Instituto Infnet

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CAROLINA ALVES DA ROCHA SILVA

TRABALHO PRÁTICO 1

Frameworks Front-end e conexão com Back-end

Prof. Vanessa Cristina Martins da Silva Frattini

Belo Horizonte
2022

CAROLINA ALVES DA ROCHA SILVA

TRABALHO PRÁTICO 1

Frameworks Front-end e conexão com Back-end

Trabalho da disciplina de Frameworks Front-end e conexão com Back-end para demonstrar as competências trabalhadas nas etapas 1 e 2.

Belo Horizonte
2022

SUMÁRIO

Questão 1

Letra A

O modelo MVVM é uma evolução do MVC, que separa as responsabilidades da aplicação entre Model, View e Controller. Quando se trata de aplicações web, o modelo MVC se torna custoso e utiliza muitos recursos, o que traz uma experiência lenta para o usuário. O padrão MVVM acaba trazendo melhoras na performance por responder mais rapidamente à interação do usuário.

A camada View é a responsável por exibir as informações para o usuário. A View-Model retém a apresentação dos dados da aplicação em determinado momento, guardando seus estados. A Model é a camada de persistência de dados e, em algumas aplicações, também contém as regras de negócios (em outras as regras de negócio aparecem na View-Model).

Letra B

As view-models são instanciados com o construtor ou suas sub-classes, como em:

```
var vm = new Vue({ /* opções */ })
```

Na view, cada instância do Vue está associada a um seletor correspondente:

```
vm.$el
```

Os dados em Vue são objetos do Javascript, assim uma model pode ser definida como:

```
new Vue({
    el: '#app',
    data: {
        title: "Bem vindo à locadora de filmes"
    }
})
```

Questão 2

O modelo MVVM é uma evolução do MVC, que separa as responsabilidades da aplicação entre Model, View e Controller. Quando se trata de aplicações web, o modelo MVC se torna custoso e utiliza muitos recursos, o que traz uma experiência lenta para o usuário. O padrão MVVM acaba trazendo melhoras na performance por responder mais rapidamente à interação do usuário.

Questão 3

Para ser considerada uma aplicação reativa, ela precisa:

- Observar as mudanças no estado
- Propagar essas mudanças por toda a aplicação
- Renderizar automaticamente as respostas a essas mudanças no estado
- Prover um feedback para as interações do usuário

Questão 4

Letra A

A diretiva *v-if*

Letra B

O *v-show* também pode ser utilizado para mostrar ou não um conteúdo. A diferença entre essas diretivas é que o *v-if* condicionalmente faz a renderização dos elementos, enquanto o *v-show* condicionalmente mostra os elementos. Isso significa que o *v-if* vai destruir e recriar os elementos quando a condição muda. O *v-show* sempre vai ter esses elementos na DOM, apenas removendo eles com CSS.

Questão 5

Usamos chaves duplas para mostrar variáveis no vue. Exemplo:

```
<p>{{ mensagem }}</p>
```

Questão 6

Letra A

O DOM-Virtual é uma abstração mais leve do DOM do browser criada pelo Vue para permitir alterações mais rápidas feitas diretamente no DOM-Virtual.

Letra B

O loop de eventos permite que o Vue monitore os dados da aplicação conforme as ações do usuário. Quando esses dados sofrem alterações, o DOM Virtual é renderizado novamente e o DOM do browser é atualizado em tempo real.

Questão 7

A diretiva *v-for* é usada para percorrer uma lista de objetos e performar determina ação para cada item da lista.

Questão 8

```
<div class="app">
    <ul>
        <li>{{ texto1 | capitalizar}}</li>
        <li>{{ texto2 | capitalizar}}</li>
        <li>{{ texto3 | capitalizar}}</li>
    </ul>
</div>

<script>
    new Vue({
        el: "#app",
        data: {
            texto1: "texto1",
```

```
            texto2: "texto2",
            texto3: "texto3"
        },
        filters: {
            capitalizar: (valor) => {
                return valor.toUpperCase();
            }
        }
    });
</script>
```

Questão 9

Se a diretiva v-html for utilizada para compor templates parciais, renderizando dinamicamente HTML sem precauções, pode permitir ataques XSS. Por isso é importante utilizar somente conteúdos confiáveis.

Questão 10

O código de resolução da questão 10 pode ser encontrado em Github [https://github.com/carolasilva99/infnet-frontend-vue].