**Projeto POKEDEX**

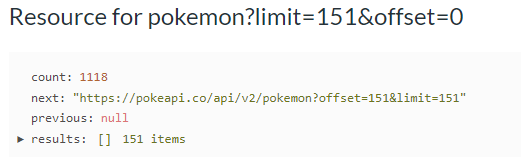
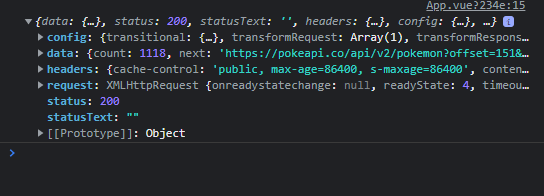
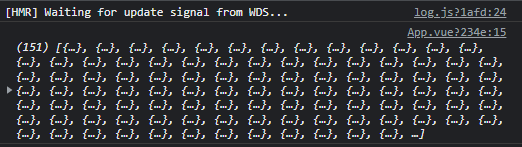
**Iniciando o projeto**

1. Neste projeto nós vamos recriar a pokedex do Pokemon no VueJs, o pokedex é um catálogo de todos os pokemons existentes na região.
2. Existe uma API na internet chamada pokeapi.co, que é gratuita e é ótima para testar aplicações FrontEnd.
3. Vamos começar criando o novo projeto VueJs
4. Na pasta onde o projeto será criado digite: vue create pokedexx
5. Após concluído a instalação vamos na pasta do projeto e vamos dar uma limpada no código.
6. O arquivo App.vue deve ficar da seguinte forma.
7. <template>
8. <div id="app">
10. </div>
11. </template>
12. <script>
13. export default {
14. name: 'App',
16. }
17. </script>
18. <style>
19. </style>
20. Durante a limpeza apagamos também o componente HelloWord

**Instalando as Bibliotecas**

1. **Biblioteca Bulma**
2. para o css do projeto
3. Parar o servidor dando cntr-C no terminal do vs-code, se estiver rodando, e na pasta do projeto digite: npm install bulma
4. **Biblioteca axios**
5. para permitir que sejam feitas requisições http ou seja requisições para outros servidores web. Adicionando o axios nós conseguimos fazer que a nossa aplicação converse com outras aplicações na web.
6. Na pasta do projeto digite: npm install axios .
7. Após a instalação das bibliotecas, rode novamente o servidor, na pasta do projeto digite:
8. Npm run serve
9. **Importando as bibliotecas**
10. No arquivo main.js importe a biblioteca bulma.
11. import "../node\_modules/bulma/css/bulma.css"
12. No arquivo App.vue importe a biblioteca axios, lembrando que esta biblioteca deve ser importância em todos os componentes que utilizarão este recurso.
13. import axios from 'axios';
14. nesse momento deve aparecer um erro no projeto, pois a variável criado axios foi importada porém ainda não está sendo utilizada, para resolver este problema momentaneamente basta comentar a linha //import axios from ‘axios’;

**Pegando os dados dos Pokemons**

1. Para darmos continuidade ao projeto devemos pegar os dados dos pokemons, para isso temos que ir até a página da pokeApi para configurarmos a quantidade pokemons que iremos pegar por vez. Então na pagina <https://pokeapi.co/>
2. Digite no campo: pokemon?limit=151&offset=0 e clicke em submit.
3. Esse comando irá fazer uma requisição para a rota pokemon passando os atributos limit e offseet permitindo os pokemons até o número 151 começando do zero.
4. 
5. Essa consulta retornará
6. 
7. Vamos pegar essa lista de 151 pokemons e vamos enviar para nossa aplicação utilizando o axios para fazer uma requisição http para a url utilizada no item 3.
8. No componente App vamos descomentar a linha que importa o axios.
9. Para fazer uma requisição http nós devemos chamar esta requisição dentro de um método.
10. Como as informações dos pokemons são vitais para a aplicação, estas informações devem estar disponíveis na inicialização da aplicação, ou seja toda vez que a página for carregada com o componente, para isso vamos utilizar o método chamado create que irá fazer uma requisição com o axios.
11. Digite o código como abaixo
12. export default {
13. name: 'App',
14. created: *function*(){
16. }
18. }
19. Dentro do método create vamos utilizar o axios e os verbos http para fazermos a requisição dos pokemons. Neste caso vamos utilizar o verbo get pois queremos somente pegar os pokemons.
20. created: *function*(){
21. axios.get()
22. }
23. Dentro da chaves do get vamos inserir a url que será utilizada para a requisição, retornando um promisse com a resposta.
24. created: *function*(){
25. axios.get("https://pokeapi.co/api/v2/pokemon?limit=151&offset=0").then(*res* *=>* {
26. console.log(*res*);
27. });
28. }
29. Abrindo o console na página do projeto é possível ver todos os dados que são carregados com a resposta.
30. 
31. Porém vamos selecionar somente os pokemons melhorando a chamada da resposta no console.log
32. created: *function*(){
33. axios.get("https://pokeapi.co/api/v2/pokemon?limit=151&offset=0").then(*res* *=>* {
34. console.log(*res*.data.results);
35. });
36. }
37. 
38. Dentro do escopo de exports defaults{ }, vamos pegar todos os itens da resposta criando uma variável dentro da função dados, lembrando que esta função sempre retorna um objeto.
39. Esta variável será pokemons e irá receber um array.
40. data(){
41. return{
42. pokemons: []
43. }
44. }
45. No método create vamos passar o array que foi capturado pela requisição http para a variável pokemons que nós criamos.
46. created: *function*(){
47. axios.get("https://pokeapi.co/api/v2/pokemon?limit=151&offset=0").then(*res* *=>* {
48. console.log(*res*.data.results);
49. this.pokemons = *res*.data.results;
50. });
51. }
52. Como sabemos que a requisição está sendo feita corretamente, vamos alterar a mensagem do console.log para “Pegou a lista de pokemons”
53. created: *function*(){
54. axios.get("https://pokeapi.co/api/v2/pokemon?limit=151&offset=0").then(*res* *=>* {
55. console.log("Pegou a lista de pokemons");
56. this.pokemons = *res*.data.results;
57. });
58. }