



RAUL BRENNER DOS SANTOS COSTA

**EASYQUEST - SISTEMA DE GERENCIAMENTO ONLINE DE
QUESTÕES PARA A PLATAFORMA MOODLE**

LAVRAS – MG

2022

RAUL BRENNER DOS SANTOS COSTA

**EASYQUEST - SISTEMA DE GERENCIAMENTO ONLINE DE QUESTÕES PARA A
PLATAFORMA MOODLE**

Trabalho de conclusão de curso apresentado à
Universidade Federal de Lavras, como parte das
exigências do curso Sistemas de Informação,
para a obtenção do título de Bacharel em
Sistemas de Informação.

Prof. Paulo Afonso Parreira Júnior
Orientador

Alexandre José de Carvalho Silva
Coorientador

LAVRAS – MG

2022

RAUL BRENNER DOS SANTOS COSTA

**EASYQUEST - SISTEMA DE GERENCIAMENTO ONLINE DE QUESTÕES PARA A
PLATAFORMA MOODLE**

Trabalho de conclusão de curso apresentado à
Universidade Federal de Lavras, como parte das
exigências do curso Sistemas de Informação,
para a obtenção do título de Bacharel em
Sistemas de Informação.

APROVADA em 22 de Fevereiro de 2022.

Prof. Paulo Afonso Parreira Júnior UFLA
Alexandre José de Carvalho Silva UFLA
Profa. Renata Teles Moreira UFLA
Prof. Heitor Augustus Xavier Costa UFLA

Prof. Paulo Afonso Parreira Júnior
Orientador

Alexandre José de Carvalho Silva
Co-Orientador

**LAVRAS – MG
2022**

RESUMO

A educação a distância vem evoluindo cada vez mais ao lado dos avanços tecnológicos. Um grande alavancador desta modalidade são os Ambientes Virtuais de Aprendizagem (AVA), por meio dos quais professores planejam e disponibilizam conteúdos aos alunos, dentre outras coisas. O Moodle é um dos AVA mais conhecidos em todo o mundo, sendo utilizado na Universidade Federal de Lavras (UFLA), por meio de sua instância conhecida como *campus virtual*. O Moodle inclui, dentre outras funcionalidades, um módulo para manutenção de questões de provas e questionários, o qual apresenta algumas deficiências em termos de facilidade de uso. Este trabalho tem como objetivo apresentar a aplicação web *EasyQuest*, que possui como intuito tornar o processo de manutenção de questões para o Moodle mais eficiente.

Palavras-chave: Educação a distância. Usabilidade. Ambiente Virtual de Aprendizagem. Moodle.

ABSTRACT

Distance education has been evolving more and more alongside technological advances. A great leverage of this modality are the Virtual Learning Environments (VLE), through which teachers plan and make content available to students, among other things. Moodle is a well-known VLE, being used at the Federal University of Lavras (UFLA), through its instance known as *Campus Virtual*. Moodle includes, among other features, a module for maintenance of test questions and questionnaires, which has some problems in terms of ease of use. This work aims to present the *EasyQuest* web application, which aims to make the Moodle question maintenance process more efficient.

Keywords: Distance education. Usability. Virtual Learning Environments. Moodle.

LISTA DE FIGURAS

Figura 2.1 – Tela que apresenta a listagem de questões cadastradas	9
Figura 2.2 – Tela que solicita ao usuário a escolha do tipo de questão a ser cadastrada . .	9
Figura 2.3 – Exemplo de código no formato Moodle/XML para uma questão de múltipla escolha	10
Figura 2.4 – Exemplo de implementação de um contador em um componente ReactJS .	13
Figura 2.5 – Estado inicial do componente <i>Counter</i>	13
Figura 2.6 – Estado do componente <i>Counter</i> após o primeiro clique	13
Figura 3.1 – Arquitetura da <i>EasyQuest</i>	17
Figura 3.2 – Módulo de exportação da <i>EasyQuest</i>	18
Figura 3.3 – Utilização do conversor na <i>EasyQuest</i>	18
Figura 3.4 – Gerenciamento de questões na aplicação <i>EasyQuest</i>	19
Figura 3.5 – Tela de criação de questão	20
Figura 3.6 – Editor de texto	20
Figura 3.7 – Editor de texto - adicionar tabela	21
Figura 3.8 – Editor de texto - adicionar imagem	21
Figura 3.9 – Editor de texto - adicionar mídia via URL	21
Figura 3.10 – Alertas - Excluir questão	22
Figura 3.11 – Alertas - Lembrete de <i>download</i>	22
Figura 3.12 – Alertas - Texto de ajuda	22
Figura 3.13 – Moodle - <i>Upload</i> de questões	23

SUMÁRIO

1	Introdução	6
2	Referencial Teórico	8
2.1	Manutenção de questões no AVA Moodle e Trabalhos Relacionados	8
2.2	SPA e ReactJS	11
3	EasyQuest	14
3.1	Requisitos	14
3.2	Arquitetura	17
3.3	Funcionalidades	18
3.3.1	Gerenciamento de Questões	19
3.3.2	Criar uma nova questão	19
3.3.3	Realizar o <i>download</i> das questões cadastradas	22
4	Avaliação da Aplicação	24
4.1	Pontos positivos	24
4.2	Dúvidas e dificuldades	24
4.3	Sugestões	25
5	Considerações finais	28
	REFERÊNCIAS	29

1 INTRODUÇÃO

A utilização de Ambientes Virtuais de Aprendizagem (AVA) tem se tornado cada vez mais comum, principalmente em instituições de ensino superior (SILVA, 2014). Durante a pandemia de COVID-19, ficou evidente o potencial que a Educação a Distância tem de viabilizar o acesso ao ensino (BRAZILIENSE, 2020), destacando ainda mais as vantagens dos AVA.

O Moodle¹ é um AVA de código aberto que possui diversas funcionalidades, entre elas, a elaboração de provas e questionários de avaliação. Apesar das facilidades proporcionadas pelo Moodle, o processo de manutenção (cadastramento, alteração, consulta e exclusão) de questões para provas e questionários requer melhorias, por causa da grande quantidade de (i) interações (por exemplo, cliques com o *mouse*) e (re)carregamentos de telas necessários para realizar esse processo de manutenção e (ii) tipos de questões e configurações disponíveis para o usuário. Se por um lado a diversidade de tipos de questões pode ser considerada uma vantagem, pois atende a diversos perfis de usuários; por outro, pode prejudicar o entendimento e a utilização do ambiente Moodle, principalmente por usuários com pouco conhecimento em informática. De acordo com pesquisa realizada pela empresa Google, em 2015, a dificuldade de utilização é uma das principais causas que levam usuários a abandonar aplicações, sendo que uma em cada cinco aplicações sendo esquecidas pelos usuários por esse e outros motivos (GOOGLE, 2015).

Neste contexto, o presente trabalho baseia-se na hipótese de que o processo de manutenção de questões para o ambiente Moodle pode ser aprimorado, reduzindo a quantidade de interações necessárias para tal, bem como a quantidade de opções de configuração disponíveis para o usuário. Assim sendo, desenvolveu-se uma aplicação computacional denominada *EasyQuest*, a qual permite a manutenção e exportação de questões para o AVA Moodle e busca minimizar os problemas relatados anteriormente. Isso é feito por meio do uso de conceitos e tecnologias modernas para desenvolvimento web, tais como o conceito de SPA (*Single Page Application*), bem como pela disponibilização criteriosa dos tipos de questões e configurações mais utilizadas pelos usuários do Campus Virtual (instância Moodle da UFLA).

A partir de uma avaliação realizada com professores, foi possível perceber que a *EasyQuest* trouxe melhorias para o processo de manutenção de questões de provas e questionários, ao simplificar a interação do usuário com o ambiente Moodle. Também foi possível encontrar pontos de melhorias para futuras versões da aplicação.

¹ <https://moodle.org>

Este trabalho encontra-se organizado como segue: no Capítulo 2, são apresentadas as principais tecnologias e conceitos utilizados ao longo deste trabalho; no Capítulo 3, por sua vez, são apresentados os detalhes de implementação e arquitetura do software desenvolvido. Por fim, os Capítulos 4 e 5 apresentam, respectivamente, os resultados obtidos por meio da avaliação da aplicação *EasyQuest* e as considerações finais deste trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo, são apresentados os recursos disponíveis para manutenção de questões no AVA Moodle, bem como as tecnologias utilizadas no desenvolvimento da solução proposta, a saber, a aplicação *EasyQuest*.

2.1 Manutenção de questões no AVA Moodle e Trabalhos Relacionados

Moodle é um dos AVA mais utilizados no mundo todo. Até Janeiro/2022, ele estava presente em 242 países, possuindo mais de 303 milhões de usuários registrados e mais de 39 milhões de cursos (MOODLE, 2022). Dentre as várias funcionalidades do Moodle, ele possui um módulo para manutenção de questões de provas e questionários, o qual permite aos usuários criar, alterar, consultar e excluir questões de diversos tipos.

Como enfatizado na introdução, para realizar o cadastramento de uma questão no Moodle, vários passos são necessários. A título de exemplo, após clicar em “Criar uma nova questão” (Figura 2.1) o usuário deve selecionar o tipo de questão que deseja cadastrar (Figura 2.2). Ao selecionar o tipo de questão desejada, o usuário deve preencher diversos campos para prosseguir, o que pode tornar o processo de manutenção de questões moroso, principalmente quando o banco de questões possui muitas questões.

Figura 2.1 – Tela que apresenta a listagem de questões cadastradas

Banco de questões

Selecione uma categoria:

A categoria padrão para as questões compartilhadas no contexto 'Gerador de questões online'.

Nenhum filtro de tag aplicado

Filtrar por tags ...

Mostrar texto da questão na lista de questões

[Opções de pesquisa](#)

Também mostrar questões de subcategorias

Também exibir questões antigas

[Criar uma nova questão ...](#)

Questão	Ações	Criado por	Última modificação por
Nome da questão / ID number		Nome / Sobrenome / Data	Nome / Sobrenome / Data
<input type="checkbox"/> Questão sobre ES	Editar	[Redacted]	8 January 2021, 16:06 PM
<input type="checkbox"/> Algoritmo para calcular Fibonacci	Editar	[Redacted]	12 July 2021, 10:43 AM
<input type="checkbox"/> Questão associativa	Editar	[Redacted]	12 July 2021, 10:43 AM
<input type="checkbox"/> Questão sobre GP	Editar	[Redacted]	8 January 2021, 16:06 PM

Com as questões selecionadas:

Fonte: do autor, 2022

Figura 2.2 – Tela que solicita ao usuário a escolha do tipo de questão a ser cadastrada

Escolha um tipo de questão para adicionar

Selecione um tipo de pergunta para ver a sua descrição.

QUESTÕES

- Arrastar e soltar sobre o texto
- Arraste e solte na imagem
- Associação
- Cálculo simples
- Associação de resposta curta aleatória
- Dissertação
- Selecionar as palavras que faltam
- Marcadores "arrasta e solta"
- Múltipla escolha
- Múltipla escolha calculada
- Numérico
- resposta curta
- Respostas embutidas (cloze)
- Verdadeiro/Falso
- Fórmulas
- Record audio/video

OUTRO

- Descrição

Fonte: do autor, 2022

Outra maneira de cadastrar questões no Moodle é por meio de seu mecanismo de importação/exportação. Para isso, é preciso que as questões a serem importadas estejam localizadas em um arquivo de texto e sigam um formato específico. Moodle/XML é um dos formatos que pode ser utilizado para a especificação de questões a serem importadas (um exemplo de questão do tipo “múltipla escolha” pode ser visto na Figura 2.3).

Figura 2.3 – Exemplo de código no formato Moodle/XML para uma questão de múltipla escolha

```
<question type="multichoice">
  <name>
    <text>Titulo da questao</text>
  </name>
  <questiontext format="html">
    <text>Enunciado da questao</text>
  </questiontext>
  <generalfeedback format="html">
    <text>Texto de feedback padrao</text>
  </generalfeedback>
  ...
  <answer fraction="{value}" format="html">
    <text>Alternativa 1</text>
    <feedback format="html">
      <text>Feedback da alternativa 1</text>
    </feedback>
  </answer>
  <answer fraction="{value}" format="html">
    <text>Alternativa 2</text>
    <feedback format="html">
      <text>Feedback da alternativa 2</text>
    </feedback>
  </answer>
</question>
```

Fonte: do autor, 2022

Este formato utiliza *tags XML (Extensible Markup Language)* para identificar as propriedades de uma questão. Algumas das principais propriedades, de acordo com formato Moodle/XML, são:

- *question*: tag raiz do documento XML. É utilizada em conjunto com o atributo *type* para identificar o tipo da questão,
- *name*: tag utilizada para informar o nome da questão, o qual aparecerá na interface gráfica da listagem de questões do Moodle (Figura 2.1),
- *questiontext*: tag utilizada para informar o enunciado da questão,
- *generalfeedback*: tag utilizada para informar o *feedback* geral de uma questão. O *feedback* é uma propriedade opcional e serve para indicar ao aluno informações sobre sua resposta, quando o professor achar pertinente fazê-lo,

- *answer*: tag utilizada para informar uma alternativa para determinada questão de múltipla escolha. Pode ser utilizada em conjunto com os atributos *fraction* e *format* para indicar o peso e o formato da resposta, respectivamente,

O mecanismo de importação/exportação de questões do Moodle é muito útil, pois permite o cadastramento de questões em lote, sem exigir do usuário executar as mesmas interações repetidas vezes. Contudo, tal mecanismo tem pouca utilidade para o usuário leigo, uma vez que colocar as questões no formato adequado para importação não é uma tarefa trivial. A importação/exportação de questões é indicada quando o usuário possui um banco de questões cadastrado em um curso do AVA e deseja migrá-lo para outro curso ou para o desenvolvimento de aplicações que visem se integrar ao AVA Moodle, como é o caso da aplicação *EasyQuest*.

É importante destacar que não foi encontrada, no mercado ou na academia, solução similar à proposta deste trabalho para integração com o módulo de questões da plataforma Moodle. O trabalho que mais se aproxima é o de (SOLOVIEV, 2018), contudo os autores fizeram uma análise qualitativa dos diferentes formatos de importação de questões do Moodle. Há ainda alguns *plugins* para importação de questões para o AVA Moodle, disponíveis no portal (MOODLE, 2022). Um desses *plugins* é o Fronter (VÉDRINE, 2022), o qual também utiliza o formato Moodle/XML. Contudo, ele faz a conversão de questões de outro AVA (denominado Fronter) para o Moodle. Outros *plugins* permitem a importação de questões para o Moodle, a partir de arquivos Word, tal como o (CAMPBELL, 2022). Contudo, isso faz com que o usuário tenha que entender e utilizar o *template* criado pelos desenvolvedores, o que pode tornar o processo de cadastramento de questões mais moroso e suscetível a erros. Além disso, o *plugin* é específico de uma ferramenta de escritório, ao contrário da *EasyQuest*, que pode ser executada em qualquer dispositivo que possua um navegador Web.

2.2 SPA e ReactJS

ReactJS¹ é um *framework* JavaScript para desenvolvimento web. No ano de 2020, ReactJS foi o *framework* mais utilizado por novos projetos, quando comparado a outros *frameworks*, tais como, Angular² e Vue³ (JETBRAINS, 2021). ReactJS foi criado por desenvolvedores do Facebook para auxiliar no desenvolvimento de aplicações da empresa, sendo utilizado

¹ <https://reactjs.org/>

² <https://angularjs.org/>

³ <https://vuejs.org/>

no próprio Facebook⁴, Instagram⁵ e Whatsapp⁶. Com ReactJS, que trabalha com o conceito de *Single Page Applications* (SPA), é possível criar aplicações web que possuem apenas um arquivo HTML, geralmente chamado de “index.html”, o qual é responsável por carregar um código JavaScript que irá gerenciar os componentes da aplicação e as interações com o usuário.

ReactJS foi o *framework* selecionado para o desenvolvimento da aplicação *EasyQuest* por causa de alguns benefícios que possui, bem como do conhecimento prévio do autor deste trabalho com relação a essa tecnologia. Com ReactJS, por exemplo, o navegador do usuário precisará fazer o *download* de apenas um *script*, minimizando a necessidade de recarregamentos de páginas, o que pode melhorar a experiência dos usuários com a aplicação, principalmente para usuários com conexões de Internet lentas.

Na Figura 2.4 é apresentado um exemplo de código desenvolvido com ReactJS. Nas versões mais recentes do *framework*, os componentes são funções que retornam código JSX (JavaScript XML), um tipo de código que combina a sintaxe do HTML com o JavaScript. No exemplo apresentado, o estado do componente se resume à constante *counter*, é criada com a função *useState* (fornecida pelo ReactJS) e inicializada com o valor 0 (zero) (linha 4). A função *useState* retorna um *array* de duas posições, sendo a primeira uma referência para o valor atual do estado criado e a segunda uma função que pode ser chamada para alterar o valor do estado. Na ação de clique no botão (linha 8), a função *setCounter* é utilizada para incrementar o valor do contador (Figuras 2.5 e 2.6).

⁴ <https://facebook.com/>

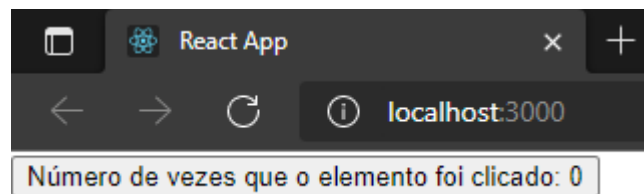
⁵ <https://instagram.com/>

⁶ <https://whatsapp.com/>

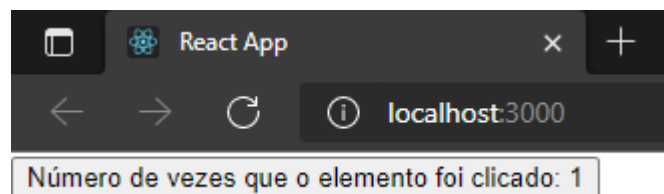
Figura 2.4 – Exemplo de implementação de um contador em um componente ReactJS

```
1 import React , {useState} from 'react'  
2  
3 const Counter = () => {  
4   const [counter, setCounter] = useState(0)  
5   return(  
6     <div>  
7       <button  
8         onClick={() => setCounter(counter+1)}>  
9         Numero de vezes que o elemento foi clicado: {counter}  
10      </button>  
11    </div>  
12  );  
13 }
```

Fonte: do autor, 2022

Figura 2.5 – Estado inicial do componente *Counter*

Fonte: do autor, 2022

Figura 2.6 – Estado do componente *Counter* após o primeiro clique

Fonte: do autor, 2022

3 EASYQUEST

Neste capítulo, são apresentados os requisitos, a arquitetura e a funcionalidade da aplicação computacional *EasyQuest*.

3.1 Requisitos

Como foi discutido na introdução deste trabalho, dois são os fatores que podem impactar na experiência do usuário quanto à manutenção de um banco de questões no Moodle. O primeiro tem a ver com a quantidade de interações e (re)carregamentos de páginas necessários para manter uma questão na aplicação. Isso levou à especificação do requisito não-funcional RNF03, conforme discutido adiante. Requisitos não-funcionais ditam as restrições sob as quais a aplicação deve funcionar (VALENTE, 2020).

O segundo fator tem a ver com a quantidade excessiva de opções oferecidas pela plataforma durante o cadastramento de uma questão. Entender quais são os tipos de questões e de opções de configuração mais utilizadas pelos professores é importante, pois permite a elaboração de um conjunto de requisitos funcionais mais alinhados às necessidades dos usuários. Requisitos funcionais são requisitos que ditam quais funcionalidades uma aplicação deve fornecer (VALENTE, 2020). Para elicitação dos requisitos funcionais da aplicação *EasyQuest*, foi realizada uma pesquisa na instalação do Moodle da Universidade Federal de Lavras (UFLA), conhecida como *campus virtual*. O *campus virtual* possui mais de 17 mil usuários matriculados em 2887 disciplinas, com uma média de 3 mil visitas diárias (VIRTUAL, 2021). Tal pesquisa, realizada com o auxílio do *plugin Question Usage Statistics*², constatou que dos 17 tipos de questões oferecidos pelo Moodle, apenas 3 deles são responsáveis por mais de 80% das questões cadastradas pelos professores, a saber “múltipla escolha”, “verdadeiro ou falso” e “dissertação”. Além disso, os recursos mais utilizados para a confecção dos enunciados das questões são texto formatado (por exemplo, negrito, itálico, alinhamentos) e arquivos de imagem.

Assim sendo, foram identificados os Requisitos Funcionais (RF) e Requisitos Não-Funcionais (RNF) para a aplicação *EasyQuest*:

- **(RF01)** O sistema deve permitir inserir, visualizar, alterar, excluir e duplicar questões;

² https://moodle.org/plugins/report_questionstats/

- **(RF02)** O sistema deve contemplar, no mínimo, três tipos de questões: múltipla escolha, verdadeiro ou falso e dissertação;
- **(RF03)** O texto de enunciado da questão pode conter formatação (negrito, itálico, alinhamento, tamanho e listas de tópicos) e imagens;
- **(RF04)** Toda questão possui título (obrigatório), enunciado (obrigatório) e *feedback* (opcional);
- **(RF05)** Questões de múltipla escolha podem conter até 5 alternativas (2 obrigatórias e as demais opcionais), sendo uma delas a alternativa correta;
- **(RF06)** Questões de verdadeiro ou falso devem conter a resposta correta, isto é, se a afirmação é verdadeira ou falsa;
- **(RF07)** O sistema deve permitir exportar as questões cadastradas para o formato MoodleXML;
- **(RF08)** O sistema deve permitir excluir todas as questões de uma só vez;
- **(RF09)** O sistema deve permitir buscar por uma questão, a partir do seu título;
- **(RNF01)** O sistema deve ser executado no navegador do usuário, sem a necessidade de instalação;
- **(RNF02)** O sistema deve implementar internacionalização para os idiomas Português e Inglês;
- **(RNF03)** O sistema deve permitir que as funções de inserção, visualização, alteração, exclusão e duplicação de questões sejam realizadas com, no máximo, 3 (três) cliques;

O RNF01 diz respeito à disponibilidade da aplicação. Como não depende de um servidor para funcionar, a *EasyQuest* deixa de depender da conexão com a internet do usuário, sendo possível até sua utilização de forma *offline*. Além disso, por ser uma SPA (*Single Page Application*), uma vez que o usuário carrega a aplicação, não serão necessários novos carregamentos

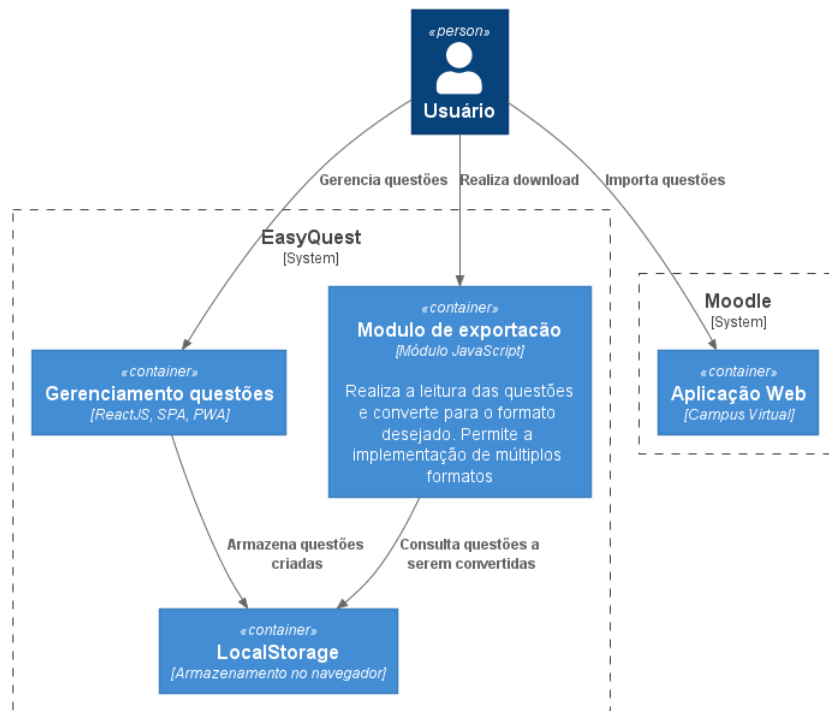
de páginas. O RNF02 visa permitir que a *EasyQuest* alcance mais usuários. Conforme mencionado, o Moodle é um AVA utilizado no mundo todo; dessa forma, ao adicionar suporte ao idioma Inglês, aumenta-se significativamente a quantidade de potenciais usuários da aplicação *EasyQuest*. Quanto ao RNF03, ele diz respeito ao real objetivo da *EasyQuest*: melhorar a experiência do usuário ao diminuir quantidade de interações necessárias para realizar seu trabalho.

Na versão atual da aplicação, todos os requisitos foram contemplados. Além disso, novos requisitos foram adicionados a partir de uma avaliação prévia, realizada com professores, conforme pode ser visto no Capítulo 4.

3.2 Arquitetura

A arquitetura elaborada para a aplicação *EasyQuest* é apresentada nesta seção. Como todos os dados das questões são armazenados no navegador do usuário, não houve a necessidade de implementação uma aplicação servidora (*back-end*). A estrutura da aplicação e a comunicação entre os módulos pode ser visualizada no diagrama da Figura 3.1.

Figura 3.1 – Arquitetura da *EasyQuest*



Fonte: do autor, 2022

É por meio do módulo Gerenciamento de questões que o usuário interage com o sistema. Esse módulo oferece as funcionalidades de inserção, visualização, edição, exclusão e download de questões, entre outras coisas. O módulo de exportação traduz um conjunto de questões criado pelo usuário para o formato MoodleXML (Figura 2.3). Ele foi desenvolvido para permitir que novos formatos de exportação sejam inseridos no futuro.

O módulo de exportação é apenas um arquivo que agrupa diferentes ferramentas de conversões. Como pode ser visto na Figura 3.2, o índice do módulo de exportação é responsável por agrupar e exportar os diferentes conversores. Desta forma, para criar um novo conversor, deve-se adicionar a importação e incluir o conversor no objeto de exportação.

Figura 3.2 – Módulo de exportação da *EasyQuest*

```
import { SingleMoodleXml, MultipleMoodleXml } from './MoodleXml';

export default {
  MoodleXml: {
    converter: SingleMoodleXml,
    multipleConverter: MultipleMoodleXml,
  },
};
```

Fonte: do autor, 2022

A utilização do conversor também é simples. Como pode ser visto na Figura 3.3, o conversor é importado na linha 1 e é utilizado na função *downloadQuestions* (linha 4). A função *downloadQuestions* recebe uma chave utilizada para recuperar questões no *LocalStorage* (linha 5), caso existam questões armazenadas, as questões são convertidas para o formato JSON e, em seguida, para o formato MoodleXML (linhas 5 a 9).

Figura 3.3 – Utilização do conversor na *EasyQuest*

```
1 import Converters from "../../shared/utils/Converters";
2 import { DownloadXmlFile } from "../../shared/utils/Utils";
3
4 const downloadQuestions = (questionsKey) => {
5   const localStorageContent = localStorage.getItem(questionsKey);
6   if (localStorageContent) {
7     const questions = JSON.parse(localStorageContent);
8     const moodleXmlQuestions =
9     Converters.MoodleXml.multipleConverter(questions);
10    DownloadXmlFile(moodleXmlQuestions, "questions.xml");
11  }
12};
```

Fonte: do autor, 2022

O *LocalStorage* é um módulo do navegador, o qual armazena informações diversas e é usado na *EasyQuest* para manter as questões criadas pelo usuário.

3.3 Funcionalidades

Nesta seção, são apresentadas as funções desenvolvidas para a aplicação *EasyQuest*¹, visando atender aos requisitos apresentados na Seção 3.1.

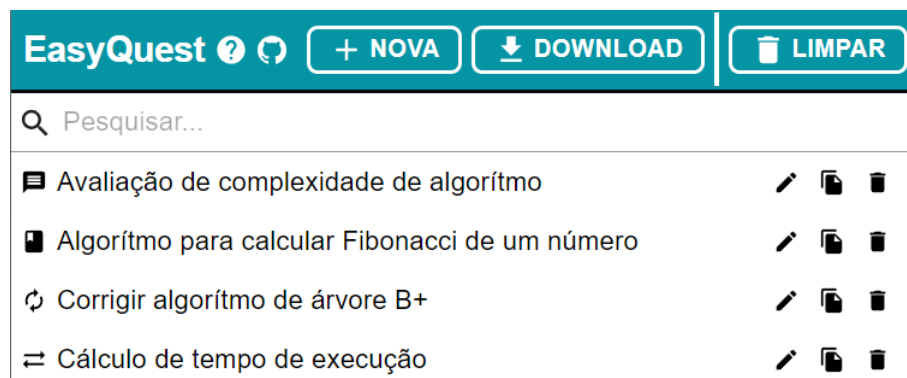
¹ *EasyQuest*. Disponível em: <https://easy-quest.web.app/>

3.3.1 Gerenciamento de Questões

Na tela inicial da *EasyQuest* (Figura 3.4), o usuário pode realizar as seguintes ações:

- Visualizar a lista de questões cadastradas;
- Visualizar questões armazenadas;
- Buscar questões pelo título;
- Criar uma nova questão;
- Realizar o *download* das questões cadastradas;
- Limpar a lista de questões;
- Editar, duplicar ou excluir uma questão cadastrada.

Figura 3.4 – Gerenciamento de questões na aplicação *EasyQuest*



Fonte: do autor, 2022

A descrição das principais ações são apresentadas nas próximas seções deste trabalho.

3.3.2 Criar uma nova questão

Ao clicar no botão “Nova” (Figura 3.4), o usuário é encaminhado à página de criação de questões (Figura 3.5). Nessa tela, o usuário pode escolher qual tipo de questão deseja criar. Por padrão, a *EasyQuest* seleciona o tipo “Múltipla Escolha”, por ser o tipo de questão mais utilizado, de acordo com a pesquisa realizada no *campus virtual*. Atualmente, a aplicação

EasyQuest permite ao usuário criar questões dos tipos dissertativa, múltipla escolha, verdadeiro/falso e associativa. O tipo “associativo” é o quarto mais utilizado no *campus virtual* e foi, posteriormente, adicionado à *EasyQuest*, após a avaliação da aplicação com usuários, conforme apresentado no Capítulo 4.

Figura 3.5 – Tela de criação de questão

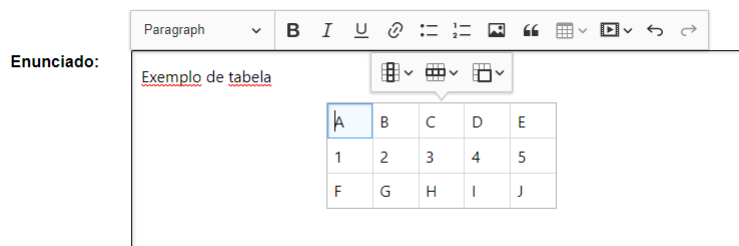
Fonte: do autor, 2022

Durante a criação de questões, *EasyQuest* fornece aos usuários algumas ferramentas para edição de texto nos campos de enunciado e de alternativas (Figura 3.6). Essas ferramentas cobrem as funcionalidades mais utilizadas pelos usuários do *campus virtual*, tais como formatação de texto, inserção de tabelas (Figura 3.7), inserção de imagens (Figura 3.8) e vídeos do YouTube (Figura 3.9).

Figura 3.6 – Editor de texto

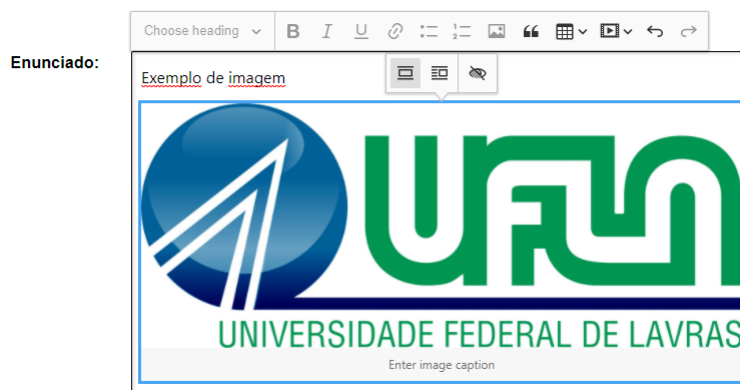
Fonte: do autor, 2022

Figura 3.7 – Editor de texto - adicionar tabela



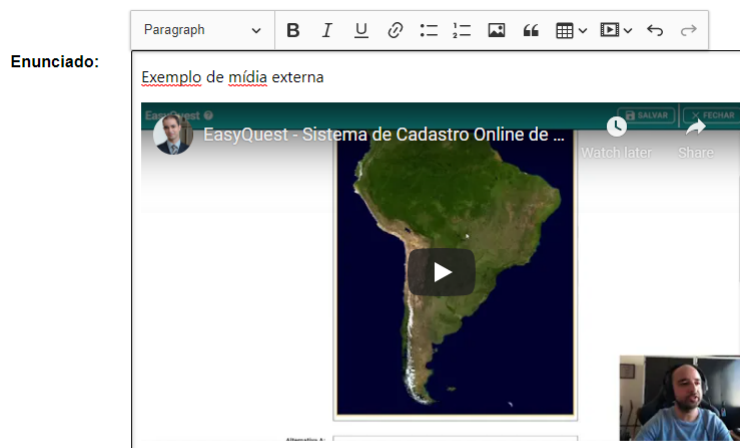
Fonte: do autor, 2022

Figura 3.8 – Editor de texto - adicionar imagem



Fonte: do autor, 2022

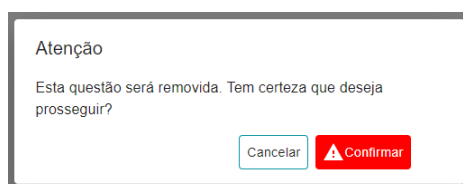
Figura 3.9 – Editor de texto - adicionar mídia via URL



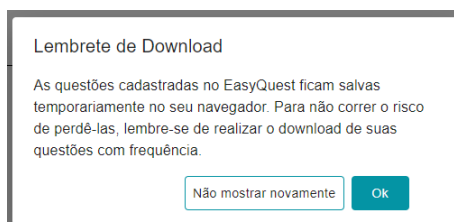
Fonte: do autor, 2022

Para prevenir erros por parte do usuário, *EasyQuest* emite alguns alertas em momentos de decisões e algumas mensagens informativas durante a utilização da aplicação (Figuras 3.10, 3.11 e 3.12).

Figura 3.10 – Alertas - Excluir questão

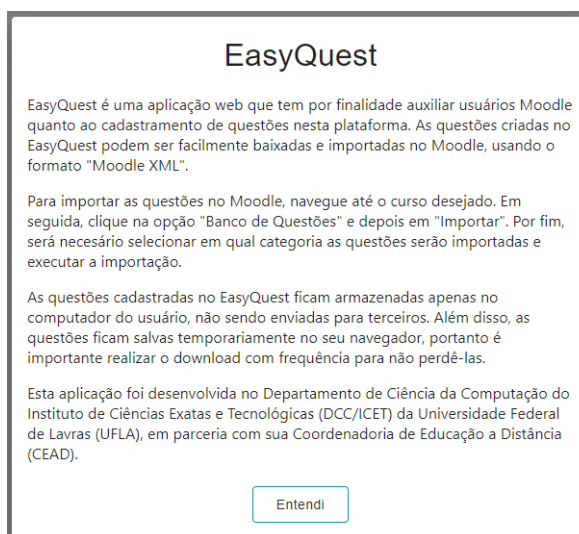


Fonte: do autor, 2022

Figura 3.11 – Alertas - Lembrete de *download*

Fonte: do autor, 2022

Figura 3.12 – Alertas - Texto de ajuda



Fonte: do autor, 2022

3.3.3 Realizar o *download* das questões cadastradas

Como mencionado anteriormente, a aplicação *EasyQuest* foi pensada para permitir que as questões criadas possam ser exportadas em vários formatos. No momento da escrita deste trabalho (Jan/2022), o formato utilizado foi o MoodleXML, um dos formatos adotados pela plataforma Moodle para realizar importação de questões de outras plataformas.

Após realizar o *download* do arquivo XML na *EasyQuest*, para realizar a importação, o usuário deve autenticar-se na sua instância Moodle, acessar a área de criação de questões

da disciplina desejada e clicar na opção de importação. O usuário deve escolher o formato MoodleXML e realizar o *upload* do arquivo gerado pela *EasyQuest* (Figura 3.13).

Figura 3.13 – Moodle - *Upload* de questões



Fonte: do autor, 2022

4 AVALIAÇÃO DA APLICAÇÃO

A avaliação da *EasyQuest* foi realizada com o auxílio de 7 (sete) professores do Departamento de Ciência da Computação da UFLA, no final da fase de desenvolvimento. Todos os professores retornaram *feedbacks* com elogios à aplicação e, em alguns casos, fizeram sugestões sobre as funcionalidades. Algumas dessas sugestões foram, posteriormente, implementadas na *EasyQuest*.

Para a realização da avaliação, foi enviado um e-mail convidando os professores a participarem do processo de avaliação da *EasyQuest*, no qual foi explicada a motivação para o desenvolvimento da aplicação e foram dadas instruções sobre como retornar os *feedbacks*.

Os dados coletados das avaliações foram divididos em três categorias: (i) Pontos positivos; (ii) Dúvidas e dificuldades; e (iii) Sugestões.

4.1 Pontos positivos

De modo geral, os elogios recebidos durante a avaliação condizem com os objetivos da aplicação. Os principais elogios apontados foram:

- Aplicação prática: os professores comentaram que acharam o processo de criar questões muito mais rápido na *EasyQuest* em comparação ao Moodle;
- Fácil utilização: os professores comentaram sobre não terem enfrentado dificuldades para realizar as operações dentro da aplicação;
- Interface amigável: os professores elogiaram o fato de a interface ser simples e objetiva, sem pontos que possam gerar dúvidas durante a utilização;
- Funcionamento correto das questões importadas no Moodle: quando importadas, todas as informações preenchidas na *EasyQuest* continuaram presentes e não foram identificados *bugs* na importação das questões para o ambiente Moodle.

4.2 Dúvidas e dificuldades

As respostas que fazem parte dessa categoria estão mais relacionadas à disponibilidade de informações sobre o funcionamento da aplicação (principalmente sobre o armazenamento

das questões). Outro ponto interessante nessa categoria é algumas das dúvidas e reclamações estejam relacionadas ao funcionamento do Moodle, não da *EasyQuest*.

- Onde e por quanto tempo as questões ficam armazenadas? As dúvidas relacionadas ao armazenamento das questões surgiram quando os professores fechavam a aplicação e abriam novamente, mas as questões ainda estavam armazenadas;
- Os ícones dos tipos de questões não fazem sentido: alguns professores mencionaram que os ícones escolhidos para representar os tipos de questões não eram intuitivos;
- Dificuldade ao utilizar a ferramenta de importação (Moodle): alguns professores mencionaram dificuldade para localizar onde realizar a *importação* das questões no Moodle.

Como mencionado, algumas das reclamações foram relacionadas à utilização do Moodle. Alguns professores chegaram a comentar que não estavam conseguindo navegar na interface de questões da plataforma Moodle. Esses professores também relataram que estavam utilizando outras ferramentas como Google Forms para realizar avaliações e que optaram por essas ferramentas pela difícil utilização do Moodle.

Na versão da *EasyQuest* avaliada pelos professores, não havia uma tela de ajuda com a explicação sobre como funciona o armazenamento das questões na aplicação. Outro ponto que contribuiu para o surgimento dessas dúvidas é o fato de a aplicação utilizar o *LocalStorage* como recurso de armazenamento. É um recurso relativamente simples, que permite armazenar dados localmente. No entanto, não garante a disponibilidade dos dados por períodos prolongados.

Os pontos que geraram reclamações relacionados à *EasyQuest* foram corrigidos na versão atual da aplicação. Os ícones de questões foram alterados, foi criada uma tela de ajuda (Figura 3.12) com explicações sobre o funcionamento da aplicação e foram adicionados alertas lembrando o usuário de realizar *download* constantemente de suas questões (Figura 3.11).

4.3 Sugestões

Alguns professores também fizeram sugestões de funcionalidades que gostariam de ver na *EasyQuest*:

- Implementar interface para dispositivos móveis: como a *EasyQuest* foi desenvolvida com o foco na utilização no *desktop*, foi feita uma sugestão para adicionar suporte dispositivos com telas menores;

- Adicionar categorização de questões: um dos comentários na avaliação sugeriu a criação de uma funcionalidade para criar categorias para questões por assunto ou disciplina por exemplo;
- Adicionar opção de anexo de arquivo: um dos professores comentou que tem o costume de anexar arquivos de código-fonte para auxiliar os alunos, por isso essa funcionalidade seria importante;
- Adicionar tela de ajuda: essa sugestão foi relacionada às dúvidas que surgiram durante o teste da aplicação, principalmente as dúvidas relacionadas ao armazenamento das questões;
- Adicionar questões associativas: na versão de testes, a opção de criar questões associativas ainda não havia sido desenvolvida, mas um dos professores comentou que costuma utilizar esse tipo de questão;
- Adicionar *feedbacks* por alternativas: na versão atual da aplicação existe apenas a funcionalidade de *feedback* geral da questão, sem a possibilidade de adicionar *feedbacks* para cada alternativa.

Das sugestões fornecidas pelos professores, foram desenvolvidas as funcionalidades de tela de ajuda e de questões associativas.

A *EasyQuest* tem como objetivo ser uma aplicação responsiva, no entanto, como o processo de criação de questões envolve a digitação de várias informações, assumiu-se que os usuários irão preferir utilizar um ambiente *desktop*. Por esse motivo, esse tipo de ambiente foi escolhido para desenvolvimento da *EasyQuest*. O suporte a dispositivos com telas menores não foi abandonado, foi apenas despriorizado e poderá ser desenvolvido em versões futuras da aplicação.

As outras sugestões foram adicionadas a um *backlog* de funcionalidades futuras e serão desenvolvidas de acordo com a ordem de prioridade. É importante destacar que para casos em que os usuários realmente precisam dessas funcionalidades, elas podem ser acessadas após a realização do *upload* das questões no Moodle, por exemplo, o *feedback* por alternativas, não disponíveis na *EasyQuest*, mas podem ser adicionados após a importação das questões no Moodle.

É válido ressaltar que essas avaliações foram realizadas com um pequeno grupo de professores do Departamento de Ciência da Computação da UFLA. Dessa forma, são professores que têm tendência a terem mais facilidade ao utilizar ferramentas computacionais. Além disso a avaliação foi realizada de maneira informal, fora de um ambiente de testes controlado e sem usar uma metodologia de avaliação específica.

No entanto, existe o planejamento para realizar uma avaliação mais adequada contemplando professores de outros departamentos.

5 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento da *EasyQuest*, foi possível perceber o tamanho do desafio que é desenvolver software para usuários reais, tentando oferecer as funcionalidades desejadas ao mesmo tempo em que se mantém a utilização do software simples e intuitiva.

Quantidade de funcionalidades e simplicidade da aplicação são medidas que geralmente se opõem. Quanto mais funcionalidades forem adicionadas a uma aplicação, mais confusa e poluída a interface pode ficar.

Um dos principais objetivos ao desenvolver a *EasyQuest* era trazer uma solução computacional que fosse uma alternativa *lightweight* para o módulo de manutenção de questões do Moodle. Apesar de ter sido avaliada apenas com professores da Universidade Federal de Lavras (UFLA), a *EasyQuest* está disponível de forma global e pode ser utilizada com qualquer instância Moodle ao redor do mundo. É importante salientar que a *EasyQuest* oferece suporte para os idiomas Português (BR) e Inglês.

Apesar de os resultados da avaliação com a *EasyQuest* terem sido satisfatórios, ainda existem melhorias a serem feitas. Algumas delas são que a *EasyQuest* ofereça suporte a outros idiomas, a outros tipos de questões e formatos de exportação do Moodle. Outro ponto de melhoria para a *EasyQuest* é oferecer novas funcionalidades com o editor dos enunciados das questões, permitindo, por exemplo, a alteração na cor do texto. É importante salientar que a *EasyQuest* é um software de código livre¹, podendo ser mantido e evoluído pela comunidade de acordo com suas necessidades.

Também cabe ressaltar a necessidade de realizar outras formas de avaliação da ferramenta, com um público maior que possa apontar falhas ainda não encontradas na aplicação.

¹ Repositório da *EasyQuest*. Disponível em: <https://github.com/pauloafpjunior/easyquest>

REFERÊNCIAS

BRAZILIENSE, C. **Universidades de todo o mundo adotam aulas on-line durante a pandemia.** 2020. <https://www.correiobraziliense.com.br/app/noticia/eu-estudante/ensino_ensinosuperior/2020/03/18/interna-ensinosuperior-2019,835070/universidades-de-todo-o-mundo-adotam-aulas-on-line-durante-a-pandemia.shtml>.

CAMPBELL, E. **Microsoft Word File Import/Export.** 2022. <https://moodle.org/plugins/qformat_wordtable>.

GOOGLE. **App-Nesia in the UK: The Need for Re-Engagement Marketing.** 2015. <<https://www.thinkwithgoogle.com/intl/en-gb/consumer-insights/consumer-trends/app-nesia-in-the-uk-the-need-for-re-engagement-marketing/>>.

JETBRAINS. **JavaScript Programming - The State of Developer Ecosystem in 2020 Infographic | JetBrains: Developer Tools for Professionals and Teams.** 2021. <<https://www.jetbrains.com/lp/devecosystem-2020/javascript/>>.

MOODLE. **Moodle statistics.** 2022. <<https://stats.moodle.org/>>.

SILVA, A. J. de C. **Desenvolvimento de uma comunidade virtual para a inserção da metodologia blended learning na Educação Básica.** [S.l.]: Universidade Federal de Lavras, Lavras, 2014.

SOLOVIEV, I. S. M. S. V. S. T. A. V. M. M. M. V. N. **Import test questions into Moodle LMS.** [S.l.]: Workshop on Cloud Technologies in Education (CTE 2018), 2018. <<https://arxiv.org/abs/2010.15577/>>.

VALENTE, M. T. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade.** [S.l.]: Independente, 2020.

VIRTUAL, C. **Campus Virtual UFLA.** 2021. <<https://campusvirtual.ufla.br/>>.

VÉDRINE, J.-M. **Frontier.** 2022. <https://moodle.org/plugins/qformat_fronter>.