



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE  
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**EDSON WESLEY ALMEIDA DO NASCIMENTO**

**SCOMATH:  
DESENVOLVIMENTO DE UMA FERRAMENTA PARA O  
MOODLE NO DOMÍNIO DE QUESTÕES DE MATEMÁTICA.**

**CAMPINA GRANDE - PB**

**2023**

**EDSON WESLEY ALMEIDA DO NASCIMENTO**

**SCOMATH:**

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA O  
MOODLE NO DOMÍNIO DE QUESTÕES DE MATEMÁTICA.**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**Orientador: Professor Dr. Maxwell Guimarães de Oliveira.**

**CAMPINA GRANDE - PB**

**2023**

**EDSON WESLEY ALMEIDA DO NASCIMENTO**

**SCOMATH:**

**DESENVOLVIMENTO DE UMA FERRAMENTA PARA O  
MOODLE NO DOMÍNIO DE QUESTÕES DE MATEMÁTICA.**

**Trabalho de Conclusão Curso  
apresentado ao Curso Bacharelado em  
Ciência da Computação do Centro de  
Engenharia Elétrica e Informática da  
Universidade Federal de Campina  
Grande, como requisito parcial para  
obtenção do título de Bacharel em  
Ciência da Computação.**

**BANCA EXAMINADORA:**

**Professor Dr. Maxwell Guimarães de Oliveira  
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Marcus Salerno de Aquino  
Examinador – UASC/CEEI/UFCG**

**Professor Tiago Lima Massoni  
Professor da Disciplina TCC – UASC/CEEI/UFCG**

**Trabalho aprovado em: 14 de Fevereiro de 2023.  
CAMPINA GRANDE - PB**

## **RESUMO (ABSTRACT)**

This work reports the development of an application and the creation of a SCORM-based package to generate math questions and automatically correct steps. The application was successfully developed and met the initial expectations. It was successfully tested in Moodle. During development, difficulties were encountered such as a lack of updated documentation and complexity of the application flow. The initial idea of developing a plugin for Moodle was not viable due to time constraints and difficulties encountered, so the use of SCORM was opted for instead.

# SCOMath: Desenvolvimento de uma ferramenta para o Moodle no domínio de questões de matemática.

Edson Wesley Almeida do Nascimento  
edson.wesley.nascimento@ccc.ufcg.edu.br  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba

Maxwell Guimarães de Oliveira  
maxwell@computacao.ufcg.edu.br  
Universidade Federal de Campina Grande  
Campina Grande, Paraíba

## RESUMO

Este trabalho relata o desenvolvimento de uma aplicação e a criação de um pacote SCORM base para gerar questões matemáticas e corrigir os passos de maneira automática. A aplicação foi desenvolvida com sucesso e atingiu as expectativas iniciais. Foi testada com sucesso no Moodle. Durante o desenvolvimento, foram encontradas dificuldades como falta de documentação atualizada e complexidade do fluxo da aplicação. A ideia inicial de desenvolver um plugin para o Moodle não foi viável devido ao tempo e dificuldades encontradas, assim, optou-se pelo uso do SCORM.

## Palavras-chaves

Moodle, SCORM, ADL, plugin, questões matemáticas, gerador de questões.

## 1. INTRODUÇÃO

O Moodle já era adotado em algumas Instituições de Ensino, porém durante a pandemia de Covid-19, medidas de isolamento foram aplicadas em todo território nacional, e com isso, algumas Instituições de Ensino Superior de modalidade presencial acabaram adotando-o para suprir a necessidade de um ambiente virtual. A Universidade Federal de Campina Grande (UFCG) foi uma das Instituições que acabou aderindo o uso de tal plataforma.

O Moodle já possuía toda uma gama de ferramentas [1] para facilitar tanto a vida dos docentes quanto dos discentes. No curso de Ciência da Computação, a plataforma foi adotada por algumas disciplinas, por exemplo: Fundamentos da Matemática para Ciência da Computação I e Introdução à Computação, onde utilizaram alguns plugins ou deixaram de realizar algumas atividades na plataforma devido à ausência de plugins mais assertivos, em que os professores tivessem uma maior flexibilidade para gerar atividades e conteúdos ou devido aos mais funcionais serem pagos, principalmente os matemáticos, o que acabou inviabilizando o uso.

Com este problema, foi identificada a necessidade de desenvolver plugins que permitam a criação de questões matemáticas, avaliando não apenas o resultado final, mas também os passos para chegar a ele, de maneira automática/semi-automática. O docente elaborará uma questão, definirá a quantidade de passos e o resultado de cada passo, gerando um pacote que pode ser utilizado em qualquer *Learning Management System* (LMS). O aluno, então, poderá interagir com o pacote disponível no Moodle, realizando o questionário e obtendo suas notas e *feedback* dos passos corretos e incorretos

(se houver erros, o passo preenchido incorretamente terá a resposta esperada exibida).

Com o desenvolvimento deste trabalho, os seguintes artefatos foram produzidos e disponibilizados:

- O produto desenvolvido está disponível no seguinte repositório do github: <https://github.com/edsonwesley/SCOMath>
- Um breve tutorial da aplicação pode ser assistido em: <https://youtu.be/56XMm8YLci0>
- Um breve tutorial do uso do pacote no Moodle pode ser assistido em: [https://youtu.be/IK1Af\\_9\\_tZc](https://youtu.be/IK1Af_9_tZc)

A seguir, na seção 2, abordaremos os itens que serviram de base para a fundamentação teórica. Na seção 3, apresentaremos uma análise entre os plugins matemáticos mais baixados. Na seção 4, teremos a solução desenvolvida, mostrando como a aplicação e o pacote funcionam e as interações dos discentes e docentes. Adiante, na seção 5, trataremos as conclusões e trabalhos futuros, falando sobre os resultados e possíveis melhorias na aplicação. Já na seção 6, mostraremos as dificuldades e lições aprendidas no decorrer da pesquisa e desenvolvimento do trabalho. Por fim, na seção 7, encerraremos com agradecimentos a todos os envolvidos nesta pesquisa.

## 2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão descritos os conceitos e definições que serão base para o desenvolvimento da aplicação.

### 2.1 Moodle

O Moodle é uma plataforma LMS *open-source* amplamente utilizada por docentes e discentes para suprir um ambiente de aprendizagem virtual. Possui diversas funcionalidades que o professor pode utilizar para desenvolver suas disciplinas, como criação de questionários, atividades e materiais complementares (como slides e PDFs). Além disso, oferece um ambiente funcional, com mais de 316 milhões de usuários e mais de 179 mil sites Moodle em todo o mundo<sup>1</sup>.

### 2.2 Plugins do Moodle

No LMS Moodle, os plugins são ferramentas desenvolvidas para realizar tarefas específicas, como criar questões ou mudar o tema do ambiente<sup>2</sup>. No contexto da criação de questões, os *plugins* são conhecidos como *Question Type* e adicionam uma

<sup>1</sup> <https://moodle.com/pt/sobre/>

<sup>2</sup> <https://moodle.org/plugins/>

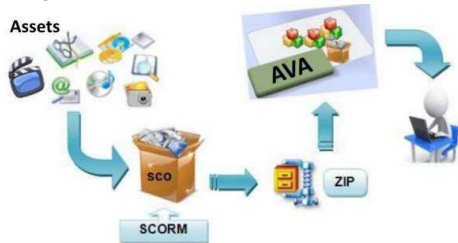
interface ao ambiente Moodle para que o professor possa criar questões usando essa nova funcionalidade.

Um exemplo de *plugin* adotado na disciplina de Introdução à Programação do curso de Engenharia Elétrica é o *Virtual Programming Lab (VPL)*<sup>3</sup>, que define um ambiente virtual de desenvolvimento para a linguagem de programação especificada pelo professor. Os alunos podem interagir com o VPL realizando atividades de programação e usufruir de ferramentas de testes automatizados, facilitando o trabalho do professor e fornecendo um *feedback* mais rápido aos alunos.

### 2.3 Scorm

O SCORM foi criado pela *Advanced Distributed Learning (ADL)*, um programa dos EUA responsável pelo desenvolvimento e aprendizagem distribuída, no início dos anos 2000. A finalidade do SCORM era resolver problemas de durabilidade, reutilização e interoperabilidade decorrentes da grande quantidade de fornecedores de cursos, que exigiam a constante adaptação de sistemas ao conteúdo, o que poderia resultar em problemas de manutenção, como a necessidade de refatoração ou até mesmo de recomeçar tudo do zero quando novos conteúdos eram adicionados ou os existentes alterados [3]. A ADL implementou uma padronização dos conteúdos, criando um sistema de "*plug-and-play*", onde os pacotes são padronizados e compatíveis com vários sistemas, solucionando os problemas mencionados. A Figura 1, demonstra o necessário para a criação de um pacote SCORM até a sua utilização por um aluno.

Figura 1 - Ciclo base de um Pacote SCORM



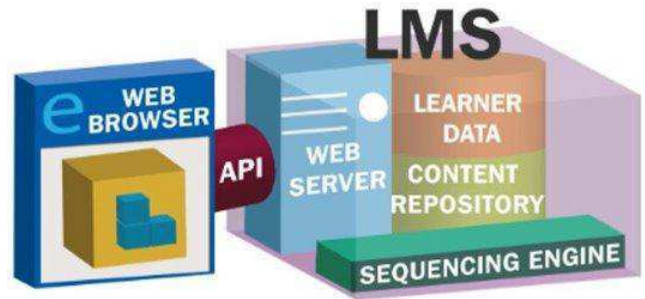
Fonte: Google [6]

O SCORM funciona de forma que o pacote e o LMS não se comunicam diretamente, necessitando de um intermediário para tal, a SCORM API. A SCORM API é responsável por estabelecer a comunicação entre o pacote e o LMS, permitindo que o LMS acesse e manipule informações contidas no pacote, como o progresso do aluno, notas e outras informações relacionadas ao conteúdo. Dessa forma, o pacote pode ser facilmente adicionado e executado em qualquer LMS compatível com a SCORM API, garantindo a interoperabilidade e a reutilização do conteúdo. A Figura 2 apresenta um exemplo, de como um pacote SCORM (bloco amarelo) não se comunica diretamente com um LMS (bloco vermelho) necessitando de uma API (SCORM API) para realizar a comunicação.

Os pacotes SCORM são arquivos *zip* que contém todos os elementos necessários para a exibição de conteúdo online, como *html*, imagens, arquivos de áudio e vídeo, JavaScript e arquivos *xml* que descrevem a estrutura de conteúdo. É guiado através do

arquivo: "*imsmanifest.xml*", onde ele é responsável por definir as *Organizations*, *Resources*, *Assets* e *SCO*s.

Figura 2 - Pacote SCORM, API e LMS



Fonte: ADL [5]

Figura 3 - Template ismanifest.xml

```
<manifest>
<metadata> ... </metadata>
<organizations>
  <organization>
    <item identifier="AGG1">
      <item identifier="SCO1" identifierref="RESOURCE1">...</item>
      <item identifier="SCO2" identifierref="RESOURCE2">...</item>
    </item>
    <item identifier="SCO3" identifierref="RESOURCE3">...</item>
    <item identifier="SCO4" identifierref="RESOURCE4">...</item>
  </organization>
</organizations>
<resources>
  <resource identifier="RESOURCE1">...</resource>
  <resource identifier="RESOURCE2">...</resource>
  <resource identifier="RESOURCE3">...</resource>
  <resource identifier="RESOURCE4">...</resource>
</resources>
</manifest>
```

Fonte: ADL [5]

Como apresentado na Figura 3, o pacote possui tags que gerenciam e guiam o comportamento do mesmo. Para um melhor entendimento, é necessário definir o conceito das tags mais importantes do *ismanifest.xml*.

- *Assets* e *Sharable Content Object (SCO)* [3]: São referências para o agrupamento de recursos, como páginas, fotos, vídeos, dentre outros recursos multimídias. A diferença entre *Asset* e *SCO*, é que o *Asset* é apenas uma unidade simples de conteúdo, enquanto um *SCO* pode ser composto por um ou mais *Assets* e é ele quem consegue comunicar com a *API (Application Programming Interface)* do SCORM.
- *Resources*: Agrupamento de *Assets* e *SCO*s.
- *Organizations*: É um agrupamento de recursos, gerando uma árvore hierárquica, geralmente cada pacote apresenta apenas uma, porém nada impede mais de uma.

<sup>3</sup> [https://moodle.org/plugins/mod\\_vpl](https://moodle.org/plugins/mod_vpl)

O SCORM (versão 2004) é considerado um sistema legado, ou seja, não recebe mais atualizações e precisa ser tratado com cuidado ao ser utilizado. Ele possui algumas limitações, como o uso de JavaScript (JS) puro devido às restrições impostas pela ADL. Isso pode causar limitações na criação e manipulação do pacote. Além disso, é preciso seguir o padrão imposto pela ADL, o que pode restringir a flexibilidade e a criação de conteúdo.

### 3. TRABALHOS RELACIONADOS

Para encontrar os *plugins* adequados para a criação de questões matemáticas, foi realizada uma pesquisa na seção de *plugins* na loja de *plugins* do Moodle, adicionando um filtro de *Question Type* e buscando pelo termo "math" [1]. Como resultado, foram encontrados 8 *plugins*, mas após uma análise detalhada, somente 4 deles foram considerados adequados para serem utilizados na versão 3.8.9 do Moodle, onde os outros 4 foram descartados por um dos seguintes motivos: funcionarem apenas na versão 1.9 do Moodle, não funcionarem ou apresentarem falhas na hora da instalação.

A investigação considerou critérios como funcionamento, limitações, praticidade, facilidade de uso, qualidade da documentação e questões relacionadas à manutenção e evolução dos *plugins*. Os resultados foram apresentados na Tabela 1, que inclui informações como nome, descrição, quantidade de *downloads*, data da última atualização, link da página do *plugin*, versões do Moodle para as quais o *plugin* foi desenvolvido, e pontos positivos e negativos de cada *plugin*.

Após selecionar os *plugins*, foi decidido instalar e testá-los localmente em um ambiente Moodle 3.8, para assim, testar tanto sua compatibilidade, quanto praticidade e facilidade de uso, buscando entender as deficiências e limitações de cada *plugin*. Para a instalação dos *plugins* e criação de questões de teste para avaliação dos mesmos, utilizou-se como base a documentação disponibilizada nas páginas dos mesmos.

**Tabela 1** - Análise dos *plugins* matemáticos

Nome	Descrição	Análise
ShortMath <sup>4</sup>	Implementa botões que inserem código latex ao serem pressionados, onde o aluno esquematiza sua resposta, posteriormente é feito um <i>match</i> entre resposta do aluno e resposta da questão.	<p><b>Pontos positivos:</b> Possui uma interface amigável, apresentando botões que já inserem o código latex para o aluno ao realizar tentativa da questão.</p> <p><b>Pontos Negativos:</b> Possui uma complexidade para criação da questão, assim como, quando testado o <i>plugin</i> falhou em fazer o <i>match</i> entre resposta da questão e resposta oferecida pelo aluno.</p>

<sup>4</sup> [https://moodle.org/plugins/qtype\\_shortmath](https://moodle.org/plugins/qtype_shortmath)

		<p><b>Informações adicionais:</b> Documentação completa. Sua última atualização foi há 9 meses. Possui 110 <i>downloads</i>.</p>
CodeRunner <sup>5</sup>	Implementa uma caixa para o desenvolvimento de códigos e validações dos mesmos, rodando testes automáticos (desenvolvidos pelo criador da questão).	<p><b>Pontos Positivos:</b> Possui uma interface amigável e um ambiente automatizado para questões de programação. O <i>Feedback</i> da questão é bastante claro.</p> <p><b>Pontos Negativos:</b> Apesar de atender os requisitos quando o assunto é programação, o CodeRunner não atende muito bem questões ligadas diretamente a matemática, dado que seu espaço para desenvolvido foca apenas em codificação.</p> <p><b>Informações adicionais:</b> Documentação completa. Sua última atualização foi há 8 semanas. Possui mais de 1000 <i>downloads</i>.</p>
Algebra <sup>6</sup>	Implementa um tipo de pergunta algébrica (Exemplo: "Qual o valor de $x + x^2$ ?) em que as respostas dos alunos são tratadas como expressões algébricas em formato latex e comparadas às respostas fornecidas pelo usuário usando as regras da álgebra.	<p><b>Pontos Positivos:</b> O aluno utiliza expressões formatadas em latex para responder às questões.</p> <p><b>Pontos Negativos:</b> O funcionamento da versão 3.8 apresenta erros ao realizar o <i>match</i> entre a resposta do aluno com a resposta da questão. Além de não ser atualizado. Não permite a avaliação do passo a passo de questões.</p> <p><b>Informações</b></p>

<sup>5</sup> [https://moodle.org/plugins/qtype\\_coderunner](https://moodle.org/plugins/qtype_coderunner)

<sup>6</sup> [https://moodle.org/plugins/qtype\\_algebra](https://moodle.org/plugins/qtype_algebra)

		<b>adicionais:</b> Sua última atualização foi há 3 anos. Possui 102 <i>downloads</i> .
STACK <sup>7</sup>	Implementa dois tipos de classes de perguntas: perguntas que possuem respostas objetivas e as que possuem respostas subjetivas. Nas questões subjetivas, o <i>plugin</i> utiliza o <i>Computer Algebra System</i> (CAS), com finalidade de analisar as respostas obtidas e entregar um <i>feedback</i> . Nas questões objetivas, o <i>plugin</i> funciona semelhante às questões de múltipla escolha já disponíveis no Moodle.	<b>Pontos Positivos:</b> Possui uma interface amigável, assim como, uma preocupação da equipe de desenvolvimento em oferecer ao usuário um maior gerenciamento das suas perguntas, alguns exemplos, como: gerar perguntas aleatórias em modelos estruturados, <i>feedback</i> das respostas dos alunos são analisadas através do CAS.  <b>Pontos Negativos:</b> Apesar de muitas funcionalidades, o <i>plugin</i> não apresenta uma documentação de fácil acesso, o que torna a experiência de uso tortuosa.  <b>Informações adicionais:</b> Sua última atualização foi há 5 meses. Possui 531 <i>downloads</i> .

A partir da instalação e teste dos *plugins* listados na Tabela 1, foi observado que todos os *plugins*, exceto o "*ShortMath*", "*CodeRunner*" e "*STACK*", apresentam uma documentação e interface com poucas informações disponíveis, tornando a criação de questões mais difícil. Por exemplo, o *plugin* "Variable numeric set with units" não possui uma documentação objetiva e nenhuma informação na sua descrição na aba de *plugins*. Outros *plugins*, como o "Spreadsheet", devido a ser um código legado, apresenta dificuldades para rodar nas versões mais recentes do Moodle, não entregando os resultados esperados. É importante mencionar que esses problemas apresentados podem ser causados por falta de atualização, manutenção e evolução do *plugin*, devendo ser levado em consideração ao selecionar um *plugin* para utilizar.

#### 4. SOLUÇÃO DESENVOLVIDA

Definida a proposta inicial de desenvolver um *plugin* para o Moodle, diversas tentativas foram realizadas com o objetivo de criar uma *Proof of Concept* (PoC) para testar a viabilidade de construir um *plugin* para a plataforma do Moodle, mas foram encontradas diversas dificuldades que acabaram inviabilizando o

desenvolvimento de um *plugin* para o Moodle. Como solução, foi pensado em utilizar o SCORM para alcançar um resultado satisfatório e semelhante à criação de um *plugin*. O SCORM se mostrou adequado para o contexto e cenário atual, pois também foi desenvolvido com o objetivo de capturar interações dos usuários e gerenciá-las em um LMS. Assim, utilizou-se dos recursos oferecidos pelo Moodle de exibição e compatibilidade com o SCORM para alcançar o objetivo proposto.

#### 4.1 Aplicação

O SCOMath tem como objetivo capturar entradas do usuário, para personalizar um pacote base. O docente pode editar o pacote definindo um número de passos esperado para uma questão matemática e suas respectivas respostas e, em seguida, disponibilizá-lo para *download*. O pacote editado pode ser executado em qualquer LMS compatível com a versão 2004 do o SCORM, que é a última versão lançada e é considerada a mais completa, seguindo todo o fluxo e funcionamento estabelecido pela ADL.

O pacote base foi desenvolvido a partir de um pacote teste disponibilizado pela própria ADL, onde o mesmo sofreu alterações para corresponder ao contexto matemático. Também foi adicionado features, como a dependência *string-similarity*<sup>8</sup>, que realiza uma comparação entre a resposta esperada e resposta definida pelo usuário, abrangendo uma maior combinação. A similaridade da resposta em cada passo da questão é calculada ao realizar comparação de *strings* com a resposta definida pelo professor (gabarito) para o respectivo passo. É realizado um tratamento nas *strings*, retirando espaços e ordenando-as, para então fazer um *match* para calcular o quanto as *strings* se assemelham. Foi definido uma margem de 95% de similaridade para considerar a resposta como correta, valor que foi definido ao realizar diversas tentativas com intuito de encontrar um peso que permitisse certa flexibilidade das respostas.

#### 4.2 Arquitetura da Aplicação

O SCOMath foi desenvolvido utilizando Flask [4], um *framework* de desenvolvimento web para Python, permitindo a criação de uma arquitetura simples e fácil de manter. A rota disponível para criação de questões é: `/questions`, onde o docente pode definir as questões e respostas e por fim realizar o *download* do pacote.

A Figura 5 ilustra a arquitetura geral do SCOMath, mostrando como as diferentes partes interagem entre si. Quando o usuário interage com os *Templates* (páginas HTML), as comunicações são trocadas entre o front-end e o back-end. O pacote base faz parte do SCOMath e é armazenado em disco, onde é feita todas as alterações definidas pelo usuário, e depois disponibilizado para *download*, para então ser importado em um LMS. O fluxo de dados é direcionado pelos Scripts e a API, permitindo a interação entre o usuário e o pacote SCORM produzido no SCOMath. O SCOMath é separado em basicamente quatro itens, onde:

- *Templates*: São responsáveis por modularizar o conteúdo HTML do *front-end*, além de capturar as entradas do usuário.
- *Scripts*: Possui funções para alterar o pacote SCORM, escrevendo as questões e respostas de acordo com o padrão definido.

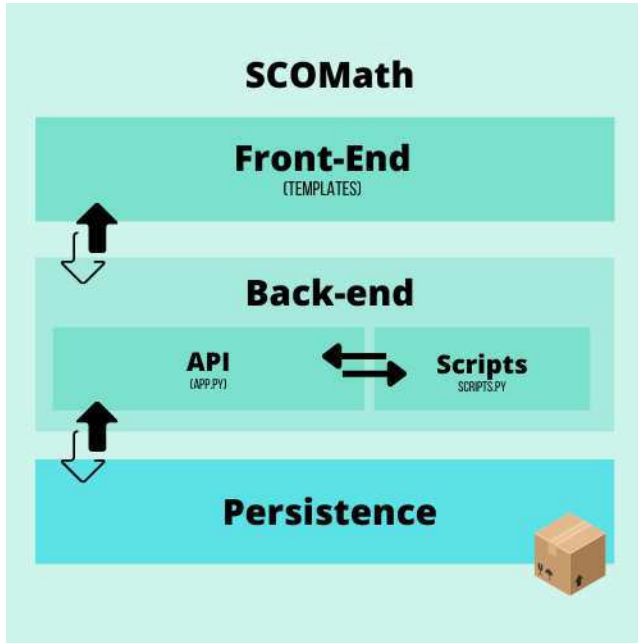
<sup>7</sup> [https://moodle.org/plugins/qtype\\_stack](https://moodle.org/plugins/qtype_stack)

<sup>8</sup> <https://www.npmjs.com/package/string-similarity>



- *API*: app.py é responsável por gerenciar a comunicação entre os templates e scripts, fazendo a ligação entre a camada de visualização e a camada de lógica.
- *Persistence*: Local onde o pacote base do SCOMath é armazenado.

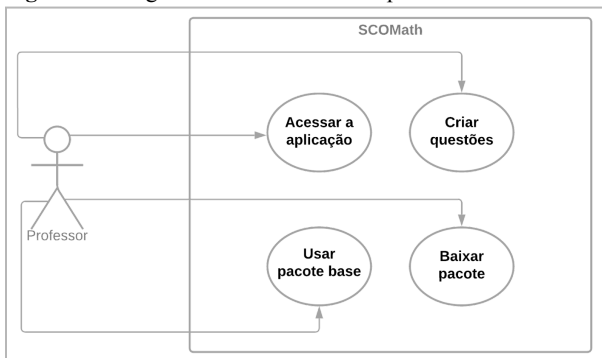
Figura 5 - Diagrama Arquitetural da Aplicação



Fonte: Os autores

O professor interage diretamente com o SCOMath ao acessá-lo, onde definirá a quantidade de passos, perguntas e respostas. Ao preencher os campos, o professor fará o *download* do pacote, como podemos ver no diagrama de casos de uso exibido na Figura 6.

Figura 6 - Diagrama de caso de uso do professor no SCOMath

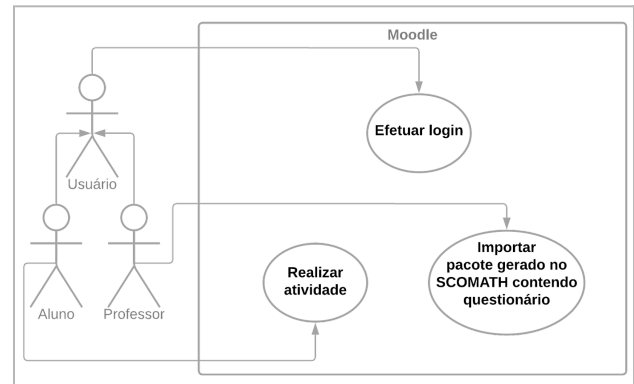


Fonte: Os autores

Na Figura 7, podemos ver o cenário onde o professor faz o *upload* em um LMS, por exemplo, o Moodle. Ao realizar a importação do pacote, o mesmo ficará disponível para os alunos realizarem a atividade. Para fazer o *upload* do pacote no

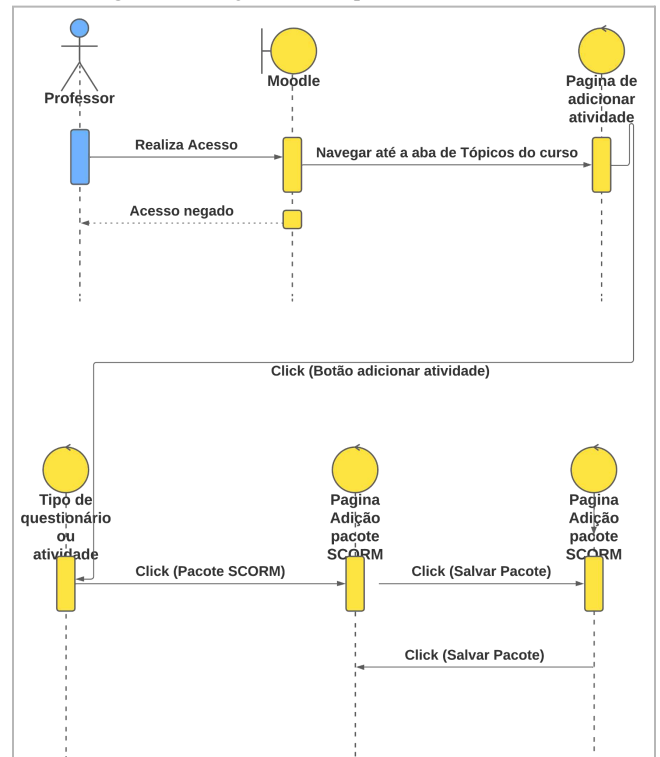
Moodle, o professor necessitará estar logado e ter o cargo de Professor, após isso ele irá navegar até a página do seu curso, buscar o botão de “adicionar atividade”, escolher o tipo de atividade desejada, selecionar a opção pacote SCORM, nomear a atividade e fazer o *upload* do questionário produzido no SCOMath e exportado como um pacote SCORM. O diagrama de seqüência exibido na Figura 8 ilustra esse processo.

Figura 7 - Diagrama de caso de uso do aluno e professor no Moodle



Fonte: Os autores

Figura 8 - Diagrama de seqüência do Professor



Fonte: Os autores

### 4.3 Funcionalidades

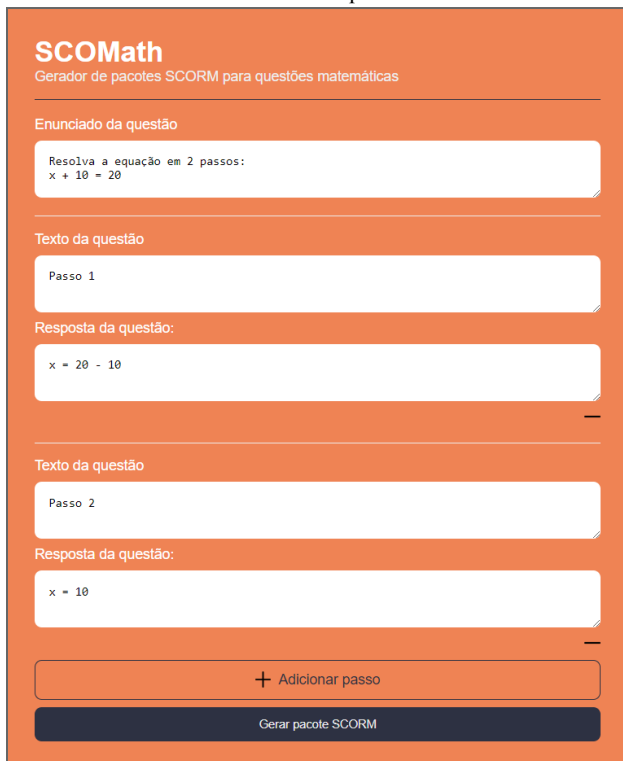
As funcionalidades do SCOMath podem ser divididas em duas perspectivas: a do professor e a do aluno. A seguir será demonstrado um passo a passo da perspectiva do professor e posteriormente do aluno.

**Visão do Professor** para a criação de um pacote:

**Passo 1:** O professor entrará no SCOMath e será redirecionado para área de criar questões para criar uma questão.

**Passo 2:** O professor definirá a quantidade de passos esperados pelos alunos ao responder tal questão. Após finalizar a quantidade de passos e definir suas respectivas respostas (gabaritos), deverá clicar no botão “Gerar pacote SCORM”. A Figura 9 apresenta um exemplo da tela do SCOMath na realização deste passo.

**Figura 9** - Exemplo de interação do professor com o SCOMath ao criar nova questão



Fonte: Os autores

**Passo 3:** Ao clicar no botão “Gerar pacote SCORM” o professor escolherá um local em sua máquina para salvar o pacote.

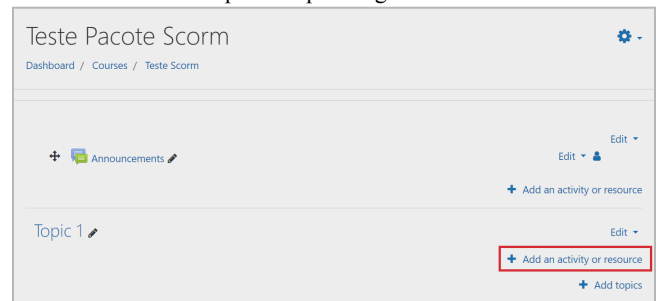
Ao realizar o *download* do pacote, o professor deverá fazer o *upload* do mesmo no Moodle ou em outro LMS compatível com SCORM. Para isso, deverá seguir os seguintes passos:

**Passo 4:** Deverá ir à página do curso, e posteriormente na opção de “Adicionar uma atividade ou recurso”, ilustrado na Figura 10.

**Passo 5:** Ao clicar em “Adicionar uma atividade ou recurso”, o professor deverá selecionar a opção *SCORM Package*, como demonstrado na Figura 11.

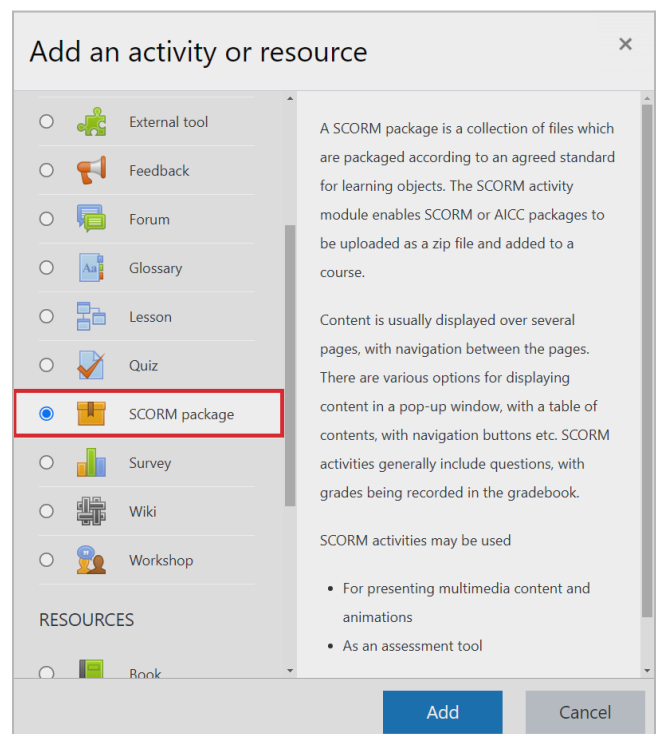
**Passo 6:** Após selecionar a opção *SCORM Package*, deverá clicar no botão Adicionar (*Add*). Onde será encaminhado para a seguinte tela como mostrado Figura 12. Em *General > Name*, o professor deverá preencher o nome da atividade, já na aba *Package* deverá realizar o *upload* do pacote baixado anteriormente, resultado da aplicação. Após isso, poderá utilizar recursos mais avançados disponibilizados pelo Moodle, para um maior controle da atividade e integração com o curso (lista de participantes, tabela de notas, restrições de acesso, etc.), porém com estas configurações básicas, a atividade já está pronta para ser disponibilizada para os alunos.

**Figura 10** - Interação do professor com o LMS Moodle, para importar o pacote gerado



Fonte: Os autores

**Figura 11** - Interação do professor com o LMS Moodle, para escolher o tipo de atividade a ser importada

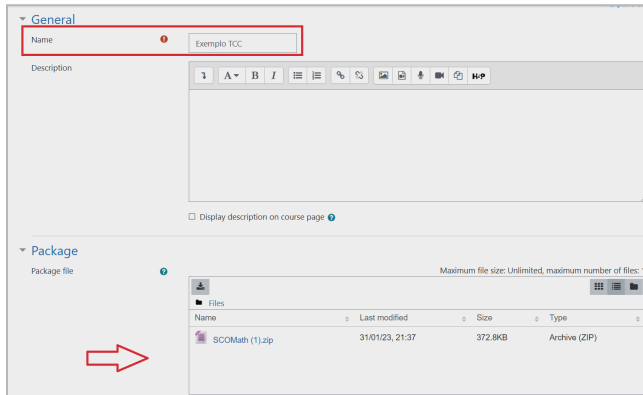


Fonte: Os autores

**Passo 7:** Ao realizar a adição do pacote SCORM, o mesmo ficará disponível para ser respondido pelos alunos, como ilustrado na Figura 13.

Ao finalizar os passos de criação de pacote, de *download* e de *upload* no Moodle, o ciclo do professor chega ao fim. Onde agora basta o aluno responder à atividade proposta.

**Figura 12** - Interação do professor com o LMS Moodle, para definir o nome da atividade, assim como importar o pacote gerado pelo SCOMath.



Fonte: Os autores

**Figura 13** - Interação do professor com LMS Moodle, para visualizar o pacote SCORM adicionado



Fonte: Os autores

**Visão do Aluno:** o aluno irá interagir com a atividade seguindo os passos a seguir.

**Passo 1:** Ao abrir a atividade, o aluno irá se deparar com tal tela. Ao selecionar o modo Normal e clicar no botão Enter, como ilustrado na Figura 14.

**Passo 2:** Após clicar no botão “Enter”. O aluno deverá responder à atividade conforme os passos definidos pelo professor, como podemos observar na Figura 15.

**Passo 3:** Ao preencher a atividade, o mesmo deve clicar em “Enviar questionário” para a submissão do mesmo. A partir de então, o aluno obterá sua nota calculada pelos acertos em cada passo e um *feedback* automático, informando se a resposta estava correta ou não, e qual seria a resposta correta, assim como, a resposta final da questão. Um exemplo dessa tela de consulta/*feedback* é ilustrado na Figura 16.

Após a submissão do questionário, o aluno é avaliado e sua nota é salva no Moodle, finalizando assim, o ciclo de interação do aluno.

#### 4.4 Avaliação preliminar

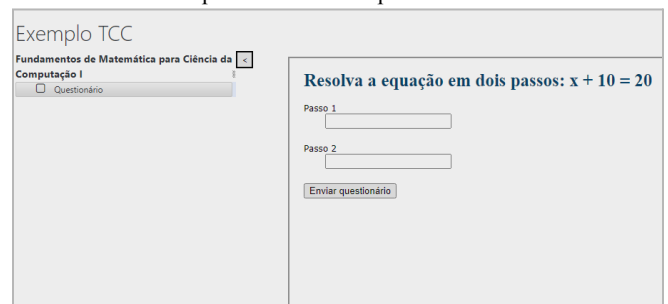
Foram realizados testes utilizando o LMS Moodle, que se mostraram satisfatórios no sentido de produzir questões matemáticas em que fosse possível avaliar, de forma automática, os erros e acertos em cada passo dado pelo aluno, e não apenas a resposta final. Infelizmente, devido a restrições de tempo para a execução deste trabalho, não foi possível avaliar com professores e alunos. No entanto, os testes mostraram que o SCOMath é simples de utilizar, desde que o professor tenha alguma noção básica de Python, necessária para rodar a aplicação localmente. Além disso, verificou-se que o SCOMath consegue resolver o objetivo a que se propõe, embora com algumas limitações, que serão detalhadas na seção seguinte.

**Figura 14** - Interação do aluno com o pacote SCORM e Moodle, para iniciar o questionário



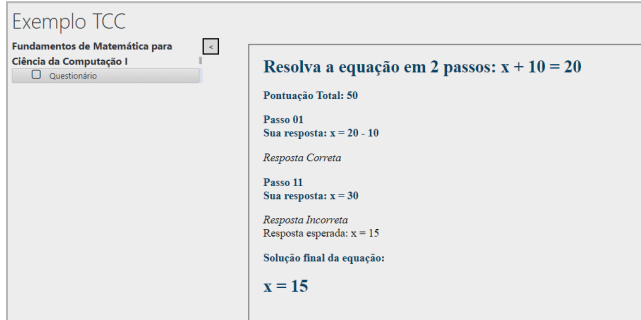
Fonte: Os autores

**Figura 15** - Interação do aluno com o pacote e Moodle, para preencher suas respostas



Fonte: Os autores

**Figura 16** - Visualização do feedback ao finalizar envio do questionário



Fonte: Os autores

## 4.5 Limitações da Aplicação

O SCOMath, assim como o pacote gerado a partir dele, possuem limitações. O pacote, por usar o SCORM, acaba sendo pouco maleável, devido à estrutura do SCORM. Outro fator limitante é que, para realizar alterações no funcionamento de um pacote, é necessário um conhecimento prévio do funcionamento do SCORM. Quanto ao SCOMath, ele possui a limitação de trabalhar apenas em um pacote, sendo necessário o usuário ter um conhecimento básico em programação, tal qual na linguagem de programação Python, para deixar a aplicação rodando localmente. Por fim, a limitação mais impactante é que a aplicação atualmente não possui suporte para a edição de pacotes gerados. Outra limitação é na forma de como cada passo é avaliado, pois ela impõe algumas limitações no contexto da matemática. Por exemplo: se o professor definir como resposta de um passo a expressão: “ $x = 2 - 5$ ”, caso o aluno responda com a expressão “ $x = -5 + 2$ ”, o *feedback* será entregue como resposta incorreta, dado que o método utilizado atualmente não conseguirá interpretar este passo como correto, embora matematicamente ele esteja.

## 5. CONCLUSÃO E TRABALHOS FUTUROS

Como apresentado, o trabalho foi dividido em duas partes: SCOMath e Pacote SCORM. A aplicação desenvolvida utiliza um pacote padrão, pré-configurado, onde será preenchido com as questões e respostas definidas pelo professor e por fim disponibilizado para *download*, onde conseguirá executar em qualquer LMS que seja compatível com o SCORM 2004.

Os resultados alcançados foram satisfatórios entregando um *Minimum Viable Product* (MVP) funcional, atingindo as expectativas geradas, dado a ideia inicial de criar um *plugin* e posteriormente sua adaptação para criação de um pacote para gerar questões. O SCOMath cumpre sua finalidade de gerar um pacote SCORM, respeitando seu fluxo, onde o mesmo é bastante rígido, e ao testar o pacote no Moodle (versão 3.89) o mesmo funcionou como esperado.

Um caminho para trabalhos futuros é na linha de investigar métodos, além de simplesmente comparação de strings, para uma melhoria na verificação de uma resposta correta em cada passo, que esteja mais apropriada com questões matemáticas, conforme exposto nas limitações. Fica também sugestão de trabalho futuro o SCOMath suportar a edição de pacotes. O caminho de receber um arquivo, extrair e comprimi-lo já foi desenvolvido, faltando adequar o template já existente a importação de um pacote. Outro ponto a ser considerado é sobre

a expansão do SCOMath, para outros tipos de questões, assim como, criar várias questões de passo a passo.

## 6. DIFICULDADES E LIÇÕES APRENDIDAS

Durante o desenvolvimento inicial do tema, diversas dificuldades foram encontradas. A documentação do Moodle para a criação de *plugins*, além de ser bastante densa, apresenta comunicados informando que a página de documentação está desatualizada, onde encaminham para outra página com uma documentação atual, porém vários trechos de códigos ainda não se encontram na nova documentação, assim como, códigos apresentados são classificados como legado, pela própria documentação.

Além da documentação, foi percebido que, para o desenvolvimento de um *plugin* para o Moodle, são necessários vários arquivos de código-fonte, possuindo uma alta complexidade de entendimento do fluxo da aplicação. Observou-se uma carência de material técnico que pudesse fornecer um passo a passo para a criação de um tipo específico de *plugin*, assim como, a baixa disponibilidade de tutoriais e documentos explicando básico do funcionamento de um *plugin*.

Desde o início da pesquisa até seu fim, diversas lições foram aprendidas, sendo uma delas: estudar bem o escopo e a viabilidade da ideia antes de partir para o desenvolvimento, uma vez que esperava-se encontrar uma documentação mais direcionada e objetiva, porém foi encontrada uma documentação faltando informações e descontinuada. Outra lição foi a questão da resiliência. Durante o desenvolvimento e estudo da ideia inicial do PTCC, foi visto que o desenvolvimento de um *plugin* de fato não era nada viável dado o tempo e dificuldades, como: documentação, complexidade do sistema, pouca familiaridade com a linguagem PHP, e tais problemas acabaram sendo contornados, de certo modo, usando SCORM como alternativa. Por fim, a experiência adquirida durante todo o processo acabou me tornando melhor como aluno e como pessoa, aprendendo a lidar com eventuais problemas que surjam durante minha vida profissional e pessoal.

## 7. AGRADECIMENTOS

Queridos amigos, familiares e orientador, é com imensa gratidão que escrevo este texto de agradecimento a cada um de vocês. Concluir meu Trabalho de Conclusão de Curso não teria sido possível sem o apoio incansável de cada um de vocês.

Em primeiro lugar, quero agradecer a Deus por sua direção e bênçãos em minha vida. Sem a Sua graça e guia, eu não conseguiria ter alcançado esse objetivo.

Quero agradecer também minha família, Edson Arantes, Lúcia Gomes e Maria Vitória, por seu amor incondicional e apoio constante durante toda essa jornada. Eu sou imensamente grato por terem estado ao meu lado.

E aos meus amigos Gutemberg Silva, Euclides Filho, Igor Franca, Abedess, Eduardo Henrique, Matheus Araujo, Gabriel Nascimento, Hericles Emanuel, Arthur Dantas, João Pedro, Roberto Honório, Lourival Netto, Rafael Dantas, Yuri Nogueira, Carlos Arthur, Gabriel Almeida, Diego Lima, Palmério Júnior, Wislayne Silva, Luan Carlos e Edilberto Júnior, suas palavras de incentivo e disponibilidade em me ajudar foram valiosas e eu nunca esquecerei sua amizade.

Por fim, quero agradecer meu orientador Maxwell Guimarães por sua paciência, orientação e suporte durante todo o processo. Sua contribuição para a conclusão bem-sucedida do meu TCC foi essencial. Obrigado a todos vocês. Eu amo cada um de vocês!

## 8. REFERÊNCIAS

- [1] Moodle plugins directory. Disponível em: <<https://moodle.org/plugins/?q=math%20type:qtype%20sort-by:publish>>. Acesso em: 10 ago. 2022.
- [2] The SCORM content packaging specification. Disponível em: <<https://scorm.com/scorm-explained/technical-scorm/content-packaging/>>. Acesso em: 15 dez. 2022.
- [3] Sharable Content Object Reference Model (SCORM®). Disponível em: <<https://adlnet.gov/past-projects/scorm/>>. Acesso em: 18 dez. 2022.
- [4] Welcome to Flask — Flask Documentation (2.2.x). Disponível em: <<https://flask.palletsprojects.com/en/2.2.x/>>
- [5] ADL. SCORM Users Guide for Programmers. Disponível em: <[https://adlnet.gov/assets/uploads/SCORM\\_Users\\_Guide\\_for\\_Programmers.pdf](https://adlnet.gov/assets/uploads/SCORM_Users_Guide_for_Programmers.pdf)>. Acesso em: 19 dez. 2022.
- [6] MIBLOGTECNICO. Migración de cursos de SCORM 1.2 a SCORM 2004. Disponível em: <<https://miblogtecnico.wordpress.com/2013/01/30/525/>>. Acesso em: 1 fev