

KAUÊ OLIVEIRA DRIGO

QUIZ PARA APRENDIZAGEM EM SOLDAGEM



**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA**

2018

KAUÊ OLIVEIRA DRIGO

QUIZ PARA APRENDIZAGEM EM SOLDAGEM

Trabalho de conclusão de curso apresentado à Faculdade de Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **Bacharel em Engenharia Mecânica**.

Área de concentração: Soldagem

Orientador: Prof. Dr. Volodymyr Ponomarov

UBERLÂNDIA – MG

2018

AGRADECIMENTOS

Ao professor Volodymyr, por ter confiado em minha capacidade para assumir este projeto, por estar sempre disponível, e pela atenção e cuidado dedicados tanto a mim quanto ao trabalho.

Ao meu irmão, por ter me orientado nos primeiros passos.

DRIGO, K. O., Quiz da Soldagem. 2018. 28 p. Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia, Uberlândia.

Resumo

O objetivo é criar um jogo no formato Quiz para web com a temática sobre Soldagem, a ser utilizado como ferramenta de aprendizagem e avaliação de conhecimento. O jogo foi desenvolvido com HTML, CSS e JavaScript. Como resultado, a aplicação possui vinte níveis de perguntas de dificuldade progressiva, que podem ser respondidas com ou sem tempo limitado, e animações que interagem com o desempenho do jogador.

Palavras-chave: *Quiz, Soldagem, Aprendizagem.*

DRIGO, K. O., Quiz da Soldagem. 2018. 28 p. Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia, Uberlândia.

Abstract

The goal is to create a Quiz type game for web about Welding, to be utilized as learning tool and evaluation of knowledge. The game was developed using HTML, CSS, and JavaScript. There are twenty question levels in the final product, with progressive difficulty, which can be answered in limited or unlimited time, and animations that Interact with player's performance.

Keywords: *Quiz, Welding, Learning.*

SUMÁRIO

Capítulo 1 –	Introdução	1
Capítulo 2 –	Revisão Bibliográfica	3
2.1 –	HTML.....	3
2.2 –	CSS.....	4
2.3 –	JavaScript.....	5
Capítulo 3 –	Metodologia.....	6
3.1 –	Organização dos arquivos	6
3.2 –	Código	7
Capítulo 4 –	Resultados e Discussão.....	10
4.1 –	Características.....	11
4.1.1 –	Jogabilidade.....	11
4.1.2 –	Design.....	12
4.2 –	Alterações no quiz	13
4.2.1 –	Banco de perguntas	13
4.2.2 –	Imagens	17
4.3 –	Teste de Campo	17
Capítulo 5 –	Conclusão	20
Capítulo 6 –	Referências Bibliográficas	22

CAPÍTULO 1

INTRODUÇÃO

Jogar é uma atividade que faz parte da vida humana há milhares de anos. Embora o início dessa atividade não possa ser exatamente especificado, sabe-se que desde o seu princípio ela passou por diversas modificações que refletiam o contexto cultural envolvido, com prática adotada por todas as faixas etárias.

Os jogos sempre foram importantes para transmitir valores da sociedade, e servir como ferramenta de formação do indivíduo. Quando crianças brincam, por exemplo, de compradores e vendedores, eles estão utilizando, de maneira lúdica, princípios que serão importantes na vida adulta. Também é possível assimilar habilidades como trabalho em equipe, capacidade de adaptação, criatividade, entre outras.

Um dos aspectos mais importantes de um jogo é a possibilidade de criar um cenário fictício, que possui suas próprias regras, portanto um mundo a parte, onde os jogadores podem experimentar situações que de outra maneira seriam inacessíveis. Esse é um dos motivos pelos quais sua utilização é muitas vezes recreativa, já que, segundo GOFFMAN (1959), o indivíduo geralmente vive em uma rotina rígida, com várias idealizações a serem seguidas, quando escondem vontades que tentam ser satisfeitas, e o jogo é um ambiente seguro para tal. Também é o caso citado de crianças imitando a vida adulta, ou ainda pode ser o caso de um profissional que deseja treinar o seu ofício, e aqui é onde se torna interessante esse aspecto dos jogos.

Embora o caráter educacional sempre estivera presente nos jogos, esse aspecto recebeu pouca atenção no passado. Eis que com o advento do mundo virtual, dado a

popularização dos computadores, outra espécie de jogo surgiu, o *Serious Games*, definido por DÖRNER et al. (2016) como jogos eletrônicos criados com ao menos uma intenção além do entretenimento, seja para aprendizagem, saúde, motivação, etc. As novas tecnologias permitem criar novos ambientes, com novas regras, que facilitam a imersão do jogador (PROTOPSALTIS et al., 2010a) e resultam em experiências similares ao real.

Nesse novo contexto, muitos desenvolvedores começaram a criar ambientes específicos para desenvolver habilidades específicas. É o caso do treino de pilotos de Fórmula 1 em simuladores, exemplo que foi estendido à formação de novos condutores no Brasil. Essa nova prática se torna atraente pela possibilidade de manter a motivação do jogador com a introdução de *feedbacks* constantes, e do progresso do jogo estar atrelado sempre a ações ativas (PROPOSALTIS et al., 2010b).

Essa alternativa está gradualmente entrando na área de ensino, a qual carece de novas ferramentas para aprendizagem que coincidam com os novos perfis de alunos. A aplicação deste tipo de jogo transcende o entretenimento, aproveitando seu caráter lúdico para ensinar. Não são conhecidas aplicações similares na área de Soldagem. Assim, o objetivo é criar um jogo tipo quiz para testar o conhecimento dos alunos da área de Soldagem, explorando a curiosidade e competitividade destes para os estimular a buscarem respostas e assim melhorarem seu conhecimento.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

A internet surgiu nos anos 1960 como um projeto militar durante a guerra fria, no intuito de ser uma ferramenta de comunicação dentro dos Estados Unidos que sobrevivesse a possíveis ataques soviéticos às linhas de comunicações vigentes na época, que se concentravam no Pentágono. Aos poucos a internet foi liberada a acadêmicos, que a utilizaram como meio de trocar informações científicas. As linguagens de programação mais usadas pela Internet são HTML, CSS, JavaScript cujas sucintas descrições são apresentadas a seguir.

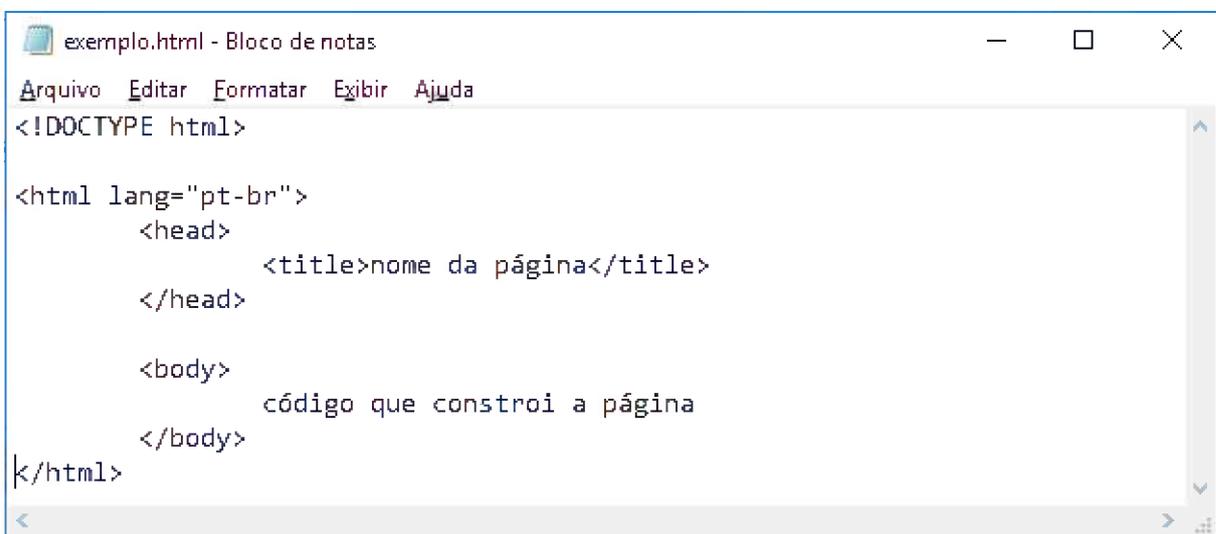
2.1 HTML

Em 1990, o cientista Tim Berners-Lee, procurando uma alternativa para melhorar a troca de informações entre acadêmicos, desenvolveu o HTML, uma linguagem que permitia a criação de páginas onde as informações seriam melhores exibidas visualmente. HTML é um acrônimo para *HyperText Markup Language* (Linguagem de Marcação de HiperTexto). Talvez a contribuição mais importante do HTML seja a utilização do *hypertext*, isto é, até então todo o caminho de acesso na internet era linear. Para Tim, a utilização de *links* de hipertexto parecia possível, a questão era apenas encontrar a maneira correta de implementar (ADDISON WESLEY LONGMAN, 1998). Com a utilização do *hypertext* as páginas passaram a referenciar outras páginas, criando um caminho dinâmico de informação.

O HTML não é uma linguagem de programação no termo convencional. Seu papel é marcar o texto, separando-o por categorias com auxílio de *tags* que são interpretadas pelos navegadores, que por sua vez aplicam o resultado visual no texto. Atualmente as *tags* também

possuem a função de dizer aos mecanismos de busca (Google, Bing, Yahoo, etc.) o que está contido nas páginas, fornecendo informações tais como título e idioma.

A estrutura de um documento HTML é simples e eficaz, um dos principais motivos por ter sido adotada como o padrão nas páginas de internet. Figura 2.1 mostra a estrutura básica de um documento HTML. A *tag* DOCTYPE informa ao navegador que tipo de documento ele está prestes a ler; a *tag* html determina a região em que será escrito em linguagem HTML, por meio de uma *tag* de abertura e outra de fechamento; a *tag* head guarda informações do documento, como título e autoria; dentro da *tag* body está localizado todo o texto que aparecerá na página, que também pode ser incluído dentro de outras *tags* para indicar parágrafos, alinhamento do texto, salto de linha, etc.

A screenshot of a Notepad window titled "exemplo.html - Bloco de notas". The window contains the following HTML code:

```
<!DOCTYPE html>

<html lang="pt-br">
  <head>
    <title>nome da página</title>
  </head>

  <body>
    código que constroi a página
  </body>
</html>
```

Figura 2.1 – Estrutura básica de um documento HTML

É comum utilizar junto ao HTML outras linguagens para aperfeiçoar a página, seja no aspecto visual ou no aspecto funcional. Também é uma boa prática separar em arquivos diferentes os códigos de cada parte. Assim, a estrutura da página é o arquivo principal HTML, que invoca os demais, e os secundários fazem modificações nesse. A linguagem CSS é utilizado para dizer como os elementos devem ser exibidos, e conta com recursos que permitem dinamização visual. Já o JavaScript é uma linguagem para mudar o comportamento da página, permitindo interação dessa com o usuário.

2.2 CSS

O CSS (*Cascading Style Sheets*) nasceu da necessidade dos programadores para web de controlar a aparência de suas criações. Até então, o HTML permitia apenas que a informação fosse estruturada. Tentativas foram feitas para agregar ao HTML mudanças de estilo, que logo se mostraram impraticáveis em caso de códigos muito grandes e complexos.

Tim Berners-Lee considerava ser função de cada navegador decidir pelas configurações de estilo, mas o que seguiu foi a oferta cada vez menor de opções nas propriedades visuais das páginas (BOS, 2016). Em 1994 Håkon Wium Lie desenvolveu o primeiro esboço do CSS, que depois de muita discussão e desenvolvimento, foi finalmente bem implementado no Internet Explorer 3 em 1996, para se tornar nos anos seguintes um padrão para todos os *browsers*.

A adoção do CSS se deve principalmente por permitir a separação da estrutura de um site e seu estilo em arquivos diferentes. Essa característica possibilita que alterações na visualização sejam executadas sem risco de comprometer o conteúdo. Além disso, o CSS pode usar as próprias *tags* do HTML para modificar, por exemplo, todos os parágrafos de uma vez, ou ainda modificar algum elemento específico pela utilização de uma *id*. No caso de várias páginas que utilizem o mesmo *layout*, basta que os vários arquivos HTML referenciem o mesmo arquivo CSS, evitando repetições desnecessárias. Também há a possibilidade de usar mais de um arquivo para definir a aparência de um site.

2.3 JavaScript

O primeiro desenvolvimento do JavaScript se deu em um período de semanas. Ele foi concebido às pressas por Brendan Eich à serviço da Netscape, que na época dominava o mercado de *browsers*. A ideia do projeto era a criação de uma linguagem para promover interação dos usuários com os *websites*, como animações, botões e jogos (MDN WEB DOCS), que fosse sintaticamente semelhante ao Java, e simples o suficiente para ser utilizada por desenvolvedores amadores em conjunto com o HTML.

Sua primeira implementação foi integrada ao Netscape em 1995, e permitiu que se interagisse com as páginas de web. O impacto foi tão grande na experiência dos usuários que os competidores foram forçados a reagir e aplicar o JavaScript ou cópias aos seus próprios produtos. Não demorou muito para que essa linguagem galgasse o posto de padrão em todos os navegadores.

Atualmente o JavaScript é muito aplicado em forma de derivações, como o AJAX e o JQuery, bibliotecas bem elaboradas que executam funções complexas, enquanto facilitam o desenvolvimento do código.

CAPÍTULO 3

METODOLOGIA

O quiz foi desenvolvido utilizando a tríade HTML-CSS-JavaScript, escolha efetuada conforme o conhecimento do autor. Seguem os detalhes da concepção e desenvolvimento do projeto, que será apresentado como resultado final no capítulo seguinte.

3.1 Organização dos arquivos

A primeira etapa do desenvolvimento se preocupou com a organização dos arquivos. Eles estão dispostos conforme mostrado na Figura 3.1.

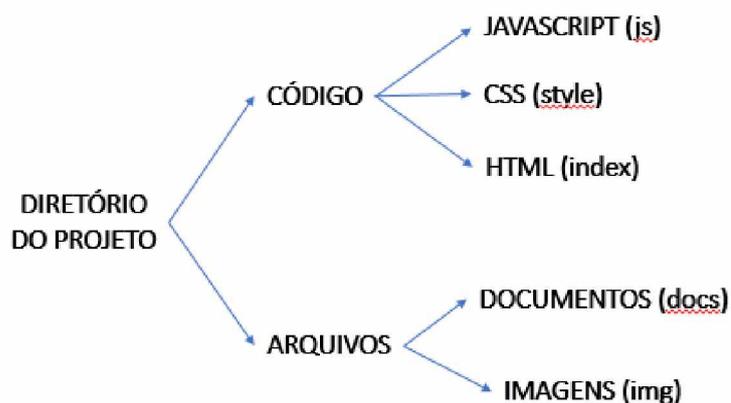


Figura 3.1 – Organização dos arquivos

A preocupação foi permitir modificações posteriores localizadas. Assim, o projeto foi organizado na separação do código com os demais arquivos. O diretório “**docs**” guarda instruções do projeto e diretrizes discutidas durante seu desenvolvimento, tais como número de perguntas e esboços do *layout*. O diretório “**img**” contém todas as imagens que serão

utilizadas no quiz, e um *backup* de versões anteriores. O diretório “js” armazena todo o código em JavaScript, incluindo as perguntas e respostas. No diretório “style” estão as definições visuais, que incluem cor das letras, cor de fundo, etc. Por último, no diretório raiz está o arquivo “index”, o HTML que comanda todo o jogo, e abrindo-o, o quiz é iniciado.

3.2 Código

No desenvolvimento do código é comum fazer uso de alguma Interface de Desenvolvimento (IDE). Foi utilizado o Visual Studio (Fig. 3.2). Esse programa possui uma região principal para a visualização do código, e ajuda a escrevê-lo facilitando algumas atividades, como por exemplo a inclusão de símbolos que abrem e fecham comandos (parêntesis, chaves, etc.). À direita há a organização do projeto. É possível navegar nas pastas e acessar qualquer arquivo, os quais podem ser abertos simultaneamente pelo sistema de abas.

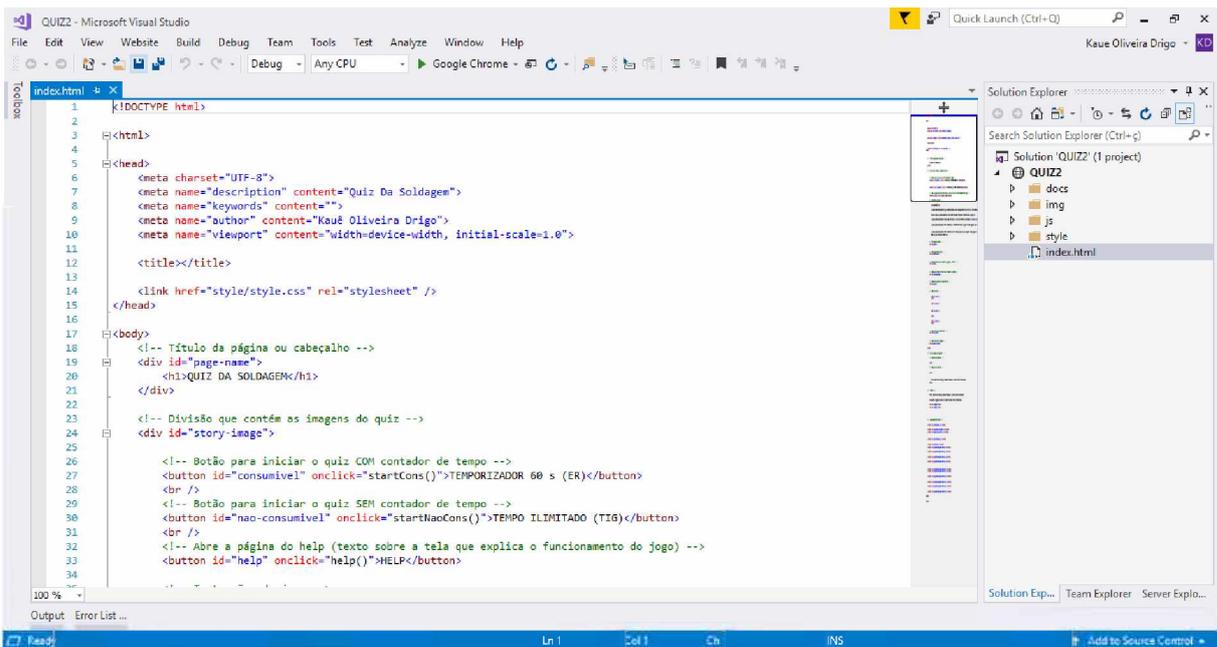


Figura 3.2 – Visual Studio

Foi feita a escolha por desenvolver o quiz em uma única página. Isto é, ao clicar em um botão ou avançar de etapa, ao invés de ser direcionado para outro endereço que contenha a função requisitada pelo usuário, o quiz faz modificações na página utilizando o JavaScript. Dessa maneira, o código HTML teve como principal função ser referência, portanto Fig. 3.3 representa a estrutura recorrente em seu corpo.

```

<!-- Personagem soldador -->
<div id="welder">
</div>

<!-- Personagem Darth Vader -->
<div id="darth-vader">
</div>

<!-- Tempo restante para responder a pergunta, se houver -->
<div id="time">
</div>

<!-- Animação do tempo restante para responder a pergunta -->
<div id="time-animation">
</div>

<!-- Animação na ponta da tocha/eletrodo -->
<div id="spark">
</div>

```

Figura 3.3 – Fragmento do corpo HTML

Foram criadas diversas *div*, regiões determinadas no programa, cujas características são ditadas pelo CSS, e posteriormente modificadas pelo JavaScript, se necessário. Todo o quiz está dentro de duas *div* principais. Um deles contém o apelo visual, enquanto a outra localiza as perguntas e respostas, portanto o quiz de fato. Dentro dessas estão localizadas muitas outras, com objetivos específicos, a exemplo das personagens, a animação de tempo, e os balões de fala.

O código CSS tem estrutura representada em Fig. 3.4. A indicação *#c-image1* faz referência ao *id* de uma *div*, dizendo o que será modificado pelas indicações seguintes, que, por sua vez, determinam a posição, tamanho, e qual imagem será alocada nesta região. No caso, as definições se referem ao logo da UFU, presente no rodapé do quiz na página inicial. O padrão é mantido para os outros elementos, salvo as especificidades de cada um.

```

#c-image1 {
  width: 150px;
  height: 34px;
  position: absolute;
  left: 150px;
  top: 0px;
  background-image: url('../img/ufu.png');
  background-size: contain;
  background-repeat: no-repeat;
}

```

Figura 3.4 – Fragmento do código CSS

A parte escrita em JavaScript tem uma estrutura mais complexa se comparada às anteriores. Seu código foi separado em diversas partes, para permitir a alteração localizada das funções. A estrutura dessa organização conta com um código principal, que coordena todas as outras funções. Figura 3.5 mostra a disposição dos arquivos.

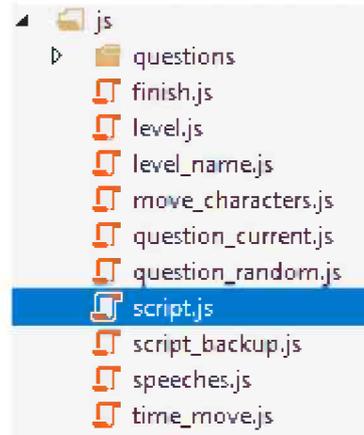


Figura 3.5 – Arquivos JavaScript

O arquivo em destaque é o principal. Ele coordena todas as modificações que podem ocorrer no quiz. Uma parte das ações está escrita nele, como a inclusão de várias imagens, enquanto outras partes, exemplo da movimentação das personagens, são ordenadas aqui, mas executadas por outras partes (no caso, pelo arquivo *move_character.js*). Dessa maneira, se por alguma razão for decidido que as personagens devem se movimentar de outra maneira, a dizer mais rapidamente, basta acessar o código específico e modificar a velocidade, sem risco de comprometer o código principal, onde estarão as condições para a movimentação.

CAPÍTULO 4

RESULTADOS E DISCUSSÃO

O resultado da concepção do quiz é apresentada em seguida, e pode ser vislumbrado em Fig. 4.1.

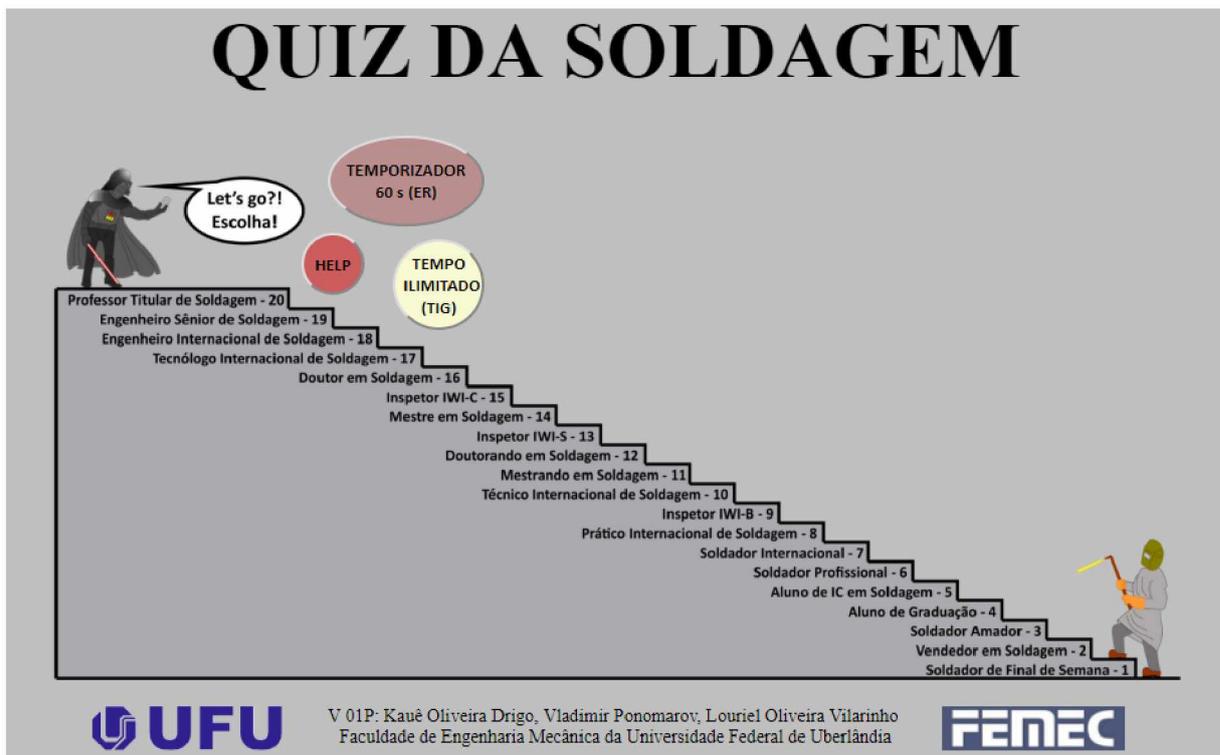


Figura 4.1 – Página inicial do quiz

4.1 Características

As características do quiz podem ser separadas em dois grupos: a jogabilidade e o design. Essa divisão corresponde aproximadamente às regiões inferior e superior da Fig. 4.2, respectivamente.

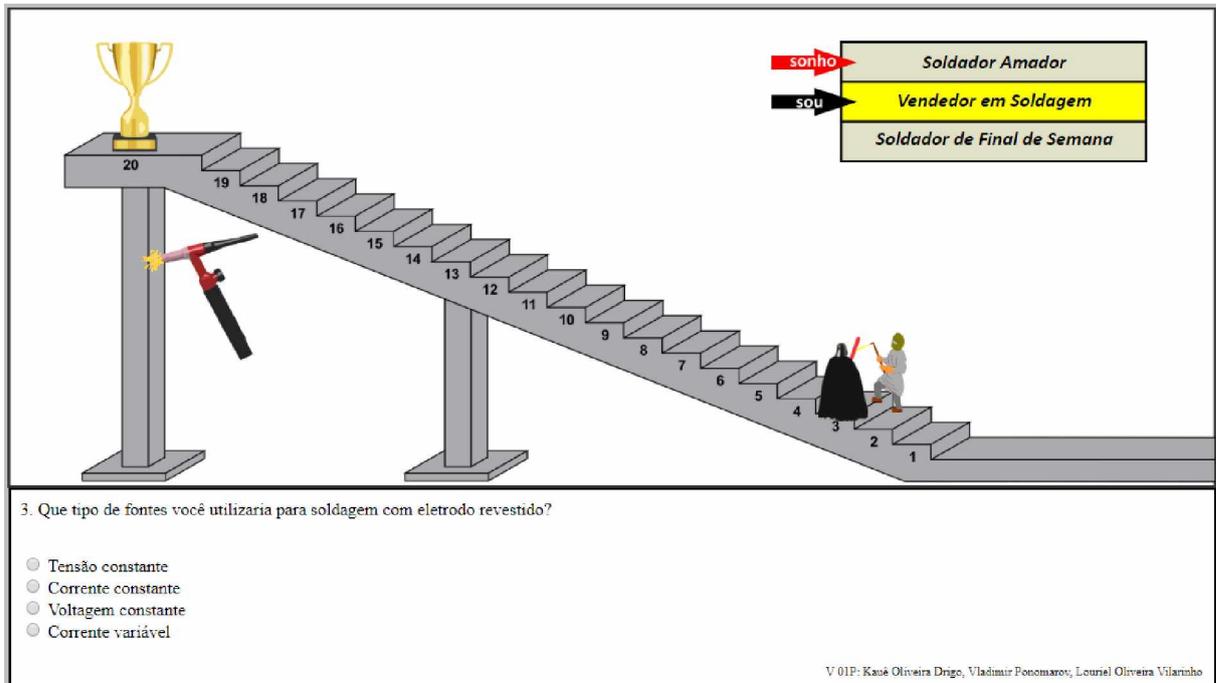


Figura 4.2 – Quiz da Soldagem

4.1.1 Jogabilidade

O quiz foi desenvolvido com vinte níveis de perguntas. Cada nível sucessivo corresponde ao aumento gradual da dificuldade, e possui um banco de aproximadamente dez perguntas, com quatro alternativas cada, sendo apenas uma correta, selecionadas sempre aleatoriamente. É possível escolher ainda se haverá tempo para selecionar a resposta certa (60 segundos), ou se o tempo é ilimitado. A primeira opção de tempo, chamada “Eletrodo Revestido”, faz referência à sua natureza consumível, e é ideal para testar o conhecimento do jogador. Já a segunda opção faz referência ao processo TIG, em que o consumo do eletrodo não é considerável, com objetivo de permitir ao jogador pensar e pesquisar a resposta, e usar o jogo como meio de estudar o conteúdo relacionado à soldagem nele contido.

As instruções do jogo podem ser acessadas pelo botão “HELP” em vermelho na página inicial (Fig. 4.1), que mostrará ao jogador Fig. 4.3. Basta selecionar, então, algum dos outros dois botões para iniciar o quiz (tanto na página HELP quanto na inicial).

INSTRUÇÕES

O jogo contém perguntas sobre soldagem em diversos níveis de dificuldade, acessados progressivamente à medida que o jogador acerta as perguntas do quiz. No total são 20 perguntas, com 4 alternativas de resposta cada, mas apenas uma delas está correta. Qualquer resposta errada finaliza o jogo.

Escolha uma alternativa, e pressione ENTER para confirmar a resposta.

TEMPO LIMITADO (ER: ELETRODO REVESTIDO): O jogador tem 60 segundos para responder cada pergunta. Caso o tempo se esgote, o jogo é finalizado, caracterizando derrota do jogador.

TEMPO ILIMITADO (TIG): O tempo disponível para responder cada pergunta é ilimitado. Modo de jogo ideal para estudar.



Figura 4.3 – Instruções do quiz

O quiz é finalizado quando uma resposta incorreta (ou nenhuma) é indicada, se o tempo para resposta esgotar (se a opção “TEMPORIZADOR” for selecionada), ou ainda se vinte perguntas são corretamente respondidas sucessivamente. Se não acontecer o último caso, o jogador poderá reiniciar o quiz com um *click* no balão representado por Fig. 4.4, que aparecerá nesta situação.



Figura 4.4 – Balão de reinício do quiz

4.1.2 Design

A parte visual do jogo busca engajamento do jogador, para tornar a atividade de aprender e/ou testar o conhecimento mais lúdica. Assim, já na página inicial é apresentado Fig. 4.5, que relaciona a dificuldade do nível da pergunta com um título relacionado à soldagem. Além disso, há também uma batalha intelectual entre o soldador e o famoso personagem Darth Vader. O jogador, representado pelo soldador, sobe a escada a cada nível vencido, tentando alcançar seu algoz.

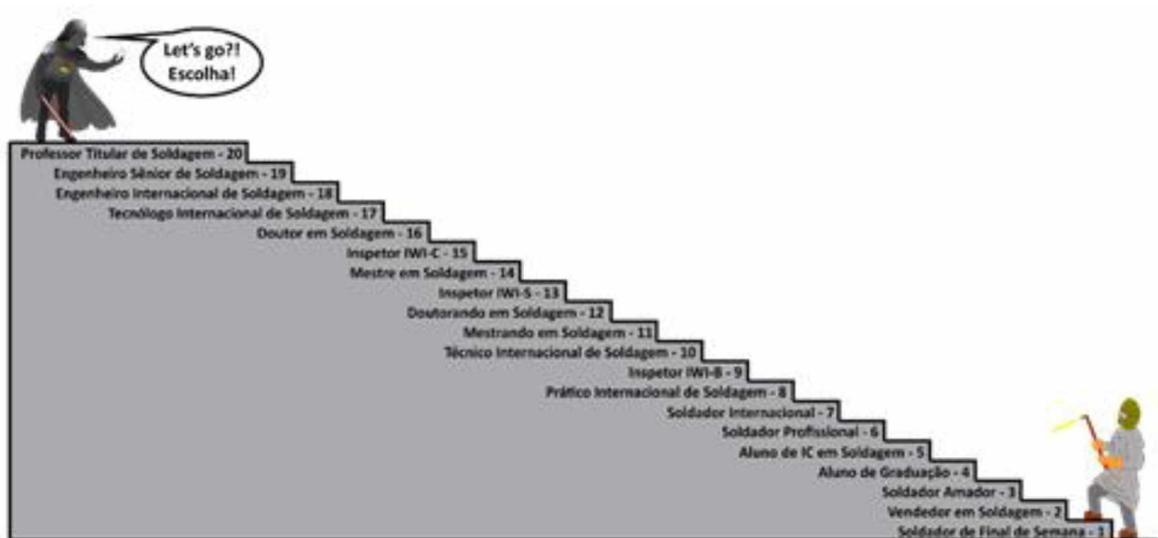


Figura 4.5 – Níveis do quiz

Se o jogador perde, o soldador sai de cena descendo todos os degraus subidos, e é repreendido por seu rival (Fig. 4.6). É a representação teatral de que o conhecimento do jogador foi insuficiente, e mais estudo se faz necessário. Se o jogador chega até o fim, quem sai de cena é o Darth Vader (Fig. 4.7), vencido pelo conhecimento de seu oponente.



Figura 4.6 – Saída do soldador derrotado

4.2 Alterações no quiz

Duas alterações simples (e necessárias) de serem executadas serão descritas a seguir.

4.2.1 Banco de perguntas

O número de perguntas em cada nível do jogo pode ser alterado sem mudanças no código principal, basta acrescentá-la no local correto. As perguntas estão armazenadas no diretório “**questions**”, dentro do diretório “**js**” (...js\questions). Figura 4.10 mostra que cada nível possui seu próprio arquivo.



Figura 4.10 – Arquivos das perguntas

Para fazer alterações em algum dos arquivos, e, portanto, nas perguntas dentro dos níveis, eles devem ser abertos utilizando o bloco de notas (Fig. 4.11). Outros programas podem ser utilizados, conforme preferência. O procedimento será similar.

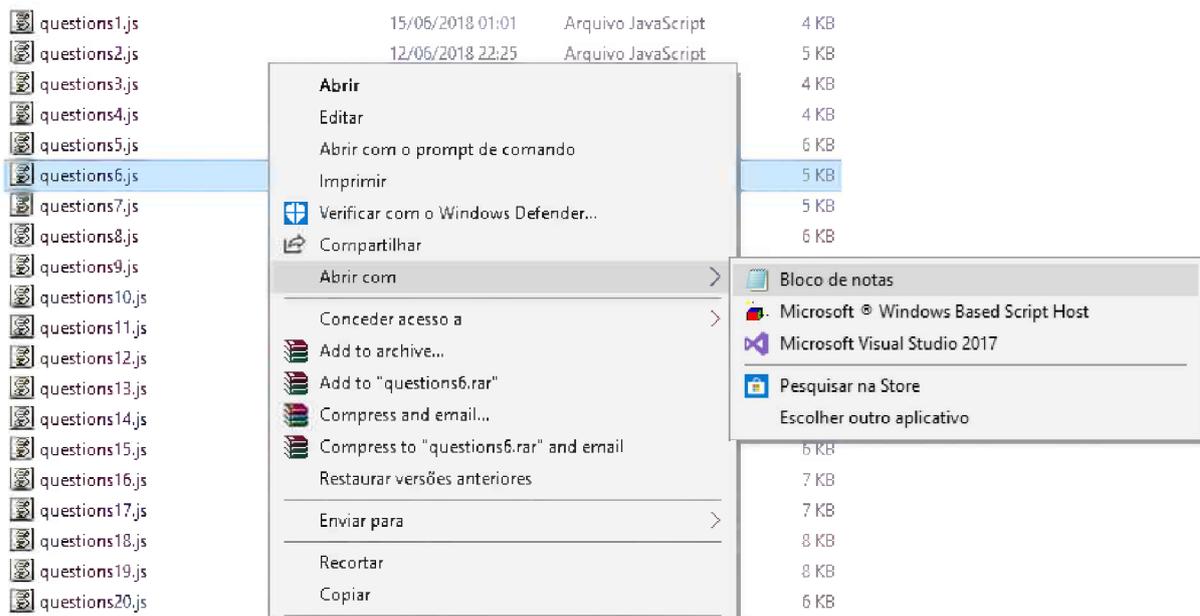


Figura 4.11 – Como abrir o arquivo das perguntas

Ao abrir o arquivo, deve ser localizado em seu fim a seção representada pela Fig. 4.12. Em seguida, copiar a parte indicada em azul, e colar logo à frente das chaves de fechamento indicado em amarelo (Fig. 4.13). Então basta preencher os campos com as informações correspondentes. Repetir o procedimento se forem adicionadas mais perguntas. Para alterar alguma pergunta, abrir o arquivo correspondente ao seu nível, encontra-la em meio às perguntas, e executar as mudanças. Observar que as respostas estão armazenadas em uma estrutura do tipo lista em que o índice dos elementos começa em zero, assim a “alternativa 1” corresponde ao índice “0”, a “alternativa 2” ao índice “1”, e assim sucessivamente. Na lacuna “alternativa correta” deve ser inserido o índice correspondente.

```

}
/*
{
  question: "pergunta",
  answers: [
    "alternativa 1",
    "alternativa 2",
    "alternativa 3",
    "alternativa 4"
  ],
  correctAnswer: "alternativa correta"
}*/

```

Figura 4.12 – Adição de novas perguntas, parte 1

```

    },
    {
      question: "pergunta",
      answers: [
        "alternativa 1",
        "alternativa 2",
        "alternativa 3",
        "alternativa 4"
      ],
      correctAnswer: "alternativa correta"
    }
  ],
  /*,
  {
    question: "pergunta",
    answers: [
      "alternativa 1",
      "alternativa 2",
      "alternativa 3",
      "alternativa 4"
    ],
    correctAnswer: "alternativa correta"
  }
  */
];

```

Figura 4.13 – Adição de novas perguntas, parte 2

4.2.2 Imagens

Uma alteração que também poderá ser feita rapidamente são as imagens do quiz. Para executar essa alteração a pasta **img** deve ser acessada. Em seguida, retirar a imagem que será substituída (preferencialmente colocar na pasta **op**), e colocar a nova imagem em seu lugar, conservando o mesmo nome. A alteração terá sido feita. Se o quiz estiver aberto, atualizar a página para ver as mudanças. Se forem observadas as proporções, nenhuma alteração no código será necessária.

4.3 Teste de campo

Houve a oportunidade de testar o funcionamento do quiz no 44º Congresso Nacional de Soldagem (CONSOLDA), que fortuitamente ocorreu na Universidade Federal de Uberlândia. A aplicação ocorreu sob supervisão do professor orientador Volodymyr Ponomarov, que reportou a ausência de falhas, e destacou a semelhança dos níveis finais alcançados pelos jogadores com suas qualificações em soldagem. Figuras 4.14, 4.15 e 4.16 são registros do evento.



Figura 4.14 – Teste CONSOLDA 1



Figura 4.15 – Teste CONSOLDA 2



Figura 4.16 – Teste CONSOLDA 3

CAPÍTULO 5

CONCLUSÃO

O quiz construído atende às ambições iniciais do projeto de conceber um jogo com fim educativo para ser utilizado no ensino de soldagem. Adotando como ponto principal para sua efetividade o engajamento do jogador, uma tela que interage com o progresso do jogador foi construída, contendo animações de tempo, indicação da dificuldade das perguntas, e uma “batalha intelectual” entre o jogador (representado pelo soldador) e seu rival Darth Vader. Por ser uma construção para web, poderá ser acessado por jogadores de diferentes localidades.

Um teste foi feito no 44º CONSOLDA. Nas poucas execuções do quiz nenhuma falha foi reportada, e o jogo ofereceu aos jogadores condições suficientes para testarem seus conhecimentos. É preciso ainda colher mais *feedback* para entender se o jogo funciona como o pretendido, isto é, se representa de fato uma opção legítima para estudar soldagem e testar conhecimento nesse tópico. Em todo caso, são indicadas melhorias na aparência geral do quiz, como estudo do jogo de cores, melhorar o formato da animação dos níveis, e um troféu final relacionado à soldagem. Também é viável acrescentar falas para ambos os personagens do jogo, relacionadas a algumas perguntas e/ou níveis específicos, como curiosidades e estatísticas.

Para a manutenção do quiz é saudável adicionar sazonalmente novas perguntas, a fim de cativar jogadores antigos, evitando que os mesmos fiquem presos nas mesmas questões. Tendo em mente que os níveis mais repetidos serão os iniciais, diminuindo a frequência progressivamente até o último, esta proporção também deve ser mantida no banco de perguntas, assim a recorrência de perguntas será menor, e a experiência do jogador melhor.

Também é vantajoso que o quiz passe a trabalhar com banco de dados. O objetivo é criar um perfil do usuário e interagir com o mesmo de maneira personalizada. Além disso, levantar estatisticamente a utilização do jogo, para entender como ele tem sido empregado pelos jogadores, se os objetivos do projeto são concretizados, e tomar decisões para alterações futuras com base nestas informações.

Uma última sugestão para o quiz seria a separação das perguntas em temas (Processos de soldagem, Materiais, Projeto, Fabricação) para melhorar a experiência dos jogadores que busquem por algo mais específico, a exemplo de estudar para avaliações. A separação por temas pode ser mais ambiciosa e abordar outras matérias do curso de Graduação em Engenharia Mecânica, ou mesmo de pós-Graduação, mas por ora a adoção da primeira sugestão é mais indicada.

CAPÍTULO 6

REFERÊNCIAS BIBLIOGRÁFICAS

ADDISON WESLEY LONGMAN. **A history of HTML**. 1998. Disponível em: <<https://www.w3.org/People/Raggett/book4/ch02.html>>. Acesso em: 26 dez. 2018.

BOS, B. **A brief history of CSS until 2016**. 17 dez. 2016. Disponível em: <<https://www.w3.org/Style/CSS20/history.html>>. Acesso em: 02 ago. 2018.

DÖRNER, R. et al. **Serious Games: Foundations, Concepts and Practice**. 1 ed. Suíça: Springer, 2016. p. 2-4.

GOFFMAN, E. **Presentation of Self in Everyday Life**. 1 ed. Nova Iorque: Anchor Books; Toronto: Random House of Canada, 1959. p. 34-51.

MDN Web Docs. **JavaScript basics**. Disponível em: <https://developer.mozilla.org/bm/docs/Learn/Getting_started_with_the_web/JavaScript_basics>. Acesso em: 26 dez. 2018.

PROTOPSALTIS, A. et al. Creative Learning with Serious Games Workshop. In: Fun and Games 2010, 3, 2010, Leuven. **Creative Learning with Serious Games**. 3p.