­Monique Conceição Rodrigues de Oliveira

DESAFIO 5.1 – SPA Angular



A diferença é que em SPA(Single Page Application ) a página principal é renderizada apenas 1 única vez e todas as outras informações serão carregadas conforme a necessidade do usuário, já a MPA(Multiple Page Application) todas as páginas da aplicação tem a necessidade de serem carregadas.



Em 1989 Tim Berners-Lee(Cientista da computação ) propôs um projeto audacioso e de grande importância que de forma resumida eram hipertextos para dinamizar a passagem de um texto para outro de forma mais rápida e dinâmica, o famoso World Wide Web(www); o projeto começou a tomar forma em 1990 e essa explosão ocorrida nos anos 90 abriu espaço para que muitas empresas estreassem seus sites na World Wide Web. Na época por conta das limitações tecnológicas a maioria das páginas nada mais eram como grandes banners de divulgação das empresas.



WEB 1.0- Início da internet, sites de conteúdo estático e pouca interatividade dos internautas e diversos diretórios de links;

WEB 2.0- Chamada de web participativa , foi a revolução dos blogs e chats, das mídias sociais colaborativas das redes sociais e do conteúdo produzido pelos próprios internautas.

WEB 3.0- É uma internet onde teremos toda informação de forma organizada os humanos e as máquinas entendem para que nos ajudem a responder pesquisa com solução concreta, personalizada e ideal; muito mais ampla que um conceito mobile, as pessoas estão e estarão conectadas 24 horas por dia 7 dias da semana por meio de dos celulares, smartphones, SmartTV’s, IPod’s, Tablets, carros.

Texto

Descrição gerada automaticamente

**|**- O Angular é um framework front-end do JavaScript codado em TypeScript, surgiu após o Angular JS como uma evolução do mesmo com a ideia de criação de componentes;

**||**- Total relação porque o Angular trabalha com o conceito de página única;

**|||**-

Se instala uma dependência do node e essa dependência cria no terminal um comando (ng) e partir daí são criados os projetos/ componentes; módulos:

*npm i -g @angular/cli*

Através do comando ng-new é criado um novo projeto:

*ng-new nomeDoProjeto*

Dentro da pasta do projeto o mesmo será compilado gerando os arquivos html, css e typeScript.

*npm start*

Dando tudo certo O cmd deve informar o localhost da aplicação. Isso significa que o ambiente está configurado para criar uma pequena aplicação.

**IV**- Angular CLI é uma ferramenta de linha de comando para ajudar a gerenciar as aplicações criadas em Angular.

Exemplos:

criar projeto: ng new *hello-world*

cria um novo prefixo no lugar do padrão app: ng new *hello-world* –prefix lib

configura o tipo de style à ser utilizado no projeto: ng new *hello-world* –style scss

**V-** Angular.json é um arquivo de configuração aonde algumas mudanças podem ser realizadas em seu objeto conforme a necessidade da aplicação, como por exemplo alterar o InlineTemplate de true para false- o que significa que no projeto o html ficará em um arquivo separado do typeScript;

**VI-**

Angular Router, Angular Forms, Angular HttpClient, Angular Animations, Angular PWA, Angular Schematics.

**VII-**

O package.json serve para configurar as ferramentas do repositório e armazenar os pacotes que são instalados.

**VIII-**

Por ser um ambiente de aplicação do Angular(informações de configuração json) serve para informar ao sistema quais os arquivos usar quando utilizar o ng build e ou o ng serve, PROD ou DEV.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Repositório salvo:**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

|- Text Interpolation é um recurso do Angular usado por padrão com as chaves duplas{{ }} para delimitar valores que serve para criar uma expressão dentro do template(html). Essas informações podem vir totalmente no html como por exemplo um calculo ou receber um dado de um arquivo aonde está a parte lógica para mostrar seu resultado no html.

II- A manipulação de dados entre os arquivos e as alterações de forma dinâmica.

III- **Repositório salvo:**

Interface gráfica do usuário, Texto, Aplicativo, Teams

Descrição gerada automaticamente

1. É uma das formas de fazer a comunicação que entre o HTML e TS aonde conseguimos ter a saída de um atributo do componente para o template, não é necessário a utilização de interpolação e pode-se usar p [].
2. Assim como one way data binding ele vai em uma direção apenas, do codigo para o HTML.
3. Criar no html uma frase com interpolação aonde o seu valor recebido será a variável dentro do componente criada no arquivo ts.
4. Event Binding são eventos de entrada através do html. Os nomes dos eventos são declarados dentro dos parênteses, ex: (click).
5. <button (click)= “clicar()”Clique aqui></button>
6. No html foi criada uma tag button com um evento click aonde ao clicar no botão é chamado a função clicar() e executado a ação contida na mesma;
7. Compartilhamento de dados entre os arquivos bem como a manipulação dos mesmos. Ex: alteração de dados de um produto.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. São tags personalizadas criadas para gerar uma determinados comportamentos.
2. Diretivas de atributos(altera a aparência ou comportamento do elemento ou outra diretiva), estruturais(modifica o layout adicionando ou removendo elementos do DOM) e componentes(diretivas como um modelo);

**NGSTYLE:** Utilizada para definir o o comportamento e estilização do elementos (cores, font-size, etc...).  
**NGSWITCH:** Estrutura de condição que cria scripts capazes de executar diferentes blocos de código de acordo com as condições.

**NGFOR**: percorre a lista de um array assim como o for para executar uma ação, assim com o For.  
**NGIF**: executa uma ação mediante a condição ser verdadeira, faz o papel do IF.  
**NGMODULE:** Ajuda na organização do aplicativo com a criação de blocos.

**NGCLASS**: Vincula dinamicamente uma ou mais classes css a um elementos HTML.

Texto

Descrição gerada automaticamente

1. Template-Drive:

Fazem uso dos “FormsModule”

São assíncronos

Reactive Foms:

“São baseados no ReactFormsModule”

São síncronos

1. Html mais clean, flexível e mais fácil de dar manutenção.
2. São necessários os módulos: FormsModule, ReactiveFormsModule e DataFormModule.
3. Nós temos o AbstractControl, FormControl, FormGroup, FormArray e FormBuilder.
4. Importe o ReactiveFormsModule na sua aplicação(ele possui todas a diretivas necessárias de um formulário reactive), criar um FormControl(para ser realizado o binding do formulário) o memo deve ser salvo no componente e setado no template.
5. Os formulários reativos são criados no próprio componente, através das linguagens TS e JS, as informações são sincronizadas com o DOM e o HTML , ou seja, a maior parte do código estará no componente e não no HTML.
6. Através dos dados inseridos o evento input captura essa informação e a mesma é atualizada no FormControl que por sua vez atualiza esse valor através do observables.
7. A validação de um formulário serve para garantir que as informações inseridas esteja de acordo com a solicitação garantindo assim que o formulário seja submetido somente quando os dados estiverem sido preenchidos de forma correta. Podemos deixar os campos como obrigatórios(ngrequired), mostrando mensagens para o usuário caso o mesmo não preencha os dados corretamente(ngdirty), inserindo cores nas bordas dos campos, habilitando o botão de envio apenas mediante todos os campos preenchidos e de forma correta entre outros.