

Israel Henrique Silva de Lima

**Um plugin de edição de Mapas Conceituais
para Ambientes Moodle integrado ao CMPaaS**

Vitória

2016

Israel Henrique Silva de Lima

Um plugin de edição de Mapas Conceituais para Ambientes Moodle integrado ao CMPaaS

Monografia apresentada ao Curso de Engenharia de Computação do Departamento de Informática da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Bacharel em Engenharia de Computação.

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Departamento de Informática

Orientador: Prof. M.Sc. Wagner de Andrade Perin

Coorientador: Prof. Ph.D. Davidson Cury

Vitória

2016

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Agradecimentos

Agradeço primeiramente a Deus por todas as graças alcançadas na vida e durante todo o curso.

Aos meus pais, pela educação, apoio emocional, financeiro e tempo dedicado a minha vida.

Em especial a Jalany, pelo carinho, paciência e apoio durante todo o tempo e principalmente o desenvolvimento desta monografia.

Ao meu orientador Wagner, por toda a ajuda oferecida no elaboração deste trabalho.

Ao meu amigo Josmar, pelas longas conversas, orientações e broncas.

Aos meus amigos Piero e Jefferson, pelos momentos de alegria e diversão.

E a todos aqueles que contribuíram direta ou indiretamente nos diversos aspectos de minha formação acadêmica e pessoal.

*“Normal people believe that if it ain’t broke, don’t fix it.
Engineers believe that if it ain’t broke, it doesn’t have enough features yet.
(Scott Adams)*

Resumo

Este trabalho apresenta a concepção e o desenvolvimento de um *plugin* de edição de mapas conceituais em cursos gerenciados pelo Moodle com integração à uma plataforma de serviços de mapas conceituais, o CMPaaS. Aborda a importância do Moodle na Educação a Distância e detalha os processos e tecnologias utilizadas na construção de um *plugin* para ele. Discute também como é possível integrar o Moodle com outras plataformas de serviços.

Palavras-chave: Mapas Conceituais. Moodle. CMPaaS.

Abstract

This work presents the design and development of a concept map editing plugin in courses managed on Moodle with integration to a platform of Conceptual mapping services, the CMPaaS. Addresses the importance of Moodle in Education Distance and details the processes and technologies used to build a plugin for him. It also discusses how it is possible to integrate Moodle with other services.

Keywords: conceptual maps. Moodle. CMPaaS.

Lista de ilustrações

Figura 1 – Um exemplo de mapa conceitual	13
Figura 2 – (a) Integração do Portal do Conhecimento com o CMPaaS. (b) Integração do CMPaaS com serviços externos	16
Figura 3 – Interface de um curso no Moodle	18
Figura 4 – Casos de Uso	21
Figura 5 – Diagrama de Sequencia do Caso de Uso Salvar Mapa	21
Figura 6 – Diagrama de Sequencia do Caso de Uso Sincronizar Mapa	22
Figura 7 – Um plugin de elaboração de mapas conceituais	24
Figura 8 – Editor com barra de formatação	25
Figura 9 – Componentes do GoJS	26
Figura 10 – Um diagrama gerado em GoJS	26
Figura 11 – Código dos elementos Shape e TextBlock	27
Figura 12 – Rotina que altera a cor de preenchimento de um nó	28
Figura 13 – Configurando a tarefa para utilizar o <i>plugin</i>	31
Figura 14 – Lista de <i>plugins</i> instalados	31
Figura 15 – Configurando a tarefa para utilizar o plugin	32
Figura 16 – Editor de mapas	32
Figura 17 – Sumário da tarefa	33
Figura 18 – Mapa que utiliza os recursos de formação	34
Figura 19 – Menus dropdown da barra de ferramentas	34

Sumário

1	INTRODUÇÃO	10
1.1	Objetivo	10
1.2	Metodologia	11
1.3	Estrutura deste trabalho	11
2	MAPAS CONCEITUAIS E EDUCAÇÃO	13
2.1	Origem e definição de mapas conceituais	13
2.2	Mapas conceituais e educação	14
2.3	Mapas conceituais e tecnologia da informação	14
2.3.1	CMPaaS	15
2.4	Moodle e Educação à Distância (EaD)	17
3	O PROJETO	20
3.1	Arquitetura	20
3.2	Tecnologias utilizadas	22
3.2.1	JavaScript	22
3.2.2	Json	22
3.2.3	PHP	23
3.3	Trabalhos correlatos	23
4	DESENVOLVIMENTO	25
4.1	Serviço de Edição de Mapas	25
4.1.1	Implementação das funções de formatação	26
4.2	Implementação no Moodle	28
4.2.1	Desenvolvimento do plugin	29
4.2.2	Integração com o CMPaaS	30
5	IMPLEMENTAÇÃO E TESTES	31
5.1	Como instalar	31
5.2	Como usar	32
5.3	Provas de Conceitos	33
6	CONSIDERAÇÕES FINAIS	35
6.1	Resultados	35
6.2	Conclusão	35
6.3	Trabalhos futuros	36

REFERÊNCIAS 37

1 Introdução

Os mapas conceituais são uma ferramenta gráfica criada na década de 70, com o objetivo de ser uma representação do conhecimento. São muito fáceis de utilizar e possuem ampla aplicação na educação, podendo ser usados para apresentar o conteúdo de uma aula, de um livro ou capítulo ou ainda como método de avaliação. Por ser muito simples, podem ser elaborados com lápis e papel, porém é possível alcançar novas aplicações ao se utilizar recursos computacionais.

A proposta deste trabalho é justamente utilizar a computação para ampliar a aplicação dos mapas conceituais. Neste contexto foi pensado na criação de uma ferramenta que permita o uso dos mapas no ensino a distância. Assim, este projeto visa o desenvolvimento de um *plugin* para o gerenciador de cursos *Modular Objected-Oriented Dynamic Learning Environment* (Moodle) que introduza os mapas conceituais nos cursos existentes nesta plataforma.

O Moodle foi escolhido para ser a plataforma de implementação deste projeto, pois é uma plataforma *open-source*, ou seja, pode ser livremente modificada, e também é modular, permitindo que novas ferramentas sejam adicionadas a ela facilmente. Além é claro, de ser amplamente utilizada por instituições de ensino no mundo todo. Avaliando a lista de *plugins* do Moodle, foi constatado que não há nenhuma ferramenta que permita o uso de mapas conceituais em tarefas de cursos gerenciados pela plataforma em suas versões mais recentes.

O produto deste trabalho permitirá que estudantes dos cursos gerenciados pelo Moodle realizem atividades com a elaboração de mapas conceituais. Adicionado a isso, o editor de mapas que será implementado também irá possuir integração com o *Conceptual Maps as a Service* (CMPaaS), uma plataforma que oferece diversos serviços de manipulação de mapas. Alguns desses serviços necessitam de conteúdo previamente existente para funcionar, como por exemplo o serviço de inferência de mapas. Devido ao Moodle ser muito utilizado, um *plugin* de elaboração de mapas conceituais implementado nele poderia gerar uma grande quantidade de conteúdo, que os serviços da plataforma CMPaaS poderiam utilizar, por exemplo, o serviço de geração de inferências. Por esse motivo o *plugin* desenvolvido nesse trabalho também será integrado ao CMPaaS.

1.1 Objetivo

O produto final deste trabalho será um *plugin* que disponibiliza um editor de criação de mapas conceituais em cursos da plataforma Moodle. O *plugin* utilizará a plataforma

CMPaaS como solução de armazenamento dos mapas conceituais na nuvem.

Para alcançar o objetivo principal foram definidos os seguintes objetivos específicos:

- Estudar o que são mapas conceituais e suas aplicações na educação.
- Estudar a arquitetura do Moodle e seus *plugins*.
- Entender a plataforma CMPaaS.
- Desenvolver um novo *plugin* para o Moodle segundo a arquitetura da plataforma.

1.2 Metodologia

Para alcançar o objetivo deste trabalho inicialmente foi feita uma revisão bibliográfica visando entender o que são mapas conceituais e a sua importância para a educação. Foi feito também um estudo da arquitetura do Moodle, de forma a entender o seu funcionamento e como podem ser desenvolvidos novos *plugins* para adicionar novos recursos à plataforma.

Após isto houve a realização de um estudo sobre quais tecnologias poderiam ser utilizadas no desenvolvimento de um *plugin* para o Moodle com a funcionalidade de responder tarefas com a elaboração de mapas conceituais e que pudesse utilizar o repositório de mapas do CPMaaS.

Para finalizar, o *plugin* foi implementado no Moodle e foram realizados testes de suas funcionalidades.

1.3 Estrutura deste trabalho

Este trabalho será dividido em sete capítulos.

O capítulo “Introdução” apresenta a forma que o trabalho foi planejado e organizado. Este capítulo contém uma introdução sobre o projeto, seus objetivos e a metodologia de desenvolvimento.

O capítulo “Mapas Conceituais e Educação” expõe o que são mapas conceituais e a sua importância no ensino. Também conceitua educação a distância e apresenta a plataforma Moodle.

O capítulo “CMPAAS” faz uma descrição sobre a plataforma CMPaaS e a sua arquitetura.

O capítulo “O Projeto” descreve como foi planejado o desenvolvimento do *plugin*, apresentando suas funcionalidades e como será implementado.

O capítulo “Desenvolvimento” detalha como o *plugin* foi desenvolvido.

O capítulo “Implementação e Testes” apresenta o resultado final do projeto, demonstrando sua instalação, configuração e utilização.

O capítulo “Considerações Finais” faz uma avaliação dos resultados obtidos no trabalho e propõe trabalhos futuros.

2.2 Mapas conceituais e educação

Os mapas conceituais podem ter diversas aplicações na educação. Podem por exemplo ser utilizados para representar o conhecimento adquirido em uma aula, ou então o conteúdo de um capítulo de um livro. Por se tratar de uma representação que é alimentada com novos conceitos conforme eles são adquiridos, pode ser utilizado pelo professor para acompanhar a aprendizagem de um estudante durante todo o período de um curso. Ademais, como os mapas conseguem representar a estrutura cognitiva de um indivíduo, eles podem ser utilizados também como método de avaliação de aprendizagem (PERIN, 2014).

A confiabilidade da avaliação da aprendizagem tradicional, como textos dissertativos e questões de múltipla escolha, tem sido questionada por pesquisadores e educadores. Nestes tipos de teste apenas os acertos são contabilizados e os erros são descartados, implicando em importantes informações para a avaliação serem desprezadas. As provas tradicionais não proporcionam a possibilidade do estudante apresentar como construiu o seu aprendizado. Elas conseguem avaliar somente a aprendizagem mecânica, não mostrando como que o aprendiz alterou suas estruturas cognitivas. Os mapas conceituais tem se destacado como alternativa a avaliação tradicional, pois conseguem demonstrar com facilidade as modificações cognitivas que ocorrem durante o processo de aprendizagem do estudante (DUTRA; FAGUNDES; CAÑAS, 2002).

Neste contexto, o *plugin* que será desenvolvido neste trabalho tem como finalidade permitir que estudantes de cursos gerenciados pelo Moodle respondam tarefas avaliativas com a elaboração de mapas conceituais. Assim poderá ampliar muito o uso de mapas como método de avaliação, já que o Moodle é uma plataforma utilizada por diversas instituições.

2.3 Mapas conceituais e tecnologia da informação

A construção de mapas conceituais é muito simples, uma caneta e um pedaço de papel são ferramentas suficientes para a confecção desse grafo. Porém a tarefa de revisá-lo, armazená-lo, e editá-lo a longo prazo pode ser muito cansativa e complexa. A introdução do uso de computadores na confecção de mapas pode facilitar a tarefa (NOVAK; CAÑAS, 2006).

Os primeiros programas de computadores criados com este objetivo se limitavam a mostrar apenas os mapas na tela, sem oferecer nenhum recurso adicional a ferramenta. Com a popularização do uso de computadores e da Internet houve um aumento na oferta de aplicações que tem como objetivo a construção de mapas gráficos. Porém há uma grande confusão entre usuários e desenvolvedores sobre a diferença entre mapas conceituais, organogramas e mapas mentais, devido a semelhança entre os diagramas (PERIN, 2014). Um mapa mental, por exemplo, é muito semelhante a um mapa conceitual

pois apresenta ligações entre ideias, mas ao contrario de um mapa mental não possui uma palavra descritiva ligando essas ideias e criando organicidade entre as ideias. Portanto uma aplicação de criação de mapas mentais não permite a construção de um mapa conceitual.

Na seção 2.3 de sua dissertação de mestrado, [Perin \(2014\)](#) realizou uma pesquisa sobre o estado da prática em mapas conceituais. Nela foi feito um levantamento sobre softwares que permitem a criação de mapas conceituais e quais são as funcionalidades oferecidas por eles. Nesta pesquisa foram avaliadas 16 ferramentas de edição de mapas, das quais apenas o CMapTools foi desenvolvido com o objetivo de oferecer suporte específico para mapas conceituais.

Percebendo esta carência, um grupo de pesquisadores do laboratório de Informática na Educação da Universidade Federal do Espírito Santo (Ufes) propôs uma plataforma de serviços para mapas conceituais, conhecida por CMPaaS ([PERIN; CURY, 2016](#)) cujo objetivo é fornecer serviços avançados para facilitar os processos e as aplicações dos mapas conceituais, tanto no ensino quanto no mercado. Neste trabalho, interessa-nos a criação de uma ferramenta integrada ao CMPaaS que proporcione a aplicação dos mapas conceituais na Educação a Distância.

2.3.1 CMPaaS

Na [seção 2.2](#) vimos como os mapas conceituais são importantes para educação, seja para auxiliar o estudante a aprender novos conceitos, ou como método de avaliação, além de algumas outras aplicações educacionais.

Apesar de haver este grande interesse acadêmico em ferramentas computacionais para criação e manipulação de mapas, a fragmentação das pesquisas nesta área impede que o desenvolvimento destas aplicações ocorra de forma acelerada. Assim, a integração de ferramentas já existentes facilitaria o progresso de novas pesquisas.

Se houvesse uma infraestrutura de apoio a criação e manipulação de mapas conceituais o desenvolvimento de novas ferramentas seria facilitado. Um pesquisador poderia criar uma nova solução mais facilmente se não houvesse a necessidade dele se preocupar com toda uma estrutura de gestão de mapas. Por exemplo, alguém que queira desenvolver uma aplicação que avalie mapas conceituais poderia utilizar o conteúdo existente previamente em uma solução computacional de criação e armazenamento de mapas.

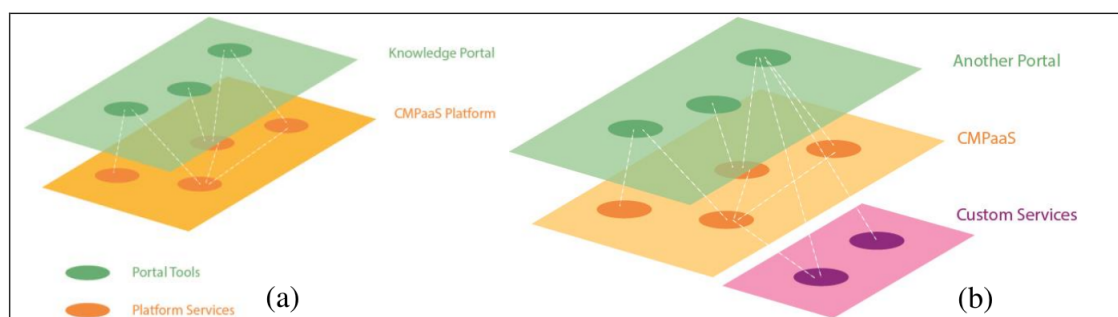
Um outro problema apontado por [Perin \(2014\)](#) é a dificuldade que a sociedade tem para acessar os resultados das pesquisas.

Consideramos importante a criação de um mecanismo de acesso eficiente aos resultados das pesquisas científicas, para que a comunidade possa contribuir para a evolução delas. O que propomos, portanto, é o lançamento de bases para uma convivência mais estreita entre o mundo acadêmico e a sociedade em geral ([PERIN, 2014](#)).

Assim foi proposto por [Perin \(2014\)](#) a criação de uma plataforma denominada CMPaaS, cujo objetivo é possibilitar que a comunidade em geral acesse o resultado de pesquisas acadêmicas e prover uma infraestrutura que permita que esta comunidade crie e estenda suas funcionalidades.

Os serviços do CMPaaS são oferecidos através da Internet, podendo ser acessado de qualquer plataforma. Mas é necessário que exista um portal ou site associado a plataforma CMPaaS para que as ferramentas possam ser acessadas através de um navegador de Internet. Este portal foi nomeado como "Portal do Conhecimento" ([PERIN, 2014](#)).

Figura 2 – (a) Integração do Portal do Conhecimento com o CMPaaS. (b) Integração do CMPaaS com serviços externos



Fonte: [Perin \(2014, p. 81\)](#)

A [Figura 2](#) ilustra a arquitetura do CMPaaS e como as aplicações fornecidas pelo portal utilizam os serviços da plataforma, muitas vezes consumindo mais de um dos serviços oferecidos. Além disso, é ilustrado também como um portal externo pode aproveitar os serviços disponíveis no CMPaaS, como é o caso do plugin que será apresentado posteriormente neste trabalho. Trata-se de um editor de mapas conceituais que armazena os mapas na plataforma. Assim são utilizados dois serviços fornecidos pelo CMPaaS: o de armazenagem de mapas e o de autenticação.

A estrutura interna da plataforma é composta por cinco camadas: serviços externos são os serviços oferecidos pelo CMPaaS aos usuários (é o que aparece no site para o usuário); processos de negócio são as rotinas que ocorrem internamente no CMPaaS; componentes internos são as aplicações internas da plataforma; serviços de aplicações externas são serviços produzidos por ferramentas externas e são consumidos por componentes do CMPaaS; e os serviços de aplicações internas são serviços produzidos por

Atualmente a plataforma CMPaaS oferece os seguintes serviços:

- Criação, edição e formatação de mapas conceituais.
- Serviço de cadastramento e autenticação de usuários.

- Persistência e repositório de mapas.
- Controle de versão de mapas conceituais.
- Importação e exportação de mapas para o CmapTools
- Serviço de inferência para mapas, que permite que o usuário elabore perguntas sobre os mapas em linguagem natural.
- Criação de ontologias rasas a partir de um mapa.
- Geração automática de mapas a partir de textos.
- Serviço de mesclagem de mapas conceituais.
- Validação estrutural de mapas.

2.4 Moodle e Educação à Distância (EaD)

Segundo [CHAVES \(1999\)](#) EaD é um ensino que ocorre quando há uma separação física entre mestre e aprendiz, sendo utilizadas tecnologias de transmissão de voz, dados e imagens para a comunicação entre ambos. A EaD também é definida no Decreto nº 5.622 de 19 de dezembro de 2005 ([BRASIL, 2005](#)).

Art. 1o Para os fins deste Decreto, caracteriza-se a educação a distância como modalidade educacional na qual a mediação didático-pedagógica nos processos de ensino e aprendizagem ocorre com a utilização de meios e tecnologias de informação e comunicação, com estudantes e professores desenvolvendo atividades educativas em lugares ou tempos diversos ([BRASIL, 2005](#)).

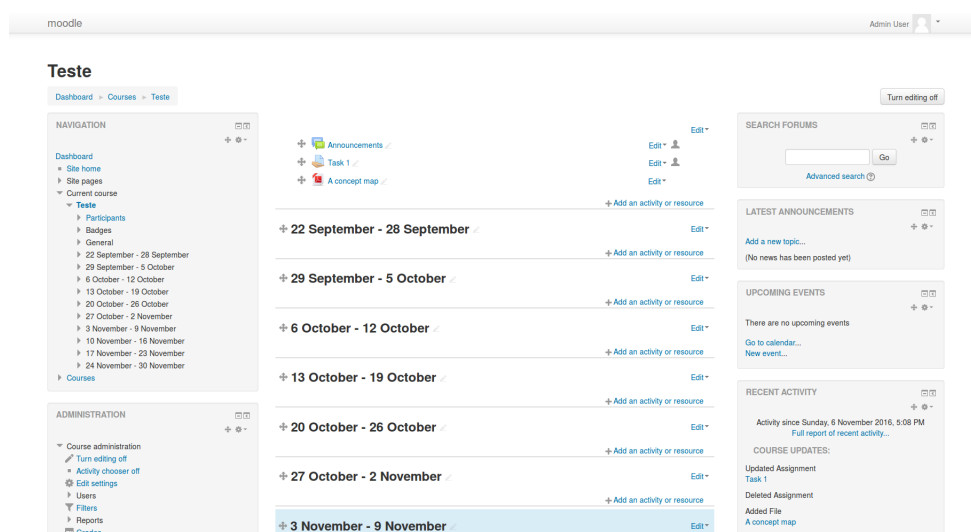
Em síntese é possível definir a educação a distância como um método de ensino no qual estudante e professor ficam separados fisicamente e a interação é feita através de tecnologias de comunicação (que podem ser dos mais diversos) de maneira a contornar esta separação.

Atualmente a educação a distância está relacionada ao uso de um Ambiente Virtual de Aprendizagem (AVA), que nada mais é que um sistema computacional cuja finalidade é prover suporte a atividades mediadas pelas tecnologias de informação e comunicação ([ALMEIDA, 2010](#)). É um ambiente que tem como objetivo integrar mídias e recursos, além de apresentar informações de forma organizada e permitir interação entre estudantes, tutores e professores ([FRANCISCATO et al., 2008](#)). Um exemplo de um AVA são os cursos que podem ser criados e gerenciados no Moodle.

O Moodle é um sistema gerenciador de cursos que foi criado por Martin Dougiamas em 1999. Ele é uma aplicação *open-source*, o que significa que ele é livre para ser instalado, utilizado, modificado e distribuído ([DOUGIAMAS; TAYLOR, 2003](#)).

O Moodle trabalha por padrão com cinco perfis de usuário: administrador, criador de cursos, professor, aluno e visitante. O administrador é o responsável técnico, é ele quem realiza a instalação e configuração do ambiente, além de manter ele funcionando corretamente. Já o criador de cursos tem como responsabilidade a criação e configuração dos cursos disponíveis na plataforma. Por sua vez, o professor tem como função o acompanhamento dos estudantes e a inserção de recursos e tarefas nos cursos. O aluno é quem realiza o curso, é ele quem vai utilizar os recursos e tarefas disponíveis no AVA. Por fim, o visitante é um usuário que só tem acesso às informações disponíveis na tela inicial do sistema. A Figura 3 mostra a interface de um curso no Moodle.

Figura 3 – Interface de um curso no Moodle



Fonte: Elaborada pelo autor

Ele foi escolhido para ser utilizado neste trabalho pois é amplamente usado por instituições em todo o mundo, possuindo mais de 73 mil sites registrados em 232 países (MOODLE, 2016b), e possui uma grande comunidade que contribui para correção de erros e criação de novas ferramentas. Além disso ele é utilizado para gerenciar os AVAs da Ufes, e é muito utilizado também em diversas instituições no país, sendo o Brasil o terceiro maior utilizador da plataforma, conforme pode ser visto na Tabela 1.

O Moodle é composto por módulos instaláveis, configuráveis e estendíveis. Assim, é possível desenvolver *plugins* e componentes que adicionam novas funcionalidades à plataforma, como é o caso do que está sendo desenvolvido neste trabalho.

O que propomos neste projeto é, portanto, a criação de um novo *plugin* para o Moodle que permita ao usuário construir, armazenar e recuperar seus mapas dentro desta plataforma. Além disto, propomos uma arquitetura para integração do Moodle com o CMPaaS de modo a permitir que mapas produzidos dentro da Plataforma Moodle sejam sincronizados com a Nuvem oferecida pelo projeto CMPaaS. No capítulo a seguir será

possível conhecer as características básicas do CMPaaS e os serviços oferecidos por esta plataforma.

Tabela 1 – Os 10 países que mais utilizam o Moodle.

País	Sites Registrados
Estados Unidos	10.131
Espanha	7.067
Brasil	4.401
Reino Unido	3.486
México	3.464
Alemanha	2.444
Itália	2.414
Austrália	2.324
Colômbia	2.264
Rússia	1.993

Fonte: [Moodle \(2016b\)](#)

3 O Projeto

3.1 Arquitetura

O Moodle é composto por vários *plugins* independentes, sendo cada um deles responsável por uma funcionalidade que a plataforma oferece. O escopo deste trabalho é o desenvolvimento de um novo *plugin*. Neste projeto foi criado *plugin* do tipo *assignment submission plugin*, que tem com função a submissão de uma tarefa. O *plugin* de tarefa permite que um professor crie uma atividade, avaliativa ou não, para os estudantes realizarem.

Este tipo de *plugin* é dividido em três partes:

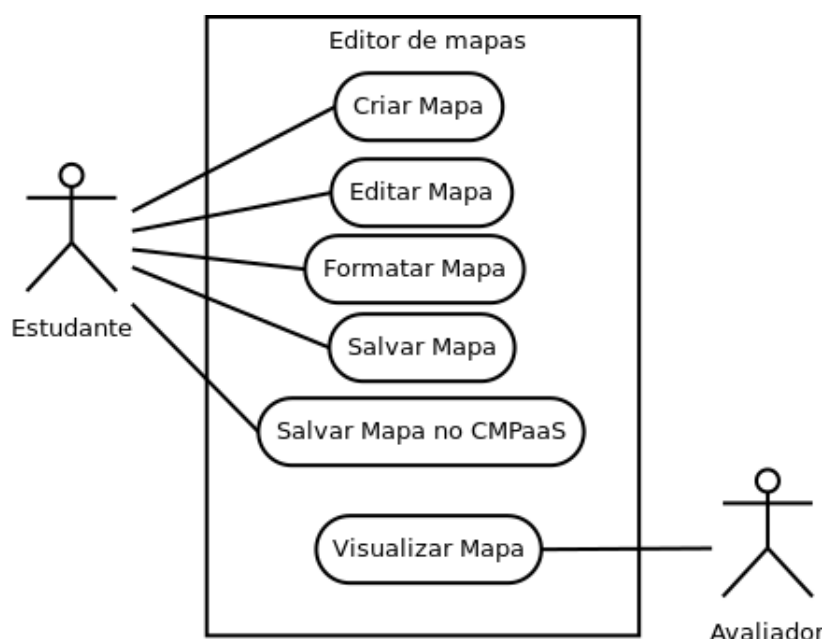
- configurações, que são opções de comportamento do *plugin*.
- sumário de envio, que será visto pelo estudante e avaliadores na tela inicial da tarefa.
- formulário de envio, que é onde o estudante responde a tarefa.

Este *plugin* oferece as seguintes funcionalidades:

- criação de um novo mapa conceitual utilizando um editor de mapas. Esta funcionalidade é representada pelo caso de uso Criar Mapa.
- armazenagem do mapa criado no banco de dados do Moodle. Esta funcionalidade é representada pelo caso de uso Salvar Mapa.
- edição de um mapa previamente salvo no banco de dados do Moodle. Esta funcionalidade é representada pelo caso de uso Editar Mapa.
- alteração da aparência de um mapa que está sendo criado ou modificado no editor. Esta funcionalidade é representada pelo caso de uso Formatar Mapa.
- possibilidade de salvar o mapa que está sendo criado na plataforma CMPaaS. Esta funcionalidade é representada pelo caso de uso Salvar Mapa no CMPaaS.
- visualizar um mapa salvo para que possa ser avaliado. Esta funcionalidade é representada pelo caso de uso Visualizar Mapa.

Os atores que irão utilizar o *plugin* são o Estudante, que irá criar e editar mapas conceituais que serão submetidos a avaliação, e um Avaliador, que irá visualizar os mapas criados pelos estudantes e avaliá-lo. Os atores e casos de uso são representados na [Figura 4](#).

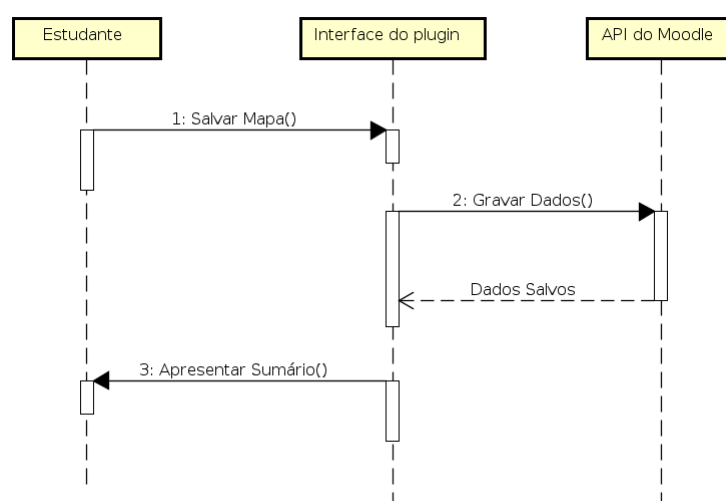
Figura 4 – Casos de Uso



Fonte: Elaborado pelo autor

O caso de uso Salvar Mapa é apresentado na [Figura 5](#). Quando o estudante solicita que o mapa seja salvo a interface do *plugin* chama a rotina da API do Moodle responsável por enviar os dados do formulário para serem armazenados. Após esta rotina ser executada e o mapa ser armazenado um sumário do envio é apresentado ao estudante.

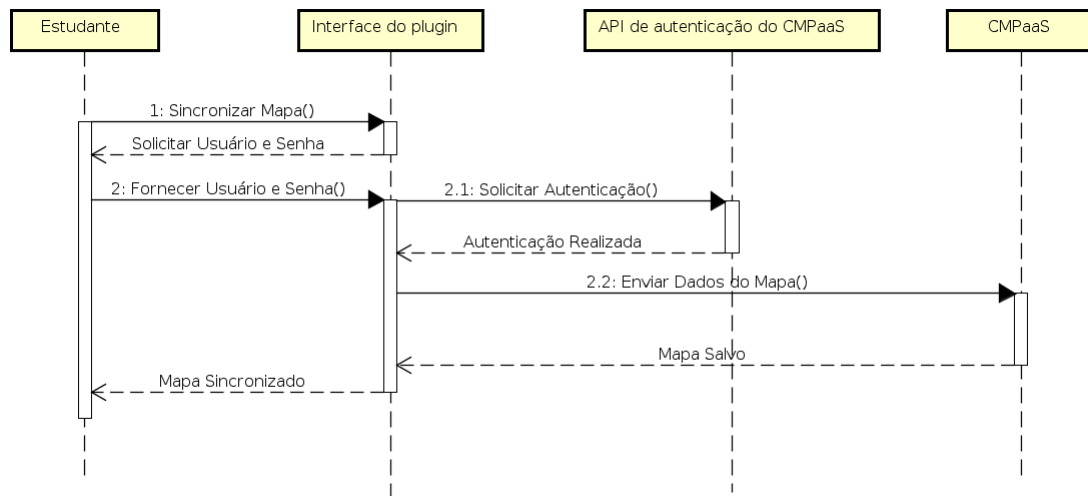
Figura 5 – Diagrama de Sequencia do Caso de Uso Salvar Mapa



Fonte: Elaborado pelo autor

O caso de uso Sincronizar Mapa é apresentado na [Figura 6](#). É este caso de uso que

Figura 6 – Diagrama de Sequencia do Caso de Uso Sincronizar Mapa



Fonte: Elaborado pelo autor

realiza a integração do *plugin* com a plataforma CMPaaS. Quando o estudante executa a ação de sincronização do mapa a interface do *plugin* chama uma rotina para que seja feita autenticação no portal. Depois que o estudante é autenticado no CMPaaS os dados do mapa são enviados para serem armazenados na plataforma.

3.2 Tecnologias utilizadas

3.2.1 JavaScript

O JavaScript é uma linguagem orientada a objeto criada em 1995 por Brendan Eich que foi idealizada para permitir que pessoas que não são programadoras pudessem estender as funcionalidades de sites de Internet (RICHARDS et al., 2010). Ela é uma linguagem interpretada, e apesar de ter sido criada para desenvolvimento web, ela é uma linguagem de proposito geral, e pode ser usada para o desenvolvimento de qualquer tipo de aplicação (FLANAGAN, 2006).

Ela foi escolhida para ser utilizada neste projeto pois o editor de mapas disponível no CMPaaS foi desenvolvido com esta tecnologia e, além disso, trata-se de uma linguagem compatível com todos os navegadores modernos.

3.2.2 Json

O *JavaScript Object Notation* (Json) é uma linguagem em formato de texto cuja função é a serialização de dados estruturados. A serialização é o processo de transformar objetos em um fluxo de *bytes* para ser armazenado em um banco de dados ou disco. Ele

pode ser utilizado para representar tipos primitivos, como cadeia de caracteres e números, ou tipos estruturados, como objetos e vetores (CROCKFORD, 2006). Ele é utilizado para intercâmbio de dados entre os serviços da plataforma CMPaaS.

3.2.3 PHP

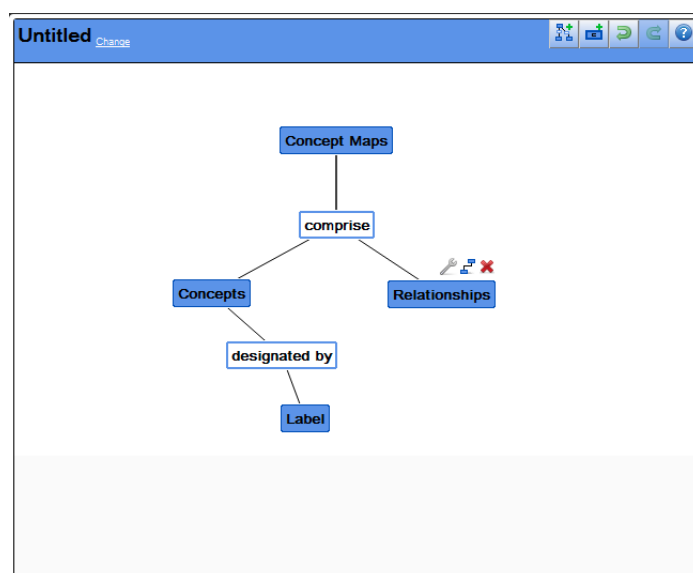
PHP é um acrônimo recursivo para *PHP: Hypertext Preprocessor*, originalmente ele foi criado como uma linguagem de script estruturada cuja finalidade era o desenvolvimento de aplicações com a funcionalidade de geração de HTML dinâmico. Com o passar do tempo a linguagem evoluiu e passou a oferecer recursos para o desenvolvimento orientado a objeto (MINETTO, 2007).

A plataforma Moodle e seus *plugins* são desenvolvidos em PHP, por isto esta linguagem foi utilizada neste projeto.

3.3 Trabalhos correlatos

Existe um *plugin* semelhante ao desenvolvido neste trabalho no repositório do Moodle. Este *plugin* possibilita que estudantes respondam questionários com a elaboração de um mapa conceitual. As principais funcionalidades dele são a possibilidade de criar um número ilimitado de conceitos e relacionamentos, compatibilidade com padrões de internacionalização e persistência de mapas com a utilização de XML. Infelizmente a última versão dele foi lançada em fevereiro de 2015 e não é compatível com as versões mais atuais do Moodle. A Figura 7 mostra um mapa criado neste *plugin*.

Figura 7 – Um plugin de elaboração de mapas conceituais



Fonte: Moodle (2016a)

4 Desenvolvimento

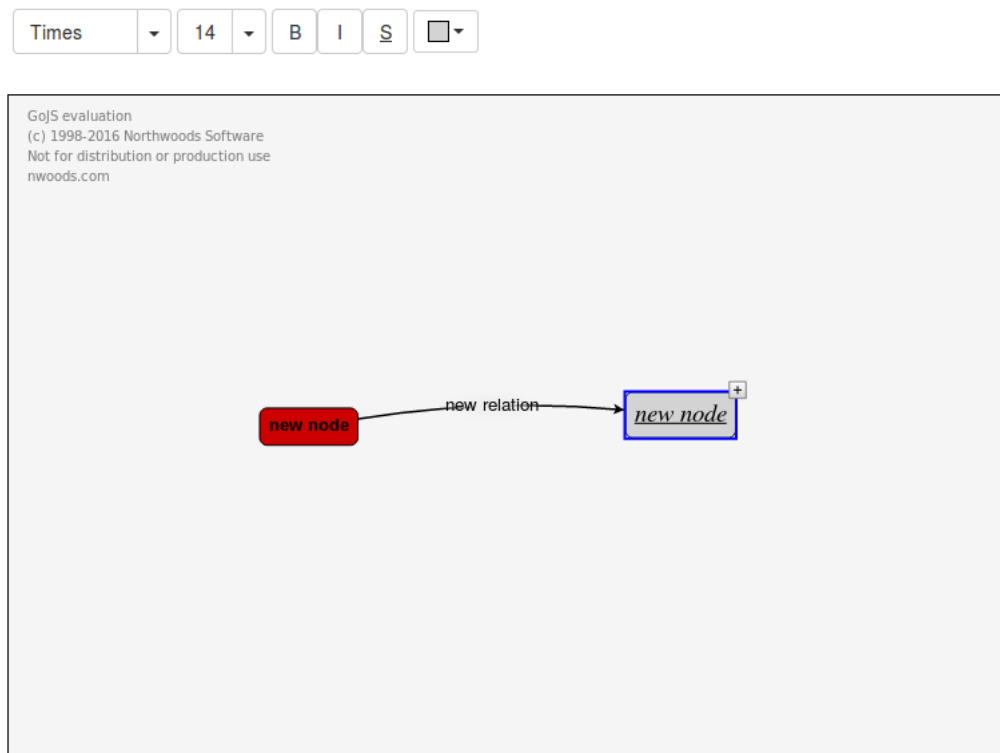
4.1 Serviço de Edição de Mapas

O serviço de edição de mapas foi implementado com a utilização do editor de mapas conceituais disponível no CMPaaS. Ele foi desenvolvido utilizando GoJS, que é uma biblioteca em JavaScript para criação de diagramas para navegadores *Web*.

Este editor salva os mapas no formato Json, possibilitando que outras aplicações utilizem os dados que ele produz. Foi concebido desta forma para acompanhar a proposta do CMPaaS de permitir a interoperabilidade dos serviços oferecidos.

A primeira parte do trabalho foi realizar um aprimoramento deste editor, adicionando funções para a formatação dos mapas criados. Não existia por exemplo, a possibilidade de alterar a fonte da letra dos conceitos ou a cor de fundo dos nós. Uma barra de ferramenta foi adicionada ao editor original para que funções de formatação estivessem disponíveis.

Figura 8 – Editor com barra de formatação



Fonte: Elaborada pelo autor

4.1.1 Implementação das funções de formatação

O editor de mapas é dividido em dois componentes, um Model e um Diagram. O Model é o que contém os dados do mapa que está sendo criado, nele os nós e links são descritos por vetores de objetos JavaScript. Já o Diagram é usado para visualizar os dados contidos no modelo.

Na [Figura 9](#) temos um trecho de código que gera um diagrama simples em GoJS. Nele é criado um Model que possui um vetor com três objetos JavaScript, nomeados Alpha, Beta e Gamma. Este vetor de objetos da classe Node representa três nós que serão apresentados no diagrama. O Model nomeado myModel é então adicionado ao Diagram chamado myDiagram. Este trecho de código gera o diagrama ilustrado na [Figura 10](#).

Figura 9 – Componentes do GoJS

```
myDiagram =
    $(go.Diagram, "myDiagram",
        { initialContentAlignment: go.Spot.Center,
          "toolManager.mouseWheelBehavior": go.ToolManager.WheelZoom,
          "undoManager.isEnabled": true,
          "clickCreatingTool.archetypeNodeData": { text: "new node" }
        });

var myModel = $(go.Model);
myModel.nodeDataArray = [
    { key: "Alpha" },
    { key: "Beta" },
    { key: "Gamma" }
];
myDiagram.model = myModel;
```

Fonte: Elaborada pelo autor

Figura 10 – Um diagrama gerado em GoJS



Fonte: Elaborada pelo autor

Cada nó do mapa criado pelo editor é representado por um objeto da classe `Node`, que por sua vez, é composto por blocos que determinam a sua aparência. Os blocos que o editor utiliza são o `Shape` e o `TextBlock`. A classe `Shape` é utilizada para mostrar uma forma geométrica colorida. Já a classe `TextBlock` tem como função mostrar um texto. Ambas as classes possuem diversas propriedades que servem para determinar a sua aparência e o seu comportamento no diagrama.

Os trechos de código abaixo são usados para criar um `Shape` e um `TextBlock`. A forma geométrica desenhada é um retângulo com largura de 40 pixels, altura de 60 pixels, com margem de 4 pixels e preenchimento na cor vermelha. Já o `TextBlock` criado é um texto “a Text Block” de cor vermelha.

Figura 11 – Código dos elementos `Shape` e `TextBlock`

```
$(go.Shape, "Rectangle", { width: 40, height: 60, margin: 4, fill: "red" })
$(go.TextBlock, { text: "a Text Block", stroke: "red" })
```

Fonte: Elaborada pelo autor

As novas funcionalidades de formatação adicionadas ao editor modificam as propriedades dos objetos das classes `Shape` e `TextBlock`, alterando assim sua aparência. Foram criados seis botões de formatação com as funções de alterar o tipo de fonte, aumentar o tamanho da fonte, alterar o estilo do texto para negrito, itálico e sublinhado, e para alterar a cor de preenchimento dos nós. Os botões de formatação de fonte alteram a propriedade *font* da `TextBlock`, que deve ser uma *string Cascading Style Sheets* (CSS). Já o botão que altera a cor do nó modifica a propriedade *fill* da `Shape`.

As funcionalidades de formatação foram implementadas em JavaScript no arquivo `toolbar.js`. Elas foram desenvolvidas de forma semelhante, utilizando eventos disparados pelos botões da barra de ferramenta.

Para realizar alteração de estilo da fonte e de cor do nó foram implementadas funções que modificam os atributos dos nós. Para cada botão da barra de ferramenta foi criada uma rotina que é executada quando o evento de clique é acionado.

A [Figura 12](#) mostra a rotina que altera a cor de preenchimento de um nó. Inicialmente é criada uma espera para o evento que será disparado pelo botão, no caso o evento *input*. Quando ele ocorre é chamada a função que altera a propriedade *fill* do objeto `Shape` de todos os nós selecionados, mudando assim a cor dos mesmos. As funcionalidades de estilo da fonte são implementadas com a mesma rotina, o que muda é a propriedade alterada, que passa a ser a *font* do objeto `TextBlock` de todos os nós selecionados.

Figura 12 – Rotina que altera a cor de preenchimento de um nó

```
colorButton = document.getElementById("color");
colorButton.addEventListener("input", function() {
    myDiagram.startTransaction("change color");
    var it = myDiagram.selection.iterator;
    while (it.next()) {
        var node = it.value;
        var shape = node.findObject("SHAPE");
        if (shape !== null) {
            shape.fill = colorButton.value;
        }
    }
    myDiagram.commitTransaction("change color");
});
```

Fonte: Elaborada pelo autor

4.2 Implementação no Moodle

Para implementar o serviço de edição de mapas no Moodle foi utilizado um *plugin* já existente na plataforma como modelo. Conforme visto no [Capítulo 3](#) o *plugin* escolhido para a implementação da funcionalidade de edição de mapas foi o de envio de tarefa. Todos os *plugins* deste tipo possuem uma estrutura de arquivo padrão, que será detalhada abaixo.

- `version.php`: este arquivo contém informações sobre a versão do *plugin*, é utilizado para que o Moodle instale e atualize o *plugins* corretamente.
- `settings.php`: este arquivo permite que se adicione opções personalizadas para a página configuração do *plugins*.
- `lang/en/submission_nomedoplugin.php`: este é o arquivo de linguagem, ele é usado para internacionalização do *plugin*.
- `db/access.php`: é utilizada para adicionar capacidades adicionais ao *plugin*. Este arquivo é opcional, não sendo necessário se o *plugin* não tiver capacidades adicionais.
- `db/upgrade.php`: este arquivo define a rotina de atualização do *plugin*.
- `db/install.xml`: este arquivo define as tabelas de banco de dados que o *plugin* vai utilizar.
- `db/install.php`: contém o código de instalação do *plugin*.
- `db/locallib.php`: é o arquivo mais importante, é ele que define todas as funcionalidades do *plugin*.

4.2.1 Desenvolvimento do plugin

O *plugin* utilizado como base para o desenvolvimento deste trabalho foi o de envio de texto online, nele o estudante tem a possibilidade responder uma tarefa por meio de um texto escrito diretamente na plataforma. Este projeto foi desenvolvido aplicando engenharia reversa a este *plugin*, buscando entender a sua funcionalidade e assim descobrir como modificá-lo para transformá-lo em um editor de mapas.

O *plugin* de envio de texto online pode ser dividido em duas partes, uma é o editor de textos onde o estudante realiza a sua tarefa e a outra é o sumário da atividade onde tanto o estudante quanto o avaliador tem acesso ao conteúdo da tarefa enviada. O *plugin* desenvolvido neste trabalho mantém esta estrutura, alterando o editor de textos por um de mapas conceituais e alterando o sumário de forma que o conteúdo apresentado nele passa a ser um Json do mapa ao invés de um texto.

As alterações feitas no *plugin* de envio de texto online se concentraram no arquivo `locallib.php`, que é o que determina o seu comportamento e funcionalidades. Dentro deste arquivo, as mudanças ocorreram nas funções `get_form_elements()`, que constrói o formulário de envio de tarefa, e no arquivo `save()`, que realiza a submissão do conteúdo criado pelo estudante.

A primeira parte do trabalho foi substituir o formulário de envio de texto pelo editor de mapas conceituais. Esta modificação foi realizada na função `get_form_elements()`. O código que realizava a inserção do editor de texto foi substituído por um outro, que insere um `iframe` contendo o editor de mapas do CMPaaS.

Assim, o *plugin* já apresentava o editor de mapas, porém ainda era necessário salvar o mapa criado no banco de dados do Moodle. Para realizar isto, o Json do editor contido no `iframe` precisava ser salvo em algum campo de formulário. Como solução, foi criada uma entrada de dados, oculta na página da tarefa, responsável por receber o Json do mapa criado no editor.

Quando o estudante aciona o botão de submissão da tarefa, o Json respectivo ao mapa que foi criado é salvo em um campo de formulário e então armazenado no banco de dados do Moodle.

Para finalizar, foi necessário criar uma forma de o editor de mapas carregar o mapa salvo pelo estudante, para caso ele tivesse interesse em editar um mapa já armazenado. A solução para isto foi realizar o caminho inverso que foi feito ao salvar o mapa criado. Ao carregar o formulário de edição de mapas o Json armazenado no banco de dados é carregado em um campo oculto e um código JavaScript se encarrega de obter o conteúdo dele e carregar no mapa.

A primeira parte do desenvolvimento do *plugin* foi concluída, o estudante conseguia criar um mapa no editor, salvar o seu conteúdo na plataforma e editar novamente caso

fosse necessário. Ainda havia a necessidade de permitir que o avaliador tivesse acesso ao conteúdo gerado pelo aluno. Além disso, falta a integração com o CMPaaS e utilização de seus recursos.

4.2.2 Integração com o CMPaaS

A integração do *plugin* com o CMPaaS é feita com a utilização de dois serviços oferecidos pela plataforma: o serviço de autenticação e o de persistência de mapas.

O serviço de autenticação é responsável por autorizar o acesso do usuário do Moodle ao CMPaaS. Ao acionar o botão de sincronização, é enviado uma mensagem requisitando autorização para que o usuário do Moodle tenha acesso ao serviço de mapas da plataforma. A API de autenticação do CMPaaS então responde concedendo acesso ao usuário do *plugin* ao repositório de mapas.

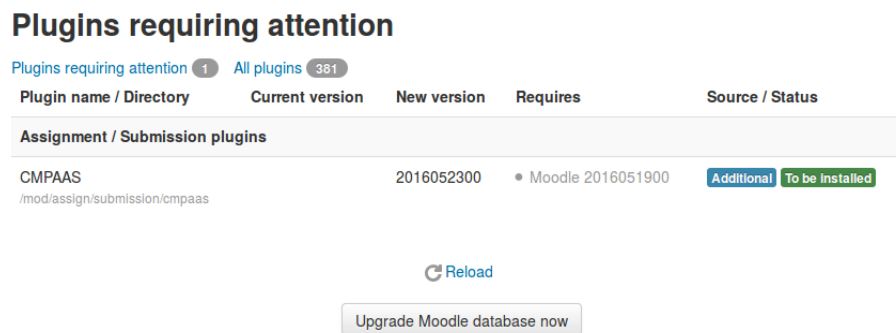
Após a rotina de autenticação ter sido realizada o editor de mapas envia um Json com os dados do diagrama a ser sincronizado. Então o serviço de mapas do CMPaaS recebe o Json e realiza a persistência do mesmo em sua base de dados.

5 Implementação e Testes

5.1 Como instalar

A pasta contendo os arquivos do *plugin* deve ser copiada para o diretório do Moodle `mod/assign/submission/`. Após copiar os arquivos para este local é necessário acessar o Moodle com um usuário com perfil de administrador. Assim que efetuar login irá aparecer uma mensagem de instalação de *plugin*, conforme a [Figura 13](#).

Figura 13 – Configurando a tarefa para utilizar o *plugin*



Fonte: Elaborada pelo autor

Ao clicar no botão de atualização do banco de dados do Moodle o *plugin* será instalado e poderá ser visto na lista de *plugins*, conforme [Figura 14](#).

Figura 14 – Lista de *plugins* instalados

Assignment / Submission plugins ⚙					
CMPAAS <small>assignsubmission_cmpaas</small>	2016052300	Enabled	Settings	Uninstall	Additional
Submission comments <small>assignsubmission_comments</small>	2016052300	Enabled		Uninstall	
File submissions <small>assignsubmission_file</small>	2016052300	Enabled	Settings	Uninstall	
OneNote submissions <small>assignsubmission_onenote</small>	3.1.0.1 2016062001	Enabled	Settings	Uninstall	Additional
Online text <small>assignsubmission_onlinetext</small>	2016052300	Enabled	Settings	Uninstall	

Fonte: Elaborada pelo autor

Depois do *plugin* ser instalado é necessário realizar a configuração da tarefa para utilizá-lo. Para que a tarefa utilize o *plugin* para submissão de dados é necessário selecioná-lo

na tela de configuração da atividade, conforme ilustrado na [Figura 15](#).

Figura 15 – Configurando a tarefa para utilizar o plugin

▼ Submission types

Submission types ☒ CMPAAS ☐ File submissions ☐ Online text ☐ OneNote submissions

Word limit ☐ Enable

Maximum number of uploaded files

Maximum submission size

Word limit ☐ Enable

Maximum number of uploaded OneNote pages

OneNote page size

Fonte: Elaborada pelo autor

Figura 16 – Editor de mapas

Tarefa

Arial

Sincronizar Mapa

GeJS evaluation
(c) 1998-2016 Northwoods Software
Not for distribution or production use
nwoods.com

Save changes Cancel

Fonte: Elaborada pelo autor

5.2 Como usar

A utilização do *plugin* é bem simples. Ao acessar uma tarefa o estudante deve clicar no botão de envio de tarefa e ao fazer isto o editor de mapas ira ser mostrado na tela,

como na [Figura 16](#). Após criar ou editar o mapa o aluno deve clicar no botão “Salvar alterações” e o Json será salvo no banco de dados da plataforma e aparecerá no sumário, conforme [Figura 17](#). Ao clicar no botão Sincronizar Mapa o diagrama será sincronizado na Nuvem da plataforma CMPaaS.

Figura 17 – Sumário da tarefa

Submission status	
Submission status	Submitted for grading
Grading status	Not graded
Due date	Monday, 17 October 2016, 1:00 AM
Time remaining	Assignment was submitted 24 days 11 hours late
Last modified	Thursday, 10 November 2016, 12:10 PM
CMAPAAS	<div><div></div><div><pre>{ "class": "go.GraphLinksModel", "nodeDataArray": [{ "text": "Capítulo 1", "key": "-1", "loc": "-210.72041083170965 -56.56608954917805", "text": "Mapas Conceituais", "key": "-239.37692552041673 7.193817273917574", "key": "-2", "text": "pg", "key": "-3", "loc": "105.16707740625003 -178.52745996875012", "text": "Introdução", "key": "-252.56325178125002 -118.689755934375005", "key": "-4", "text": "Capítulo 2", "key": "-55.547746890625035 -100.05013790625003", "key": "-5", "text": "O que são", "key": "-6", "loc": "277.00478203125004 182.299521796875", "text": "Origem", "key": "-238.72180319229176 244.2439551801676", "key": "-7", "text": "Quem criou", "key": "-165.34591573916674 153.26424210204252", "key": "-8", "text": "Importancia", "key": "-119.78266395791675 70.17831238329248", "key": "-9", "text": "EAD", "key": "-34.56441057812481 -19.644399468750038", "key": "-10", "text": "Moodle", "key": "-4.067960390624819 58.08114768749999", "key": "-11", "text": "Mapas e EAD", "key": "-75.0930233474983 56.74105204687493", "key": "-12", "text": "Capítulo 3", "key": "-57.65034029687496 -119.78162589062502", "key": "-13", "text": "Cmpaas", "key": "-108.89972418749988 35.35560053124999", "key": "-14", "text": "new node", "key": "-34.65408426083326 153.26424210204254", "key": "-15", "text": "new node", "key": "-16, 'loc': -60 224"], "linkDataArray": [{ "from": -3, "to": -1, "text": "" }, { "from": -3, "to": -4, "text": "" }, { "from": -3, "to": -5, "text": "" }, { "from": -2, "to": -6 }, { "from": -2, "to": -7 }, { "from": -1, "to": -2 }, { "from": -2, "to": -8 }, { "from": -2, "to": -9 }, { "from": -5, "to": -10 }, { "from": -10, "to": -11 }, { "from": -10, "to": -12 }, { "from": -3, "to": -13 }, { "from": -13, "to": -14 }, { "from": -8, "to": -15 }] }</pre></div></div>
Submission comments	► Comments (0)

Fonte: Elaborada pelo autor

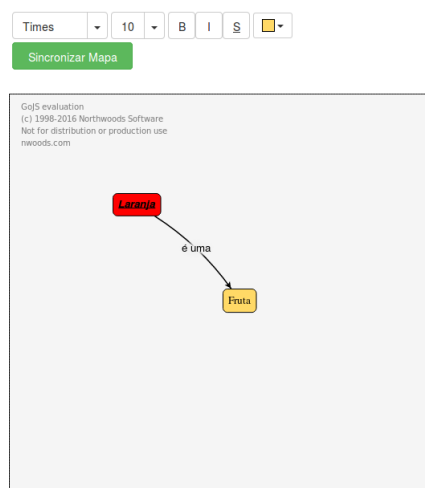
5.3 Provas de Conceitos

Nesta seção será apresentada a utilização de todas as funcionalidades oferecidas pelo *plugin*. Elas podem ser divididas em dois grupos: recursos para elaboração e edição dos mapas e persistência dos diagramas.

Os recursos para edição de mapas são as funções de formatação da aparência do diagrama disponíveis na barra de ferramentas. Elas permitem alterar a cor de preenchimento dos nós e o estilo da fonte de cada conceito. A [Figura 18](#) mostra um mapa no qual foi aplicado estas funções. Já a [Figura 19](#) apresenta o conteúdo dos menus *dropdown* de fonte e cor de preenchimento do nó.

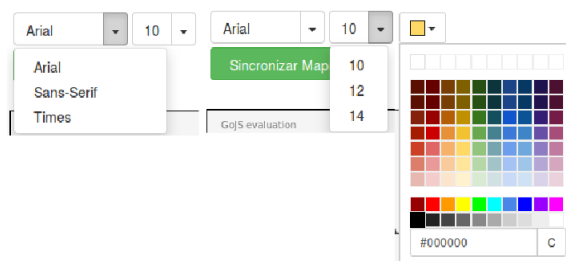
Já a persistência do diagrama pode ser dividida em duas etapas: a gravação do mapa no Moodle e a sincronização com o CMPaaS. A primeira ocorre quando o usuário clica no botão Salvar Alterações, que foi apresentado na [Figura 16](#). Já a sincronia com o CMPaaS ocorre quando o estudante utiliza o botão Sincronizar Mapa, que pode ser visto na cor verde na [Figura 18](#).

Figura 18 – Mapa que utiliza os recursos de formação



Fonte: Elaborada pelo autor

Figura 19 – Menus dropdown da barra de ferramentas



Fonte: Elaborada pelo autor

6 Considerações finais

6.1 Resultados

O objetivo deste trabalho foi a criação de uma ferramenta que permitisse a utilização de mapas conceituais em cursos gerenciados pela plataforma Moodle. Para realizar isto foi idealizado o desenvolvimento de um *plugin* de envio de tarefa que oferecesse a possibilidade do estudante responder uma atividade com a elaboração de um mapa conceitual. O *plugin* deveria prover também uma interface para um professor ou tutor visualizar o mapa criado pelo estudante e avaliar o mesmo. Adicionalmente ofereceria também uma interface de integração com o plataforma CMPaaS.

O protótipo desenvolvido neste trabalho alcançou o objetivo de oferecer uma ferramenta para o estudante responder uma tarefa com mapas conceituais. O *plugin* desenvolvido possui um editor de mapas que permite a elaboração de diagramas e modificação de sua aparência, além de realizar a persistência dos mapas criados.

A integração com o CMPaaS também foi implementada, assim o estudante tem a possibilidade de sincronizar o mapa criado no editor com o serviço de armazenamento na Nuvem oferecido pela plataforma. Já a interface para que o mapa seja visualizado para avaliação não foi implementada no protótipo, porém o *plugin* desenvolvido permite que este recurso seja adicionado em trabalhos futuros.

6.2 Conclusão

Neste trabalho foi apresentada a definição de mapas conceituais e a sua importância para a educação. Neste contexto foi discutido como eles podem ser utilizados no ensino e foi levantado quais são as ferramentas disponíveis para criação e manipulação de mapas. Neste levantamento foi identificada uma carência de aplicações computacionais que proporcionam suporte a criação de mapas conceituais e a necessidade do desenvolvimento de novas ferramentas com esta finalidade.

Foi pensado então no projeto de uma ferramenta que proporcionasse a aplicação de mapas conceituais na Educação a Distância. Escolhemos o Moodle como a plataforma onde seria realizada a implementação deste projeto. Ele foi escolhido por ser um gerenciador de cursos open-source amplamente utilizado em todo o mundo e por ser facilmente extensível através de inclusão de novos módulos.

Um *plugin* que permite a aplicação de mapas conceituais em tarefas dos cursos do Moodle foi então desenvolvido. Este *plugin* possibilita que os estudantes respondam

tarefas com a elaboração de mapas conceituais e oferece a possibilidade de sincronização do mapa na plataforma CMPaaS.

O *plugin* que foi criado neste trabalho trata-se de um protótipo, ou seja, não está totalmente funcional e carece de melhorias. No entanto serve como base para a criação de uma ferramenta que pode ser adicionada a lista de repositórios de *plugin* do Moodle, ampliando assim a possibilidade do uso de mapas conceituais no ensino.

6.3 Trabalhos futuros

O protótipo desenvolvido neste trabalho teve foco na interface para o estudante criar mapas conceituais, não possuindo o recurso de apresentar o mapa elaborado nele para um avaliador. Assim, uma proposta de trabalho futuro é a criação de uma forma de o avaliador visualizar o mapa submetido pelo estudante.

Ademais é sugerido o desenvolvimento de trabalhos que visam a melhoria da experiência do usuário que utiliza o editor de mapas. Visto que este trabalho não teve como foco a criação de uma interface otimizada.

Referências

ALMEIDA, M. E. B. de. Tecnologia e educação a distância: Abordagens e contribuições dos ambientes digitais e interativos de aprendizagem. *Revista Brasileira de Educação a Distância*, p. 6, 2010.

AUSUBEL, D. P. The psychology of meaningful verbal learning. Grune and Stratton, New York, 1963.

BRASIL. Decreto 5.622, de 19 de dezembro de 2005. Regulamenta o artigo 80 da Lei no 9.394, de 20 de dezembro de 1996, que estabelece as diretrizes e bases da educação nacional. 2005. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/d5622.htm>. Acesso em: 08/11/2016.

CHAVES, E. Tecnologia na educação, ensino a distância, e aprendizagem mediada pela tecnologia: conceituação básica. *Revista Educação da Faculdade de Educação da Pontifícia Universidade Católica de Campinas*, v. 3, n. 7, 1999.

CROCKFORD, D. The application/json media type for javascript object notation (json). 2006.

DOUGIAMAS, M.; TAYLOR, P. Moodle: Using Learning Communities to Create an Open Source Course Management System. *Research.Moodle.Net*, p. 1–16, 2003. Disponível em: <http://research.moodle.net/pluginfile.php/15/mod_data/content/1121/Moodle-Dougiamas-2003.pdf>.

DUTRA, Í. M.; FAGUNDES, L. d. C.; CAÑAS, A. J. Um ambiente integrado para apoiar a avaliação da aprendizagem baseado em mapas conceituais. *XIII Simpósio Brasileiro de Informática na Educação - SBIE - Unisinos*, p. 49–59, 2002.

FLANAGAN, D. *JavaScript: the definitive guide*. [S.l.]: "O'Reilly Media, Inc.", 2006.

FRANCISCATO, F. T. et al. Avaliação dos Ambientes Virtuais de Aprendizagem Moodle, TelEduc e Tidia - Ae: um estudo comparativo. *Novas Tecnologias na Educação*, v. 6, n. 2, p. 5–10, 2008.

MINETTO, E. L. Frameworks para desenvolvimento em php. *São Paulo: Novatec*, 2007.

MOODLE. *Moodle Plugins*. 2016. Disponível em: <https://moodle.org/plugins/qtype_conceptmap>. Acesso em: 09/12/2016.

MOODLE. *Moodle Statistics*. 2016. Disponível em: <<https://moodle.net/stats/>>. Acesso em: 16/11/2016.

NOVAK, J. D. Results and Implications of a 12-Year Longitudinal Study of Science Concept Learning. *Research in Science Education*, v. 35, n. 1, p. 23–40, mar 2005. ISSN 0157-244X. Disponível em: <<http://link.springer.com/10.1007/s1165-004-3431-4>>.

NOVAK, J. D.; CAÑAS, A. J. The origins of the concept mapping tool and the continuing evolution of the tool. *Information Visualization*, v. 5, n. 3, p. 175–184, 2006. ISSN 1473-8716. Disponível em: <<http://ivi.sagepub.com/lookup/doi/10.1057/palgrave.ivs.9500126>>.

PERIN, W.; CURY, D. Uma Plataforma de Serviços para Mapas Conceituais. n. Cbie, p. 230, 2016. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/6703>>.

PERIN, W. d. A. *iMap – UM MECANISMO DE INFERÊNCIA PARA MAPAS CONCEITUAIS*. 120 p. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2014.

RICHARDS, G. et al. An analysis of the dynamic behavior of JavaScript programs. *Proceedings of the 2010 ACM SIGPLAN conference on Programming language design and implementation*, p. 1, 2010. ISSN 03621340.