

**FELIPE CHAGAS RODRIGUES DE SOUZA**

**MARIA ISABEL NOGUEIRA DE OLIVEIRA**

**MARIA ALICE DE OLIVEIRA**

**VOLODYMIR PONOMAROV**

**APERFEIÇOAMENTO DO QUIZ PARA A  
AUTOAVALIAÇÃO DO CONHECIMENTO  
DOS ALUNOS EM SOLDAGEM**

**APERFEIÇOAMENTO DO QUIZ PARA A  
AUTOAVALIAÇÃO DO CONHECIMENTO  
DOS ALUNOS EM SOLDAGEM**



FELIPE CHAGAS RODRIGUES DE SOUZA  
MARIA ISABEL NOGUEIRA DE OLIVEIRA  
MARIA ALICE DE OLIVEIRA  
VOLODYMYR PONOMAROV

**APERFEIÇOAMENTO DO QUIZ PARA A  
AUTOAVALIAÇÃO DO CONHECIMENTO  
DOS ALUNOS EM SOLDAGEM**

1ª Edição

Quipá Editora  
2024

Copyright © dos autores e autoras. Todos os direitos reservados.

Esta obra é publicada em acesso aberto. O conteúdo dos capítulos, os dados apresentados, bem como a revisão ortográfica e gramatical são de responsabilidade de seus autores, detentores de todos os Direitos Autorais, que permitem o download e o compartilhamento, com a devida atribuição de crédito, mas sem que seja possível alterar a obra, de nenhuma forma, ou utilizá-la para fins comerciais.

Conselho Editorial: Me. Adriano Monteiro de Oliveira, Quipá Editora  
Dra. Anny Kariny Feitosa, Instituto Federal do Ceará (IFCE) / Dra. Érica Priscilla Carvalho L. Machado, Universidade Federal do Rio Grande do Norte (UFRN) / Dra. Francione Charapa Alves, Universidade Federal do Cariri (UFCA) / Dra. Mônica Maria Siqueira Damasceno, Instituto Federal do Ceará (IFCE) / Esp. Ricardo Damasceno de Oliveira, Universidade Regional do Cariri (URCA)

#### Dados Internacionais de Catalogação na Publicação (CIP)

---

L732c Souza, Felipe Chagas Rodrigues De  
Aperfeiçoamento do quiz para a autoavaliação do conhecimento dos alunos em soldagem / Felipe Chagas Rodrigues de Souza [et. al.]. — Iguatu, CE : Quipá Editora, 2024.

50p. : il.

ISBN 978-65-5376-307-4

1. Aprendizagem. 2. Soldagem. I. Título.

CDD

---

Obra publicada pela Quipá Editora em abril de 2024.

Quipá Editora  
[www.quipaeditora.com.br](http://www.quipaeditora.com.br)  
[@quipaeditora](https://www.instagram.com/quipaeditora)

## RESUMO

O objetivo é aperfeiçoar o jogo que foi anteriormente desenvolvido pelo aluno Kauê Oliveira Drigo (PFC de 2018, FEMEC, UFU) de tal forma que o jogo possa ser hospedado num site e armazene informações dos participantes. O jogo aprimorado com HTML, CSS, JavaScript, PHP e SQLite. O resultado contém uma tabela de Scores com as seguintes informações: pontuação, nome, instituição, país, contador para países e contador para as vezes que o jogo foi jogado. Também foram acrescentadas opções como idioma, categorias das perguntas e novas perguntas.

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

# SUMÁRIO

<b>CAPÍTULO 1.....</b>	<b>06</b>
------------------------	-----------

INTRODUÇÃO

<b>CAPÍTULO 2.....</b>	<b>09</b>
------------------------	-----------

REVISÃO BIBLIOGRÁFICA

<b>CAPÍTULO 3.....</b>	<b>20</b>
------------------------	-----------

METODOLOGIA

<b>CAPÍTULO 4.....</b>	<b>35</b>
------------------------	-----------

RESULTADOS E DISCUSSÃO

<b>CAPÍTULO 5.....</b>	<b>42</b>
------------------------	-----------

CONCLUSÃO

<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>50</b>
--	-----------

# CAPÍTULO 1

## INTRODUÇÃO

“A civilização da terceira onda tem sido chamada de sociedade da informação. Poucos se perguntam por que a informação se tornou tão importante na terceira onda. A razão está no fato de que os sistemas sociais, isto é, a sociedade se desmassificou, e, conseqüentemente, se complexificou, a tal ponto que, hoje, é impossível geri-la sem informação e sem tecnologia da informação (computadores e telecomunicações)” (TOFLER citado por CHAVES, 1998).

O conhecimento universitário precisa ser compartilhado para seu aprimoramento, e agora na era digital, há ferramentas, como jogos, que possibilitam uma facilidade de compartilhamento de informações muito forte na disseminação do conhecimento. Por isso o jogo proposto neste

trabalho, precisa estar à disposição de qualquer pessoa de uma maneira simples e rápida: hospedada na internet.

A internet, embora tenha surgido como um projeto militar, só começou a expandir-se quando se transformou em um meio para a troca de informações usado por grupos de pesquisadores de diferentes universidades (SENAC, 1998). O e-mail foi criado nessa época, justamente para viabilizar a troca de mensagens entre os pesquisadores.

Desde então, a troca de informações pela internet só evoluiu, e hoje este trabalho pode usufruir de tais tecnologias que podem: fornecer o jogo, captar e armazenar os dados, e então devolver estatísticas (que posteriormente serão base de novos estudos). O jogo tem um papel fundamental no processo de internacionalização da faculdade, pois permite interação de qualquer pessoa de qualquer país e na opção da língua inglesa (idioma considerado universal) e ainda ajuda a quantificar o interesse estrangeiro em trabalhos brasileiros.

Este jogo é pioneiro em sua categoria a abrirá portas para que mais modelos similares venham a se desenvolver, aumentando o interesse acadêmico em aprendizado online, de forma a complementar e também revisar estudos visto em sala. Outros jogos online que dizem abordar soldagem de forma didática, na verdade acabam por ter a soldagem como um tema, e não como objeto de estudo. A exemplo disso, na Figura 1.1 podem ser observadas exigências de logística com a temática de soldagem, mas nada que seja realmente didático.

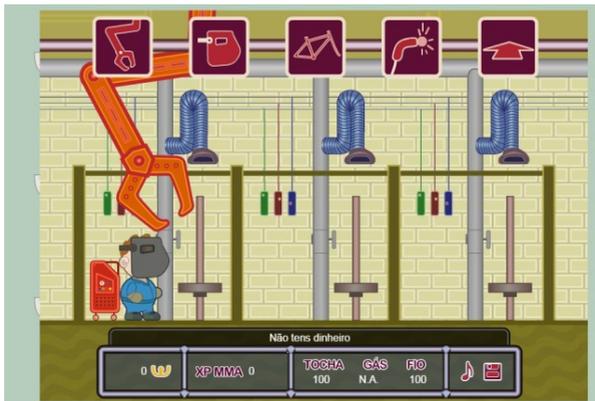


Figura 1. Weld Play Game.

## CAPÍTULO 2

### REVISÃO BIBLIOGRÁFICA

As linguagens a mais usadas pela Internet são HTML, CSS, JavaScript cujas sucintas descrições são apresentadas a seguir, junto também das linguagens usadas para hospedar o programa e armazenar dados dos usuários neste jogo (PHP, SLN e SQLite).

#### **HYPertext MARKUP LANGUAGE - HTML**

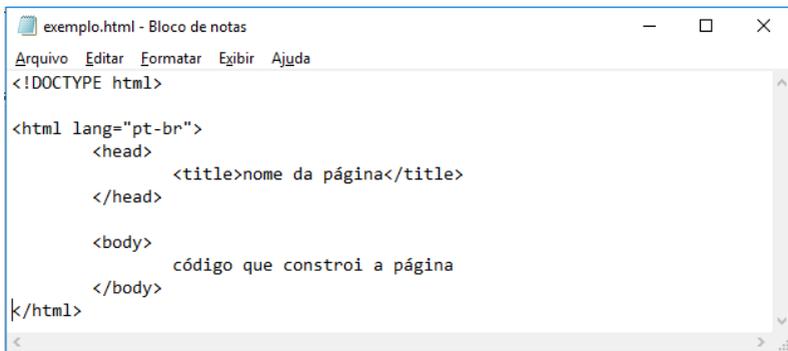
Em 1990, o cientista Tim Berners-Lee, procurando uma alternativa para melhorar a troca de informações entre acadêmicos, desenvolveu o HTML, uma linguagem que permitia a criação de páginas onde as informações seriam melhores exibidas visualmente. HTML é um acrônimo para *HyperText Markup Language* (Linguagem 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. 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. 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. O 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.

## **CASCADING STYLE SHEETS - 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. 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.

## **JAVASCRIPT - JS**

A ideia do projeto do JavaScript era a criação de uma linguagem para dinamizar a web, 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.

## **SQLITE**

Foi desenvolvido por D. Richard Hipp e é uma biblioteca de código aberto (open source) desenvolvido na linguagem C que permite a disponibilização de um pequeno banco de dados na própria aplicação, sem a necessidade de acesso a um SGDB separado. A estrutura de banco junto com a aplicação é denominada de “banco de dados embutido” e é indicada para aplicações de pequeno porte, que utilizam poucos dados.

A grande vantagem dos bancos de dados embutidos está em sua simplicidade: é mais prático implementar e administrar do que a implementação de SGDB’s separados, utilizando soluções como SQL Server e Oracle. Por outro lado, a performance e limitação, de recursos são desvantagens do SQLite e soluções semelhantes. Para escolher a opção mais adequada, devem ser levados em consideração parâmetros como os exemplificados a seguir.

- Quando o uso do SQLite é recomendado:
- Aplicativos básicos desktop / mobile
- Pequenos Web Sites
- Sistemas utilizados por poucas pessoas
- Quando o uso do SQLite não é recomendado:
- Sistemas Web / aplicações cliente servidor
- Sites que recebem quantidade elevadíssima de visitas

## **ARQUIVO DE SOLUÇÃO (SOLUTION) - SLN**

Uma solução é uma estrutura para organizar projetos no Visual Studio. A solução mantém as informações de estado para projetos em .sln (baseado em texto, compartilhado) e arquivos .suo (opções de solução de binário, especificada pelo usuário). Se o VSPackage é carregado como resultado de que está sendo referenciado no arquivo. sln, o ambiente chama ReadSolutionProps para ler no arquivo. sln.

O arquivo. sln contém informações baseadas em texto, que o ambiente usa para localizar e carregar os parâmetros de nome-valor para os dados persistentes e o projeto VSPackages faz referência a ele. Quando um usuário abre uma solução, o ambiente percorre os preSolution, Project, e postSolution informações no arquivo. sln para carregar a solução, projetos dentro da solução, e todas as informações persistentes anexado à solução.

Arquivo de cada projeto contém informações adicionais de leitura pelo ambiente para preencher a hierarquia com itens do projeto. A persistência de dados de hierarquia é controlada pelo projeto; os dados não são normalmente armazenados no arquivo. sln, embora você possa escrever intencionalmente informações do projeto para o arquivo se você optar por fazê-lo. Para obter mais informações relacionadas à persistência, consulte persistência do projeto e abrindo e salvando itens de projeto.

## **PERSONAL HOME PAGE - PHP**

Atualmente conhecida como Hypertext Preprocessor, é uma linguagem de programação amplamente utilizada por desenvolvedores ao redor de todo o mundo para a construção de uma série de aplicações, a exemplo de websites dinâmicos, pois permite a interação com o usuário por meio de links, formulários e parâmetros de URL.

Criada em 1994 e com o código-fonte publicado no ano posterior, foi originalmente escrita por Rasmus Lerdorf sob a sigla que significa Personal Home Page. Inicialmente, o principal objetivo de Rasmus era substituir através do php o conjunto de scripts Perl que era utilizado para manter sua página web pessoal no ar.

Com a crescente evolução da tecnologia, o desenvolvedor reescreveu a linguagem como um script engine, ou seja, como um programa executado no lado do servidor (e não do usuário) capaz de interpretar os scripts php e

gerar páginas HTML. Como também era capaz de interpretar formulários, essa evolução da linguagem ficou conhecida como PHP/FI.

O PHP tem sua principal utilização em aplicativos web. Isso porque, diferentes de linguagens de programação como o Javascript, que possui uma função similar, o PHP roda do lado do servidor, e não na máquina do cliente. Isso significa que além de ser enviado HTML puro para o cliente, o PHP ainda consegue interagir com as aplicações existentes no servidor, como o banco de dados, sem ser necessário expor o código-fonte para o cliente. Com isso, ele é ideal para desenvolver programas que necessitam lidar com informações confidenciais, já que senhas e outras informações de caráter pessoal e intransferível não serão mostradas.

## **CAPÍTULO 3**

### **METODOLOGIA**

O quiz foi desenvolvido utilizando HTML-CSS-JS-PHP-SQLite, escolha efetuada conforme o conhecimento do autor. Seguem os detalhes da concepção e desenvolvimento do projeto.

#### **OPÇÃO DE IDIOMAS**

A primeira modificação foi inserir a opção de acessar questões em inglês, pois é um idioma universal. Para realizar esta alteração, foi preciso de pouca alteração no código, e a maior parte do trabalho se deu em traduzir e reinserir as questões nesta nova categoria.

A Figura 3.1.1 mostra a leitura da opção de idioma selecionado, enquanto a Figura 3.1.2 mostra o arquivo script.js usando tal informação.

```
if(lingua[0].checked)
    quizStart(true,"pt", subjectValue);
else
    quizStart(true,"en", subjectValue);
```

Figura 3.1.1 – A variável “língua” recebe a opção selecionada na interface do usuário

```
////////////////////////////////////
/ FUNÇÃO PRINCIPAL //////////////////////////////////////
////////////////////////////////////
unction quizStart(eletrodoRevestido, language, subjects) {
    console.log("Hey " + language);
    var i;
    var questionLevel = questionCurrent(questionNumber, langu
    var aux = questionRandom(questionLevel);
    console.log("Entre com subject " + questionLevel[aux].su
```

Figura 3. Logo no início do Quiz, é decidido qual banco de perguntas será chamado: português ou inglês

Apartir de então, o programa irá interagir com a biblioteca selecionada (Figura 3.1.3). Mas se não encontrar a

pasta do idioma selecionado, o programa não funciona corretamente, ao contrário da opção de categorias de perguntas que será visto em seguida. Ou seja, o idioma é um restrição não uma preferência.

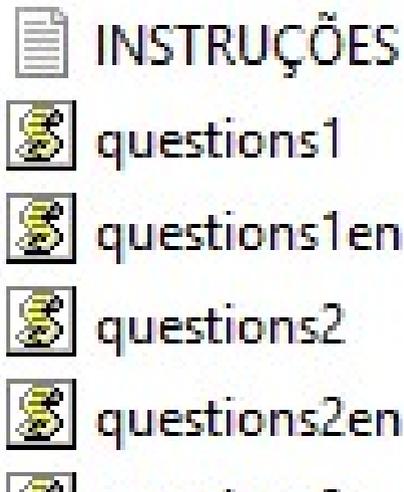


Figura 4. Na pasta que contém as questões tem-se nos 2 idiomas para cada nível

## OPÇÃO DE SUBJECT

As perguntas da disciplinas soldagem pode ser divididas em 4 categorias: **Processos**(0), **Materiais**(1), **Projeto**(2) e **Fabricação**(3). Para isso, foi criado uma nova variável chamada "subject" que é uma preferência da categoria que o usuário pode escolher. Ou seja, as perguntas são selecionadas de forma randômica, e ao escolher uma categoria (ou mais de uma) elas são sorteadas apenas dentro da(s) categoria(s) escolhidas. Porém caso falte questões na categoria, o programa irá buscar qualquer outra pergunta fora da categoria escolhida, ao contrário da opção de idioma, em que o programa pára de funcionar caso não encontre as questões selecionadas. Assim sendo, é possível escolher uma categoria, ou várias ao mesmo tempo, dando até 14 combinações. Sendo que a opção em que todas categorias estão selecionadas é a que vem sempre inicialmente marcada na tela inicial, independente da opção selecionada num jogo anterior. A leitura da categoria selecionada e a variável que

indica na questão sua categoria, podem ser vistos nas Figuras 3.2.1 e 3.2.2, respectivamente.

```
var questionLevel = questionCurrent(questionNumber, language);
var aux = questionRandom(questionLevel);
console.log("Entrei com subject " + questionLevel[aux].subject);
var trials = 0;
while(!subjects.includes(parseInt(questionLevel[aux].subject)) &&
    aux = questionRandom(questionLevel);
    console.log("entrei");
```

Figura 5. A variável "subject" definindo como a função randômica vai agir

```
question: "Você tem qu
          + " você v
answers: [
    "2",
    "4",
    "7",
    "5"
],
correctAnswer: "3",
subject: "2"
```

Figura 6. Classificação da questão

## COLETA DE DADOS DO USUÁRIO

A partir daqui, já se começa a utilizar a extensão php, pois os códigos precisam gerar páginas. Temos as seguintes novas rotinas na Figura 3.3.1. Os dados do usuário são **nome**, **universidade**, **país** e **pontuação final**.

Estas rotinas obedecem a seguinte lógica, de forma resumida:

*Ao fim do programa a função **script.js**, envia para o **insertscore.php** os dados do usuário. O **conexaobd.php** confere a existência do banco de dados chamado **score.sqlite**, e o gera caso ele tenha sido apagado. O **conexaobd.php** é um requisito para que o **insertscore.php** seja executado, para que este último por sua vez, possa trabalhar com os dados e prepará-los para então o **listaScores.php** gerar uma tabela com os dados em uma página html. O **score.php** apenas fazer a contagem de jogos totais e por países. A seguir*

será explicado com mais detalhes o funcionamento de cada um deles.

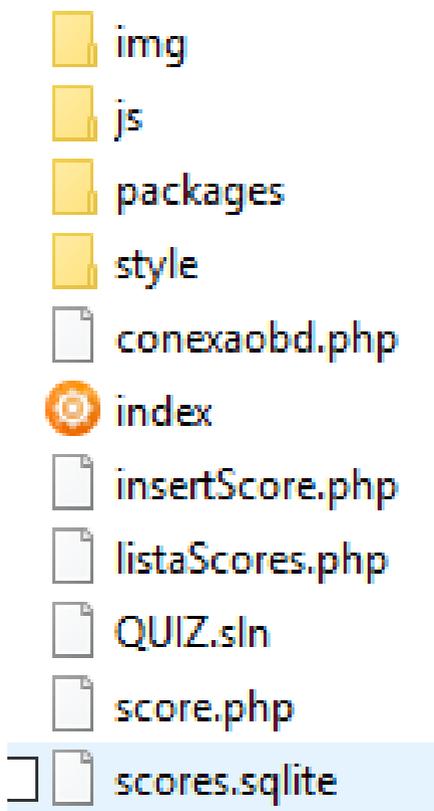


Figura 7. Novas rotinas.

## SCRIPT.JS – 331 linhas

Este programa já vem da versão anterior do Quiz e precisou apenas de algumas modificações para que então acionasse o **listaScores.php**. Na Figura 3.3.1 podemos ver um exemplo de tal modificação:

```
function insertScore() {
    method = "post";

    var form = document.getElementById("optionForm");
    form.setAttribute("method", method);
    form.setAttribute("action", "insertScore.php");

    var hiddenField = document.createElement("input");
    hiddenField.setAttribute("type", "hidden");
    hiddenField.setAttribute("name", "score");
    hiddenField.setAttribute("value", questionNumber);

    form.appendChild(hiddenField);

    form.submit();
}
```

Figura 8. trecho com exemplo de modificação ao longo do **scrip.js**

## CONEXAOBD.PHP – 29 linhas

Quando o programa é executado, em algum momento o **insertScore.php** será solicitado é ele quem chama o **conexaobd.php** (conexão ao banco de dados). Quando **conexaobd.php** é executado ele confere a existência do banco de dados **Score.sqlite** e, em caso de ausência ele é criado instantaneamente. Figura 3.2.2 mostra o **conexaobd.php**.

```
19         PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
20     ));
21     if (!$db) {
22         $error = (file_exists('scores.sqlite')) ? "Im
23         die($error);
24     }
25     $db->query("CREATE TABLE IF NOT EXISTS scores (
```

Figura 9. Momento em que o **conexaobd.php** faz um “if” para conferir a existência do **scores.sqlite**.

## **SCORE.SQLITE**

Este é um banco de dados, é aqui que é armazenado as informações dos jogos anteriores. Em caso de necessidade de um back-up, é este arquivo que deve ser salvo. Em caso de necessidade de limpar a lista de scores basta excluir este arquivo. O banco de dados do tipo SQLite foi escolhido pois este é o mais simples e não precisa de um servidor trabalhando em paralelo para funcionar, ou seja, é basicamente um arquivo.

## **INSERTSCORE.PHP – 43 linhas**

Aqui os dados obtidos no programa são processados, armazenados em novas variáveis para então serem trabalhados no objetivo final de exibí-los numa página html. Um detalhe importante é um sistema de segurança para filtrar os dados que serão armazenados, pois é possível que pessoas mal-intencionadas insiram rotinas de programas ao preencher "nome" e/ou "universidade", e então danificar o programa. O

filtro de segurança pode ser visto na Figura 3.3.4.1 Já na Figura 3.3.4.2 podemos ver o trecho que faz a inserção de novos dados ao banco de dados.

```
18
19     $studentName = filtraEntrada($_POST["studentName"]);
20     $university = filtraEntrada($_POST["university"]);
21     $country = $_POST["country"];
22     $score = $_POST["score"];
23
24     try{
```

Figura 10. Filtro nas variáveis com o nome e universidade.

```
25
26     $conexao = conectabd();
27
28     $sql = "INSERT INTO scores( name, country, university, score )
29     VALUES ( '$studentName', '$country', '$university', '$score' )";
30
```

Figura 3.3.4.2 – Inserção de dados no banco de dados.

## **SCORE.PHP – 71 linhas**

Esta rotina é responsável pela coleta dados sobre a utilização do jogo, permitindo quantificar o número total de partidas jogadas, bem como o número de visitas provenientes

de cada país. Essas informações podem ser valiosas para entender a popularidade do jogo em diferentes regiões e para orientar estratégias de marketing e desenvolvimento. A Figura 3.3.5 exemplifica uma das funções contadoras, mostrando como os dados são coletados e organizados para análise.

```
63
64     function getTotal($conexao) {
65         $sql = "SELECT count(country) as total FROM scores;";
66         $result = $conexao->query($sql);
67         if(!$result)
68             throw new Exception('Ocorreu uma falha ao gerar o relat
69         return $result->fetch( PDO::FETCH_ASSOC )['total'];
70     }
71     ?>
```

Figura 3.3.5 – Exemplo de uma das funções contadoras.

### LISTASCORE.PHP – 143 linhas

Este arquivo possui uma parte estática (nunca muda) e também uma parte dinâmica (que varia). A parte dinâmica varia de acordo com os dados que o **listaScore.php** busca no banco de dados precisa estar num servidor, pois ele acessa o

banco de dados. Por isso é preciso estar em um servidor para utilizar este programa.

A principal tarefa do **listaScore.php** é juntar todas as informações dos programas apresentados anteriormente e apresentá-los em uma tabela, a qual será esboçada numa página html. Para visualizar a lista de scores então, basta adicionar “[/listaScores.php](#)” ao final do link que estiver hospedando o programa.

Na figura 3.3.6 pode-se ver um trecho do algoritmo que esboça os resultados do jogo.

```

48
49 - <body>
50   <h2 class="listagens">TOTAL DE PARTICIPANTES:</h2>
51   <?php
52     echo "<h2 class='bigTotal'>{$total}</h2>"
53   >
54 - <div class="container lista">
55   <h2 class="listagens">SCORES</h2>
56   <div class="table-responsive">
57     <table class="table">
58       <thead>
59         <tr class="tabelalistagens">
60           <th>Nome</th>
61           <th>Universidade</th>
62           <th>País</th>
63           <th>Nível Máximo</th>
64         </tr>
65       </thead>

```

Figura 3.3.6 – Trecho que esboça os Scores.

O **listaScore.php** também é responsável por criar o vetor que ordena as pontuações dos jogadores.

## REVISOR DE SINTAXE DA LINGUAGEM

Ao contrário de programas como compiladores em linguagem C que acusam erro em caso de sintaxe errada, isso não ocorre em javascript. O programa não acusa nenhum

erro, apenas funciona de maneira incorreta ou nem funciona. Por isso foi utilizado um seguinte site (Figura 3.4) para conferir as sintaxes, principalmente para inserir novas questões no programa.

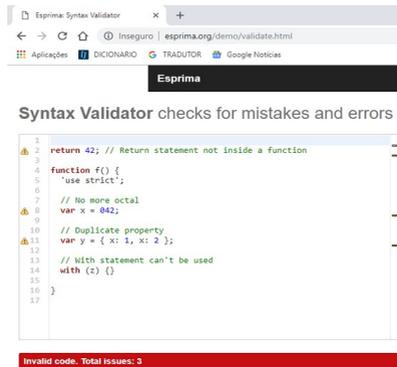


Figura 3.4 – Esprima.org

# CAPÍTULO 4

## RESULTADOS E DISCUSSÃO

O resultado da concepção do Quiz é apresentada em seguida, e pode ser vislumbrado em Figura 4.1 e comparada pela versão anterior na Figura 4.2

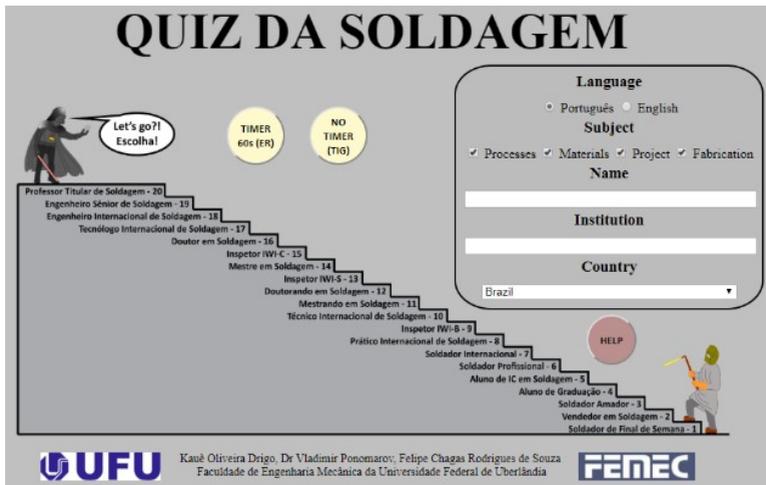


Figura 4.2 – Página inicial atual.

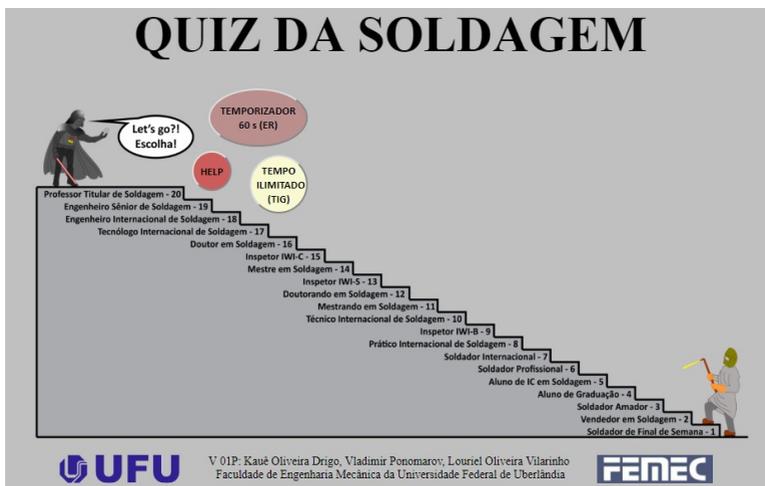


Figura 4.2 – Página inicial anterior.

## CARACTERÍSTICAS

Para agilizar a jogabilidade, prevendo que os jogadores joguem algumas vezes seguidas, a página inicial vem preenchida visando a maior parte do público alvo. Assim sendo, temos que o idioma estará em português, todas as categorias das disciplinas marcadas e o país sendo o Brasil. O jogo só se inicia se Todos os itens estiverem preenchidos

## HOSPEDANDO O QUIZ EM UM SERVIDOR

A seguir há uma série de passos ensinando como hospedar o quiz em um servidor, usando de exemplo o Umblar. Será feito também comentário para a hospedagem no site pretendido: a página da FEMEC.

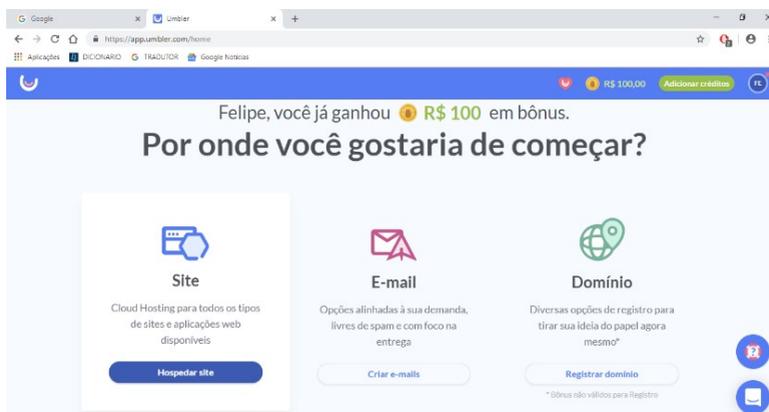


Figura 4.2.1 – Site Umblar e sua interface simples para criar sites

No site da FEMEC não há possibilidade de incompatibilidade com este tipo de aplicação, porém pode

haver uma restrição para o PHP. Neste caso é preciso liberar a restrição do servidor para a paste que conterà o Quis.



Figura 4.2.2 – Aqui escolhemos o nome do site. No caso do site da FEMEC já tem-se um domínio definido, será feito apenas uma subdivisão nele.



Figura 4.2.3 – O site Umblor pergunta o tipo de aplicativo usado, que no caso é o PHP.

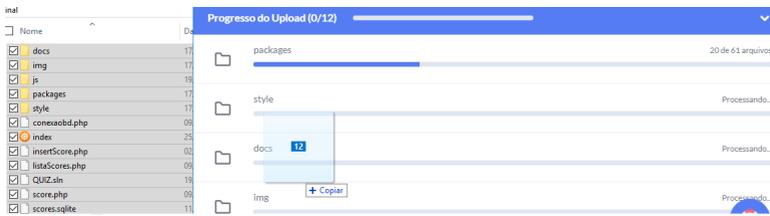


Figura 4.2.4 – Basta inserir todos arquivos no site e o programa já irá funcionar.



Figura 4.2.5 – Será gerado um link para uma página HTML a qual terá o jogo. Os arquivos PHP que criam esta página.

## LINKS DE TESTE E IMAGENS DO JOGO HOSPEDADO

Conforme visto anteriormente o jogo está disponível para todos em um link, com o **“index.html”** sendo o responsável

por juntar todas as partes de Quiz e inicializa-lo. Ver Figura 4.3.1. os links de **teste** (válidos até final de março de 2019) para o jogo e para os scores são, respectivamente:

<http://testequiz00-com.umbler.net/index.html>

<http://testequiz00-com.umbler.net/listaScores.php>

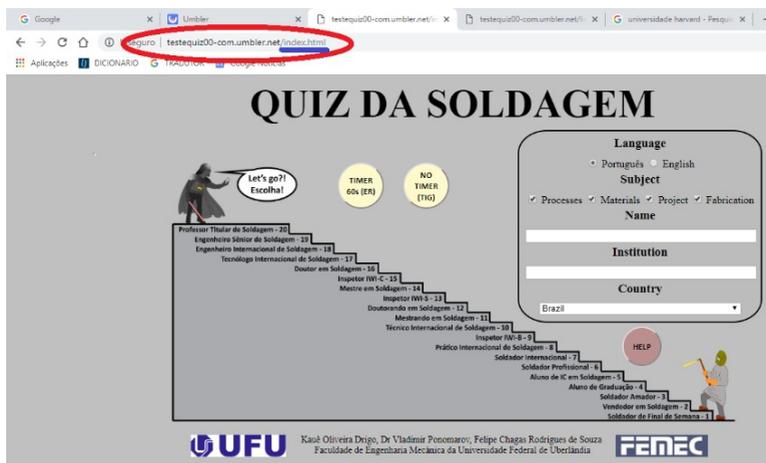


Figura 4.3.1 – Link para todos poderem jogar o Quis.

Já para visualizar os resultados, é preciso acessar o link com o final “**listaScores.php**” que é o responsável por gerar a

tabela (Figura 4.3.2). Nela tem-se todos os resultados armazenados no banco de dados.

TOTAL DE PARTICIPANTES:  
4

SCORES			
Nome	Universidade	Pais	Nível Máximo
Felipe	UFU	Brazil	10
vladin	gringa	Ukraine	3
ilgal	harvard	Mexico	2
felipe teste serio	ufu	Brazil	1

PAISES

Pais	Participantes
Brazil	2
Mexico	1
Ukraine	1

Figura 4.3.1 – Link para lista de Scores

## **CAPÍTULO 5**

### **CONCLUSÃO**

Todas as propostas para aperfeiçoamento do Quiz foram atingidas. O jogo foi testado inúmeras vezes e nenhum bug foi encontrado na versão final, e ficou com o tamanho de 31Mb. Já é possível a hospedagem do Quiz no site da FEMEC, conforme a proposta do professor. O jogo somará pontos para as atividades que se enquadram na categoria de “internacionalização da faculdade”, e terá como sustentação de tal proeza os dados obtidos na lista de scores.

Antes da hospedagem é preciso apenas elevar ainda mais o número de questões de 309(atuais até dezembro de 2018) para 500, e traduzí-las para o inglês.

Para a próxima etapa de aperfeiçoamento sugiro 4 itens que iam além da proposta deste TCC, mas que terão grande valor para o Quiz:

### **Inserir um botão “Enter” para as respostas e “Scores” [alteração simples]**

Ao final do TCC foi percebido um problema: quando acessado por smartphones, é impossível confirmar as opções de respostas selecionadas. Portanto é preciso um botão com a função do “enter” para confirmar a opção escolhida e prosseguir no quiz, quando acessado por um smartphone. Exemplo na figura 5.1.1.

Também houve discussão para inserir um botão para que o usuário veja o score (ao lado do help por exemplo), porém para isso é preciso definir o domínio em que o programa será hospedado. Figura 5.1.2.

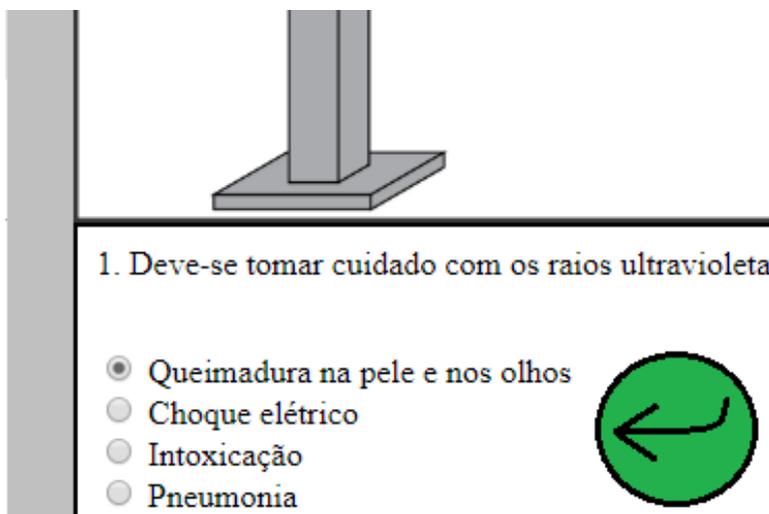


Figura 5.1.1 – Exemplo de botão “enter”

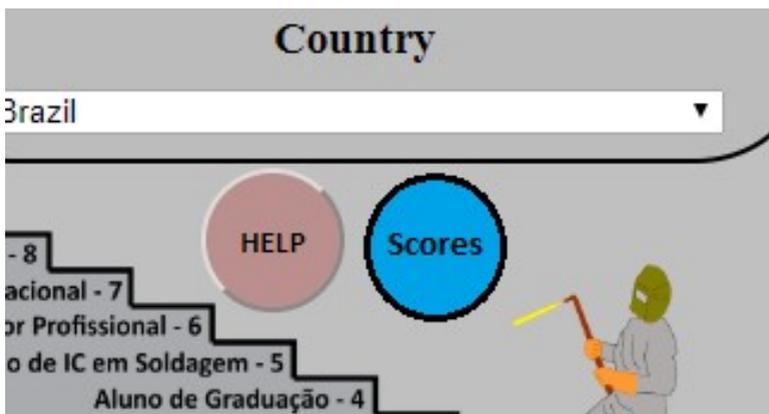


Figura 5.1.1 – Exemplo de botão “Scores”

### **Perguntas com imagens [alteração complexa]**

Inserir perguntas com imagens é muito importante, principalmente na categoria processos. O número de questões nessa categoria ficou limitado devido esta restrição.

### **Pontuação com o Timer [alteração simples]**

O usuário que escolher a opção de jogo com tempo, não deveria entrar para o score. Também pode ser interessante na lista de Scores com os resultados inserir uma coluna com dados sobre o tempo gasto pelo jogador.

OBS: para isso seria necessário corrigir um pequeno erro: o timer pára de marcar o tempo quando o jogo é minimizado ou simplesmente deixa de ser a janela principal na tela do usuário.

### **Nova estrutura: XAMPP**

Está é a modificação sugerida mais importante, uma mudança radical na estrutura do jogo. A utilização da

aplicação chamada XAMPP irá dar novas possibilidades para o Quiz.

Com o XAMPP é possível simular um provedor e não será preciso colocar o programa em um servidor real para testar os PHPs.

O "SQLite" não é a melhor opção para o caso, e através do XAMPP é criado um "mysql" que dará mais funcionalidades.

Também será possível criar uma interface para que um Administrador tenha a possibilidade de editar as questões direto no servidor, sem a necessidade de ficar fazendo sucessivos uploads de arquivos. Aperfeiçoando a interface é possível que extensões ".xls" (Figura 5.4.1) tornem-se "csv", e então de forma similar à estrutura do Excel, será possível inserir e editar questões numa interface simples, diretamente no servidor e já moldá-las automaticamente para uma estrutura que o programa precisa (Figura 5.4.2).

A	B	C	D	E	F	G	H
NÍVEL	PERGUNTA	RESPOSTA 0	RESPOSTA 1	RESPOSTA 2	RESPOSTA 3	GABARITO	Tema: 0.1.2.3
2	A corrente contínua (CC) é a corrente elétrica que ...	passa através de um condutor em uma só direção	nunca acaba	passa através de um condutor, ora em uma direção, ora na direção oposta, com uma regularidade alta	tem sempre o valor constante	0	0
2	A corrente alternada (CA) é a corrente que ...	muda seu valor de repente	pode ser ligada e desligada com frequência	passa através de um condutor, ora em uma direção, ora na direção oposta, com uma frequência determinada	oscila entre dois níveis de valor sem mudar sua direção	2	0
2	A tomada da rede pública fornece ...	corrente contínua, pois a tensão disponível (127 ou 220 V) não muda de valor	corrente contínua, visto que ela não varia, a não ser se houver uma mudança na resistência no circuito elétrico	corrente alternada, com uma polaridade estabelecida	corrente alternada	3	0
2	Qual dos materiais abaixo tem maior condutividade térmica (condutor melhor isolador)?	Ar	Alumínio	Aço	Baquelite	1	1
2	Como o soldador deve manusear os eletrodos para manter livre de mistura e contaminantes (sujeira)?	Retirar os eletrodos da embalagem e lavá-los bem com água.	Manter os eletrodos que não serão utilizados de imediato na embalagem	Retirar os eletrodos da embalagem e ventá-los bem antes da soldagem	Retirar os eletrodos da embalagem e aquecê-los todos, mesmo sem previsão de uso mediana	1	0
2	Um soldador realizou um metro de solda em 200 s. Qual a velocidade de soldagem em cm/min?	30 cm/min	60 cm/min	0,5 cm/min	50 cm/min	0	3
2	Qual dos materiais abaixo tem pior condutividade térmica (melhor isolante térmico)?	Cobre	Alumínio	Aço	Baquelite	3	1
2	Qual dos materiais abaixo tem maior resistência mecânica?	Cobre	Alumínio	Aço	Baquelite	2	2
2	Atopas é medida em ...	Pa	N/m	N/m	kg	3	2
2	Which scale number according to EN 163 gives the highest darkness?	0	10	12	16	3	3
2	A welder should protect himself against radiation by using ...	a welding curtain	self darkening glasses	ear protectors	proper clothing and a welder's shield with adequate filters	3	3

Figura 5.4.1 – Tabela usada para organizar as questões atualmente

Para plotar a tabela é recomendado usar um “jschar”. Este utilizará uma biblioteca para gerar a tabela de scores de forma mais didática. Assim haverá mais possibilidade de plotar gráficos e estatísticas (por exemplo, qual a pergunta que foi respondida errada mais vezes).

```

    ,
      {
        question: ""
          + "",
        answers: [
          "",
          "",
          "",
          ""
        ],
        correctAnswer: "",
        subject: ""
      }

```

Figura 5.4.2 – Estrutura atual para inserir uma questão

Para finalizar, o XAMPP pode ainda dar a possibilidade de que o jogo fique hospedado diretamente a um computador, bastando que a CTI (Centro de Tecnologia e Informação) da UFU cadastre o IP do computador nos servidores da UFU. Neste caso o computador precisa estar ligado 24h para que o jogo permaneça online, porém inicialmente é uma maneira de agilizar a correção de bugs e problemas com as questões.

## REFERÊNCIAS BIBLIOGRÁFICAS

ASTON, B. **A brieft history of JavaScript**. 01 abr. 2015. Disponível em: <<https://medium.com/@benastontweet/lesson-1a-the-history-of-javascript-8c1ce3bffb17>>. Acesso em: 25 out. 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: 25 out. 2018.

CHAVES, Eduardo. **Resumo de palestra de Alvin Tofler In: EDUTECCNET**. [Citado em 8 Nov. 1998 21:19]. Disponível na Internet: [edutec@edutecnet.com.br](mailto:edutec@edutecnet.com.br).

EIS, D. **O básico: O que é HTML?**. Jan. 2011. Disponível em: <<https://tableless.com.br/o-que-html-basico/>>. Acesso em: 25 out. 2018.

HIDAYAT, A. **Syntax Validator**. Disponível em: <<http://esprima.org/demo/validate.html>>. Acesso em: 19 dez. 2018.

LIEW, G. V. **Arquivo de solução (.Sln)**. Disponível em:<<https://docs.microsoft.com/pt-br/visualstudio/extensibility/internals/solution-dot-sln-file?view=vs-2017>>. Acesso em: 19 dez. 2018.

PEYROTT, S. **A Brief History of JavaScript**. 16 jan. 2017. Disponível em: <<https://auth0.com/blog/a-brief-history-of-javascript/>>. Acesso em: 25 out. 2018.

SQLITE.ORG. **What Is SQLite?**. Disponível em:<<https://www.sqlite.org/index.html>>. Acesso em: 19 dez. 2018.

UMBLER.COM. **UMBLER**. Disponível em:<<https://app.umbler.com/home>>. Acesso em: 19 dez. 2018.

UNIVERSITY OF WASHINGTON. **A Brief History of HTML**. Disponível em: <[https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html\\_history.html](https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html)>. Acesso em: 25 out. 2018.

ISBN 978-655376307-4



9

786553

763074