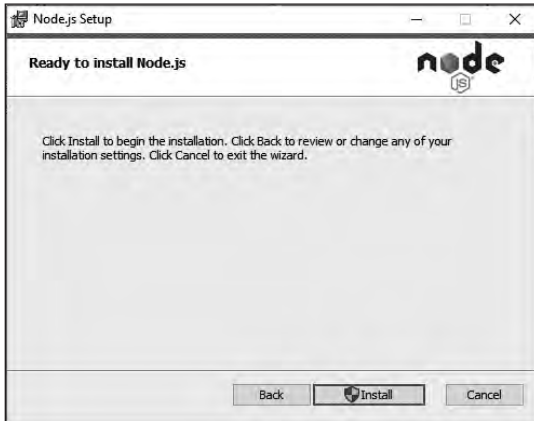


Figura 1.9 - Tela de confirmação da instalação do Node.js.

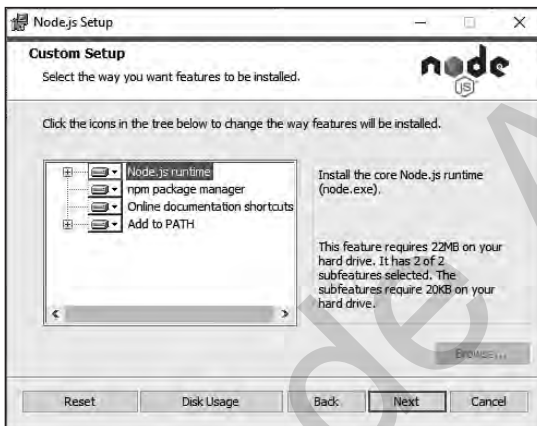


Figura 1.10 - Tela de configuração dos itens a serem instalados.

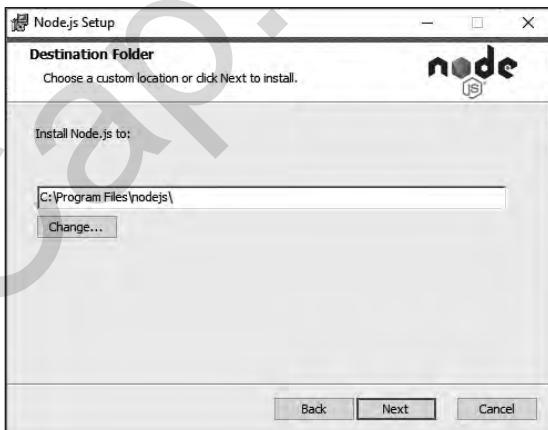


Figura 1.11 - Tela de especificação do local de instalação do Node.js.



Figura 1.12 - Tela de finalização do processo de instalação.

Tendo concluído a instalação, reinicie o sistema e, quando ele estiver pronto, acesse novamente o prompt de comando do Windows ou o **PowerShell** e execute outra vez o comando **node -v**. Deve ser apresentada uma mensagem com a versão do Node.js (Figura 1.13). Você também pode verificar a versão do gerenciador de pacotes NPM por meio do comando **npm -v** (Figura 1.14).

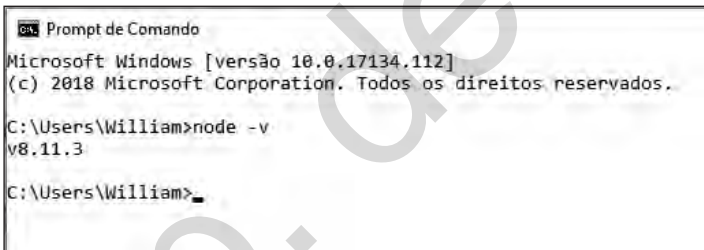


Figura 1.13 - Comando para visualização da versão do Node.js.

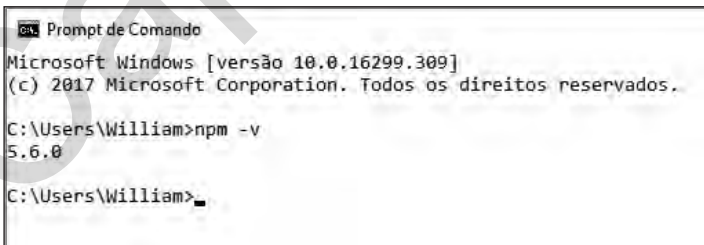
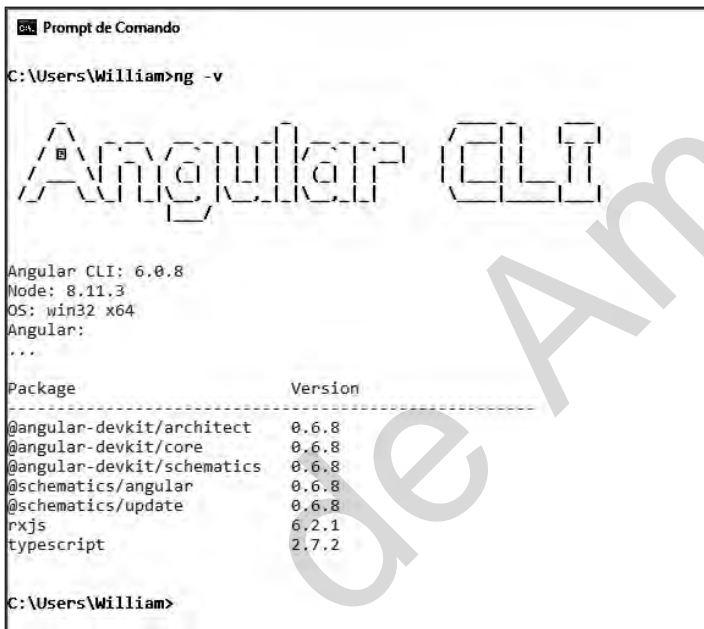


Figura 1.14 - Comando para visualização da versão do gerenciador de pacotes NPM.

Agora que já temos o Node.js instalado, assim como o gerenciador NPM, chegou a hora de instalar o Angular CLI, que é a ferramenta de linha de comando do Angular 6 responsável por facilitar o gerenciamento de projetos. Ainda com a janela do prompt de comando ou do **PowerShell** aberta, digite o comando `npm install -g @angular/cli` e tecla [ENTER] para executá-lo. Se já houver uma versão anterior instalada, ela será atualizada automaticamente.

Após ter finalizado o processo de instalação ou atualização, execute o comando `ng -v` para ver se a instalação foi bem sucedida. Em caso afirmativo, deve ser mostrada uma mensagem similar à da Figura 1.15, contendo a versão do Angular.



```

C:\Users\William>ng -v

Angular CLI
-----
Angular CLI: 6.0.8
Node: 8.11.3
OS: win32 x64
Angular:
...

Package          Version
-----
@angular-devkit/architect    0.6.8
@angular-devkit/core        0.6.8
@angular-devkit/schematics   0.6.8
@schematics/angular         0.6.8
@schematics/update          0.6.8
rxjs                       6.2.1
typescript                2.7.2

C:\Users\William>

```

Figura 1.15 - Mensagem com informação da versão do Angular CLI.

A instalação no Linux não difere muito. É apresentado aqui o processo em uma distribuição Ubuntu, versão 17.10, utilizando seu repositório de pacotes, mas pode ser empregado em qualquer distribuição baseada no Debian. Para distribuições que utilizam outras ferramentas de gerenciamento de pacotes, é necessário consultar o manual do usuário.

Abra uma sessão de **Terminal de Console** (Figura 1.16) e verifique se o Node.js está instalado por meio do comando `node -v`. Em caso negativo, deve-se verificar também se o utilitário `curl` está instalado, uma vez que ele será utilizado na criação de um arquivo de script para instalação do Node.js. Para efetuar essa verificação, digite na janela do terminal a expressão `curl` e tecla [ENTER]. Se for exibida a mensagem da Figura 1.17, significa que ele precisa ser instalado.

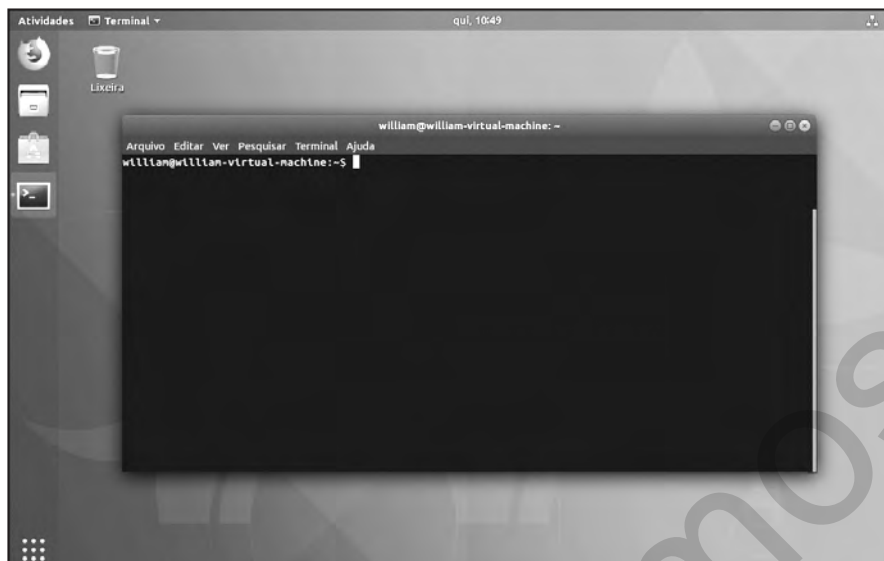


Figura 1.16 - Tela do Terminal de Console do Ubuntu Linux.

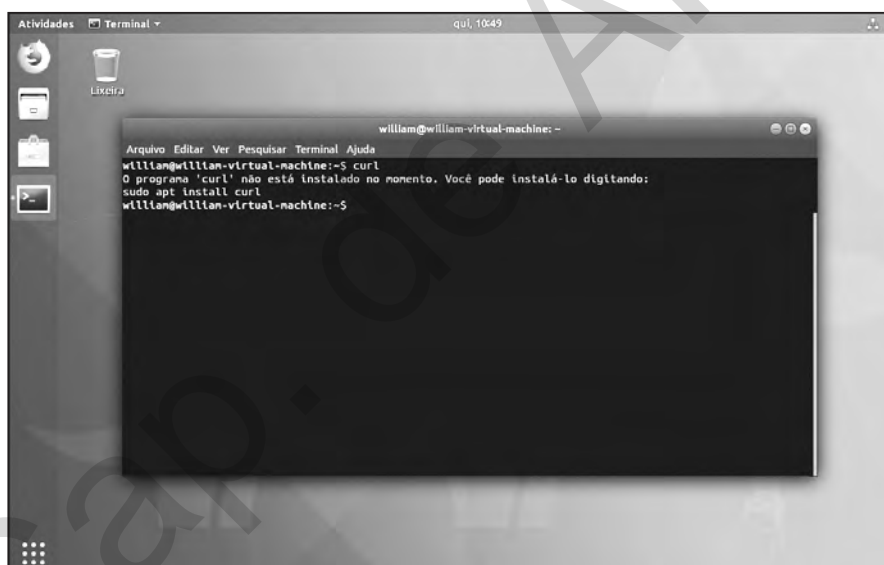


Figura 1.17 - Verificação se a ferramenta curl está instalada.

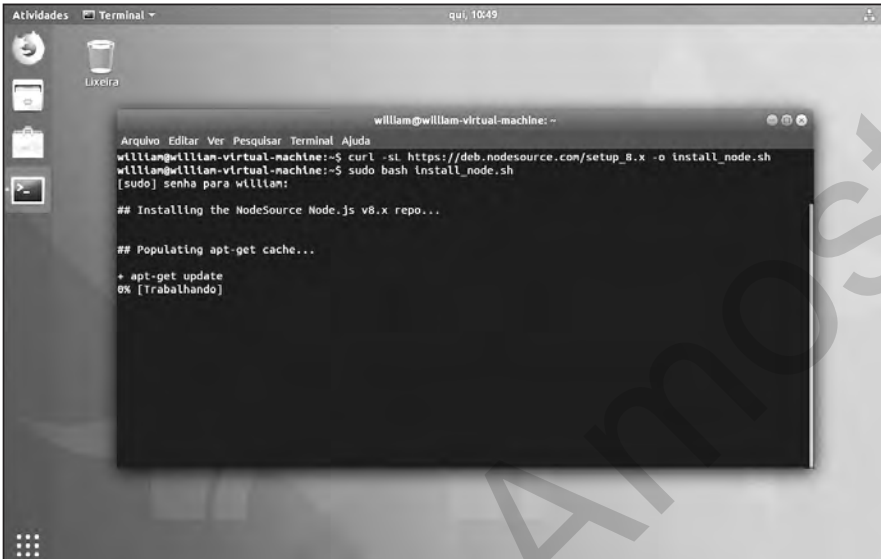
A instalação é efetuada com a execução do comando **sudo apt-get curl**.

Em seguida, execute os comandos apresentados pela seguinte listagem:

```
curl -sL https://deb.nodesource.com/setup_8.x -o install_node.sh
sudo bash install_node.sh
```

Veja a Figura 1.18. Ao fim da execução do último comando, você deverá ver uma tela similar à da Figura 1.19. Execute, ainda, o seguinte comando:

```
sudo apt-get install nodejs
```



```

william@william-virtual-machine: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
william@william-virtual-machine:~$ curl -sL https://deb.nodesource.com/setup_8.x -o install_node.sh
william@william-virtual-machine:~$ sudo bash install_node.sh
[sudo] senha para william:
## Installing the NodeSource Node.js v8.x repo...

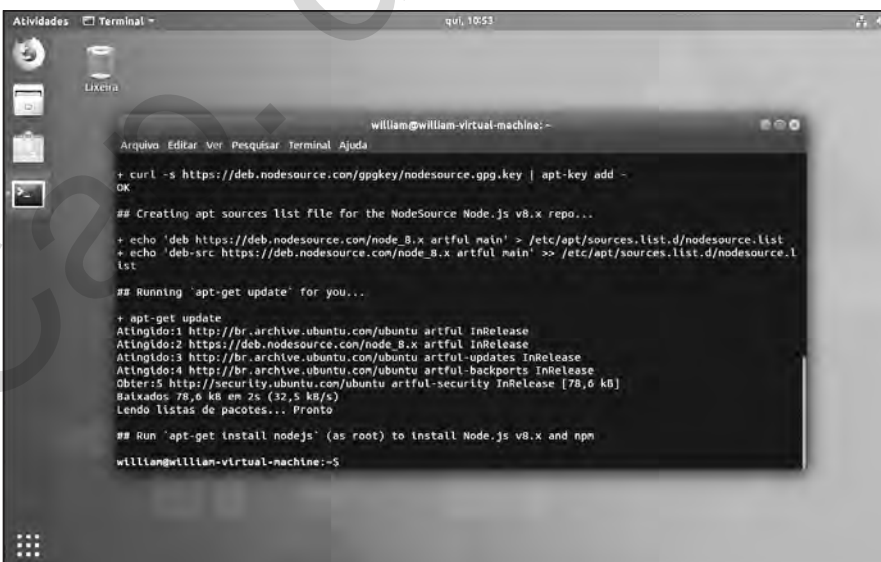
## Populating apt-get cache...

+ apt-get update
0% [Trabalhando]

```

Figura 1.18 - Execução de comandos para instalação do Node.js.

Quando for finalizada a instalação, execute os comandos `node -v` e `npm -v`, com os quais você poderá verificar a versão de ambos e se eles foram instalados corretamente. Veja a Figura 1.20.



```

william@william-virtual-machine: ~
Arquivo Editar Ver Pesquisar Terminal Ajuda
+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | apt-key add -
OK
## Creating apt sources list file for the NodeSource Node.js v8.x repo...
+ echo 'deb https://deb.nodesource.com/node_8.x artful main' > /etc/apt/sources.list.d/nodesource.list
+ echo 'deb-src https://deb.nodesource.com/node_8.x artful main' >> /etc/apt/sources.list.d/nodesource.l
ist
## Running 'apt-get update' for you...
+ apt-get update
Atingido:1 http://br.archive.ubuntu.com/ubuntu artful InRelease
Atingido:2 https://deb.nodesource.com/node_8.x artful InRelease
Atingido:3 http://br.archive.ubuntu.com/ubuntu artful-updates InRelease
Atingido:4 http://br.archive.ubuntu.com/ubuntu artful-backports InRelease
Obter:5 http://security.ubuntu.com/ubuntu artful-security InRelease [78,6 kB]
Baixados 79,6 kB em 7s (32,5 kB/s)
Lendo listas de pacotes... Pronto
## Run 'apt-get install nodejs' (as root) to install Node.js v8.x and npm
william@william-virtual-machine:~$

```

Figura 1.19 - Instalação do Node.js concluída.

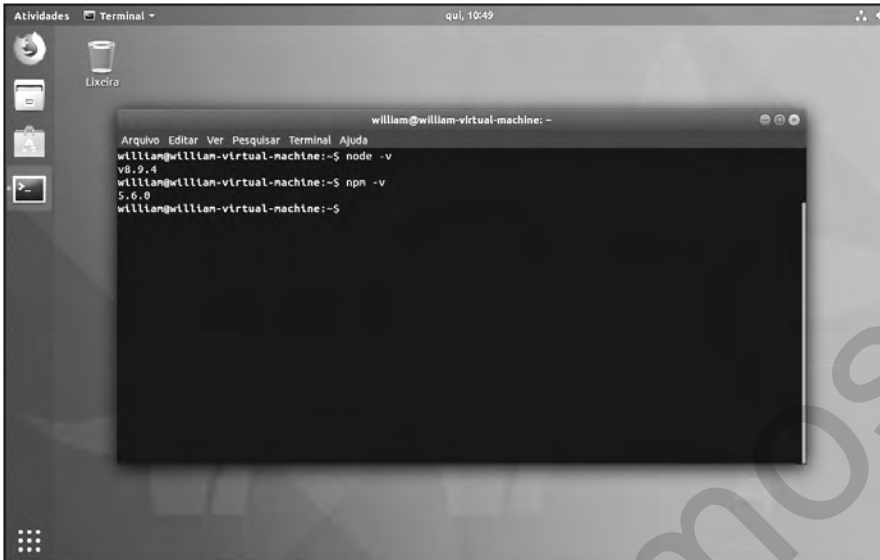


Figura 1.20 - Verificação da versão do Node.js e do gerenciador NPM.

Para instalar o Angular CLI no Ubuntu, deve ser executado o comando **sudo npm install -g @angular/cli** na janela do terminal.

Na Figura 1.21 podemos ver a mensagem de confirmação do Angular CLI instalado com sucesso.

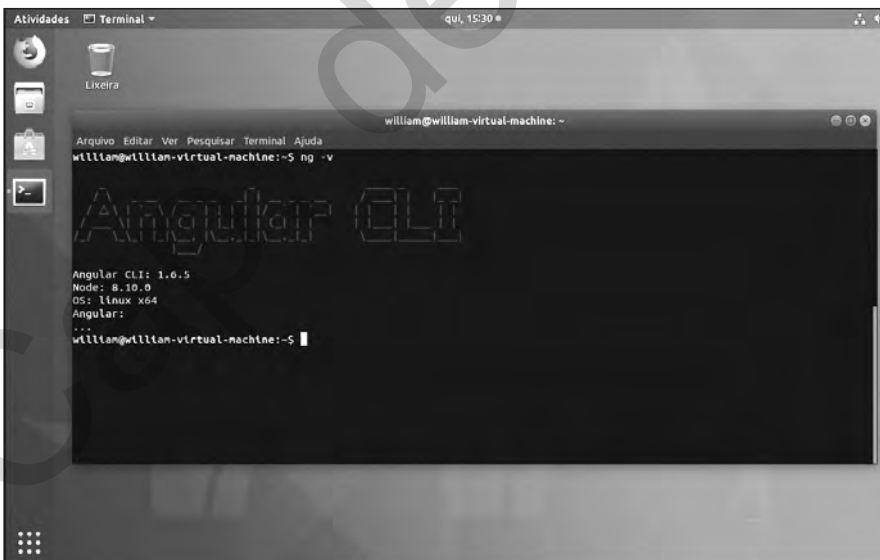
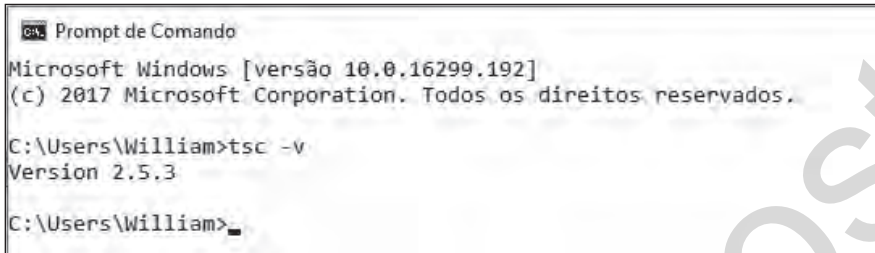


Figura 1.21 - Mensagem com informação da versão do Angular CLI.

Uma vez que o Angular 6 faz uso da linguagem TypeScript, é preciso que ela também esteja instalada na máquina. A versão mais recente da linguagem TypeScript, na época de produção deste livro, era a 2.9.2. Você pode verificar qual a versão instalada executando o comando `tsc -v` no prompt de comando do Windows ou na tela de terminal do Linux, como mostra o exemplo da Figura 1.22.

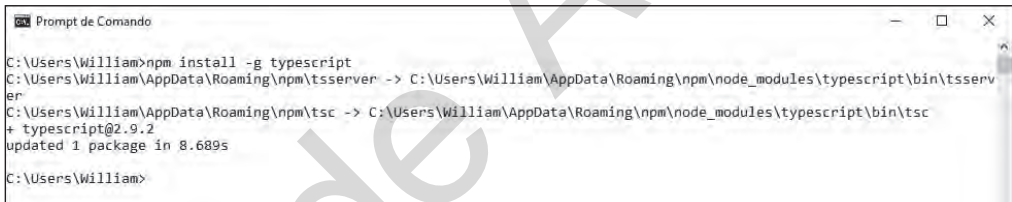


```

C:\Users\William>tsc -v
Version 2.5.3
C:\Users\William>
  
```

Figura 1.22 - Verificação da existência da linguagem TypeScript.

Para instalar a versão mais recente, ou atualizar a que se encontra instalada atualmente, execute o comando `npm install -g typescript` no Windows, ou o comando `sudo apt install node-typescript`, no Ubuntu. Veja o exemplo da Figura 1.23.



```

C:\Users\William>npm install -g typescript
C:\Users\William\AppData\Roaming\npm\tsserver -> C:\Users\William\AppData\Roaming\npm\node_modules\typescript\bin\tsserver
C:\Users\William\AppData\Roaming\npm\tsc -> C:\Users\William\AppData\Roaming\npm\node_modules\typescript\bin\tsc
+ typescript@2.9.2
updated 1 package in 8.689s
C:\Users\William>
  
```

Figura 1.23 - Instalação/atualização da linguagem TypeScript.

Com todos esses procedimentos efetuados, estamos prontos para criar nossos projetos com o Angular 6. A criação de um projeto envolve a instalação do próprio framework e a definição de várias pastas e arquivos necessários ao funcionamento do Angular 6. Esse processo será abordado no próximo capítulo.

1.3 O padrão MVC

MVC (sigla de *Model-View-Controller*) é um padrão de projeto de software que tem por objetivo nortear o processo de separação do código da aplicação de acordo com a funcionalidade e/ou objetivo, levando a um agrupamento em categorias, conhecidas na engenharia de software como camadas.

Pretende-se, com isso, isolar partes do código de modo que elas sejam as mais independentes possíveis, porém com um alto grau de conectividade. Isso significa que um

objeto pode existir sem a necessidade de outro, mas eles podem ser vinculados de maneira bastante simplificada, por meio de uma interface bem definida.

Em termos práticos, a separação em camadas oferece um ganho enorme em relação à facilidade e redução de tempo nos processos de manutenção do sistema.

Essa divisão também auxilia muito a execução de testes unitários, uma vez que se pode testar uma função ou um objeto sem que haja necessidade de outros componentes externos já terem sido definidos/implementados.

Na camada *Model* (Modelo), temos os códigos responsáveis pela modelagem ou representação da estrutura lógica dos dados que o sistema deverá manipular. É importante não confundir com o processo de modelagem de dados empregado na área de banco de dados.

A camada *View* (Visão) compreende a parte do sistema que representa a interface com o usuário e que engloba, na maioria das vezes, a exibição dos dados armazenados em memória ou em um banco de dados em disco (via modelo da camada *Model*).

Por fim, a camada *Controller* (Controlador) funciona como uma ponte entre as camadas *Model* e *View*. Nela, os códigos agem de acordo com os eventos ocorridos na *View* ou na *Model*. Tecnicamente falando, essa camada consiste em rotinas (métodos ou funções) que leem dados da *View* e atualizam as estruturas correspondentes na camada *Model*. No lado inverso, os dados são lidos na *Model* e exibidos na camada *View*. Isso significa que *View* e *Model* nunca interagem diretamente entre si.

A Figura 1.24 apresenta um diagrama que ilustra bem o relacionamento existente entre essas três camadas dentro de um sistema.

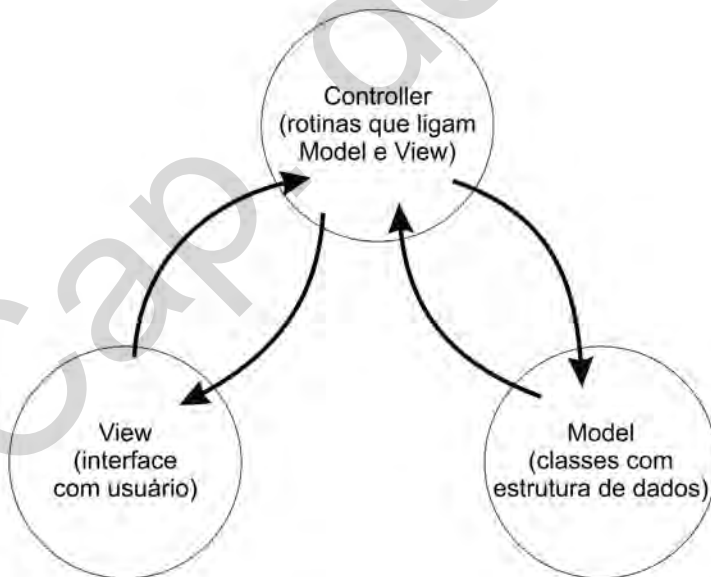


Figura 1.24 - Diagrama do relacionamento entre camadas do padrão MVC.

1.4 Aplicações de página única

Conforme mencionado anteriormente, o framework Angular tem sido empregado principalmente no desenvolvimento de aplicações que fazem uso de um conceito denominado *Single-Page Applications* (SPA).

Esse tipo de aplicação web se caracteriza por apresentar uma tela que corresponde à página principal, com as demais páginas sendo carregadas dentro dela, em áreas específicas, em vez de serem carregadas por inteiro de maneira a sobrepor a página principal. Isso imita o comportamento de aplicações para desktop ou cliente/servidor, criadas com linguagens de programação como C++, Java, C# ou Object Pascal (Delphi).

A forma como as páginas visualizadas pelos usuários são construídas em uma SPA difere consideravelmente do método tradicional empregado em websites regulares. Nesses últimos, geralmente construídos com tecnologias *server-side*, como Java, PHP ou ASP.NET, o servidor web recupera os dados de um banco de dados e então gera a página HTML que deve ser enviada ao navegador do usuário. Em outras palavras, ele executa todo o trabalho pesado, o que pode resultar em lentidão se houver muitas conexões de usuários ao mesmo tempo.

Com a arquitetura SPA, toda a parte do processamento relacionada com a construção da página em si é delegada ao navegador do usuário. O servidor web tem apenas a função de fornecer os dados requisitados, não sendo, portanto, sobrecarregado, mesmo com uma grande quantidade de requisições a serem atendidas.

No próximo capítulo, veremos como instalar e configurar o servidor web Apache no Windows e no Ubuntu.

Exercícios

1. Defina o conceito de framework.
2. Quais são os requisitos mínimos para se instalar o Angular CLI?
3. Qual é o comando que permite verificar a versão instalada do node, caso ele esteja?
4. Cite o comando que deve ser executado para a instalação do Angular CLI.
5. O que são aplicações de página única (Single-Page Applications)?
6. Explique o conceito de cada uma das camadas no padrão MVC.