

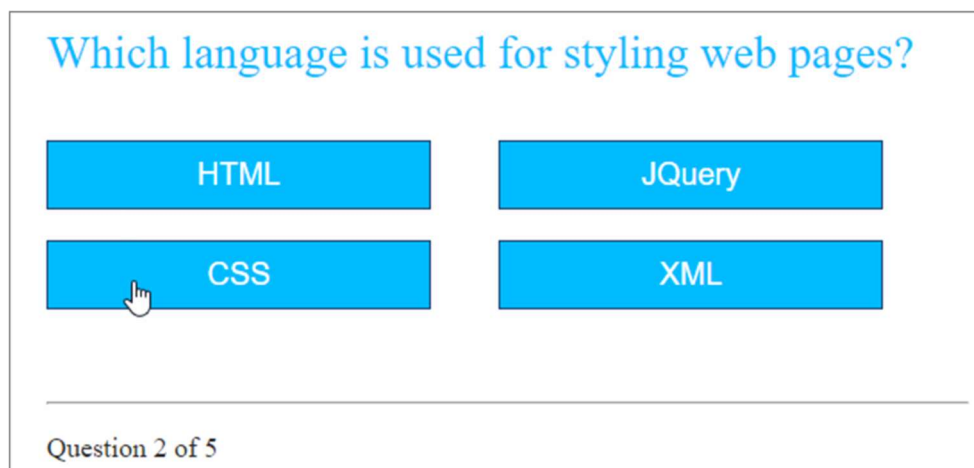
Jogo: Bolsonaro: verdadeiro ou falso?

Escondendo elementos da DOM com VueJS

Bolsonaro: verdadeiro ou falso? é um quiz game, o tradicional jogo de perguntas e respostas que, na prática, demanda conhecimentos básicos da lógica de programação, sendo construído em cima de comandos de seleção (if/else) e do uso de contadores.

Mas aqui foi proposto um desafio adicional. O objetivo era que cada pergunta tivesse um visual ligeiramente diferente, usando fotos, vídeos e conteúdo compartilhado em redes sociais, ou seja, alternando diferentes elementos. Isto exigia a criação de um código próprio, porque, geralmente, os quizzes costuma repetir uma mesma interface gráfica.

Por exemplo, no script disponibilizado no site Web Dev Trick¹, todas as perguntas tem exatamente o mesmo formato



Tela 1: layout do quiz do site WebdevTrick

Isto é feito, no Javascript, através de arrays, como mostramos no código abaixo: a pergunta, as opções e a resposta correta ficam dentro do array e o script vai passando as perguntas, uma a uma.

¹ Disponível em < <https://webdevtrick.com/create-javascript-quiz-program/>>.

```
// create questions here
var questions = [
  new Question("Hyper Text Markup Language Stand For?", ["JavaScript",
"XHTML", "CSS", "HTML"], "HTML"),
  new Question("Which language is used for styling web pages?", ["HTML"
, "jQuery", "CSS", "XML"], "CSS"),
  new Question("Which is not a JavaScript Framework?", ["Python Script"
, "jQuery", "Django", "NodeJS"], "Django"),
  new Question("Which is used for Connect To Database?", ["PHP", "HTML"
, "JS", "All"], "PHP"),
  new Question("Webdevtrick.com is about..", ["Web Design", "Graphic De
sign", "SEO & Development", "All"], "All")
];
```

Código 1: array para reúne dados do quiz

A mesma fórmula é encontrada em diversos outros scripts disponíveis na internet, como o do Sitepoint².

Em *Bolsonaro: verdadeiro ou falso?* por conta da necessidade de inserir vídeos e scripts na página o objetivo era pré-carregar blocos distinto de código HTML. A ideia pode ser resolvida em Javascript carregando tags <DIV> ocultas, que vão sendo exibidas conforme o usuário vai respondendo as perguntas.

Para simplificar este código, que exigiria manipular múltiplas vezes o comando `document.getElementById("NOMEDADIV").style.display`, optou-se por adotar por um framework que reduzisse esta digitação de código.

Optou-se por usar o framework VueJS, que tem como uma de suas características a uso de um DOM Virtual, recurso que permite renderizar trechos de código antes de exibir na DOM, ou seja, na tela do navegador. Desta forma, todas as cinco perguntas, cada uma com sua formatação, carregariam ao iniciar a página.

O DOM Virtual foi usado em diversos trechos do código. Por exemplo, o trecho abaixo (ainda numa versão provisória do código) exibe o resultado do quiz.

```
<p>Você acertou {{ acertos }} de 5 perguntas</p>
<template v-if="acertos==5">Mito</template>
<template v-else-if="acertos==4">Fraquejada</template>
<template v-else-if="acertos==3 || acertos==2">Tem que melhorar isso aí, talkei?</template>
<template v-else>Ponta da Praia</template>
```

Código 2: trecho do código HTML, com VueJS, que usa o VirtualDOM.

² Disponível em <<https://www.sitepoint.com/simple-javascript-quiz/>>.

No código acima temos a tag `<template>`, que é uma tag subutilizada do HTML5, que mostra um container cujo elemento não é renderizado, ou seja, que não é exibido na tela. No VueJS usamos ela com maior frequência: todos os elementos entre `<template>` e `</template>` são renderizados pela Virtual DOM, sendo exibidos quando solicitados.

O código exibe ainda as diretivas `v-if`, `v-else-if` e `v-else` do VueJS. Assim, como no tradicional `if/else`, o VueJS vai testar a condição, no caso o número de acertos no quiz, e só apresentará na DOM a condição que for atendida.

Se o usuário, pelas ferramentas de desenvolvedor do Chrome, olhar este trecho do código, só verá uma linha, da condição que foi atendida:

```
<p>Você acertou 1 de 5 perguntas</p>  
" Ponta da Praia" == $0
```

Código 3: Do código 2, a DOM exibe apenas a condição atendida do código.

A ideia era repetir esta técnica em toda a aplicação. Considerando que o código tem sete áreas distintas (uma introdução, as 5 perguntas e a tela com as respostas, cada uma delas ficaria em uma seção correspondente (ou seja, delimitada em uma tag `<SECTION>` e `</SECTION>`). Estas telas seriam todas carregadas de uma única vez, sendo exibidas conforme o usuário fosse avançando as perguntas.

Mas o plano não foi possível de ser executado porque uma das perguntas do quiz consistia em fazer o usuário interagir com uma mensagem publicada na conta do presidente Jair Bolsonaro na rede social Twitter.



Tela 2: Tweet do presidente

O problema ocorreu porque incorporar um tweet consiste em agregar o seguinte código na página:

```
<blockquote class="twitter-tweet"><p lang="pt" dir="ltr">- Em torno de 40% do Fundo Amazônico vai para as... ONGs, refúgio de muitos ambientalistas. Veja a matança das baleias patrocinada pela Noruega. <a href="https://t.co/46hpQnHSJA">pic.twitter.com/46hpQnHSJA</a></p>&mdash; Jair M. Bolsonaro (@jairbolsonaro) <a href="https://twitter.com/jairbolsonaro/status/1163242553837330432?ref_src=twsrc%5Etfw">August 19, 2019</a></blockquote> <script async src="https://platform.twitter.com/widgets.js" charset="utf-8"></script>
```

Código 4: código para incorporar uma mensagem do Twitter numa página.

No lugar de trabalhar com uma tag embed ou um iframe, que são formas mais tradicionais de incorporar um conteúdo externo em uma página, o Twitter disponibiliza o texto da mensagem dentro de uma tag de citação, a <BLOCKQUOTE> e configura seu layout através de um script externo (incluída na última linha do código, onde se lê <script async src="https://platform.twitter.com/widgets.js" charset="utf-8"></script>).

O Virtual DOM do VueJS é capaz de manipular códigos HTML e CSS mas não processar (ou pré-processar) um código Javascript externo, de forma que todas as tentativas de inserir este tweet da página e só renderizar ele posteriormente deram errado.

Para contornar o problema buscou-se para um solução mais simples, que é o uso da diretiva v-show do VueJS. O v-show é um comando que permite ao programador manipular qualquer tag do HTML, escondendo ou exibindo ela via CSS usando os parâmetros true ou false. Ou seja, ele funciona de forma semelhante a ocultar um elemento da página usando o document.getElementById("NOMEDADIV").style.display do Javascript.

```
<section v-show="area[1]">
  <div style="float:left;width: 50%;">
    <h2>"Em 2013, um acordo entre o governo petista e a ditadura cubana trouxe ao Brasil 10 mil médicos sem nenhuma comprovação profissional. Foram impedidos de trazer cônjuges e filhos, tiveram 75% de seus salários confiscados pelo regime e foram impedidos de usufruir de direitos fundamentais, como o de ir e vir." </h2>
  <p>
```

Código 5: exemplo do uso do v-show, ocultando uma das questões do quiz.

Desta forma, o eixo central da nossa aplicação, as perguntas do quiz, estão todas carregadas na página inicial, apenas ocultas, configuradas com o parâmetro v-show de cada uma delas setado como falso. Conforme o usuário avança, o script oculta a pergunta já respondida e exhibe a pergunta a ser respondida até que, no final, todas as perguntas e respostas se abrem para que ele possa revisar o que errou e o que acertou.

Bolsonaro: verdadeiro ou falso? foi construído, portanto, no uso alternado destas duas possibilidades que o VueJS apresenta: guardar trecho de código no Virtual DOM ou esconder elementos na manipulação do CSS. Não era a intenção inicial do projeto alternar entre estes dois recursos, mas só com ela foi possível superar uma dificuldade técnica que surgiu durante o desenvolvimento deste game.