



**UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE
UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**



JOSÉ OSVALDO MESQUITA CHAVES

**Uma ferramenta de apoio ao processo de ensino-aprendizagem
em disciplinas de programação de computadores por meio da
integração dos Juízes Online ao Moodle**

MOSSORÓ – RN

2014

JOSÉ OSVALDO MESQUITA CHAVES

**Uma ferramenta de apoio ao processo de ensino-aprendizagem
em disciplinas de programação de computadores por meio da
integração dos Juizes Online ao Moodle**

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semi-Árido – como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Orientadora: Profa. Dra. Angélica Félix de Castro – UFERSA.

Coorientador: Prof. Dr. Rommel Wladimir de Lima – UERN.

MOSSORÓ – RN

2014

**Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.**

Chaves, José Osvaldo Mesquita

Uma ferramenta de apoio ao processo de ensino-aprendizagem em disciplinas de programação de computadores por meio da integração dos Juízes Online ao Moodle. / José Osvaldo Mesquita Chaves - Mossoró, RN, 2014.

80 f.

Orientador(a): Profa. Dra. Angélica Félix de Castro.

Dissertação (Mestrado em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-Árido. Programa de Pós-Graduação em Ciência da Computação.

1. Disciplinas de Programação. 2. Juízes Online. 3. Moodle. 4. Ensino - Aprendizagem. I. Castro, Angélica Félix de. II. Universidade do Estado do Rio Grande do Norte. III. Título.

UERN/BC

CDD 004


JOSÉ OSVALDO MESQUITA CHAVES

**Uma ferramenta de apoio ao processo de ensino-aprendizagem
em disciplinas de programação de computadores por meio da
integração dos Juizes Online ao Moodle**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de Mestre em Ciência da Computação.


APROVADA EM: 24 / 02 / 2024.

BANCA EXAMINADORA



Profa. Dra. Angélica Félix de Castro – UFERSA

Presidente



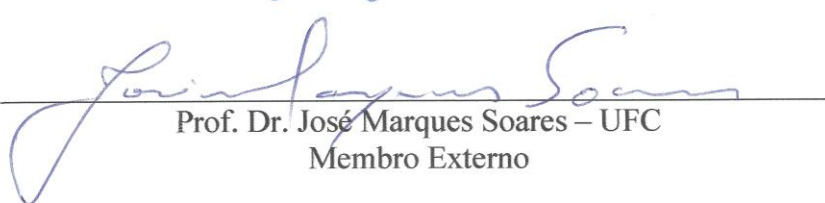
Prof. Dr. Rommel Wladimir de Lima – UERN

Membro Interno



Prof. Dr. Marcos Evandro Cintra – UFERSA

Membro Interno



Prof. Dr. José Marques Soares – UFC

Membro Externo

Dedico este trabalho a meu companheiro e amigo Marcos Vinicius de Andrade, por me fazer acreditar no sonho e por me acompanhar nesta árdua aventura na busca pelo conhecimento e crescimento pessoal.

AGRADECIMENTOS

Primeiramente a Deus, sem o qual eu não estaria aqui. Obrigado Papai, por tudo!

A minha família, em especial àqueles que mesmo sem completo conhecimento do que se tratava o mestrado, apoiaram e colaboraram pra que esse objetivo fosse alcançado. Muito obrigado a minha Mãe Socorro, minhas irmãs Suelen e Suiane, minhas cunhadas Sônia e Zildete, e a Débora, pelo mutirão realizado para equipar uma moradia provisória enquanto eu estava longe de casa e pelo, sempre caloroso, aconchego a cada retorno ao lar.

A Professora Dra. Angélica Félix de Castro, minha orientadora, e ao Professor Dr. Rommel Wladimir de Lima, meu coorientador, pelo apoio, paciência, direcionamento, vibração a cada conquista e confiança durante todas as fases da pesquisa. Sem o apoio, orientação e motivação de vocês eu não chegaria tão longe. #Foco #Fé #FacaNaCaveira!

A minha querida amiga (e ex-professora) Patrícia Jardim, que mesmo distante, sempre foi um apoio constante, principalmente nos momentos mais difíceis e angustiantes. Essa é uma amizade que pretendo carregar por toda minha vida.

Ao Professor Msc. Sebastião Alves Filho, que foi fundamental na idealização da ferramenta desenvolvida neste trabalho.

Aos professores e colegas do PPgCC UERN/UFERSA, pelo compartilhamento de conhecimento, incentivo e amizade. Meu horizonte se expandiu com tudo que foi dito e feito em cada momento no decorrer destes dois anos.

A Rosita Rodrigues, secretária do PPgCC-UERN, que ao longo destes dois anos demonstrou cuidado e preocupação cada vez que precisei ir a sua sala tratar dos mais diversos assuntos relacionados ao mestrado. Muito obrigado por todo o carinho, cuidado e atenção.

Aos companheiros de laboratório, Marcos Vinicius de Andrade Lima e Karl Hansmuller Alelaf Ferreira, pelo compartilhamento de conhecimento, apoio nas horas de enfrentar o “bucha” da vez, parceria e suporte na realização da pesquisa. Somos guerreiros!

A CAPES e FAPERN, que viabilizaram a realização deste trabalho por meio da concessão de bolsa e compra de equipamento, respectivamente.

E por fim, a todos que, direta ou indiretamente, contribuíram para a minha formação. Mesmo aqueles que não estão sendo citados aqui, por medo de correr o risco de esquecer algum nome, o meu MUITO OBRIGADO!

“Os nossos maiores problemas não estão nos obstáculos do caminho, mas na escolha da direção errada.” (Augusto Cury)

RESUMO

Disciplinas que envolvem programação são essenciais em cursos de computação, e exigem um grande envolvimento por parte do professor que, muitas vezes sobrecarregado por suas tarefas, não consegue realizar o devido acompanhamento de seus alunos. A espera do aluno para tirar dúvidas ou apresentar seus resultados são elementos que acabam contribuindo para a desmotivação, desinteresse, e em alguns casos até desistência do aluno, colaborando com o aumento no índice de evasão do curso. Isso pode ser ainda mais grave quando a modalidade de ensino é à distância, onde o contato pessoal entre professor e aluno é mínimo ou inexistente. Com a sobrecarga relacionada à tarefa de elaborar, submeter, avaliar e fornecer o devido feedback das atividades a seus alunos, o professor tem como alternativa a utilização de ferramentas automatizadas que sirvam de auxílio para a realização de suas tarefas. Diante deste cenário, e visando contribuir com a melhoria do ensino e aprendizagem em disciplinas de programação, este trabalho apresenta uma ferramenta para a automatização do processo de elaboração, submissão e avaliação de atividades práticas de programação de maneira a reduzir a sobrecarga de tarefas do professor. A ferramenta se fundamenta no processo de avaliação automática realizado pelos sistemas de juízes online, processo semelhante ao adotado em maratonas de programação. A ferramenta apresentada é integrada à plataforma Moodle como um novo módulo para adição de funcionalidades, podendo apoiar o ensino presencial e semipresencial ou à distância.

Palavras-Chave: Disciplinas de Programação, Juízes Online, Moodle, Ensino, Aprendizagem.

ABSTRACT

Disciplines which involve programming are essential in computing courses, and demand a high involvement of the teachers who, most of times, are overloaded by their tasks, so that cannot properly monitor their students. The long student waiting period to clarify doubts or present their results are elements that contribute to discouragement, disinterest, and, in some cases, even the student giving up, collaborating to the increasing of the course dropout rate. This can be even more serious when regarding to distance education modality, where the personal contact between teacher and student is minimal or nonexistent. With the overload related to the tasks of preparing, submitting, evaluating and providing feedback of the activities to their students, the teachers have as an alternative, the using of automated tools that serves as an aid in performing their tasks. Before this scenario, and aiming to contribute to the improvement of programming courses teaching and learning, this paper presents a tool for automating the process of preparation, submission and evaluation of practical programming activities in order to reduce the overload of teachers' tasks. The tool bases on automatic assessment process conducted by online judges systems, a similar process used in programming marathons. The presented tool is integrated into the Moodle platform as a new module for adding functionalities and can support teaching in cases of face-to-face classroom, semi-distance or distance modality.

Keywords: Programming Disciplines, Online Judges, Moodle, Teaching, Learning.

LISTA DE TABELAS

Tabela 1 – Níveis de dificuldade para atribuição a questão	32
Tabela 2 – Resultados da submissão	34
Tabela 3 – Tabela de publicações.....	67
Tabela 4 – Primeiras impressões da ferramenta	68

LISTA DE FIGURAS

Figura 1 – Arquitetura final de integração dos ambientes.....	29
Figura 2 – Arquitetura interna do MOJO	30
Figura 3 – Ranking das linguagens de programação mais populares.....	35
Figura 4 – Trecho de código exemplificando a utilização da biblioteca cURL	36
Figura 5 – Conjunto de Tabelas do MOJO.....	37
Figura 6 – Disciplinas de programação em um curso no Moodle.....	40
Figura 7 – Editando uma disciplina no Moodle	41
Figura 8 – Adicionando uma nova atividade à disciplina	41
Figura 9 – Fluxo do processo ESA com o MOJO.....	43
Figura 10 – Adicionando uma nova atividade do tipo MOJO.....	44
Figura 11 – Formulário principal do MOJO.....	45
Figura 12 – Total de questões e opção para adicionar novas	46
Figura 13 – Tela para adição de novas questões ao MOJO.....	47
Figura 14 – Campos do formulário preenchidos com dados da questão selecionada	48
Figura 15 – Tela principal da disciplina com atividades adicionadas	49
Figura 16 – Tela de visualização da questão pelo professor	49
Figura 17 – Tela principal da disciplina na visão do aluno	50
Figura 18 – Tela de visualização da questão com destaque nas opções do aluno.....	51
Figura 19 – Tela de submissão de questão pelo aluno	52
Figura 20 – Popup exibindo a descrição da questão na tela de submissão.....	53
Figura 21 – Solução de questão pronta para ser enviada.....	54
Figura 22 – Tela de resultados de submissão do aluno	55
Figura 23 – Popup exibindo código-fonte do aluno	55
Figura 24 – Resultado do aluno atualizado	56

Figura 25 – Questão com pendência no resultado	56
Figura 26 – Atividades recentes do aluno	57
Figura 27 – Envio de código para avaliação e recebimento do resultado	58
Figura 28 – Região MOJO Resultados	59
Figura 29 – Organograma das opções de visualização dos resultados pelo professor	60
Figura 30 – Tela de visualização dos resultados pelo professor.....	61
Figura 31 – Opção de envio de feedback pelo professor.....	62
Figura 32 – Tela para envio de feedback pelo professor	62
Figura 33 – Popup exibindo feedback enviado pelo professor.....	63
Figura 34 – Instalação de novo módulo ao Moodle	77
Figura 35 – Módulos de atividades instalados.....	78

LISTA DE SIGLAS

API – Application Programming Interface;

AVA – Ambiente Virtual de Aprendizagem;

CURL – Cliente URL;

ESA – Elaboração, Submissão e Avaliação;

LORDI – Laboratório de Redes e Sistemas Distribuídos;

MEC – Ministério da Educação;

MOCA – Módulo de Carga e Atualização;

MOJO – Módulo de Integração com os Juízes Online;

MOODLE – Modular Object-Oriented Dynamic Learning Environment;

MOP – Módulo Principal;

PHP – Hypertext Preprocessor

SGBD – Sistema Gerenciador de Banco de Dados;

SPOJ – Sphere Online Judge;

TIC – Tecnologias da Informação e Comunicação;

VPL – Virtual Programming Lab;

WS – Web Services.

SUMÁRIO

1. INTRODUÇÃO	16
1.1 PROBLEMÁTICA E JUSTIFICATIVA	18
1.2 OBJETIVOS	19
1.3 ORGANIZAÇÃO DO TEXTO	19
2. AMBIENTES ENVOLVIDOS E TRABALHOS RELACIONADOS	21
2.1 MOODLE	21
2.1.1 Potencial para Extensão	22
2.2 JUÍZES <i>ONLINE</i>	22
2.2.1 SPOJ Brasil	23
2.3 TRABALHOS RELACIONADOS	24
2.3.1 Ferramentas não Integradas ao Moodle	24
2.3.2 Ferramentas Integradas ao Moodle	25
2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO	26
3. UM AMBIENTE INTEGRADO PARA AUXILIAR PRÁTICAS DE PROGRAMAÇÃO DE COMPUTADORES	27
3.1 INTRODUÇÃO	27
3.2 TECNOLOGIAS UTILIZADAS NO DESENVOLVIMENTO DA FERRAMENTA	28
3.3 MÓDULO DE INTEGRAÇÃO COM JUÍZES <i>ONLINE</i> (MOJO)	29
3.3.1 Arquitetura Interna do MOJO	30
3.3.2 Principais Funcionalidades do MOJO	31
3.3.3 Linguagens de Programação Disponíveis	35
3.4 NECESSIDADES DA INTEGRAÇÃO	36
3.4.1 Necessidade de Obtenção de Informação	36
3.4.2 Necessidade de tabelas exclusivas	37
3.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO	37
4. INTERAÇÃO COM O MOJO	39
4.1 INTRODUÇÃO	39
4.2 INTERAGINDO NO MOODLE	40
4.3 PROCESSO ESA COM O MOJO	42
4.3.1 Automatização do Processo	42
4.4 FUNCIONAMENTO DA FERRAMENTA	44
4.4.1 Configurações Iniciais do MOJO	45
4.4.2 Professor Elaborando e Submetendo Questão com o MOJO	46
4.4.3 Aluno Submetendo a Solução de uma Questão com o MOJO	50
4.4.4 Avaliação Automática com o MOJO	58
4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO	63

5. CONCLUSÃO	65
5.1 LIMITAÇÕES DA FERRAMENTA	65
5.2 PUBLICAÇÕES	66
5.3 PRIMEIRAS IMPRESSÕES DA FERRAMENTA	68
5.4 CONSIDERAÇÕES FINAIS	69
5.4.1 Trabalhos Futuros	70
5.4.2 Concluindo	70
REFERÊNCIAS	73
APÊNDICE A – NOÇÕES BÁSICAS PARA O DESENVOLVIMENTO DE NOVOS MÓDULOS	76
APÊNDICE B – ESTRUTURA DAS TABELAS EM .XML	79

1. INTRODUÇÃO

Disciplinas de programação são essenciais aos estudantes de computação e de outros cursos que possuem essa disciplina em sua grade curricular, pois constituem a base para muitas áreas de aplicação da informática. O devido aprendizado dessas disciplinas torna o indivíduo apto a utilizar a lógica de programação na resolução de diversos problemas, fator importante e útil em disciplinas mais avançadas e na própria carreira.

Por sua vez, as Tecnologias de Informação e Comunicação (TIC) têm contribuído de maneira significativa para a ampliação das estratégias de ensino e aprendizagem, proporcionando diferentes ferramentas e artefatos que apoiam o processo de aprendizagem (Amaral *et al.*, 2011). Dessa forma, o sistema educacional tem caminhado rumo a novas possibilidades e sendo enriquecido com novas técnicas que visam um melhor desenvolvimento intelectual dos alunos.

A utilização dessas novas tecnologias tem sido de suma importância para auxiliar professores e alunos. Podem-se citar exemplos de sua utilização no auxílio a disciplinas de programação. As ferramentas de TIC aplicadas a tais disciplinas podem facilitar o aprendizado de programação, reduzindo os índices de reprovação e o mau desempenho em outras matérias que tenham programação como pré-requisito.

Por outro lado, problemas relacionados à qualidade e quantidade dos laboratórios e equipamentos disponibilizados para essas disciplinas, número de alunos por turma, necessidade de monitoramento e tratamento individualizado do aluno são algumas das dificuldades encontradas no aprendizado de programação. Dificuldades que refletem em problemas (índices de reprovação e mau desempenho) que podem colaborar para a evasão do curso.

Sobre evasão de curso, de acordo com o jornal Folha de São Paulo, que analisou dados do Ministério da Educação (MEC), existe uma alarmante taxa de 28%, em média, de evasão nos cursos de Ciência da Computação (Takahashi, 2009) nas instituições de ensino superior brasileiras.

Segundo Mota *et al.* (2009), observa-se, principalmente, nos primeiros semestres dos cursos de graduação em computação, uma quantidade relevante de alunos que reprova, desiste ou obtém um baixo rendimento nas disciplinas que focam em programação. Isso é

ocasionado, na maioria dos casos (mas não exclusivamente), devido à complexidade em desenvolver lógica de programação ou à sintaxe da linguagem de programação associada.

No sentido de melhorar esta situação, muitas ferramentas de TIC têm sido propostas para auxiliar o professor no ensino de programação. Um exemplo a ser citado é a utilização dos Ambientes Virtuais de Aprendizagem (AVA), categoria que tem como forte representante o Moodle (Kumar *et al.*, 2011). Porém, mesmo com o advento dessas ferramentas, algumas barreiras ainda são encontradas pelo professor no ensino de programação, como, por exemplo, a dificuldade que se tem ao avaliar todas as atividades de uma turma extensa de alunos em pouco tempo.

No que diz respeito à avaliação de atividades de programação (atividades práticas de codificação), existem alguns sistemas como, por exemplo, os sistemas de Juízes *Online* (Zhigang *et al.*, 2012) que são utilizados em maratonas de programação, e conhecidos por seu amplo repositório de questões e pelo seu processo de avaliação automática de código-fonte. A avaliação feita por estes sistemas gera respostas como: certo, errado, erro de compilação e erro em tempo de execução.

As ferramentas de auxílio existentes, somadas às pesquisas nesta área podem contribuir não apenas para minimizar os problemas de evasão e dificuldade do aprendizado de programação, como também para melhorar a qualidade do processo de ensino/aprendizagem.

Geralmente, quer seja na modalidade de Educação a Distância (EaD) ou na modalidade de educação presencial, essas ferramentas de auxílio fornecem um ambiente que permite ao aluno criar seus algoritmos e codificá-los em alguma linguagem de programação. Porém, é de responsabilidade do professor a elaboração, submissão, avaliação das atividades e fornecimento de *feedback* a todos os seus alunos. Todo este processo, quando realizado para várias turmas com numerosos alunos, acaba causando desgaste (físico e mental) ao professor e uma sobrecarga de suas atividades.

Com base no cenário atual da tecnologia educacional, no que diz respeito ao ensino de programação, é possível integrar dois ou mais ambientes distintos de maneira complementar para um propósito comum. Desta forma, fazendo surgir um novo sistema que contém as características dos sistemas originais. Esta integração é um processo complexo, pois envolve alguns riscos, como, por exemplo, o de comprometer a evolução do *software*. Porém, mesmo com toda a complexidade envolvida, este ainda é o cenário

mais comumente encontrado. Exemplos deste tipo de integração serão apresentados na seção de trabalhos relacionados do capítulo 2.

Visando contribuir com a melhoria do processo de ensino e aprendizagem de programação, este trabalho apresenta o Módulo de Integração com os Juízes *Online* – MOJO (Chaves *et al.*, 2013a), uma ferramenta que integra os Juízes *Online* ao Moodle. Esta ferramenta baseia-se em sistemas utilizados em maratonas de programação com o objetivo de avaliar automaticamente códigos-fonte, e automatiza o processo de Elaboração, Submissão e Avaliação – ESA (Chaves *et al.*, 2013b) de atividades de programação propostas pelo professor. O objetivo é fornecer um ambiente único que facilite o gerenciamento de recursos e permita realizar o devido acompanhamento dos alunos, além de fornecer o devido *feedback* das atividades.

Dando prosseguimento ao capítulo, as subseções seguintes abordam a problemática, justificativa e os objetivos deste trabalho.

1.1 PROBLEMÁTICA E JUSTIFICATIVA

Mesmo os Ambientes Virtuais de Aprendizagem atuais, que apresentam um conjunto de ferramentas de propósito geral e podem ser empregados em vários cursos, raras vezes são concebidos levando em consideração a possibilidade de extensão ou de integração com outras plataformas. Uma exceção a este modelo é o Moodle, pois possui documentação específica para a agregação de novas funcionalidades.

No que diz respeito às disciplinas de programação, mesmo com o uso das ferramentas existentes nos ambientes virtuais, como, por exemplo, disponibilização de notas de aulas, exercícios e fóruns de discussão; percebe-se que estas ferramentas não são o bastante para solucionar a dificuldade do professor de acompanhar e dar o *feedback* necessário no aprendizado de todos os seus alunos.

Como forma de contribuir para melhorar as condições de ensino/aprendizagem em disciplinas de programação, e se aproveitando da possibilidade de extensão oferecida pelo Moodle, este trabalho apresenta uma ferramenta que integra em um mesmo ambiente os sistemas de Juízes *Online* e o Moodle.

Com o ambiente resultante da integração, o professor pode gerenciar os recursos didáticos necessários (atividades, disponibilização de notas e material de estudos) sem a necessidade de utilizar várias ferramentas diferentes. Além de automatizar o processo de

Elaboração, Submissão e Avaliação (ESA) de atividades de programação, propiciando um melhor acompanhamento e *feedback* aos alunos.

1.2 OBJETIVOS

Este trabalho tem como principal objetivo fornecer uma ferramenta que integre Juízes *Online* e Moodle em um mesmo ambiente, proporcionando o auxílio necessário ao professor de disciplinas de programação. A ferramenta é instalada ao Moodle na forma de um módulo, proporcionando um ambiente coeso que permita:

- Auxílio necessário ao professor no que diz respeito ao processo de Elaboração, Submissão e Avaliação de atividades de programação (ESA);
- Acompanhamento de resultados por meio de uma mesma interface disponível no ambiente virtual;
- *Feedback* mais rápido ao aluno;
- Reuso das atividades.

Visando alcançar este objetivo, os seguintes pontos foram levados em consideração:

- Compreensão do funcionamento dos Juízes *Online*;
- Estudo aprofundado e compreensão de como deve ocorrer a integração entre o Moodle e os Juízes;
- Estabelecimento do ambiente utilizado para desenvolvimento da ferramenta (linguagem de programação, banco de dados, servidor);
- Projeto e implementação da ferramenta (MOJO);
- Teste e validação das funcionalidades desenvolvidas.

1.3 ORGANIZAÇÃO DO TEXTO

Para um melhor entendimento, o texto da dissertação está organizado em capítulos da seguinte forma:

- **Capítulo 2 – Ambientes Envolvidos e Trabalhos Relacionados:** Apresenta e descreve os ambientes envolvidos na integração, além de enumerar os principais trabalhos relacionados a esta pesquisa;
- **Capítulo 3 – Um Ambiente Integrado para Auxiliar Práticas de Programação de Computadores:** Apresenta uma visão da ferramenta desenvolvida, as tecnologias envolvidas na sua concepção, suas funcionalidades e as necessidades relacionadas à integração;
- **Capítulo 4 – A Ferramenta em Ação:** Descreve a integração da ferramenta com o Moodle, além de abordar o processo ESA e como ele ocorre com a utilização do MOJO;
- **Capítulo 5 – Conclusão:** São abordadas as limitações da ferramenta e as publicações que derivaram da pesquisa, além de abordar os resultados iniciais e o que se pretende realizar em trabalhos futuros.

2. AMBIENTES ENVOLVIDOS E TRABALHOS RELACIONADOS

O capítulo anterior apresentou alguns dos problemas referentes ao ensino de programação, seja na modalidade de ensino presencial ou à distância. Diante disso, algumas ferramentas e ambientes foram desenvolvidos na tentativa de minimizar ou até mesmo de resolver tais problemas.

Neste contexto, o presente capítulo tem por objetivo apresentar a fundamentação teórica que serve de suporte necessário à elaboração do trabalho dissertativo e ao desenvolvimento da ferramenta que dá título à pesquisa. O capítulo apresenta e descreve os principais ambientes envolvidos na integração, além de listar uma série de trabalhos que estão relacionados à ferramenta desenvolvida.

2.1 MOODLE

Moodle é um acrônimo para *Modular Object-Oriented Dynamic Learning Environment*. Foi desenvolvido pelo australiano Martin Dougiamas em 1999 e possui tradução para diversos idiomas, incluindo o português. É uma plataforma de código aberto (*open source*) para aprendizagem eletrônica (*e-learning*) que é utilizada em todo o mundo, em 193 países, com mais de 400.000 usuários registrados (Kumar *et al.*, 2011).

O Moodle faz parte do grupo de Ambientes Virtuais de Aprendizagem (AVA). Esses ambientes permitem, por meio da Internet, a produção de conteúdos e canais variados de comunicação, bem como o gerenciamento de dados e controle total de informações. Além disso, possui uma comunidade formada por professores, pesquisadores, e, principalmente, programadores que mantém um Portal na *Web*¹ funcionando como uma central de informações, discussões e colaborações.

No contexto educacional, a ferramenta oferece a professores e alunos um ambiente capaz de reunir a maioria das informações e eventos relevantes, associados a uma disciplina de determinado curso. Essas características a tornam uma importante ferramenta auxiliar no ensino superior, podendo ser utilizada tanto na modalidade de ensino à distância como na modalidade de ensino presencial ou semipresencial.

¹ Acesse em: <http://www.moodle.org>

A plataforma conta com uma diversidade de recursos educacionais, permitindo larga flexibilidade em sua configuração e utilização. Por meio do seu desenvolvimento extremamente modular é possível, ainda, a inclusão de novos recursos que se adaptem às necessidades de quem utiliza o ambiente. Porém, na prática, para agregar um recurso externo ao Moodle como, por exemplo, um novo módulo, é preciso ter conhecimento de sua estrutura, assim como saber utilizar os recursos internos por meio de suas bibliotecas de desenvolvimento.

2.1.1 Potencial para Extensão

De acordo com França (2012), muitos administradores de ambientes de aprendizagem vêm aderindo ao Moodle devido ao seu caráter de sistema aberto, baseado em uma forte filosofia educacional, com uma comunidade crescente de usuários que contribui para o desenvolvimento e apoio a usuários novatos.

O modelo de desenvolvimento do Moodle prevê a sua extensão permitindo, assim, o desenvolvimento e disponibilização de novas funcionalidades por meio de *plugins* para que outros usuários possam usufruir dos novos recursos. Uma das formas de extensão e criação de novas funcionalidades é a implementação de blocos ou módulos. O APÊNDICE A mostra as noções básicas para o desenvolvimentos de novos módulos para o Moodle. Esta característica se mostra relevante no desenvolvimento da ferramenta de integração que é o objetivo principal deste trabalho.

Todo esse potencial oferecido pela plataforma para a criação e extensão de funcionalidades justifica a integração do Moodle com outras ferramentas como, por exemplo, os sistemas de Juízes *Online* abordados na seção seguinte.

2.2 JUÍZES ONLINE

A maioria dos programas de natureza algorítmica necessita apenas obter como entrada um padrão de dados devidamente formatados e, a partir desses dados, realizar o devido processamento. Após o processamento, os resultados são apresentados de maneira formatada em uma saída padronizada. Dessa maneira, é possível que a avaliação de programas seja feita automaticamente utilizando uma ferramenta que gere os dados de entrada e outra que obtenha e verifique os resultados obtidos.

Os Juízes *Online* são sistemas que compilam, executam e testam códigos-fonte para julgar se estão corretos (Zhigang *et al.*, 2012). Basicamente, este processo de avaliação automática ocorre da seguinte maneira: o juiz recebe o código-fonte, durante a execução do código, o juiz utiliza dados formatados como a entrada do programa, processa esses dados e realiza a comparação dos resultados obtidos com os resultados esperados, dando uma resposta apropriada com base nessas comparações (certo, errado, erro de compilação e erro de execução).

Este método de avaliação automática de códigos-fonte é utilizado em muitos concursos de programação no Brasil e no mundo, como, por exemplo, *ACM Collegiate Programming Contest International* (ACM-ICPC, 2014) e *TopCoder* (Inc., 2014).

Os sistemas de Juízes *Online* podem ser facilmente encontrados na Internet e como exemplos podem ser citados o *Timus Online Judge* (Timus Online Judge Team, 2014), o *UVA Online Judge* (Universidad de Valladolid, 2014) e o *SPOJ Brasil* (Sphere Research Labs, 2014a), este último utilizado no desenvolvimento deste trabalho. Nestes sistemas são disponibilizados vários problemas para resolução.

Segundo Zhigang *et al.* (2012), os sistemas de Juízes *Online* são tão atraentes que muitos alunos começam a praticar suas habilidades de programação neles. Essa característica reforça como estes sistemas podem ser vantajosos àqueles que pretendem aprender a programar. A seguir são apresentadas as principais características do *SPOJ Brasil*, juiz *online* utilizado neste trabalho.

2.2.1 SPOJ Brasil

A plataforma *SPOJ* é centrada em torno de um sistema de juiz *online* que serve para a avaliação automática de programas submetidos pelo usuário. O *SPOJ Brasil* é a versão brasileira da plataforma *SPOJ*. Algumas de suas características mais importantes incluem:

- Suporte a mais de 45 linguagens e compiladores de programação, incluindo C, C++, PHP, Java, C#;
- Disponibiliza cerca de mais de 500 questões para resolução 24 horas por dia;
- Disponibiliza fóruns de discussão, ranking de classificação, além de links de notícias e informações;

- Disponibiliza problemas de concursos regionais, seletivas e olimpíadas de programação em português.

Nesta primeira versão do MOJO optou-se por utilizar este juiz, devido sua interface amigável, número de questões e por possuir versão em português. Estas características são essenciais para a integração dos ambientes.

2.3 TRABALHOS RELACIONADOS

O uso de ferramentas de TIC para dar suporte à educação, mais especificamente em disciplinas de programação, já vem sendo explorado há alguns anos. Esta seção relaciona algumas dessas ferramentas. Para uma melhor organização e entendimento, esta seção se divide em subseções, onde são apresentadas ferramentas que não estão e que estão integradas a algum Ambiente Virtual de Aprendizagem, neste caso específico, o Moodle.

2.3.1 Ferramentas não Integradas ao Moodle

Entre os ambientes que fornecem apoio à submissão e avaliação automática de atividades de programação destacam-se a WEB-CAT (Edwards, 2004) e a PROGTEST (Souza *et al.*, 2012). WEB-CAT é um ambiente *Web* que visa incentivar o desenvolvimento orientado a testes, apoiando a submissão e avaliação automática de atividades de programação. As ferramentas de teste e de análise são disponibilizadas na WEB-CAT por meio de *plugins*. Cada *plugin* possui critérios de avaliação e relatórios próprios, associados a uma linguagem de programação particular. Dentre as linguagens permitidas, encontram-se Java, C, C++ e Pascal.

A PROGTEST (semelhantemente a WEB-CAT) é um ambiente *Web* automatizado que apoia a submissão e avaliação de atividades práticas de programação, e também se baseia em atividades de teste de *software*. Atualmente, dá suporte a apenas duas linguagens de programação (Java e C) e utiliza um programa referência (programa oráculo) que deve ser fornecido pelo professor para avaliação das atividades dos alunos, além de utilizar diferentes ferramentas para testes.

Uma iniciativa que utiliza Juízes *Online* é a de Santos e Ribeiro, que propõe o *JOnline* (Santos e Ribeiro, 2011) um sistema que possui um Juiz *Online* próprio no qual é

possível acessar problemas para serem resolvidos e submeter códigos-fonte para a visualização do resultado. Além da programação colaborativa, o sistema ainda adiciona funcionalidades didáticas ao juiz, auxiliando o aluno no processo de aprendizagem.

Embora essas ferramentas tragam importantes características para auxiliar no ensino de programação, elas não estão integradas a um ambiente que forneça outros recursos educacionais como, por exemplo, suporte a gestão de conteúdo e ferramenta de discussão. Estes são importantes recursos, principalmente no que diz respeito às disciplinas ministradas à distância, recursos que um AVA como o Moodle pode fornecer. E em algumas ferramentas específicas, além do professor ter que criar programas referências para auxiliar na correção das atividades, tem-se ainda a limitação de se trabalhar restrito a poucas linguagens de programação.

2.3.2 Ferramentas Integradas ao Moodle

Em um contexto semelhante ao abordado nesta pesquisa, algumas iniciativas foram realizadas no sentido de integrar recursos de apoio as disciplinas de programação ao ambiente Moodle, como é o caso da iniciativa de Sirotheau *et al.* (2011), do BOCA-LAB (França, 2012) e do *Onlinejudge* para Moodle (Onlinejudge, 2014). Em Sirotheau *et al.* (2011), com o objetivo de contribuir para uma melhor compreensão do estudante no aprendizado de programação, a ferramenta *JavaTool* (Mota *et al.*, 2009) foi integrada ao Moodle juntamente com o avaliador automático de Moreira e Favero (2009), propiciando uma maneira de visualizar e simular programas no Moodle e permitindo a combinação de técnicas para avaliação da complexidade do código. Desta forma, colaborando para uma melhor avaliação e *feedback* das atividades.

O BOCA-LAB foi desenvolvido no Departamento de Engenharia de Teleinformática (DETI) da Universidade Federal do Ceará (UFC) e surgiu da extensão de um sistema utilizado em maratonas de programação – o BOCA (Campos e Ferreira, 2004). O BOCA-LAB foi integrado ao Moodle por meio de *Web Services* (WS). A ferramenta é capaz de compilar e executar programas escritos em diversas linguagens de programação. Os programas submetidos são então avaliados quanto a erros de compilação e execução em um processo automático, além de tratar problemas de similaridade de código.

O *Onlinejudge* para Moodle é composto por *plugins* e também foi desenvolvido para gerenciar a submissão e avaliação de códigos-fontes no Moodle. Ele pode ser

integrado a outras duas aplicações: a *Sandbox* (Sandbox, 2014) e a *Ideone* (Sphere Research Labs, 2014b). A *Sandbox* se restringe a submissões que executem em C e C++ no ambiente Linux. Já a *Ideone* permite escrever códigos-fonte em aproximadamente 40 linguagens de programação diferentes, sendo executados diretamente a partir do navegador. Entretanto, a *Ideone* é uma aplicação comercial e permite submissão de apenas 2000 códigos-fonte por mês em uma conta gratuita (Zhigang *et al.*, 2012). Sem a integração o *Onlinejudge* para Moodle permite apenas as linguagens C e C++.

Outra importante iniciativa é o *JAssess* (Yusof *et al.*, 2012), que propõe um sistema intermediário desenvolvido para fornecer, no Moodle, uma maneira prática de gerenciar a apresentação de exercícios de programação em Java desenvolvidos pelos alunos. Um ponto fraco encontrado no *JAssess* é o fato da ferramenta se limitar a exercícios que utilizem somente Java como linguagem de programação, enquanto outras linguagens como, por exemplo, C e C++ não são atendidas pela ferramenta.

Estas ferramentas também trazem importantes contribuições para o ensino de programação. Porém, mesmo com a integração ao Moodle, o professor ainda precisa realizar o processo de elaborar, submeter, avaliar e dar o devido *feedback* das atividades para cada aluno de sua(s) turma(s).

2.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Visando diminuir a sobrecarga do professor em relação ao processo ESA de atividades de programação, esta pesquisa, em complemento aos demais trabalhos relacionados, apresenta uma ferramenta que integre os Juízes *Online* (neste caso o SPOJ Brasil) ao Moodle. Unindo o processo de avaliação automática e as várias questões disponibilizadas pelos Juízes às funcionalidades de gerenciamento do Moodle, proporcionando um ambiente único e coeso que forneça:

- 1) Auxílio necessário e maior agilidade nas tarefas do professor com a automatização do processo ESA de atividades de programação;
- 2) Acompanhamento de resultados através de uma mesma interface disponível no ambiente integrado;
- 3) *Feedback* mais rápido ao aluno;
- 4) Suporte às linguagens de programação C, C++, C#, Java e PHP;
- 5) Reuso das atividades.

3. UM AMBIENTE INTEGRADO PARA AUXILIAR PRÁTICAS DE PROGRAMAÇÃO DE COMPUTADORES

Neste capítulo será apresentado o Módulo de Integração com os Juízes *Online*, que será o responsável pela integração dos Juízes *Online* ao Moodle, além de apresentar as tecnologias utilizadas na concepção da ferramenta.

3.1 INTRODUÇÃO

Conforme dito anteriormente, o Moodle oferece a professores e alunos um ambiente que reúne as principais informações e eventos relativos a uma disciplina de determinado curso. Com base nisto, sua utilização em disciplinas voltadas para o ensino de programação, seja na modalidade presencial ou à distância, pode trazer importantes benefícios no que diz respeito à organização da mesma.

Por meio desta plataforma, é permitido ao professor disponibilizar, em um espaço reservado à turma, todo o material didático digital relativo, além de propor atividades e recebimento de trabalhos no próprio ambiente virtual. Além disso, com a possibilidade de fazer lançamentos de notas, é possível aos alunos o acompanhamento de seu rendimento em cada atividade. A plataforma também facilita a interação entre o professor e o aluno por intermédio de ferramentas agregadas como fóruns e chats (França, 2012).

Entretanto, no que diz respeito ao processo de compilação e avaliação automática de códigos fonte, o Moodle, por padrão, não oferece qualquer suporte. Porém, é possível, para este propósito, instalar alguns *plugins* como, por exemplo, o *Onlinejudge* para Moodle, apresentado na seção de trabalhos relacionados, mas, conforme dito, ele possui limitações como, por exemplo, o número de linguagens de programação permitidas.

Diante desta perspectiva, e visando diminuir a sobrecarga de tarefas do professor no que diz respeito ao processo ESA, foi feito um levantamento com alguns professores que lecionam disciplinas de programação na Universidade do Estado do Rio Grande do Norte (UERN), Campus Mossoró. Com base nesse levantamento, foi possível idealizar uma ferramenta capaz de realizar a integração entre os Juízes *Online* ao Moodle.

Pelo fato de não existir qualquer ferramenta semelhante que realize esta integração, permitindo a utilização do processo de avaliação automática e das questões dos juízes no Moodle, foi necessário o desenvolvimento de uma nova ferramenta.

Na seção seguinte, são discutidas as tecnologias que foram consideradas para a integração dos sistemas de Juízes *Online* e Moodle. Na sequência, são apresentados maiores detalhes sobre a arquitetura interna da ferramenta, bem como as adaptações que foram necessárias para o ambiente de integração concebido.

3.2 TECNOLOGIAS UTILIZADAS NO DESENVOLVIMENTO DA FERRAMENTA

Com a limitação de nenhuma *Application Programming Interface* (API) ou *Web Service* (WS) ser disponibilizado pelo juiz utilizado na ferramenta, foi necessário realizar implementações bem específicas para a integração. Esta parte, em especial, foi a que demandou mais tempo e atenção no decorrer do desenvolvimento do módulo.

Para estas implementações, e visando facilitar a integração com a plataforma Moodle, foi utilizada a linguagem de programação PHP (The PHP Group, 2014a), mesma linguagem utilizada no desenvolvimento da plataforma Moodle. O PHP possui diversas características, entre as quais se destacam as seguintes vantagens (Sousa e Porto, 2010):

- É uma linguagem totalmente aberta;
- Possui suporte a diversos sistemas operacionais, como Linux, Windows e Unix e diversas outras plataformas;
- Possui baixa curva de aprendizagem, já que o mesmo possui características de outras linguagens, tendo comandos semelhantes às linguagens C e Java, possibilitando, assim, uma rápida assimilação;
- Dispõe de uma rica e extensa API, disponibilizando funções que vão desde o tratamento de *Strings* e *Arrays*, até comandos de manipulação de *Sockets*;
- Compatibilidade com diversos bancos de dados, velocidade e código-fonte aberto são outras vantagens que tem ajudado para o rápido crescimento do PHP.

Em conjunto com a linguagem PHP, também foi utilizada *JavaScript* (W3Schools, 2014a), uma linguagem de programação *client-side*, que possibilita que os *scripts* sejam executados do lado do cliente e interajam com o usuário sem a necessidade de passar pelo servidor.

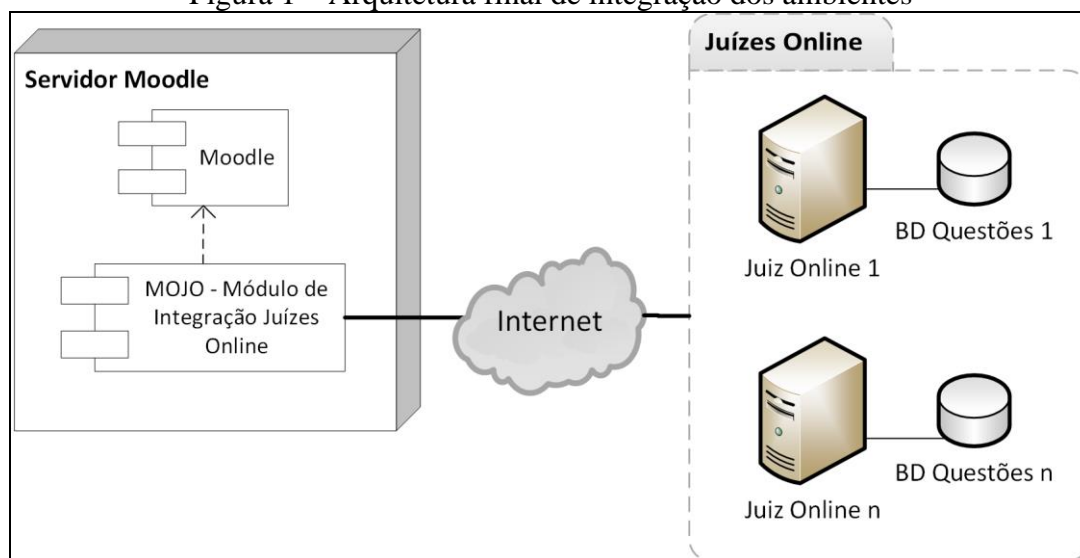
Ainda sobre o uso da linguagem PHP, foi utilizada a biblioteca *client URL* ou simplesmente *cURL* (The PHP Group, 2014b), uma biblioteca bem específica para casos em que se precisa trabalhar obtendo partes específicas da estrutura HTML (W3Schools, 2014b) de uma página *Web*. Esta biblioteca permite a conexão e comunicação com diferentes tipos de servidores com a utilização de diferentes tipos de protocolos. Para a correta utilização da biblioteca é importante ter a extensão **php_curl** habilitada no PHP.

Em relação à base de dados adotada, optou-se por utilizar como Sistema Gerenciador de Banco de Dados (SGBD) o *PostgreSQL* (The PostgreSQL Global Development Group, 2014), por sua característica *open source* (*software* livre). E também pela facilidade em sua utilização e suporte, pois o mesmo SGBD já vem sendo utilizado no Laboratório de Redes e Sistemas Distribuídos (LORDI), onde a pesquisa foi desenvolvida.

3.3 MÓDULO DE INTEGRAÇÃO COM JUÍZES *ONLINE* (MOJO)

O MOJO é a ferramenta propriamente dita (encapsulada em um módulo instalado no Moodle) responsável pela integração dos ambientes. Este módulo é o responsável pela interação que ocorre entre o Moodle e os Juízes *Online* envolvidos nas operações. Na arquitetura final de integração dos ambientes, apresentada na Figura 1, o Moodle fornece a interface e o conjunto de funcionalidades necessárias à gestão e ao acompanhamento das atividades de programação, e o MOJO fornece a comunicação e interação com os Juízes *Online* que fornecem as questões e o processo de avaliação automática.

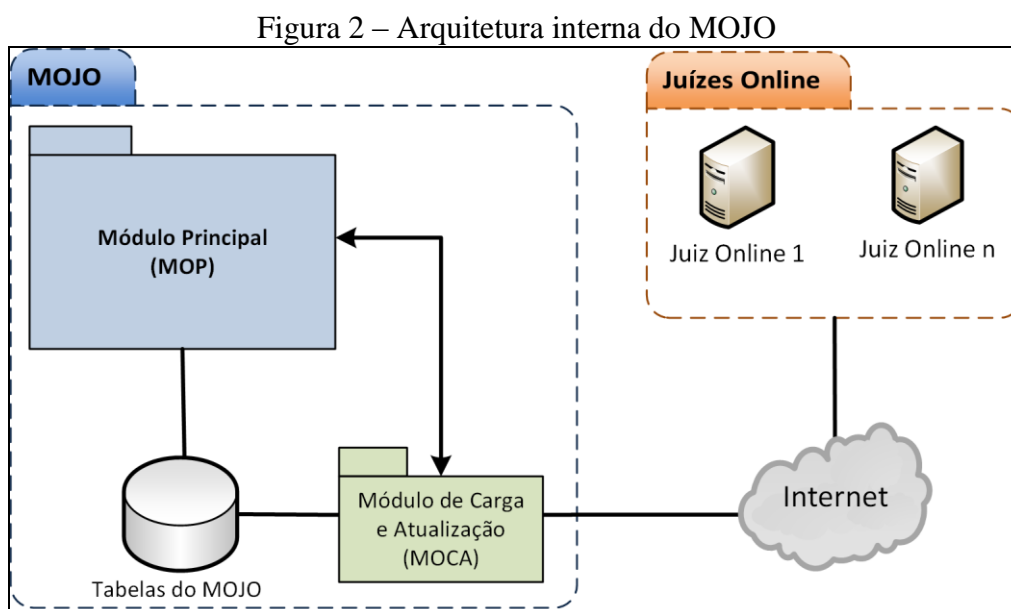
Figura 1 – Arquitetura final de integração dos ambientes



As seções seguintes abordam a arquitetura interna do MOJO, explicando cada um dos seus componentes, e as principais funcionalidades disponibilizadas pela ferramenta para viabilizar a integração.

3.3.1 Arquitetura Interna do MOJO

Com a finalidade de garantir a integração entre os ambientes (Moodle e Juízes Online), a arquitetura do MOJO é composta por dois módulos que se comunicam entre si: o Módulo Principal (MOP) e o Módulo de Carga e Atualização (MOCA), além de um conjunto de tabelas exclusivas do MOJO, apresentados na Figura 2 e detalhados a seguir.



- **Módulo Principal (MOP):** O MOP gerencia, controla e fornece todas as funcionalidades necessárias ao bom funcionamento do MOJO, tais como a disponibilização das questões e dos resultados no Moodle. Conforme dito anteriormente, o MOP interage diretamente com o Módulo de Carga e Atualização, desta forma, gerenciando suas ações.
- **Módulo de Carga e Atualização (MOCA):** O MOCA é o responsável, mediante delegação do MOP, pela carga de questões na ferramenta e por sua constante atualização. Em um primeiro momento, o MOJO realiza uma carga inicial de questões. Esta carga é feita pelo MOCA que interage, por

meio de requisições *Web*, com o repositório de questões dos Juízes *Online*, obtendo as questões e as incorporando ao MOJO. Além disso, o MOCA também oferece o monitoramento do repositório de questões dos juízes a fim de descobrir se novas questões foram adicionadas; em caso afirmativo, é realizada a atualização das questões no MOJO. Toda operação que envolve persistência de dados no MOJO é de responsabilidade do MOCA.

- **Tabelas do MOJO:** Consiste de um conjunto de tabelas exclusivamente criadas para o MOJO, pois foi identificada a necessidade de criação das mesmas para o funcionamento correto da ferramenta. Estas tabelas interagem diretamente com as tabelas do Moodle e são responsáveis por armazenar dados como, por exemplo, as questões disponibilizadas pelos juízes e os resultados de submissões.

A seção seguinte é responsável por apresentar e descrever as principais funcionalidades disponibilizadas pelo MOJO.

3.3.2 Principais Funcionalidades do MOJO

Após estudo das funcionalidades necessárias à utilização do MOJO e suas limitações (implementações específicas para incorporação do juiz *online* utilizado) quanto ao contexto deste trabalho, foi feito um levantamento e definido um conjunto de funcionalidades a serem implementadas:

- Importação de questões;
- Atualização de questões;
- Atribuição de nível a questão;
- Submissão de questões para resolução;
- Submissão de códigos-fonte;
- Visualização de resultados;
- Atualização de resultado da submissão.

A seguir é descrito, em detalhes, o funcionamento de cada uma destas funcionalidades.

a) Importação de questões

Funcionalidade diretamente relacionada ao Módulo de Carga e Atualização. É a operação responsável por realizar a importação das questões dos juízes para o MOJO. A operação ainda informa a quantidade de questões que foram importadas. As informações obtidas na importação de cada questão são:

- i) Código da questão;
- ii) Juiz responsável pela questão;
- iii) Título da questão;
- iv) Descrição da questão;
- v) Linguagem de programação permitida.

b) Atualização de questões

Outra funcionalidade que está diretamente relacionada ao Módulo de Carga e Atualização. Esta operação é responsável por verificar se houve a adição de novas questões no repositório de questões do juiz. Havendo novas questões disponíveis é realizada a atualização no MOJO.

A busca por novas questões leva em consideração a quantidade de questões que havia anteriormente no MOJO, além de considerar o código da questão, para que não exista o risco de duplicação de questões.

c) Atribuição de nível a questão

Por meio desta operação o professor pode, com base no conteúdo da questão, definir um determinado nível de dificuldade para uma questão, sendo 5 (cinco) os níveis de dificuldade disponíveis, conforme é mostrado na Tabela 1.

Tabela 1 – Níveis de dificuldade para atribuição a questão

NÍVEL	DESCRIÇÃO
1	Muito Fácil
2	Fácil
3	Normal
4	Difícil
5	Muito Difícil


O professor pode realizar a atribuição de nível ao submeter uma nova atividade para resolução ou ao editar a mesma no Moodle. Em versões futuras da ferramenta,

pretende-se utilizar regras para esta atribuição, como, por exemplo, a quantidade de erros/acertos de determinada questão.

d) Submissão de questões para resolução

Por meio desta operação o professor pode analisar e selecionar, em uma lista de questões, a questão que melhor se aplica a seus alunos. O professor então submete a questão para resolução pelos alunos. A questão fica visível para resolução pelos alunos como atividade no Moodle no período de tempo estabelecido pelo professor. O professor pode submeter quantas questões achar necessário podendo estabelecer uma pontuação máxima para cada questão.

e) Submissão de códigos-fonte

Por meio desta operação são enviados os dados do código-fonte, fornecido pelo aluno, para o processo de avaliação automática nos Juízes *Online*. Uma vez submetido os dados do código-fonte, o MOJO armazena essas informações em sua base de dados, guardando o resultado desta submissão como 'Pendente'. O processamento desses códigos é realizado nos servidores dos juízes. A forma que o processamento do código acontece nos juízes é de responsabilidade dos mesmos. O MOJO apenas envia o código para avaliação e recebe a resposta (*feedback*) do processo. Vale ressaltar que enquanto o resultado desta questão estiver 'Pendente', sinalizado pelo ícone , a questão estará bloqueada para novas submissões pelo aluno que está com a pendência.

f) Visualização de resultados

Esta operação fornece a opção de recuperar o resultado da avaliação automática realizada pelos Juízes *Online*. O resultado é recuperado utilizando o código da questão, obtendo o *feedback* da base de dados do MOJO e retornando essas informações para visualização. Professores e alunos tem acesso a essa opção.

O professor terá a opção de visualizar os resultados:

- **Por aluno:** opção onde é possível visualizar as questões de um aluno específico e seus resultados, podendo também consultar os códigos-fonte enviados.

- **Por questão:** opção onde é possível visualizar os alunos que submeteram código para determinada questão e seus resultados, podendo também consultar os códigos-fonte enviados.

O aluno, se comparado ao professor, terá apenas a opção de visualizar seus próprios resultados por questão e também pode consultar o código-fonte enviado. Os itens disponíveis para visualização por professor e aluno são:

- Código identificador da submissão (ID Submissão);
- Título da questão;
- Resposta com o resultado da questão;
- Linguagem de programação utilizada na questão;
- Data em que a submissão foi realizada;
- Código-fonte utilizado na submissão da questão.

g) Atualização de resultado da submissão

Esta funcionalidade pode ser utilizada tanto pelo aluno que deseja atualizar o resultado de sua submissão, como também pode ser utilizada pelo professor, para atualizar o resultado da submissão de determinado aluno. A operação é responsável por obter o resultado da submissão, diretamente no juiz *online*. O MOJO obtém este resultado e atualiza a submissão do aluno. Para um melhor entendimento pelo aluno foram estabelecidos os padrões de resultados exibidos na Tabela 2.

Tabela 2 – Resultados da submissão

RESULTADO JUIZ ONLINE	RESULTADO MOJO
Aceito	Resposta Correta
resposta errada	Resposta Errada
erro de compilação	Resposta Errada: Erro de Compilação
tempo limite excedido	Resposta Errada: Tempo Limite Excedido
erro em tempo de execução	Resposta Errada: Erro em Tempo de Execução

Conforme mostrado na Tabela 2, percebe-se que o resultado retornado pelo juiz *online* pode não ser claro o bastante para um aluno mais inexperiente, devido a isso, foi necessário adaptar esse resultado para ser retornado pelo MOJO. Nesta primeira versão da ferramenta, foi adotada esta padronização, mas em versões futuras pretende-se implementar uma funcionalidade que permita ao professor padronizar seu próprio tipo de resultado, caso ache necessário.

3.3.3 Linguagens de Programação Disponíveis

Para esta primeira versão da ferramenta, optou-se por disponibilizar para utilização, no desenvolvimento dos códigos-fonte, 5 (cinco) linguagens de programação, que de acordo com TIOBE Software BV (2014), fazem parte do grupo das 10 linguagens de programação mais populares atualmente². São estas: C, Java, C++, C# e PHP. A Figura 3 ilustra o *ranking* do TIOBE com as linguagens citadas e suas posições quanto à popularidade, com base na comparação realizada entre janeiro de 2013 e janeiro de 2014.

Ainda seria possível disponibilizar outras linguagens, mas por questão de praticidade resolveu-se utilizar estas, que geralmente são as comumente utilizadas no ensino de disciplinas de programação.

Figura 3 – Ranking das linguagens de programação mais populares

Jan 2014	Jan 2013	Mudança	Linguagem de Programação
1	1		C
2	2		Java
3	3		Objective-C
4	4		C++
5	5		C#
6	6		PHP
7	7		(Visual) Basic
8	8		Python
9	10	↑	JavaScript
10	22	↑↑	Transact-SQL

Fonte: TIOBE Software BV (2014) – Figura adaptada pelo autor.

² Dados relativos a janeiro de 2014.

3.4 NECESSIDADES DA INTEGRAÇÃO

Conforme dito na subseção de Tecnologias Utilizadas, o juiz *online* utilizado na integração não fornece qualquer API ou funcionalidades via *Web Service*. Essas características dificultaram, mas não impediram o desenvolvimento da ferramenta. A seguir são explicadas algumas das necessidades relativas à integração entre os ambientes, como a utilização da *cURL*, para obtenção de informações dos juízes, e a necessidade da criação de um conjunto de tabelas próprio para atender às funcionalidades do MOJO.

3.4.1 Necessidade de Obtenção de Informação

O MOJO é uma ferramenta que se propõe a obter questões, de programação, de terceiros (Juízes *Online*) para resolução pelo aluno. Para a obtenção dessas questões, e demais informações necessárias ao MOJO, fez-se o uso da *cURL*, uma biblioteca da linguagem PHP. A biblioteca em si é bem específica para casos em que se precisam obter informações de uma página HTML.

A partir dos pedaços de informações “brutos” obtidos com a utilização da *cURL*, foi feito o devido tratamento, fazendo uso das funções disponibilizadas pela própria linguagem PHP, até a obtenção da informação “lapidada”. Para um melhor entendimento, a Figura 4 exibe um trecho de código utilizando a biblioteca *cURL* para obtenção de informações de uma página.

Figura 4 – Trecho de código exemplificando a utilização da biblioteca *cURL*

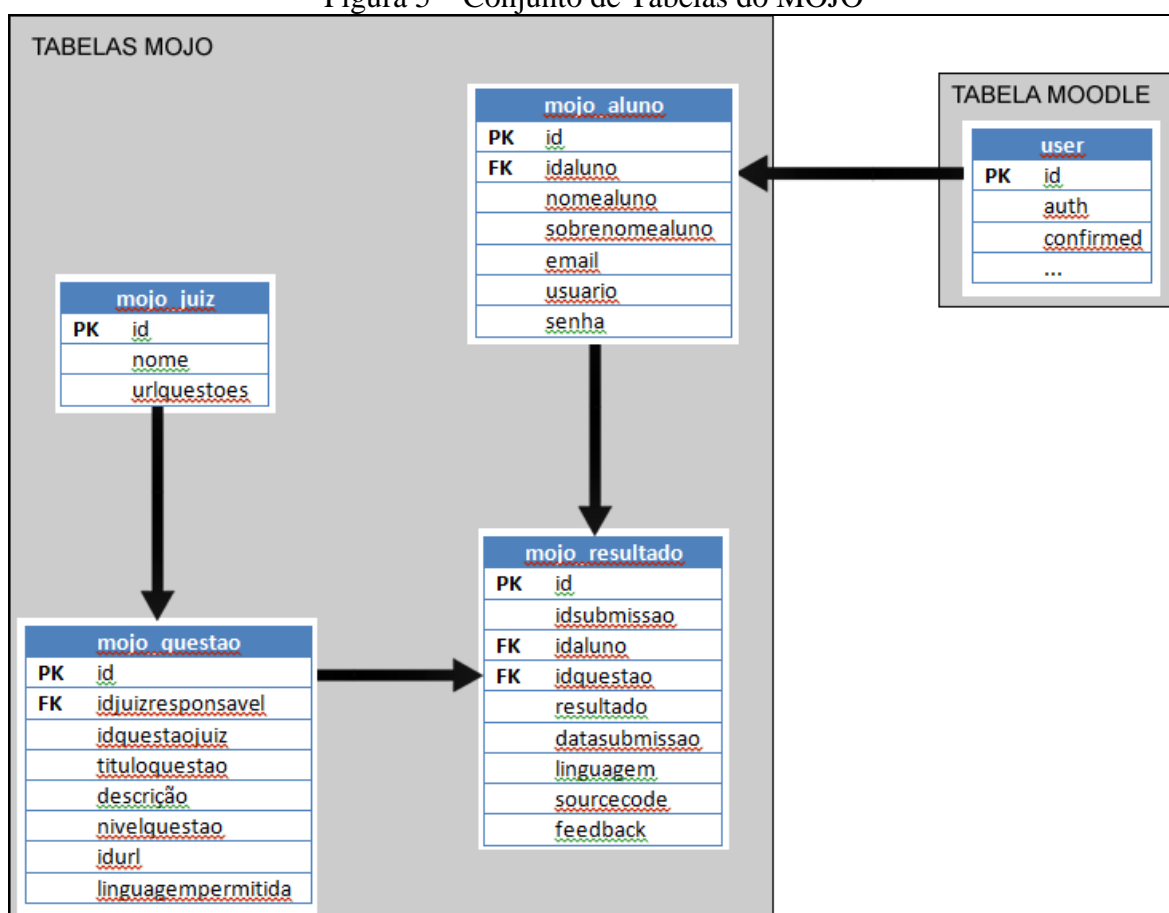
```
1. $ch = curl_init();  
2. $timeout = 0;  
3. curl_setopt($ch, CURLOPT_URL, $paginaHTML);  
4. curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
5. curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, $timeout);  
6. $url = curl_exec($ch);  
7. curl_close($ch);
```

Conforme a Figura 4, a variável **\$paginaHTML**, na linha 3, recebe o endereço URL da página que se deseja obter a informação. Essa informação é colocada dentro de outra variável a **\$url**, na linha 6. Uma vez em posse dessa informação é possível manipulá-la até a obtenção do trecho de informação necessário.

3.4.2 Necessidade de tabelas exclusivas

Percebeu-se, ainda, a necessidade de criação de um conjunto de novas tabelas para o perfeito funcionamento do MOJO, conforme ilustra a Figura 5. Estas novas tabelas, que interagem diretamente com as tabelas do Moodle, tem entre as principais finalidades, armazenar as questões dos juízes e os resultados de submissões das mesmas, além de armazenar informações dos alunos, permitindo o correto direcionamento do *feedback* gerado pela ferramenta.

Figura 5 – Conjunto de Tabelas do MOJO



3.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou a ferramenta que dá título a este trabalho dissertativo: o MOJO – Módulo de Integração com os Juízes *Online*, uma ferramenta inovadora que disponibiliza as questões dos Juízes *Online* no Moodle e utiliza o mecanismo de avaliação automática dos próprios juízes para avaliar as questões.

O capítulo também apresentou as tecnologias utilizadas no desenvolvimento da ferramenta, a arquitetura geral do ambiente integrado, a arquitetura interna do MOJO e as principais características que tornaram possível a integração desejada. Além disso, o capítulo ainda tratou das principais funcionalidades da ferramenta, das linguagens de programação disponíveis nesta primeira versão e das necessidades relevantes à integração.

O próximo capítulo aborda a integração da ferramenta com o Moodle, a automatização do processo ESA e como ele ocorre no ambiente integrado pelo MOJO.

4. INTERAÇÃO COM O MOJO

No capítulo anterior foi possível ter uma visão geral do MOJO, as tecnologias adotadas no seu desenvolvimento, sua arquitetura interna e particularidades que possibilitam a concepção do ambiente integrado. Também foram apresentadas suas principais funcionalidades e as devidas adaptações que precisaram ser feitas para tornar possível a integração desejada.

Este capítulo é o grande responsável por apresentar a ferramenta em ação, sua integração com o Moodle e seu funcionamento passo-a-passo. Além de aprofundar um pouco mais quanto ao processo ESA, o que é e para quem é. O capítulo pode servir como um tutorial para a utilização da ferramenta. O que será bem útil, em um primeiro momento, para os professores que não estiverem tão familiarizados com a ferramenta ou com os ambientes envolvidos. Vale ressaltar que, ao final do capítulo, existe um link para um vídeo na *Web* onde é possível assistir a uma demonstração do MOJO na prática.

4.1 INTRODUÇÃO

Com base no Moodle, um AVA com várias funcionalidades voltadas para a educação e que permite sua extensão por meio da criação de novos módulo, e nos Juízes *Online*, atraentes sistemas (Zhigang *et al.*, 2012) que fornecem questões para resolução e um processo de avaliação automática dessas questões, foi desenvolvido um novo módulo de atividades para o Moodle.

Este módulo, conhecido como Módulo de Integração com Juízes *Online* ou simplesmente MOJO, traz para dentro do Moodle as questões disponibilizadas pelos Juízes *Online*, para que o aluno possa desenvolver suas habilidades de programação. O módulo envia a solução da questão, desenvolvida pelo aluno, para ser avaliada pelo juiz *online* responsável. O MOJO, então, obtém a resposta do processo de avaliação automática da questão e exibe o resultado final para os interessados (professores e alunos).

As seções seguintes explicam, detalhadamente, como é o funcionamento deste novo módulo e como ele poderá auxiliar professores e alunos no ensino/aprendizagem de disciplinas de programação de computadores.

4.2 INTERAGINDO NO MOODLE

Conforme dito anteriormente, o Moodle oferece a professores e alunos um ambiente que reúne informações e eventos relevantes, associados a uma disciplina. Como o foco deste trabalho são as disciplinas de programação, tomemos como exemplo um curso de computação e suas disciplinas. Como forma de ilustrar o exemplo, a Figura 6 mostra a tela inicial do Moodle com algumas disciplinas de programação cadastradas e seus respectivos professores.

Figura 6 – Disciplinas de programação em um curso no Moodle

The screenshot displays the Moodle interface. On the left is a navigation menu with sections: 'Navegação' (containing 'Página inicial', 'Minha página inicial', 'Páginas do site', 'Meu perfil', 'Meus cursos'), 'Configurações' (containing 'Configurações da página inicial' with sub-items like 'Alterar edição', 'Editar configurações', 'Usuários', 'Backup', 'Restaurar', 'Banco de questões'), and 'Administração do site' (containing 'Avisos', 'Inscrição', 'Opções avançadas', 'Usuários', 'Cursos', 'Notas', 'Local', 'Idioma', 'Plugins', 'Segurança', 'Aparência', 'Página principal', 'Servidor', 'Relatórios', 'Desenvolvimento', 'Ajuda para atualização de questões'). The main content area is titled 'Cursos disponíveis' and lists three courses: 'Laboratório de Desenvolvimento de Sistemas 2' (Teacher: Balthazar Mesquita), 'Laboratório de Desenvolvimento de Sistemas 1' (Teacher: Balthazar Mesquita), and 'Programação 1' (Teacher: Osvaldo Mesquita). On the right, there is a calendar for January 2014.

Para um melhor entendimento, é necessário explicar que o Moodle trata as disciplinas cadastradas como cursos, por isso no exemplo da Figura 6 é possível observar “Cursos disponíveis” ao invés de “Disciplinas disponíveis”. Mas que fique claro que o exemplo ilustra as disciplinas (de programação) de um determinado curso (de computação).

Nos cursos disponibilizados no Moodle, o professor dispõe de uma variedade de atividades que irão auxiliar seu trabalho na disciplina, são exemplos dessas atividades: Fórum, Chat, Tarefas, Wiki. O professor que leciona a disciplina, estando devidamente cadastrado no Moodle, pode ir adicionando essas atividades à disciplina. Uma vez adicionada, a atividade é disponibilizada aos alunos cadastrados na disciplina.

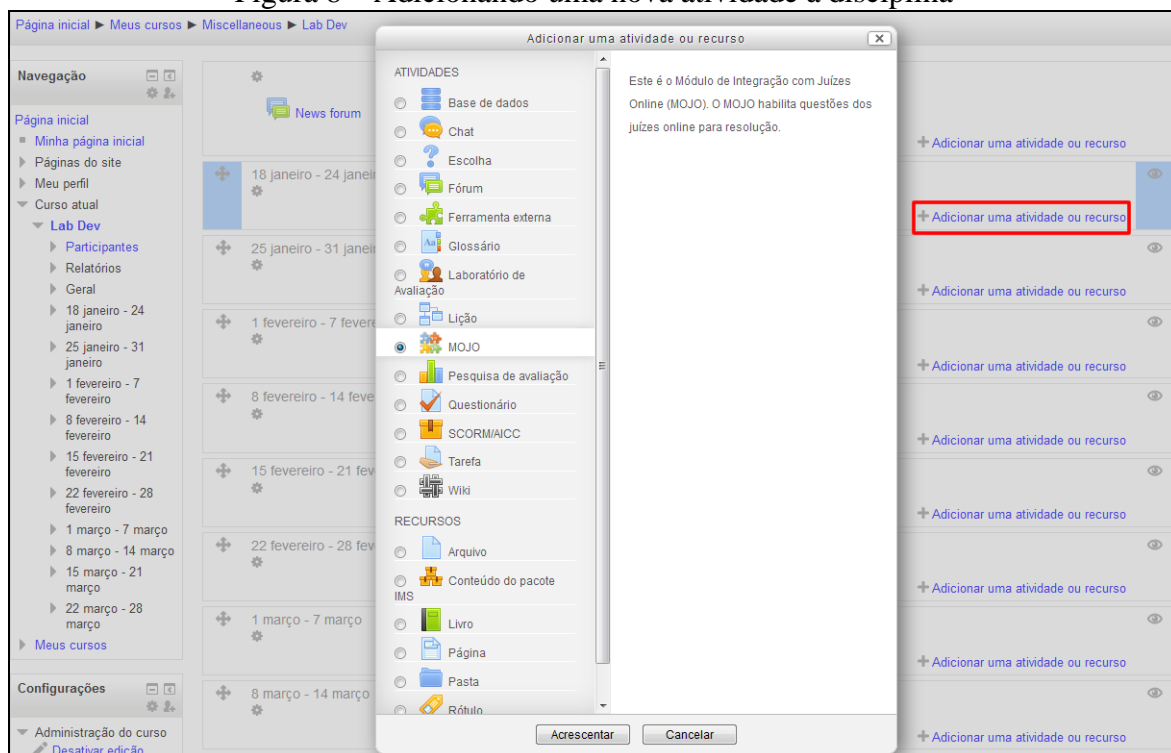
Para a adição de uma nova atividade à disciplina, o professor deve acessar a página principal de sua disciplina, onde ele pode observar a opção “Ativar edição”, conforme destacado na Figura 7.

Figura 7 – Editando uma disciplina no Moodle



Depois de ativada a edição da disciplina, uma nova opção intitulada “Adicionar uma atividade ou recurso” é disponibilizada. Ao selecionar esta opção, uma nova tela exibindo as opções de tipos de atividades surge, como mostrado na Figura 8.

Figura 8 – Adicionando uma nova atividade à disciplina



Dentre os tipos de atividades disponíveis, mostrados na Figura 8, é possível observar a atividade do tipo “MOJO” que é a atividade disponibilizada pelo novo módulo desenvolvido nessa pesquisa. Ao selecionar a atividade, um pequeno diálogo de ajuda é exibido ao lado e o usuário tem a opção de “Acrescentar” a atividade ou de “Cancelar”, conforme botões mostrados na ilustração.

Dados os primeiros passos para adição de uma nova atividade, a seção seguinte aborda o processo ESA e como é o funcionamento da ferramenta integrada ao Moodle a partir do momento que uma nova atividade do tipo “MOJO” é adicionada à disciplina.

4.3 PROCESSO ESA COM O MOJO

O processo ESA, citado no decorrer deste trabalho, nada mais é que o processo de Elaborar, Submeter e Avaliar uma atividade de programação, realizado pelo professor (Chaves *et al.*, 2013c). O MOJO automatiza este processo.

De acordo com o que foi dito em capítulos anteriores, o professor fica sobrecarregado por ter que realizar este processo para cada aluno em cada uma de suas turmas. Com a automatização do ESA, o professor aproveita o tempo ganho com essa automatização para realizar um acompanhamento mais próximo de seus alunos, principalmente daqueles que enfrentam maiores dificuldades no aprendizado da disciplina. Esse acompanhamento é importante, pois serve como motivação ao aluno para continuar no curso. A recompensa do professor vem com a melhoria no desempenho de seus alunos.

4.3.1 Automatização do Processo

No ambiente integrado pelo MOJO, o processo ESA automatizado possui 3 (três) importantes personagens que desempenham diferentes e importantes papéis. Esses personagens são: o professor, o aluno e o juiz *online*. A colaboração entre os envolvidos permite que o a ferramenta trabalhe da seguinte forma:

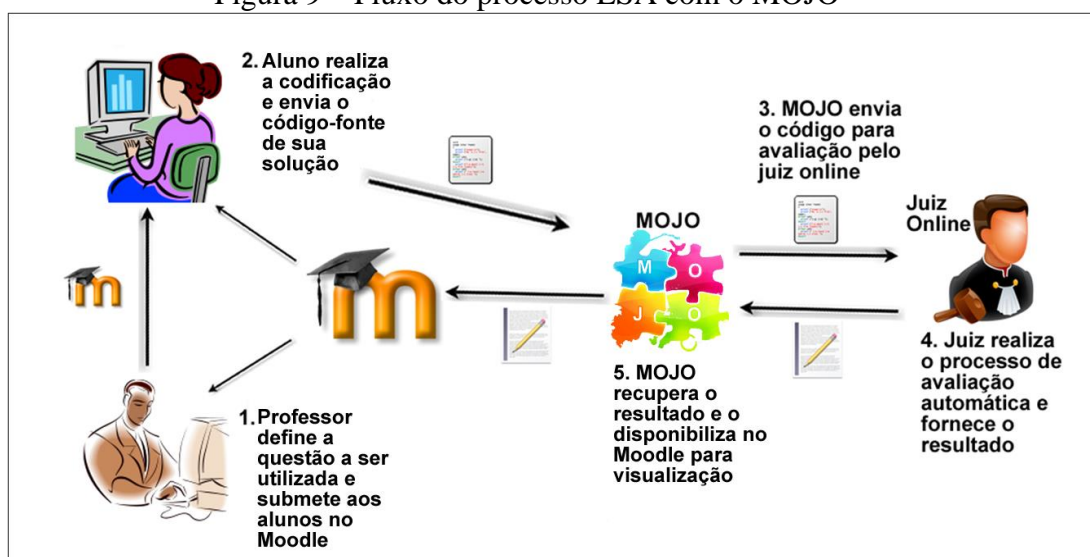
- a) Facilitando a **Elaboração** de atividades: uma vez que os Juízes *Online* fornecem suas questões e as mesmas passam a fazer parte do MOJO, o professor vai poder selecionar uma dessas questões, sem a necessidade dele mesmo precisar elaborar a questão. Vale ressaltar que o professor não precisa elaborar a questão, mas ele ainda precisa analisar e selecionar a questão que melhor se aplica aos alunos de sua turma. Por outro lado, o professor não pode elaborar sua própria questão, ele apenas pode utilizar as questões já existentes no MOJO.
- b) Facilitando a **Submissão** de atividades: neste ponto o MOJO possui 2 (dois) tipos distintos de “submissão”. Um tipo de submissão é a que o professor realiza após

selecionar a questão no MOJO e ela é automaticamente submetida para resolução dos alunos no Moodle. Esse tipo de submissão já é realizado pelo Moodle após escolha e configuração das atividades. A novidade do MOJO está no tipo de submissão realizado pelo aluno. O aluno, percebendo uma nova atividade, irá a analisar a questão e desenvolver uma solução possível para ela. Com sua solução pronta, o aluno a envia para avaliação. Essa submissão é feita a partir do Moodle, pelo MOJO, diretamente ao juiz responsável por avaliar a questão.

- c) Facilitando a **Avaliação** automática de atividades: após submissão da solução da questão pelo aluno e envio da mesma para avaliação no juiz, o MOJO entra em contato com o juiz da questão para obter o resultado da avaliação e o disponibilizar para visualização dos interessados (professores e alunos) no Moodle.

Para um melhor entendimento, a Figura 9 ilustra o fluxo completo deste processo.

Figura 9 – Fluxo do processo ESA com o MOJO



Conforme ilustrado na Figura 9, o fluxo do processo se divide em 5 (cinco) etapas. Sintetizando bem cada uma delas, temos:

- **Etapa 1:** o professor, no Moodle, define a questão e a submete para resolução pelos alunos (Elaboração e Submissão).
- **Etapa 2:** o aluno visualiza, desenvolve e submete uma solução para a questão (Submissão).

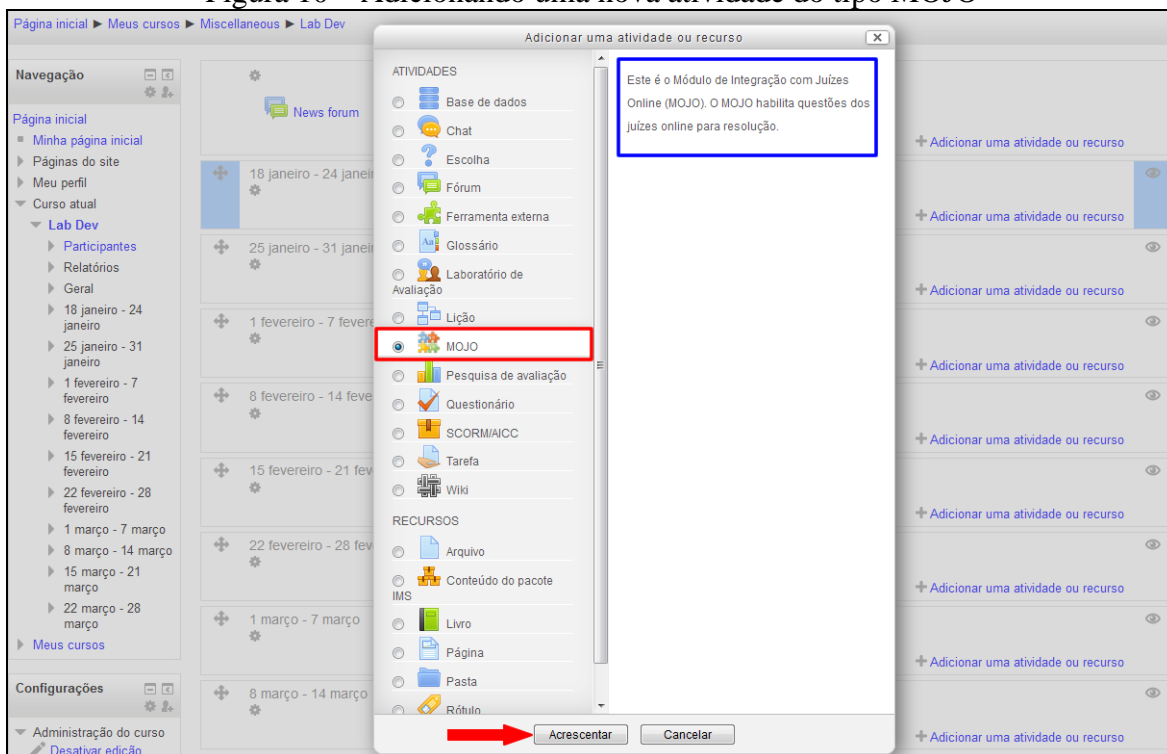
- **Etapa 3:** o MOJO entra em contato com o juiz responsável pela questão e envia o código-fonte para avaliação (Submissão).
- **Etapa 4:** o Juiz *Online* realiza os devidos processos de avaliação automática para a solução proposta e devolve o resultado (Avaliação).
- **Etapa 5:** o MOJO obtém o resultado da avaliação e o disponibiliza no Moodle para visualização pelo professor e pelo aluno (Avaliação).

Na prática o processo é bem mais complexo, por isso a próxima seção aborda esse funcionamento de forma mais detalhada.

4.4 FUNCIONAMENTO DA FERRAMENTA

Com o processo ESA devidamente apresentado é possível dar prosseguimento às explicações sobre o funcionamento da ferramenta no Moodle. Esta seção é a responsável por detalhar este funcionamento, desde o momento da seleção do tipo de atividade “MOJO” até a visualização dos resultados e envio de *feedback* das submissões no Moodle.

Figura 10 – Adicionando uma nova atividade do tipo MOJO



A Figura 10 mostra a tela de um professor que selecionou a atividade do tipo MOJO para ser adicionada à disciplina. Nesta tela é possível observar, além dos botões de “Acrescentar” e “Cancelar”, uma breve descrição da atividade, essa descrição serve de ajuda ao professor na escolha do tipo de atividade a ser proposta aos alunos.

4.4.1 Configurações Iniciais do MOJO

Ao clicar no botão para acrescentar a atividade o MOJO é ativado pela primeira vez. O carregamento do módulo na sua primeira utilização pode ser um pouco mais demorado, pois neste primeiro momento é realizada uma carga inicial de questões no MOJO, além da importação dos dados necessários aos alunos.

Figura 11 – Formulário principal do MOJO

The screenshot displays the MOJO configuration interface. At the top, a status bar indicates 'Acrescentando um(a) novo(a) MOJO em 27 fevereiro - 5 março'. The form is organized into three main sections:

- Geral:** Contains a 'Name*' text input field.
- Lista de questões:** Includes a 'Juiz Online' dropdown menu set to 'SPOJ Online Judge', a 'Lista' dropdown menu with options '1-Investindo no mercado de ações 1', '2-Investindo no mercado de ações 2', '3-Atleta mentiroso', and '4-Cinefinlândia', and a 'Total de Questões' input field set to '393'. A blue link 'Adicionar Questões' is also present.
- Questão:** Features a 'Título' text input, a 'Código' text input, and a 'Descrição*' rich text editor with a standard toolbar. Below the editor is a 'Caminho:' field with the value 'p'. At the bottom, there are fields for 'Nível', 'Linguagem', and 'Nota máxima' (set to '100').

A primeira carga é realizada para que o professor tenha questões para utilizar em suas atividades no primeiro acesso, mas ele pode adicionar mais questões posteriormente.

A importação de dados dos alunos ocorre para que o MOJO realize o devido controle das submissões e encaminhamento do *feedback*, pois esses dados são utilizados pela ferramenta para realizar a autenticação do aluno no juiz *online*, uma vez que cada aluno necessita de credenciais de acesso para a utilização do módulo, este mecanismo de autenticação é explicado melhor na seção “4.4.3 Aluno Submetendo a Solução de uma Questão com o MOJO”.

Após as devidas configurações iniciais, o professor tem acesso ao formulário principal do MOJO, ilustrado na Figura 11 e pode realizar a submissão de questões para seus alunos, dentre outras funcionalidades, conforme detalhado na seção seguinte.


4.4.2 Professor Elaborando e Submetendo Questão com o MOJO

No formulário disponibilizado ao professor, exibido na Figura 11, é possível identificar 3 (três) regiões distintas e suas particularidades, detalhadas a seguir:

- 1) Região **Geral**: contém um campo obrigatório que dá nome a atividade;
- 2) Região **Lista de questões**: exibe o campo “Juiz Online” que já vem preenchido para informar a qual juiz as questões pertencem, além de uma lista com todas as questões que o professor pode selecionar pra submeter a seus alunos (a lista informa o número e o título da questão). Essa região também exibe um campo que informa o total de questões que foram importadas para o MOJO, além de disponibilizar um link com a opção “Adicionar Questões” (conforme destacado na Figura 12), onde é possível importar novas questões para o MOJO.

Figura 12 – Total de questões e opção para adicionar novas

Lista de questões

Juiz Online  SPOJ Online Judge

Lista

- 235-Competição de placas de carros
- 236-Baixar as trincas
- 237-Emoticons
- 238-Achando os assentos

Total de Questões 393

[Adicionar Questões](#)

Quando o professor clica na opção de adicionar novas questões uma nova tela é exibida, conforme ilustra a Figura 13.

Figura 13 – Tela para adição de novas questões ao MOJO

Esta tela exibe os concursos com questões disponíveis pelo juiz (em destaque na Figura 13). O professor que desejar realizar a importação de questões no MOJO precisa apenas selecionar um dos concursos disponíveis. A ferramenta então verifica se existe questão a ser importada para aquele concurso ou não. Se existir nova questão a ferramenta realiza o cadastro da mesma e atualiza o total de questões, caso não exista, esta informação é retornada ao professor e ele pode tentar importar questões de outro concurso.


- 3) Região **Questão**: exibe os campos relativos aos dados da questão, esses campos serão preenchidos quando o professor selecionar uma das questões da lista. O campo “Título” é preenchido com o título da questão (o mesmo da lista). O campo “Código” é preenchido com o identificador da questão no juiz ao qual ela pertence, o que facilita o direcionamento do envio da questão para avaliação no juiz. O campo “Descrição” é obrigatório para não permitir que o professor submeta uma questão sem conteúdo a seus alunos. O campo “Nível” é onde o professor pode realizar a atribuição do nível de dificuldade à questão de acordo com a Tabela 1, caso a mesma já não tenha sido nivelada. O campo “Linguagem” exibe as linguagens de programação nas quais a solução da questão pode ser desenvolvida. Já o campo “Nota Máxima” é onde o professor pode estabelecer o valor máximo de pontuação para aquela questão. O ícone  ao lado do campo fornece informações importantes sobre o campo. A Figura 14 mostra um exemplo de questão selecionada e seus campos preenchidos.

Figura 14 – Campos do formulário preenchidos com dados da questão selecionada

Geral	
Name*	Atividade de Teste no MOJO
Lista de questões	
Juiz Online 🤖	SPOJ Online Judge
Lista	<ul style="list-style-type: none"> 235-Competição de placas de carros 236-Baixar as trincas 237-Emoticons 238-Achando os assentos
Total de Questões	393
Adicionar Questões	
Questão	
Título	Baixar as trincas
Código	TRINCAS
Descrição*	<div style="border: 1px solid #ccc; padding: 5px;"> <p>Família da font: Tamanho da font: Parágrafo</p> <p>B <i>I</i> <u>U</u> ABC x₂ x₃ [Listas] [Links] [Images] [Tables] [Code] [HTML]</p> <p>Os habitantes de uma pequena ilha caribenha na região conhecida como Triângulo das Bermudas adoram gastar suas noites quentes de verão jogando cartas. Como um tributo para a região onde moram, todos seus jogos de cartas têm alguma conexão com triângulos. Um dos jogos mais populares na ilha é conhecido como Trincas, e tem regras bem simples.</p> <p>O jogo é jogado entre dois amigos, com um baralho de cartas padrão. As cartas são distinguidas apenas por seus valores, de 1 (Ás) a 13 (Rei). As cartas são embaralhadas e colocadas em uma pilha no centro da mesa, viradas para baixo. Essa pilha é chamada de pilha. Os dois jogadores jogam em turnos. A cada turno, um jogador</p> <p>Caminho: p</p> </div>
Nível	3 - Normal
Linguagem	C, C#, C++, JAVA e PHP
Nota máxima	100

A Figura 14 mostra a questão “Baixe as trincas” selecionada e os dados relativos a ela preenchendo os devidos campos. É importante ressaltar que o campo de descrição da questão, possui uma barra de rolagem que permite uma visão completa das informações que descrevem a questão, estas informações compreendem o enunciado da questão, os dados de entrada/dados de saída e exemplos desses dados.

Este formulário principal de atividade no MOJO ainda possui outras 2 (duas) regiões: a região “**MOJO Resultados**” que permite ao professor visualizar os resultados das submissões de seus alunos, e a região “**Configurações comuns de módulos**” que não é detalhada aqui, pois como o próprio nome sugere, é uma região comum a todos os módulos de atividade do Moodle, não sendo específica do MOJO.

Com a devida questão selecionada, o professor tem as opções de “Salvar e voltar ao curso”, “Salvar e mostrar” e “Cancelar”. Essas opções são explicadas a seguir:

- a) Na opção **Salvar e voltar ao curso**, os dados da atividade são salvos e o professor é redirecionado para a tela principal da disciplina, onde é possível visualizar a atividade recém-adicionada, além de outras atividades que porventura tenham sido adicionadas anteriormente, como pode ser visto na Figura 15.

Figura 15 – Tela principal da disciplina com atividades adicionadas

Como pode ser visto na Figura 15, o professor tem uma visão de todas as atividades adicionadas, além de ter acesso a certas funcionalidades, como, por exemplo, editar a atividade, excluí-la, duplicá-la e outras que não serão detalhadas aqui, pois não são específicas do funcionamento do MOJO.

Figura 16 – Tela de visualização da questão pelo professor

- b) Na opção **Salvar e mostrar**, os dados da atividade são salvos e o professor é redirecionado para uma tela onde ele pode visualizar a questão submetida. Esta tela é semelhante a que é disponibilizada ao aluno para visualização da questão. A Figura 16 ilustra o exemplo desta tela.

A diferença entre a tela de visualização de questão do professor e do aluno é que na visualização do aluno existe a opção de submeter uma solução para a questão. Esta opção não é disponibilizada para o professor.

- c) Na opção **Cancelar**, o professor cancela a ação de adicionar a atividade e ele é redirecionado para a tela principal da disciplina onde ele pode adicionar outras atividades ou acessar outras opções, caso assim deseje.

Apesar dessas 3 (três) opções não serem específicas do MOJO, o detalhamento de suas funções é de fundamental importância para entendimento na utilização da ferramenta.

Esta seção detalhou o comportamento da ferramenta do ponto de vista do professor, desde a definição da questão até sua submissão na forma de uma atividade no Moodle. A seção seguinte detalha o comportamento da ferramenta do ponto de vista do aluno.

4.4.3 Aluno Submetendo a Solução de uma Questão com o MOJO

Após o professor adicionar uma nova atividade contendo uma nova questão do MOJO, a mesma estará disponível para visualização e resolução pelo aluno na tela principal da disciplina, conforme mostra a Figura 17.

Figura 17 – Tela principal da disciplina na visão do aluno

The screenshot displays the Moodle course interface for 'Laboratório de Desenvolvimento de Sistemas 1'. The main content area shows the 'Agenda do Curso' with a list of activities. Two activities are highlighted with red boxes: '1ª Atividade Lab Dev 1' and 'Atividade de Teste no MOJO'. The right sidebar contains a search bar, a 'Pesquisar nos Fóruns' section, and a 'Próximos eventos' section. At the bottom right, the 'Atividade recente' section shows a recent activity report for 'Atividade de Teste no MOJO' dated 24 January 2014, 03:53.

Período	Atividade	Check
23 janeiro - 29 janeiro	1ª Atividade Lab Dev 1	<input type="checkbox"/>
	Atividade de Teste no MOJO	<input type="checkbox"/>
30 janeiro - 5 fevereiro		<input type="checkbox"/>
6 fevereiro - 12 fevereiro		<input type="checkbox"/>
13 fevereiro - 19 fevereiro		<input type="checkbox"/>
20 fevereiro - 26 fevereiro		<input type="checkbox"/>
27 fevereiro - 5 março		<input type="checkbox"/>
6 março - 12 março		<input type="checkbox"/>
13 março - 19 março		<input type="checkbox"/>

Conforme mostrado na Figura 17, é possível observar as atividades adicionadas pelo professor na região “Agenda do Curso”. Além disso, é possível observar ao lado direito alguns blocos contendo informações como atividades recentes, que fornece uma visão das atividades do usuário com base em um intervalo de tempo específico, esta funcionalidade será abordada posteriormente.

Quando o aluno seleciona uma atividade (do tipo MOJO) adicionada pelo professor, ele é redirecionado para uma tela onde é possível visualizar os dados relativos à questão que compõe a atividade (título da questão, descrição, nível da questão, linguagem permitida e nota máxima). Nesta tela, o aluno tem as seguintes opções:

- a) **Resultados**, onde ele pode consultar seus resultados para aquela questão (caso já exista algum registro de submissão para a questão).
- b) **Submeter Solução**, onde ele pode submeter um código-fonte para ser avaliado como uma possível solução da questão.
- c) **Continuar**, essa opção redireciona o aluno para a página principal da disciplina e ele pode escolher outra atividade.

Figura 18 – Tela de visualização da questão com destaque nas opções do aluno

The screenshot displays the 'Atividade de Teste no MOJO' interface. On the left, a navigation sidebar shows the user's path: 'Página inicial' > 'Meus cursos' > 'Lab Dev 1' > '23 janeiro - 29 janeiro' > 'Atividade de Teste no MOJO'. The main content area features a title 'Baixe as trincas' and a rich text editor with a toolbar. Below the editor, the question description is visible, detailing a card game called 'Trincas' from the Bermuda Triangle. A control panel on the right includes fields for 'Nível' (3 - Normal), 'Linguagem' (C, C#, C++, JAVA e PHP), and 'Nota Máxima' (100). At the bottom, three buttons are highlighted with red boxes: 'Resultados', 'Submeter Solução', and 'Continuar'.

A Figura 18 exibe a tela de visualização da questão na visão do aluno com as opções descritas em destaque. A opção de “Resultados” será abordada posteriormente.

Caso o aluno opte pela opção “Submeter Solução” para a questão, mas o mesmo ainda não possua cadastro nos Juízes *Online* isso não será problema, pois a ferramenta gera, com base nas informações do aluno, as credenciais de acesso (usuário e senha) e as cadastra automaticamente no juiz, sem necessidade de intervenção por parte do professor ou do aluno.

Conforme dito anteriormente, nas configurações iniciais do MOJO é feita a importação dos alunos do Moodle para o MOJO, essa importação é importante para manter a consistência das informações no ambiente integrado. Porém, para utilização de todas as funcionalidades do MOJO, o aluno necessita das credenciais de acesso ao juiz *online* (neste caso, no SPOJ Brasil). Para resolver este problema a ferramenta, na primeira tentativa do aluno de submeter código, gera as credenciais necessárias permitindo a liberação do aluno para submissão de códigos.

As credenciais são fornecidas pela ferramenta de forma a tornar ainda mais ágil o processo, evitando que o aluno (ou professor) precise realizar um cadastro a mais, neste caso, se deslocando ao site do juiz para realizar o cadastro. A necessidade dessas credenciais é conhecida apenas pelo MOJO, ou seja, o processo de geração e cadastro das credenciais não fica sendo de conhecimento do aluno.

Feito os devidos esclarecimentos, sobre as credenciais necessárias, é possível dar prosseguimento às explicações a partir do momento em que o aluno clica em “Submeter Solução”. Neste momento ele é direcionado para a tela de submissão de solução da questão. Esta tela é mostrada na Figura 19.

Figura 19 – Tela de submissão de questão pelo aluno

Laboratório de Desenvolvimento de Sistemas 1 Você acessou como **Mojo Teste** (Sair)
Português - Brasil (pt_br)

Navegação

- Página inicial
 - Minha página inicial
- Páginas do site
- Meu perfil
- Meus cursos
- Cursos

Configurações

- Minhas configurações de perfil

MOJO Submissão de questão

Atividade de Teste no MOJO

Título: ?

Linguagem: (Menu suspenso aberto com opções: C, C#, C++, Java, PHP)

Solução:

Na tela exibida na Figura 19 é possível identificar os campos necessários à submissão da solução de uma questão, estes campos são:


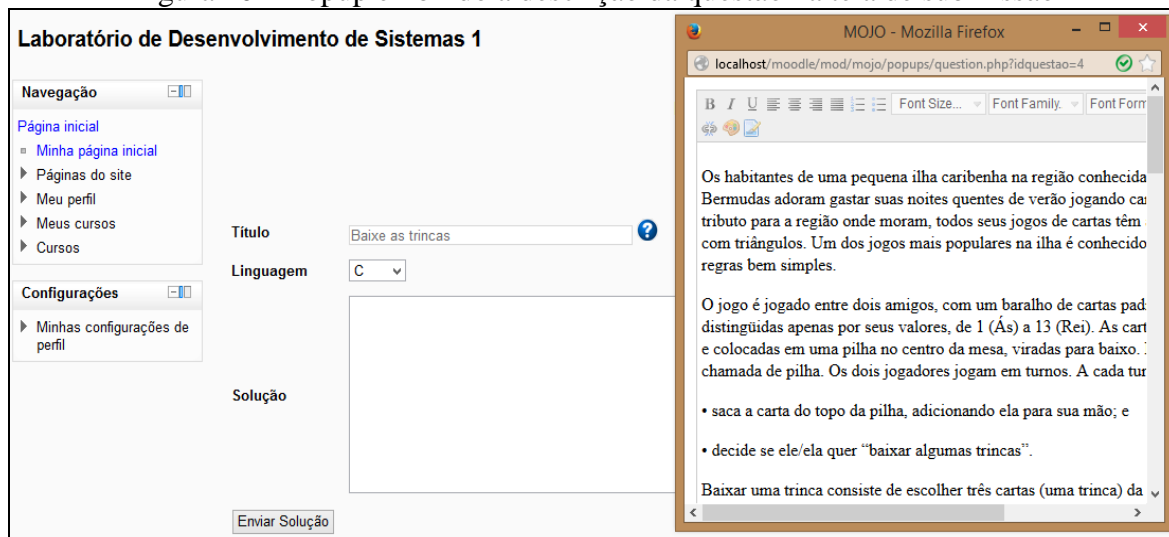
- i) **Título:** Título da questão que está sendo resolvida. O ícone  se clicado abre um *popup* que exhibe a descrição da questão (ver Figura 20).

Figura 20 – Popup exibindo a descrição da questão na tela de submissão



Esta informação é importante caso o aluno tenha dúvidas quanto ao enunciado ou dados da questão que está sendo respondida.

- ii) **Linguagem:** Lista das linguagens de programação com as quais a solução da questão pode ser codificada. O aluno seleciona apenas uma linguagem por vez. Nesta primeira versão da ferramenta, a opção de linguagem ficou aberta para escolha do aluno, se percebida a necessidade do professor em restringir a uma linguagem específica, essa melhoria pode ser facilmente implementada em versões futuras.
- iii) **Solução:** Campo reservado para a solução da questão propriamente dita. O aluno deve inserir o código-fonte neste espaço. Como nesta versão do MOJO ainda não existe um editor de código-fonte integrado, o aluno pode realizar a codificação no seu editor de preferência e depois transferir seu código para este campo.

A Figura 21 ilustra uma solução de questão prestes a ser enviada para avaliação.

Figura 21 – Solução de questão pronta para ser enviada

The screenshot shows the 'Laboratório de Desenvolvimento de Sistemas 1' interface. On the left, there is a navigation menu with options like 'Página inicial', 'Minha página inicial', 'Páginas do site', 'Meu perfil', 'Meus cursos', and 'Cursos'. Below it are 'Configurações' and 'Minhas configurações de perfil'. The main area is titled 'MOJO Submissão de questão' and 'Atividade de Teste no MOJO'. It contains a form with the following fields:

- Título:** A text input field containing 'Baixe as trincas' with a help icon.
- Linguagem:** A dropdown menu set to 'C'.
- Solução:** A large text area containing C code:

```
#include "stdio.h"
int main()
{
    int N;
    int X, Total, i;

    while(scanf("%d", &N) == 1)
    {
        Total = 0;
        for(i = 0; i < N; i++)
        {
            scanf("%d", &X);
            Total = Total + X;
        }
        printf("%d\n", Total);
    }
    return 0;
}
```
- Enviar Solução:** A button at the bottom left, highlighted with a red box.


Conforme a Figura 21, todos os campos estão devidamente. Caso o aluno tente submeter uma solução sem o preenchimento adequado dos campos, o MOJO exibe uma mensagem de advertência e não permite a submissão até todos os campos estarem devidamente preenchidos.

Com todos os campos preenchidos adequadamente, o aluno pode clicar em “Enviar Solução” (em destaque na figura) para enviar sua solução para avaliação. No envio para avaliação, o MOJO realiza os 2 (dois) seguintes processos:

1. Envio do código-fonte para avaliação diretamente no juiz responsável pela questão. Neste processo, o MOJO envia os parâmetros necessários para o juiz da questão que realizará os processos necessários para avaliação automática (compilação, execução, testes).
2. Registro da submissão na base de dados do MOJO. Neste processo a ferramenta insere em uma tabela específica os dados relativos à submissão da solução, esses dados compreendem o identificador do aluno que submeteu a solução, o identificador da questão resolvida, um resultado (que por padrão é inserido como ‘Pendente’, sinalizando o status da submissão), a linguagem utilizada na codificação, data em que foi realizada a submissão e o código-fonte utilizado na solução. Esses dados serão utilizados para a visualização de resultados das submissões do aluno.

Com o envio da solução e os devidos processos realizados, o aluno é automaticamente redirecionado para a página de resultados da atividade respondida por ele. Esta página pode ser vista na Figura 22.

Figura 22 – Tela de resultados de submissão do aluno

ID Submissão	Título	Resultado	Linguagem	Data Submissão	Fonte
10940201	Baixe as trincas	Pendente	C	2014-01-25 17:06:12-02	

Conforme a Figura 22, é possível encontrar todas as informações relativas à submissão do aluno. Vale ressaltar que esta tela é a mesma acessada quando o aluno seleciona a opção “Resultados” na tela de visualização da questão.


A tela de resultados do aluno é o local onde ele pode acompanhar o andamento de suas submissões. Nesta tela, além do aluno ficar sabendo qual o status do resultado de sua submissão ele ainda pode consultar o código-fonte submetido por ele, pois ao clicar no ícone  um *popup* exibindo o código-fonte do aluno é mostrado, este exemplo é ilustrado na Figura 23.

Figura 23 – Popup exibindo código-fonte do aluno


```

#include "stdio.h"

int main()
{
    int N;
    int X, Total, i;

    while (scanf("%d", &N) == 1)
    {
        Total = 0;
        for (i = 0; i < N; i++)
        {
            scanf("%d", &X);
            Total = Total + X;
        }
        printf("%d\n", Total);
    }
    return 0;
}

```

Ainda de acordo com a Figura 22, é possível identificar o ícone , esse ícone é mostrado sempre que o aluno possui submissões ‘Pendentes’. Para solucionar o problema

da pendência o aluno deve clicar sobre este mesmo ícone. No clique do ícone, o resultado da submissão será atualizado, levando em consideração o identificador da submissão, com o resultado da avaliação realizada pelo juiz. A Figura 24 ilustra essa atualização.

Figura 24 – Resultado do aluno atualizado

ID Submissão	Título	Resultado	Linguagem	Data Submissão	Fonte
10940201	Baixe as trincas	Resposta Errada	C	9/03/2014 13:34:39	

Uma observação importante sobre a pendência no resultado de um aluno é que, enquanto o resultado estiver pendente a questão, na qual existe pendência, ficará bloqueada para novas submissões até que o aluno ou o professor atualize seu resultado. Um exemplo de questão bloqueada por pendência é ilustrado na Figura 25.

Figura 25 – Questão com pendência no resultado

Você acessou como **Mojo Teste** (Sair)
Português - Brasil (pt_br)

Página inicial ► Meus cursos ► Lab Dev 1 ► 23 janeiro - 29 janeiro ► Atividade de Teste no MOJO

Atividade de Teste no MOJO

Título: Baixe as trincas

Descrição: Os habitantes de uma pequena ilha caribenha na região conhecida como Triângulo das Bermudas adoram gastar suas noites quentes de verão jogando cartas. Como um tributo para a região onde moram, todos seus jogos de cartas têm alguma conexão com triângulos. Um dos jogos mais populares na ilha é conhecido como Trincas, e tem regras bem simples. O jogo é jogado entre dois amigos, com um baralho de cartas padrão. As cartas são distinguidas apenas por seus valores, de 1 (Ás) a 13 (Rei). As cartas são embaralhadas e colocadas em uma pilha no centro da mesa, viradas para baixo. Essa pilha é chamada de pilha. Os dois jogadores jogam em turnos. A cada turno, um jogador...

Nível: 3 - Normal
Linguagem: C, C#, C++, JAVA e PHP
Nota Máxima: 100

Resultados

Esta questão possui submissão pendente! ?

Continuar

Ou seja, o aluno só pode submeter nova solução para a questão quando o resultado da submissão anterior for definitivo, por definitivo entenda-se “Resposta Correta” ou

“Resposta Errada”, de acordo com a Tabela 2. Após a definição do resultado a questão volta a aceitar novas submissões e o ícone de pendência desaparece da visualização de resultados. O aluno pode realizar quantas submissões achar necessário.

Para que o aluno possa conferir todas suas atividades recentes no MOJO, ele tem disponível o bloco de “Atividade recente”, conforme mostrado na Figura 17 em destaque, junto às atividades na tela principal da disciplina. Por meio desse bloco o aluno terá acesso a todas as suas atividades para aquela disciplina, com base em parâmetros que o próprio aluno pode estabelecer.

Com a exibição das atividades recentes, o aluno pode ter uma visão de quais atividades ele realizou e como está seu desempenho em cada uma delas. Caso o aluno prefira, ele pode clicar em seu resultado e ser redirecionado diretamente para a página de resultados da atividade para resolver uma pendência, por exemplo. A Figura 26 mostra a tela que exhibe as atividades do aluno, destacando as atividades realizadas no período.

Figura 26 – Atividades recentes do aluno

Página inicial ► Meus cursos ► Lab Dev 1 ► Atividade recente ► Mojo Teste

Laboratório de Desenvolvimento de Sistemas 1: Mojo Teste

Filtros

Desde 21 janeiro 2014 03 50 Ativar

Mostrar a atividade recente

Fórum Fórum de notícias

23 janeiro - 29 janeiro

MOJO 1ª Atividade Lab Dev 1

Resultado : *Pendente*
Mojo Teste - quinta-feira, 23 janeiro 2014, 03:01

MOJO Atividade de Teste no MOJO

Resultado : *Resposta Errada*
Mojo Teste - sábado, 25 janeiro 2014, 19:06

Esta seção detalhou o comportamento da ferramenta na visão do aluno, desde a visualização da questão, passando pela submissão de solução para a questão até a

visualização dos resultados de submissões. A seção seguinte aborda a última etapa do processo ESA, a avaliação. Desde o momento em que o MOJO envia a solução para ser avaliada até o momento em que o resultado é disponibilizado para professores e alunos.

4.4.4 Avaliação Automática com o MOJO

O processo de avaliação automática é realizado exclusivamente pelos Juízes *Online* em seus servidores, este processo não será detalhado aqui, pois é um processo específico dos juízes e não realizado pelo MOJO. O que a ferramenta faz é enviar a solução para avaliação e captar o resultado dessa avaliação.

Quando o aluno clica em “Enviar” uma solução para a questão ele ativa uma importante operação do MOJO, o envio de código-fonte para avaliação no juiz. Uma vez que a solução do aluno foi enviada, o juiz responsável por avaliá-la recebe o código e realiza todo o procedimento de avaliação automática necessário.

Conforme explicado na seção *JUÍZES ONLINE*, o juiz compila o código-fonte e o executa. Durante a execução são utilizados dados formatados que funcionam como a entrada do programa, esses dados são processados e um resultado é fornecido. Com base nesse resultado é realizada a comparação dos resultados esperados com os resultados obtidos. O juiz *online* utiliza essas comparações para fornecer uma resposta apropriada como resultado da avaliação.

A resposta fornecida pelo juiz é a informação utilizada pelo MOJO para atualizar o resultado da submissão do aluno. Neste caso, o MOJO obtém, com base em alguns parâmetros (usuário do aluno e questão respondida), o resultado fornecido pelo juiz.

Figura 27 – Envio de código para avaliação e recebimento do resultado

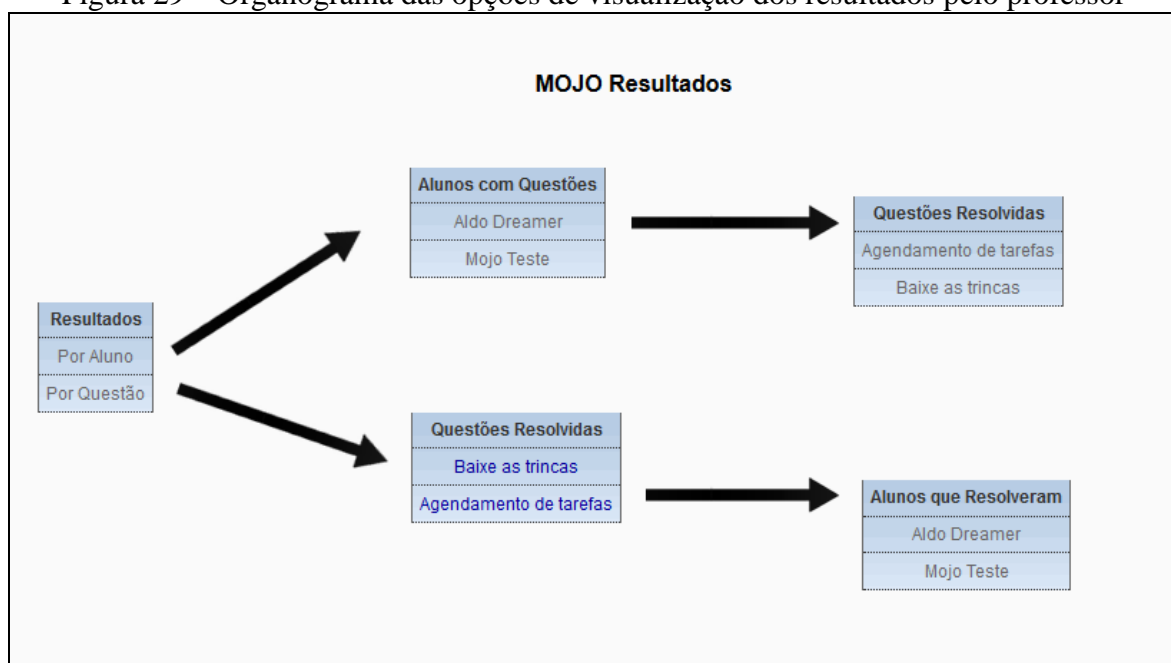


ao selecionar a questão, o professor pode visualizar os resultados daquele aluno para a questão.

- **Por questão:** Selecionando esta opção, o professor obtém a lista das questões que tiveram submissões para elas. Ao selecionar uma das questões da lista, o professor tem acesso a todos os alunos que enviaram solução para aquela questão e ao selecionar o aluno, o professor pode visualizar os resultados daquele aluno para a questão.

Para um melhor entendimento, a Figura 29 mostra um organograma de como estão divididas essas opções de visualização dos resultados pelo professor.

Figura 29 – Organograma das opções de visualização dos resultados pelo professor



Explicando melhor: o professor clica na opção de visualização que preferir, por exemplo, ao escolher percorrer o caminho do tipo de resultados “Por Aluno” ele é direcionado para uma tela que exhibe os “Alunos com Questões” que exhibe todos os alunos que resolveram questões. Por sua vez, ao selecionar um desses alunos o professor é direcionado para a tela de “Questões Resolvidas” que exhibe todas as questões que foram resolvidas pelo aluno selecionado na tela anterior. Ao final do caminho, o professor tem acesso à tela de visualização dos resultados da questão ou do aluno selecionado, caso tenha percorrido o outro caminho. A tela de resultados visualizada pelo professor é mostrada na Figura 30.

Figura 30 – Tela de visualização dos resultados pelo professor

Voce acessou como Balthazar Mesquita [Sair]
Português - Brasil (pt_br)

Página inicial ► Meus cursos ► Lab Dev 1

Navegação



Página inicial

- Minha página inicial
- Páginas do site
- Meu perfil
- Meus cursos
 - Lab Dev 2
 - Lab Dev 1
 - Participantes
 - Relatórios
 - Geral
 - 23 janeiro - 29 janeiro
 - 30 janeiro - 5 fevereiro


MOJO Resultados - Aldo Dreamer


1ª Atividade Lab Dev 1 - Agendamento de tarefas

Acrescentar um bloco
Adicionar...


ID	Juiz Online	Título	Resultado	Linguagem	Data Submissão	Fonte
10	SPOJ Online Judge	Agendamento de tarefas	Pendente	C#	2014-01-27 16:04:06-02	 

Continuar

É possível ver, em destaque, na Figura 30 que o professor, assim como o aluno, tem a opção de visualizar o código-fonte submetido para resolver a questão, para isso basta que ele clique no ícone . O funcionamento ocorre semelhante ao que acontece para o aluno, ao clicar no ícone um *popup* é aberto exibindo o código-fonte submetido.

Ainda, se o resultado da questão do aluno estiver como “Pendente”, o professor pode atualizar o resultado clicando no ícone . Este processo de atualização foi explicado na seção anterior e funciona praticamente da mesma forma para o professor e para o aluno. A diferença é que o professor pode atualizar o resultado de qualquer um de seus alunos, já o aluno só pode atualizar seu próprio resultado. Sempre que houver pendência de resultado este ícone fica visível para ambos (professor e aluno).

Essa tela de resultados pode ajudar o professor a acompanhar o progresso e desempenho do aluno na realização da atividade. Por meio desse acompanhamento, o professor pode identificar, com base nos erros e acertos, quais alunos enfrentam mais dificuldades em resolver as questões. Com isso o professor pode realizar um acompanhamento mais próximo desses alunos, avaliando os códigos submetidos por eles e dando dicas de melhorias, por meio da tela de *feedback*.

Ainda na tela de resultados, o professor tem a opção de enviar o *feedback* para cada submissão do aluno. Esta opção é exclusiva do professor e se torna disponível quando a submissão do aluno já tem um resultado definitivo (Resposta Correta ou Resposta Errada). Esta opção é indicada pelo ícone  que sinaliza que o professor pode enviar um *feedback* para aquela submissão.

A Figura 31 ilustra a situação em que esta funcionalidade é disponibilizada.

Figura 31 – Opção de envio de feedback pelo professor

ID	Juiz Online	Título	Resultado	Linguagem	Data Submissão	Fonte
10973789	SPOJ Online Judge	Soma	Resposta Correta	C	30/01/2014 21:25:39	

A Figura 31 destaca o resultado da submissão do aluno e o ícone para envio de *feedback* disponível, o professor ao clicar no ícone será direcionado para a tela de envio de *feedback* mostrada na Figura 32.

Figura 32 – Tela para envio de feedback pelo professor

A tela mostrada na Figura 32 exibe os campos necessários ao envio de *feedback* do professor. O professor deve preencher o campo “Feedback” com o que pode ser uma dica para resolução da questão, uma correção ou o que ele preferir, e então clica no botão “Enviar”. Para facilitar a utilização do professor, a tela ainda permite a exibição da descrição da questão e do código-fonte enviado (ícones em destaque na figura).

Após clicar em enviar, o *feedback* dado pelo professor é gravado e fica disponível para visualização na tela de resultados do aluno. O diferencial desta funcionalidade está no

fato do professor poder enviar *feedback* para cada uma das submissões realizadas pelo aluno e não apenas um *feedback* por atividade.


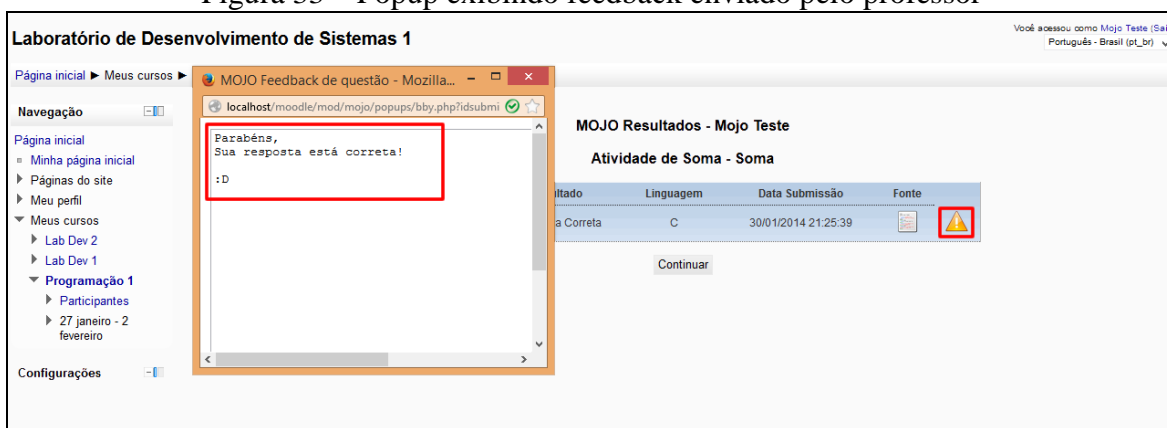
Para que o aluno tenha acesso ao *feedback* enviado pelo professor, o ícone  passa a ser exibido na tela de visualização de seus resultados, sinalizando que existe *feedback* do professor para aquela submissão. Quando o aluno clica no ícone em destaque é exibido um *popup* que mostra o *feedback* enviado pelo professor. A tela que ilustra esse *popup* é mostrada na Figura 33.

Figura 33 – Popup exibindo feedback enviado pelo professor



Com a possibilidade de visualizar o *feedback* enviado pelo professor (mostrado na Figura 33), o aluno pode colocar em prática as dicas e observações feitas pelo professor e a cada nova submissão ir melhorando seus resultados. Além disso, o aluno ainda vai sentir que o professor está acompanhando o andamento de suas atividades.

Para o professor, o mecanismo de envio de *feedback* melhora o acompanhamento de seus alunos, facilitando a comunicação e o envio de dicas e observações.

4.5 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo abordou o funcionamento do MOJO como ferramenta de integração entre os ambientes Moodle e Juízes *Online*. Integração que automatiza o processo ESA de atividades de programação. O capítulo ainda abordou um passo-a-passo de cada etapa do processo ESA. Mostrando desde a definição de uma atividade (Elaboração) pelo professor, passando pela Submissão de uma solução pelo aluno até a Avaliação automática dessa solução no juiz *online*. Além disso, foi explorada a parte de visualização de resultados e

feedback de submissões, onde é possível para o professor realizar um acompanhamento de seus alunos, e para o próprio aluno acompanhar o seu desempenho.

Para um melhor entendimento do funcionamento, foi disponibilizado um vídeo na Web onde é demonstrada a utilização do MOJO, o vídeo pode ser encontrado no seguinte link: <http://greenbeans.com.br/osvaldomesquita/mojo/>.

O próximo capítulo encerra a dissertação abordando as limitações encontradas na ferramenta e as primeiras impressões daqueles que conheceram a ferramenta. Além disso, o capítulo ainda destaca algumas das publicações que foram frutos dessa pesquisa, e encerra com as melhorias que se pretende realizar em futuras versões do MOJO.

5. CONCLUSÃO

No capítulo anterior foi apresentado um passo-a-passo do funcionamento da ferramenta, a automatização do processo ESA e todas as funcionalidades do MOJO disponíveis a professores e alunos, além de focar na integração da ferramenta ao Moodle como um módulo de atividades instalado na plataforma.

Este capítulo é o responsável por encerrar esta dissertação. Sendo assim, o capítulo aborda toda a parte de conclusões da pesquisa e desta 1ª versão do MOJO. O capítulo aborda as limitações da ferramenta, além de apresentar algumas publicações, derivadas da pesquisa, que foram responsáveis por apresentar o MOJO em renomados eventos nacionais e até mesmo um evento internacional. O capítulo se encerra com as devidas considerações finais e propostas de trabalhos a serem realizados em futuras versões.

5.1 LIMITAÇÕES DA FERRAMENTA

O desenvolvimento da pesquisa encontrou algumas dificuldades pelo caminho. Para que fosse possível continuar com o desenvolvimento da pesquisa, superando essas dificuldades, algumas implementações bem específicas precisaram ser feitas e algumas necessidades precisaram ser atendidas.

As dificuldades encontradas trouxeram com elas algumas limitações ao MOJO, sendo a principal delas a dependência da estrutura HTML das páginas dos juízes utilizados na ferramenta. Isso significa que, caso haja uma atualização nas páginas dos juízes, alterando sua estrutura (nomes de campos, variáveis, tabelas) o MOJO será diretamente impactado por isso, sendo necessária uma atualização da ferramenta com base nas alterações realizadas no juiz.

Isso não aconteceria se houvesse, por parte dos juízes, a disponibilização de API's ou funcionalidades via *Web Services*. Sobre isso, pretende-se, como um trabalho futuro, apresentar aos Juízes *Online* uma proposta de padronização no acesso às suas funcionalidades, facilitando a integração e diminuindo essa dependência da estrutura da página. De forma que nenhuma das partes se sinta prejudicada, pois o MOJO tem como único, e exclusivo objetivo, auxiliar no processo de ensino/aprendizagem em disciplinas de programação.

Embora possua suas limitações, o MOJO já foi bem divulgado. Essa divulgação aconteceu por meio de publicações em eventos e periódicos, na área de informática na educação, e é sobre essas publicações que trata a seção seguinte.

5.2 PUBLICAÇÕES

Durante o desenvolvimento da pesquisa, e da ferramenta, foram feitas algumas publicações em eventos relevantes na área de informática na educação. Essas publicações apresentavam a proposta da ferramenta e como ela pretende melhorar o ensino/aprendizagem de programação auxiliando professores e alunos. As principais publicações são abordadas nesta seção.

O trabalho e a ferramenta desenvolvida apresentados e descritos nesta dissertação foram apresentados, em junho de 2013, no X Congresso Brasileiro de Ensino Superior a Distância (ESUD), em Belém, e foi publicado nos anais do evento com o título “**MOJO: Uma Ferramenta para Auxiliar o Professor em Disciplinas de Programação**” (Chaves *et al.*, 2013a).

Com a pesquisa já mais amadurecida, ela foi apresentada em novembro de 2013 no XXIV Simpósio Brasileiro de Informática na Educação (SBIE), em Campinas, e com o título: “**Integrando Moodle e Juízes Online no Apoio a Atividades de Programação**” (Chaves *et al.*, 2013b) foi publicada no anais do evento.

O trabalho também foi apresentado, em dezembro de 2013, no XXII Ciclo de Palestras Novas Tecnologias, Rio Grande do Sul, e publicado no Vol.11, Nº 3 (2013) na RENOTE - Revista Novas Tecnologias na Educação, com o título “**Uma Ferramenta Baseada em Juízes Online para Apoio às Atividades de Programação de Computadores no Moodle**” (Chaves *et al.*, 2013c).

O trabalho ainda teve publicações apresentadas em dois eventos do Congresso da Sociedade Brasileira de Computação (CSBC) em 2013: o II Workshop de Desafios da Computação Aplicada à Educação (DesafIE!) e o XXVI Workshop sobre Educação em Computação (WEI).

Além disso, um dos artigos derivados da pesquisa foi agraciado com o prêmio de 2º lugar na categoria *Best Paper* na Escola Potiguar de Computação e suas Aplicações (EPOCA) em novembro de 2013, tendo sido publicado nos anais do evento com o título “**MOJO: Uma Ferramenta para Integrar Juízes Online ao Moodle**”. A premiação

nesse evento rendeu o convite para a publicação de uma versão estendida do artigo na revista HOLOS, uma publicação *online* do Instituto Federal do Rio Grande do Norte (IFRN), que tem como objetivo publicar artigos que contribuam para o estudo de temas interdisciplinares e áreas afins. Este artigo foi submetido e no momento aguarda publicação.

A pesquisa ainda teve uma publicação em evento internacional na XVIII Conferência Internacional sobre Informática na Educação (TISE) em dezembro de 2013. Além desses eventos, a pesquisa também foi apresentada na III Semana de Computação, organizada e realizada pela Universidade do Estado do Rio Grande do Norte (UERN) em dezembro de 2013.

A Tabela 3 exibe todas as publicações derivadas desta pesquisa.

Tabela 3 – Tabela de publicações

EVENTO/PERIÓDICO	ARTIGO	STATUS
X Congresso Brasileiro de Ensino Superior a Distância (ESUD)	MOJO: Uma Ferramenta para Auxiliar o Professor em Disciplinas de Programação	Publicado
II Workshop de Desafios da Computação Aplicada à Educação (Desafie!)	Uma Ferramenta para Auxiliar na Elaboração, Submissão e Correção de Atividades Práticas em Disciplinas de Programação	Publicado
XXVI Workshop sobre Educação em Computação (WEI)	MOJO: uma ferramenta de auxílio à elaboração, submissão e correção de atividades em disciplinas de programação	Publicado
XXIV Simpósio Brasileiro de Informática na Educação (SBIE)	Integrando Moodle e Juízes Online no Apoio a Atividades de Programação	Publicado
RENTE - Revista Novas Tecnologias na Educação	Uma Ferramenta Baseada em Juízes Online para Apoio às Atividades de Programação de Computadores no Moodle	Publicado
Escola Potiguar de Computação e suas Aplicações (EPOCA)	MOJO: Uma Ferramenta para Integrar Juízes Online ao Moodle	Publicado
XVIII Conferência Internacional sobre Informática na Educação (TISE)	Uma Ferramenta de Auxílio ao Processo de Ensino-Aprendizagem para Disciplinas de Programação de Computadores	Publicado
Revista HOLOS	MOJO: Uma Ferramenta para Integrar Juízes Online ao Moodle no Apoio ao Ensino e Aprendizagem de Programação	Aguardando publicação

Embora a ferramenta desenvolvida ainda não tenha sido liberada para utilização, e com isso não tendo sido possível a coleta de dados sobre sua utilização prática fora do âmbito laboratorial, foi possível coletar, nos eventos em que a mesma foi apresentada, alguns dados sobre a importância e relevância do MOJO. Esses dados são apresentados como as primeiras impressões da ferramenta na seção seguinte.

5.3 PRIMEIRAS IMPRESSÕES DA FERRAMENTA

A ferramenta ainda não foi liberada para utilização prática de professores e alunos. Portanto, as impressões abordadas nesta seção são fundamentadas nas diversas exposições realizadas sobre a ferramenta durante os eventos em que a mesma foi apresentada.

Para coleta dos dados iniciais da ferramenta, foram realizadas entrevistas com um total de 7 (sete) professores de disciplinas de programação, para os quais foram apresentados o MOJO e seus objetivos. Nessas entrevistas, após a devida apresentação da ferramenta foram feitas as seguintes perguntas a cada um dos professores:

- 1) Você concorda que com a diminuição da sobrecarga de tarefas o professor pode melhorar o acompanhamento de seus alunos?
- 2) Você acredita que o MOJO pode reduzir o tempo gasto pelo professor no processo ESA?
- 3) Qual sua opinião sobre a utilização de questões já prontas disponibilizadas pelos Juízes *Online*?
- 4) Você concorda que o MOJO pode auxiliar no fornecimento mais rápido de *feedback* ao aluno?

Com base nas respostas dadas pelos entrevistados, foram obtidos os seguintes dados apresentados na Tabela 4.

Tabela 4 – Primeiras impressões da ferramenta

Nº DE PROFESSORES	AVALIAÇÃO DOS PROFESSORES
7	Professores concordaram que, com a diminuição na sobrecarga de tarefas, é possível um melhor acompanhamento dos alunos.
6	Professores concordaram que o MOJO pode ajudar a fornecer um <i>feedback</i> mais rápido ao aluno.
5	Professores concordaram que o MOJO pode reduzir o tempo gasto no processo de Elaboração, Submissão e Avaliação.
3	Professores fizeram observações quanto à legibilidade do código em questões já prontas.
2	Professores sentiram falta da possibilidade de poderem elaborar suas próprias atividades e terem as mesmas avaliadas automaticamente no MOJO.
2	Professores fizeram observações sobre a clareza no resultado da avaliação realizada pelos juízes.

Percebe-se, com base nos dados apresentados, que os professores acreditam que com a utilização da ferramenta, acarretando na diminuição da sobrecarga de trabalho do professor e no *feedback* mais rápido aos alunos, é possível realizar um melhor acompanhamento dos alunos, principalmente dos que encontram mais dificuldades no aprendizado da disciplina.

Alguns dos professores fizeram observações quanto a possíveis problemas de legibilidade do código (similaridades, cópia e cola) que possam ocorrer. Além disso, alguns sinalizaram que preferem elaborar suas próprias questões, mas que um processo de avaliação automática para estas questões pode ser bem útil.

Ainda foram feitas observações sobre o resultado da avaliação realizada pelos juízes, segundo os entrevistados o resultado não é claro o suficiente e precisa ser melhorado. Inclusive essa melhoria já foi realizada, uma vez que para dar maior clareza ao resultado, o MOJO passou a padronizá-los de acordo com a Tabela 2.

Todas as considerações apontadas pelos entrevistados são de fundamental importância para futuras melhorias na ferramenta, assim como é importante uma avaliação prática da mesma por professores e alunos para corroborar as vantagens de sua utilização. A seção seguinte aborda essas melhorias futuras e as considerações finais sobre a ferramenta.

5.4 CONSIDERAÇÕES FINAIS

A integração dos Juízes *Online* com o Moodle, pelo MOJO, tem como objetivo fornecer um ambiente coeso que visa diminuir a sobrecarga de trabalho do professor, automatizando o processo de Elaboração, Submissão e Avaliação de atividades de programação. Seja ele professor na modalidade de ensino presencial, semipresencial ou à distância.

Como resultado da utilização desse ambiente único, espera-se a melhoria na qualidade do ensino/aprendizagem em disciplinas de programação, tendo em vista que com a redução do tempo gasto pelo professor no processo ESA, ele pode utilizar o tempo ganho para realizar um melhor acompanhamento dos alunos.

Além disso, quanto à questão da evasão de curso, sabe-se que outros fatores como, por exemplo, problemas pessoais e inaptidão para o curso podem ser responsáveis por reprovações e desistências. Mas, se considerarmos que, com o aumento da disponibilidade

de tempo, propiciado pela utilização da ferramenta, o professor pode atuar de maneira a evitar o desestímulo ou eventual sensação de abandono por parte do aluno, o MOJO estará, desta forma, auxiliando a evitar a evasão do curso.

5.4.1 Trabalhos Futuros

A ferramenta ainda não foi liberada oficialmente para utilização prática. Por isso, para uma real avaliação do seu desempenho junto a professores e alunos, é necessário realizar uma avaliação prática. Em relação a isso, a primeira versão estável da ferramenta já está em fase de liberação, ela então será disponibilizada em uma turma regular que esteja cursando alguma disciplina de programação. Desta forma, será possível verificar a aplicação prática do MOJO e possíveis ajustes a serem realizados em futuras versões.

Como trabalhos futuros, pretende-se integrar ao MOJO um Laboratório Virtual de Programação, como o VPL (VPL, 2014), que além de auxiliar o aluno na edição e teste de seus códigos, também permitirá ao professor elaborar suas próprias questões e avaliá-las automaticamente. Com este laboratório virtual também se pretende melhorar os problemas de legibilidade de código apontados nas primeiras impressões da ferramenta.

Além disso, pretende-se criar um indicador de dificuldade, com base na quantidade de erros e acerto do aluno, que manterá o professor informado sobre as dificuldades encontradas pelos alunos. Também está prevista a categorização das questões quanto ao que é trabalhado em cada uma delas como, por exemplo, questões que trabalham com manipulação de *array*, *strings* ou laços de repetição.

A inclusão de outros Juízes *Online* também está prevista, ampliando ainda mais o a quantidade de questões no MOJO, inclusive com questões de língua estrangeira.

5.4.2 Concluindo

Esta dissertação descreve um trabalho que se contextualiza na utilização de recursos computacionais para oferecer suporte ao professor e aluno de disciplinas de programação. Seu desenvolvimento teve como maior motivação a escassez de recursos humanos que impede que o ensino/aprendizado se desenvolva com qualidade, escassez que acarreta desmotivação, desinteresse e uma possível evasão do curso por parte do aluno.

Essa realidade é percebida pela sobrecarga de tarefas do professor, que pode lecionar em uma ou mais turmas de numerosos alunos. De maneira bem específica, este trabalho propõe um ambiente que integra os sistemas de Juízes *Online* à plataforma Moodle. Esta tecnologia irá auxiliar o professor a melhor gerenciar os seus esforços no acompanhamento das atividades de seus alunos. Atividades que exigem maior atenção e que pode, em turmas numerosas, dificultar o aprendizado de qualidade. O auxílio necessário será dado pela automatização de um processo já conhecido pelo professor, o processo ESA (Elaboração, Submissão e Avaliação).

Muitas ferramentas são desenvolvidas com a proposta de apoiar a educação, incluindo o apoio a atividades de programação. Muitas dessas ferramentas foram e vem sendo construídas com diferentes propósitos e quando possibilitam que suas funcionalidades sejam integradas entre si, podem poupar o investimento no desenvolvimento de recursos que já existem, mas que operam de forma independente.

A integração de funcionalidades, presentes em diferentes ferramentas, pode fornecer ao professor o acesso a um ambiente coeso que possua os recursos necessários ao acompanhamento das atividades em um determinado curso. Assim, é possível reduzir a sobrecarga relativa à utilização de diversas ferramentas.

Com base nessa perspectiva, a principal contribuição desta pesquisa é um novo módulo para o Moodle que permite a integração de dois ambientes *Web*: o Moodle, plataforma bastante utilizada no ensino à distância, e o sistema de Juízes *Online*, sistemas utilizados em maratonas de programação e conhecidos por sua variedades de questões e pelo processo de avaliação automática de códigos-fonte.

A ferramenta desenvolvida, nomeada como Módulo de Integração com os Juízes *Online* (MOJO), disponibiliza, no Moodle, as questões e o processo de avaliação automática dos Juízes *Online*. A ferramenta é integrada a plataforma Moodle pela instalação de um *plugin* que passa a oferecer um novo tipo de atividades que o professor pode propor a seus alunos.

O ambiente integrado visa diminuir a sobrecarga de tarefas do professor, uma vez que ele não precisa fazer uso de várias ferramentas ao mesmo tempo, pois o ambiente integrado contém as informações e recursos necessários. Como consequência, espera-se a melhoria na qualidade do ensino/aprendizado de programação, uma vez que o professor terá uma maior disponibilidade de tempo para dar atenção aos alunos, principalmente os que enfrentam maior dificuldade na disciplina.

Quanto à instalação e utilização da ferramenta é importante ressaltar que: devido às mudanças ocorridas na padronização para o desenvolvimento de novos módulos para o Moodle a partir da versão 2.0, o MOJO não funciona em versões da plataforma anteriores a esta, pois a ferramenta foi desenvolvida seguindo os novos padrões de desenvolvimento de módulos.

A primeira versão da ferramenta está disponível para *download* no seguinte link: <http://greenbeans.com.br/osvaldomesquita/mojo/> e em breve na comunidade oficial do Moodle.

REFERÊNCIAS³

ACM-ICPC. The ACM-ICPC International Collegiate Programming Contest. Disponível em: <<http://icpc.baylor.edu/>>. Acesso em: 20 de jan. de 2014.

AMARAL E. M. H; ÁVILA, B; ZEDNIK, H; TAROUCO, L. Laboratório Virtual de Aprendizagem: Uma Proposta Taxonômica. **RENOTE: Revista Novas Tecnologias na Educação**, v. 9 n. 2, 2011.

CAMPOS, C. P; FERREIRA, C. E. BOCA: Um sistema de apoio para competições de programação. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI 2004), 12, 2004, Salvador.

CHAVES, J. O. M; CASTRO, A. F; LIMA, R. W; LIMA, M. V. A; FERREIRA, K. H. A. MOJO: Uma Ferramenta para Auxiliar o Professor em Disciplinas de Programação. In: CONGRESSO BRASILEIRO DE ENSINO SUPERIOR A DISTÂNCIA (ESUD 2013), 10, 2013, Belém.

_____. Integrando Moodle e Juízes Online no Apoio a Atividades de Programação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2013), 24, 2013, Campinas.

_____. Uma Ferramenta Baseada em Juízes Online para Apoio às Atividades de Programação de Computadores no Moodle. **RENOTE: Revista Novas Tecnologias na Educação**, v. 11 n. 3, 2013.

EDWARDS, S. H. Using software testing to move students from trial-and-error to reflection-in-action. **SIGCSE Bulletin**, v. 36, n. 1, março 2004, p. 26–30.

FRANÇA, A. B. **Sistema de apoio a atividades de laboratório de programação com suporte ao balanceamento de carga e controle de plágio**. 145 f. Dissertação (Mestrado em Engenharia de Teleinformática) – Programa de Pós-Graduação em Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza, 2012.

INC., T. TopCoder. Disponível em: <<http://www.topcoder.com/>>. Acesso em: 20 de jan. de 2014.

KUMAR S; GANKOTIYA, A. K; DUTTA, K. A Comparative Study of Moodle with other e-Learning Systems. In: INTERNATIONAL CONFERENCE ON ELETRONICS COMPUTER TECHNOLOGY (ICECT 2011), 3, 2011, Kanyakumari, p. 414-418.

MOREIRA, M. P; FAVERO, E. L. Um Ambiente Para Ensino de Programação com Feedback Automático de Exercícios. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI 2009), 17, 2009, Belém.

³De acordo com: ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 6023: informação e documentação: referências: elaboração. Rio de Janeiro, 2002.

MOTA, M. P; PEREIRA, L. W. K; FAVERO, E. L. JavaTool: Uma Ferramenta Para Ensino de Programação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2009), 20, 2009, Florianópolis.

ONLINEJUDGE. Online judge plugin for Moodle. Disponível em: <<https://github.com/hit-moodle/onlinejudge/>>. Acesso em: 21 de jan. de 2014.

SANDBOX. Sandbox. Disponível em: <<https://github.com/openjudge/sandbox/>>. Acesso em: 21 de jan. de 2014.

SANTOS, J. C. S; RIBEIRO, A. R. L. JOnline: proposta preliminar de um juiz online didático para o ensino de programação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2011), 22, 2011, Aracaju.

SIROTHEAU, S; BRITO, S. R; SILVA, A. S; ELIASQUEVICI, M. K; FAVERO, E. L; TAVARES, O. L. Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de autoavaliação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2011), 22, 2011, Aracaju.

SOUSA, G. R; PORTO, V. S. **Criação de Plugins para a Ferramenta Moodle**. 80 f. Monografia (Bacharel em Ciência da Computação) – Curso de Bacharelado em Ciência da Computação, Universidade Católica de Brasília, Taguatinga, 2010.

SOUZA, D. M; MALDONADO, J. C; BARBOSA, E. F. Aspectos de Desenvolvimento e Evolução de um Ambiente de Apoio ao Ensino de Programação e Teste de Software. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE 2012), 23, 2012, Rio de Janeiro.

SPHERE RESEARCH LABS. IDE ONE. Disponível em <<http://ideone.com/>>. Acesso em: 22 de jan. de 2014.

_____. SPOJ Brasil. Disponível em <<http://br.spoj.pl/>>. Acesso em: 16 de jan. de 2014.

TAKAHASHI, F. Matemática e ciências da computação têm alta taxa de abandono. **Folha de São Paulo**, São Paulo, 06 abr. 2009. Disponível em: <<http://www1.folha.uol.com.br/folha/educacao/ult305u546576.shtml>>. Acesso em: 12 de jan. de 2014.

THE PHP GROUP. PHP: Hypertext Preprocessor. Disponível em: <<http://www.php.net/>>. Acesso em: 13 de jan. de 2014.

_____. Biblioteca Cliente URL. Disponível em: <http://php.net/manual/pt_BR/book.curl.php>. Acesso em: 20 de jan. de 2014.

THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL. Disponível em: <<http://www.postgresql.org/>>. Acesso em: 13 de jan. de 2014.

TIMUS ONLINE JUDGE TEAM. Timus Online Judge. Disponível em: <<http://acm.timus.ru/>>. Acesso em: 21 de jan. de 2014.

TIOBE SOFTWARE BV. TIOBE Index for January 2014. Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>. Acesso em: 23 de jan. de 2014.

UNIVERSIDAD DE VALLADOLID. UVA Online Judge. Disponível em: <<http://uva.onlinejudge.org/>>. Acesso em: 06 de jan. de 2014.

VPL. Virtual Programming Lab. Disponível em: <<http://vpl.dis.ulpgc.es/>>. Acesso em: 20 de jan. de 2014.

W3SCHOOLS. JavaScript Tutorial. Disponível em: <<http://www.w3schools.com/js/default.asp>>. Acesso em: 06 de jan. de 2014.

_____. HTML Tutorial. Disponível em: <<http://www.w3schools.com/html/default.asp>>. Acesso em: 12 de jan. de 2014.

YUSOF, N; ZIN, N. A. M; ADNAN, N. S. Java programming assessment tool for assignment module in moodle e-learning system. **Procedia - Social and Behavioral Sciences**, v. 56, 2012, p. 767-773.

ZHIGANG, S; XIAOHONG, S; NING, Z; YANYU, C. Moodle Plugins for Highly Efficient Programmin Courses. In: MOODLE RESEARCH CONFERENCE, 1, 2012, Heraklion, p. 157-163.

APÊNDICE A – NOÇÕES BÁSICAS PARA O DESENVOLVIMENTO DE NOVOS MÓDULOS

O Moodle permite sua extensão por meio da criação de novos módulos que adicionam novas funcionalidades à plataforma. Mas existem algumas regras básicas que devem ser seguidas para que um novo módulo seja devidamente desenvolvido e reconhecido pelo Moodle.

Como o foco deste trabalho não é ensinar como desenvolver um novo módulo desde o princípio, e sim apresentar o novo módulo já desenvolvido, aqui são apresentadas apenas algumas das regras básicas para a criação de módulos. Para um maior aprofundamento, recomenda-se a visita ao site⁴ que contém a documentação específica para quem pretende desenvolver novas funcionalidades e adicioná-las ao Moodle.

Feito o devido esclarecimento, a seguir são listadas as regras básicas que o desenvolvimento de um novo módulo deve respeitar para ser reconhecido pela plataforma:

- Os arquivos que fazem parte do novo módulo (ou bloco) devem estar num mesmo diretório com um nome que, por padrão, não deve conter números ou caracteres especiais e deve ser uma palavra única, sem espaços em branco. Esse novo diretório deve ficar localizado no Moodle dentro da pasta /mod (no caso de módulos) ou na pasta /blocks (caso seja bloco).
- O acesso a base de dados, por padrão, deve ser feito utilizando as funções que constam no arquivo **datalib.php** localizado na pasta /lib do Moodle. Esse acesso padronizado auxilia o acesso a diferentes bases de dados, pois não se restringe às especificidades da linguagem do SGBD utilizado.
- Os textos devem ser manuseados por meio das funções pré-definidas pelo Moodle (**get_string()** e **print_string()**). Desta forma, é importante que se crie palavras-chave para os textos, uma espécie de dicionário que facilite a tradução dos mesmos para outras línguas. As funções indicadas buscam as palavras-chave no arquivo de língua localizado no diretório /lang do módulo. Os arquivos

⁴ http://docs.moodle.org/dev/Developer_documentation

contendo as variáveis de tradução devem ter o nome do módulo seguido da extensão **.php**, como, por exemplo, **mojo.php** localizado em `/mod/mojo/lang/pt_br`. É importante lembrar que é obrigatória a existência de pelo menos um diretório de linguagem no novo módulo em desenvolvimento e, por padrão, esse diretório é o de língua inglesa, por exemplo, `/mod/mojo/lang/en`.

- Se for necessária a criação de novas tabelas na base de dados para o módulo, deve existir no diretório do novo módulo uma pasta denominada `/db` onde deve estar localizado um arquivo denominado **install.xml**. Esse arquivo deve conter toda a estrutura das tabelas a serem instaladas. O APÊNDICE B traz o arquivo **.xml** descrevendo a estrutura das tabelas necessárias à ferramenta desenvolvida na pesquisa.

Depois de cumpridas as etapas básicas, o novo módulo deve estar disponível para instalação na tela principal do administrador no Moodle, conforme mostra a Figura 34.

Figura 34 – Instalação de novo módulo ao Moodle

Número de plugins que requerem sua atenção: 1						
Mostrar a lista completa de plugins instalados						
Nome do plugin	Diretório	Fonte	Versão atual	Nova versão	Requer	Status
Módulos de atividades						
MOJO	/mod/mojo	Contribuições		2014010800		A ser instalado

[Recarregar](#)

Na tela de instalação, mostrada na Figura 34, basta que o administrador clique no botão “Atualizar base de dados do Moodle agora” e o módulo é instalado.

Figura 35 – Módulos de atividades instalados

Página inicial ► Administração do site ► Plugins ► Módulos de atividades ► Gerenciar atividades

Navegação

Página inicial

- Minha página inicial
- Páginas do site
- Meu perfil
- Meus cursos

Favoritos do administrador

Marcar esta página

Configurações

- Minhas configurações de perfil
- ▼ Administração do site
 - ⚙ Avisos
 - ⚙ Inscrição
 - ⚙ Opções avançadas
 - Usuários

Atividades

Módulo de atividades	Atividades	Versão	Ocultar/Mostrar	Excluir	Configurações
Tarefa	0	2012112900	👁	Excluir	Configurações
Tarefa (2.2)	0	2012112900	👁	Excluir	Configurações
Livro	0	2012112900	👁	Excluir	Configurações
Chat	0	2012112900	👁	Excluir	Configurações
Escolha	0	2012112900	👁	Excluir	
Base de dados	0	2012112901	👁	Excluir	Configurações
Pesquisa	0	2012112900	👁	Excluir	Configurações
Pasta	0	2012112900	👁	Excluir	Configurações
Fórum	1	2012112901			Configurações
Glossário	0	2012112900	👁	Excluir	Configurações
Conteúdo do pacote IMS	0	2012112900	👁	Excluir	Configurações
Rótulo	0	2012112900	👁	Excluir	
Lição	0	2012112900	👁	Excluir	Configurações
MOJO	0	2014010800	👁	Excluir	
Página	0	2012112900	👁	Excluir	Configurações

Na instalação, o Moodle vai realizar a criação das tabelas necessárias, atualizar as demais tabelas referentes à utilização do novo módulo, dentre outros procedimentos. Depois da instalação o novo módulo vai figurar entre os módulos de atividades disponíveis, como ilustrado na Figura 35.

Explicado, de forma básica, como se dá a criação de um novo módulo, vale ressaltar que a versão da plataforma Moodle utilizada no desenvolvimento da ferramenta foi a Moodle 2.4.

APÊNDICE B – ESTRUTURA DAS TABELAS EM .XML

```

<?xml version="1.0" encoding="UTF-8" ?>
<XMLDB PATH="mod/mojo/db" VERSION="2014010800"
  COMMENT="XMLDB file for Moodle mod/mojo"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../../../../../lib/xmldb/xmldb.xsd">

<TABLES>
  <TABLE NAME="mojo" COMMENT="mojo table" NEXT="mojo_aluno">
    <FIELDS>
      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="true" NEXT="name"/>
      <FIELD NAME="name" TYPE="char" LENGTH="100" NOTNULL="true"
        SEQUENCE="false" PREVIOUS="id" NEXT="course"/>
      <FIELD NAME="course" TYPE="int" LENGTH="10" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="false" PREVIOUS="name" NEXT="judgeid"/>
      <FIELD NAME="judgeid" TYPE="int" LENGTH="10" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="false" PREVIOUS="course"
        NEXT="onlinejudge"/>
      <FIELD NAME="onlinejudge" TYPE="char" LENGTH="50" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="false" PREVIOUS="judgeid"
        NEXT="questionid"/>
      <FIELD NAME="questionid" TYPE="int" LENGTH="10" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="false" PREVIOUS="onlinejudge"
        NEXT="questiontitle"/>
      <FIELD NAME="questiontitle" TYPE="char" LENGTH="100" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="false" PREVIOUS="questionid"
        NEXT="questioncode"/>
      <FIELD NAME="questioncode" TYPE="char" LENGTH="20" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="false" PREVIOUS="questiontitle"
        NEXT="maxgrade"/>
      <FIELD NAME="maxgrade" TYPE="int" LENGTH="10" NOTNULL="true"
        UNSIGNED="true" SEQUENCE="false" PREVIOUS="questioncode"
        NEXT="timemodified"/>
      <FIELD NAME="timemodified" TYPE="int" LENGTH="10" NOTNULL="false"
        UNSIGNED="false" SEQUENCE="false" PREVIOUS="maxgrade"/>
    </FIELDS>
    <KEYS>
      <KEY NAME="primary" TYPE="primary" FIELDS="id"/>
    </KEYS>
  </TABLE>
  <TABLE NAME="mojo_aluno" COMMENT="students table"
    PREVIOUS="mojo" NEXT="mojo_juiz">
    <FIELDS>
      <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
        SEQUENCE="true" NEXT="idaluno"/>
      <FIELD NAME="idaluno" TYPE="int" LENGTH="10" NOTNULL="true"
        SEQUENCE="false" PREVIOUS="id" NEXT="nomealuno"/>
      <FIELD NAME="nomealuno" TYPE="char" LENGTH="100" NOTNULL="true"
        SEQUENCE="false" PREVIOUS="idaluno" NEXT="sobrenomealuno"/>
      <FIELD NAME="sobrenomealuno" TYPE="char" LENGTH="100" NOTNULL="true"
        SEQUENCE="false" PREVIOUS="nomealuno" NEXT="email"/>
      <FIELD NAME="email" TYPE="char" LENGTH="100" NOTNULL="true"
        SEQUENCE="false" PREVIOUS="sobrenomealuno" NEXT="usuario"/>
      <FIELD NAME="usuario" TYPE="char" LENGTH="100" NOTNULL="true"
        SEQUENCE="false" PREVIOUS="email" NEXT="senha"/>
      <FIELD NAME="senha" TYPE="char" LENGTH="100" NOTNULL="true"
        SEQUENCE="false" PREVIOUS="usuario"/>
    </FIELDS>

```

```

<KEYS>
  <KEY NAME="primary" TYPE="primary" FIELDS="id" NEXT="user"/>
  <KEY NAME="user" TYPE="foreign" FIELDS="idaluno" REFTABLE="user"
    REFFIELDS="id" PREVIOUS="primary"/>
</KEYS>
</TABLE>

<TABLE NAME="mojo_juiz" COMMENT="judges table"
  PREVIOUS="mojo_aluno" NEXT="mojo_questao">
  <FIELDS>
    <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="true" NEXT="nome"/>
    <FIELD NAME="nome" TYPE="char" LENGTH="200" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="id" NEXT="urlquestoes"/>
    <FIELD NAME="urlquestoes" TYPE="char" LENGTH="255"
      NOTNULL="true" SEQUENCE="false" PREVIOUS="nome"/>
  </FIELDS>
  <KEYS>
    <KEY NAME="primary" TYPE="primary" FIELDS="id"/>
  </KEYS>
</TABLE>

<TABLE NAME="mojo_questao" NEXT="mojo_resultado">
  <FIELDS>
    <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="true" NEXT="idjuizresponsavel"/>
    <FIELD NAME="idjuizresponsavel" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="id" NEXT="idquestaojuiz"/>
    <FIELD NAME="idquestaojuiz" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="idjuizresponsavel" NEXT="tituloquestao"/>
    <FIELD NAME="tituloquestao" TYPE="char" LENGTH="200" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="idquestaojuiz" NEXT="descricao"/>
    <FIELD NAME="descricao" TYPE="text" NOTNULL="true" SEQUENCE="false"
      PREVIOUS="tituloquestao" NEXT="nivelquestao"/>
    <FIELD NAME="nivelquestao" TYPE="char" LENGTH="50" NOTNULL="false"
      SEQUENCE="false" PREVIOUS="descricao" NEXT="idurl"/>
    <FIELD NAME="idurl" TYPE="char" LENGTH="20" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="nivelquestao" NEXT="linguagempermitida"/>
    <FIELD NAME="linguagempermitida" TYPE="char" LENGTH="200" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="idurl"/>
  </FIELDS>
  <KEYS>
    <KEY NAME="primary" TYPE="primary" FIELDS="id" NEXT="judge"/>
    <KEY NAME="judge" TYPE="foreign" FIELDS="idjuizresponsavel"
      REFTABLE="mojo_juiz" REFFIELDS="id" PREVIOUS="primary"/>
  </KEYS>
</TABLE>

<TABLE NAME="mojo_resultado" PREVIOUS="mojo_questao">
  <FIELDS>
    <FIELD NAME="id" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="true" NEXT="idsubmissao"/>
    <FIELD NAME="idsubmissao" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="id" NEXT="idaluno"/>
    <FIELD NAME="idaluno" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="idsubmissao" NEXT="idquestao"/>
    <FIELD NAME="idquestao" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="idaluno" NEXT="resultado"/>
    <FIELD NAME="resultado" TYPE="char" LENGTH="50" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="idquestao" NEXT="datasubmissao"/>
    <FIELD NAME="datasubmissao" TYPE="int" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="resultado" NEXT="linguagem"/>
    <FIELD NAME="linguagem" TYPE="char" LENGTH="10" NOTNULL="true"
      SEQUENCE="false" PREVIOUS="datasubmissao" NEXT="sourcecode"/>
    <FIELD NAME="sourcecode" TYPE="text" NOTNULL="true" SEQUENCE="false"
      PREVIOUS="linguagem" NEXT="feedback"/>
  </FIELDS>

```



```
<FIELD NAME="feedback" TYPE="text" NOTNULL="false"
  SEQUENCE="false" PREVIOUS="sourcecode"/>
</FIELDS>
<KEYS>
  <KEY NAME="primary" TYPE="primary" FIELDS="id" NEXT="aluno"/>
  <KEY NAME="aluno" TYPE="foreign" FIELDS="idaluno"
    REFTABLE="mojo_aluno" REFFIELDS="id" PREVIOUS="primary"
    NEXT="questao"/>
  <KEY NAME="questao" TYPE="foreign" FIELDS="idquestao"
    REFTABLE="mojo_questao" REFFIELDS="id" PREVIOUS="aluno"/>
</KEYS>
</TABLE>
</TABLES>
</XMLDB>
```