

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO**

Jogo de perguntas e respostas em um plugin  
de jogos no auxílio à aprendizagem para o Moodle

**Pedro Igor Porfírio Sampaio**

**PROJETO FINAL DE GRADUAÇÃO**

**CENTRO TÉCNICO CIENTÍFICO - CTC**

**DEPARTAMENTO DE INFORMÁTICA**

**Curso de Graduação em Ciência da Computação**

Rio de Janeiro, Novembro de 2015.



**Pedro Igor Porfírio Sampaio**

**Jogo de perguntas e respostas em um plugin de jogos no  
auxílio à aprendizagem para o Moodle**

Relatório de Projeto Final, apresentado ao  
**Departamento de Informática** da PUC-Rio como requisito  
parcial para a obtenção do título de Bacharel em Ciência da  
Computação.

Orientadora: Simone Diniz Junqueira Barbosa

Rio de Janeiro  
Novembro de 2015.

## Resumo

Sampaio, Pedro Igor Porfírio. Barbosa, Simone Diniz Junqueira. Jogo de perguntas e respostas em um plugin de jogos no auxílio à aprendizagem para o Moodle. Rio de Janeiro, 2015. 60p. Projeto Final II – Departamento de Ciência da Computação. Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho teve por objetivo especificar e desenvolver um plugin para o Moodle que agrupe jogos educacionais relacionados à matéria do curso, bem como um jogo que se encaixe nesse perfil. O plugin reside nos blocos do Moodle e foi projetado com design inspirado em agregadores de multimídia. O jogo reside dentro do plugin, assim como todos os jogos a serem desenvolvidos para este plugin deverão residir, e seu design foi inspirado em jogos de trivia. O plugin possui uma forma automatizada de visualização de seus jogos, com fácil inserção, remoção e alteração destes.

Palavras-chave: Departamento de Informática, Moodle, Jogos de Aprendizagem, Relatórios, Pesquisa

## **Abstract**

Sampaio, Pedro Igor Porfírio. Barbosa, Simone Diniz Junqueira. A trivia game in a plugin of games in the aid of learning for Moodle. Rio de Janeiro, 2015. 60p. Projeto Final II – Departamento de Ciência de Computação. Pontifícia Universidade Católica do Rio de Janeiro.

This project aimed to specify and develop a plugin for the Moodle that groups learning games related to the course material as well as a game that fits this profile. The plugin lies in the Moodle blocks and its design was inspired by multimedia aggregators. The game lies inside the plugin, much like every game of this plugin will lie, and its design was inspired by trivia games. The plugin has an automated form of displaying its games, with an easy way of inserting, removing and modifying them.

Keywords: Department of Informatics, Moodle, Learning Games, Reports, Research

## Sumário

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>5</b>
<b>2</b>	<b>SITUAÇÃO ATUAL .....</b>	<b>7</b>
<b>3</b>	<b>OBJETIVOS .....</b>	<b>9</b>
<b>4</b>	<b>ATIVIDADES REALIZADAS .....</b>	<b>11</b>
4.1	AMBIENTE DE DESENVOLVIMENTO .....	11
4.2	ESTUDOS E DEFINIÇÃO DA LINGUAGEM .....	11
4.3	ESTUDOS CONCEITUAIS PARA O PLUGIN .....	11
4.4	ESTUDOS CONCEITUAIS PARA O JOGO .....	12
4.5	DEFINIÇÃO DO JOGO .....	13
4.5.1	<i>Configuração .....</i>	<i>13</i>
4.5.2	<i>Objetivo .....</i>	<i>13</i>
4.5.3	<i>Jogabilidade .....</i>	<i>14</i>
4.6	ESTUDOS DE CONTROLE DE VERSÃO .....	14
4.7	TESTE E PROTÓTIPOS PARA APRENDIZADO E DEMONSTRAÇÃO .....	15
4.8	MÉTODO .....	16
4.9	CRONOGRAMA .....	17
<b>5</b>	<b>PROJETO E ESPECIFICAÇÃO DO SISTEMA .....</b>	<b>18</b>
5.1	PLUGIN .....	18
5.1.1	<i>Instalação do plugin .....</i>	<i>18</i>
5.1.2	<i>Instalação de jogos no plugin .....</i>	<i>20</i>
5.1.3	<i>Adequação ao Moodle .....</i>	<i>20</i>
5.1.4	<i>Banco de dados .....</i>	<i>24</i>
5.1.5	<i>Outros arquivos .....</i>	<i>26</i>
5.1.6	<i>Interface .....</i>	<i>27</i>
5.2	JOGO .....	29
5.2.1	<i>Instalação .....</i>	<i>29</i>
5.2.2	<i>Banco de dados .....</i>	<i>30</i>
5.2.3	<i>Estrutura web e página principal .....</i>	<i>31</i>
5.2.4	<i>Configurações .....</i>	<i>33</i>
5.2.5	<i>Modo jogador único .....</i>	<i>35</i>
5.2.6	<i>Modo multijogador .....</i>	<i>39</i>
5.2.7	<i>Notificações .....</i>	<i>43</i>
5.2.8	<i>Conquistas .....</i>	<i>47</i>
5.2.9	<i>Rankings .....</i>	<i>50</i>
<b>6</b>	<b>IMPLEMENTAÇÃO E AVALIAÇÃO .....</b>	<b>54</b>
6.1	PLANEJAMENTO E EXECUÇÃO DE TESTES FUNCIONAIS .....	54
6.2	PLANEJAMENTO E EXECUÇÃO DE TESTES COM USUÁRIOS .....	55
6.3	COMENTÁRIOS SOBRE A IMPLEMENTAÇÃO .....	55
<b>7</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>57</b>
<b>8</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>59</b>

# 1 Introdução

Moodle (Modular Object Oriented Distance Learning) é um sistema para gerenciamento de cursos (SGC) - um programa para computador com o objetivo de auxiliar educadores a criar cursos online de qualidade. Tais sistemas de educação via Internet são algumas vezes também chamados de Sistemas de Gerenciamento de Aprendizagem (SGA) ou Ambientes Virtuais de Aprendizagem (AVA).

Atualmente o Moodle alcança mais de 69 milhões de usuários em 224 países, sendo o Brasil o terceiro em número de sites registrados no sistema, atrás apenas da Espanha, em segundo, e dos Estados Unidos, em primeiro.

Como qualquer outro meio de aprendizagem, o Moodle requer a motivação do usuário aprendiz, e este nem sempre é motivado pelo estudo. Essa falta de motivação para o estudo costuma ser mais comum em ambientes de ensino médio ou inferior, onde o aluno, em geral, é menos motivado que o de ensino superior.

Tendo em vista que nos dias de hoje, com a explosão do número de usuários de smartphones, jogos são parte da vida da população como nunca antes, utilizar alguns de seus elementos no ensino pode torná-lo mais atraente para os alunos pouco motivados.

Jogos com foco na aprendizagem podem parecer, à primeira vista, não tão interessantes para os alunos. Porém, se combinados com elementos característicos dos jogos de hoje em dia, eles podem agregar um interesse tão grande quanto o dos jogos mais tradicionais. Temos um bom exemplo no jogo "Perguntados", da empresa Etermax. "Perguntados" é um jogo de perguntas e respostas relacionadas a conhecimentos gerais de diversos tópicos. Seu sucesso foi evidente, tendo mais de 50 milhões de downloads só na loja de aplicativos da Google.

Uma das funcionalidades do Moodle é a capacidade de ativar, desativar, instalar e desinstalar plugins que podem ser usados para acrescentar funcionalidades ao sistema.

No momento, o Moodle, junto de suas aplicações padrões, ainda pode encontrar dificuldades de inspirar motivação nos alunos mais desmotivados.

Uma boa forma de motivar esses alunos seria o desenvolvimento de um

plugin para o Moodle que agrupe jogos relacionados à aprendizagem da matéria dos cursos. Esses jogos devem possuir elementos de jogos tradicionais visando atrair mais os alunos que não veem o estudo com afinco.

O Projeto Final proposto por este meio consistiu no desenvolvimento deste plugin, bem como o desenvolvimento de um jogo para este plugin que se encaixe no perfil descrito.

## 2 Situação Atual

Existem várias funcionalidades e plugins no Moodle que possuem características semelhantes a jogos, seja de aprendizagem (e.g. "Quiz") ou gerais.

**Preview Optional CDF Exam**

[Start again](#)

---

**1**

The type of resume that is sometimes viewed with suspicion by employers due to the lack of specific information about work history is:

Marks: --/1

Choose one answer.

- ☐ a. Scientific reasoning
- ☒ b. Theory
- ☐ c. Literature review
- ☐ d. Intervention

That answer is correct!

[Submit](#)

**Correct**

Marks for this submission: 1/1.

---

**3**

"What brought you here today?" is an example of a .

Marks: --/1

Choose one answer.

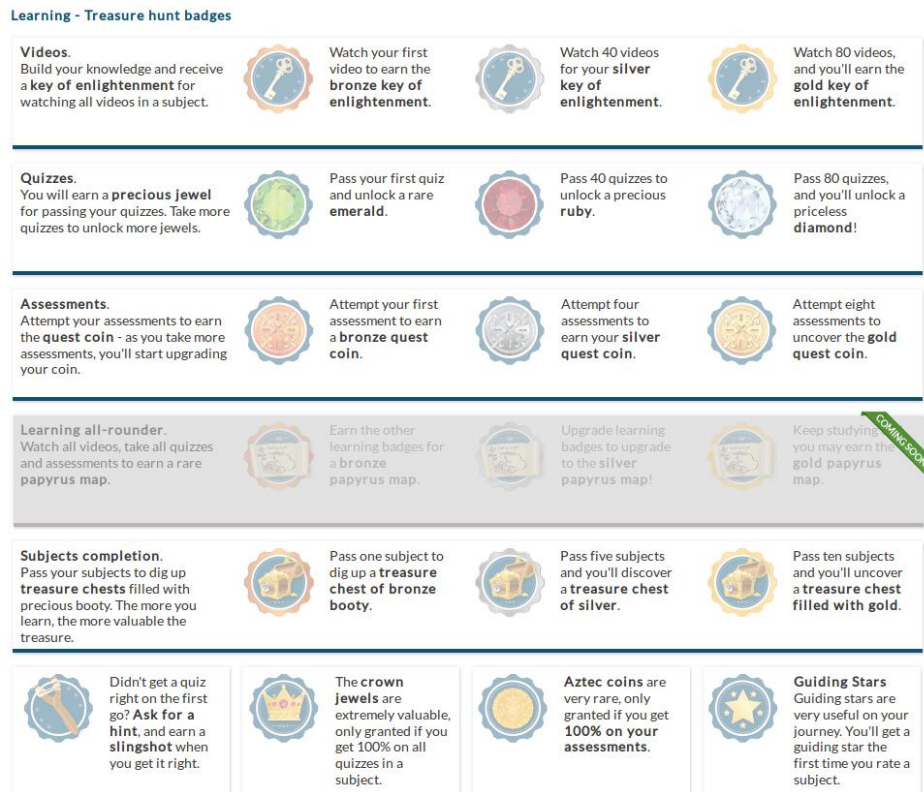
- ☐ a. Relevant question
- ☐ b. Open-ended question
- ☐ c. Irrelevant question

**Figura 1: Exemplo de "Quiz" do Moodle**

O "Quiz", aplicação de questionários padrão do Moodle, permite uma variedade de modelos para perguntas, desde questões associativas até questões aritméticas (Figura 1).

O próprio Moodle oferece um elemento muito comum nos jogos atuais: o sistema de conquistas, intitulado, no Moodle, de "Badges" (Figura 2).





**Figura 2: "Badges" - sistema de conquistas do Moodle**

Presente no Moodle desde a versão 2.5, o sistema de "Badges" é o equivalente ao sistema de conquistas dos jogos atuais, que consiste em recompensar jogadores com "medalhas" quando atingem certos critérios. No "Badges", os critérios estão relacionados às atividades gerais do curso Moodle, como assistir vídeos e realizar questionários. Apesar do elemento de jogo estar presente, o sistema não está necessariamente atingindo o grupo de alunos desmotivados, pois sua utilização é baseada nas aplicações já existentes, que podem não ser vistos por este grupo com motivação.

Outro elemento ausente, considerado por parte dos jogadores como principal motivador na hora de jogar um jogo, é o recurso do multijogador, onde um jogador interage com outro de forma cooperativa ou competitiva.

Por fim, apesar de o Moodle possuir elementos de jogos visando à motivação do aluno, estes não estão diretamente vinculados a um jogo, e sim ao programa do curso Moodle em geral. Com essa separação, é possível que os alunos percam a identificação da funcionalidade com os jogos que estão acostumados a jogar, e não se sintam motivados para sua utilização.

### 3 Objetivos

Este trabalho propõe e implementa um plugin para o Moodle que agrupe jogos educacionais relacionados com a matéria do curso. Para isso, o plugin reutiliza depósitos de perguntas e respostas já existentes em um curso cadastrado no Moodle.

Os jogos agrupados pelo plugin podem acessar o depósito de perguntas e respostas, utilizando-as de acordo com suas funcionalidades. Inicialmente o plugin contém um jogo, desenvolvido pelo presente aluno.

Os jogos desenvolvidos para este plugin poderão ter recursos multijogador, sistemas de conquistas e sistemas de rankings. Para ajudar os jogos a implementarem tais funcionalidades, o plugin lhes fornece uma identificação do usuário jogador e uma lista dos alunos inscritos no curso.

Pensando na satisfação e na facilidade de uso, o plugin e o jogo desenvolvido devem possuir interface dinâmica, agradável e que se assemelhe às características de um jogo, bem como qualquer jogo a ser desenvolvido para este plugin deverá ter.

O plugin possui dois tipos de usuários: o professor e o aluno, sendo estes possuidores de direitos de acesso diferentes. O professor terá acesso às funcionalidades de administrador, com direitos para modificações no plugin e nos jogos, enquanto o aluno terá acesso restrito à utilização do plugin e dos jogos. Para ter esse controle, é necessária a autenticação prévia dos utilizadores no sistema Moodle.

Após o desenvolvimento do plugin agregador, serão desenvolvidos dois jogos: um jogo será desenvolvido pelo aluno de Engenharia de Computação Alex Brito de Paiva, e outro pelo presente aluno.

O presente aluno desenvolverá um jogo de perguntas e respostas de múltipla escolha, onde o jogador tentará obter a maior pontuação acertando-as em um período de tempo, agregando pontos e desbloqueando conquistas. O jogador poderá ser desafiado por outros jogadores, bem como os desafiar. Para o recurso de desafio, o jogador desafiante terá que escolher outro na lista de jogadores (alunos do curso). O jogador desafiado poderá aceitar ou recusar o desafio.

O desafio será o recurso competitivo do jogo, e o que estará em disputa será a maior pontuação em um grupo de perguntas. O vencedor será o único recompensado, agregando a vitória ao seu perfil, podendo alcançar conquistas e melhores posições no ranking.

Essa valorização da competição é um dos grandes motivos das horas gastas em jogos hoje em dia, e só é possível na presença do recurso de multijogador, um recurso ausente nas aplicações de perguntas e respostas presentes no Moodle (e.g. "Quiz").

## **4 Atividades Realizadas**

### **4.1 Ambiente de desenvolvimento**

O conhecimento prévio do ambiente de desenvolvimento do Moodle era mínimo, tendo apenas na instalação do Moodle algo familiar. Já tendo usado anteriormente o software "XAMPP" como servidor web e, como consequência ("XAMPP" é um acrônimo para "X-OS, Apache, MySQL, PHP, Perl"), a plataforma Apache, MySQL e PHP, este foi o caminho escolhido para a instalação do ambiente Moodle.

Paralelamente, o conhecimento de desenvolvimento de plugins para o Moodle era inexistente. Por isso, um estudo através da própria documentação do Moodle foi realizado para aquisição de conhecimento relacionado ao desenvolvimento de plugins para a plataforma e os padrões exigidos.

### **4.2 Estudos e definição da linguagem**

Na proposta do projeto ficou pendente a linguagem que seria utilizada no desenvolvimento back-end do plugin e do jogo. A dúvida existente era entre a linguagem Java e a linguagem PHP. Foi decidido que a linguagem a ser utilizada no back-end será PHP, pois o Moodle é uma plataforma que a utiliza como linguagem, e isso pesou mais que o fato do conhecimento prévio da linguagem PHP ser básico.

Como o conhecimento das linguagens de programação a serem utilizadas no desenvolvimento era díspar, o estudo da linguagem PHP foi privilegiado, já que a experiência com esta linguagem era pequena, enquanto que o estudo da linguagem Javascript, a qual o conhecimento prévio era mais robusto, tornou-se apenas uma revisão do que já foi aprendido anteriormente. Apesar disso, foram estudadas bibliotecas de apoio à linguagem Javascript (e.g. "jQuery") para facilitar seu uso e prover novas funcionalidades, com objetivo de dispor de maiores possibilidades no desenvolvimento front-end do plugin e do jogo.

### **4.3 Estudos conceituais para o plugin**

Interfaces de agregadores de multimídias foram estudadas (e.g. "Netflix") visando absorver os moldes mais agradáveis de interação para o plugin, além de formas de transmitir informações sobre os jogos presentes no plugin de modo afável, através

de recursos gráficos e textuais.

A partir deste estudo, concluiu-se que uma interface semelhante a um carrossel seria uma opção segura para a listagem de jogos no plugin. Essa solução gráfica é utilizada para diversos propósitos quando se trata de divulgação de catálogos, como sites de notícia (e.g. globo.com) e lojas virtuais (e.g. App Store da Apple). Através de interações dinâmicas, é transmitido ao aluno o catálogo de jogos à sua disposição de forma simples e intuitiva, como demonstrado na Figura 3.



**Figura 3: Exemplo de interface carrossel**

#### **4.4 Estudos conceituais para o jogo**

Em adição aos estudos que visam um entendimento maior sobre interfaces de agregadores de multimídias, também foram estudadas interfaces de jogos que se encaixam no perfil de perguntas e respostas proposto (e.g. "Perguntados", jogo com mais de cinquenta milhões de downloads na "Google Play", da Google), a fim de extrair características que tornam a interface de perguntas e respostas mais agradável e que passe a sensação de jogo ao usuário, peça chave para o que foi proposto no projeto.

Após o estudo de jogos que se encaixam no perfil, foi decidido que o jogo "Perguntados" servirá de inspiração para o desenvolvimento do jogo proposto, pois, além de possuir muitos jogadores, compartilha diversas características definidas para a gamificação da aprendizagem (e.g. sistema de conquistas e sistema multijogador).

**Perguntados (Figura 4):** Perguntados é um jogo para dispositivos Android e iOS, onde os usuários têm como meta: vencer seus amigos em desafios de conhecimentos gerais. O desafio é feito através de perguntas divididas por categorias. Além do celular, pode ser jogado também na Web, por meio do

Facebook.



Figura 4: Jogo Perguntados

## 4.5 Definição do Jogo

O jogo desenvolvido reúne as perguntas e respostas já existentes no curso Moodle, e as utiliza para moldar um jogo de perguntas e respostas. Foram definidas algumas funcionalidades e características que devem estar presentes no jogo proposto.

### 4.5.1 Configuração

O jogo deve permitir a configuração de parâmetros como o número de perguntas em um desafio e o tempo em que o jogador tem para responder uma pergunta.

A configuração desses parâmetros deve ser permitida apenas se o usuário for identificado como administrador, ou seja, apenas os professores devem ter a permissão de modificá-los, ajustando-os à sua preferência.

### 4.5.2 Objetivo

O objetivo do jogo é obter a maior pontuação dentro de um desafio, que consiste em um grupo de perguntas e respostas, acertando o maior número de questões que for capaz, adquirindo conquistas e desafiando outros jogadores a fim de definir quem é o melhor jogador.

### **4.5.3 Jogabilidade**

O jogador inicia um jogo, um grupo de perguntas e repostas em formato múltipla escolha é carregado e dividido em fases que correspondem a uma pergunta e um limite de tempo em que o jogador deve responder a pergunta exibida. O jogador, então, pode escolher uma resposta para a pergunta proposta. Se não souber a resposta, o jogador pode escolher a opção de pular a questão, porém não haverá uma questão substituta, fazendo com que o jogador perca a chance de acertar uma das questões do grupo. Este recurso existe para desvalorizar o "chute", sem ter seu custo reduzido, pois pular uma questão resulta em uma penalidade menor que fornecer uma resposta errada. Além dessas opções, o jogador deve poder também, ao sair da página, desistir do desafio a qualquer momento, perdendo qualquer progresso feito.

O jogo deve poder ser realizado em dois modos, jogador único e multijogador.

Se desejar jogar sozinho para praticar, o jogador deve poder escolher o modo "jogador único", porém se desejar desafiar outros jogadores deve optar pelo modo "multijogador". Neste modo, o jogador deve poder escolher um desafiante dentro da lista de jogadores (alunos do curso) e desafiá-lo, saindo-se vencedor aquele que obtiver a maior pontuação em um grupo de questões. O vencedor terá a vitória agregada a seu perfil, podendo alcançar conquistas de modo multijogador e subir posições no ranking de jogadores.

## **4.6 Estudos de controle de versão**

Um estudo relacionado à metodologia de desenvolvimento foi realizado para garantir um desenvolvimento seguro e ágil. Foi utilizado para o controle de versionamento o sistema Git, afim de garantir um melhor fluxo no desenvolvimento do plugin, visto que foi realizado por dois desenvolvedores em paralelo. A ferramenta utilizada para gerenciar o Git foi o Bitbucket, visto que já era conhecida por ambos os desenvolvedores, o que garante um ambiente familiar para se utilizar.

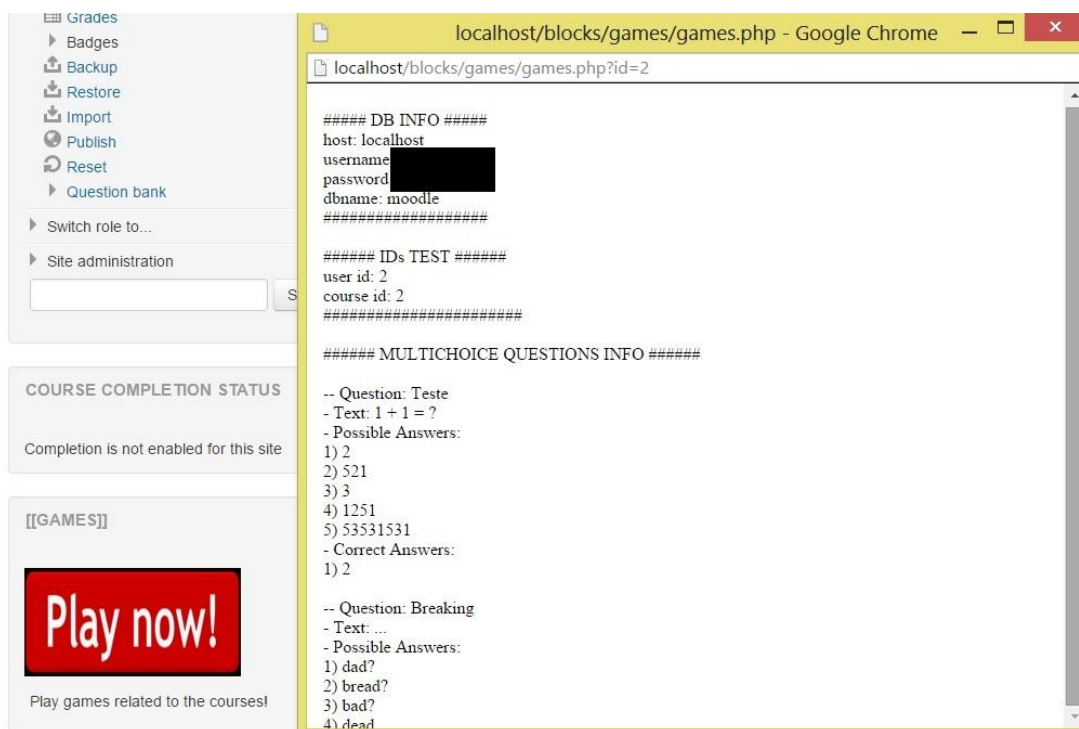
Bitbucket é um serviço de hospedagem de projetos de software que suporta repositórios que usam o sistema Git como controle de versão. O seu serviço é baseado em Python e possui versões gratuita (para até cinco desenvolvedores) e comercial. Esta foi a principal ferramenta para gerenciamento e compartilhamento

dos códigos desenvolvidos.

## 4.7 Teste e protótipos para aprendizado e demonstração

Para uma aprendizagem relacionada ao desenvolvimento de plugins para o Moodle, assim como um teste de viabilidade do que foi proposto, foi desenvolvido um simples plugin do tipo bloco, seguindo os passos propostos no tutorial de "Blocks" presente na documentação do Moodle, visando a adequação do plugin nos padrões existentes de desenvolvimento de plugins para o Moodle.

Após a instalação do bloco seguindo os padrões do Moodle, foi verificado que o bloco é capaz de redirecionar livremente o usuário para qualquer página e, sendo essa livre para qualquer tipo de desenvolvimento web, foi criada uma página de teste (Figura 5) que verifica se as funcionalidades básicas que o plugin deve fornecer aos seus jogos são viáveis, como a disponibilização dos identificadores do usuário e do curso atual, e das questões nos depósitos de questões do curso.



**Figura 5: Plugin bloco simples com testes de funcionalidades básicas**

Para a disponibilização dos identificadores, não foi necessário o acesso ao banco de dados do Moodle. O identificador de usuário atual foi obtido através da variável global "\$USER", obtida pela importação do módulo "config.php" do diretório da



instalação Moodle, enquanto que o identificador de curso atual foi obtido através da própria url da página do curso, que indica no parâmetro "id" o identificador do atual curso (Figura 6).

```
20 // gets course id by url param (get)
21 $course_id = filter_input(INPUT_GET, 'id');
22
23 // needed for global functions and vars
24 require_once __DIR__ . '/../../config.php';
25 require_login();
26
27 // stores user id to register player
28 $user_id = $USER->id;
```

**Figura 6: Método de obtenção dos identificadores de usuário e de curso**

Diferentemente da disponibilização dos identificadores, a disponibilização das questões do depósito de questões necessitou de consultas ao banco de dados do Moodle. Para o teste, o depósito foi povoado com algumas questões de múltipla escolha.

Através das tabelas "mdl\_question" e "mdl\_question\_answers", onde "mdl\_" é o prefixo de tabelas padrão na instalação do Moodle, foi possível obter as questões e suas respostas sem maiores problemas (Figura 7).

```
$rows_questions = $this->db_select("SELECT `id`, `name`, `questiontext`
FROM `".$prefix."question`
WHERE `qtype` = 'multichoice'");
$rows_answers = $this->db_select("SELECT `id`, `answer`, `fraction`
FROM `".$prefix."question_answers`
WHERE `question` = ".$row_question['id']."");
```

**Figura 7: Consultas ao banco de dados para obtenção das questões (deslocadas)**

No protótipo, assim como na versão atual do plugin, são disponibilizadas apenas as questões do tipo múltipla escolha.

Mais detalhes sobre o banco de dados, suas tabelas e as consultas realizadas sobre este estão descritos mais adiante neste documento.

## 4.8 Método

O processo de desenvolvimento do plugin e do jogo seguiu princípios da

metodologia de desenvolvimento ágil (Agile Manifesto, 2001), principalmente nos quesitos que explicitam que o funcionamento do software está acima de documentação abrangente, e que a capacidade de respostas a mudanças está acima de seguir um plano pré-definido.

## 4.9 Cronograma

O cronograma a seguir detalha as etapas do projeto final I e II:

<b>Mês</b>	<b>Tarefas</b>
04/15	<ul style="list-style-type: none"> <li>• Estudos sobre o ambiente de desenvolvimento do Moodle</li> <li>• Estudos sobre desenvolvimento de plugins para o Moodle</li> <li>• Estudos sobre a linguagem PHP</li> </ul>
05/15	<ul style="list-style-type: none"> <li>• Estudos sobre a linguagem PHP</li> <li>• Revisão da linguagem Javascript</li> <li>• Estudos sobre IHC de jogos e agregadores de multimídias</li> </ul>
06/15	<ul style="list-style-type: none"> <li>• Estudos sobre jogos que se encaixam no perfil do plugin</li> <li>• Estudos sobre o banco de dados do Moodle</li> <li>• Estudos sobre bibliotecas de apoio a linguagem Javascript</li> <li>• Preparação do ambiente de desenvolvimento do Moodle</li> <li>• Redação do Relatório do Projeto Final I</li> </ul>
07/15	<ul style="list-style-type: none"> <li>• Preparação da plataforma git "Bitbucket"</li> <li>• Desenvolvimento do plugin</li> </ul>
08/15	<ul style="list-style-type: none"> <li>• Desenvolvimento do plugin</li> </ul>
09/15	<ul style="list-style-type: none"> <li>• Desenvolvimento do plugin e do jogo</li> </ul>
10/15	<ul style="list-style-type: none"> <li>• Desenvolvimento do plugin e do jogo</li> </ul>
11/15	<ul style="list-style-type: none"> <li>• Desenvolvimento do plugin e do jogo</li> <li>• Redação do Relatório do Projeto Final II</li> </ul>
12/15	<ul style="list-style-type: none"> <li>• Criação de slides para apresentação do Projeto Final</li> <li>• Revisão final do projeto</li> </ul>

## 5 Projeto e Especificação do Sistema

O desenvolvimento deste trabalho foi dividido em duas frentes: o plugin agregador de jogos educacionais e o jogo de perguntas e respostas que se encaixe no perfil proposto.

### 5.1 Plugin

O objetivo do plugin é agregar seus jogos em uma interface que se assemelhe com outros agregadores de multimídia populares, bem como prover funcionalidades básicas aos seus jogos, como a disponibilização de questões de múltipla escolha, além de outras que são melhor detalhadas a seguir.

Além disso, também foi visada uma fácil instalação e administração do plugin para diminuir o esforço necessário por parte dos professores e administradores do curso.

#### 5.1.1 Instalação do plugin

Primeiramente, é necessário colocar a pasta do plugin na pasta "blocks" da instalação do Moodle, onde se encontram todos os plugins do Moodle do tipo bloco. A pasta "blocks" se localiza, por padrão, na pasta raiz da instalação Moodle.

Para facilitar a instalação do plugin, diversas das configurações necessárias para o funcionamento deste são obtidas automaticamente das configurações da instalação do Moodle. Para isso, é necessário configurar a pasta raiz da instalação do Moodle no arquivo "config.ini" (Figura 8), localizado na pasta raiz do plugin. Esta é a única configuração necessária para o funcionamento do plugin.

```
; EDIT WITH YOUR MOODLE DIR ROOT  
; ex1: C:/xampp/htdocs/moodle  
; ex2: C:/xampp/htdocs  
root = C:/xampp/htdocs
```

**Figura 8: Arquivo "config.ini" - configuração da pasta raiz "root" da instalação**

Agora já é possível fazer o reconhecimento do plugin pelo site de sua instalação Moodle. Normalmente, um novo plugin será reconhecido automaticamente ao entrar no site do Moodle e uma página de instalação de plugins irá ser carregada (Figura

9). Caso isso não aconteça, é necessário ir até as configurações de administração e abrir a página de instalação de plugins (Figura 9) manualmente. Para isso, estando em uma conta de administrador, basta ir nas notificações de administração, localizado no bloco de administração e na aba "Site administration".

#### Plugins check

This page displays plugins that may require your attention during the upgrade. Highlighted items include new plugins that are about to be installed, updated plugins that are about to be upgraded and any missing plugins. Additional plugins are highlighted if there is an available update for them. It is recommended that you check whether there are more recent versions of plugins available and update their source code before continuing with this Moodle upgrade.

[Check for available updates](#)

Last check done on 24 August 2015, 3:24 PM

**Number of plugins requiring your attention: 2**

[Display the full list of installed plugins](#)

Plugin name	Directory	Source	Current version	New version	Requires	Status
<b>Blocks</b>						
[pluginname.block_gamesplugin]	/blocks/gamesplugin	Additional		2015111820	Moodle 2010112400	To be installed
simplehtml		Additional	2015082416			Missing from disk!

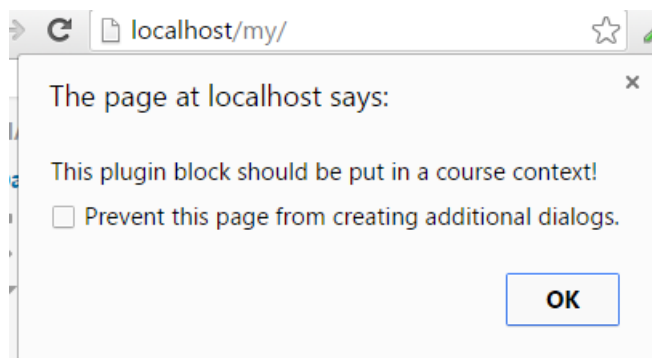
[Reload](#)

[Upgrade Moodle database now](#)

**Figura 9: Página de instalação de plugins do Moodle**

Após a instalação do plugin, falta somente adicioná-lo a uma página como um plugin do tipo bloco. Para tal, é necessário habilitar a edição da página em que o plugin será adicionado. Será possível então adicionar o bloco do plugin, através de seu nome na lista de blocos.

É importante frisar que o plugin só funcionará corretamente se seu bloco for colocado dentro do contexto de um curso, pois este plugin está diretamente ligado a um curso Moodle, já que todos os seus jogos devem estar relacionados ao curso atual ao qual o usuário pertence. Caso seja colocado fora de um curso, o plugin não funcionará (Figura 10).



**Figura 10: Erro: plugin adicionado fora de um curso**

### 5.1.2 Instalação de jogos no plugin

Pensando em facilitar a instalação de jogos, o plugin foi projetado para reconhecer automaticamente seus jogos, bem como as informações de cada um destes.

Para isso, os jogos a serem reconhecidos precisam seguir alguns padrões simples:

- A pasta do jogo deve se localizar dentro da pasta "learning\_games", localizada na pasta raiz do plugin.
- Para a visualização do logo do jogo na interface do plugin, é necessário que este seja nomeado "logo", e sua extensão precisa ser ".gif".
- É necessário possuir um arquivo PHP chamado "description.php" com as informações do jogo no formato demonstrado na Figura 11.

```
<?php

$name = 'Quiz Game'; // The game name
$description = 'A quiz where you measure your knowledge of the course!'; // A short description of the game
$category = 'quiz'; // the category of the game (board, quiz, rpg, action, other)
$author = 'Pedro'; // The author of the game
$file_name = 'quiz.php' // The name of the main game file
```

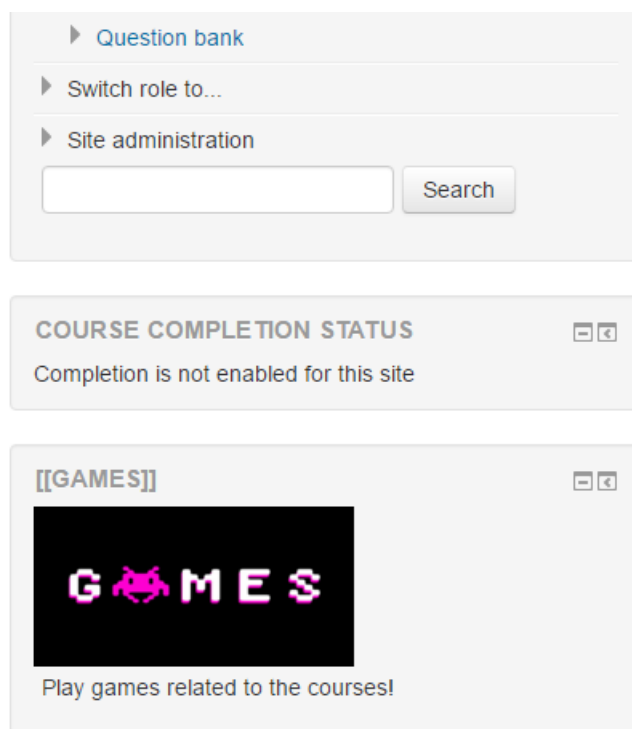
**Figura 11: "description.php" do jogo Quiz Game**

Deste modo, o jogo será reconhecido automaticamente e suas informações serão mostradas na interface do plugin. O resultado pode ser visto mais à frente neste documento, no tópico de interface do plugin (Seção 5.1.6).

Para adicionar/remover jogos do plugin, basta adicionar/remover as pastas dos jogos na pasta "learning\_games". Como o plugin sempre checa seus jogos ao inicializar, este sempre se mantém atualizado em relação aos seus jogos.

### 5.1.3 Adequação ao Moodle

O plugin foi desenvolvido para o sistema Moodle, e por isso, segue os padrões para a validação de plugin pelo sistema. Mais especificamente, foi desenvolvido como um plugin do tipo "Block" que ocupa a parte lateral da página do curso (Figura 12).



**Figura 12: Plugin de jogos educacionais estilo Moodle Block**

Para ser validado pelo Moodle como um plugin do tipo "Block", é necessário implementar um módulo PHP na pasta raiz do plugin, que estende a classe "block\_base" e implementa o método "get\_content()", atribuindo propriedades ao objeto bloco, como título do bloco, texto do conteúdo do bloco, rodapé (footer) do conteúdo do bloco, entre outros (Figura 13).

```

<?php

class block_games extends block_base {

    public function init() {
        $this->title = get_string('games', 'block_games');
        // changes current working directory to the plugin directory
        chdir(__DIR__);
    }

    public function get_content() {
        if ($this->content != null) {
            return $this->content;
        }

        $this->content = new stdClass;
        $this->content->text = file_get_contents(getcwd() . "/block_index.html");
        $this->content->footer = get_string('games:blockfooter', 'block_games');

        return $this->content;
    }
}

```

**Figura 13: Módulo "block\_games.php" - seta propriedades do objeto bloco**

Além disso, é necessário definir as "capabilities" do plugin, ou seja, as permissões que diferentes tipos de usuários do plugin vão possuir. As "capabilities" são definidas, por padrão do Moodle, dentro de um arquivo PHP chamado "access.php", que deve se localizar em uma subpasta do plugin chamada "db". As "capabilities" devem ser definidas dentro de um array de mesmo nome, com o formato especificado a seguir (Figura 14), onde basicamente é associado à "capability" um nome, um contexto, e os tipos de usuários que possuem permissão para tal "capability".

Tanto para o plugin quanto para os jogos, que receberão as permissões do plugin, apenas duas "capabilities" são necessárias: uma para administradores e outra para utilizadores (Figura 14). Isso se dá pelo fato de terem sido definidos apenas dois tipos diferentes de usuário neste projeto: o professor, com permissões de administrador, e o estudante, com permissões de utilização.

```

// Ability to use the plugin as a 'student'.
'block/games:attempt' => array(
    'riskbitmask' => RISK_SPAM,
    'cansite' => 'write',
    'contextlevel' => CONTEXT_COURSE,
    'archetypes' => array(
        'student' => CAP_ALLOW,
        'teacher' => CAP_ALLOW,
        'editingteacher' => CAP_ALLOW,
        'manager' => CAP_ALLOW
    )
),

// Ability to edit the plugin settings
'block/games:manage' => array(
    'riskbitmask' => RISK_SPAM,
    'cansite' => 'write',
    'contextlevel' => CONTEXT_COURSE,
    'archetypes' => array(
        'editingteacher' => CAP_ALLOW,
        'manager' => CAP_ALLOW
    )
),

```

**Figura 14: Capabilities de utilização e de administração**

Após definirmos as “capabilities”, podemos checar se um usuário possui uma “capability” facilmente, como mostrado a seguir na figura 15.

```

// needed for global functions and vars
require_once __DIR__ . '/../../config.php';
require_login();

// gets course id by url param (get)
$course_id = filter_input(INPUT_GET, 'id');

// gets context of the course using its id
$context = context_course::instance($course_id);

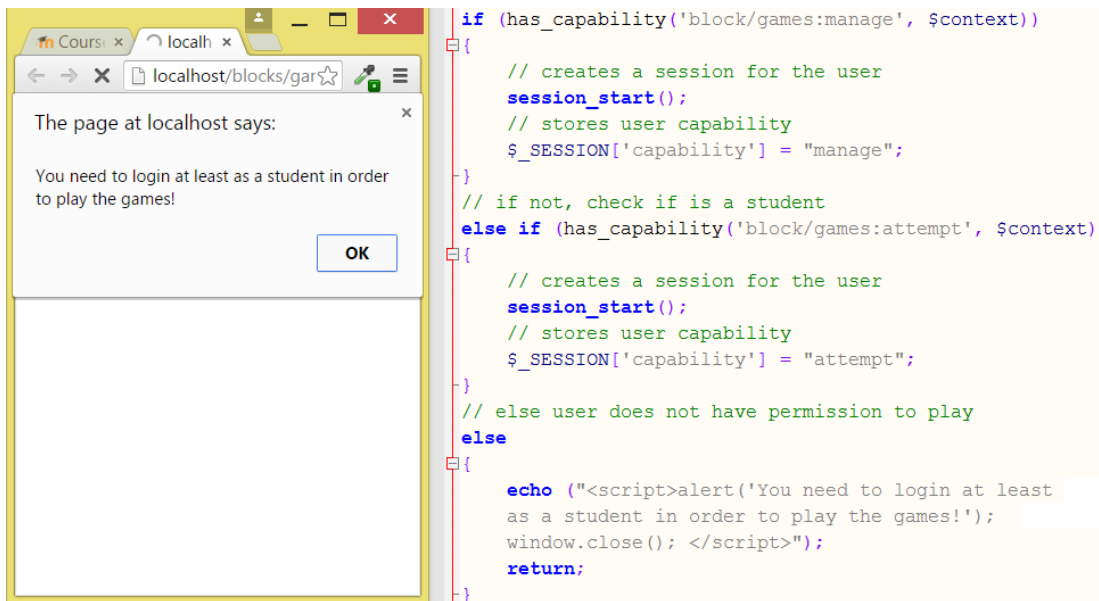
// check if user is capable of managing (editing teacher or manager)
if (has_capability('block/games:manage', $context))
{
    // creates a session for the user
    session_start();
    // stores user capability
    $_SESSION['capability'] = "manage";
}

```

**Figura 15: Checa se usuário atual possui capability "block/games:manage"**

Deste modo, podemos responder a uma ação do usuário de acordo com a “capability” deste (Figura 16).





**Figura 16: Usuário sem permissão para utilizar o plugin (guest)**

Por último, é necessário possuir um arquivo PHP na pasta raiz do plugin que possua informações da versão atual do plugin (Figura 17). Assim, o Moodle acompanhará automaticamente as atualizações do plugin quando houver modificações neste arquivo.

```

<?php

$plugin->version = 2015111820; // YYYYMMDDHH (year, month, day, 24-hr time)
$plugin->requires = 2010112400; // YYYYMMDDHH (This is the release version for Moodle 2.0)
$plugin->release = '1.0.0'; // release

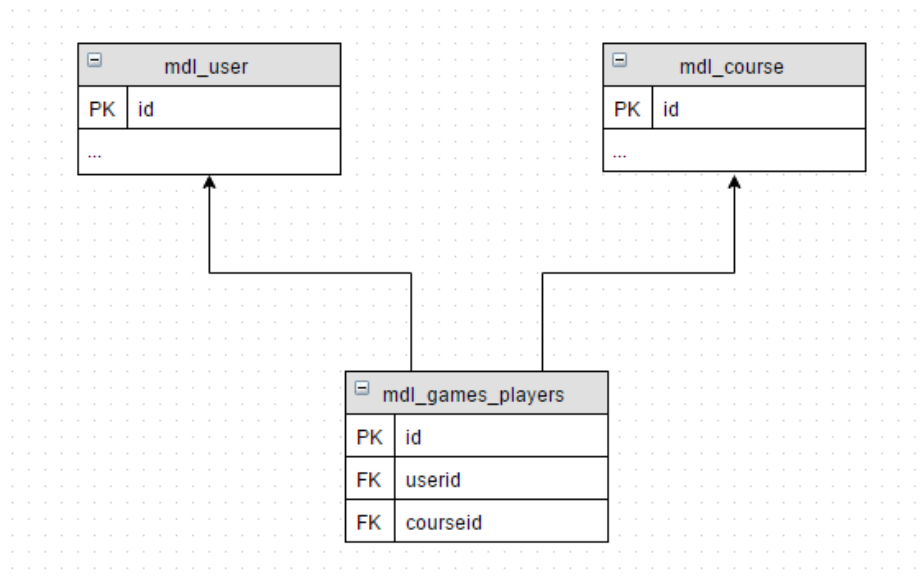
```

**Figura 17: Arquivo "version.php" com informações da versão do plugin**

#### 5.1.4 Banco de dados

Não é necessária nenhuma configuração extra relacionada ao banco de dados da instalação do Moodle. O plugin obtém as configurações necessárias a partir da própria instalação do Moodle.

Apenas uma tabela é criada e disponibilizada pelo plugin, a tabela "mdl\_games\_players", onde "mdl\_" é o prefixo de tabelas padrão na instalação do Moodle. Essa tabela contém registros de todos os jogadores do plugin, onde um jogador é uma combinação de usuário e curso. Um usuário do Moodle pode ter mais de um registro de jogador, basta este estar matriculado em mais de um curso que possui o plugin. A Figura 18 demonstra, de forma simplificada, este relacionamento.



**Figura 18: Modelo simples das tabelas do plugin (apenas mdl\_games\_players)**

Os jogadores são registrados automaticamente no momento em que um usuário entra no plugin em um curso no qual este ainda não possui registro. As funcionalidades de criação, registro e consulta a esta tabela se encontram no módulo "dbhelper.php", bem como outras funcionalidades detalhadas a seguir.

O plugin visa facilitar a vida dos desenvolvedores de seus jogos. Com este objetivo, é disponibilizado em um módulo PHP "dbhelper.php", que se encontra na pasta raiz do plugin, diversas funcionalidades que são de interesse comum aos jogos. Este módulo segue boas práticas para conexão com o banco de dados (Lionite, 2014), garantindo uma conexão única com este. A utilização deste módulo e seus métodos é recomendada aos desenvolvedores de jogos, e também é recomendado que qualquer novo método ou classe que opera sobre o banco de dados seja feito utilizando o módulo "dbhelper.php", que disponibiliza as funcionalidades básicas de conexão e consulta ao banco, garantindo a unicidade da conexão e evitando problemas relacionados ao banco de dados.

Além dessas funcionalidades básicas, o módulo concede diversas outras de nível mais alto que representam um interesse comum aos jogos desenvolvidos para este plugin. A seguir alguns de seus métodos que concedem funcionalidades comuns aos jogos são descritos a seguir:

- **db\_get\_playerID(\$userid, \$courseid)** - Dados um id de usuário e um id de curso, retorna id de jogador.

- **db\_get\_players(\$courseid)** - Dado um id de curso, retorna todos os jogadores deste curso.
- **db\_get\_multichoice\_questions(\$category)** - Dada uma categoria, retorna todas as questões de múltipla escolha desta categoria. Se não receber nenhuma categoria, retorna todas as questões registradas.
- **db\_get\_categories()** - Retorna a lista de categorias existentes no banco de questões do Moodle.
- **db\_get\_user\_name(\$userid)** - Dado um id de usuário, retorna o nome completo deste usuário.
- **db\_get\_player\_name(\$playerid )**- Dado um id de jogador, retorna o nome completo deste jogador.
- **db\_get\_course\_name(\$courseid)** - Dado um id de curso, retorna o nome deste curso.
- **get\_userID()** - Retorna o id do atual usuário "logado" no sistema Moodle.
- **get\_courseID()** - Retorna o id do atual curso em que o usuário "logado" entrou no plugin.
- **get\_user\_capability()** - Retorna uma string que representa o tipo de capability do atual usuário "logado" no sistema Moodle ("attempt" ou "manage").
- **db\_get\_category\_name(\$category)** - Retorna o nome de uma categoria de questões através do identificador recebido.

#### 5.1.5 Outros arquivos

- **block\_index.html** - Texto HTML que é carregado em "block.games.php" como o texto do conteúdo do bloco do plugin.
- **cfg.php** - Módulo responsável por obter as informações necessárias para o plugin a partir da instalação do Moodle.
- **games.php** - Módulo principal do plugin. Responsável pelo reconhecimento dos jogos, pela obtenção e propagação de dados como id do curso e capability do usuário, pelo registro automático de jogadores, além de gerar a interface do plugin.
- **js/jquery\_block\_actions.js** - Script que controla as ações realizadas sobre o bloco do plugin.
- **obj/multichoice\_question.php** - Objeto que representa uma questão do

tipo múltipla escolha, auxiliando na disponibilização deste tipo de pergunta pelo módulo dbhelper.php.

- **obj/player.php** - Objeto que representa um jogador, auxiliando na disponibilização de jogadores pelo módulo dbhelper.php.
- **css/** - Pasta com arquivos do tipo CSS referentes ao plugin, com todos os estilos que foram utilizados na interface deste.
- **images/** - Pasta com todas as imagens utilizadas na interface do plugin.

### 5.1.6 Interface

Como já dito anteriormente, a interface do plugin foi inspirada em agregadores de multimídia populares. Além disso, houve um cuidado para que esta seja simples e agradável ao usuário, sendo também informativa o suficiente. Para auxiliar no desenvolvimento da interface, foi utilizada a biblioteca CSS "PrettyPhoto".

Para ilustrar seu funcionamento foram adicionados alguns jogos ainda não desenvolvidos para fins de exemplo, além do jogo "Quiz Game" implementado pelo presente aluno (Figura 19).

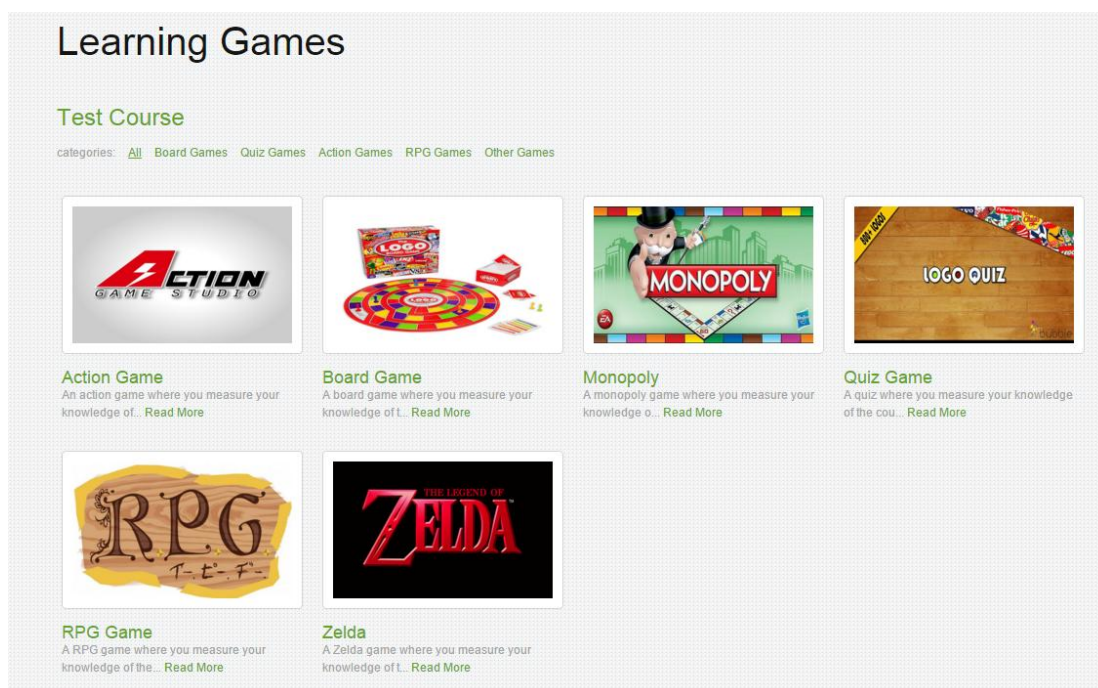


Figura 19: Interface do plugin com jogos adicionados

Além de visualizar todos os jogos disponíveis no plugin, o usuário pode filtrá-los de acordo com categorias pré-definidas no plugin, visualizando apenas os jogos da

categoria desejada (Figura 20).

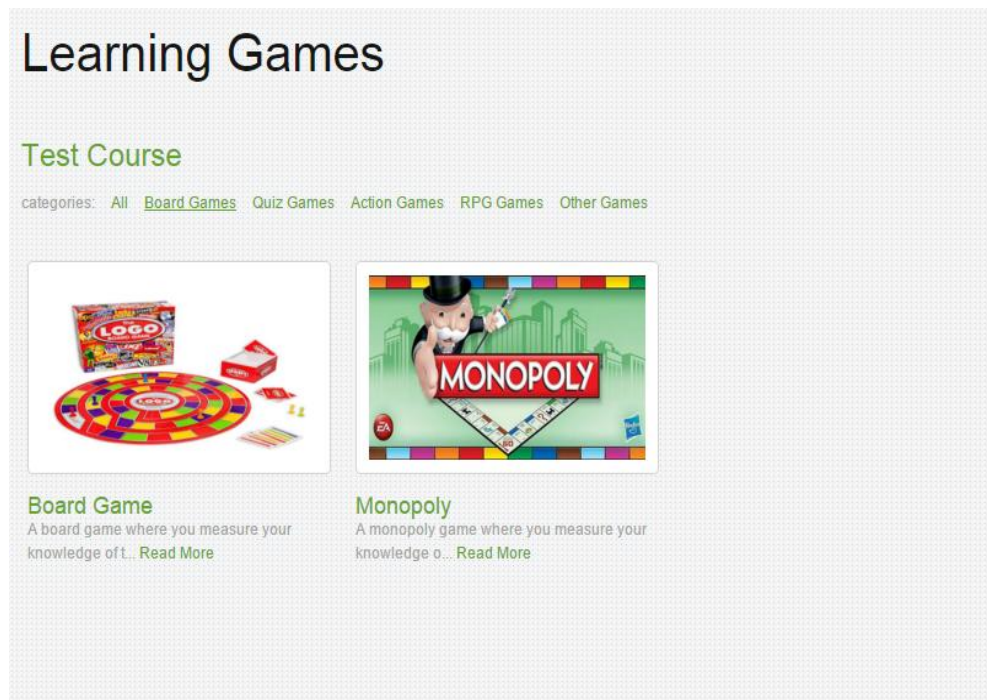


Figura 20: Interface do plugin com filtro de jogos de tabuleiro

Por último, é possível visualizar com mais detalhes a descrição de um jogo do plugin ao passar o ponteiro do mouse sobre o texto realçado "Read More" (Figura 21).

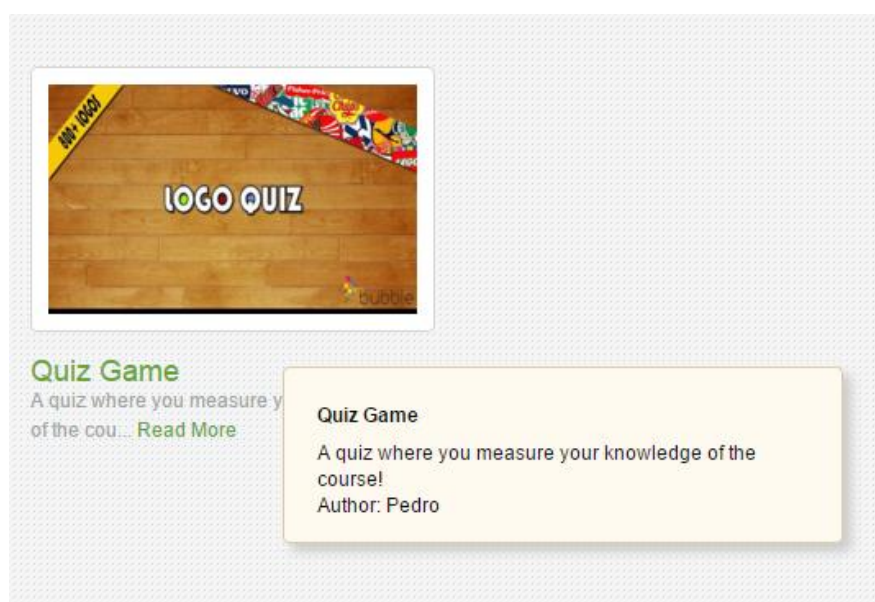


Figura 21: Interface "Read More" do plugin

## 5.2 Jogo

O jogo desenvolvido "Quiz Game" visa motivar os alunos dos cursos Moodle através de recursos comuns aos jogos populares de hoje em dia e de sua interface inspirada em jogos de trivia, como já descrito anteriormente neste documento.

O funcionamento do jogo está de acordo com o que foi previamente definido neste documento, na seção de definição do jogo (Seção 4.5).

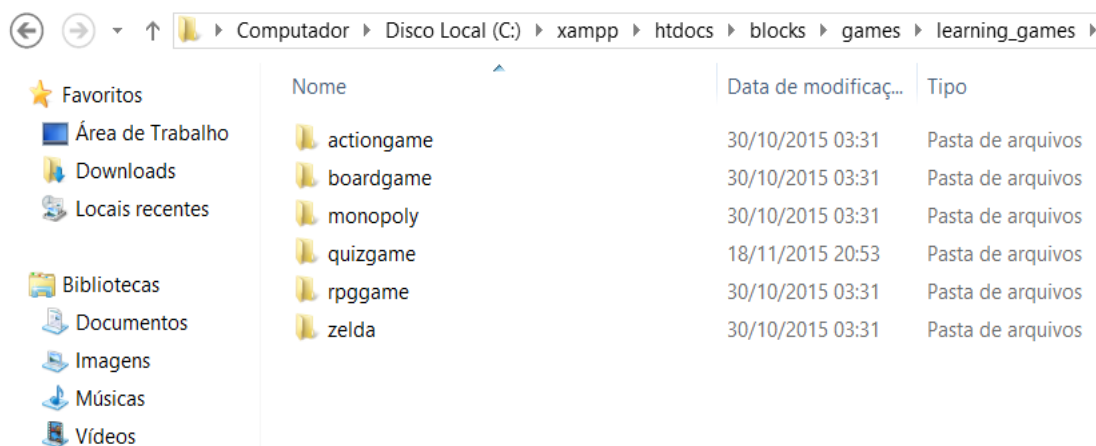
Durante seu desenvolvimento foram utilizadas as linguagens PHP, Javascript, SQL, CSS, HTML e a biblioteca javascript JQuery.

Os arquivos que compõem o jogo foram divididos da seguinte forma:

- **quizgame/** - Pasta raiz do jogo. Contém todos os arquivos PHP e INI, além do arquivo necessário para a visualização do logo do jogo no plugin, "logo.gif".
- **quizgame/css** - Pasta que contém todos os arquivos CSS utilizados pelo jogo.
- **quizgame/images**- Pasta que contém todos os arquivos de imagem utilizados pelo jogo.
- **quizgame/js**- Pasta que contém todos os arquivos de script javascript utilizados pelo jogo.
- **quizgame/sounds**- Pasta que contém todos os arquivos de áudio utilizados pelo jogo.

### 5.2.1 Instalação

A instalação do jogo se resume apenas à extração da pasta raiz do jogo na pasta "learning\_games" (Figura 22), localizada na pasta raiz do plugin. Como o jogo segue os padrões para o reconhecimento de jogos pelo plugin já discutidos neste documento, ele será automaticamente listado na interface na próxima vez em que o plugin for acessado.



**Figura 22: Pasta learning\_games que contém os jogos instalados no plugin**

### 5.2.2 Banco de dados

Aproveitando-se do auxílio que o plugin concede aos seus jogos, não é necessária nenhuma configuração relacionada ao banco de dados da instalação do Moodle para o funcionamento do jogo, pois, como recomendado anteriormente neste documento, o jogo utiliza as funcionalidades disponibilizadas pelo módulo "dbhelper.php" do plugin, que resgata todas as configurações necessárias para o acesso ao banco de dados da própria instalação do Moodle.

O módulo responsável por estender as funcionalidades do módulo "dbhelper.php" e por implementar as novas funcionalidades relacionadas ao banco de dados, necessárias para o funcionamento do jogo, localiza-se na pasta raiz do jogo, e se chama "db.php".

O módulo PHP "db.php" estende as funcionalidades do módulo "dbhelper.php" por composição, e é responsável por todas as operações do jogo relacionadas ao banco de dados da instalação Moodle, como a criação e as operações de registros nas tabelas que o jogo necessita para seu funcionamento.

O jogo implementa três tabelas necessárias para seu funcionamento. São estas:

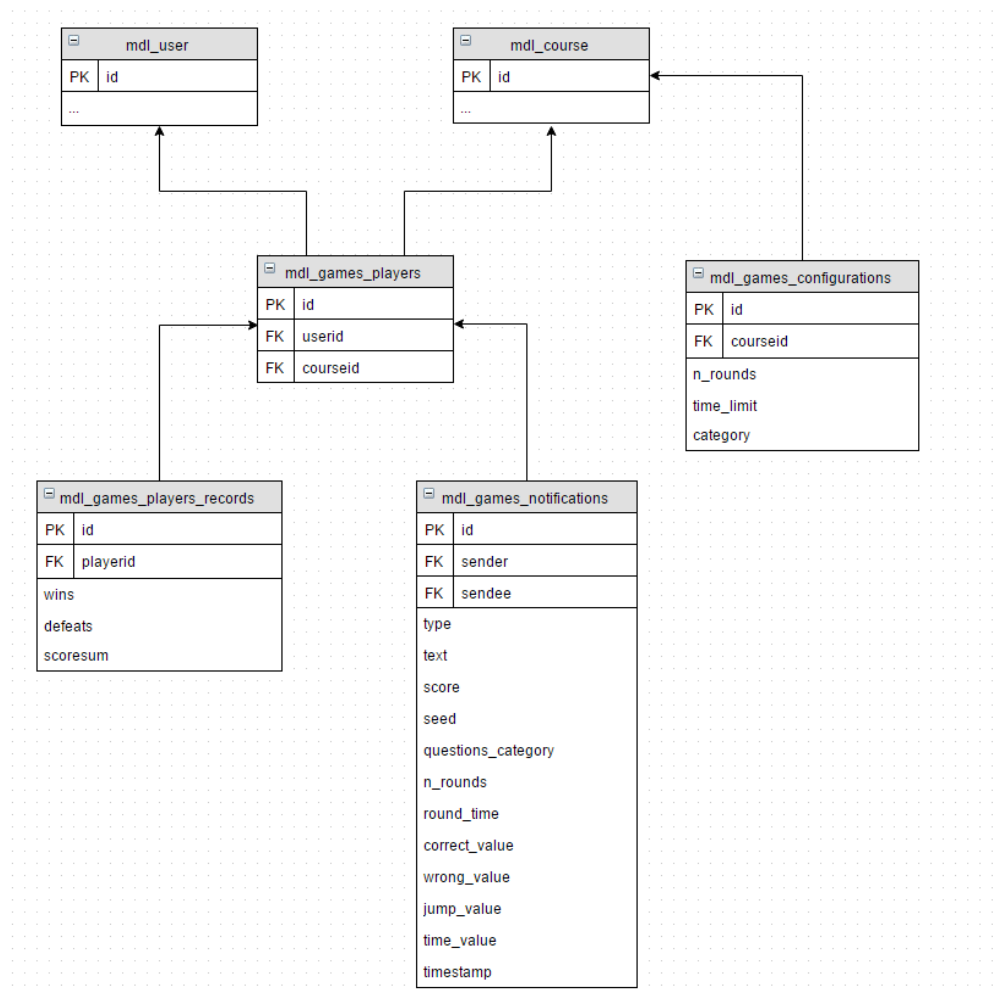
- "mdl\_games\_configurations": Tabela que armazena as configurações vigentes de um determinado curso.
- "mdl\_games\_players\_records": Tabela que armazena as informações relacionadas aos recordes dos jogadores. Como um jogador é diretamente relacionado a um curso, as informações relacionadas aos

registros são indiretamente ligados a um curso.

- "mdl\_games\_notifications": Tabela que armazena as notificações de jogo existentes no atual momento. Uma notificação é enviada por um jogador e recebida por outro jogador. Por isso, está relacionada com a tabela de jogadores e indiretamente relacionada a um curso.

Todos os nomes das tabelas acompanham o prefixo de tabelas padrão na instalação do Moodle, o prefixo "mdl\_".

O relacionamento entre as tabelas descritas acima é demonstrado de forma simplificada pela figura 23.



**Figura 23: Modelo simples das tabelas do jogo**

### 5.2.3 Estrutura web e página principal

Para uma interação mais fluida e dinâmica, atributo priorizado para se assemelhar a



interface de jogos e aumentar a satisfação do usuário, o jogo reside em apenas uma página na web. Todo novo conteúdo gerado pelo servidor é carregado dinamicamente a esta página via AJAX, com auxílio da biblioteca javascript JQuery.

A página principal é gerada pelo módulo PHP "quiz.php". Além do conteúdo que é carregado dinamicamente na página principal, esta possui também conteúdos fixos, como o "container" que envolve todo o conteúdo do jogo em uma interface similar à de um quadro negro, e a música de fundo, que, beneficiada por essa estrutura de páginas, pode ser reproduzida em todas as páginas geradas pelo jogo sem interrupções entre páginas.

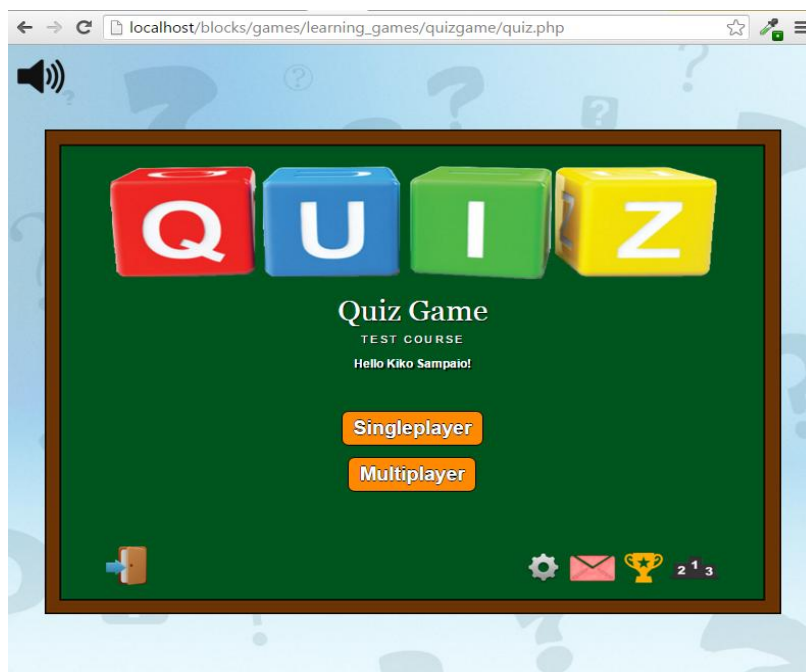
Novas páginas geradas pelo servidor são carregadas dinamicamente dentro de uma div específica da página principal, identificada como "content", utilizando a biblioteca javascript JQuery para as requisições AJAX (Figura 24).

```
$( "#content" ).load( "./game.php?&reload=1" ,  
{ "round": 1 , "score": 0, "acceptedchallenge" : "true", "seed" : seed,  
"lastround" : "false", "challenger" : challenger, "cscore" : cscore, "nid" : nid, "n_rounds" : n_rounds,  
"questions_category" : questions_category , "jump_value" : jump_value , "time_value" : time_value,  
"wrong_value" : wrong_value, "correct_value" : correct_value, "round_time" : round_time} );
```

**Figura 24: Página gerada por "game.php" sendo carregada com parâmetros post via AJAX à div "content" da página principal**

Caso o conteúdo de uma página gerada ultrapasse os limites do "container" representado pelo quadro negro, os usuários podem mover o conteúdo na direção desejada pelos métodos comuns para tal funcionalidade, como usar a roda do mouse ou utilizar as setas do teclado.

Assim que o jogo é iniciado, é carregada uma primeira página na div "content" da página principal. Esta página representa o menu inicial do jogo, e é gerada pelo módulo PHP "menu.php". Nela é possível acessar as outras páginas do jogo, bem como sair deste e voltar para a página do plugin (Figura 25). Todas as ações e requisições AJAX desta página são controladas pelo script javascript "load.js", localizado na pasta "js" da pasta raiz do jogo.



**Figura 25: Página gerada por "menu.php" carregada no início do jogo.**  
**Repare que a URL da página continua como a da página principal "quiz.php"**

As outras páginas do jogo e suas funcionalidades são detalhadas nos próximos tópicos deste documento.

#### **5.2.4 Configurações**

O jogo permite aos utilizadores com permissões de administração - professores e administradores - a alteração de suas configurações.

Utilizando a tabela "mdl\_games\_configurations", já descrita anteriormente, é possível configurar um jogo em um contexto diferente para cada curso. Essa funcionalidade foi visada pois um curso pode naturalmente precisar de parâmetros diferentes de outros cursos, como, por exemplo, mais tempo para responder suas perguntas.

Todos os métodos necessários para a persistência dos dados das configurações estão localizados no módulo PHP "db.php". Todas as funcionalidades a seguir que necessitam de operações sobre o banco de dados utilizaram métodos localizados neste módulo.

Para acessar a página de configurações, basta clicar no ícone que se assemelha a uma engrenagem localizado no menu principal (Figura 26). Este ícone

só é visível para os usuários que possuem a "capability" de administração.



Figura 26: Ícone engrenagem que carrega a página de configurações

O módulo PHP "options.php", localizado na pasta raiz do jogo, é o responsável por gerar a página de configurações (Figura 27), e o script javascript "options.js", localizado na pasta de scripts javascript "js", é o responsável por responder as ações do usuário sobre esta página, enviando requisições AJAX ao servidor se e quando necessário.

A imagem mostra uma interface web com o título "Options of Test Course" em um cabeçalho amarelo. Abaixo, há explicações para os campos: "Rounds: The number of rounds of the game", "Time: The time for each round of the game" e "Category: Moodle question bank category id". O campo "Rounds" tem opções 7, 15 e 30, com 7 selecionado. O campo "Time" tem opções 30s, 60s e 120s, com 60s selecionado. O campo "Category" tem um botão "Default for TestCourse" com uma seta. No canto inferior direito, há dois botões: "Save" e "Return".

Figura 27: Página de configurações gerada por "options.php"

Nesta página o usuário administrador pode escolher a quantidade de rounds e o tempo para responder cada questão do jogo. Além disso, é possível escolher a categoria das questões no banco de questões a serem utilizadas no jogo. Todas as categorias existentes são mostradas como opção em uma lista (Figura 28). As configurações são salvas através do botão "Save". Para descartar as mudanças,

basta retornar ao menu principal sem salvar, através do botão "Return".

Todas as questões cadastradas no banco de questões do Moodle possuem uma categoria. Para criar módulos de questões de acordo com o contexto atual da matéria do curso, basta salvar essas questões em uma categoria diferente, podendo então selecionar a nova categoria na página de configurações.



**Figura 28: Lista de categorias existentes**

Se o utilizador administrador desejar parâmetros diferentes nas configurações de tempo e de número de rounds, basta modificar estes no arquivo "options.ini" (Figura 29), localizado na pasta raiz do jogo.

```
; options of round time and number of rounds
; to be displayed in the options menu to managers
; WARNING: in order to be displayed correctly,
; only three options of each must exist

rounds[op1] = 7
rounds[op2] = 15
rounds[op3] = 30
time[op1] = 30
time[op2] = 60
time[op3] = 120
```

**Figura 29: Arquivo "options.ini"**

### 5.2.5 Modo jogador único

O modo jogador único se resume a uma partida completado jogo com as configurações vigentes. Neste modo não há desafiantes, tampouco recompensas. O jogador pode utilizá-lo para praticar e para aprendizagem.

A página da partida (Figura 30) pode ser acessada através do botão "Singleplayer", localizado no menu principal. A página é gerada pelo módulo PHP "game.php", e as ações sobre essa página são controladas pelo script javascript "actions.js".



**Figura 30: Página de partida gerada pelo módulo "game.php"**

Nesta página é possível visualizar todas as informações da atual rodada – as questões e suas respostas, o tempo restante para responder, a questão atual e a quantidade de questão da partida. Vale ressaltar que a ordem das respostas de uma questão é sempre sorteada com o objetivo de evitar que o jogador faça uma ligação entre a resposta da questão e sua posição na ordem das respostas.

Durante uma rodada, o jogador pode responder a pergunta, pular a questão, ou o tempo para respondê-la pode acabar. Dependendo do que aconteceu, a pontuação do jogador é afetada de modo diferente, como já explicada no tópico de definição do jogo. A influência que cada tipo de evento causa na pontuação de uma partida é definida no arquivo "config.ini" (Figura 31) e pode ser modificada livremente.

```
;round time limit
round_time = 60

;correct answer value
correct_value = 25

;wrong answer value
wrong_value = -10

;jump question value
jump_value = -4

;time limit reach value
time_value = -3
```

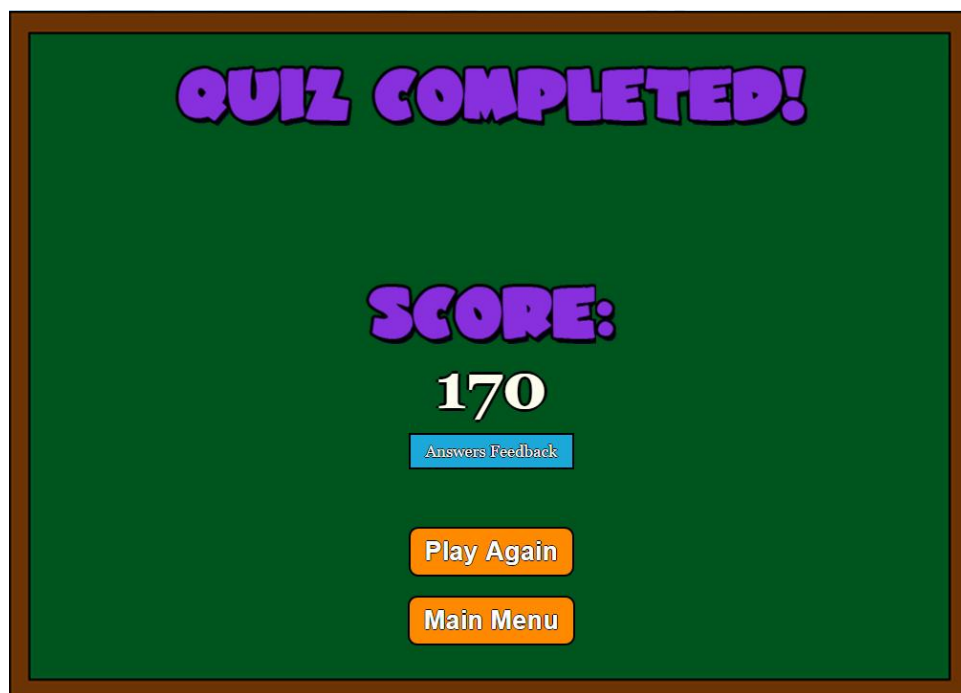
**Figura 31: Arquivo "config.ini"**

Uma partida tem um número "n" de rodadas (rounds), onde esse número "n" é definido nas configurações de jogo por um usuário administrador do curso ou é o valor padrão da instalação do jogo.

Para realizar uma partida completa, a página gerada pelo módulo PHP "game.php" é carregada via AJAX "n" vezes, atualizando as informações da página de acordo com a atual rodada.

As questões que irão compor as rodadas da partida são sorteadas antes da primeira rodada, e são enviadas para todos os próximos rounds através de uma espécie de "seed", capaz de gerar de volta qualquer das questões sorteadas.

Ao final da última rodada, é gerada uma nova página que contém todas as informações sobre a partida que acabou de ocorrer (Figura 32). Essa página é gerada pelo módulo PHP "end.php", e as ações sobre essa página são controladas pelo script javascript "end.js".



**Figura 32: Página de final da partida gerada por "end.php"**

O jogador pode então visualizar seu desempenho na partida, simbolizado pela pontuação, e pode escolher entre jogar de novo ou voltar para o menu principal.

Exclusivamente no modo jogador único, há uma opção extra para o jogador, onde este pode ver uma espécie de relatório contendo um feedback de suas respostas na partida. Para visualizar esse feedback, basta clicar na opção "Answers Feedback", localizado embaixo da pontuação da partida.

A página de feedback (Figura 33) é gerada pelo módulo PHP "feedback.php", e as ações sobre essa página são controladas pelo script javascript "load.js". Nela é possível visualizar cada questão e resposta dada pelo jogador, diferenciando as respostas certas das erradas para ajudar na aprendizagem do jogador.



**Figura 33: Página gerada pelo módulo "feedback.php"**

Essa dinâmica de partida se repete para o modo multijogador, porém neste modo há a interação competitiva com outros jogadores, com recompensas aos vencedores dos desafios. O modo multijogador é mais bem detalhado nos próximos tópicos.

### **5.2.6 Modo multijogador**

O modo multijogador, semelhante ao modo jogador único, se resume a uma partida completa do jogo com as configurações vigentes. A diferença é que neste modo há a interação com outros jogadores de forma competitiva, além de recompensas aos que se saírem vitoriosos de um desafio. Esta é a funcionalidade mais importante para a motivação dos alunos, visto que a competição é considerada um dos fatores mais motivadores de um jogo.

Como no modo jogador único, uma partida ocorre de forma completa com as configurações vigentes do curso. Porém, antes de entrar na partida, o jogador precisa escolher um outro jogador do curso para desafiar. Para isso, assim que a opção "Multiplayer" é escolhida, uma página que lista os outros jogadores do curso e suas informações é carregada (Figura 34). Essa página é gerada pelo módulo PHP "multiplayer.php", e as ações sobre essa página são controladas pelo script javascript "multiplayer.js".





**Figura 34: Página com lista de jogadores gerada pelo módulo "multiplayer.php"**

Nesta página é possível ver as informações do jogador atual, e também as informações dos outros jogadores do curso candidatos ao desafio. É possível visualizar o nome, o número de vitórias e derrotas, e a quantidade acumulada de pontuação em jogos do tipo multijogador.

Após escolher um desafiante, basta clicar em cima de seu bloco para o desafiar. Para evitar erros, uma página de confirmação será carregada (Figura 35). Todas as páginas de confirmações que são necessárias para o jogo são geradas pelo módulo PHP "confirmation.php" e controladas pelo script javascript "confirmation.js".

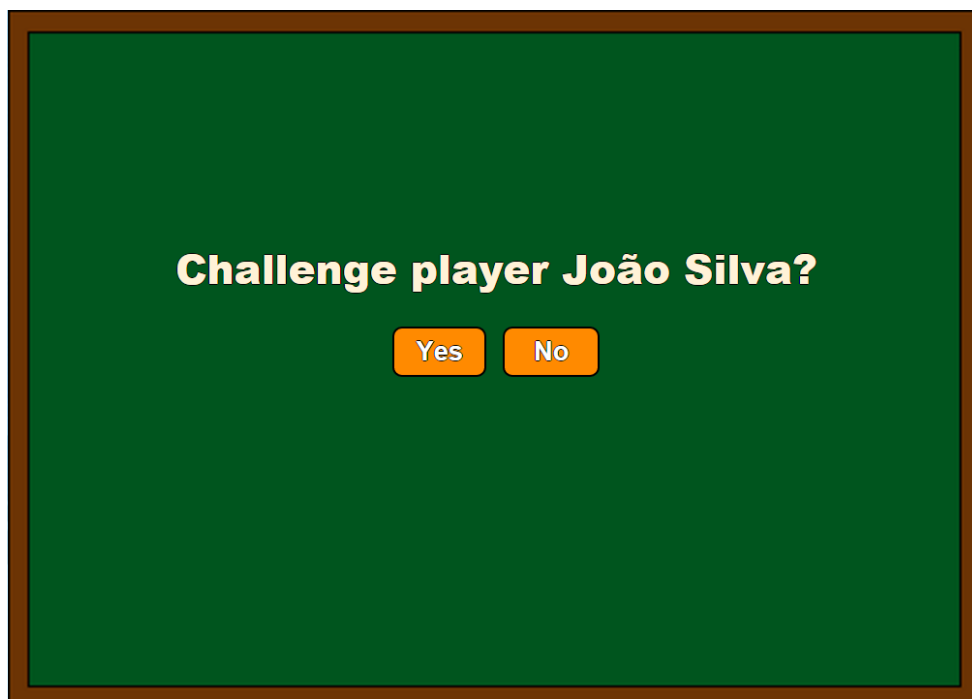


Figura 35: Página de confirmação de desafio gerada por "confirmation.php"

Confirmado o desafio, uma partida é iniciada com a mesma dinâmica de uma partida modo jogador único, diferindo apenas na página final da partida (Figura 36).

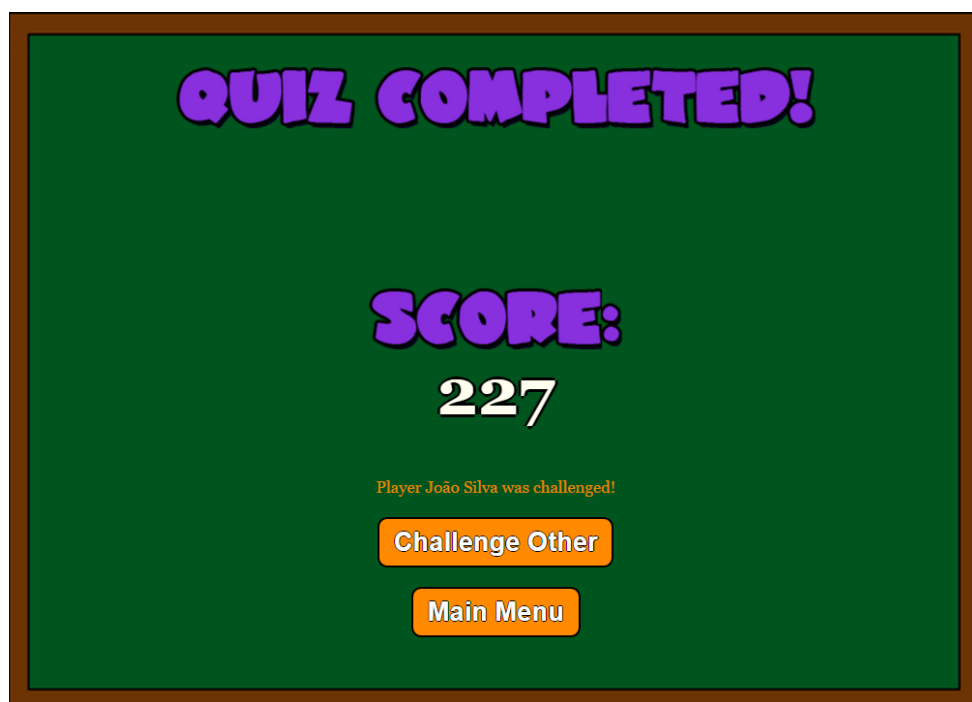
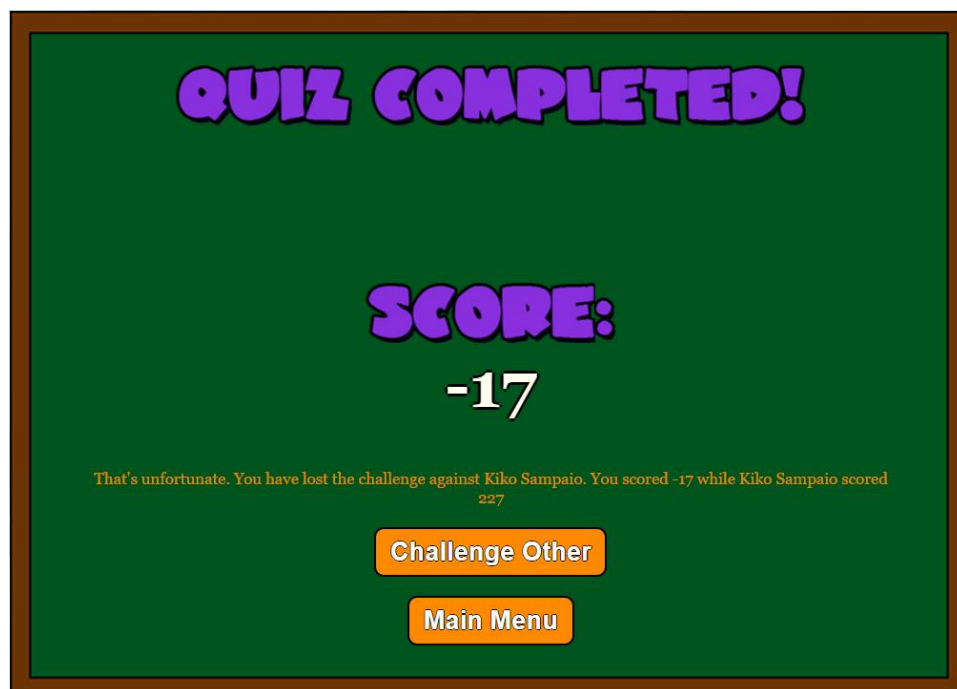


Figura 36: Página de final de um desafiante multijogador gerada por "end.php"

Além das diferenças na interface da página final, como o texto indicando que o jogador desafiado foi desafiado com sucesso, e a opção de desafiar outro jogador ao invés de jogar novamente uma partida modo jogador único, no final de uma partida modo multijogador uma notificação de desafio é enviada para o jogador desafiado.

Cabe ao jogador desafiado aceitar ou recusar o desafio. Caso o desafio seja recusado, uma nova notificação é enviada de volta ao jogador desafiante informando que o desafio foi recusado. Caso o desafio seja aceito, o jogador desafiado irá jogar uma partida com as mesmas questões e configurações que o jogador desafiante jogou. Após a partida, uma nova página de final de partida é gerada, desta vez com informações sobre o vencedor do desafio (Figura 37).



**Figura 37: Página de final de partida multijogador gerada por "end.php" vista pelo jogador desafiado**

Tendo já informado ao jogador desafiado as informações finais sobre o desafio, uma notificação é enviada ao jogador desafiante com as mesmas informações, para que este saiba o resultado final do duelo. Além disso, são atualizadas as informações no banco de dados relacionadas aos recordes dos jogadores envolvidos no desafio. As atualizações são feitas através do módulo PHP "db.php" e são realizadas na tabela "mdl\_games\_players\_records".

A funcionalidade de notificações utilizada no modo multijogador é detalhada nos próximos tópicos deste documento.

### **5.2.7 Notificações**

Para tornar possível a interação entre os jogadores do curso de forma assíncrona, foi desenvolvido um sistema de notificações que se assemelha bastante com um sistema de troca de mensagens, porém com uma diferença - as mensagens são definidas pelos eventos que geram as notificações, ao invés de serem textos criados pelos usuários.

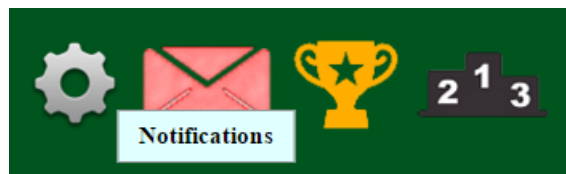
O sistema conta com a tabela "mdl\_games\_notifications", já descrita anteriormente. É nesta tabela que todas as notificações e suas informações são armazenadas.

Uma notificação consiste basicamente de um texto, um remetente e um destinatário. Para acomodar as diferentes necessidades do jogo, estas são caracterizadas também por um tipo. Dependendo do tipo de notificação, uma notificação pode ter mais informações anexas a si.

Os tipos de notificação criados para acomodar as necessidades do jogo são:

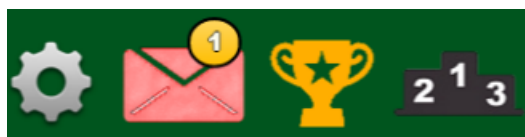
- Notificação de desafio: Notificação enviada ao fim de uma partida multijogador. Para esse tipo de notificação é necessário também anexar à notificação informações sobre o duelo, como a *seed* geradora das questões e as configurações em que a partida foi jogada pelo desafiante.
- Notificação de mensagem: Notificação simples que apresenta apenas um texto.
- Notificação de conquista: Notificação enviada a jogadores quando estes atingem novas conquistas.

Para visualizar as notificações que possui, o jogador pode entrar na página de notificações, gerada pelo módulo PHP "notification.php" e controlada pelo script javascript "notification.js", através do menu principal. Para isso, basta clicar no ícone que se assemelha a uma carta (Figura 38).



**Figura 38: Ícone carta que carrega página de notificações**

Para poupar tempo do jogador, é possível visualizar a quantidade de notificações que este possui antes mesmo de entrar na página de notificações. Caso haja notificações para o jogador, um círculo amarelo com o número de notificações que este possui aparece em cima do ícone de carta (Figura 39). Não é necessário atualizar a página para este ícone aparecer, pois há um script javascript ("notechecker.js") com um timer que checa de tempos em tempos se há alguma notificação para o jogador, e caso haja, ele carrega o círculo informativo.



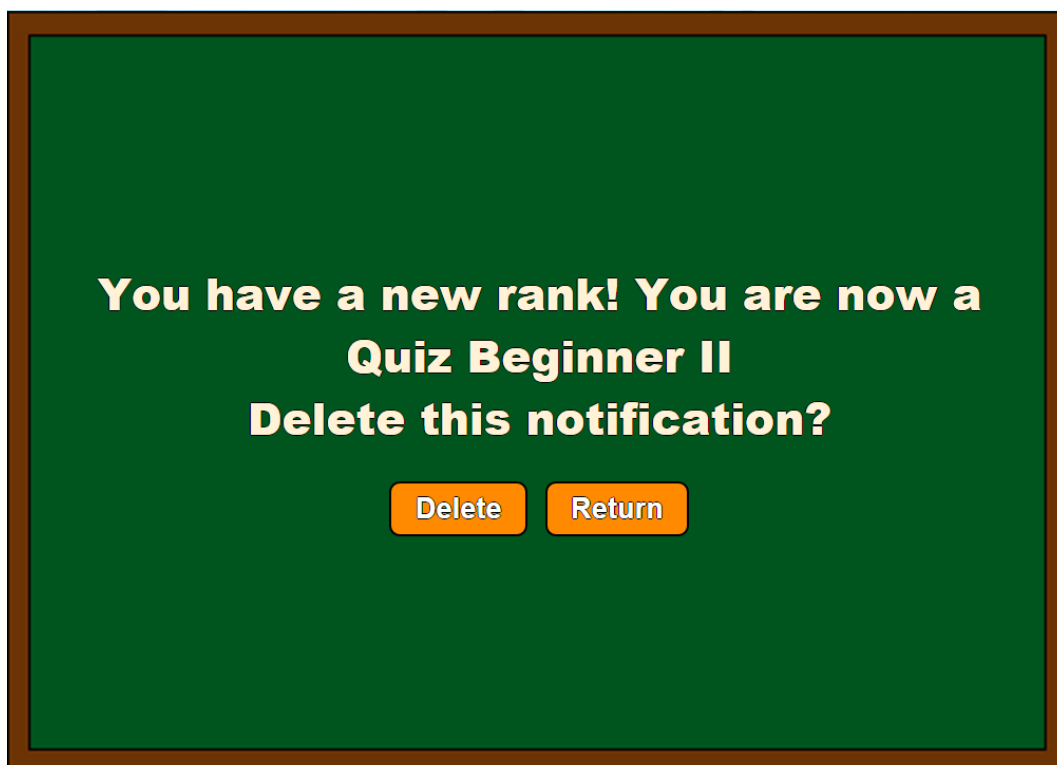
**Figura 39: Ícone carta com círculo informativo**

Assim que carregada, a página de notificações exibe as notificações do jogador em uma lista parecida com a lista de jogadores a serem desafiados do modo multijogador, na qual cada notificação é exibida de forma diferente de acordo com seu tipo (Figura 40).



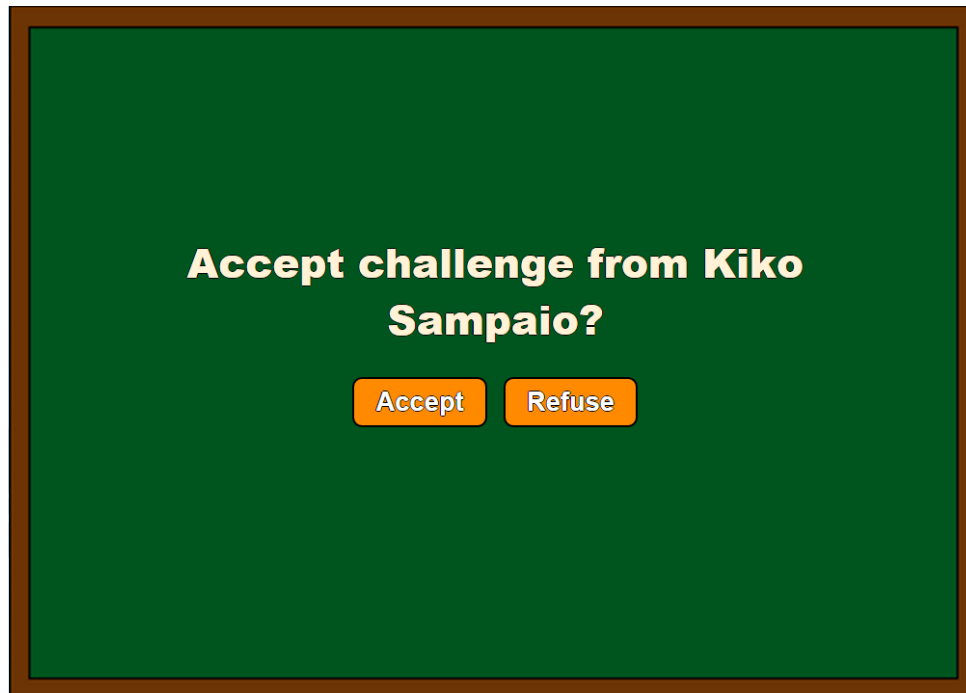
**Figura 40: Página de notificações carregada por "notifications.php" com notificações de todos os tipos existentes no jogo**

Ao clicar em uma notificação, uma página de confirmação contendo o texto da notificação e opções de confirmação contextualizadas com o tipo de notificação aberta é exibida (Figura 41). Esta página é carregada pelo módulo PHP "confirmation.php" e é controlada pelo script javascript "confirmation.js".



**Figura 41: Página de confirmação de uma notificação do tipo conquista**

O jogador pode, basicamente, deletar ou não deletar a notificação aberta. Caso a notificação seja do tipo desafio (Figura 42), o jogador pode aceitar ou recusar o desafio. Caso aceite, uma partida com as questões e configurações do jogo do desafiante, obtidas do próprio registro da notificação no banco de dados, iniciará. Nesse caso, a notificação de desafio será deletada ao final da partida. Caso o jogador recuse o desafio, a notificação é deletada instantaneamente.



**Figura 42: Página de confirmação de uma notificação do tipo desafio**

### **5.2.8 Conquistas**

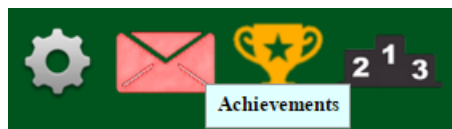
Um outro sistema muito comum nos jogos de hoje em dia é o de conquistas, que representa um progresso que é alcançado pelos triunfos dos jogadores.

Há várias formas de se motivar um jogador dando a sensação de progresso no jogo, como as conquistas do tipo missão, que são alcançadas quando o jogador realiza alguma façanha específica.

Para a versão atual do jogo, foi desenvolvido apenas um tipo de conquista, baseado nos "ranks", ou patentes, comuns aos jogos de hoje em dia. Este tipo é o que mais realça o progresso do jogador e, possivelmente, o que mais o motiva, pois possui elementos de RPG (Role-Playing Games), um tipo de jogo dos mais populares no qual a evolução e o progresso do jogador é priorizada.

É possível visualizar seu atual "rank" e seu atual progresso acessando a página de conquistas através do menu principal. Para isso, basta clicar no ícone que se assemelha a um troféu (Figura 43).





**Figura 43: Ícone troféu que carrega a página de conquistas**

Ao clicar no ícone, a página de conquistas é carregada (Figura 44). Esta é gerada pelo módulo PHP "achievement.php" e é controlada pelo script javascript "load.js".



**Figura 44: Página de conquistas gerada por "achievement.php"**

Nesta página, é possível visualizar o "rank" e o progresso de vitórias atual do jogador, além da quantidade necessária para atingir o próximo "rank".

Um "rank" é associado a um nome, um ícone e à quantidade de vitórias necessárias para atingi-lo. Todos os "ranks" existentes e as informações relacionadas ao nome, a quantidade de vitórias e a ordem destes são obtidas do arquivo "rank.ini" (Figura 45), e podem ser facilmente modificadas.

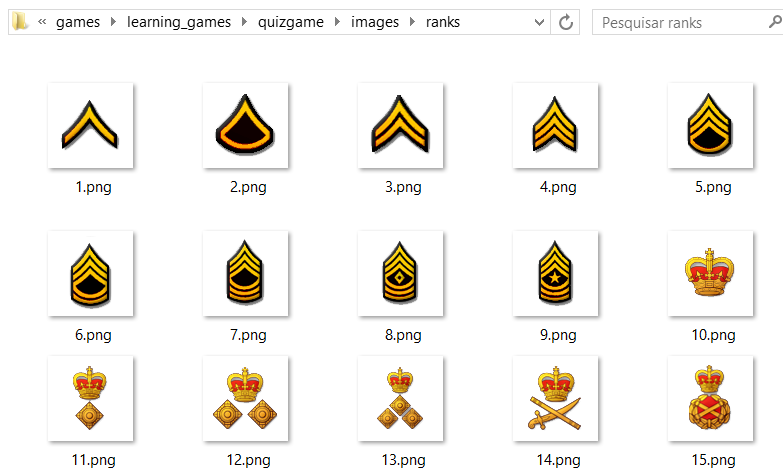
```

1 ; List of ranks and the number of wins to achieve it
2 ; WARNING: In order to properly be represented in the game
3 ; the ranks must be in ascending order
4 ; format:
5 ; rank[Name Of the Rank] = Number of Wins to Achieve the Rank
6 rank[Quiz Beginner I] = 0
7 rank[Quiz Beginner II] = 3
8 rank[Quiz Beginner III] = 9
9 rank[Quiz Beginner IV] = 18
10 rank[Quiz Beginner V] = 30
11 rank[Quiz Intermediate I] = 45
12 rank[Quiz Intermediate II] = 65
13 rank[Quiz Intermediate III] = 80
14 rank[Quiz Intermediate IV] = 110
15 rank[Quiz Advanced I] = 150
16 rank[Quiz Advanced II] = 200
17 rank[Quiz Advanced III] = 260
18 rank[Quiz Advanced IV] = 330
19 rank[Quiz Advanced V] = 410
20 rank[Quiz Master] = 500

```

**Figura 45: Arquivo "rank.ini" e as configurações dos ranks**

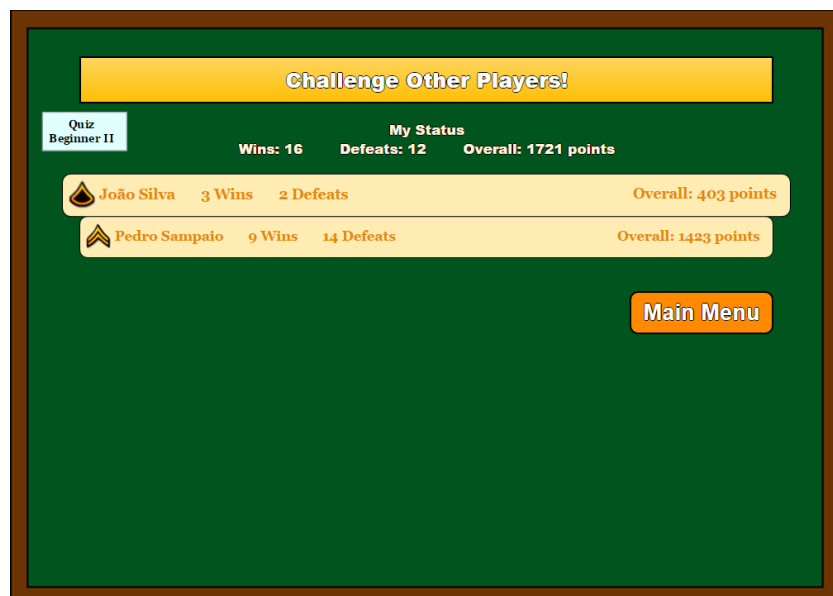
Os ícones associados aos "ranks" são relacionados a partir da ordem em que os "ranks" são definidos neste arquivo, onde o primeiro "rank" definido é o "rank" inicial de todos os jogadores, e o último é o último "rank" possível de se atingir. Os ícones se encontram dentro da pasta "ranks" (Figura 46), localizado na pasta de imagens do jogo "images".



**Figura 46: Ícones dos ranks enumerados de acordo com a ordem de um rank**

Os ícones devem estar no formato de imagem "PNG" e ser enumerados de forma que o ícone do primeiro "rank" seja chamada de "1", e o ícone do último "rank" seja a quantidade de "ranks" disponíveis no jogo.

Para valorizar o sistema de "ranks", é possível visualizar os "ranks" de outros jogadores quando estes são listados em um contexto de desafio (Figura 47).



**Figura 47: "Ranks" dos jogadores listados no modo multijogador**

Deste modo, os jogadores poderão exibir seu progresso e conquistas para os outros, motivando-os a sempre estar com o "rank" mais alto. Para ajudar a motivar esta competição, é também recomendável que os ícones e os nomes dos "ranks" mais altos sejam gradativamente mais atrativos.

Como já demonstrado no tópico de notificações, ao atingir um novo "rank" o jogador recebe uma notificação do tipo conquista informando sobre sua nova conquista.

O sistema de conquistas, diferentemente dos outros sistemas do jogo, não está relacionado a um curso, portanto este se apresenta da mesma forma para todos os cursos presentes no sistema Moodle.

### **5.2.9 Rankings**

A última funcionalidade a ser descrita por este documento, é a mais óbvia quando se pensa em competição - a classificação dos jogadores do curso. Não há competição sem um primeiro colocado, segundo colocado, ou um último colocado.

O sistema de rankings visa exatamente obter este efeito através do posicionamento dos jogadores, de acordo com uma ordem desejada, em uma tabela acessível para todos os jogadores do curso. Assim, todos podem ver suas colocações em relação aos outros jogadores do curso, bem como ver aquele mais

bem colocado, ou aquele de pior colocação.

Para gerar tal tabela, o sistema utiliza as informações armazenadas na tabela "mdl\_games\_players\_records". Nela é possível acessar todas as informações referentes ao desempenho do jogador no curso, como já detalhado anteriormente.

Diferentemente do sistema de conquistas, este sistema está diretamente relacionado a um curso Moodle e seus jogadores. Um ranking de um curso precisa ser diferente do ranking de um outro curso, pois cada curso possui seus jogadores e suas questões.

É possível acessar a página de ranking através do ícone que se assemelha a um pódio, localizado no menu principal (Figura 48).



**Figura 48: Ícone pódio que carrega a página de ranking**

A página de ranking (Figura 49) é gerada pelo módulo PHP "ranking.php" e é controlada pelo script javascript "ranking.js".



Figura 49: Página de ranking ordenada por vitórias, gerada por "ranking.php"

Nesta página podemos ver informações sobre o jogador atual e sobre todos os outros jogadores do curso. É possível ordenar a página de acordo com qualquer uma das informações exibidas: vitórias, derrotas, acumulado de pontos e *ratio*, onde *ratio* é a razão entre o número de vitórias e o número de derrotas.

O *ratio* é a medida que melhor dimensiona a qualidade de um jogador, pois este não depende da quantidade de tempo em que um jogador passou desafiando outros, diferentemente das outras medidas. A figura 50 a seguir demonstra como a página de ranking deste mesmo curso fica quando ordenada por *ratio*. Nela, podemos ver como um jogador com menos tempo de jogo pode assumir a primeira colocação do ranking quando este é ordenado por *ratio*.



Figura 50: Página de ranking ordenada por ratio, gerada por "ranking.php"

## **6 Implementação e Avaliação**

Todas as funcionalidades presentes no plugin e no jogo foram testadas de forma manual, procurando abordar todos os possíveis casos de interação do usuário com o sistema, assim como os diferentes comportamentos possíveis de cada método implementado.

Durante o desenvolvimento do jogo e do plugin, foi visada uma compatibilidade entre os navegadores mais comuns (cross-compatibility). Para isso, ambos foram testados nos navegadores Mozilla Firefox, Google Chrome e Internet Explorer.

### **6.1 Planejamento e execução de testes funcionais**

Durante o desenvolvimento, os testes manuais cobriram todas as funcionalidades implementadas de forma incremental, onde a cada nova funcionalidade um conjunto de testes foram realizados sobre esta.

Basicamente, os testes se preocuparam com as fronteiras condicionais e as entradas e saídas de métodos implementados.

Os erros encontrados durante o desenvolvimento foram achados por meio de debug simples, onde a saída de erro utilizada foi a padrão.

Para realizar os testes, foram criados módulos de testes responsáveis por testar as funcionalidades de um módulo específico. Para fins de demonstração, na versão final do sistema ainda é possível acessar o módulo de testes do módulo "db.php", chamado "dbtest.php".

Em relação a utilização das funcionalidades pela interface do sistema, foram testados diversos casos de uso através de observações de seus comportamentos de acordo com a interação realizada. A seguir, alguns dos casos de uso mais gerais que foram testados:

#### **Plugin:**

- Filtrar os jogos do plugin por categorias.
- Ler mais informações sobre um jogo.
- Entrar em um jogo.

#### **Jogo:**

- Jogar o modo jogador único.

- Jogar o modo multijogador.
- Tirar a música de fundo.
- Modificar as configurações do jogo.
- Visualizar suas notificações.
- Visualizar suas conquistas.
- Visualizar o ranking do curso.

Todos estes casos de uso abrangem diversos outros, que também foram testados para a identificação de erros e comportamentos não desejados em várias situações, como: múltiplos disparos de cliques do mouse, atualizações de páginas forçadas, redimensionamento de janela, ampliação e redução de tela, exibição de strings com caracteres especiais, entre outros.

Qualquer erro ou comportamento não desejado foram resolvidos instantaneamente. Durante o desenvolvimento, não houve nenhum tipo de catalogação de erros.

## **6.2 Planejamento e execução de testes com usuários**

Além de todas as versões produzidas do plugin e do jogo, várias das possíveis mudanças de interface foram expostas para potenciais usuários do sistema, os tornando parte ativa do desenvolvimento do sistema, importante princípio de uma metodologia ágil, utilizada durante o desenvolvimento.

Durante a execução dos testes com os usuários, foram priorizadas as respostas destes em relação à satisfação geral do uso do sistema.

Mudanças na cor e nos ícones de botões são exemplos de resultados obtidos através dos testes com os usuários.

## **6.3 Comentários sobre a implementação**

Para a realização dos testes, foi necessário carregar diversos tipos de dados no sistema. Dentre estes, estão as perguntas múltipla escolha necessárias para a realização dos jogos. Para isso, foi criado um arquivo no formato Aiken, reconhecido pelo Moodle, contendo diversas questões que foram importadas para o banco de questões Moodle através da funcionalidade de importação de questões do site Moodle. Este arquivo continua presente na versão final do jogo, localizado na pasta "inc", que se encontra na pasta raiz do plugin, com o nome "bdcapitais.txt", e possui



sessenta e seis questões sobre capitais de países.

Além disso, foi necessário criar diferentes tipos de usuários para testar as permissões existentes no sistema, além de ser necessário testar as interações de diferentes usuários durante os recursos multijogadores do jogo.

Diversos problemas foram encontrados graças aos testes realizados. De todos os erros observados, apenas um não foi corrigido.

O erro descrito pela mensagem de erro "error: could not query the database (db\_get\_course\_name()) Error? You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near " at line 1" acontece, como exposto pela mensagem de erro, no método "db\_get\_course\_name()", do módulo PHP "dbhelper.php". Este método é usado em diversos momentos no sistema, e em apenas um destes momentos este erro ocorre.

Apesar da ocorrência do erro, o método funciona perfeitamente, bem como a consulta SQL a qual o erro se refere. Por este fato, e pelo tempo levado para resolver o erro sem sucesso, o erro passou a ser ignorado.

Para o acompanhamento dos erros que podem ocorrer durante a utilização do sistema pelos usuários, os erros foram direcionados para um arquivo log de erros localizado na pasta "log", na pasta raiz do plugin.

## 7 Considerações finais

O projeto buscou contribuir para o ambiente de aprendizagem Moodle, tornando o estudo dos seus cursos mais atrativo para aqueles que não veem o estudo com afinco. Para isso, este tomou funcionalidades comuns aos jogos de mais sucesso dos dias de hoje, visando aquelas que possuem como fator principal a motivação do jogador, e as aplicou na forma de um jogo de aprendizagem em um plugin que agrupa jogos que se encaixam neste perfil.

O público alvo beneficiado por esse trabalho são os professores e os alunos que utilizam o sistema Moodle, que agora dispõem de uma nova opção de aprendizagem através do plugin e de seu jogo, bem como os desenvolvedores do ambiente Moodle, que podem se aproveitar das facilidades do plugin para desenvolver mais jogos de aprendizagem para os cursos Moodle.

O trabalho aprimorou bastante os conhecimentos do desenvolvedor relacionados à programação web. PHP, Javascript, HTML, CSS são algumas das linguagens em que o presente aluno possuía pouco ou nenhuma experiência. Entender como essas interagem entre si foi um dos conhecimentos adquiridos mais interessantes para o aluno. Além disso, foram aprimorados diversos outros conhecimentos relacionados às metodologias utilizadas para o desenvolvimento. O projeto foi de grande importância também para o reforço da organização do aluno para um desenvolvimento seguro dentro de um prazo determinado. Uma última contribuição a ser detalhada neste documento, é o aprimoramento obtido pelo aluno em relação a redação de relatórios e documentos de um projeto de uma escala maior dos que são vistos durante o curso.

Uma oportunidade clara para trabalhos futuros é o desenvolvimento do sistema para dispositivos móveis, visto que a maioria dos alunos se beneficiariam da adaptação para a esta plataforma, considerando que estes estão sempre conectados à internet através de seus dispositivos. Além disso, diversas melhorias podem ser implementadas, tanto no plugin quanto no jogo. A seguir, algumas sugestões de melhorias ao projeto:

### **Plugin:**

- Suporte para mais tipos de perguntas além de múltipla escolha.
- Adição de um menu de configuração para administradores fornecendo

opções de manipulação dos jogos do plugin.

- Criação de um servidor acessível pelo plugin que agrupe os jogos desenvolvidos para o plugin para facilitar a adição destes.

**Jogo:**

- Melhorias na interface.
- Melhorias nos efeitos sonoros.
- Adição de mais tipos de conquistas.

É sugerida também, neste caso para ambos, a parametrização das strings em arquivos externos para facilitar desde mudanças simples até traduções completas para outros idiomas.

## 8 Referências Bibliográficas

1. Pulino Filho, A.R; **Introdução ao Moodle - Ambiente de Aprendizagem**; Universidade de Brasília; Agosto de 2004.
2. CLASSROOMAID, **Gamifying Learning with Moodle**; <http://classroom-aid.com/2013/11/18/gamifying-learning-with-moodle-gbl/>; Setembro de 2013.
3. Moodle™, **Moodle Statistics**; <https://moodle.net/stats/>; Abril de 2015.
4. Staa, A.v.; *Programação Modular*; Rio de Janeiro; Campus; 2000.
5. Apache Friends, **XAMPP**; <https://www.apachefriends.org>; 2015.
6. Moodle™, **Moodle Documentation**; <https://docs.moodle.org/>; Junho de 2015.
7. DEVMEDIA, **Bibliotecas Javascript**; <http://www.devmedia.com.br/top-5-bibliotecas-javascript-que-voce-precisa-conhecer/27681>; 2015.
8. Google, **Google Play**; <https://play.google.com/store/apps/>; Junho de 2015.
9. Moodle™, **Blocks**; <https://docs.moodle.org/dev/Blocks>; Abril de 2015.
10. Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas, **Agile Manifesto**; <http://agilemanifesto.org/>; 2001.
11. Lionite, **Using PHP with MySQL**; <https://www.binpress.com/tutorial/using-php-with-mysql-the-right-way/17>; Março de 2014.
12. Caron, Stéphane; <http://www.no-margin-for-errors.com/projects/prettyphoto-jquery-lightbox-clone/>; 2015.
13. The JQuery Foundation, **JQuery API**; <https://api.jquery.com/>; 2015.
14. W3Schools, **JQuery** **AJAX** **Methods**; [http://www.w3schools.com/jquery/jquery\\_ref\\_ajax.asp](http://www.w3schools.com/jquery/jquery_ref_ajax.asp); 2015.
15. Moodle™, **Aiken Format**; [https://docs.moodle.org/24/en/Aiken\\_format](https://docs.moodle.org/24/en/Aiken_format); 2015.