



**Ernane Rosa Martins
(Organizador)**

Princípios e Aplicações da Computação no Brasil 3

Atena
Editora
Ano 2019

Ernane Rosa Martins
(Organizador)

Princípios e Aplicações da Computação no Brasil 3

Atena Editora
2019

2019 by Atena Editora
Copyright © Atena Editora
Copyright do Texto © 2019 Os Autores
Copyright da Edição © 2019 Atena Editora
Editora Chefe: Profª Drª Antonella Carvalho de Oliveira
Diagramação: Lorena Prestes
Edição de Arte: Lorena Prestes
Revisão: Os Autores



Todo o conteúdo deste livro está licenciado sob uma Licença de Atribuição Creative Commons. Atribuição 4.0 Internacional (CC BY 4.0).

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores. Permitido o download da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

Conselho Editorial

Ciências Humanas e Sociais Aplicadas

Profª Drª Adriana Demite Stephani – Universidade Federal do Tocantins
Prof. Dr. Álvaro Augusto de Borba Barreto – Universidade Federal de Pelotas
Prof. Dr. Alexandre Jose Schumacher – Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso
Prof. Dr. Antonio Carlos Frasson – Universidade Tecnológica Federal do Paraná
Prof. Dr. Antonio Isidro-Filho – Universidade de Brasília
Prof. Dr. Constantino Ribeiro de Oliveira Junior – Universidade Estadual de Ponta Grossa
Profª Drª Cristina Gaio – Universidade de Lisboa
Prof. Dr. Deyvison de Lima Oliveira – Universidade Federal de Rondônia
Prof. Dr. Edvaldo Antunes de Faria – Universidade Estácio de Sá
Prof. Dr. Eloi Martins Senhora – Universidade Federal de Roraima
Prof. Dr. Fabiano Tadeu Grazioli – Universidade Regional Integrada do Alto Uruguai e das Missões
Prof. Dr. Gilmei Fleck – Universidade Estadual do Oeste do Paraná
Profª Drª Ivone Goulart Lopes – Istituto Internazionele delle Figlie di Maria Ausiliatrice
Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense
Profª Drª Keyla Christina Almeida Portela – Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso
Profª Drª Lina Maria Gonçalves – Universidade Federal do Tocantins
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Prof. Dr. Marcelo Pereira da Silva – Universidade Federal do Maranhão
Profª Drª Miranilde Oliveira Neves – Instituto de Educação, Ciência e Tecnologia do Pará
Profª Drª Paola Andressa Scortegagna – Universidade Estadual de Ponta Grossa
Profª Drª Rita de Cássia da Silva Oliveira – Universidade Estadual de Ponta Grossa
Profª Drª Sandra Regina Gardacho Pietrobon – Universidade Estadual do Centro-Oeste
Profª Drª Sheila Marta Carregosa Rocha – Universidade do Estado da Bahia
Prof. Dr. Rui Maia Diamantino – Universidade Salvador
Prof. Dr. Urandi João Rodrigues Junior – Universidade Federal do Oeste do Pará
Profª Drª Vanessa Bordin Viera – Universidade Federal de Campina Grande
Prof. Dr. Willian Douglas Guilherme – Universidade Federal do Tocantins

Ciências Agrárias e Multidisciplinar

Prof. Dr. Alan Mario Zuffo – Universidade Federal de Mato Grosso do Sul
Prof. Dr. Alexandre Igor Azevedo Pereira – Instituto Federal Goiano
Profª Drª Daiane Garabeli Trojan – Universidade Norte do Paraná
Prof. Dr. Darllan Collins da Cunha e Silva – Universidade Estadual Paulista
Profª Drª Diocléa Almeida Seabra Silva – Universidade Federal Rural da Amazônia
Prof. Dr. Fábio Steiner – Universidade Estadual de Mato Grosso do Sul
Profª Drª Girlene Santos de Souza – Universidade Federal do Recôncavo da Bahia
Prof. Dr. Jorge González Aguilera – Universidade Federal de Mato Grosso do Sul
Prof. Dr. Júlio César Ribeiro – Universidade Federal Rural do Rio de Janeiro
Profª Drª Raissa Rachel Salustriano da Silva Matos – Universidade Federal do Maranhão
Prof. Dr. Ronilson Freitas de Souza – Universidade do Estado do Pará
Prof. Dr. Valdemar Antonio Paffaro Junior – Universidade Federal de Alfenas

Ciências Biológicas e da Saúde

Prof. Dr. Benedito Rodrigues da Silva Neto – Universidade Federal de Goiás
Prof. Dr. Edson da Silva – Universidade Federal dos Vales do Jequitinhonha e Mucuri
Profª Drª Elane Schwinden Prudêncio – Universidade Federal de Santa Catarina
Prof. Dr. Gianfábio Pimentel Franco – Universidade Federal de Santa Maria
Prof. Dr. José Max Barbosa de Oliveira Junior – Universidade Federal do Oeste do Pará
Profª Drª Magnólia de Araújo Campos – Universidade Federal de Campina Grande
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Profª Drª Vanessa Lima Gonçalves – Universidade Estadual de Ponta Grossa
Profª Drª Vanessa Bordin Viera – Universidade Federal de Campina Grande

Ciências Exatas e da Terra e Engenharias

Prof. Dr. Adélio Alcino Sampaio Castro Machado – Universidade do Porto
Prof. Dr. Alexandre Leite dos Santos Silva – Universidade Federal do Piauí
Profª Drª Carmen Lúcia Voigt – Universidade Norte do Paraná
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná
Prof. Dr. Fabrício Menezes Ramos – Instituto Federal do Pará
Prof. Dr. Juliano Carlo Rufino de Freitas – Universidade Federal de Campina Grande
Profª Drª Neiva Maria de Almeida – Universidade Federal da Paraíba
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista

Dados Internacionais de Catalogação na Publicação (CIP) (eDOC BRASIL, Belo Horizonte/MG)	
P957	<p>Princípios e aplicações da computação no Brasil 3 [recurso eletrônico] / Organizador Ernane Rosa Martins. – Ponta Grossa, PR: Atena Editora, 2019. – (Princípios e aplicações da computação no Brasil; v. 3)</p> <p>Formato: PDF Requisito de sistema: Adobe Acrobat Reader Modo de acesso: World Wide Web Inclui bibliografia ISBN 978-85-7247-702-4 DOI 10.22533/at.ed.024191510</p> <p>1. Computação. 2. Informática. 3. Programação de computador. I. Martins, Ernane Rosa. II. Série.</p> <p style="text-align: right;">CDD 004</p>
Elaborado por Maurício Amormino Júnior – CRB6/2422	

Atena Editora
Ponta Grossa – Paraná - Brasil
www.atenaeditora.com.br
contato@atenaeditora.com.br

APRESENTAÇÃO

O volume 3 da referida obra “Princípios e Aplicações da Computação no Brasil” apresenta 15 novos capítulos, que abordam assuntos importantes sobre o panorama atual da computação no Brasil, tais como: desenvolvimento de aplicativos móveis, VoIP, modelagem computacional, simulação, recurso educacional aberto, projeto de banco de dados, mobile learning, robótica, avaliação automática de código-fonte e agrupamento difuso multivariado.

Deste modo, esta obra reúne debates e análises acerca de questões relevantes, tais como: desenvolvimento de uma aplicação móvel para realização de recargas, consultas e simulação de saldo na carteira de transporte público, usando o método Design Constructal para melhorar a configuração geométrica de um novo layout para TCSA com quatro dutos; análise dos benefícios da tecnologia VoIP como alternativa de baixo custo a um serviço de telefonia institucional gerenciado por minicomputadores Raspberry e adaptadores ATA; análise do comportamento mecânico de placas finas de materiais compósitos laminados reforçados por fibras submetidas a esforços de flexão, comparando-as com o de uma placa de aço, usando modelagem computacional utilizando o software Abaqus 2017 Student Version®; apresentação de uma arquitetura de sistemas de simulação para área de convergência; desenvolvimento de uma abordagem para auxiliar o docente na criação de atividades com REA's que quando executados pelos alunos gerem o registro de suas experiências; análise da gestão de recursos humanos na administração pública por meio de um projeto de banco de dados; análise dos recursos disponíveis no WhatsApp Messenger, que oferecem possibilidades de uso no ambiente escolar como recurso pedagógico que pode auxiliar o processo de ensino e aprendizagem; simulação de um agente robótico capaz de realizar busca e exploração visual num ambiente virtual; e o desenvolvimento de uma ferramenta de análise e avaliação automática de código-fonte como alternativa para a otimização do processo de ensino-aprendizagem de programação.

Nesse sentido, esta obra apresenta enorme potencial para contribuir com análises e discussões aprofundadas sobre assuntos relevantes da área da computação, podendo servir de referência para novas pesquisas e estudos nesta área. Agradecemos em especial aos autores dos capítulos apresentados, e desejamos aos leitores, inúmeras reflexões proveitosas sobre as temáticas abordadas nesta obra.

Ernane Rosa Martins

SUMÁRIO

CAPÍTULO 1	1
WALLET BUS - TECNOLOGIA E TRANSPORTE PÚBLICO	
Carlos Adriano Carneiro Pestana Juan Castro Ribeiro Leonardo Carlos Balbino	
DOI 10.22533/at.ed.0241915101	
CAPÍTULO 2	16
UM PROJETO CONSTRUCTAL DE TROCADORES DE CALOR SOLO-AR COMPOSTOS POR QUATRO DUTOS	
Ruth da Silva Brum Liércio André Isoldi Jairo Valões de Alencar Ramalho Luiz Alberto Oliveira Rocha Michel Kepes Rodrigues Elizaldo Domingues dos Santos	
DOI 10.22533/at.ed.0241915102	
CAPÍTULO 3	30
TELEFONIA DIGITAL DE BAIXO CUSTO COM VOIP E RASPBERRY: UM ESTUDO DE CASO NO IFPI CAMPUS ANGICAL DO PIAUÍ	
Francisco Alysson da Silva Sousa Matheus Lopes Pereira	
DOI 10.22533/at.ed.0241915103	
CAPÍTULO 4	40
SIMULAÇÃO NUMÉRICA PARA ANÁLISE DE DEFLEXÕES E TENSÕES EM PLACAS FINAS DE AÇO E DE MATERIAI COMPÓSITOS SUBMETIDAS A ESFORÇOS DE FLEXÃO	
João Vítor de Azambuja Carvalho Daniel Helbig	
DOI 10.22533/at.ed.0241915104	
CAPÍTULO 5	49
SIMULAÇÃO DE ÁREA DE CONVERGÊNCIA NUVEM-HPC	
Alessandro Kraemer	
DOI 10.22533/at.ed.0241915105	
CAPÍTULO 6	63
RECREATE: RECOMENDAÇÃO DE REA'S POR MEIO DA DESCOBERTA DO ESTILO DE APRENDIZAGEM PELA EXPERIÊNCIA DO ESTUDANTE E EXPECTATIVA DO PROFESSOR	
Marco Antonio Colombo Da Silva Marilde Terezinha Prado Santos	
DOI 10.22533/at.ed.0241915106	
CAPÍTULO 7	76
PROJETO DE BANCO DE DADOS COM FOCO NA ALOCAÇÃO DEPARTAMENTAL DE SERVIDORES PÚBLICOS E SUAS ATRIBUIÇÕES FUNCIONAIS	
José Nazareno Alves Rodrigues	
DOI 10.22533/at.ed.0241915107	

CAPÍTULO 8	87
M-LEARNING ATRAVÉS DO WHATSAPP: UMA ANÁLISE DE RECURSOS	
Nilson Alves dos Santos	
Mayara Kaynne Fragoso Cabral	
Beatriz Francisco Coelho	
DOI 10.22533/at.ed.0241915108	
CAPÍTULO 9	98
EXPLORAÇÃO E BUSCA VISUAL ROBÓTICA EM AMBIENTE SIMULADO	
Felipe Duque Belfort de Oliveira	
Hansenclever de França Bassani	
Aluizio Fausto Ribeiro Araujo	
DOI 10.22533/at.ed.0241915109	
CAPÍTULO 10	112
ESTUDO SOBRE A ESPECTROSCOPIA NIR PARA DETECÇÃO DONÍVEL ALCOÓLICO NO SANGUE	
Luis Gustavo Longo da Silva	
Cláudio Luis D'Elia Machado	
Fabiano Sandrini Moraes	
Rafael Galli	
DOI 10.22533/at.ed.02419151010	
CAPÍTULO 11	125
ESTUDO COMPUTACIONAL DO COEFICIENTE DE IMPACTO VERTICAL EM LONGARINAS DE PONTES RODOVIÁRIAS BRASILEIRAS	
Vinícius Heidtmann Avila	
Mauro de Vasconcellos Real	
Márcio Wrague Moura	
DOI 10.22533/at.ed.02419151011	
CAPÍTULO 12	140
DESENVOLVIMENTO DE UMA ABORDAGEM DE CLASSIFICAÇÃO DE TRAÇOS DE PERSONALIDADE DO BIG-FIVE VIA ANÁLISE DE TEXTOS EM PORTUGUÊS DO BRASIL	
Angelo Travizan Neto	
Taís Borges Ferreira	
Márcia Aparecida Fernandes	
DOI 10.22533/at.ed.02419151012	
CAPÍTULO 13	148
CODETEACHER: UMA FERRAMENTA PARA CORREÇÃO AUTOMÁTICA DE TRABALHOS ACADÊMICOS DE PROGRAMAÇÃO EM JAVA	
Francisco Alan de Oliveira Santos	
Plácido Soares das Chagas Segundo	
Mardoqueu Sousa Telvina	
DOI 10.22533/at.ed.02419151013	
CAPÍTULO 14	158
CLASSIFICAÇÃO DE TENDÊNCIAS POLÍTICAS EM NOTÍCIAS VIA MINERAÇÃO DE TEXTO E REDES NEURAIS SEM PESO	
Rafael Dutra Cavalcanti	
DOI 10.22533/at.ed.02419151014	

CAPÍTULO 15	169
ANÁLISE DE DADOS DE EXPRESSÃO GÊNICA POR MEIO DE MÉTODOS DE AGRUPAMENTO DIFUSO MULTIVARIADO	
Bruno Almeida Pimentel	
Marcilio Carlos Pereira de Souto	
DOI 10.22533/at.ed.02419151015	
SOBRE O ORGANIZADOR.....	182
ÍNDICE REMISSIVO	183

CODETEACHER: UMA FERRAMENTA PARA CORREÇÃO AUTOMÁTICA DE TRABALHOS ACADÊMICOS DE PROGRAMAÇÃO EM JAVA

Francisco Alan de Oliveira Santos

Instituto Federal de Educação do
Maranhão – IFMA
Coelho Neto – Maranhão

Plácido Soares das Chagas Segundo

Instituto Federal de Educação do
Maranhão – IFMA Grajaú – Maranhão

Mardoqueu Sousa Telvina

Instituto Federal de Educação do
Maranhão – IFMA
Coelho Neto – Maranhão

RESUMO: Exercícios práticos são essenciais no aprendizado de uma linguagem de programação. Com o objetivo de auxiliar o professor na avaliação de exercícios de programação, este artigo apresenta o CodeTeacher, uma ferramenta de análise e avaliação automática de código-fonte como alternativa para a otimização do processo de ensino-aprendizagem de programação. A intenção é fomentar a discussão na comunidade educacional em Computação e propor uma solução no sentido de encontrar novas abordagens para esse desafio, contribuindo para o progresso do ensino de programação.

PALAVRAS-CHAVE: Avaliação Automática, Exercícios de Programação, Java.

ABSTRACT: Practical exercises are essential to learn a programming language. With the aim of assisting the teacher in the assessment of programming exercises, this paper presents CodeTeacher, a tool for automatic analysis and evaluation of source code as an alternative for the optimization of the teaching-learning programming process. The intention is to call the educational computing community to discussion and propose a solution to find new approaches to this challenge, thus contributing to the progress of teaching programming.

KEYWORDS: Automatic Assessment, Programming Exercises, Java.

1 | INTRODUÇÃO

O ensino de programação envolve a retenção de vários conhecimentos, como: noções de lógica, técnicas de programação, uso correto da sintaxe e recursos de uma linguagem, além da aplicação de boas práticas de desenvolvimento de software, dentre outros. A avaliação desses quesitos requer uma análise minuciosa por parte do professor, uma vez que é necessária a revisão do código produzido pelos alunos para identificar a manifestação do conhecimento adquirido, geralmente materializado na forma de trabalhos práticos [Prior 2003]. Devido à quantidade de detalhes

a serem observados, faz-se necessário o acompanhamento individual para uma aprendizagem mais acurada. [Tobar et al 2001].

Porém, a tarefa de avaliar o conhecimento do aluno a partir da análise do código constitui um desafio para o ensino de programação, principalmente quando as turmas são extensas e a quantidade de turmas por professor também - realidade comum em cursos de programação [De Oliveira et al 2015]. Tal realidade acaba prejudicando, senão inviabilizando, a capacidade de avaliação do professor, dado o grande volume de trabalhos a serem corrigidos e as restrições de tempo para correção e entrega de notas [Nunes 2004], além de ser uma atividade repetitiva, trabalhosa e que pouco acrescenta ao docente. Por causar uma sobrecarga de atividades ao professor, tais fatores tendem a afetar a qualidade das avaliações.

Diante dessa dificuldade, surgiram alternativas para otimização desse processo por meio da automatização e execução de testes de código-fonte [Hollingsworth 1960, Ebrahimi 1994]. A tentativa de substituir a análise visual e execução manual de programas trouxe à luz ferramentas de correção assistida de trabalhos de programação. Contudo, embora tenha ocorrido uma considerável evolução desses sistemas [Romli et al. 2010], de acordo com De Oliveira et al. (2015), ainda há deficiências consideráveis com relação à efetividade dessas ferramentas, principalmente no que tange em avaliar se os objetivos educacionais almejados foram de fato alcançados.

Este artigo apresenta uma ferramenta de análise e avaliação automática de código-fonte como alternativa para a otimização do processo de ensino-aprendizagem de programação. A intenção é incentivar a discussão na comunidade educacional em Computação e propor uma solução no sentido de encontrar novas abordagens para esse desafio, contribuindo para o progresso do ensino de programação.

O texto está organizado como segue: Na Seção 2, apresentamos os trabalhos relacionados. Na Seção 3, apresentamos o CodeTeacher e descrevemos os tipos de análise que o compõem. Na Seção 4, explicamos funcionamento da ferramenta. A Seção 5 traz um estudo de caso para averiguar a viabilidade da ferramenta. E por fim, na Seção 6, concluímos com as considerações finais e trabalhos futuros.

2 | TRABALHOS RELACIONADOS

Dentre as muitas ferramentas de apoio à prática da programação com as finalidades de submissão, execução e avaliação de exercícios, destacam-se o ProgTest, o PCodigo, o BOCA e o MOJO.

O ProgTest [De Souza et al. 2011] é um sistema de apoio automatizado à avaliação de submissões de programas escritos em Java. Juntamente com os programas também são submetidos seus respectivos casos de testes. O ProgTest compila os programas do aluno e submete-os à execução dos testes.

O PCodigo é um complemento do Ambiente Virtual de Aprendizagem (AVA) Moodle [Moodle 2011] para execução em massa e de análise de programas [De Oliveira et al 2015], desenvolvido na linguagem C, porém é aplicável a diferentes linguagens de programação. Integrado ao Moodle, recebe soluções de atividades de programação submetidas por alunos, executa-as e emite relatórios de avaliação para professores.

O BOCA On-line Contest Administrator [Campos and Ferreira 2004, França et al. 2011] é um sistema de internet para submissão de exercícios e correção on-line de código. É a plataforma atual usada nas competições de programação promovidas pela Sociedade Brasileira de Computação (SBC).

MOJO [Chaves 2013] é uma ferramenta que integra o conceito os Juizes On-line (JO) [Kurnia et al. 2001] ao Moodle. Consiste em um módulo que visa auxiliar o professor no processo de elaboração, submissão e correção de questões de programação.

O diferencial do CodeTeacher em relação às soluções já existentes, além de ser focado em Java, é que ele consiste em dar mais flexibilidade à avaliação do professor, permitindo uma análise mais holística. Além de orientar a avaliação por uma perspectiva pedagógica [Ihantola et al. 2010]. Também possui a característica de ser extensível, isto é, sua arquitetura modular favorece a inclusão de novos recursos.

Conseqüentemente, com a possibilidade de proporcionar feedback imediato para o aluno, a ferramenta tem o potencial de obter maior engajamento deste, pois a conscientização do aluno sobre sua situação tende a causar uma reação no sentido de obter melhores resultados. Além de possibilitar maior transparência, tornando visual o processo de avaliação, expondo os resultados e reportando todos os interessados no aprendizado.

Para o professor, há ainda a possibilidade de percepção das deficiências de uma turma através da identificação de erros recorrentes em uma mesma avaliação, bem como a oportunidade de um acompanhamento mais próximo do progresso da disciplina, com maior clareza sobre o rendimento individual e coletivo da turma.

3 | CODETEACHER

O CodeTeacher é uma aplicação desktop feita em Java, portanto independente de plataforma, que se fundamenta na abordagem de análise estática, dinâmica e estático-dinâmica [De Oliveira2015]. É um sistema de análise de código-fonte baseada na configuração de critérios pré-definidos de avaliação. A aplicação foi feita para aceitar projetos em Java, independente da IDE de desenvolvimento utilizada, pois só precisa que as classes compiladas (arquivos com a extensão .class) sejam submetidas à apreciação da ferramenta.

Para detecção automática de inconformidades no código a partir da definição de critérios avaliativos pré-configurados pelo professor, é utilizado o recurso da programação reflexiva ou metaprogramação [Horstmann 2000]. Esse paradigma provê a capacidade de examinar a estrutura e o estado (metainformações) de um programa e o poder de alterar seu comportamento em tempo de execução. Essa funcionalidade é provida por algumas linguagens de programação, dentre elas, o Java.

O foco atual de utilização do CodeTeacher concentra-se nas disciplinas iniciais e intermediárias de programação. É importante salientar também que a ferramenta abrange somente trabalhos práticos de codificação em Java, não abraçando análise textual de respostas a exercícios e questões subjetivas.

Quatro tipos de avaliação são possíveis, denominadas conforme segue: análise estrutural, análise comportamental, análise de saída padrão e análise conceitual. A seguir serão descritos os tipos de análise do CodeTeacher.

3.1 Análise conceitual

Neste tipo de análise, são verificados os elementos estáticos do código, como declaração correta dos atributos e métodos de uma classe, podendo também ser verificados os seus modificadores de acesso, bem como se determinado membro é de classe ou instância.

Outra possibilidade é a análise do uso de herança por meio da navegação pela hierarquia de classes implementada pelo aluno. De modo similar, esta análise permite verificar se determinada classe implementa ou não uma dada interface. É possível ainda flexibilizar a análise considerando apenas alguns elementos na avaliação. Por exemplo, pode-se configurar um critério que avalie se há, no escopo definido, algum método que retorne dados e/ou receba parâmetros de determinados tipos, não sendo necessário informar o nome do método a ser buscado.

Para tornar mais livre a implementação do aluno, sem contudo, prejudicar a efetividade da avaliação, é possível lançar mão de expressões regulares, usando coringas na especificação de critérios. Pode-se, por exemplo, definir que deve existir um método cujo nome inicie com um prefixo tal como “cadastrar...”. Para isso, seria preciso apenas informar no nome do método a seguinte expressão: “cadastrar*”, onde o asterisco (*) significaria qualquer cadeia de caracteres. Similarmente, seria possível verificar a existência de um método com o sufixo “?Dados”, sendo o sinal de interrogação correspondente a qualquer caractere.

3.2 Análise Comportamental

Para Ala-Mutka (2005) e Rahman et al. (2008), corretude e funcionalidade são importantes itens de avaliação. A análise comportamental consiste em atestar a

corretude do código a partir de sua testagem funcional com vistas a simular o comportamento do programa em um ambiente ou cenário real. É uma análise baseada em entrada/saída que testa os serviços providos por um objeto, isto é, verifica-se a saída correta de um programa a partir de entradas fornecidas previamente e compara-se com resultados previstos. O código é executado e são cheçadas as respostas do objeto a estímulos externos, para isso, mensagens são passadas a um objeto no intuito de encontrar uma resposta esperada. O professor modela um conjunto de casos a serem avaliados e submete-os à apreciação da ferramenta e, através da combinação das entradas fornecidas e saídas previstas, é possível inferir a qualidade do programa quanto às suas funcionalidades, sendo possível com isso afirmar se ele corresponde aos requisitos especificados pelo professor.

Um exemplo de uso dessa abordagem pode ser a checagem do retorno de métodos a partir da invocação com passagem de uma lista parâmetros pré-estabelecidos e da definição dos respectivos retornos esperados

3.3 Análise de saída padrão

Nesta análise verifica-se se o código executado realiza a impressão de algum texto na saída padrão do sistema (console). É comum em disciplinas iniciais de programação a criação de programas que escrevam dados ou mensagens na saída padrão. Geralmente são dados resultantes de cálculos efetuados pela aplicação ou mesmo mensagens informativas. Normalmente essas informações são apresentadas de forma textual em uma interface de linha de comandos. A ferramenta captura a saída padrão, interrompendo o fluxo de impressão e desviando-o para um proxy que armazena o conteúdo impresso e, em seguida, verifica se o que é impresso pelo programa do aluno é equivalente ao texto definido no critério de avaliação configurado pelo professor. Este deve informar o conteúdo a ser impresso para que a comparação de igualdade seja realizada e, nesse caso, é considerado sucesso a correspondência exata entre os termos comparados.

3.4 Análise conceitual

Neste tipo de análise podem ser definidos padrões e métricas a serem contemplados. Esse mecanismo permite avaliar a aplicação de conceitos da Programação Orientada a Objetos (POO) como emprego de herança, polimorfismo, grau de encapsulamento, entre outros [Horstmann 2000]. Dessa forma, é possível descobrir se há classes abstratas que não são estendidas ou interfaces não implementadas, por exemplo. O uso de polimorfismo pode ser identificado e avaliado considerando, por exemplo, a presença de métodos sobrecarregados e/ou sobrescritos. Tais fatores podem ser enquadrados em uma escala de gradação definida pelo professor. Como exemplo, pode-se medir

o nível de encapsulamento configurando um percentual mínimo de membros encapsulados a ser alcançado.

Embora ainda não estejam disponíveis no CodeTeacher, outros conceitos e métricas podem ser usados neste tipo de análise, como grau de coesão e acoplamento, pois novas métricas e conceitos podem ser agregados à ferramenta na forma de adição de funcionalidade, com a inclusão de plug-ins de extensão.

4 | ETAPAS DA AVALIAÇÃO

A seguir são descritos os passos para utilização da ferramenta, todos os pormenores envolvendo o processo total de avaliação são detalhados nos tópicos a seguir. O fluxo de atividades é mostrado na Figura 1.



Figura 1. Etapas da avaliação

4.5 Seleção de artefatos

Nesta etapa são indicados quais serão os elementos escolhidos para serem avaliados. Os ativos são selecionados explicitamente pelo usuário, primeiramente, através da indicação de quais diretórios devem ser acessados para buscar os artefatos de código a serem analisados, podendo ser, projetos, classes, ou pacotes de classes. Essa indicação determina o escopo da avaliação. A convenção utilizada pelo *CodeTeacher* para associar um determinado conjunto de artefatos de código a um aluno específico é que todos os arquivos contidos em uma pasta pertencem a um mesmo aluno. Por isso, recomenda-se que haja uma pasta com o nome de cada estudante, pois o nome da pasta servirá de referência para identificar o aluno.

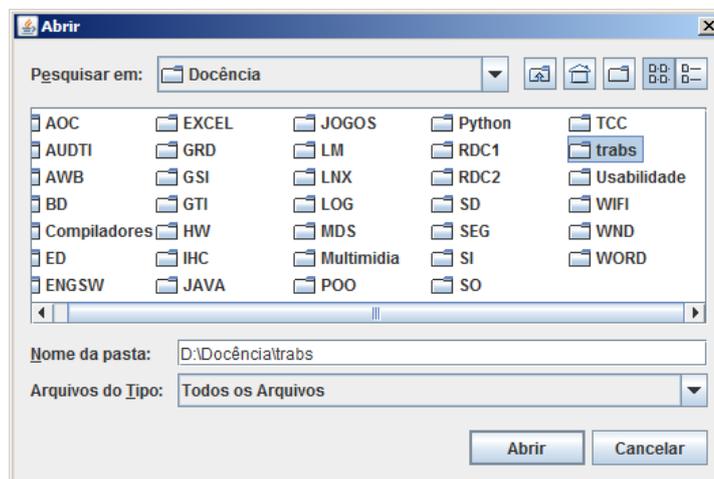


Figura 2. Seleção de artefatos

4.6 Configuração de critérios

Uma vez que os elementos de *software* foram escolhidos, prossegue-se com a criação dos itens que irão compor a avaliação. Assim, um conjunto de critérios deve ser informado pelo professor para serem aplicados durante a análise. Os critérios são definidos usando-se a interface gráfica, onde são indicados quais serão os itens avaliativos considerados. O formato dos critérios varia de acordo com o tipo de análise, porém cada critério deve possuir um valor para compor a nota do aluno. A valoração dos critérios é estabelecida conforme são atribuídos pesos de acordo com o grau de importância considerado pelo avaliador. É neste momento que o professor alimenta o sistema com seu julgamento prévio das competências esperadas do aluno com a realização da atividade prática em questão, no sentido de alcançar os objetivos de aprendizagem estabelecidos.

Depois de criados os critérios, há a possibilidade salvá-los em um arquivo para uso ou edição posterior, evitando assim que o trabalho de configurar todos os critérios seja repetido no caso de uma avaliação posterior.

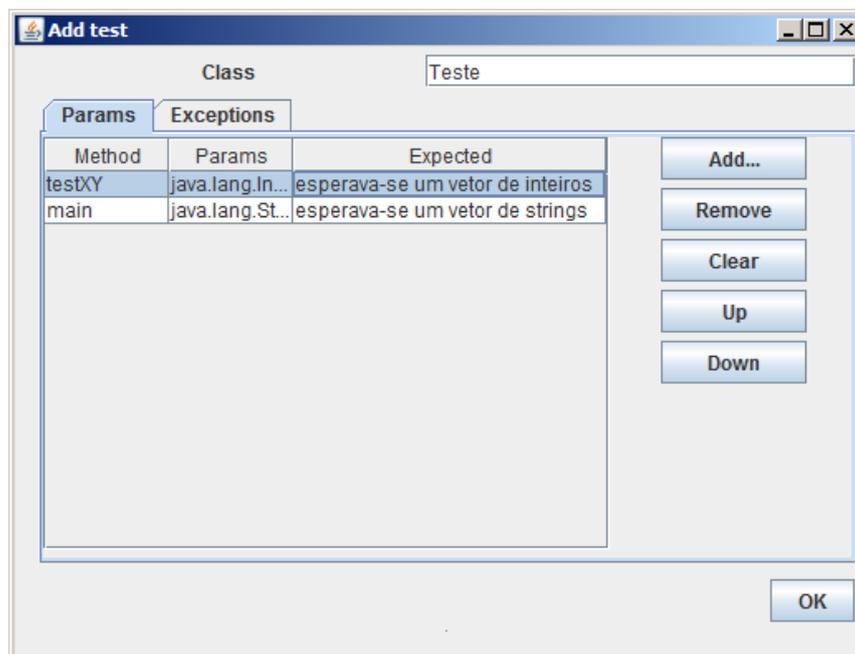
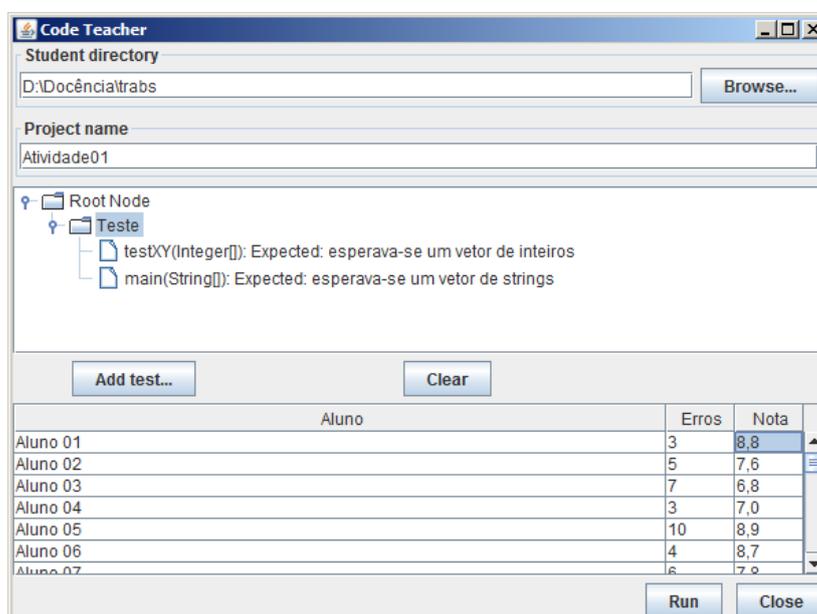


Figura 3. Configuração de critérios

4.7 Execução de testes

A estratégia de execução é definida pelo professor, no sentido de tornar flexível o processo de avaliação. A execução dos testes pode ser em massa ou classe a classe, podendo ser alterada posteriormente, de forma que seja possível aplicar avaliações diferenciadas e com foco em aspectos determinados, a critério do professor. As notas dos alunos são calculadas de acordo com a quantidade total de pontos, obtida através do somatório de todos os critérios. Considera-se que, antes do início da execução, cada aluno possua todos os pontos, porém à medida que vão sendo encontradas irregularidades no seu código, os pontos correspondentes aos critérios não atendidos são debitados do total de pontos do aluno. Por fim, a nota final é contabilizada baseada no percentual de acertos do aluno.



4.8 Relatório de desempenho

Após o término da execução de todos os testes, avança-se para a fase de predição das notas. Os resultados são tabelados e é gerado um relatório de desempenho onde constam as notas dos alunos da turma. Inicialmente o relatório é apresentado na sua forma resumida, contendo um sumário das notas, com informações estatísticas como média da turma e desvio-padrão. Porém, é possível analisar o desempenho individual obtendo um detalhamento do resultado de cada aluno. Outras opções de visualização também são disponibilizadas, como por exemplo, um extrato dos tipos de erros mais cometidos, percentual de acerto, índice de aproveitamento, entre outros. Com essas informações complementares é possível identificar dificuldades generalizadas na turma e planejar estratégias para sanar tais deficiências.

Nesta fase, há ainda uma opção de exportação dos dados resultantes para uma planilha, caso seja necessário manipular tais dados para geração de gráficos, por exemplo, ou mesmo para reportar os alunos acerca de seu desempenho. A geração do arquivo também serve de evidência documental para indicar a evolução da turma no decorrer da disciplina ministrada.

```

Tipo                |Valor| Descrição
Método não encontrado | 5 | Método testXY(Integer[]) não encontrado na classe Test
Método não encontrado | 5 | Método main(String[]) não encontrado na classe Test
Total Erros: 10
Percentual Acertos: 80
Nota: 8.0

```

Figura 5. Relatório de Desempenho

5 | ESTUDO DE CASO

Para atestar a efetividade do CodeTeacher como objeto de aprendizagem, ele foi adotado experimentalmente em uma instituição de ensino real, onde o conjunto de testes foi composto de respostas de exercícios de programação de uma turma regular do curso técnico em informática. Foram selecionadas 5 atividades práticas de programação para compor o corpus do experimento. Para cada atividade foram coletadas respostas em formato de código-fonte, que foram utilizadas como a base de soluções, totalizando 100 respostas.

Para a experimentação da solução proposta, as respostas foram submetidas às análises estrutural, comportamental, de saída padrão e conceitual, respectivamente, sendo a nota final a média aritmética entre estas notas. Para cada análise foram elaborados critérios avaliativos como sendo respostas-modelo, que levavam em consideração aspectos tidos como relevantes em um aluno iniciante em programação, envolvendo o uso básico dos principais elementos do Java, como sintaxe da

linguagem, aspectos procedurais e recursos básicos de orientação a objetos. Os valores dos critérios foram atribuídos individualmente e de forma arbitrária por dois professores. Caso houvesse divergência nas atribuições de valor de cada professor, estas eram confrontadas e passavam por um processo de revisão onde os dois discutiam até chegarem a um consenso.

Posteriormente, as respostas foram avaliadas manualmente por dois professores, que atribuíram notas de 0 a 10. Cada professor recebeu um checklist com os itens avaliativos idênticos aos critérios definidos na aplicação. Os professores deliberadamente assinalaram cada item como “Atendido” ou “Não atendido” e o cálculo da nota foi obtido a partir do percentual de itens atendidos. Este conjunto de exercícios já avaliados compõe a base de comparação do experimento.

5.9 Resultados

Foi feita uma análise comparativa entre as notas obtidas via *CodeTeacher* e as notas atribuídas pelos professores. A Figura 2 ilustra essa comparação.

Questão	#1	#2	#3	#4	#5
Grau de Similaridade (%)	90,69	98,45	85,35	99,89	93,36
Média	93,55				
Análise Comportamental					
Questão	#1	#2	#3	#4	#5
Grau de Similaridade (%)	91,87	95,34	89,32	91,46	88,87
Média	91,37				
Análise de Saída Padrão					
Questão	#1	#2	#3	#4	#5
Grau de Similaridade (%)	90,21	92,35	96,89	96,33	97,99
Média	94,75				
Análise Conceitual					
Questão	#1	#2	#3	#4	#5
Grau de Similaridade (%)	90,69	85,81	75,35	79,89	73,26
Média	81				
Média Geral (%)	90,17				

Figura 2. Análise comparativa

Comparando as notas previstas e as notas reais, a correção automática obteve precisão de 90,17 %, o que demonstra a viabilidade do uso da ferramenta com

ÍNDICE REMISSIVO

A

Abaqus 40, 41, 43, 46, 48

Agrupamento 81, 140, 169, 170, 175, 176

Aplicação 5, 6, 28, 31, 33, 34, 37, 49, 50, 51, 52, 53, 58, 59, 60, 61, 68, 70, 80, 83, 89, 129, 135, 148, 150, 152, 157, 169, 174

Aplicativo móvel 1, 7, 88

Aprendizado online 98, 100

Aprendizagem 63, 64, 65, 66, 67, 69, 70, 71, 72, 73, 74, 75, 87, 88, 89, 90, 92, 93, 94, 95, 96, 97, 100, 101, 140, 147, 148, 149, 150, 154, 156

Aprendizagem colaborativa 89, 95, 140

Área de convergência 49, 50, 51, 52, 58, 60, 61, 62

Atenção visual 98, 101, 103, 106, 107, 110

Avaliação automática 148, 149

B

Banco de dados 8, 59, 66, 76, 77, 79, 81, 82, 84, 85, 86, 170, 182

Busca visual 98, 99, 100, 101, 102, 107, 110

C

Carteira estudantil 1

Celular 87, 88, 90, 91, 96, 179

Centrais de dados 49, 50, 54, 62

Coefficiente de impacto 125, 130, 138

Condicionamento de Ar 16

Custos 30, 31, 32, 36, 37

D

Design constructal 16, 18, 19, 28

Desvio funcional 76

Difuso 169

E

Efeitos dinâmicos 125, 126, 127, 130, 136, 137

Energias renováveis 16

Ensino 2, 34, 38, 63, 64, 74, 87, 88, 90, 93, 94, 95, 96, 97, 148, 149, 156

Espectroscopia 112, 114, 115, 116, 118, 121, 122

Etilômetro 112, 113, 114, 116

Exercícios de programação 148, 156

Expressão gênica 169, 171, 174

G

Gestão pública 76, 85

I

Infravermelho 112, 114, 115, 116, 117, 118, 121

J

Java 53, 148, 149, 150, 151, 156

L

LIWC 140, 143, 144, 146

M

Materiais compósitos 40, 41, 42, 43, 44, 46, 47

Mineração de textos 158, 166, 167

Mobile learning 87, 88, 89, 91, 93, 95, 96, 97

Modelagem computacional 16, 40, 42, 125, 138

Modelo de estilo de aprendizagem 63, 65, 69

Modelos de cargas móveis 125

Multivariado 169, 172, 176, 178

P

Pontes rodoviárias 125, 126, 138, 139

Processamento de linguagem natural 140, 159

Processo ensino-aprendizagem 63

R

Recomendação 63, 64, 67, 70, 71, 72, 73

Recurso educacional aberto 63, 64

Recursos humanos 76, 79, 86

Redes 7, 30, 31, 32, 33, 35, 37, 38, 39, 81, 97, 158, 161, 167

Redes neurais sem peso 158

Robótica 98, 99, 100, 101, 110

S

Simulação 5, 13, 14, 40, 41, 43, 49, 50, 52, 54, 57, 58, 60, 61, 62, 68, 102

Simulação numérica 40, 41, 43

Sistemas de informação 76, 77, 79, 80, 86, 182

T

Tecnologia 1, 30, 32, 33, 34, 38, 39, 63, 90, 97, 112, 176, 182

Telefonia 30, 31, 32, 33, 34, 35, 37, 38, 39

Transporte público 1, 2, 3, 5, 6, 7, 15

Trocadores de Calor Solo-Ar (TCSA) 16, 17

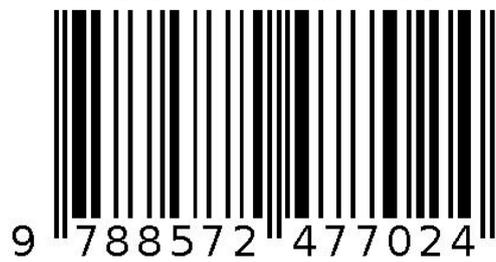
V

VoIP 30, 32, 33, 34, 38, 39

W

Whatsapp 87, 91, 96, 97

Agência Brasileira do ISBN
ISBN 978-85-7247-702-4



9 788572 477024