**Aplicação desenvolvida para vaga de dev front-end jr**

A aplicação consiste em dois campos do tipo '*select*', onde o segundo campo é preenchido de acordo com a seleção do primeiro, ou seja, ao selecionarmos um continente, o segundo campo é preenchido automaticamente com os países referentes ao continente em questão.

**Opções tomadas durante o desenvolvimento**

Durante o desenvolvimento optou-se pelo uso do *const* ao invés do *var*, devido que o const não permite que a variável seja reatribuída, sobrescrita, ou seja, quando definidos uma variável com o const, não podemos reatribuir um valor para ela, com ocorre com o uso do *var*. Mas pode-se observar que para objetos o const permite que modifiquemos o objeto, mas não nos permite re-declarar o mesmo. Além de garantir o escopo de bloco, coisa que não acontece com o var, gerando vários erros no código.

**Requisição e preenchimento do ‘select’ pelo Javascript**

No nosso arquivo ‘index.html’, importamos a biblioteca *axios* para nos auxiliar no processo de requisição de dados. Com o *axios* no nosso projeto, vamos para o ‘index.js’ realizar a requisição.

A requisição por sua vez, é do tipo ‘GET’, pois estamos solicitando dados do servidor, onde recebemos dados no formato json (JavaScript Notation Object). Para que seja possível manipular esses dados no nosso arquivo, definimos um objeto global *data* (vazio) e colocamos os dados recebidos da requisição dentro desse objeto. Assim, já estamos prontos para realizar a manipulação dos dados.

Definimos então nossa função showContinents, que recebe como parâmetro os dados recebidos na requisição para que possamos fazer a população do primeiro campo *select*. Na função, selecionamos o nosso elemento HTML com a DOM. Como nossos dados estão dentro um array, utilizamos o método *map*, para iterar em cima de cada objeto e coletar o nome de cada objeto, através do cont.name. Na função que vai dentro do método *map*, nós criamos o elemento *option* para cada um dos continentes presentes nos nossos dados e adicionamos ele ao nosso elemento HTML select através do *appendChild().*

Já podemos observar que temos o primeiro select preenchido com dados. Vamos ao segundo:

Para que seja possível popular nosso segundo select, é necessário que sejamos capaz de ‘percerber’ a seleção do usuário no primeiro campo, para isso utilizamos a propriedade *onchange*, que tem uma função passada como parâmetro.

Essa função vai ser responsável por enviar para nosso javascript a seleção que o usuário fez, e assim podemos ser capazes de selecionar os países corretos para popular o nosso segundo campo.

Dentro dessa função fazemos a seleção dos dois campos select, onde no primeiro select acessamos o value do elemento. Logo após, fazemos uma verificação no segundo campo, para sabermos se ele está vazio ou não, caso esteja com conteúdo o mesmo é removido através de um loop for, para que possamos inserir países de outro continente sem que haja o acúmulo dos países e nos retorne dados errados.

Logo após essa verificação, fazemos uma iteração no nosso objeto, para da mesma forma que fizemos para o primeiro campo, possamos popular nosso segundo campo, mas agora com uma condição, só os países que estão contido no continente selecionado serão adicionados ao nosso segundo campo.

Para enriquecer um pouco a aplicação, utilizaremos uma API que nos fornecerá dados dos países selecionados.

Da mesma forma que realizamos a requisição ‘GET’ anteriormente para coletar os dados dos continentes, agora faremos o mesmo tipo de requisição para receber os dados dos países, passando como parâmetro para a API o nome do país que desejamos receber as informações.

Após receber os dados, fazemos o tratamento do mesmo, para selecionar apenas os dados que são do nosso interesse, que são a capital de cada país, a bandeira e a população.

Selecionados os dados, está na hora que mostrarmos ao usuário um pequeno card com essas informações. Onde fazemos a seleção dos devidos elementos onde cada dado deve estar inserido, usando a DOM, após essa seleção utilizamos o setAttribute para o elemento *img*, e para os demais, já que são strings, utilizamos o *innerHTML*.

Importante notar que o card não está disponível quando abrimos nossa aplicação, ele fica disponível apenas depois de selecionarmos o nosso continente desejado.