

## Desafio 5.1

### a) Explique de forma sucinta a diferença entre SPA e MPA;

SPA ou Single Page Application é uma página que funciona dentro do próprio navegador, não precisando ser recarregada durante seu uso. MPA ou Multi-page Application é uma página que é recarregada toda vez que seu conteúdo muda, normalmente usado para sites com grande quantidade de informações, como e-commerce.

### b) Como eram as páginas WEB na década de 90 e 1ª década dos anos 2000?

Nos anos 90 foram criados o famoso HTTP (protocolo de Transferência de Hipertexto) e a linguagem HTML (Linguagem de Marcação de Hipertexto), o que facilitou a navegação e a criação de páginas respectivamente. Nessa década também foi criado o Mosaic, o primeiro navegador. Logo em seguida foi criado o Netscape e posteriormente o Google.

Nos anos 2000 as páginas utilizavam poucas imagens e cores fortes, pois a internet era discada ainda, as páginas eram estáticas e não havia interatividade com o usuário.

### c) O que é WEB 1.0, WEB 2.0 e WEB 3.0?

Web 1.0 começou com a criação do WWW, páginas que era possível visitar, mas não interagir.

Web 2.0 as páginas já permitem maior interação, como criação de conteúdos e compartilhamentos entre si. A Web 2.0 foi o criador e difusor da portabilidade e alcance da internet para o maior número de pessoas.

Web 3.0 essa interatividade é ainda maior, utilizando-se de algoritmos que conseguem personalizar seu uso, buscando conteúdos relevantes para você. A versão 3.0 é construída sobre os conceitos centrais de descentralização, abertura e maior atuação do usuário.

### d) Com suas palavras, responda às seguintes perguntas referentes ao Angular:

#### I. De forma resumida o que é o Angular?

É um framework da Google em parceria com a Microsoft para criação de interfaces utilizando HTML, CSS e TS.

#### II. Qual a relação do Angular com SPA?

O Angular facilita a criação de SPA, pois podemos destacar os componentes, templates, diretivas, roteamento, módulos, serviços, injeção de dependências e ferramentas de infraestrutura que automatizam tarefas. Inclusive, por ter vários componentes fica mais fácil dar manutenção quando necessário.

#### III. Em uma aplicação básica, o que é essencial ter?

Módulos, Componentes, Templates, Metadata, Data binding, Diretivas, Serviços e Injeção de dependências

#### **IV. O que é o Angular CLI e qual a sua importância? Dê exemplos.**

O Angular CLI é baseado no Webpack, uma ferramenta que ajuda a processar e agrupar nossos diversos arquivos TypeScript, JavaScript, CSS, HTML e imagens. Por baixo dos panos, o Webpack é responsável por estruturar o ponto de entrada da aplicação, ou seja, a raiz contextual (arquivo de inicialização). Você pode por exemplo usar o comando `'ng g component nome-do-componente'` para criar um novo componente sem precisar fazê-lo de forma manual, pulando etapas e otimizando seu tempo.

#### **V. Para que serve o arquivo angular.json?**

O arquivo angular.json contém as configurações globais da aplicação, também define o arquivo que vai iniciar a aplicação, fazendo o SPA.

O arquivo angular.json fornece padrões de configuração de todo o espaço de trabalho e também específicas do projeto para ferramentas de construção e desenvolvimento fornecidas pela CLI do Angular. Os valores de caminho fornecidos na configuração são relativos à pasta do espaço de trabalho raiz.

#### **VI. Quais pacotes (libs) são considerados primários (default)?**

São módulos, componentes, diretivas e serviços.

#### **VII. Para que serve o arquivo package.json?**

Contém todas as dependências da nossa aplicação. Ele é um repositório central de configurações de ferramentas, por exemplo. Ele também é onde npm armazena os nomes e versões dos pacotes instalados.

#### **VIII. Para que serve o arquivo environment.ts?**

É o arquivo de configuração do ambiente para o ambiente de desenvolvimento.

A instalação do Angular é abordado no curso Angular Avançado, que foi passado no início da sua jornada, sendo assim presumo que seu ambiente esteja totalmente configurado. Então vamos dar start em um novo projeto para trabalharmos com algumas manipulações. Após criar seu projeto e rodar o mesmo. No `app.component.html` deixe apenas:

#### **e) Sobre Text Interpolation, responda:**

##### **I. O que é?**

Serve para renderizar dados dinâmicos, de modo que a página vai mudando com o tempo.

## **II. Quais os benefícios de se ter interpolação em uma aplicação Angular?**

Você consegue deixar a página dinâmica, não precisando alterar o código toda vez que quiser uma alteração.

## **III. Crie uma interpolação com o seu nome, com nome do seu guia, data do desafio e a sua turma do desafio.**

Criado.

### **f) A respeito de Bindings, responda:**

#### **I. O que é Property Binding e quais as suas aplicações?**

É o valor atribuído ao componente, pode ser aplicado no HTML, CSS, etc.

#### **II. Qual é o passo a passo do fluxo das informações com Property Binding?**

O fluxo de informações ocorre em apenas uma via, do component para o template.

#### **III. Dê um exemplo de Property Binding e explique o seu funcionamento.**

Pode alterar a cor de uma DIV, definindo essa propriedade no CSS e depois declarando-a no HTML utilizando colchetes [ ] para mostrar para o angular que um property binding

#### **IV. O que é Event Binding?**

É quando uma lógica de programação é 'escutada' no HTML., sua direção é do template para o component.

#### **V. Dê um exemplo de Event Binding e explique o seu funcionamento.**

Clicar um botão no browser é um evento 'clique'. No HTML o angular escuta esse evento utilizando parênteses ( ).

#### **VI. O que é Two-Way-Data-Binding e em quais cenários devemos usá-lo?**

É um evento e propriedade acontecendo juntos, utilizando colchete e parenteses [()] 'banana in a box'. É utilizado quando queremos que tanto o template quanto o component fiquem atualizados ao mesmo tempo

#### **VII. Dê um exemplo de Two-Way-Data-Binding e explique.**

Preenchendo um formulário com nome, idade, endereço, etc

O component e template conversam ao mesmo tempo, as alterações ocorrem em tempo real.

### **g) Diretivas:**

#### **I. O que é uma diretiva Angular e qual o seu propósito?**

É uma forma de passar instruções para o código HTML, de modo a anexar um comportamento específico a um elemento.

## **II. Quais tipos de diretivas existem?**

São 2 tipos: atributo (NgIf, NgFor, NgSwitch) e estruturais (NgClass, NgStyle, NgModel).

## **III. O Angular possui algumas diretivas default, explique as 6 principais e dê um exemplo de cada.**

NgIf - mesmo comportamento da condicional 'If' nas linguagens de programação, cria ou remove um elemento na condição que você fornecer.

Exemplo: Verificar se o usuário está logado.

NgFor - é uma diretiva Angular que permite percorrer um array, ou qualquer objeto iterável, e exibir cada item do array como elemento na tela.

Exemplo: listar nomes de um Array.

NgSwitch - usado quando você precisa comparar o valor de uma variável com múltiplos valores.

Exemplo: quando você quer exibir um conteúdo de uma barra de navegação.

NgClass - permite adicionar ou remover várias classes.

Exemplo: alterar a cor de um botão quando o mesmo for clicado em um evento click.

NgStyle - permite adicionar ou remover várias estilos..

Exemplo: alterar cor e tamanho do texto, fonte do texto, cor de fundo, etc.

NgModel - adiciona um two-way-data-binding ao HTML.

Exemplo: atualiza a informação guardada em um formulário em tempo real conforme o usuário digita login e senha por exemplo.

## **h) Reactive forms:**

### **I. Qual a diferença entre template-driven e reactive forms?**

O template-driven é criado e configurado no HTML, enquanto o reactive forms é criado no próprio componente.

### **II. Quais benefícios temos ao usar um formulário reativo?**

O template HTML fica mais limpo, é fácil de criar e fazer manutenção.

### **III. Quais módulos são necessários para criar os formulários reativos?**

FormsModule, ReactiveFormsModule.

**IV. Quais classes base temos para formulários reativos no Angular?**

AbstractControl, FormControl, FormGroup, FormArray, FormBuilder.

**V. Como é feita a configuração de um formulário reativo usando separadamente as classes base?**

É feita de duas formas, utilizando FormGroup e FormControl. Ou utilizando o FormBuilder.

**VI. Como se dá o fluxo de informações entre o formulário reativo e o componente em Angular?**

É feita utilizando a diretiva formGroup.

**VII. Como é feita a captura dos dados de um formulário reativo e apresentado na tela?**

Utilizamos o FormControlName para linkar o formulário reativo com o template.

**VIII. Explique para o que serve a validação de um formulário e como isto pode ser apresentado para o usuário.**

A validação de um formulário serve para verificar se as informações que você pediu ao usuário são válidas para determinadas categorias, como por exemplo nome e email. Isto pode ser apresentado ao usuário utilizando a função validator do angular.