



Instituto Infnet

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CAROLINA ALVES DA ROCHA SILVA

TRABALHO PRÁTICO 3

Frameworks Front-End e conexão com Back-end

Prof.

Belo Horizonte
2022

CAROLINA ALVES DA ROCHA SILVA

TRABALHO PRÁTICO 3
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 6 e 7.

Belo Horizonte
2022

SUMÁRIO

Questão 1.....	3
Questão 2.....	4
Questão 3.....	5
Questão 4.....	6
Questão 5.....	7
Questão 6.....	8
Questão 7.....	9
Questão 8.....	10
Questão 9.....	11
Questão 10.....	12
Letra A	12
Letra B	12
Letra C	12

Questão 1

Componentes devem ser utilizados porque ajudam a modularizar o código e evitar repetição do mesmo. Exemplo de criação de um componente simples:

```
1  <div id="app">
2  |   <numeros-positivos></numeros-positivos>
3  </div>
4
5  <script>
6  Vue.component('numeros-positivos', {
7  |   template: '<p>{{ numerosPositivos.length }} números positivos</p>',
8  |   data() {
9  |     return {
10 |       numeros: [-5, 0, 2, -1, 1, 0.5]
11 |     };
12 |   },
13 |   computed: {
14 |     numerosPositivos() {
15 |       return this.numeros.filter((numero) => numero >= 0);
16 |     }
17 |   }
18 | });
19
20 new Vue({
21 |   el: "#app"
22 | })
23 </script>
```

Questão 2

Utilizam-se as “props” para passar informações entre componente pai e componente filho, como mostrado na imagem abaixo:

```
Vue.component('exibir-preco', {  
  props: {  
    preco: Number,  
    unidade: String  
  }  
});
```

Não é possível passar informações de pai para filho.

Questão 3

O código da questão 3 pode ser encontrado [aqui](https://github.com/carolasilva99/infnet-frontend-vue/tree/main/TP3)
[https://github.com/carolasilva99/infnet-frontend-vue/tree/main/TP3].

Questão 4

O código da questão 4 pode ser encontrado [aqui](https://github.com/carolasilva99/infnet-frontend-vue/tree/main/TP3)
[https://github.com/carolasilva99/infnet-frontend-vue/tree/main/TP3].

Questão 5

Single Page Applications são aplicações web que não precisam recarregar a página completa durante as transições. O Vue Router permite que isso aconteça gerenciando as rotas no browser ao invés de requisitar o carregamento de páginas ao servidor quando uma URL muda.

Questão 6

O modo hash é o modo usado por padrão no Vue Router, que utiliza um jogo da velha (#) para gravar o caminho a ser renderizado, ou seja, para acessar a rota “/form” seria necessário digitar “http://localhost:8080/#/form”. Usando o modo history o jogo da velha deixa de ser necessário e, ao acessar a URL sem o #, a página continua funcionando e o browser não precisa renderizar novamente.

Questão 7


As rotas dinâmicas são muito utilizadas para páginas de consulta de determinado item. Em um supermercado online por exemplo, ela seria usada quando eu clico em um determinado produto para ver mais detalhes dele.

Questão 8

O Vuex é a biblioteca de gerenciamento de estados do Vue. Ela permite que determinados dados do “store” sejam compartilhados entre todos os componentes de uma aplicação. Ela é a versão do Vue do Redux do React. Ambas se baseiam numa arquitetura Flux, criada pelo Facebook. O Flux promove uma arquitetura unidimensional de dados que enviados das actions, para o dispatcher, para a store e por fim para a View, que é onde elas são exibidas. As operações síncronas podem ser feitas nas mutações do Vuex, enquanto as assíncronas devem ser feitas com Actions.

Questão 9

A store do Vuex deve ser criada em aplicações maiores e mais complexas em que determinados dados sejam necessários por toda a aplicação e não possam ser utilizadas as propriedades.



```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

Questão 10

Letra A

Precisamos criar o Store, que vai armazenar todo o estado do aplicativo em um objeto e manter a consistência de dados na aplicação.

Letra B

Precisamos criar as mutations e actions.

Letra C

Utilizamos o getter para realizar a busca.