

## Situação de Aprendizagem – Desenvolvimento Web com Angular 19

Após feito sua avaliação deverá ser adicionado a um repositório no gitHub e enviado o link do projeto para o instrutor no e-mail - [prof.desemble@gmail.com](mailto:prof.desemble@gmail.com) – com assunto “nome de escala - Atividade 30PTS - ANGULAR 19”

## Título: Aplicando Componentização e Serviços em Angular 19 Standalone

## Contexto:

Você foi contratado como desenvolvedor(a) frontend júnior para uma startup de soluções digitais. Seu primeiro desafio é criar uma página inicial funcional com dois componentes independentes. Cada componente deverá conter funcionalidades específicas, utilizando boas práticas de componentização, serviços e integração com API REST pública.

## Desafio (Produto Esperado):

Desenvolver uma aplicação Angular 19 (modo standalone) contendo:

1. **Página inicial (home.component) com dois botões:**
  - Calcular Área ao Quadrado
  - Autenticar e Listar Dados
  - Criar as rotas para cada botão ao clicar seja direcionado para os seus devidos componentes.
2. **Componente calcularQuadrado:**
  - Ao ser acionado, deverá exibir inputs para largura e altura.
  - Deve usar um **service exclusivo** para calcular o quadrado da área (largura × altura)<sup>2</sup>.
  - Exibir o resultado diretamente no .html do componente, com formatação em **CSS**.
3. **Componente AuthApi:**
  - Ao ser acionado, deve consumir dados da API: <https://jsonplaceholder.typicode.com/posts>
  - Utilizar um **service exclusivo** para a requisição HTTP.
  - Exibir os campos title e body em uma tabela estilizada, diretamente no .html do componente (sem uso de templates externos).

**Importante:** Não utilizar template no componente. A coleta e exibição das informações devem acontecer somente nos arquivos component.html dos respectivos componentes, com estilos aplicados via CSS.

## Critérios de Avaliação (30 pontos totais):

Critério	Pontuação
Estrutura Angular standalone corretamente implementada	5 pts
Página inicial funcional com navegação e botões	5 pts
Componente calcularQuadrado funcional e bem estruturado	5 pts
Service calcularQuadrado.service.ts corretamente utilizado	5 pts
Componente AuthApi funcional e com consumo correto da API	5 pts
Service auth-api.service.ts implementado e integrando corretamente com o componente	3 pts
Estilização CSS aplicada corretamente em ambos os componentes	2 pts