

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

ALEX BADO

MÓDULO PARA AVALIAÇÃO AUTOMÁTICA DE PROGRAMAS PARA MOODLE

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2015

ALEX BADO

MÓDULO PARA AVALIAÇÃO AUTOMÁTICA DE PROGRAMAS PARA MOODLE

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – COADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Juliano Rodrigo Lamb.

MEDIANEIRA

2015



TERMO DE APROVAÇÃO

Módulo para avaliação automática de programas para MOODLE

Por

Alex Bado

Este Trabalho de Diplomação (TD) foi apresentado às 7:40 h do dia 12 de junho de 2015 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. M.Sc. Juliano Rodrigo Lamb
UTFPR – Câmpus Medianeira
(Orientador)

Prof. Dr. Pedro Luiz de Paula Filho
UTFPR – Câmpus Medianeira
(Convidado)

Prof. M.Sc. Cesar Alfredo Cardoso
UTFPR – Câmpus Medianeira
(Convidado)

Prof. M.Sc. Juliano Rodrigo Lamb
UTFPR – Câmpus Medianeira
(Responsável pelas atividades de TCC)

RESUMO

BADO, Alex. Módulo para avaliação automática de programas para MOODLE. 2015. 57 f. Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas), Universidade Tecnológica Federal do Paraná. Medianeira. 2015.

Cursos da área da tecnologia da informação possuem várias disciplinas relacionadas a programação, sendo que estas possuem uma carga considerável de exercícios relacionados ao desenvolvimento, o que exige muito tempo e esforço do professor na correção individual. Sistemas corretores automáticos de programas são uma tentativa de auxiliar o professor nesse processo, automatizando a correção dos programas. Os corretores automáticos de programas têm como principal objetivo disponibilizar um feedback da atividade do aluno logo após ele fazer o envio do arquivo com código fonte de seu programa, sem a necessidade do professor ler o seu código para encontrar possíveis erros. Considerando este cenário, este trabalho tem como objetivo o desenvolvimento de um módulo corretor de programas, para o ambiente virtual MOODLE, por meio da linguagem PHP. Para que seja possível a correção, o programa deve permitir a entrada de valores por meio da linha de comando e, estes valores quando submetidos a execução do programa devem gerar um retorno, que será identificado pelo módulo. O professor alimentará valores *default* para entrada e saída, os quais servirão de base para a correção. O plugin tem ainda algumas customizações para auxiliar o professor. Os resultados se mostraram satisfatórios, de modo que o processo é agilizado, observando que este processo deve ser ampliado com suporte para outras linguagens.

Palavras chaves: Corretores Automáticos de Programas, Módulo para MOODLE.

ABSTRACT

BADO, Alex. *Corrector module of exercises for MOODLE programing*. 2015. *Final project for the course of Technology in Analyses and Developing Systems from Universidade Tecnológica Federal do Paraná Medianeira* 2015.

Courses in the Information Technology Area have a large number of disciplines related to programming, also, they have considerable amount of exercises related to development, which requires a lot of time and effort from the teacher correcting each work individually. Automatic corrector systems programs are an attempt to help the teacher in this process, creating a pattern for corrections. Automatic corrector programs have as main goal to provide feedback of the student's activity after they upload the file with source code of their program without the need for the teacher to read their code to find possible errors. Considering this scenario, this work has as objective the development of a corrector program module for the virtual MOODLE environment, throughout the PHP language. For the correction to be possible the program must allow the entrance of values throughout the command line and, these values when submitted to the program execution must create a return that will be identified throughout the module. The teacher will feed default values for the income and outcome, which will be used as base for the correction. The plugin has a few more customizations to help the teacher. The results were satisfactory, because the process was streamlined, noting that this process should be extended with support for other languages.

Key words: Corrector program auto. MOODLE Module.

“A persistência é o caminho do êxito.” (Charles Chaplin)

LISTA DE SIGLAS

ABED	Associação Brasileira de Educação a Distância
API	<i>Application Programming Interface</i>
AVA	Ambiente Virtual de Aprendizagem
CGI	<i>Common Gateway Interface</i>
CSUS	<i>California State University Sacramento</i>
EAD	Educação a Distância
GPL	<i>General Public License</i>
HTML	<i>Hyper Text Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electric and Electronic Engineers</i>
MOODLE	<i>Modular Object-Oriented Dynamic Learning Environment</i>
ODBC	<i>Open Database Connectivity</i>
PHP	<i>Personal Home Page</i>
SBC	Sociedade Brasileira de Computação
SCORM	<i>Sharable Content Object Reference Model</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
URL	<i>Universal Resource Locator</i>

LISTA DE FIGURAS

Figura 1- Ciclo de vida do desenvolvimento de um <i>software</i>	15
Figura 2 - Testes de <i>software</i>	17
Figura 3 - Tela inicial do ambiente TELEDUC	20
Figura 4 - Tela inicial AVA Aulanet	22
Figura 5 - Tela inicial do MOODLE	24
Figura 6 - Fluxograma de processos de criação de cursos no MOODLE	24
Figura 7 - Tela inicial do URI Online Judge	29
Figura 8- Fluxograma do processo de correção de atividades no BOCA	31
Figura 9 - Diagrama de implementação do módulo	33
Figura 10 - Fluxograma de processos do funcionamento do módulo.	34
Figura 11 - Editor XMLDB	35
Figura 12 - Arquivos do banco de dados do módulo.....	36
Figura 13 - Compilação e execução de um programa em JAVA	42
Figura 14 - Compilação de um programa em JAVA com erro.	43
Figura 15 - Adicionando uma atividade de correção de programas	45
Figura 16 - Formulário de criação do corretor de programas.....	46
Figura 17 - Envio de uma atividade para correção	47
Figura 18 - Tela exibida ao enviar um programa com erro de código	48
Figura 19 - Tela exibida ao aluno após o envio da atividade	49
Figura 20 - Tela da atividade exibida ao aluno após a data final	49
Figura 21 - Tela de avaliação de atividades	50
Figura 22 - Campo “Permitir reenvio”	51
Figura 23 – Opção para deletar um arquivo já enviado.....	51

LISTA DE QUADROS

Quadro 1 - Agentes de um sistema corretor de código.....	26
Quadro 2 - Lista de Requisitos do módulo	32
Quadro 3 - Trecho de código de tradução de idiomas do módulo.....	37
Quadro 4 – Código fonte do programa de soma de dois números	38
Quadro 5 - Exemplo de código em PHP.....	39
Quadro 6 - Código responsável pela compilação e avaliação das atividades.....	52

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVO GERAL.....	12
1.2 OBJETIVOS ESPECÍFICOS	12
1.3 JUSTIFICATIVA	12
2 REFERENCIAL TEÓRICO	14
2.1 ENGENHARIA DE <i>SOFTWARE</i>	14
2.1.1 Qualidade de <i>Software</i>	15
2.2 TESTE DE <i>SOFTWARE</i>	16
2.3 AMBIENTES VIRTUAIS DE APRENDIZAGEM.....	19
2.3.1 Teleduc	20
2.3.2 Aulanet	21
2.3.3 MOODLE	22
2.4 VANTAGENS E DESVANTAGENS	25
2.5 CORRETORES AUTOMÁTICOS DE CÓDIGO	26
2.5.1 Pc2	26
2.5.2 Mooshak	27
2.5.3 URI <i>Online Judge</i>	28
2.5.4 BOCA	29
3 MATERIAIS E MÉTODOS	32
3.1 DESENVOLVIMENTO MODULAR PARA MOODLE.....	35
3.2 CÓDIGO USADO NOS TESTES.....	37
3.3 TECNOLOGIAS UTILIZADAS	38
3.3.1 PHP.....	38
3.3.2 WampServer	39
3.3.3 Banco de dados MYSQL.....	40
3.3.4 Netbeans	40
3.3.5 JAVA.....	41
4 RESULTADOS E DISCUSSÕES	44
4.1 VISÃO GERAL DO MÓDULO	44
4.2 CRIANDO E CONFIGURANDO UMA ATIVIDADE DE AVALIAÇÃO	44

4.3 ENVIO DA ATIVIDADE PELO ALUNO	47
4.4 AVALIAÇÃO DAS ATIVIDADES	47
4.4.1 Avaliação de atividade com erro de compilação	47
4.4.2 Avaliação de atividade compilada	48
4.4.3 Múltiplos envios	50
4.5 COMPILAÇÃO DAS ATIVIDADES	52
5 CONSIDERAÇÕES FINAIS.....	53
5.1 CONCLUSÃO.....	53
5.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO	53
REFERÊNCIAS BIBLIOGRÁFICAS	54

1 INTRODUÇÃO

Os ambientes virtuais de aprendizagem (AVA) são ambientes disponibilizados na internet, que agregam muitas ferramentas para gestão de cursos presenciais e não presenciais. Estes ambientes são utilizados por diversas instituições de ensino, visando não só aproximar o aluno do professor ou tutor, como também dar um controle melhor das atividades desenvolvidas pelos alunos no decorrer do curso.

Dentre os vários ambientes virtuais de aprendizagem disponíveis no mercado, o MOODLE é considerado por muitos como um dos melhores, sendo utilizado em países de todo planeta em diversos cursos, tanto na Educação a Distância (EAD) ou como apoio em cursos presenciais (SILVA, 2010). O MOODLE permite que a criação de cursos, onde é possível propor questionários online, atividades de envio de arquivos, fóruns, disponibilização de materiais entre diversas outras funcionalidades.

Em disciplinas de programação, professores levam um tempo considerável na correção de exercícios de desenvolvimento, devido à grande complexidade em analisar os erros encontrados no código. O uso de um AVA nesse caso, ajuda a ter um melhor controle das atividades dos alunos, mas ainda não traz a agilidade necessária, já que o professor tem que corrigir todos os trabalhos individualmente.

O uso de um sistema corretor automático de códigos, pouparia o professor da correção manual dos trabalhos. O BOCA é um dos principais corretores de códigos automático, utilizado como uma ferramenta de apoio à competições de programação. Foi desenvolvido com linguagem de programação PHP, para ser usado na Maratona de Programação da Sociedade Brasileira de Computação (SBC). O sistema também é usado no apoio a disciplinas em que façam uso de submissão e correção de trabalhos durante as aulas. O BOCA disponibiliza os problemas, placar e troca de mensagens além de suportar usuários remotos, permite com que os usuários enviem seus códigos fonte para a correção, podendo esta ser automática ou personalizada.

A utilização do BOCA não resolveria por completo o problema dos professores na correção dos exercícios de programação enviados por seus alunos. Uma solução para esse caso, seria o desenvolvimento de um módulo que realize a correção dos programas/atividades automaticamente, assim como o BOCA. O módulo integrado ao MOODLE permitiria ao professor criar uma atividade, propondo ao aluno o desenvolvimento de um programa e o aluno teria sua atividade avaliada automaticamente em tempo real pelo sistema.

O objetivo deste trabalho está no desenvolvimento de um módulo para correção de exercícios de programação integrado a um AVA, facilitando o controle das atividades pelos professores, já que estes não precisariam mais fazer a correção dos códigos fontes dos alunos de forma manual.

1.1 OBJETIVO GERAL

Desenvolver um módulo de correção automática para o MOODLE, que seja capaz de efetuar a correção dos exercícios de programação, enviados pelos alunos em um curso.

1.2 OBJETIVOS ESPECÍFICOS

São objetivos específicos deste trabalho:

- a. Estudar o ambiente virtual de aprendizagem MOODLE;
- b. Estudar sistemas compilação automática de programas;
- c. Realizar um estudo sobre as ferramentas e linguagens utilizadas para o desenvolvimento do módulo;
- d. Identificar as funcionalidades e as particularidades da aplicação a ser desenvolvida;
- e. Desenvolver a aplicação utilizando as tecnologias e linguagens estudadas;
- f. Apresentar os resultados obtidos.

1.3 JUSTIFICATIVA

Disciplinas de programação possuem uma quantidade elevada de exercícios práticos de desenvolvimento nas mais diversas linguagens de programação. Devido a esta grande quantidade de exercícios, os professores demandam muito tempo para realizar a correção das atividades, já que estas geralmente são complexas, exigindo muito tempo na correção. Com isso acompanhamento dos alunos nestas disciplinas, pode ficar prejudicado, impossibilitando a

correção de todas as atividades dos alunos, além de um grande tempo de espera por parte do aluno para receber a avaliação de seu exercício.

Para auxiliar o professor na correção das atividades de seus alunos, nesse trabalho é proposto o desenvolvimento de um módulo corretor automático de programas para o AVA MOODLE, ele automatizaria toda a correção das atividades. Por meio do módulo, o professor poderia criar uma atividade, propondo ao aluno o desenvolvimento de um programa, que assim que enviado, seria corrigido automaticamente e instantaneamente, para isto, o professor deverá informar na criação da atividade, quais os parâmetros de entrada que serão passados ao programa enviado e o resultado de saída esperado pelo programa após a correção.

2 REFERENCIAL TEÓRICO

Este capítulo aborda tópicos relacionados aos estudos realizados a fim de apresentar os objetivos do trabalho.

2.1 ENGENHARIA DE *SOFTWARE*

O desenvolvimento de *softwares* tem crescido muito na sociedade, devido ao grande uso de computadores nas mais diversas áreas do conhecimento humano. Com isto, surgiu a engenharia de *software*, com a intenção de estabelecer regras e técnicas para desenvolvimento de aplicações computacionais.

Um *software* não é apenas um programa de computador como muitos pensam, mas sim um conjunto de dados como a documentação e configurações necessárias para seu funcionamento. A engenharia de *software* trata os aspectos relacionados a processos, métodos, técnicas, ferramentas e ambientes para desenvolvimento de *softwares*. Também são utilizadas metodologias de gerenciamento de projetos, além de outras tecnologias, como banco de dados, ambientes de programação, linguagens de programação e de qualidade de *software*. Esta engenharia faz o uso de modelos para especificar, projetar, desenvolver e manter o *software*, facilitando assim todo processo de criação da aplicação até mantimento dele após a entrega ao usuário.

Segundo Bauer (1969, p. 231), “a Engenharia de *Software* é a criação e a utilização de sólidos princípios de engenharia a fim de obter *software* de maneira econômica, que seja confiável e que trabalhe em máquinas reais[...]”.

A engenharia de *software* inicialmente decompõe o problema a ser tratado em partes menores, para que uma solução seja desenvolvida para cada parte. Com cada parte solucionada, é necessário fazer a integração de cada uma delas, para isto deve-se estabelecer uma arquitetura para o desenvolvimento.

A Figura 1 mostra o ciclo de vida do desenvolvimento de um *software*.

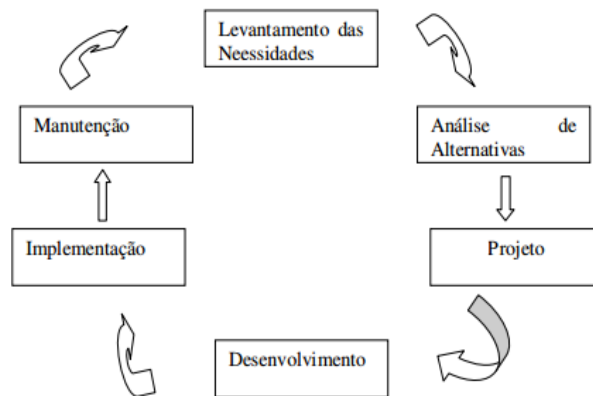


Figura 1- Ciclo de vida do desenvolvimento de um *software*

Fonte: Gordon & Gordon (2006)

2.1.1 Qualidade de *Software*

A engenharia de *software* tem como principal objetivo desenvolver uma aplicação de qualidade, tanto ao usuário quanto ao desenvolvedor. Para o usuário, um *software* de qualidade, tem que atender as suas necessidades, ser fácil de usar e eficiente, já para o desenvolvedor, uma boa aplicação vai além disto, ela precisa ser organizada, fácil de entender para outros desenvolvedores e de fácil manutenção.

Como em outros setores, a qualidade do processo de *software* é fundamental para um produto final de boa qualidade. Abordagens de qualidade de processo, assim como a série de padrões ISO 9000, sugerem que melhorando a qualidade do processo de *software*, é possível melhorar a qualidade do produto final. A premissa por detrás dessa afirmativa é a de que com processos bem estabelecidos, que incorporam mecanismos sistemáticos para acompanhar o desenvolvimento e avaliar a qualidade, no geral, conduzem a produtos finais de qualidade. Por exemplo, quando uma empresa fabricante de algum produto certificada ISO 9001 (uma das normas da série ISO 9000), isto não garante que todos os produtos fabricados por ela são de qualidade, mas sim que eles passam por um bom processo produtivo, o que pode levar a produtos de boa qualidade (FALBO, 2005).

Um processo de *software*, em uma abordagem de Engenharia de *Software*, envolve diversas atividades que podem ser classificadas quanto ao seu propósito em:

- a) Atividades de Desenvolvimento (Técnicas ou de Construção): são as atividades diretamente relacionadas ao processo de desenvolvimento do *software*, ou seja, que contribuem diretamente para o desenvolvimento do produto de *software* a ser entregue ao cliente. São exemplos de atividades de desenvolvimento: especificação e análise de requisitos, projeto e implementação.
- b) Atividades de Gerência: são aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto, tais como realização de estimativas, elaboração de cronogramas, análise dos riscos do projeto etc.
- c) Atividades de Garantia da Qualidade: são aquelas relacionadas com a garantia da qualidade do produto em desenvolvimento e do processo de *software* utilizado, tais como revisões e inspeções de produtos (intermediários ou finais) do desenvolvimento.

As atividades de desenvolvimento formam a espinha dorsal do desenvolvimento e, de maneira geral, são realizadas segundo uma ordem estabelecida no planejamento. As atividades de gerência e de controle da qualidade são, muitas vezes, ditas atividades de apoio, pois não estão ligadas diretamente à construção do produto final: o *software* a ser entregue para o cliente, incluindo toda a documentação necessária. Essas atividades, normalmente, são realizadas ao longo de todo o ciclo de vida, sempre que necessário ou em pontos pré-estabelecidos durante o planejamento, ditos marcos ou pontos de controle.

2.2 TESTE DE SOFTWARE

O teste de *software* tem como objetivo a execução de um produto antes da entrega ao cliente e identificar possíveis defeitos, erros e falhas a serem corrigidas. A definição de teste de *software* se resume em executar a aplicação de maneira controlada e verificar se seu comportamento corre de acordo com o que foi especificado. As atividades de teste são de natureza destrutiva e não construtiva, pois visa aumentar a confiabilidade de um sistema ao exibindo todos problemas, porém antes da entrega final ao cliente.

Existem algumas definições sobre o teste *software*, o IEEE - *Institute of Electrical and Electronics Engineers* (IEEE 610, 1990) define o seguinte padrão para engenharia de *software* sobre defeitos, erros e falhas:

- a) Defeito: é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Por exemplo, uma instrução ou comando incorreto.
- b) Erro: é uma manifestação concreta de um defeito num artefato de *software*. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro.
- c) Falha: é o comportamento operacional do *software* diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.

A Figura 2 mostra a diferença entre cada um destes conceitos. Defeitos (universo físico) são causados por um mau uso de uma tecnologia por parte do desenvolvedor, podendo assim gerar um erro no produto, ou seja, a construção do *software* não seguiu corretamente a especificação (universo da informação). Os erros podem gerar falhas, que são comportamentos não esperados que afetam diretamente o usuário final do *software* (universo do usuário).



Figura 2 - Testes de *software*

Fonte: NETO (2008)

A atividade de teste é composta por alguns elementos que visam formalizar a atividade. Estes elementos são:

- a) Caso de Teste: descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado (CRAIG e JASKIEL, 2002).
- b) Procedimento de Teste: é uma descrição dos passos necessários para executar um caso (ou um grupo de casos) de teste (CRAIG e JASKIEL, 2002).

- c) Critério de Teste: serve para selecionar e avaliar casos de teste de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto (ROCHA et al., 2001). Os critérios de teste podem ser utilizados como:
- d) Critério de Cobertura dos Testes: permitem a identificação de partes do programa que devem ser executadas para garantir a qualidade do *software* e indicar quando o mesmo foi suficientemente testado (RAPPS e WEYUKER, 1982). Ou seja, determinar o percentual de elementos necessários por um critério de teste que foram executados pelo conjunto de casos de teste.
- e) Critério de Adequação de Casos de Teste: Quando, a partir de um conjunto de casos de teste T qualquer, ele é utilizado para verificar se T satisfaz os requisitos de teste estabelecidos pelo critério. Ou seja, este critério avalia se os casos de teste definidos são suficientes ou não para avaliação de um produto ou uma função (ROCHA et al., 2001).
- f) Critério de Geração de Casos de Teste: quando o critério é utilizado para gerar um conjunto de casos de teste T adequado para um produto ou função, ou seja, este critério define as regras e diretrizes para geração dos casos de teste de um produto que esteja de acordo com o critério de adequação definido anteriormente (ROCHA et al., 2001).

Existem diversas maneiras de se testar um *software*. As técnicas de teste são definidas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste. Elas contemplam diferentes perspectivas do *software* e impõe-se a necessidade de se estabelecer uma estratégia de teste que contemple as vantagens e os aspectos complementares dessas técnicas. As técnicas existentes são: técnica funcional e estrutural.

- a) **Técnica Funcional (caixa preta):** O teste funcional funciona como uma caixa preta, leva em conta apenas comportamentos externos, ignorando comportamentos internos. No teste são fornecidos dados de entrada e após a execução, o resultado de saída é comparado com o resultado esperado previamente conhecido. O teste terá um resultado positivo se a saída da execução do teste for igual ao resultado esperado. O componente de *software* a ser testado pode ser um método, uma função interna, um programa, um componente, um conjunto de programas e/ou componentes ou mesmo uma funcionalidade. A técnica de teste funcional é aplicável a todos os níveis de teste (PRESSMAN, 2005).

- b) **Técnica Estrutural:** Trabalha diretamente com o código fonte do *software*, leva em conta apenas comportamentos internos. Segundo Pressman essa técnica trabalha diretamente sobre o código fonte do componente de *software* para avaliar aspectos tais como: teste de condição, teste de fluxo de dados, teste de ciclos e teste de caminhos lógicos.

2.3 AMBIENTES VIRTUAIS DE APRENDIZAGEM

Os Ambientes Virtuais de Aprendizagem são ferramentas que auxiliam tutores e professores em cursos à distância e presencias. Eles dispõem de diversos recursos para a criação, tutoria e gestão de atividades que normalmente se apresentam em forma de curso. A intenção de um ambiente virtual se aplica em estreitar o relacionamento entre alunos e professores, podendo ser usado para envio de atividades, disponibilização de materiais, fóruns interativos e como canal de comunicação entre aluno e professor.

Segundo CIÊNCIA & COGNIÇÃO (2008), um AVA eficiente é composto por itens fundamentais ao processo de aprendizado, são eles Conteúdo, Formato, Infraestrutura e Pedagogia. O primeiro item define a abordagem do tema que será proposto ao estudante, bem como a forma de representação deste conteúdo no ambiente virtual. Já o formato compreende os parâmetros curriculares determinados pela instituição de ensino e os recursos humanos que serão utilizados (professores, monitores, técnicos). A infraestrutura relaciona-se com os equipamentos computacionais que serão utilizados para oferecer o ambiente virtual de maneira satisfatória, o que inclui *softwares* e *hardwares*. Por fim a pedagogia aplicada como forma de abordagem didática a partir dos tópicos do conteúdo programático, visando determinar a metodologia de ensino mais adequada.

Existem vários Ambientes Virtuais de Aprendizagem comerciais e gratuitos disponíveis no mercado: Eleven, Edmodo, Aulanet, Claroline, eFront, Atutor, OLAT, Docebo, Dokeos, Ilias, Openlms, MOODLE, Sakai, E-proinfo e Teleduc. Entre estes se destacam, o Teleduc que possui grande aceitação no Brasil, o Aulanet que foi a primeira plataforma de educação a distância desenvolvida no Brasil e o MOODLE é o de maior utilização em todo o mundo (SILVA, 2010), desde sua criação em 2002, já foi utilizado em diversas instituições em mais de 200 países para atender as mais diversas necessidades.

2.3.1 Teleduc

O Teleduc é um ambiente virtual de educação à distância para criação, participação e administração de cursos na Web. Foi concebido para a formação de professores de informática educativa, baseado na metodologia de formação desenvolvida por pesquisadores do Núcleo de Informática Aplicada à Educação à Distância da Unicamp (Universidade Estadual de Campinas, Brasil). O Teleduc é um *software* livre; pode-se redistribuí-lo e/ou modificá-lo sob os termos da GNU *General Public License* (TELEDUC, 2007).

As funcionalidades oferecidas pelo Teleduc estão agrupadas em três grupos: ferramentas de coordenação, de administração e de comunicação. As ferramentas de coordenação têm como objetivo principal organizar as ações no curso, informando os alunos as intenções da proposta e contribuindo para uma melhor organização e compreensão da dinâmica de trabalho. As ferramentas de comunicação só podem ser acessadas dentro do ambiente, a exemplos do correio, bate-papo, fóruns de discussão e mural. Entretanto com o desenvolvimento de novas versões, já se pode enviar *e-mail* externo aos destinatários, porém a resposta só pode ser dada a partir do acesso ao ambiente. As ferramentas de administração têm como objetivo dar apoio ao gerenciamento do curso por meio do suporte e da administração e não são visíveis aos alunos (ROCHA, 2003).

Na Figura 3 é exibido o layout do ambiente virtual de aprendizagem Teleduc na visão do aluno.

The screenshot displays the TELEDUC web interface. At the top left is the TELEDUC logo. The main header area shows the course title "INFORMÁTICA NA EDUCAÇÃO MATEMÁTICA". A left sidebar contains a navigation menu with items like "Estrutura do Ambiente", "Dinâmica do Curso", "Agenda", "Material de Apoio", "Perguntas Frequentes", "Parada Obrigatória", "Fóruns de Discussão", "Bate-Papo", "Correio", "Grupos", "Perfil", "Diário de Bordo", "Portfólio", "Acessos", "Intermap", and "Busca". The main content area is titled "Agenda - Agenda Atual" and shows a section for "Agendas Anteriores" with a "Título" and "Texto" field. The text in the "Texto" field reads: "Caros alunos: Na nossa aula do dia 19/04/2011, tivemos nossos dois Chats. A ferramenta se 'comportou' direito e com um percalço ou outro conseguimos cumprir a meta. Na próxima aula do dia 26/04, próximo, iremos fazer um fechamento das ideias para a avaliação do 03/05/2011. Boa Páscoa a todos!". At the bottom, there are logos for TELEDUC, nied, Instituto de Computação, and UNICAMP, along with the copyright notice: "2010 - Teleduc - Todos os direitos reservados. All rights reserved - NIED - UNICAMP".

Figura 3 - Tela inicial do ambiente TELEDUC

Fonte: ROCHA (2010)

2.3.2 Aulanet

O ambiente Aulanet foi desenvolvido no Laboratório de Engenharia de *Software* do Departamento de Informática da Universidade PUC-RJ, para administração, criação e manutenção e participação em cursos a distância.

O funcionamento do ambiente é fundamentado no princípio da cooperação entre alunos e professores, tendo como suporte as interfaces da internet. Tem como objetivo, a interação no curso, a utilização por pessoas não especialistas na área, e a reutilização de conteúdos já existentes na mídia digital (PPGED-UFBA, 2015).

Vavassori e Raabe (2003) destacam o docente, o aprendiz e o administrador como os principais atores do ambiente, ressaltando a função dos mesmos. Para ele a construção da proposta educativa é de responsabilidade do docente, que conta com uma série de ferramentas como o fórum, chat, portfólio, etc. O aprendiz é de suma importância na dinâmica do ambiente, já que sua participação ativa é fundamental no desenvolvimento de processos cooperativos com os demais participantes. As questões mais técnicas, relativas a matrícula e ao registro no AVA, são de responsabilidade do administrador, que opera continuamente no ambiente.

Completam ainda os autores, que o ambiente dispõe de algumas ferramentas para a disponibilização dos cursos, que são: os mecanismos de comunicação, de coordenação e de cooperação. Os mecanismos de comunicação possibilitam o envio de mensagens aos docentes e à coordenação do curso, além da criação dos grupos de discussão, grupos de interesse, debates por meio de chats, e o contato com os participantes conectados no ambiente. Já os mecanismos de coordenação como avisos, planos de aula, tarefas, avaliação e relatórios de participação permitem a criação de atividades e um melhor acompanhamento dos estudantes.

Tendo como foco o trabalho coletivo, os mecanismos de cooperação disponibilizados no ambiente como a bibliografia, a “webliografia”, a documentação, o *download*, a coautoria do aprendiz permite que os sujeitos possam cooperar uns com os outros, socializando seus saberes e construindo novas habilidades (VAVASSORI E RAABE, 2003).

A Figura 4 mostra o layout tela inicial do AVA Aulanet.

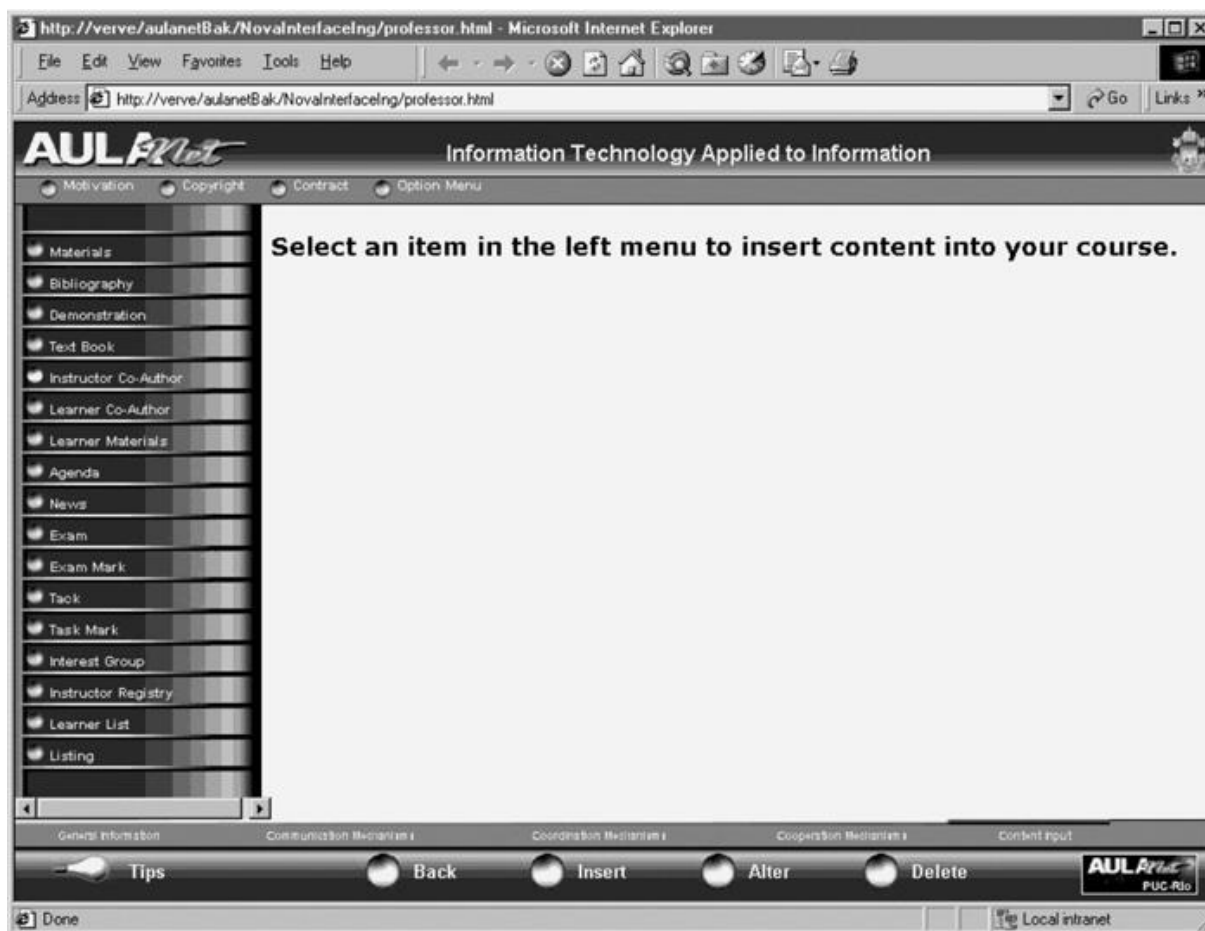


Figura 4 - Tela inicial AVA Aulanet

Fonte: MOREIRA (2012)

2.3.3 MOODLE

O MOODLE é um sistema *web* para gerenciamento de cursos de aprendizagem, de código aberto e gratuito. O projeto desenvolvido pelo australiano Martin Dougiamas iniciou na década de 90 com sua primeira versão lançada em 2002, tendo como seu principal objetivo permitir que processos de ensino-aprendizagem ocorram não apenas a partir da interatividade, mas também, pela interação, ou seja, priorizando a construção/reconstrução do conhecimento, a autoria, a produção de conhecimento em colaboração com os pares e a aprendizagem do aluno (MOREIRA, 2012).

O MOODLE é uma ferramenta de auxílio para a educação a distância (EAD), em que podem ser criadas salas de estudos online, disponibilizar materiais didáticos, aplicar exercícios avaliativos, fóruns, etc. Considerado o ambiente virtual de maior aceitação no mundo ele

encontra-se em 224 países, possuiu mais de 60 pacotes de idioma e está em cerca de 53300 sites, destes mais de 3600 só no Brasil (MOODLE, 2015). Sua principal utilização é para suporte a educação à distância e como apoio a cursos presenciais. Roda em todas as plataformas que conseguem executar a linguagem PHP e como base de dados podem ser utilizados o MySQL, PostgreSQL, Oracle, Access, Interbase ou ODBC.

A autenticação no MOODLE é feita por meio de um *login* de acesso para cada um de seus usuários, sendo que estes podem ter diferentes perfis de acesso, como por exemplo, perfil de professor, estudante, administrador, etc. Os níveis de acesso podem ser configurados, adicionando ou removendo permissões, também pode-se criar novos perfis adicionando as permissões necessárias.

Por se tratar de uma ferramenta de código aberto, o MOODLE pode ser adaptado para o fim que ele será utilizado, alterações no seu código nativo podem ser feitas sem muita complexidade, novos campos podem ser inseridos nos cadastros, seu layout pode ser facilmente alterado, além de novas funcionalidades que podem ser desenvolvidas incorporadas ao ambiente.

As principais vantagens do MOODLE são (MOODLE, 2015):

- a) Criação de layout;
- b) Criação e adaptação de novas funcionalidades;
- c) Compartilhamento de conteúdo;
- d) Instalação de novos módulos;
- e) Adequado para aulas 100% online;
- f) Simples, leve, eficiente, compatível;
- g) Um site MOODLE pode suportar milhares de cursos;
- h) Segurança total.
- i) Mais de 60 pacotes de idioma;

A interface gráfica do MOODLE é amigável, de fácil acesso aos recursos e configurações. A Figura 5 mostra a tela inicial com o layout padrão do MOODLE na versão 2.8.

The screenshot shows the Moodle interface. At the top, it says 'Moodle' and 'Português - Brasil (pt_br)'. The user is identified as 'Alex Bado'. The main heading is 'Ambiente Virtual de Aprendizagem'. On the left, there is a 'NAVEGAÇÃO' menu with options like 'ADMINISTRAÇÃO', 'Configurações da página inicial', 'Ativar edição', 'Editar configurações', 'Usuários', 'Filtros', 'Relatórios', 'Backup', 'Restaurar', 'Banco de questões', 'Minhas configurações de perfil', and 'Administração do site'. The main content area is titled 'Categorias de Cursos' and shows two categories: 'Redes de Computadores' with sub-items 'Turma 01 (1)' and 'Turma 02', and 'Programação Web' with sub-item 'Turma 01 (1)'. On the right, there is a 'CALENDÁRIO' for 'março 2015' with a grid of days.

Figura 5 - Tela inicial do MOODLE

Fonte: Autoria própria

Para criar cursos online no MOODLE, não são necessários conhecimentos em linguagens de programação, todo o processo de criação e configuração do curso é feita por meio da interface gráfica do ambiente com apenas alguns cliques. A Figura 6 mostra o fluxograma de processos para se criar um novo curso no MOODLE.

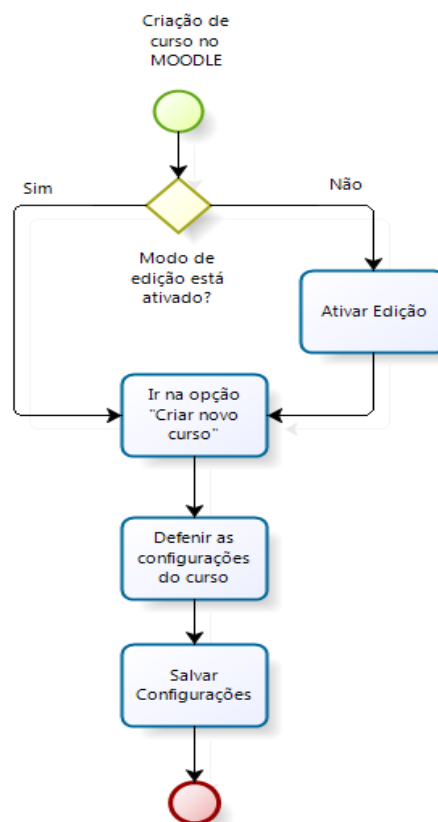


Figura 6 - Fluxograma de processos de criação de cursos no MOODLE

Fonte: Autoria própria

O MOODLE permite criar categorias e subcategorias para poder melhor organizar o ambiente. Alguns recursos que podem ser utilizados em um curso online no MOODLE são: fóruns interativos, wiki, pacotes SCORM, questionários online, atividades de envio de arquivos, videoconferências, chats entre outros recursos. Ele também permite criação de novos recursos que podem ser facilmente incorporados a versão nativa.

Por se tratar de uma ferramenta de código aberto, o MOODLE possui uma comunidade de desenvolvedores muito ativa, muitos disponibilizam seus códigos na comunidade online MOODLE.org, facilitando assim o desenvolvimento de novos recursos para o AVA.

2.4 VANTAGENS E DESVANTAGENS

Os ambientes virtuais de aprendizagem possuem diversas vantagens que auxiliam tanto na educação à distância como na educação presencial. Em cursos presenciais um AVA pode ser utilizado para disponibilizar os materiais utilizados em salas de aula, propor o desenvolvimento de atividades em que o aluno pode fazer o envio pelo próprio sistema, aplicar provas ou questionários avaliativos online, criar fóruns de discussão, comunicação entre os participantes do curso, entre outros recursos. Os AVAs dão suporte a cursos totalmente à distância com recursos como: *chats*, videoconferências, utilização de webcams, conteúdos interativos na forma de pacotes SCORM (*Sharable Content Object Reference Model*), etc.

Apesar da grande quantidade de recursos que os AVAs possuem hoje, eles ainda possuem algumas desvantagens, a principal é o controle da evasão de alunos. Um curso mal estruturado, pode se tornar cansativo e de difícil entendimento para o aluno, o que pode levar o mesmo a desistir do curso.

Segundo a Associação Brasileira de Educação a Distância (ABED), algumas desvantagens em se utilizar AVAs, estão na falta de troca de experiências entre alunos e professores e de convivência humana e contato pessoal (SCHUELTER, 2005).

2.5 CORRETORES AUTOMÁTICOS DE CÓDIGO

Os corretores automáticos de código possuem como objetivo validar a consistência de um algoritmo por meio de vários casos de teste. Este processo é semelhante ao utilizado pela engenharia de *software* no teste da caixa preta ou funcional. Esta técnica deve procurar explorar a maior variedade possível do espaço de solução, a fim de poder testar as mais diversas possibilidades de soluções e casos existentes.

O funcionamento dos corretores automáticos consiste em analisar, após uma coleção de entrada, como o algoritmo proposto se comporta e quais resultados correspondem para cada caso de teste. O objetivo deste tipo de corretor não é avaliar o desempenho ou a forma de como o algoritmo desenvolvido pelo aluno é constituído, pois não é o foco da disciplina, o objetivo é validar se o algoritmo desenvolvido pelo aluno está correto. No Quadro 1 são destacadas as obrigações de cada agente de um sistema corretor automático de código.

Aluno	Professor	Servidor
Interpretar	Informar Problema	Receber
Resolver	Entrada Padrão	Comparar
Submeter	Saída esperada	Avaliar

Quadro 1 - Agentes de um sistema corretor de código.

Fonte: Autoria própria

Os principais corretores automáticos de *softwares* foram desenvolvidos com o objetivo de apoiar competições de programação. Basicamente eles possuem três atores: competidor, juiz e administrador. Os principais corretores automáticos de *software* são: Pc2, Mooshak, BOCA e o URI Online Judge, dentre outros.

2.5.1 Pc2

O Pc2 (*Programming Contest Control*) é um sistema projetado para suportar operações de maratonas de programação em vários ambientes de computação. O Pc2 permite que as

equipes de desenvolvimento apresentem seus problemas por meio do sistema para a correção dos juízes. Os juízes podem recompilar o programa apresentado, executa-lo, verificar o código fonte e os resultados de execução, e assim enviar uma resposta à equipe (PC2, 2012).

O Pc2 também possui uma opção para corrigir automaticamente os programas, assim a correção é feita pelo próprio sistema e não por juízes. O sistema possui um mecanismo que automaticamente exibe as posições das equipes na competição. Ele também permite que os membros das equipes solicitem aos juízes esclarecimentos sobre a correção e que os juízes retornem uma resposta as equipes (PC2, 2012).

O Pc2 possui uma variedade de opções configuráveis que permitem ao administrador da maratona adaptar o sistema de acordo com necessidades da competição, como por exemplo, o número de equipes, problemas e linguagens no concurso, o método de pontuação a ser aplicado, quais problemas serão tratados por quais juízes, se as equipes serão automaticamente notificadas do resultado e a frequência das atualizações automáticas do painel de avaliação. Existem também mecanismos previstos para editar o banco de dados de pontuação interna e para a recuperação de vários tipos de dados simples e complexos (SOUZA, 2009). O sistema é projetado para permitir que equipes de programadores usem qualquer ferramenta da linguagem, que pode ser chamado a partir de uma linha de comando gerando um arquivo executável.

O PC2 foi desenvolvido na Califórnia State University, Sacramento (CSUS), e está disponível no site <http://www.ecs.csus.edu/pc2> (PC2, 2008). Escrito em Java suporta ser executado em qualquer versão Java 1.5 (ou superior). Suporta os sistemas operacionais Windows (98 / ME / 2000 / XP / Vista / 7), Mac OS X (10.4+) e uma variedade de sistemas baseados em Unix, incluindo Solaris, Linux e FreeBSD (SOUZA, 2009).

2.5.2 Mooshak

O Mooshak é um sistema de código aberto, utilizado em maratonas de programação. Ele corrige e avalia automaticamente códigos de programas enviados por seus usuários, para isto é necessário que um problema seja proposto no sistema juntamente com o resultado de saída esperado (MOOSHAK, 2008).

O sistema inicialmente foi desenvolvido para uso em maratonas de programação, mas seu uso em cursos de programação vem aumentando, para dar uma maior agilidade na correção

dos exercícios desenvolvidos pelos alunos. Após o aluno submeter o arquivo com o código fonte de seu programa, o Mooshak retorna um *feedback* instantâneo, apontando erros e pré-avaliando o exercício. (MOOSHAK, 2008)

2.5.3 URI Online Judge

O URI *Online Judge* é um portal disponibilizado ao público em 01 de fevereiro de 2012, serve como uma importante ferramenta no apoio nos estudos de alunos de disciplinas de programação. O portal foi desenvolvido como uma plataforma EAD, interativa e com uma interface amigável. O URI é capaz de corrigir códigos automaticamente em tempo real (TONIN E BEZ, 2008).

A construção do URI foi projetada para atender as necessidades dos professores e principalmente dos alunos, suas principais características são (TONIN E BEZ, 2008).

- a) Ambiente amigável, organizado e de acesso 24 horas por dia;
- b) Correção de programas em tempo real disponibilizando visualmente aos alunos seus erros;
- c) Ambiente disponível nos idiomas inglês, português e espanhol;
- d) Disponibilizar aos professores a opção de criar disciplinas com alunos e disponibilizar exercícios a eles;
- e) Classificar os exercícios por assunto e níveis de dificuldade;
- f) Oferecer um ambiente em que os alunos possam interagir, trocando experiências sobre os exercícios;

O URI Online Judge pode ser dividido em três ambientes diferentes: o site principal meio de acesso dos estudantes, um ambiente para os professores e ambiente de fóruns.

O site está disponível em três idiomas (inglês, espanhol e português), e foi desenvolvido com uma interface muito intuitiva, para facilitar o uso pelos estudantes. No site é possível visualizar os assuntos por categorias, são 8 no total, permitindo o usuário focar no que é o de seu interesse. Também possível visualizar o nível de cada problema por meio de cores, os níveis podem variar de 1 a 10.

O ambiente utilizado pelo professor permite uma melhor utilização da ferramenta em sala de aula, oferecendo recursos para o professor poder ter um melhor controle e um

acompanhamento do progresso dos alunos, podendo assim auxiliar os alunos que estão com dificuldades.

O fórum possui uma divisão por categorias e dentro de cada uma delas existe um tópico exclusivo para cada problema pertencente a aquela categoria. Além disso, o fórum possui moderadores voluntários de diversas localidades do Brasil que o acompanham diariamente respondendo dúvidas e ajudando a deixá-lo organizado. A Figura 7 mostra a tela inicial do URI Online Judge.

The screenshot shows the URI Online Judge dashboard. At the top, there is a navigation bar with links: HOME, PERFIL, CONFIGURAÇÕES, NEWS, FÓRUM, ACADEMIC, CONTESTS, BUSCAR, PROBLEMAS, SUBMITER, SUBMISSÕES, ESTATÍSTICAS, RANKS, SAIR. The main content area is divided into several sections:

- Left Sidebar:**
 - URI logo
 - ONLINE JUDGE PROBLEMS & CONTESTS
 - Bem-vindo, ALEX BADO
 - TOP 20 list: Gabriel Dalalio, Wyllian, Thalyson Nepomuceno, Junior Andrade, Matheus Leão, Dâmi Henrique, Caio Russi, Luciano Ribeiro, Abner Samuel P. Palmeira, Welton Cardoso, André Alves, Fúlvio Abrahão de Paula.
- Central Dashboard:**
 - DASHBOARD
 - Este é o seu dashboard. Aqui você encontrará algumas coisas interessantes.
 - [DIA 32] SEU PROGRESSO ATÉ AGORA
 - 00.00%
 - 14/06/2015 09:45:12 (UTC-3)
- Grid of Interactive Cards:**
 - PROBLEMAS:** Explore nosso repositório, dividido em 8 grandes categorias. EXPLORE
 - FUNCIONALIDADES:** Vote nas próximas funcionalidades a serem desenvolvidas. VOTAR
 - RANK:** Visualize o ranking principal do portal URI Online Judge. VISUALIZAR
 - UNIVERSIDADES:** Selecione a sua Universidade e confira o ranking. EXPLORE
 - CONTESTS:** Pratique para competições de programação! VISUALIZAR
 - NEWS:** Confira as últimas novidades sobre o URI Online Judge. VISUALIZAR

Figura 7 - Tela inicial do URI Online Judge

Fonte: URI Online Judge (2015)

2.5.4 BOCA

O corretor de códigos BOCA, é um sistema *web* utilizado em maratonas de programação, ele permite que as equipes de programadores enviem a resolução das questões da maratona ao sistema, este automaticamente faz a correção e retorna as equipes se a atividade está correta ou errada, permitindo a equipe fazer um novo envio, desde que o tempo da prova

não tenha esgotado. Caso a atividade enviada ao BOCA esteja errada, o sistema não mostra aonde que está erro para equipe, apenas informa que a atividade não está correta (CAMPOS E FERREIRA, 2004).

O BOCA possuiu autenticação, controle de tempo e disponibilização de resultados em tempo real. O sistema é totalmente desenvolvido em linguagem de programação PHP, podendo ser executado em qualquer sistema operacional. Para armazenamento dos dados o sistema é flexível, suportando diversos SGDB (Sistema Gerenciador de Banco de Dados). Para a sua execução ainda é necessário um servidor de páginas em que sistema possa estar incubado.

O sistema BOCA tem como recursos, um chat/fórum que funciona apenas de forma dirigida e controlada, para que assim o aluno possa tirar dúvidas relacionadas a prova sem precisar que o professor precise se deslocar até o aluno. As dúvidas podem ser respondidas de forma individual ou ainda de forma coletiva (CAMPOS E FERREIRA, 2004).

O BOCA pode ser dividido em cinco partes: `time`, `juiz`, `administrador`, `staff` e `placar`. Cada parte, possui uma interface própria que corresponde a um perfil de usuário.

O usuário `Time` pode enviar os programas para correção, alterar informações pessoais, enviar dúvidas aos professores e verificar o placar *online* da maratona. O `time` também pode solicitar por meio do sistema a impressão de seus códigos fontes e solicitar ajuda ao `staff`. Ao fazer o envio da atividade o `time` deve informar a linguagem utilizada para solucionar o problema proposto.

O usuário `Juiz` pode responder perguntas e dúvidas dos times de formar individual ou de forma coletiva, ou criar perguntas globais para serem respondidas a todos. O `juiz` também pode corrigir as atividades enviadas pelos times e definir no sistema qual será a resposta que o `time` irá receber.

O `administrador` é usuário de maior acesso do sistema, é ele que configura as informações sobre a maratona, as linguagens a serem utilizadas e as questões da prova. O `administrador` também é responsável pela inclusão dos times, dos juizes, placares, `staff` e novos perfis de `administrador` na maratona. O `administrador` tem acesso total as configurações do sistema, todo o serviço necessário para colocar a maratona em funcionamento pode ser feito pelo `administrador`.

A principal função do `staff` durante a prova controlar o acesso e a movimentação no local da prova, imprimir os arquivos solicitados pelos times e entregar a impressão a eles, mostrar de forma visual o placar para as equipes e dar assistência aos times com problemas

relacionados a hardware e *software*. Todas interações do staff apesar de serem físicas, todas as solicitações a ele podem ser feitas pelo BOCA.

O usuário *placar* é necessário para poder disponibilizar pela rede o resultado em tempo real da competição para pessoas que não estão diretamente ligadas a ela. A interface deste usuário não permite nenhuma alteração ou configuração no sistema, serve somente como leitura do placar parcial para pessoas que não podem ter acesso a configuração do sistema, como por exemplo os técnicos das equipes.

A Figura 8, mostra o fluxograma do processo de correção de atividades do sistema BOCA.

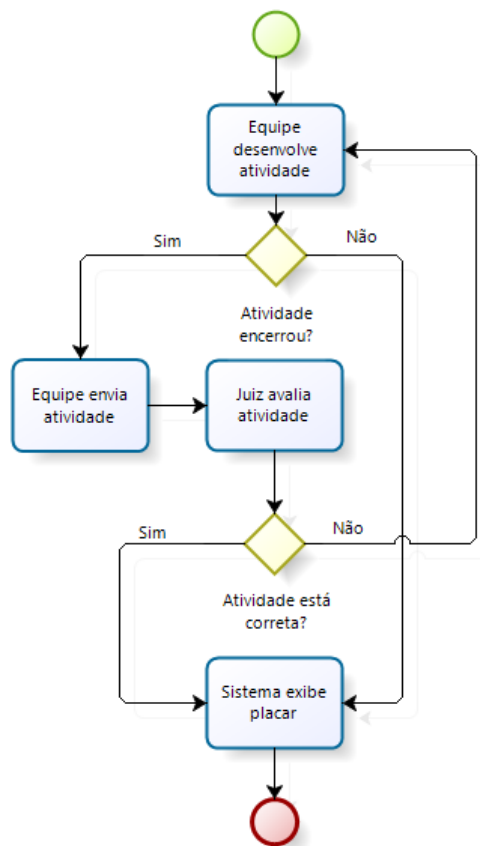


Figura 8- Fluxograma do processo de correção de atividades no BOCA

Fonte: Autoria própria

3 MATERIAIS E MÉTODOS

Este capítulo apresenta todas as ferramentas necessárias para o desenvolvimento do módulo para o Ambiente Virtual de Aprendizagem MOODLE. O módulo irá dar apoio aos professores de disciplinas de programação com a correção automática das atividades enviadas pelos seus alunos, poupando o tempo do professor e dando um retorno mais eficiente ao aluno sobre a sua atividade.

Para iniciar o desenvolvimento do módulo, foi realizado um levantamento de requisitos do sistema, por meio deste levantamento, foi possível identificar as funcionalidades do módulo e as ferramentas necessárias para o desenvolvimento.

O Quadro 2 mostra a lista de requisitos do sistema.

ID	Funcionalidade	Descrição	Ator
R1	Envio de atividade	Permitir o envio de arquivo um arquivo por meio da atividade	Aluno
R2	Avaliação de atividade automática	Avaliar automaticamente atividades enviadas pelos alunos	Sistema
R3	Avaliação de atividade manual	Permitir a correção manual das atividades enviadas pelos alunos	Professor
R4	Escrever Feedback	Permitir que o professor possa escrever um comentário sobre a atividade do aluno	Professor
R5	Download de arquivo	Permitir o professor fazer download do arquivo enviado pelo aluno	Professor
R6	Reenviar atividade	Permitir o Reenvio de um arquivo	Aluno
R7	Ver nota	Permitir que o aluno visualize a nota de sua atividade	Aluno

Quadro 2 - Lista de Requisitos do módulo

Fonte: Autoria própria

Para o desenvolvimento do módulo foi utilizado um servidor local com o sistema operacional Microsoft Windows 8.1 Professional. Para poder executar a aplicação, foi utilizado o ambiente para execução de aplicações *web* WampServer, este possuiu um conjunto de programas necessários para que fosse feita instalação do AVA MOODLE versão 2.8. Para realizar a compilação e execução dos exercícios de programação, foi instalado o compilador JAVA. A Figura 9 mostra o diagrama de implantação do módulo.

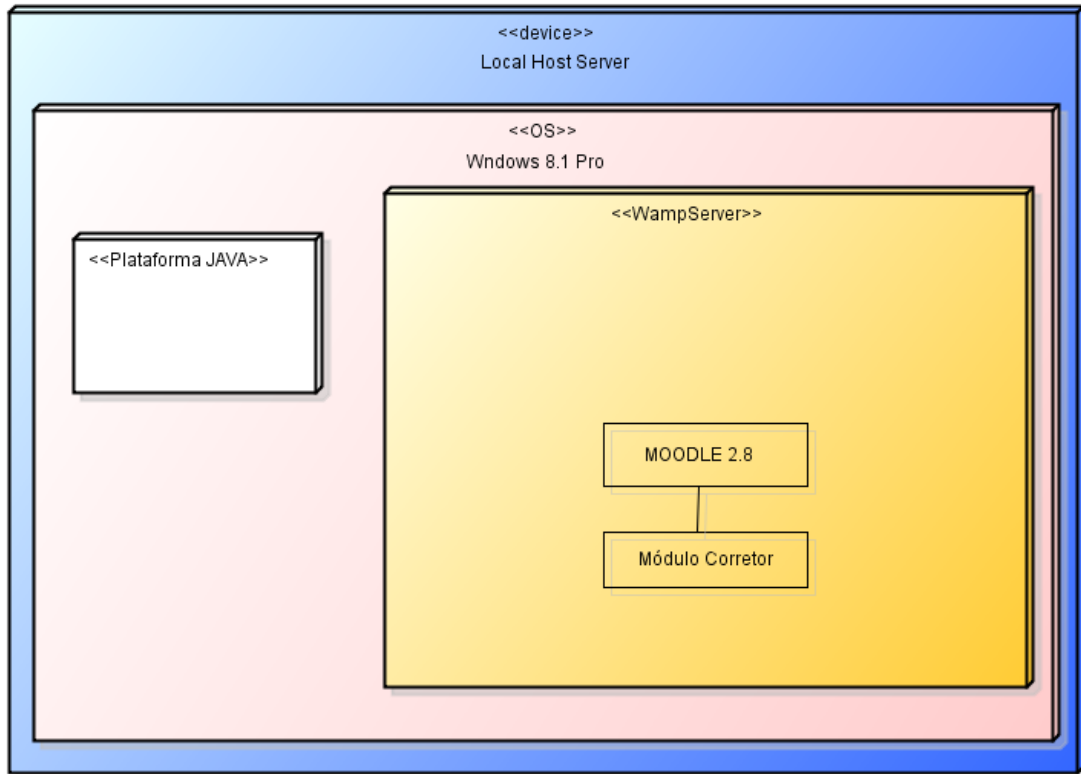


Figura 9 - Diagrama de implementação do módulo

Fonte: Autoria própria

O módulo permite o professor criar uma atividade no MOODLE para que seus alunos enviem seus exercícios de programação. O aluno deverá enviar o arquivo com a extensão .java e seu código-fonte deve permitir a entrada de parâmetros externos via argumentos para que possa ser feita a avaliação automática. A Figura 10 mostra o fluxograma de processos do funcionamento do módulo corretor de programas.

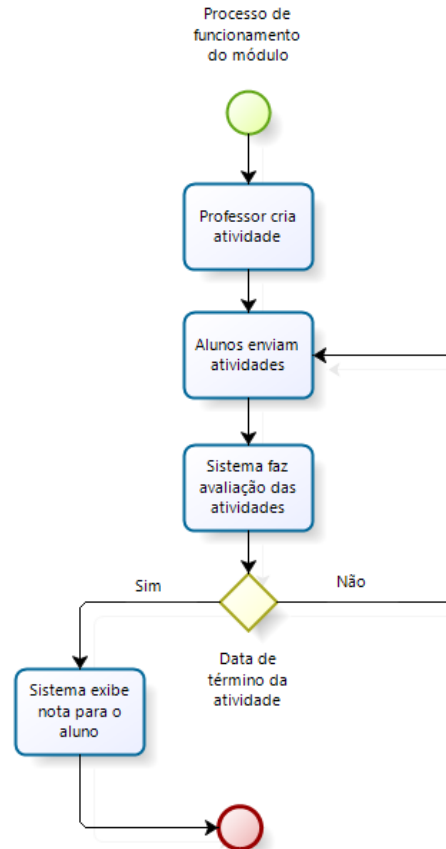


Figura 10 - Fluxograma de processos do funcionamento do módulo.

Fonte: Autoria própria

Para poder armazenar informações do módulo corretor de programas e das atividades enviadas por ele, são necessárias duas tabelas de banco de dados, sendo que uma será responsável por guardar as informações sobre a criação do módulo (título da atividade, descrição, parâmetros de entrada, resposta correta, data de início e fim da atividade, reenvio, etc) e outra será responsável por guardar informações sobre as atividades enviadas pelos alunos (nota, data de envio, feedback, etc). Estas tabelas foram criadas por meio da ferramenta Editor XMLDB do MOODLE. O módulo é compatível com todos os bancos de dados suportados pelo MOODLE e a partir da versão 2.4 do MOODLE.

3.1 DESENVOLVIMENTO MODULAR PARA MOODLE

Para iniciar o desenvolvimento de módulos para o MOODLE, existe um módulo padrão disponível para *download* em https://github.com/MOODLEhq/MOODLE-mod_newmodule, por meio dele, encontra-se um modelo de como se desenvolver para MOODLE.

Para a criação de tabelas de banco dados para módulo não é necessário conhecimento em linguagens de banco de dados, já que o MOODLE possui um recurso chamado Editor XMLDB. Esta ferramenta cria os arquivos XML que especificam como MOODLE deve criar suas tabelas no banco de dados, também é possível por meio dele a edição das tabelas. Para utilizar o XMLDB, é necessário acessar o bloco de administração, ir na opção administração do site, desenvolvimento Editor XMLDB. A Figura 11 mostra a edição de uma tabela no banco por meio do editor.

Figura 11 - Editor XMLDB

Fonte: Autoria própria

Antes de utilizar o Editor XMLDB, é necessário a criação de uma pasta chama db no diretório do novo módulo (Ex: mymodule/db), nesta pasta o editor irá criar automaticamente todos os arquivos necessários para trabalhar com o banco. A Figura 12, exhibe a pasta db e os arquivos criados pelo Editor XMLDB.

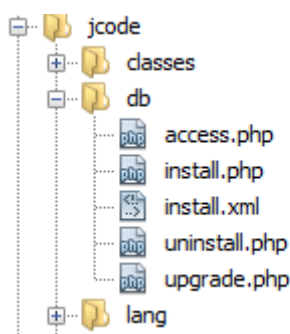


Figura 12 - Arquivos do banco de dados do módulo.

Fonte: Autoria própria

Cada arquivo da pasta jocode/db tem uma responsabilidade específica no módulo, os arquivos são:

- a) **Access.php:** arquivo em que é configurada as permissões dos tipos de usuários do MOODLE;
- b) **Install.xml:** este arquivo é utilizado na instalação do módulo, ele é responsável por criar as novas tabelas no banco de dados;
- c) **Install.php:** Comandos básicos para instalação, para casos em que é necessário executar algum processo extra na instalação do módulo.
- d) **Uninstall.php:** comando básico para desinstalar o módulo. Também pode ser alterado para casos em que é necessário realizar alguma alteração extra;
- e) **Upgrade.php:** arquivo utilizado quando é necessário fazer uma atualização no banco de dados, principalmente quando o MOODLE já está em ambiente de produção.

O controle de idiomas do módulo é feito dentro da pasta lang, dentro dela, é criada uma subpasta para cada idioma que o módulo será traduzido. A pasta de cada idioma contém um arquivo php nomeado com o mesmo nome do módulo (Ex: mymodule/lang/em/mymodule.php), neste arquivo é informada a tradução de cada string do módulo. O Quadro 3, mostra um trecho do código responsável pela tradução.

```

$string['grade'] = 'Nota';
$string['disabled'] = 'Sem Data';
$string['feedback'] = 'Comentário';
$string['remove_alert'] = 'Deseja realmente eliminar a sua postagem?';
$string['remove'] = 'Deletar';
$string['removed'] = 'Arquivo removido com sucesso.';
$string['jcode:addinstance'] = 'Adicionar novo Jcode';
$string['jcode:manager'] = 'Gerenciar Jcode';
$string['jcode:view'] = 'Visualizar Jcode';
$string['jcode:submit'] = 'Enviar atividade no Jcode ';

```

Quadro 3 - Trecho de código de tradução de idiomas do módulo

Fonte: Autoria própria

Outros arquivos necessários para a criação de um novo módulo para o MOODLE são:

- a) Lib.php: no arquivo lib.php é feita a comunicação entre o módulo e o MOODLE, por meio deste arquivo são criadas e deletadas instancias do módulo.
- b) Mod_form.php: é o formulário de criação do modulo, utiliza a API de formulário do MOODLE;
- c) Index.php: é executado na listagem de atividades por tipo. Mostra uma lista com todas as atividades do mesmo tipo com as descrições de cada uma.
- d) View.php: é o arquivo executado ao acessar diretamente a atividade, serve como página inicial da atividade.
- e) Version.php: Arquivo de controle de versão do modulo, além da sua própria versão ele controla se um modulo precisa de atualização, também com a versão requerida ele informa a versão do MOODLE que ele pode ser instalado, e o cron indica a frequência com que o MOODLE irá executar a função de manutenção do modulo.

3.2 CÓDIGO USADO NOS TESTES

Para realizar testes no módulo desenvolvido, foi utilizado um programa em JAVA que realiza soma de dois números, este programa também foi utilizado para demonstrar o funcionamento do módulo no decorrer deste trabalho. Este programa foi desenvolvido de modo que permite a entrada de parâmetros externos na sua execução. O quadro 4 mostra o código-

fonte do programa JAVA que realiza a soma de dois números, utilizado para demonstrar o funcionamento do módulo.

```
public class Soma{
    public static void main(String args[]){

        int resultado = 0;
        int valor1 = Integer.parseInt(args[0]);

        int valor2 = Integer.parseInt(args[1]);
        resultado = valor1 / valor2;
        System.out.println(resultado);

    }
}
```

Quadro 4 – Código fonte do programa de soma de dois números

Fonte: Autoria própria

3.3 TECNOLOGIAS UTILIZADAS

Este capítulo tem como objeto mostrar as tecnologias utilizadas no desenvolvimento do trabalho.

3.3.1 PHP

O PHP é uma linguagem de programação de código aberto que permite criar sites *web* dinâmicos, possibilitando uma interação com o usuário por meio de formulários, parâmetros da URL e links. O código PHP é executado em um servidor, sendo enviado para o cliente somente em HTML puro. Desta maneira é possível interagir com bancos de dados e aplicações existentes no servidor, com a vantagem de não expor o código fonte para o cliente. Isso pode ser útil quando o programa trabalha com senhas ou qualquer tipo de informação confidencial.

A linguagem PHP surgiu como um sucessor de um produto chamado PHP/FI. Criado por Rasmus Lerdorf em 1994, a linguagem começou com um conjunto simples de binários Common Gateway Interface (CGI) escrito em linguagem C, sendo que no começo era apenas utilizado para o acompanhamento de visitas para um curriculum online, sendo nomeado de "Personal Home Page Tools". Com o passar do tempo Rasmus foi adicionando funcionalidades ao sistema tornando-o capaz de fazer interações com bando de dados e também dando aos usuários a possibilidade de desenvolver algumas pequenas aplicações *web*. Em 1995 Rasmus disponibilizou o seu código fonte sendo que assim permitiu que outros desenvolvedores fornecessem correções para alguns bugs no código. Com o passar dos anos a linguagem foi se expandindo e ganhando novas funcionalidades até se tornar uma das linguagens de programação *web* mais utilizada no mundo (PHP, 2015).

O PHP é utilizado juntamente com o HTML em uma página *web*, sendo que o interpretador identifica apenas as linhas em PHP para que seja feita a execução. Para delimitar o código PHP dentro do HTML são utilizadas as tags `<? php` ou `<? e ?>`.

O Quadro 5 mostra um trecho de código implementado na linguagem de programação PHP responsável por montar a tabela com as informações das atividades enviadas pelos alunos.

```
53
54     $table = new html_table();
55     if ($course->format == 'weeks') {
56         $table->head = array(get_string('week'), get_string('name'));
57         $table->align = array('center', 'left');
58     } else if ($course->format == 'topics') {
59         $table->head = array(get_string('topic'), get_string('name'));
60         $table->align = array('center', 'left', 'left', 'left');
61     } else {
62         $table->head = array(get_string('name'));
63         $table->align = array('left', 'left', 'left');
64     }
```

Quadro 5 - Exemplo de código em PHP

Fonte: Autoria própria

3.3.2 WampServer

WampServer é um ambiente para o sistema operacional Windows, voltado para programadores que precisam testar aplicações *web*, suportando Apache, PHP e banco de dados MySQL.

Trata-se de um pacote com vários programas, que instala em seu computador de uma só vez, o Apache, MySQL *database*, PHP5, PHPmyadmin e SQLitemanager para gerenciar bancos de dados com facilidade, permitindo construir sites/páginas dinâmicas que podem ser publicadas na internet (WAMP, 2014).

3.3.3 Banco de dados MYSQL

O MySQL é um gerenciador de banco de dados relacional que suporta SQL. É *opensource* e está entre os mais utilizados SGBDs (Sistema Gerenciador de Banco de Dados).

Foi desenvolvido pela empresa MySQL AB *Limited Company*, algumas características são: portabilidade (suporta qualquer plataforma), excelente desempenho e estabilidade, facilidade de uso, suporta triggers, possui interface gráfica (MySQL Toolkit) de fácil utilização, entre outras. É um gerenciador de banco de dados multiusuário, que dá ao *software* (programa) a possibilidade de trabalhar em rede com diversos acessos simultâneos e multitarefa. Sua implementação segue o seu próprio conjunto de comandos SQL, porém o conceito básico do SQL permanece (MORAZ, 2005).

A integração fácil do MySQL com a linguagem de programação PHP, faz com que o uso desse banco de dados seja de grande demanda. Uma de suas peculiaridades são suas licenças para uso gratuito, tanto para fins estudantis como para realização de negócios, possibilitando que empresas o utilizem livremente (MILANI, 2007).

Dentre as vantagens do MySQL estão o suporte gratuito, o suporte a diferentes plataformas, como Windows e Linux, a facilidade de uso, a performance e confiabilidade, ser um *software open source* e suportar APIs de diversas linguagens como PHP, C, C++, Java e outras (MORAZ, 2005).

3.3.4 Netbeans

O domínio de uma linguagem de programação não basta para que um desenvolvedor desenvolva um trabalho com eficiência. Para obter facilitar e agilizar o desenvolvimento de um *software*, é importante ter o domínio de um Ambiente Integrado de Desenvolvimento (IDE).

Uma IDE aumenta a produtividade do programador, pois ele automatiza parte do trabalho, liberando o programador para pensar na lógica do programa.

O NetBeans é uma IDE gratuita, permite o desenvolvimento em diversas linguagens de programação, como Java, PHP, C, C++ entre outras. Sua origem vem do projeto Xelfi, criado em 1996 por dois universitários thecos. Em 1999, após se tornar um IDE proprietário denominado NetBeans DeveloperX2, foi adquirido pela Sun Microsystems, que optou em torna-lo um projeto *open source*. Em 2009 a Oracle com a Sun Microsystems, mas decidiu manter o projeto com código aberto (NETBEANS, 2015).

O IDE NetBeans permite o desenvolvimento de aplicações para a internet, desktop ou também para dispositivos moveis. Encontra-se disponível para *download* em diversas versões no site netbeans.org.

3.3.5 JAVA

O JAVA é uma linguagem de programação orientada objetos, lançada em 1995 pela empresa Sun Microsystems, sendo comprada anos depois pela Oracle. A linguagem JAVA faz parte do núcleo da plataforma JAVA, que é um dos ambientes de desenvolvimento de aplicações mais utilizados no mundo (JAVA, 2015).

A sintaxe da linguagem de programação JAVA, deriva do C++ porém com um modelo mais simples. Como sua principal característica, todo o código é escrito dentro de uma classe e tudo é um objeto, com exceção dos tipos intrínsecos (números reais e ordinais, valores booleanos e caracteres) que não são classes por questões da funcionalidade exercida (JAVA, 2015).

As aplicações em Java normalmente podem ser executadas em qualquer plataforma que possua a instalada, independente da arquitetura do computador.

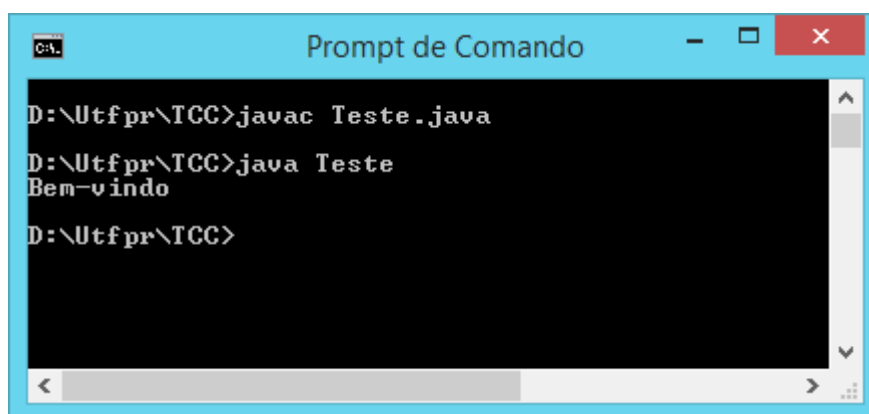
Os programas em JAVA rodam em vários sistemas operacionais por meio de uma máquina virtual chama da Java *Virtual Machine* (JVM), podendo ser emulada em sistemas que suportem a linguagem C++. Esta máquina virtual é um programa que carrega e executa os aplicativos JAVA, convertendo os bytecodes em código executável de máquina, gerenciando os aplicativos a medida que são executados.

Para o desenvolvimento de aplicações em JAVA é necessário a instalação do JDK (Java Development Kit), este pacote contém as APIs, o compilador e o JRE (Java Runtime

Environment), por meio deste conjunto de ferramentas é possível realizar todo o processo de desenvolvimento e execução do aplicativo (JAVA, 2015).

O processo de compilação do Java passa por dois processos, interpretação e compilação. Ao criar uma classe em Java e salvar com a extensão .java é gerado apenas um arquivo com código Java, não podendo assim ser executado pela JVM. Para isto é necessário que transformem este código em linguagem de máquina por meio do processo de compilação. Ao compilar o código é gerado um arquivo com a extensão .class, que são os bytecodes que poderão ser interpretados pela JVM (JAVA, 2015).

Um programa em Java pode ser compilado via linha de comando, para isto, é necessário abrir o diretório da classe JAVA e utilizar o comando “javac” seguido de um espaço e o nome da classe junto com sua extensão. Depois deste processo é possível executar o programa utilizando o comando “java” seguido de um espaço e o nome do arquivo gerado pelo processo sem a compilação (sem sua extensão). A Figura 13 mostra como é compilado e executado um programa em JAVA via linha de comando.



```

C:\> Prompt de Comando
D:\Utfpr\TCC>javac Teste.java
D:\Utfpr\TCC>java Teste
Bem-vindo
D:\Utfpr\TCC>
```

Figura 13 - Compilação e execução de um programa em JAVA

Fonte: Autoria Própria

Caso o compilador encontre algum erro no código durante o processo de compilação, ele irá exibir uma mensagem na tela informando o erro e apontando o local em que o erro foi encontrado. A Figura 14 mostra como é exibido um erro ao compilar um programa Java via linha de comando.

A screenshot of a Windows Command Prompt window titled "Prompt de Comando". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The command prompt shows the following text:

```
D:\Utfpr\TCC>javac Teste.java
Teste.java:5: error: ';' expected
System.out.println("Bem-vindo")
                        ^
1 error
D:\Utfpr\TCC>
```

The error message indicates a semicolon is missing at the end of the print statement on line 5. A caret (^) points to the end of the string "Bem-vindo".

Figura 14 - Compilação de um programa em JAVA com erro.

Fonte: Autoria Própria

4 RESULTADOS E DISCUSSÕES

Para demonstrar o funcionamento do módulo corretor de programas, foi instalado em um servidor local a versão 2.8 do AVA MOODLE, para que assim seja instalado o módulo no AVA.

4.1 VISÃO GERAL DO MÓDULO

O módulo foi desenvolvido utilizando a linguagem de programação PHP e como banco de dados foi utilizado o MySQL. O módulo é compatível com a versão 2.4 ou superior do MOODLE, também suporta todos os bancos de dados compatíveis com o MOODLE.

O módulo permite que o professor crie uma atividade no MOODLE para envio de arquivos, capaz de corrigir e avaliar os códigos-fonte escritos em linguagem de programação JAVA enviados pelos alunos automaticamente, o módulo também permite que o professor realize a correção manual das atividades, para isto, há uma opção de download das atividades recebidas, para que assim ele possa realizar a correção manual. Também é possível escrever um *feedback* para atividade do aluno.

4.2 CRIANDO E CONFIGURANDO UMA ATIVIDADE DE AVALIAÇÃO

Para utilizar o recurso, após a instalação do módulo no MOODLE, o professor deverá criar uma atividade de correção de arquivos em seu curso. Para a criação de uma atividade deste tipo, é necessário que o professor ative o modo de edição do MOODLE e adicione uma nova atividade de correção de programas. A Figura 15 mostra adição de uma atividade de correção de programas.

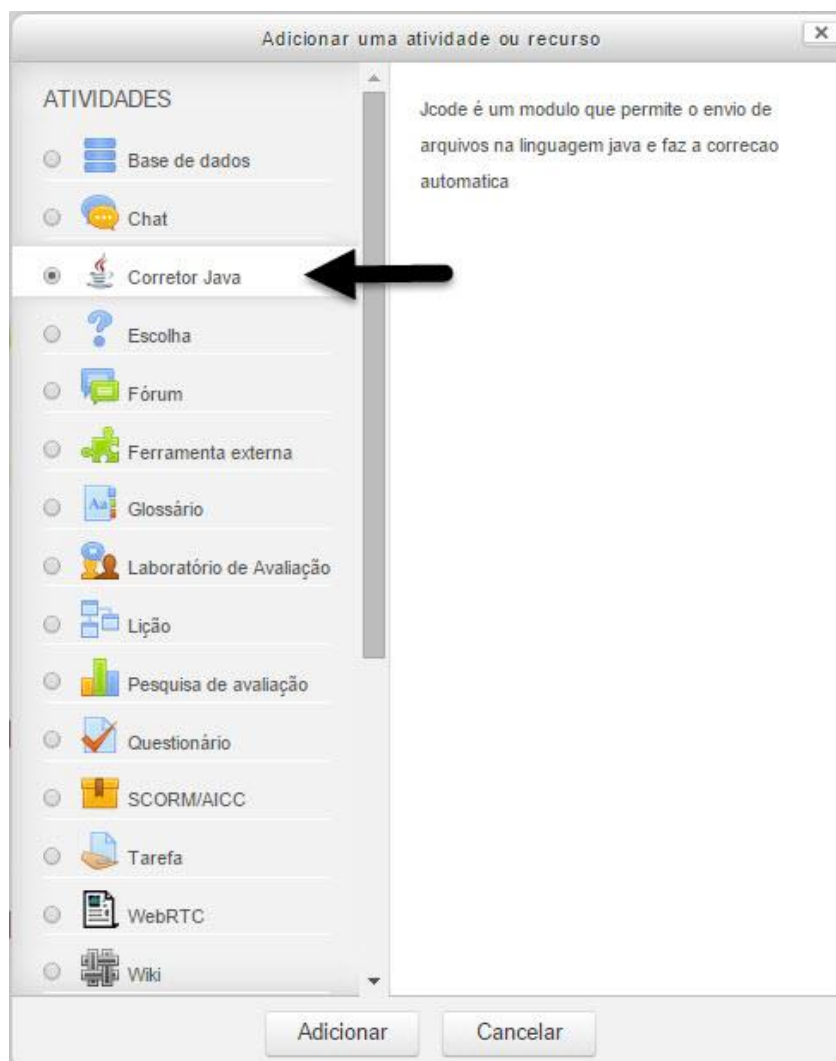


Figura 15 - Adicionando uma atividade de correção de programas

Fonte: Autoria própria

No formulário de criação da atividade, o professor terá de atribuir um título, descrever o problema para solução, informar quais os parâmetros que serão utilizados como entrada no processo de compilação automática, gabarito da atividade, permitir ou não mais de um envio por aluno, datas de início e término da atividade e configurações de notas. A Figura 16 mostra a tela de cadastro de uma nova atividade de correção de programas.

Nome da atividade*

Descrição

Exibir descrição na página do curso

Parâmetros

Resposta correta

Permitir Reenvio

Data de início Ativar

Data Final Ativar

Figura 16 - Formulário de criação do corretor de programas

Fonte: Autoria própria

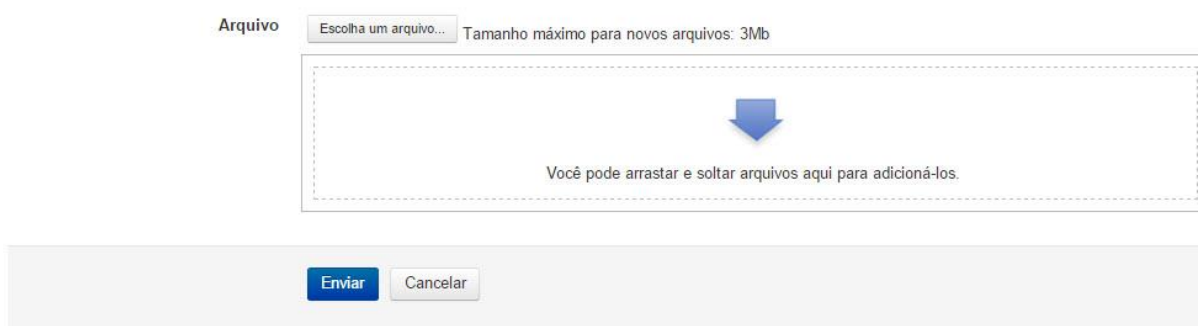
Os campos de criação de uma atividade de correção de arquivos são:

- a) **Nome da atividade:** neste campo o professor deverá informar um nome para atividade, este nome será exibido na tela inicial do curso;
- b) **Descrição:** neste campo o professor irá escrever a atividade a ser desenvolvida;
- c) **Parâmetros:** neste campo deverá ser informado quais são os valores de entrada que serão passados a atividade do aluno, no caso de mais de um valor, eles deverão estar separados por um espaço (Ex: 2 2);
- d) **Resposta correta:** neste campo o professor terá que informar o gabarito do programa. O preenchimento deste campo deverá estar de acordo com o campo “Parâmetros”, por exemplo, em um exercício que é proposto o desenvolvimento de um programa que faça a soma de dois números, o professor deverá cadastrar como resposta correta a soma dos dois valores que informou como parâmetros;
- e) **Permitir reenvio:** neste campo é dada ou não a permissão reenvio da atividade;
- f) **Data de início:** campo que indica a data em que inicia a atividade;
- g) **Data de fim:** campo que determina a data de encerramento da atividade;

4.3 ENVIO DA ATIVIDADE PELO ALUNO

Com a criação da atividade, o aluno poderá fazer o envio de seu programa para correção quando a data de início chegar. Para fazer o envio, basta que ele acesse a atividade e faça o *upload* de seu arquivo com o código fonte de sua atividade. Ao fazer o envio arquivo com o código fonte, o módulo irá tentar compilar o programa do aluno e caso não encontre nenhum erro no código, irá executá-lo passando os parâmetros informados pelo professor no cadastro da atividade, após a execução, será comparado o resultado de saída do programa, com a resposta informada pelo professor na criação da atividade, a solução será considerada correta caso esta comparação identifique que os valores são iguais. A Figura 17 mostra a tela de envio de atividades pelos alunos.

Envio de Atividade



Arquivo Escolha um arquivo... Tamanho máximo para novos arquivos: 3Mb

Você pode arrastar e soltar arquivos aqui para adicioná-los.

Enviar Cancelar

Figura 17 - Envio de uma atividade para correção

Fonte: Autoria própria

4.4 AVALIAÇÃO DAS ATIVIDADES

4.4.1 Avaliação de atividade com erro de compilação

Caso o aluno envie um programa com erros de código, (Ex: faltando fechar uma chave) não sendo possível a compilação do mesmo, o corretor não irá aceitar o envio da atividade e exibir uma tela informando que houve erro ao compilar o arquivo, reencaminhando o aluno para a tela de envio de atividades, como o sistema não aceita o envio da atividade neste caso, o professor não receberá nenhuma informação sobre a mesma e o aluno ficará com atividade

pendente. A Figura 18 mostra a tela exibida ao aluno, quando um arquivo com erro de código é enviado.

Exercício de Fixação

Desenvolva uma atividade que realize a soma de dois números.

Envio de Atividade

Erro ao compilar arquivo

Processando arquivo, aguarde por favor...

(Continuar)

Figura 18 - Tela exibida ao enviar um programa com erro de código

Fonte: Autoria própria

4.4.2 Avaliação de atividade compilada

Ao fazer o upload de um arquivo sem erros de código, o sistema irá compilar o arquivo e passar os parâmetros de entrada para a execução do programa. Após a execução do programa, o resultado de saída será comparado com a saída esperada cadastrada pelo professor, caso a comparação identifique que os dois valores são iguais o sistema irá atribuir nota cem para a atividade do aluno, caso a comparação identifique que são valores diferentes, a nota da atividade será zerada. A Figura 19 mostra a tela exibida ao aluno após o envio de uma atividade sem erros de código, neste exemplo a atividade foi configurada para exibir a nota ao aluno somente após a data final.

Exercício de Fixação

Desenvolva uma atividade que realize a soma de dois números.

Envio de Atividade

Data de entrega	domingo, 24 mai 2015, 19:44 Deletar
Data de início	domingo, 24 mai 2015, 19:15
Data Final	quinta-feira, 28 mai 2015, 19:15

Figura 19 - Tela exibida ao aluno após o envio da atividade

Fonte: Autoria própria

A nota da atividade estará disponível instantaneamente para o professor assim que o sistema receber o exercício do aluno, já este só terá acesso a nota em tempo real, caso o professor não habilite a data de término da atividade, caso contrário ele terá que aguardar a data de fim da atividade para ter acesso a nota, o *feedback* deixado pelo professor também será exibido juntamente com a nota, quando a mesma for liberada. A Figura 20 mostra a tela exibida ao aluno após a data final da atividade.

Exercício de Fixação

Desenvolva uma atividade que realize a soma de dois números.

Envio de Atividade

Data de entrega	domingo, 24 mai 2015, 19:44
Data de início	domingo, 24 mai 2015, 19:15
Data Final	domingo, 24 mai 2015, 19:15
Nota	100
Comentário	Muito bom Maria, sua atividade está correta.

Figura 20 - Tela da atividade exibida ao aluno após a data final

Fonte: Autoria própria

Caso o professor queira avaliar manualmente a atividade, o corretor de programas permite que ele faça download do arquivo enviado e realize a correção manual, editando a nota do aluno, podendo fracioná-la.

A Figura 21 apresenta a tela de avaliação das atividades dos alunos pelo professor, nesta tela é exibida a nota do aluno, campo para *feedback*, data de entrega, arquivo enviado e o resultado da execução do programa enviado. Nesta tela o professor poderá editar a nota, deixar um *feedback* e fazer o download do arquivo enviado pelo aluno.

Avaliar Envios

Aluno	Nota	Feedback	Data de Entrega	Arquivo	Resultado
Maria Augusta	<input type="text" value="0"/> / 100	<input type="text"/>	quinta-feira, 7 mai 2015, 04:25	Soma.java	1
Pedro Silva	<input type="text" value="100"/> / 100	<input type="text"/>	quinta-feira, 7 mai 2015, 04:13	Soma.java	4

Figura 21 - Tela de avaliação de atividades

Fonte: Autoria própria

4.4.3 Múltiplos envios

Após o envio da atividade o aluno só poderá reenviar um novo exercício para correção, caso o professor, no processo de criação da atividade, tenha preenchido o campo “Permitir reenvio” com a opção “Sim”. A Figura 22 mostra o campo “Permitir reenvio” na tela de criação de uma atividade.

Exibir descrição na página do curso

Parâmetros

Resposta correta

Permitir Reenvio

Data de início Ativar

Data Final Ativar

Figura 22 - Campo “Permitir reenvio”

Fonte: Autoria própria

Para o envio de um novo exercício, primeiramente o aluno deverá deletar o arquivo já enviado, após deleta-lo, o sistema irá redirecionar o aluno para a página de envio de atividades, onde ele poderia fazer o *upload* de um novo arquivo. A Figura 23 mostra a opção “Deletar” para o aluno excluir um arquivo já enviado.

Exercício de Fixação

Desenvolva uma atividade que realize a soma de dois números.

Envio de Atividade

Data de entrega	domingo, 24 mai 2015, 19:44 Deletar
Data de início	domingo, 24 mai 2015, 19:15
Data Final	quinta-feira, 28 mai 2015, 19:15

Figura 23 – Opção para deletar um arquivo já enviado

Fonte: Autoria própria

A opção “Deletar” é ocultar após a data final da atividade configurada pelo professor, deste modo, o reenvio de atividades fica bloqueado a todos os alunos.

4.5 COMPILAÇÃO DAS ATIVIDADES

No Quadro 4 é apresentado o trecho de código responsável pela compilação e execução dos códigos enviados pelos alunos. O função `exec()` do PHP é utilizada para poder executar comandos em Shell, isso se faz necessário para que o módulo possa invocar o compilador JAVA instalado no servidor. O compilador JAVA irá retornar o valor 1 caso encontre algum erro no código e o valor 0 caso o programa compile sem nenhum erro. Após compilar o programa, a variável `$file_info` armazena o caminho em que o programa foi salvo e logo em seguida ele é executado e avaliado, comparando o resultado de saída com a resposta cadastrada pelo professor na criação da atividade.

```

42 // compilando
43 exec ( "javac $save_path$filename", $resultado );
44 if ($resultado == "1") {
45     echo "Erro ao compilar arquivo";
46 }
47
48 $file_info = pathinfo ( $filename );
49 // executando o arquivo
50 exec ( "cd $save_path && java {$file_info['filename']} {$jcode->parameters} -cp 2>&1", $out, $success );
51 $response = current ( $out );
52 $out = serialize ( $out );
53
54
55 // salvando registro
56 if ($success == 0) {
57     $post_file = new stdClass ();
58     $post_file->jcode_id = $jcode->id;
59     $post_file->filename = $filename;
60     $post_file->user_id = $USER->id;
61     $post_file->submit_date = time ();
62     if ($jcode->correctanswer == $response) {
63         $post_file->grade = $jcode->grade;
64     } else {
65         $post_file->grade = 0;
66     }
67     $post_file->result = $out;
68     $DB->insert_record ( 'jcode_files', $post_file );
69 }

```

Quadro 6 - Código responsável pela compilação e avaliação das atividades

Fonte: Autoria própria

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÃO

A integração de um módulo corretor automático de programas ao AVA MOODLE reduz significativamente o trabalho do professor de disciplinas de programação, já que o módulo faz todo o trabalho de correção e avaliação automaticamente, sem a necessidade fazer a leitura manual de todo código.

Os corretores automáticos de código são ferramentas que se aplicadas em disciplinas de programação, além de facilitar o trabalho do professor, garantirá a correção e avaliação de todos os trabalhos dos alunos, evitando que alunos façam o envio de um arquivo qualquer recebam nota pela sua atividade.

5.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

Como ideia para trabalhos futuros, expandir o módulo para correção de programas em outras linguagens de programação. Também pode-se implementar uma opção ao professor, permitindo que ele possa inserir um arquivo com um conjunto de entradas e outro contendo as respectivas saídas, ao receber a atividade do aluno, o sistema compila o programa recebido, caso não encontre erros, e o executa passando-o os dados do arquivo que contém o conjunto de entradas, após isto, ele compara o resultado da execução com o arquivo que contém as saídas esperadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- AULANET. Disponível em <http://www.MOODLE.ufba.br/mod/book/view.php?id=18148&chapterid=12014>. Acesso em 08 de março de 2015.
- BEZ, Jean Luca; TONIN, Neilor A. URI ONLINE JUDGE E A INTERNACIONALIZAÇÃO DA UNIVERSIDADE
- BOCA Online Contest Administrator. Disponível em <http://www.ime.usp.br/~cassio/boca/> Acesso 15 de outubro de 2014.
- CAMPOS, Cassio P. de; FERREIRA, Carlos E. BOCA: um sistema de apoio a competições de programação, 2004.
- DEVMEDIA. Artigo Engenharia de Software – Introdução a teste de software. Disponível em <http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035> Acesso 04 de março de 2015.
- FALBO, Rodrigo A.; Engenharia de Software – Notas de Aula. Disponível em <http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf> Acesso 25 de maio de 2015.
- JAVA. Disponível em https://www.java.com/pt_BR/about/ Acesso em 29 de abril 2015.
- MOODLE. MOODLE Docs. Disponível em: https://docs.MOODLE.org/28/en/Main_page >. Acesso em: 10 junho 2014.
- MOOSHAK. Disponível em <https://mooshak.dcc.fc.up.pt/>. Acesso em 05 de março 2015.
- MySQL. MySQL 5.0 Reference Manual. Disponível em <http://dev.mysql.com/doc/refman/5.0/en/>. Acesso em 08 de junho de 2014.
- NETBEANS. Disponível em <https://netbeans.org/features/index.html> Acesso em 07 de abril de 2015.
- PC2. Disponível em <http://www.ecs.csus.edu/pc2/>. Acesso em 05 de março 2015.
- PERREIRA, Christiane dos Santos. DIAS, Samuel. Pereira. SILVA, G. A UTILIZAÇÃO DE AMBIENTES VIRTUAIS DE APRENDIZAGEM COMO FERRAMENTA DE APOIO AO ENSINO PRESENCIAL: estudos preliminares e proposta de uma metodologia de implantação no CEFET-BambuÍ. Disponível em http://www.cefetbambui.edu.br/str/artigos_aprovados/informatica/61-CO-5.pdf.
- PHP. Documentação PHP. Disponível em: <http://www.php.net/docs.php>. Acesso 06 de junho de 2014.

PRESSMAN, R. S. Engenharia de Software . 3. ed. São Paulo: MaKron Bo ok do Brasil, 1995.

RIBEIRO, Elvia N. R.; ARAÚJO GILDA. A.; MENDONÇA, Alzino F.A IMPORTÂNCIA DOS AMBIENTES VIRTUAIS DE APRENDIZAGEM NA BUSCA DE NOVOS DOMÍNIOS DA EAD, 2007. Disponível em <<http://www.abed.org.br/congresso2007/tc/4162007104526AM.pdf>> Acesso em 01 de maio de 2015

SILVA, Robson. S. MOODLE para autores e tutores. 2010.

SOUZA. Alan. Adaptação de um Corretor de Automático de Código para Auxílio da Atividade de Exercício e Prática no Ensino de Programação, 2009. Disponível em <<http://www.pergamum.udesc.br/dados-bu/000000/000000000000D/00000D51.pdf>>.

TELEDUC. Disponível em <http://fenix.nied.unicamp.br/redmine/projects/teleduc/wiki/Bem-Vindo_%C3%A0_Wiki_do_TelEduc> Acesso em 29 de março de 2015.

Wamp. WampServer. Disponível em <<http://www.wampserver.com/en/>>. Acesso em 07 de junho de 2014.