

## Modelo de um Sistema Multiagente para Recomendação de Atividades de Programação na Plataforma Moodle

Rosemary Pessoa Borges<sup>1,2</sup>, Carla Katarina de Monteiro Marques<sup>1,2</sup>, Rommel Wladimir de Lima<sup>1</sup>

<sup>1</sup>Departamento de Informática. Universidade do Estado do Rio Grande do Norte (UERN) – Mossoró, RN – Brasil

<sup>2</sup>Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte (IFRN) – Natal, RN, Brasil

{marybezerraleao, carla.katarina, rommel.lima}@gmail.com

**Abstract.** *This article presents a context sensitive tool that aims to guide the creation of activities by the teacher, making the recommendation based on the content addressed and the interactions recorded in the Virtual Learning Environment. It is hoped that this tool will support, in the first moment, the activities related to teaching, allowing the teacher to easily identify the recurring difficulties. In the second moment, as a result, one expects to obtain a better performance relative to the learning, since the students will start to develop directed and specific activities, orientated according to the needs previously identified.*

**Resumo.** *O presente trabalho apresenta uma ferramenta sensível ao contexto que objetiva orientar a criação das atividades por parte do professor, realizando a recomendação com base no conteúdo abordado e nas interações registradas no Ambiente Virtual de Aprendizagem (AVA). Espera-se que essa ferramenta apoie, no primeiro momento, as atividades relativas ao ensino, permitindo que o professor possa facilmente identificar as dificuldades mais recorrentes. No segundo momento, como resultado, espera-se obter um melhor rendimento relativo à aprendizagem, uma vez que os alunos passarão a desenvolver atividades direcionadas e específicas, norteadas de acordo com as necessidades anteriormente identificadas.*

### 1. Introdução

O processo de seleção e avaliação de atividades por parte do docente pode ser considerado como um dos fatores determinantes para o bom desempenho da turma [Perrenoud 2003]. Para a seleção das atividades, levar em conta as necessidades mais recorrentes apresentadas pelos alunos, mesmo que cada aluno tenha seu ritmo de desenvolvimento, pode determinar o sucesso ou o fracasso de uma grande parcela da turma. Nesse contexto, questiona-se como é possível identificar os conteúdos que precisam ser mais explorados e exercitados pelo professor, em tempo hábil e com bom

índice de precisão, para um melhor rendimento da turma?

Em se tratando de disciplinas que abordem linguagens de programação, diversas pesquisas, como apresentado por Divino (2015), Maciel et al (2012) e Chaves et al (2013), revelam que os alunos tendem a ter um maior índice de reprovação, o que leva a desmotivação para continuar no curso ou até mesmo na área. Já com relação a avaliação das atividades, os docentes têm o trabalho intensificado uma vez que o código desenvolvido pelos alunos é um elemento complexo de ser analisado, exigindo muito esforço e tempo do docente, tornando o processo demorado. Nesse sentido, uma ferramenta de gestão de atividade de código fonte, que auxilie na avaliação das respostas dos alunos tem se apresentado uma solução [Chaves 2014].

Toda a problemática apresentada, agrava-se principalmente quando o número de alunos envolvidos na turma é grande, isso porque, guiar esses alunos considerando que possuem perfil heterogêneo e diferentes níveis de conhecimento, dificulta ainda mais o trabalho docente tornando o processo de mediação ainda mais complexo [Marinho et al. 2016]. Diante desse contexto, esse trabalho apresenta a concepção de uma ferramenta sensível ao contexto que tem por objetivo orientar a criação das atividades por parte do professor, realizando a recomendação com base no conteúdo abordado e nas interações registradas no AVA, integrado a um módulo do Moodle responsável pelo gerenciamento de atividades relativas à criação de código fonte.

A estrutura do artigo é descrita em 6 tópicos. Além da introdução já apresentada, o tópico 2 apresenta o conceito de sistema de recomendação. O tópico 3 explana sobre sistema multiagente. O tópico 4 apresenta o Módulo de Integração com os Juizes Online. O tópico 5 apresenta a modelagem do sistema multiagente, bem como a metodologia, ferramentas e arquitetura. E, por fim, o tópico 6 conclui o trabalho com as considerações finais.

## **2. Sistema de Recomendação**

Não é de hoje que sistemas de recomendação são desenvolvidos a fim de auxiliar no aumento da capacidade e qualidade no processo de indicação, fato já bastante utilizado nas relações sociais entre seres humanos [Resnick and Varian 1997]. Na internet esse tipo de sistema cada vez mais tem se destacado. Hoje é comum um internauta receber indicações de conteúdos, ou produtos, semelhantes aos já pesquisados por ele em outro momento, mesmo que não tenha solicitado. De acordo com Reategui and Lorenzatti (2005) essa é uma das principais funcionalidade proporcionadas pelos sistemas de recomendação.

Diversas técnicas podem ser utilizadas no processo de recomendação, uma das mais antigas é a conhecida como filtragem colaborativa, onde pessoas são classificadas de acordo com comportamentos semelhantes, logo, pressupõe que suas preferências também serão similares [Herlocker et al. 2000]. Para o desenvolvimento da funcionalidade de recomendação, segundo Barcellos et al. (2007), é necessário que o processo de recomendação seja dividido em quatro etapas: identificação do usuário,

coleta de informações, estratégias de recomendação e visualização das recomendações.

No âmbito dessa pesquisa, a recomendação é realizada para identificar os conteúdos que mais geraram interações no AVA, através da consulta direta a base de dados do AVA, dispensando assim qualquer outro tipo de interação do usuário. Como sugere-se na pesquisa de Reategui and Lorenzatti (2005), onde é apresentado um sistema de recomendação de conteúdo. No cenário abordado nessa pesquisa, a existência de recursos no AVA que fornecem pistas sobre as dificuldades que a turma está encontrando durante os estudos são ideais para o docente, sendo possível assim a implementação de uma estrutura simplificada.

### **3. Sistema Multiagente**

Na Inteligência Artificial é frequente o uso do conceito de Agente Inteligente (AI). Segundo Russel e Norvig (2013), um AI é uma entidade autônoma capaz de interagir com o ambiente, com outros agentes, cooperando ou ainda competindo entre si e com inteligência para tomar decisões sem a necessidade de um outro sistema ou da interferência de seres humanos. Ainda de acordo com Russel e Norvig (2013), um AI capta as percepções oriundas de um ambiente por meio de sensores e age nesse ambiente por intermédio de atuadores.

Os sistemas categorizados como Sistemas Multiagentes (SMA), segundo Reis (2003), são formados por agentes que possuem desempenho autônomo, porém, têm ações de caráter colaborativo. Ou seja, eles se ajudam, objetivando alcançar uma meta em comum. Na atualidade, muitos são os exemplos de competições que utilizam o conceito de SMA, como por exemplo, copas de futebol utilizando robôs, competição de venda de pacotes de viagens, dentre outras competições.

De acordo com Wooldridge (2009), um SMA caracteriza-se por ser um sistema onde vários agentes atuam em conjunto sobre um ambiente na busca da resolução de um problema. Um SMA conta com elementos que possuem diferentes capacidades de percepção e ação no mundo. de acordo com Reis (2003), cada agente pertence a uma organização, onde podem ocorrer interações distintas, e estes agentes detêm uma esfera de influência diferente sobre o ambiente ao qual interage.

Para Dorça (2002), os sistemas educacionais desenvolvidos utilizando-se características de SMA, tendem a ser eficazes, pois tendem a ir de encontro a natureza da solução dos problemas de ensino-aprendizagem, que no geral são resolvidos de forma colaborativa. Mas para que isso ocorra, ainda segundo o mesmo autor, é necessário delimitar bem uma arquitetura para a aplicação, os agentes que implementam os módulos dessa arquitetura, bem como as suas interações.

### **4. Módulo de Integração com os Juizes Online**

O Módulo de Integração com os Juizes Online (MOJO), desenvolvido por Chaves (2014), tem como objetivo auxiliar o docente de forma automatizada nos processos de

elaboração (por possuir uma base de questões previamente criada), avaliação e fornecimento de *feedback* das questões respondidas aos alunos. Para que isso ocorra, a ferramenta faz uma integração entre os sistemas de Juízes Online ao Ambiente Virtual de Aprendizagem Moodle.

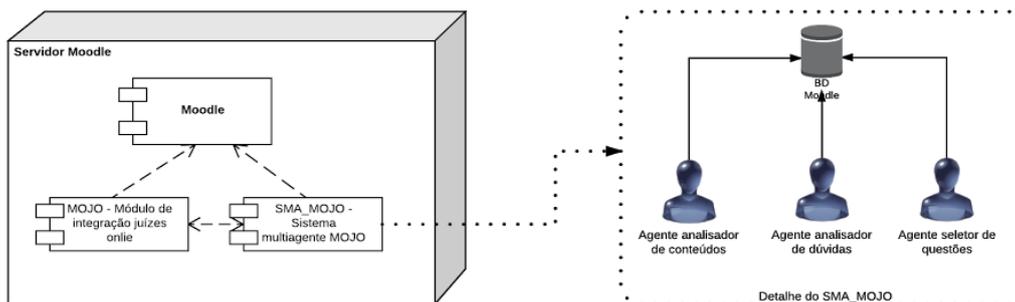
Chaves (2013) vislumbrou, ao implementar o MOJO, a integração de dois ambientes, almejando que um venha a suprir e completar ou outro, de forma que sendo integrado ao Moodle, o MOJO oferece ao professor e ao aluno, uma facilitação no *feedback* com relação ao código que foi desenvolvido, possíveis erros na compilação, dentre outros.

A integração Moodle-MOJO além de proporcionar a disponibilização de atividades através da plataforma Moodle e correção por juízes online, fornece base para um leque de funcionalidades que venham a automatizar o trabalho docente. Surge aí, a possibilidade da implementação, de forma integrada, de um sistema multiagente sensível ao contexto.

## 5. Sistema Multiagente integrado ao MOJO (SMA-MOJO)

Neste trabalho, como já discorrido anteriormente, propõe-se um sistema de recomendação de atividades baseado em um sistema multiagente orientado ao contexto do educando.

Para isso, um sistema multiagente está sendo desenvolvido, contando com agentes específicos para: (I) identificar os conteúdos disponibilizados em forma de recursos; (II) identificar as principais dúvidas citadas nos canais de comunicação (fóruns e chats); (III) selecionar questões do MOJO de acordo com as informações coletadas nos agentes citados anteriormente e (IV) recomendar essas questões ao professor no Moodle. A nova arquitetura pode ser verificada na Figura 1, onde também são apresentados os agentes descritos.



**Figura 1 - Integração entre Moodle, MOJO e o SMA, destacando-se os agentes envolvidos.**

Definir um modelo sensível ao contexto para a recomendação de atividades do módulo MOJO, dentro do Moodle, exige que sejam implementados alguns objetivos em específico. São eles: (i) definição de metadados que permitam categorizar cada questão e cada conteúdo programático disponibilizado na base de dados, sendo necessário assim

alteração na base de dados do Moodle; (ii) identificação dos assuntos recorrentes com base nas informações obtidas observando o contexto ao qual o aluno está inserido (através da análise dos recursos disponibilizados pelo professor e canais de dúvidas); (iii) seleção de questões a serem direcionadas com base nas informações obtidas no AVA e nos metadados das questões.

### 5.1 Metodologia

Se faz necessário utilizar-se de um servidor Moodle para cumprir o objetivo proposto relativo ao desenvolvimento do referido sistema multiagente, bem como utilizar-se do módulo MOJO (integrado ao Moodle). O ambiente de desenvolvimento adotado para o SMA-MOJO, utiliza a linguagem de programação JAVA 1.8 e o *framework* JADE.

Para atingir o objetivo do SMA, todos os agentes devem trocar informações e acessar o banco de dados MySQL para perceber o que está acontecendo no Moodle e sugerir as questões ao professor através da interface do MOJO. De forma que, os agentes da aplicação fazem a verificação das condições nas quais os alunos estão inseridos no AVA, assim sendo, para isso, devem levar em consideração os seguintes aspectos: (I) o assunto que está sendo desenvolvido; (II) as temáticas abordadas nos fóruns e chats; e (III) características das questões, com base nas peculiaridades de cada turma e de cada disciplina. A análise do contexto, para cada um dos aspectos apresentados acontecerá de forma diferente, e, para cada uma das características um agente diferente será utilizado.

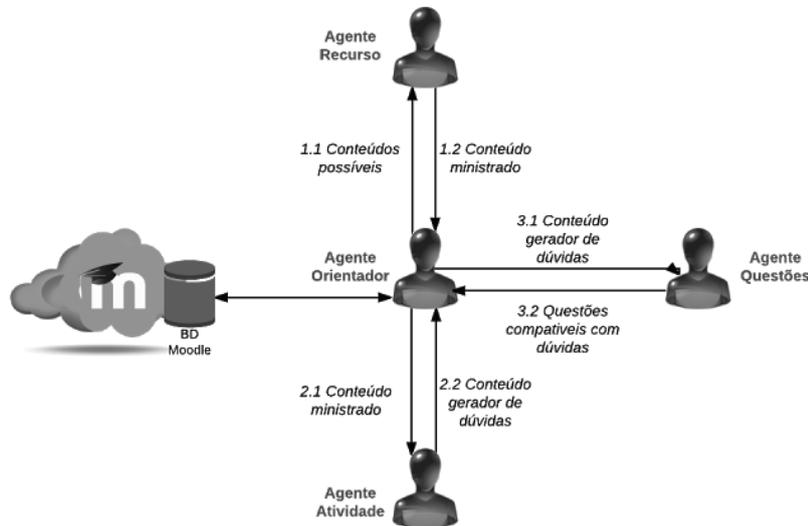
### 5.2 Arquitetura

Para o desenvolvimento de um Sistema Multiagente, como já mencionado anteriormente, é necessário que seus comportamentos sejam bem definidos, para que a colaboração entre eles flua e possam realizar da melhor forma possível a tarefa para a qual foram designados. Para esse SMA, em específico, os comportamentos configurados em cada um dos agentes inteligentes seguiram as descrições:

- **Agente analisador de conteúdos**, responsável por (i) recuperar da base de dados os conteúdos desenvolvidos em determinada sessão da disciplina e (II) analisar os dados recuperados de forma a identificar conceitos chaves da disciplina na sessão atual.
- **Agente analisador de dúvidas**, responsável por (i) recuperar da base de dados as dúvidas registradas pelos alunos nos fóruns, em determinada sessão da disciplina, (ii) recuperar da base de dados as dúvidas registradas pelos alunos nos chats, em determinada sessão da disciplina e (iii) analisar os dados recuperados de forma a identificar quais conceitos chaves trabalhados na disciplina na sessão atual, foram citados pelos alunos.
- **Agente seletor de questões**, responsável por (i) realizar a seleção das atividades que se enquadrem nos conceitos chaves identificados pelos agentes descritos anteriormente e (ii) reportar ao MOJO as questões, priorizando aquelas que mais se

aproximam das necessidades observadas.

Na Figura 2, é possível observar a arquitetura de funcionamento do SMA para recomendação de atividades.



**Figura 2 - Arquitetura de funcionamento do SMA para recomendação de atividades.**

Esse SMA, por possuir a característica de aprendizagem, permite adaptabilidade às mudanças do ambiente. Assim, em uma dada fração de tempo, que está também em constante reavaliação, o SMA deve realizar a análise dos dados para fornecer base para a funcionalidade de recomendação.

## 6. Considerações finais

Como foi possível observar, o SMA-MOJO traz uma proposta de recomendação de atividades, sem que o professor precise concentrar esforços para identificar as principais dificuldades relatadas pelos alunos nos meios de comunicação da plataforma Moodle, permitindo assim que o docente possa se concentrar em outras atividades relativas ao ensino. Vale ressaltar, que o contexto da disciplina pode ser considerada como um todo ou em partes. Isso ocorre, pois, o professor tende a optar por trabalhar sua disciplina fragmenta em sessões no Moodle, dividindo por assunto. Se assim o fizer, o SMA observará cada sessão individualmente evitando o surgimento de falsos positivos, por exemplo, no caso de uma sessão anterior que tenha abordado assunto muito polêmico, gerando inúmeras dúvidas, enquanto a sessão atual não. Vislumbra-se ainda que a utilização de ontologias no desenvolvimento da ferramenta proporcionaria maior relação entre os termos, melhor organização, e facilitaria a implementação. Fato que pode ser melhor explorado em trabalhos futuros.

## Referências

Barcellos, C. D. et al. (2007) “Sistema de Recomendação Acadêmico para Apoio a

- Aprendizagem”. In *Novas Tecnologias na Educação*, Rs, v. 5, n. 2.
- Chaves, J. O. M. et al. (2013) “Uma Ferramenta Baseada em Juizes Online para Apoio às Atividades de Programação de Computadores no Moodle”. In *Revista Novas Tecnologias na Educação*, [s.l.], v. 11, n. 3, [s.p.].
- Chaves, J. O. M. (2014) “Uma ferramenta de apoio ao processo de ensino-aprendizagem em disciplinas de programação de computadores por meio da integração dos Juizes Online ao Moodle”. 80 f. Dissertação (Mestrado em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semiárido. Mossoró, RN.
- Divino, A. (2015) “Algo+ - Um app para o auxílio na aprendizagem de programação”. In *Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015)*, [s.l.], p.372-376, 26 out. 2015.
- Dorça, F. et al. (2002) “Um Sistema Inteligente Multiagente para Educação à Distância”. In: *Congresso da Sociedade Brasileira de Computação.*, 2002, Florianópolis. Anais... . Florianópolis: CSBC.
- Herlocker, J., Konstan, J. e Riedl, J. (2000). “Explaining Collaborative Filtering Recommendations”. In *proceedings of ACM 2000 Conference on Computer Supported Cooperative Work*, December 2-6.
- Maciel, D. L. et al. (2012) “Análise de similaridade de códigos-fonte como estratégia para o acompanhamento de atividades de laboratório de programação”. *Revista Novas Tecnologias na Educação*. [s.l.], v. 10, n. 03, [s.p.].
- Marinho, C. S. S. et al. (2016) “Experiências no Uso da Metodologia Coding Dojo nas Disciplinas Básicas de Programação de Computadores em um Curso Interdisciplinar do Ensino Superior”. In: *Workshop do Congresso Brasileiro de Informática na Educação*, 5., 2016, Uberlândia. Anais... . Uberlândia: CBIE, 2016. p. 1097 - 1106.
- Perrenoud, P. (2003) “Sucesso na escola: só o currículo, nada mais que o currículo”. In *Cadernos de Pesquisa*, nº 119.
- Reategui, E. and Lorenzatti, A. (2005) “Um Assistente Virtual para Resolução de Dúvidas e Recomendação de Conteúdo”. *V Encontro Nacional de Inteligência Artificial (ENIA 2005)*. São Leopoldo, RS, Julho.
- Reis, L. P. (2003) “Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico”. 2003. 487 f. Tese (Doutorado) - Curso de Engenharia Electrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, 2003. Cap. 10. Disponível em: <[https://web.fe.up.pt/~niadr/PUBLICATIONS/thesis\\_PhD/PhD\\_LuisPauloReis.pdf](https://web.fe.up.pt/~niadr/PUBLICATIONS/thesis_PhD/PhD_LuisPauloReis.pdf)>
- Resnick, P. and Varian, H. R. (1997) “Recommender Systems”. *Communications of the ACM*, New York, v.40, n.3, pp. 55-58.
- Russel, S. and Norvig, P. (2013) “Inteligência Artificial: Uma abordagem moderna”. 3a. ed. Rio de Janeiro: Elsevier Editora.
- Wooldridge, M. (2009) “An Introduction to Multiagent Systems”. Ed. Wiley. England.