

Curso de Jogos Digitais

Disciplina de Tecnologias Web

Aula 10 – Ferramentas e Frameworks
Instalação do Framework Angular

Professor: André Flores dos Santos

Introdução

- O que é o Framework Angular?
- Angular é um framework de desenvolvimento de aplicativos web e móveis, desenvolvido e mantido pelo Google.
- Ele permite criar aplicativos web escaláveis, de alto desempenho e com uma estrutura robusta baseada em componentes.



• URL: angular.io

• URL: material.angular.io

Introdução

- **Principais recursos do Angular:**

- Arquitetura baseada em componentes: Os aplicativos Angular são construídos a partir de componentes reutilizáveis, cada um representando uma parte da interface do usuário.
- Data binding bidirecional: Angular suporta data binding bidirecional, o que significa que as alterações no ‘modelo’ são refletidas automaticamente na ‘visão’ e vice-versa.
 - Visão para Modelo: Quando a interface do usuário é atualizada (por exemplo, quando um usuário digita em um campo de formulário), o modelo de dados correspondente é automaticamente atualizado.



Introdução

• Exemplo Prático

- Vamos usar um exemplo simples com um campo de entrada de texto (input).

```
// Arquivo TypeScript do componente  
(exemplo.component.ts)  
  
import { Component } from '@angular/core';  
  
@Component({  
  selector: 'app-exemplo',  
  templateUrl: './exemplo.component.html',  
  styleUrls: ['./exemplo.component.css']  
})  
export class ExemploComponent {  
  nome: string = 'John Doe';  
}
```

Explicação:

Modelo para Visão:

No exemplo acima, o valor inicial do modelo nome é "John Doe". Esse valor é exibido no campo de entrada de texto e no parágrafo abaixo dele. Se o valor de nome for alterado programaticamente no TypeScript, por exemplo, this.nome = 'Jane Doe'; essa alteração será automaticamente refletida tanto no campo de entrada quanto no parágrafo.

Visão para Modelo:

Quando o usuário digita algo no campo de entrada de texto, o valor de nome é atualizado automaticamente.

Conforme o usuário digita, o parágrafo abaixo do campo de entrada mostra o valor atualizado instantaneamente, refletindo a mudança no modelo de dados.

```
<!-- Arquivo HTML do componente  
(exemplo.component.html) -->  
<input [(ngModel)]="nome" placeholder="Digite  
seu nome">  
<p>Nome: {{ nome }}</p>
```



Introdução

- **Principais recursos do Angular:**

- **Injeção de dependência:** Angular possui um sistema de injeção de dependência integrado que facilita a modularização, reutilização e teste de componentes.
- **Roteamento:** O roteador Angular permite navegação entre diferentes partes do aplicativo e carregamento dinâmico de componentes com base em URLs.



Introdução

- **Angular 17:**
- A última versão do Angular, Angular 17, continua a evolução do framework com melhorias de desempenho, novos recursos e correções de bugs.
- A cada nova versão, a equipe do Angular foca em melhorar a produtividade do desenvolvedor, a experiência do usuário e a eficiência do aplicativo.



Introdução

- **Lado do servidor:**

- O Angular é um framework do lado do cliente, o que significa que ele é executado no navegador do usuário.
- No entanto, é comum usar o Angular em conjunto com um backend de servidor para fornecer dados, autenticação, autorização e outras funcionalidades. Angular pode se integrar facilmente a várias tecnologias de back-end, como [Node.js](#), [Java Spring](#), [ASP.NET Core](#), entre outras.



Introdução

- **Acoplamento:**

- Angular é um framework **full-stack**, o que significa que ele fornece todas as ferramentas necessárias para desenvolver um aplicativo completo, desde a interface do usuário até a lógica de negócios e integração com o servidor.
- Embora seja possível usar Angular apenas para a parte front-end de um aplicativo, ele também oferece recursos para criar aplicativos completos, incluindo manipulação de rotas, autenticação de usuários, comunicação com APIs e muito mais.



Angular CLI

- **Angular CLI**

- Ao instalar o Angular CLI (Command Line Interface), você obtém uma poderosa ferramenta de linha de comando que simplifica o processo de desenvolvimento Angular. Uma das funcionalidades do Angular CLI é a capacidade de **gerar um servidor de desenvolvimento embutido** para facilitar o teste e a visualização de aplicativos Angular localmente.
- Quando você cria um novo projeto Angular usando o Angular CLI e executa o comando '**ng serve**', o Angular CLI inicializa um servidor de desenvolvimento local usando o Webpack Dev Server.
- Esse servidor inicia um ambiente de desenvolvimento no seu computador que permite acessar e visualizar o aplicativo Angular no navegador.



Instalação

- Processo de instalação:
- 1) Antes de instalar o Angular devemos instalar o Node.js para utilizar o npm (Node Package Manager) que é o gerenciador de pacotes que vem com o Node.js.

Baixar o Node.js:

- Acesse o site oficial do Node.js em <https://nodejs.org/>.
- Na página inicial, você verá dois botões para download: "LTS" (versão de suporte de longo prazo) e "Current" (versão atual).
- Recomendo escolher a versão LTS para garantir maior estabilidade, a menos que você tenha uma razão específica para escolher a versão atual.



The screenshot shows the official Node.js website. At the top, there's a navigation bar with links for Estudar, Sobre, Descarregar, Blogue, Documentação, Colaborar, Certificação, a search bar, and keyboard shortcuts (Ctrl + K). Below the navigation, the main heading reads "Run JavaScript Everywhere". A text block explains that Node.js is a free, open-source, cross-platform JavaScript runtime environment. A large green button labeled "Get Node.js®" is highlighted with a red box. Below it, smaller buttons for "Get security support" and "for EOL Node.js versions" are visible. To the right, a code editor window displays a simple HTTP server example in JavaScript. The code uses the 'http' module to create a server that responds with "Hello World!". It also shows how to run the server with 'node server.mjs'. A "Copy to clipboard" button is at the bottom of the code editor. At the very bottom of the page, there are links for "v22.20.0 Latest LTS" and "v24.9.0 Latest Release", along with links for "Política da Marca Comercial", "Política de Privacidade", "Código de Conduta", and "Política de Segurança".

Instalação

Instalar o Node.js:

- Após baixar o instalador, dê um duplo clique para iniciá-lo.
- O assistente de instalação será aberto. Clique em "Next" para começar.
- Aceite os termos do contrato de licença e clique em "Next".
- Escolha o local de instalação ou deixe o padrão e clique em "Next".
- Marque a opção "Automatically install the necessary tools..." se desejar que o instalador instale ferramentas adicionais úteis (como o npm) e clique em "Next".
- Clique em "Install" para iniciar a instalação.

The screenshot shows a web browser displaying the Node.js download page at nodejs.org/pt/download. At the top, there's a message about Docker installation instructions. Below it, a PowerShell command block shows how to use Docker to install Node.js. Further down, there's a note about Docker being a containerization platform. The main section is titled 'Obter um binário de Node.js® pré-compilada para' with a dropdown set to 'Windows'. Underneath, two download buttons are shown: 'Windows Instalador (.msi)' (highlighted with a red box) and 'Binário Autônomo (.zip)'. Below the buttons, there are links for reading change logs, verifying checksums, and finding previous releases.

```
1 # Docker possui instruções de instalação específicas para cada sistema operativo.
2 # Podemos consultar a documentação oficial no https://docker.com/get-started/
3
4 # Puxar ou extrair a imagem de Docker da Node.js:
5 docker pull node:22-alpine
6
7 # Criar um container de Node.js e iniciar uma sessão de Shell:
8 docker run -it --rm --entrypoint sh node:22-alpine
9
10 # Verify the Node.js version:
11 node -v # Should print "v22.20.0".
12
13 # Consultar a versão da npm:
14 npm -v # Deveria imprimir "10.9.3".
```

Docker é uma plataforma de contencionização. Se encontrarmos quaisquer problemas, podemos visitar o [sítio da Docker](#).

Ou obter um binário de Node.js® pré-compilada para Windows executando uma arquitetura x64.

[Windows Instalador \(.msi\)](#) [Binário Autônomo \(.zip\)](#)

Ler o registo de alterações para esta versão.
Ler o registo de alterações para esta versão.
Saber como verificar SHASUMS assinados.
Procuramos pelo código-fonte da Node.js? Podemos descarregar um ficheiro compactado assinado do código-fonte da Node.js.
Consultar nossos binários noturnos ou todos lançamentos anteriores ou os binários não oficiais para outras plataformas.

Instalação

Verificar a Instalação:

Para verificar se o Node.js foi instalado corretamente, abra o Prompt de Comando (cmd) ou o PowerShell.

Digite os seguintes comandos e pressione Enter após cada um:

node -v

npm -v

Se as versões do Node.js e npm forem exibidas, a instalação foi bem-sucedida.



```
C:\ Prompt de Comando
C:\Users\Andre>node -v
v22.20.0
C:\Users\Andre>npm -v
6.9.3
C:\Users\Andre>
```

Instalação

- **Processo de instalação (vamos fazer nos próximos slides e não agora):**
- **Para instalar ele globalmente e não dentro de uma pasta apenas, iremos entrar no cmd e instalar na pasta c:\Users\o nome do usuário do seu pc**
- Instalação do Angular CLI: Você instala o Angular CLI globalmente usando o npm (Node Package Manager) com o comando: `npm install -g @angular/cli`
 - Depois de instalado para ver a versão usar o seguinte comando: `ng version`
- Criação de um novo projeto Angular: Você cria um novo projeto Angular usando o Angular CLI com o comando (escolha uma pasta de destino e digite o comando no cmd dentro da pasta):
 - `ng new nome-do-projeto`
- **Execução do servidor de desenvolvimento:** Você navega até o diretório do projeto recém-criado e executa o comando: `ng serve`. Isso inicia o servidor de desenvolvimento local.
- **Acesso ao aplicativo no navegador:** Após o servidor iniciar, você pode acessar o aplicativo Angular no navegador digitando `http://localhost:4200` na barra de endereços ou no Software ‘VS Code’ ele já disponibiliza o link se tudo ocorrer certo durante a execução.
 - O projeto padrão que é criado é um template exemplo do Angular, agora podemos configurar nosso site ou aplicação como desejar.



Instalação - Observações

- **Atualização automática:** O servidor de desenvolvimento incluído com o Angular CLI é configurado para atualizar automaticamente o navegador sempre que você fizer alterações no código-fonte do seu aplicativo. Isso simplifica o processo de desenvolvimento, permitindo que você veja instantaneamente as alterações refletidas no navegador enquanto trabalha no código.
- Em resumo, o Angular CLI facilita o processo de desenvolvimento Angular, fornecendo uma maneira conveniente de criar novos projetos, iniciar servidores de desenvolvimento locais e automatizar tarefas comuns de desenvolvimento



Servidor Angular - Conceitos

- **SERVIDOR DO ANGULAR**

- O servidor de desenvolvimento utilizado pelo Angular CLI, chamado Webpack Dev Server, não é específico do Angular e é uma ferramenta bastante comum no desenvolvimento web moderno. O Webpack Dev Server é uma parte do ecossistema do Webpack, que é uma ferramenta de empacotamento de módulos amplamente utilizada na comunidade de desenvolvimento front-end.
- O Webpack Dev Server é responsável por servir os arquivos estáticos do aplicativo (como HTML, CSS, JavaScript e outros recursos) durante o desenvolvimento e por fornecer recursos adicionais, como recarregamento automático da página quando os arquivos são alterados.



TypeScript e Webpack

- **TypeScript e Webpack:**
- O Angular utiliza o TypeScript como sua linguagem principal de programação. TypeScript é uma superset do JavaScript que adiciona tipagem estática e outras características ao JavaScript.
- O Webpack é uma ferramenta de empacotamento de módulos que permite aos desenvolvedores organizar e empacotar os arquivos de um aplicativo web. Ele é frequentemente usado em conjunto com o TypeScript para compilar o código TypeScript em JavaScript e empacotar todos os recursos do aplicativo em um formato adequado para execução em navegadores.



TypeScript e Webpack

- **Desenvolvimento local com o Webpack Dev Server:**
- Durante o desenvolvimento local de um aplicativo Angular, o Angular CLI utiliza o Webpack Dev Server para servir os arquivos do aplicativo e fornecer funcionalidades como recarregamento automático da página.
- O Webpack Dev Server é configurado pelo Angular CLI para atender às necessidades específicas do desenvolvimento Angular, como compilação em tempo real do código TypeScript em JavaScript e recarregamento automático da página quando os arquivos são alterados.



Renderização – Cliente e Servidor

- Desenvolvimento local com o Webpack Dev Server:
- O Angular CLI permite que você escolha entre duas opções de renderização: "Client-side Rendering" (CSR) e "Server-side Rendering" (SSR).
 - Client-side Rendering (CSR): Essa é a opção padrão e é ideal para aplicativos em que a renderização é feita no navegador do cliente. Aqui, todo o processamento do aplicativo, incluindo a renderização dos componentes Angular, é feito no navegador do cliente.
 - Server-side Rendering (SSR): Com essa opção, o Angular CLI configura o projeto para permitir a renderização do lado do servidor. Isso significa que uma versão renderizada do aplicativo é enviada do servidor para o navegador do cliente, o que pode melhorar a velocidade de carregamento inicial e a indexação pelos mecanismos de busca.
 - Essas opções podem ser selecionadas durante a criação de um novo projeto Angular usando o Angular CLI. Ao escolher a opção de renderização do lado do servidor (SSR), o Angular CLI pode configurar o projeto para usar frameworks como Angular Universal, que permite a renderização do lado do servidor em aplicativos Angular.



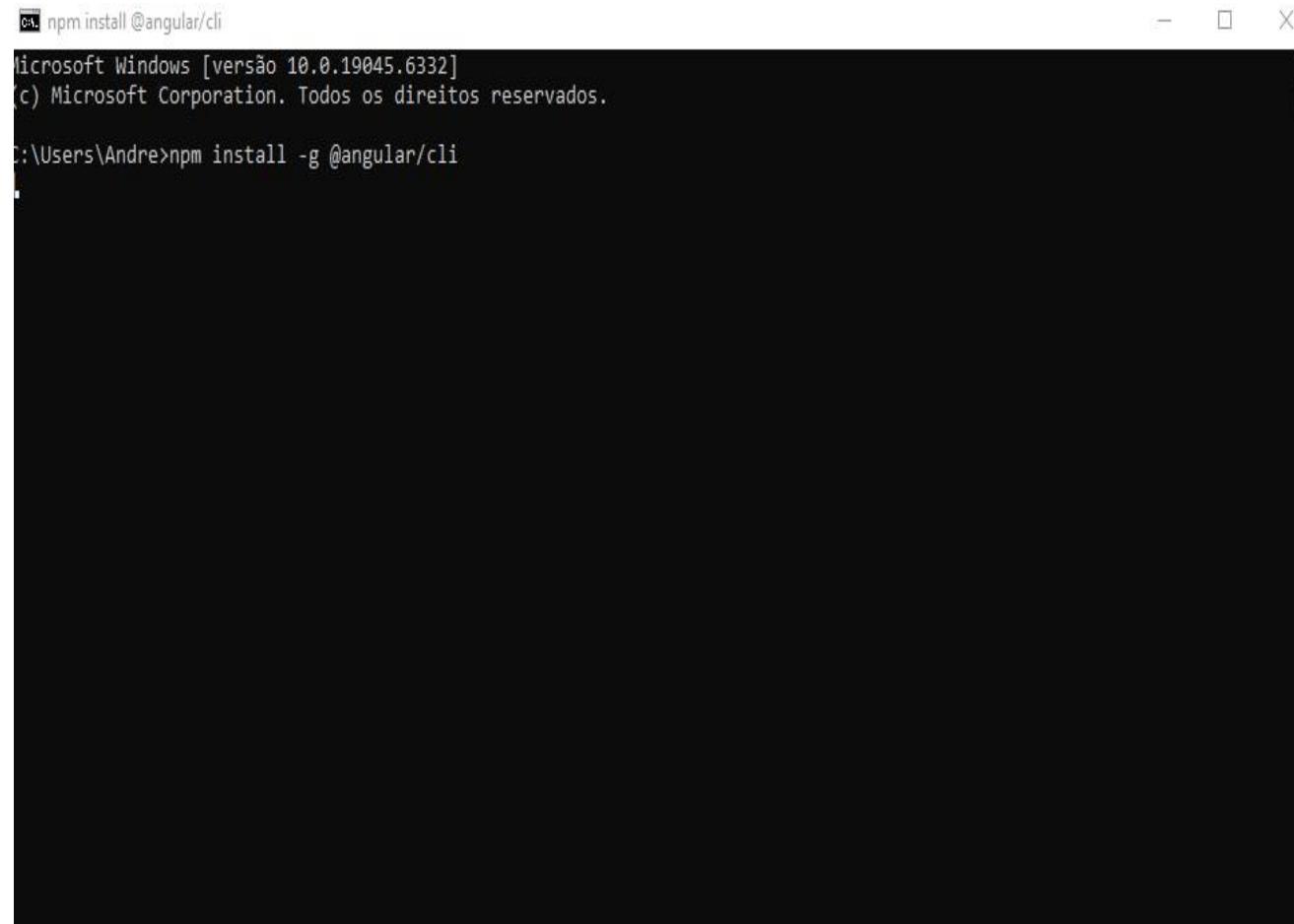
Atividade Prática

- Vamos instalar ele através de um terminal no Windows (Tecla Win+r e Digitar CMD). Como vamos instalar ele globalmente, não vai ser necessário estar dentro de uma pasta definida e sim na pasta c:\Users\o nome do usuário do seu pc

- Comando (ou dentro da pasta que escolher criar o projeto):

cmd

npm install -g @angular/cli



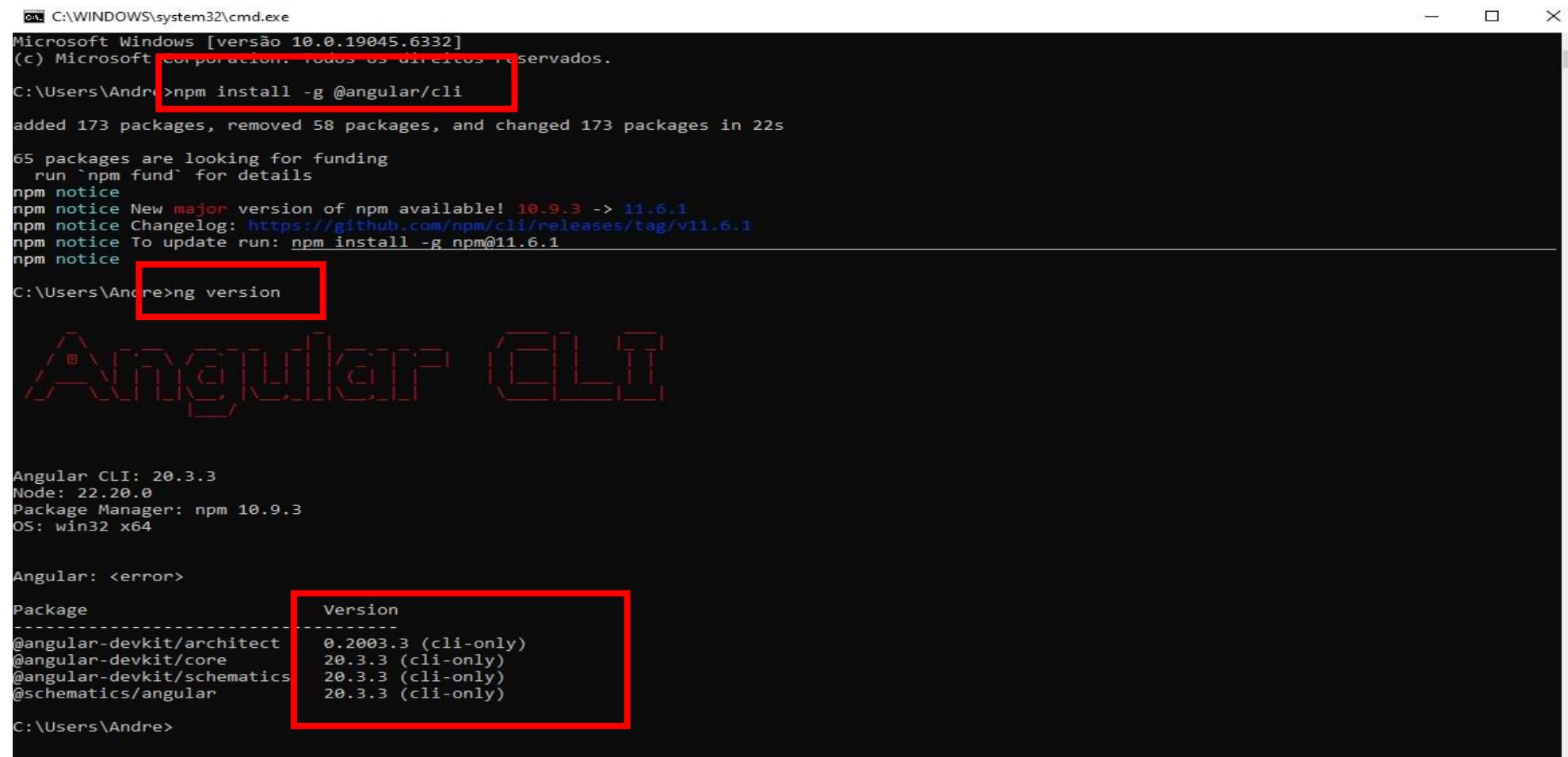
```
npm install -g @angular/cli
Microsoft Windows [versão 10.0.19045.6332]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Andre>npm install -g @angular/cli
```

Atividade Prática

- Depois de instalado para ver a versão instalada usar o seguinte comando:

`ng version`



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 10.0.19045.6332]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Andre>npm install -g @angular/cli
added 173 packages, removed 58 packages, and changed 173 packages in 22s

65 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New major version of npm available! 10.9.3 -> 11.6.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.6.1
npm notice To update run: npm install -g npm@11.6.1
npm notice

C:\Users\Andre>ng version
Angular CLI: 20.3.3
Node: 22.20.0
Package Manager: npm 10.9.3
OS: win32 x64

Angular: <error>
  Package          Version
  -----          -----
  @angular-devkit/architect    0.2003.3 (cli-only)
  @angular-devkit/core         20.3.3 (cli-only)
  @angular-devkit/schematics   20.3.3 (cli-only)
  @schematics/angular          20.3.3 (cli-only)

C:\Users\Andre>
```

Atividade Prática

- Após instalação e comando para verificar a versão instalada devemos ter algo parecido a isso:

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 10.0.19045.6332]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Andre>npm install -g @angular/cli
added 173 packages, removed 58 packages, and changed 173 packages in 22s

65 packages are looking for funding
  run `npm fund` for details
npm notice
npm notice New major version of npm available! 10.9.3 -> 11.6.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v11.6.1
npm notice To update run: npm install -g npm@11.6.1
npm notice

C:\Users\Andre>ng version
Angular CLI: 20.3.3
Node: 22.20.0
Package Manager: npm 10.9.3
OS: win32 x64

Angular: <error>
  Package          Version
  -----          -----
  @angular-devkit/architect    0.2003.3 (cli-only)
  @angular-devkit/core        20.3.3 (cli-only)
  @angular-devkit/schematics  20.3.3 (cli-only)
  @schematics/angular         20.3.3 (cli-only)

C:\Users\Andre>
```

Atividade Prática

- Agora devemos dar o comando para criar um novo projeto dentro da pasta que queremos no ‘cmd’.

Comando: ng new angular01

```
Angular CLI: 17.1.2
Node: 20.10.0
Package Manager: npm 10.2.3
OS: win32 x64

Angular:
...
Package          Version
-----
@angular-devkit/architect    0.1701.2 (cli-only)
@angular-devkit/core         17.1.2 (cli-only)
@angular-devkit/schematics   17.1.2 (cli-only)
@schematics/angular          17.1.2 (cli-only)

C:\GERAL\UFN_PROFESSOR\PROFESSOR\2024-01\G00E1_APPLICACOES_WEB_SI\Angular\Exemplos\aula01>ng new angular01
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
  SCSS  [ https://sass-lang.com/documentation/syntax#scss ]
  Sass  [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
  Less  [ http://lesscss.org ]
```



Devemos escolher um formato de ‘CSS’ entre os

4 disponíveis:

- **CSS (Escolher esse)**
- SCSS
- Sass
- Less

Pesquisar as diferenças entre cada um desses modos!

Atividade Prática

- A próxima pergunta é se desejamos ativar o SSR (Renderização do lado do Servidor) e SSG (Pré-renderização do lado Front-end)

Resposta: 'y'.

Outra pergunta: ? Do you want to create a 'zoneless' application without zone.js? (y/N)

Resposta: 'N'.

Pergunta sobre AI: **Resposta: None**

```
c:\GERAL\UFN_PROFESSOR\PROFESSOR\2025\2025-02\JOGOS_DIGITAIS\TECNOLOGIAS_WEB\EXERCICIOS\EXERCICIOS_AULA10>cd Angular_ex1

c:\GERAL\UFN_PROFESSOR\PROFESSOR\2025\2025-02\JOGOS_DIGITAIS\TECNOLOGIAS_WEB\EXERCICIOS\EXERCICIOS_AULA10\Angular_ex1>ng new angular01
✓ Which stylesheet format would you like to use? CSS [ https://developer.mozilla.org/docs/Web/CSS ]
✓ Do you want to enable Server-Side Rendering (SSR) and Static Site Generation (SSG/Prerendering)? Yes
✓ Do you want to create a 'zoneless' application without zone.js? No
? Which AI tools do you want to configure with Angular best practices? https://angular.dev/ai/develop-with-ai (Press <space> to select, <a> to
toggle all, <i> to invert selection, and <enter> to proceed)
>(*) None
( ) Claude      [ https://docs.anthropic.com/en/docs/claude-code/memory ]
( ) Cursor      [ https://docs.cursor.com/en/context/rules ]
( ) Gemini      [ https://ai.google.dev/gemini-api/docs ]
( ) GitHub Copilot [ https://code.visualstudio.com/docs/copilot/copilot-customization ]
( ) JetBrains AI [ https://www.jetbrains.com/help/junie/customize-guidelines.html ]
( ) Windsurf    [ https://docs.windsurf.com/windsurf/cascade/memories#rules ]
```



Logo após o comando o angular começará a criar o nosso projeto na pasta escolhida!! Necessário aguardar até a finalização.

Atividade Prática

- Após a criação do projeto devemos dar o comando para inicializar o servidor. Abrir a pasta do projeto no ‘**vscode**’ e dar o comando lá dentro que irá gerar o link automaticamente. Obs: dar ‘y’ se houver perguntas.
- **Comando:** `ng serve`



The screenshot shows the Visual Studio Code interface with the Angular project structure. The Explorer sidebar shows files like `app.component.css`, `app.component.html`, and `app.component.ts`. The `app.component.html` file is open in the editor, displaying a template placeholder and some CSS styles. The Terminal tab at the bottom shows the command `ng serve` being run in a PowerShell window. The status bar indicates the current path is `C:\GERAL\UFN_PROFESSOR\PROFESSOR\2024\2024-01\G00E1_APPLICACOES_WEB_SI\Angular\Exemplos\aula01\angular01`.

Atividade Prática

- Se tudo ocorreu 'ok' e sem erros o angular já disponibiliza o link para o nosso servidor com o nosso projeto: <http://localhost:4200>.
- O projeto que ele cria é um exemplo padrão. Então devemos modificar todo ele e adaptar ao nosso projeto. Devemos conhecer toda a estrutura de pastas que ele cria para entender melhor o funcionamento e roteamento do projeto.



The screenshot shows the Visual Studio Code interface with an Angular project open. The Explorer sidebar on the left lists files and folders like .angular, .vscode, node_modules, src, app, assets, favicon.ico, index.html, main.server.ts, main.ts, styles.css, .editorconfig, .gitignore, angular.json, package-lock.json, package.json, README.md, server.ts, tsconfig.app.json, tsconfig.json, and tsconfig.spec.json. The app component's HTML file is selected and shown in the main editor area. The terminal at the bottom displays the build process and the generated bundle sizes:

```
Initial chunk files      | Names          | Raw size
chunk-MPNIZYFA.mjs     | -              | 1.70 MB
polyfills.server.mjs    | polyfills.server | 557.44 kB
main.server.mjs         | main.server    | 216.22 kB
chunk-VPSODEBW.mjs     | -              | 2.51 kB
render-utils.server.mjs | render-utils.server | 423 bytes

Lazy chunk files        | Names          | Raw size
chunk-OTT6LQ5K.mjs     | xhr2           | 39.10 kB

Application bundle generation complete. [3.586 seconds]

Watch mode enabled. Watching for file changes...
→ Local: http://localhost:4200/
→ press h + enter to show help
```

Atividade Prática

- Resultado:



 Angular

Hello, angular01

Congratulations! Your app is running. 🎉

[Explore the Docs](#)

[Learn with Tutorials](#)

[CLI Docs](#)

[Angular Language Service](#)

[Angular DevTools](#)

⟳ ⌛ ⏴

Estrutura do Projeto Padrão



- Estrutura do projeto resumida:
 - `node_modules/`
 - Diretório onde estão instaladas todas as dependências do projeto, gerenciado pelo npm.
 - `src/` : Diretório principal onde reside o código-fonte da aplicação.
 - `app/` : Contém o módulo raiz (`app.module.ts`) e o componente principal (`app.component.ts`), além de outros componentes, serviços e módulos da aplicação.
 - `assets/` : Armazena arquivos estáticos como imagens, fontes, etc.
 - `environments/`
 - Contém arquivos de configuração para diferentes ambientes (desenvolvimento, produção, etc.).
 - `environment.ts`: Configuração para o ambiente de desenvolvimento.
 - `environment.prod.ts`: Configuração para o ambiente de produção.
 - `index.html`
 - Arquivo HTML principal que serve de entrada para a aplicação.
 - `main.ts`
 - Arquivo principal que inicializa a aplicação Angular.
 - `polyfills.ts`
 - Scripts necessários para compatibilidade com navegadores mais antigos.
 - `styles.scss`
 - Arquivo global de estilos para a aplicação.
 - Existem diversos arquivos e pastas no projeto, por padrão podem ser criados diferentes. Se faltar algum parâmetro essencial devemos criar manualmente e adaptando ao nosso projeto, tornando necessário realizar bastante leitura sobre diferentes contextos do Angular. Cada versão tem uma maneira de se trabalhar, o que pode apresentar algumas dificuldades aos programadores.

Tutoriais indicados



- Existem alguns tutoriais que auxiliam no aprendizado do Angular, porém precisamos tomar cuidado com as versões que estão sendo utilizadas e a versão que estamos usando. Muitas vezes é necessário adaptações.
- Tutorial completo para criar um CRUD Angular + Spring Boot Back-end (Java)
- Link: <https://www.youtube.com/watch?v=qJnjz8Fls6Q>
- Loiane Groner é uma ótima programadora nesta área, atua nos EUA.
- **Desafio para casa:**
 - Assistir este tutorial e tentar replicar o que ela está fazendo. Cuidado que a versão que ela está usando é uma versão anterior a Versão 17.
 - Objetivos:
 - Aprender a mapear e rotear todo o projeto
 - Usar o Angular Materials.
 - Trabalhar com cores e temas do Angular.

Referências

HOGAN, Brian P. HTML 5 e CSS3: Desenvolva hoje com o padrão de amanhã . Rio de Janeiro (RJ): Ciência Moderna Ltda., 2012 282 p. ISBN 978-85-399-0260-6

Fábio Flatschart. HTML 5 - Embarque Imediato, 2011. (Biblioteca Digital)

Deitel, Paul J.; Deitel, Harvey M.. Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores, 2008. (Biblioteca Digital)

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. 2. ed. São Paulo, SP: Novatec, 2014. 335 p. ISBN 978-85-7522-403-8.

Denilson Bonatti. Desenvolvimento de Jogos em HTML5, 2014. (Biblioteca Digital)

W3SCHOOL. The world's largest web development site. Acessado em 2018. Disponível em:
<http://www.w3schools.com/>.

W3C. World Wide Web Consortium. Acessado em 2018. Disponível em: <http://www.w3.org>

MORRISON, Michael. Use a cabeça JavaScript. Rio de Janeiro (RJ): Alta Books, 2008. 606 p.

<https://br.freepik.com/fotos-vetores-gratis/desenvolvimento-web>

Material do Professor Fabrício Tonetto Londero, 2023.

Obrigado pela atenção!!



Email: andre.flores@ufn.edu.br