Introdução ao

Angular

Maria Elisa Gomes de Matos

Atua como desenvolvedora de software plena na Myrp Soluções Ltda, com ampla experiência em desenvolvimento de aplicações, arquitetura de software, testes automatizados, computação em nuvem, controle de versões e desenvolvimento ágil de software. Pós-graduanda em Engenharia de Software com ênfase em DevOps na UNIFOR. Graduada em Ciências da Computação pela UVA e Técnica em Informática no GWA.

@mariaelisagmt

(88) 99794-5263

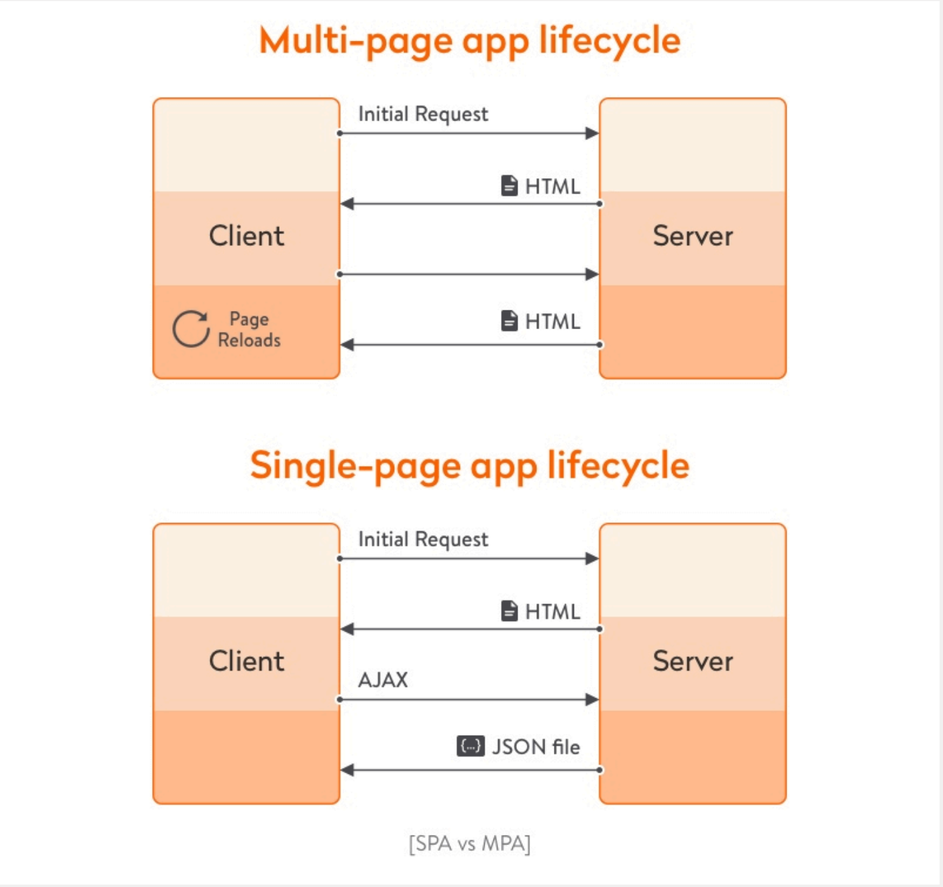
maria.elisa.gomes.de.matos@gmail.com
mariaelisagmt@edu.unifor.com

https://www.linkedin.com/in/maria-elisa-gomes-de-matos/

Introdução

Angular é um framework front-end baseado em TypeScript, desenvolvido pelo Google, usado para criar aplicações web dinâmicas e SPA (Single Page Applications).

SPA VS MPA



[SPA vs MPA]

Principais Recursos

01 Modularidade: Aplicações divididas em módulos reutilizáveis.

02 Componentes: Interface construída com componentes independentes.

03 Two-Way Data Binding: Sincronização automática entre a interface e os dados.

04 Injeção de Dependência: Facilita o gerenciamento de serviços e dados.

05 Suporte a TypeScript: Aumenta a segurança e produtividade.

Vantagens e Desavantagens

01 Suportado pelo Google.

02 Escalável para projetos pequenos e grandes.

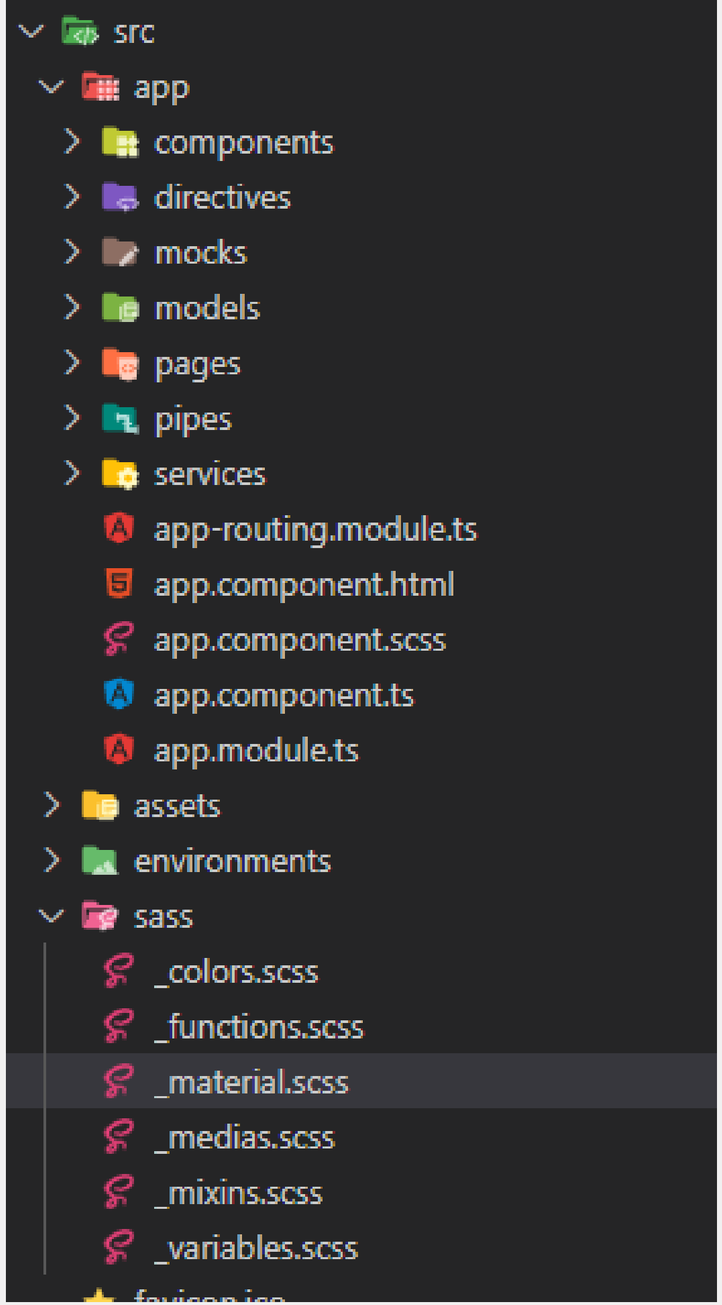
03 Comunidade ativa e documentação extensa.

04 Ferramentas nativas como Angular CLI e Angular Material.

01 Curva de Aprendizado

02 Requer conhecimento de TypeScript.

03 Pode ser complexo para projetos pequenos.



Estrutura do Projeto

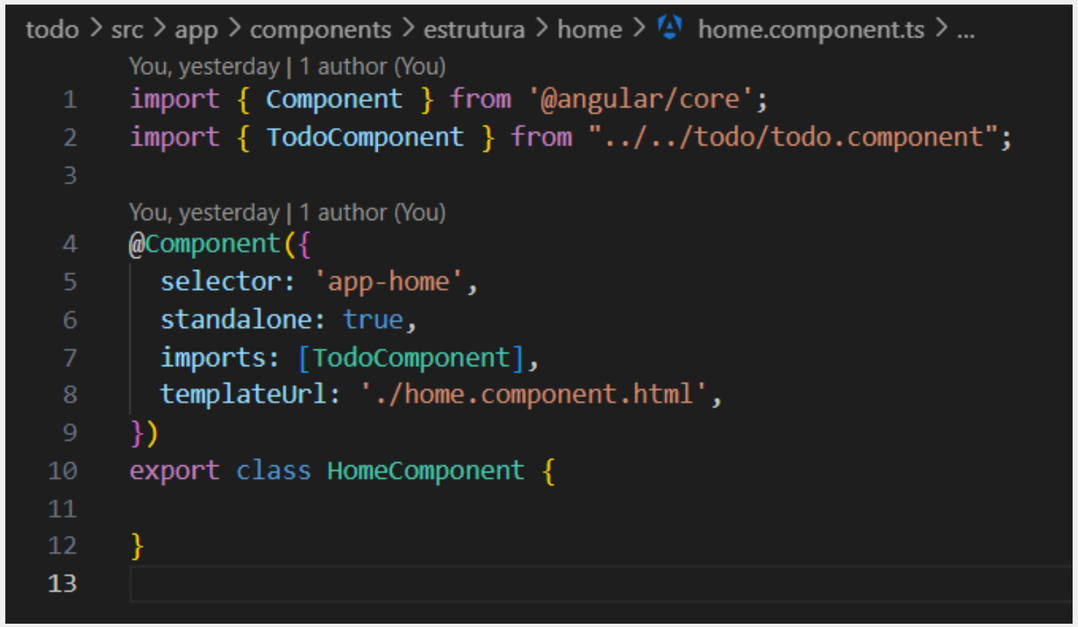
Arquivos importantes:

- src/app: Onde os componentes e serviços são criados.
- angular.json: Configuração do projeto.
- package.json: Dependências e scripts.

Fluxo básico de desenvolvimento

```
npm install -g @angular/cli
ng new my-angular-app
cd my-angular-app
ng serve
ng generate component nome-do-componente
```

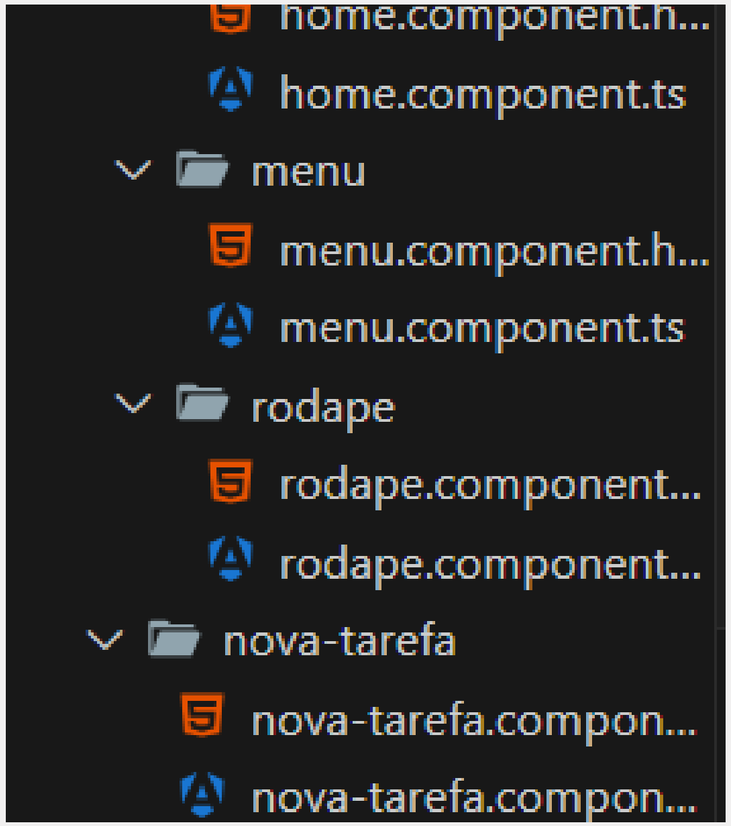
Componentes em Angular



Um componente é uma parte reutilizável da interface de usuário, que combina HTML, CSS e lógica em TypeScript.

Um projeto Angular é composto por vários componentes, que se comunicam entre si e formam a aplicação como um todo.

Arquitetura de Componentes em Angular



Cada componente tem 3 partes principais:

1. HTML Template – Define a interface visual.
2. CSS Styles – Define o estilo visual.
3. TypeScript Class – Define a lógica e comportamento do componente.

- Estrutura:
  - app.component.ts (Lógica)
  - app.component.html (Interface)
  - app.component.css (Estilo)
  - app.component.spec.ts (Testes unitários)

Ciclo de Vida de um Componente

Angular oferece métodos de ciclo de vida para que você possa controlar o comportamento do componente em diferentes fases.

1. ngOnInit() – Executado quando o componente é inicializado.
2. ngOnChanges() – Chamado quando um valor de input é alterado.
3. ngOnDestroy() – Executado quando o componente é destruído.

Comunicação entre Componente

```
@Input() titulo: string;
```

@Input() – Passa dados do componente pai para o filho.

```
@Output() clicado = new EventEmitter<void>();

onClick() {
  this.clicado.emit();
}
```

@Output() – Envia eventos do componente filho para o pai.

Service Sharing – Compartilha dados entre componentes usando serviços.

Componentes Aninhados

```
<app-header></app-header>
<router-outlet></router-outlet>
<app-footer></app-footer>
```

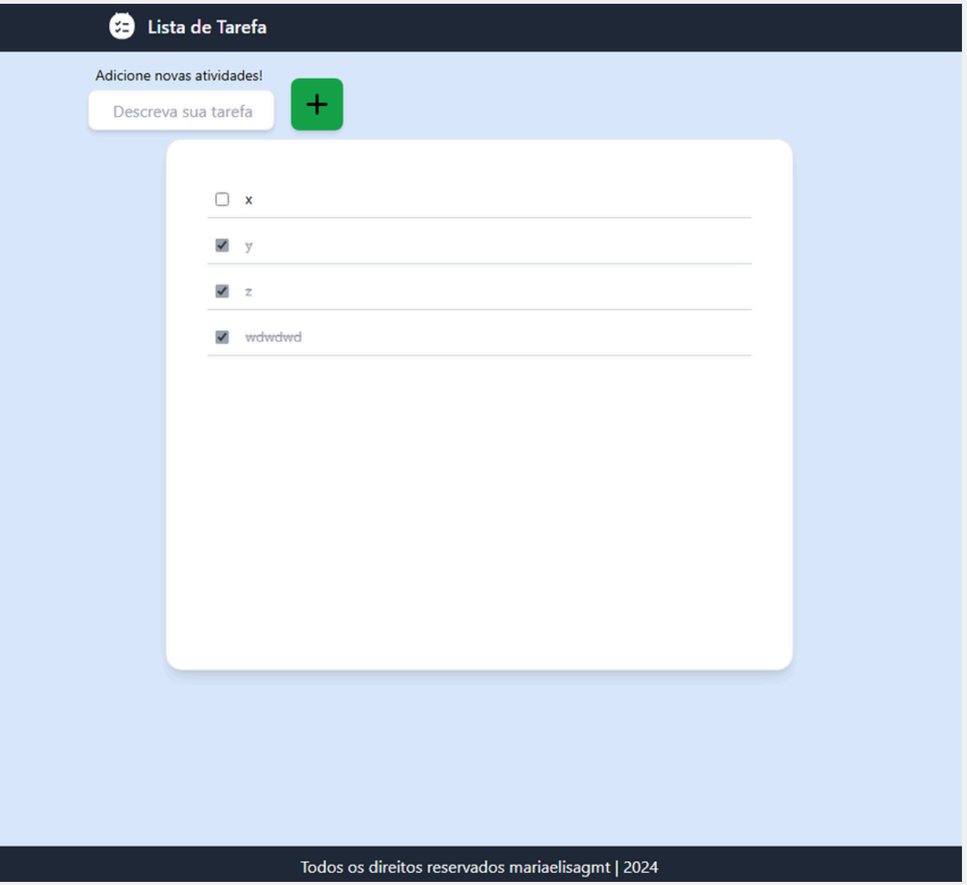
Componentes podem ser aninhados dentro de outros.

Boas Práticas para Componentes

- Simplicidade: Mantenha os componentes simples e focados em uma única função.
- Reutilização: Divida a aplicação em componentes reutilizáveis.
- Modularidade: Use módulos para agrupar componentes relacionados.
- Lazy Loading: Carregue componentes sob demanda para melhorar a performance.
- Teste de Componentes: Utilize arquivos .spec.ts para criar testes automatizados.

Laboratório

Projeto Final



1. Fork do projeto no seu github:
2. Clone do projeto no seu ambiente local
3. Criei uma branch com o modelo feature/add-page-[nome-aluno]
4. Adicione 3 Componentes
5. Adicione o Service
6. Adicione o Model
7. Melhore o layout
8. Utilize o git para controle de versão do seu código
9. Publique a aplicação no seu github

Muito Obrigada!

Caso tenham dúvida podem entrar em contato.