

Princípios e Aplicações da Computação no Brasil

Ernane Rosa Martins
(Organizador)



Atena
Editora
Ano 2019

Ernane Rosa Martins

(Organizador)

Princípios e Aplicações da Computação no Brasil

Atena Editora

2019

2019 by Atena Editora

Copyright © da Atena Editora

Editora Chefe: Profª Drª Antonella Carvalho de Oliveira

Diagramação e Edição de Arte: Geraldo Alves e Natália Sandrini

Revisão: Os autores

Conselho Editorial

- Prof. Dr. Alan Mario Zuffo – Universidade Federal de Mato Grosso do Sul
Prof. Dr. Álvaro Augusto de Borba Barreto – Universidade Federal de Pelotas
Prof. Dr. Antonio Carlos Frasson – Universidade Tecnológica Federal do Paraná
Prof. Dr. Antonio Isidro-Filho – Universidade de Brasília
Profª Drª Cristina Gaio – Universidade de Lisboa
Prof. Dr. Constantino Ribeiro de Oliveira Junior – Universidade Estadual de Ponta Grossa
Profª Drª Daiane Garabeli Trojan – Universidade Norte do Paraná
Prof. Dr. Darllan Collins da Cunha e Silva – Universidade Estadual Paulista
Profª Drª Deusilene Souza Vieira Dall’Acqua – Universidade Federal de Rondônia
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná
Prof. Dr. Fábio Steiner – Universidade Estadual de Mato Grosso do Sul
Prof. Dr. Gianfábio Pimentel Franco – Universidade Federal de Santa Maria
Prof. Dr. Gilmei Fleck – Universidade Estadual do Oeste do Paraná
Profª Drª Girlene Santos de Souza – Universidade Federal do Recôncavo da Bahia
Profª Drª Ivone Goulart Lopes – Istituto Internazionele delle Figlie de Maria Ausiliatrice
Profª Drª Juliane Sant’Ana Bento – Universidade Federal do Rio Grande do Sul
Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense
Prof. Dr. Jorge González Aguilera – Universidade Federal de Mato Grosso do Sul
Profª Drª Lina Maria Gonçalves – Universidade Federal do Tocantins
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte
Profª Drª Paola Andressa Scortegagna – Universidade Estadual de Ponta Grossa
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. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista
Prof. Dr. Urandi João Rodrigues Junior – Universidade Federal do Oeste do Pará
Prof. Dr. Valdemar Antonio Paffaro Junior – Universidade Federal de Alfenas
Profª Drª Vanessa Bordin Viera – Universidade Federal de Campina Grande
Profª Drª Vanessa Lima Gonçalves – Universidade Estadual de Ponta Grossa
Prof. Dr. Willian Douglas Guilherme – Universidade Federal do Tocantins

Dados Internacionais de Catalogação na Publicação (CIP) (eDOC BRASIL, Belo Horizonte/MG)

P957 Princípios e aplicações da computação no brasil [recurso eletrônico] /
Organizador Ernane Rosa Martins. – Ponta Grossa (PR): Atena
Editora, 2019. – (Princípios e aplicações da computação no
Brasil; v. 1)

Formato: PDF

Requisito de sistema: Adobe Acrobat Reader

Modo de acesso: World Wide Web

Inclui bibliografia

ISBN 978-85-7247-046-9

DOI 10.22533/at.ed.469191601

1. Computação. 2. Informática. 3. Redes sociais. I. Martins,
Ernane Rosa. II. Título. III. Série.

CDD 004

Elaborado por Maurício Amormino Júnior – CRB6/2422

O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de
responsabilidade exclusiva dos autores.

2019

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.

www.atenaeditora.com.br

APRESENTAÇÃO

Esta obra se propõe a permitir conhecer melhor o panorama atual da computação no Brasil por meio dos textos dos 15 capítulos que a constituem. Assim, estes trazem a reflexão temas importantes da área, tais como: performance web de e-commerce, análise de redes sociais, teoria de redes complexas, automação de teste em sistemas legados, ambiente virtual, arquitetura e organização de computadores, sistema integrado de gestão, sistema de apoio à avaliação de atividades de programação, rastreamento de objetos em vídeo, segurança da informação, ensino de programação, ensino de teoria da computação, sistemas de informação, fábrica de software, interdisciplinaridade, estilos de aprendizagem em computação, plataformas multiprocessadoras baseadas em barramentos.

Deste modo, esta obra reúne debates e análises acerca de questões relevantes, tais como: Qual o tamanho médio das páginas das lojas virtuais brasileiras e como estão em comparação com a média mundial? Quais informações estratégicas, para a segurança pública, podem ser obtidas com o uso da análise das redes sociais e complexas provenientes de uma base de dados de Tatuagens em Criminosos? A proposta de um novo ambiente virtual de simulação pode apoiar a aprendizagem? A proposta de um sistema de reconhecimento automático de possíveis soluções com mapeamento destas em escores atribuídos por professores, pode auxiliar professores na avaliação de exercícios de programação? A proposta de uma metodologia para rastreamento de múltiplos objetos em vídeos usando subtração de plano de fundo via mistura de gaussianas, morfologia matemática e o filtro de Kalman é mais precisa do que quando feita usando somente a subtração de plano de fundo? Como mensurar e priorizar a segurança da informação corporativa com base nos atuais arcabouços existentes na área? Quais páginas mais se preocupam com o usuário? Algumas ferramentas que foram propostas em trabalhos anteriores e que são utilizadas no ensino de programação atendem a nova realidade do ensino inicial de programação para crianças e jovens? Um projeto de extensão de uma Fábrica de Software, pode propiciar aos alunos capacitação nas principais tecnologias de mercado e vivência no mundo do trabalho?

Nesse sentido, este material ganha importância por constituir-se numa coletânea de trabalhos, experimentos e vivências de seus autores, tendo por objetivo reunir e socializar os estudos desenvolvidos em grandes universidades brasileiras. Certamente os trabalhos apresentados nesta obra são de grande relevância para o meio acadêmico, proporcionando ao leitor textos científicos que permitem análises e discussões sobre assuntos pertinentes à computação, por meio de linguagem clara e concisa, propiciando a aproximação e o entendimento sobre temas desta área do conhecimento. A cada autor, nossos agradecimentos a submissão de seus estudos na Editora Atena. Aos leitores, desejo proveitosa reflexão sobre as temáticas abordadas.

SUMÁRIO

CAPÍTULO 1 1

UTILIZANDO O TIPI PARA IDENTIFICAR TRAÇOS DE PERSONALIDADE DE ESTUDANTES DE UM CURSO TÉCNICO EM INFORMÁTICA

Janderson Jason Barbosa Aguiar
Joseana Macêdo Fechine Régis de Araújo
Evandro de Barros Costa

DOI 10.22533/at.ed.4691916011

CAPÍTULO 2 13

UMA AVALIAÇÃO DA PERFORMANCE WEB DE E-COMMERCE NO BRASIL

Cristiano Politowski
Gabriel Freytag
Vinícius Maran
Lisandra Fontoura

DOI 10.22533/at.ed.4691916012

CAPÍTULO 3 25

UMA ANÁLISE DOS PADRÕES DE TATUAGENS ASSOCIADOS À CRIMINALIDADE DO ESTADO DA BAHIA COM AUXÍLIO DA TEORIA DE REDES

Hernane Borges de Barros Pereira
Antônio José Assunção Cordeiro
Carlos César Ribeiro Santos
Alden José Lázaro da Silva

DOI 10.22533/at.ed.4691916013

CAPÍTULO 4 32

UM ESTUDO DE CASO DE AUTOMAÇÃO DE TESTE EM SISTEMAS LEGADOS SOBRE PLATAFORMA FLEX

Augusto Boehme Tepedino Martins
Jean Carlo Rossa Hauck

DOI 10.22533/at.ed.4691916014

CAPÍTULO 5 45

UM AMBIENTE VIRTUAL APLICADO AO ENSINO E PESQUISA EM ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Guilherme Álvaro Rodrigues Maia Esmeraldo
Edson Barbosa Lisboa

DOI 10.22533/at.ed.4691916015

CAPÍTULO 6 50

SISTEMA INTEGRADO DE GESTÃO ESPORTIVA: UMA FERRAMENTA DE APOIO AO PROGRAMA TALENTO OLÍMPICO DO PARANÁ

Robson Parmezan Bonidia
Luiz Antonio Lima Rodrigues
Rosângela Marques Busto
Jacques Duílio Brancher

DOI 10.22533/at.ed.4691916016

CAPÍTULO 7 64

SISTEMA DE APOIO À AVALIAÇÃO DE ATIVIDADES DE PROGRAMAÇÃO POR RECONHECIMENTO AUTOMÁTICO DE MODELOS DESOLUÇÕES

Márcia Gonçalves de Oliveira

Leonardo Leal Reblin

Elias Silva de Oliveira

DOI 10.22533/at.ed.4691916017

CAPÍTULO 8 75

RASTREAMENTO DE OBJETOS EM VÍDEO COM APLICAÇÕES PRÁTICAS

Karla Melissa dos Santos Leandro

Sérgio Francisco da Silva

Marcos Napoleão Rabelo

DOI 10.22533/at.ed.4691916018

CAPÍTULO 9 82

PROPOSTA DE ESTRATÉGIA DE MATURIDADE E PRIORIZAÇÃO PARA SEGURANÇA DA INFORMAÇÃO BASEADA NA ISO/IEC 27001 E 27002 ADERENTE AOS PRINCÍPIOS DA GOVERNANÇA ÁGIL

Gliner Dias Alencar

Hermano Perrelli de Moura

DOI 10.22533/at.ed.4691916019

CAPÍTULO 10 99

PROGRAMAÇÃO PARA TODOS: ANÁLISE COMPARATIVA DE FERRAMENTAS UTILIZADAS NO ENSINO DE PROGRAMAÇÃO

Silvino Marques da Silva Junior

Sônia Virginia Alves França

DOI 10.22533/at.ed.46919160110

CAPÍTULO 11 110

MODOS CONTEMPORÂNEOS DE APRENDIZADO E CONSTRUÇÃO DO CONHECIMENTO: REFLEXÕES SOBRE O ENSINO DE TEORIA DA COMPUTAÇÃO PARA SISTEMAS DE INFORMAÇÃO

Isabel Cafezeiro

Leonardo Cruz da Costa

Ricardo Kubrusly

DOI 10.22533/at.ed.46919160111

CAPÍTULO 12 123

MODELO DE FÁBRICA DE SOFTWARE ESCOLA

Edmilson Barbalho Campos Neto

Alba Sandyra Bezerra Lopes

Diego Silveira Costa Nascimento

DOI 10.22533/at.ed.46919160112

CAPÍTULO 13 135

INTERDISCIPLINARIDADE NO IF FARROUPILHA - CAMPUS SANTO ÂNGELO ATRAVÉS DA PRÁTICA PROFISSIONAL INTEGRADA

Fábio Weber Albiero

Karlise Soares Nascimento

Andréa Pereira

Joice Machado

DOI 10.22533/at.ed.46919160113

CAPÍTULO 14..... 140

IDENTIFICAÇÃO DE ESTILOS DE APRENDIZAGEM EM TURMAS DE NÍVEL TÉCNICO, GRADUAÇÃO E PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Janderson Jason Barbosa Aguiar

Joseana Macêdo Fachine Régis de Araújo

Evandro de Barros Costa

DOI 10.22533/at.ed.46919160114

CAPÍTULO 15..... 151

EXPLORAÇÃO EFICIENTE EM ESPAÇOS DE PROJETO DE COMUNICAÇÃO EM PLATAFORMAS MULTIPROCESSADORAS BASEADAS EM BARRAMENTOS

Guilherme Álvaro Rodrigues Maia Esmeraldo

Edna Natividade da Silva Barros

DOI 10.22533/at.ed.46919160115

SOBRE O ORGANIZADOR 167

UTILIZANDO O TIPI PARA IDENTIFICAR TRAÇOS DE PERSONALIDADE DE ESTUDANTES DE UM CURSO TÉCNICO EM INFORMÁTICA

Janderson Jason Barbosa Aguiar

Universidade Federal de Campina Grande
(UFCG),
Campina Grande – PB

Joseana Macêdo Fechine Régis de Araújo

Universidade Federal de Campina Grande
(UFCG),
Campina Grande – PB

Evandro de Barros Costa

Universidade Federal de Alagoas (UFAL),
Maceió – AL

RESUMO: Os ingressantes em cursos técnicos geralmente focam em se profissionalizar e em se encaminhar rapidamente para o mercado de trabalho. Embora possuam objetivos e características similares, tais alunos possuem peculiaridades. A partir do inventário TIPI (*Ten-Item Personality Inventory*), que se baseia no modelo *Big Five* de personalidade, este estudo constatou a heterogeneidade de turmas de informática em nível técnico em relação aos Traços de Personalidade dos alunos. Alfim, comenta-se como a identificação desse aspecto psicológico pode influenciar o desenvolvimento do aluno durante o processo de ensino e aprendizagem.

PALAVRAS-CHAVE: Traços de Personalidade; Modelo *Big Five*; TIPI; Educação em

Computação.

ABSTRACT: The students in technical courses usually focus on becoming professional and going quickly to the labor market. Although they have similar objectives and characteristics, such students have peculiarities. From the TIPI (*Ten-Item Personality Inventory*) — based on the *Big Five* personality model —, this study found heterogeneity of computer classes at the technical level in relation to the Personality Traits of the students. Lastly, we comment on how the identification of this psychological aspect can influence the development of the student during the process of teaching and learning.

KEYWORDS: Learning Styles; *Big Five* Model; TIPI; Computer Education.

1 | INTRODUÇÃO

Ao estudar meios de apoio ao processo de ensino–aprendizagem em cursos de computação, é comum a análise de alunos de graduação e, por vezes, pesquisas considerando o nível técnico de ensino não são geralmente realizadas. Apesar de, em nível de graduação e pós-graduação, os discentes terem mais tempo no curso para aliar o conhecimento teórico e prático, é importante também considerar como

propiciar melhorias na educação a alunos de computação em cursos técnicos.

É corriqueiro observar que alunos em cursos técnicos tendem a escolher este tipo de curso com o objetivo de, em pouco tempo, profissionalizar-se e estar apto a entrar para o mercado de trabalho, ou mesmo incrementar seu currículo e as chances de conseguir oportunidades melhores de emprego. Mesmo os indivíduos possuindo este objetivo específico, tendendo a preferir aspectos práticos (e, no caso da escolha de cursos de computação, geralmente gostarem e terem aptidões para o uso de tecnologias digitais), percebe-se que, ao entrarem nos cursos técnicos, ainda há diferenças particulares entre esses alunos que podem influenciar no seu aprendizado.

Nos últimos anos, muitas pesquisas, considerando essas particularidades, visam empregar aspectos psicológicos humanos na educação mediada por tecnologias de informação e comunicação, uma vez que esses aspectos, a exemplo da personalidade, têm influência no processo de tomada de decisão dos humanos (NUNES, 2012). É interessante indicar, curiosamente, que há sites comerciais que empregam a personalidade para identificar o perfil de interesses do usuário, a exemplo do site de recomendação de filmes Whattorent, cujas estatísticas indicam que os usuários ficaram satisfeitos com as recomendações em mais de 80% do tempo (WHATTORENT, 2015).

Embora a definição do conceito de personalidade não seja um consenso, é possível defini-la como um padrão comportamental consistente originado internamente no indivíduo (BURGER, 2000). Segundo Paixão, Fortaleza e Conte (2012), é preciso criar possibilidades para estimular alunos com diferentes tipos de personalidade nos cursos de computação.

Neste capítulo, é apresentado um relato de experiência sobre a identificação da personalidade de alunos de um Curso Técnico em Informática. A partir disso, são realizadas reflexões para estimular reproduções similares dessa experiência por parte de outros pesquisadores. O restante deste capítulo está organizado como segue: na seção 2, são apresentadas algumas pesquisas relacionadas; na seção 3, é apresentada a teoria de personalidade baseada em traços, além de estratégias de detecção desses traços de personalidade; na seção 4, é apresentado o método empregado para extração dos traços de personalidade de alunos em um curso técnico em informática; na seção 5, são apresentados os resultados da aplicação de um inventário; na seção 6, são discutidos alguns aspectos finais observados a partir dessa experiência.

2 | TRABALHOS RELACIONADOS

Alguns trabalhos analisando a personalidade de indivíduos em computação podem ser encontrados na literatura.

Ferreira e Silva (2008) abordaram como os perfis psicológicos influenciam na utilização e no processo de software. Capretz e Ahmed (2010) realizaram um trabalho em relação aos tipos de personalidade que são mais indicados para os diferentes

papéis em equipes de desenvolvimento de software.

Com enfoque em metodologias ágeis, Branco, Conte e Prikladnicki (2011) estudaram a influência dos tipos de personalidade nessas metodologias de desenvolvimento. Considerando as tarefas individuais e coletivas na Engenharia de Software, Cruz et al. (2011) abordaram a influência da personalidade nessas tarefas.

Paixão, Fortaleza e Conte (2012) realizaram um estudo sobre a personalidade no ensino de computação, discutindo implicações dos perfis psicológicos a partir de uma análise com 46 alunos dos cursos de Ciência da Computação e Sistemas de Informação da Universidade Federal do Amazonas. Posteriormente, com 80 alunos, esses autores realizaram um estudo da relação entre perfil psicológico de alunos e evasão (PAIXÃO, FORTALEZA e CONTE, 2013).

Visando o auxílio na aprendizagem e no desempenho acadêmico, Farias, Dobrões e Da Silva (2013) expuseram uma análise para adaptar atividades da área de computação por meio de adaptações considerando os perfis psicológicos dos alunos.

Apesar desses trabalhos se relacionarem à experiência descrita neste capítulo, não foram encontrados trabalhos cujo foco tenha sido a detecção e análise de traços de personalidade (segundo a teoria do *Big Five*) de alunos de cursos técnicos na área da computação.

3 | TRAÇOS DE PERSONALIDADE

Assim como não há um consenso na psicologia sobre o conceito de personalidade, também não há apenas uma única teoria para representá-la. Dentre as existentes, a teoria dos Traços de Personalidade (ALLPORT e ALLPORT, 1921) é uma das mais populares, especialmente por ser de fácil mensuração por intermédio computacional.

A versão moderna dessa teoria é conhecida como *Big Five* ou modelo dos cinco grandes fatores da personalidade. São estes os fatores/traços do *Big Five*:

- **Extroversão** (*extraversion*): Indivíduos que apresentam altos níveis de extroversão são os considerados ativos, aventureiros, barulhentos, energéticos, entusiásticos, exibidos, sociáveis, tagarelas. Por outro lado, baixos níveis de extroversão indicam um indivíduo geralmente acanhado, introvertido, quieto, reservado, silencioso, tímido.
- **Socialização** (*agreeableness*): Indivíduos que apresentam altos níveis de socialização (amabilidade) são os considerados altruístas, amigáveis, carinhosos, confiantes, cooperativos, gentis, sensíveis, simpáticos. Por outro lado, baixos níveis de socialização indicam um indivíduo geralmente antipático, brigão, bruto, crítico, frio, insensível.
- **Realização** (*conscientiousness*): Indivíduos que apresentam altos níveis de realização (conscientização/conscienciosidade) são os considerados con-

fiáveis, conscientes, eficientes, minuciosos, organizados, práticos, precisos, responsáveis. Por outro lado, baixos níveis de realização indicam um indivíduo geralmente desatento, descuidado, desorganizado, distraído, imprudente, irresponsável.

- **Neuroticismo** (*neuroticism*): Indivíduos que apresentam altos níveis de neuroticismo (instabilidade emocional) são os considerados ansiosos, apreensivos, emotivos, instáveis, nervosos, preocupados, temerosos, tensos. Por outro lado, baixos níveis de neuroticismo indicam um indivíduo geralmente calmo, contido, estável, indiferente, sereno, tranquilo.
- **Abertura** (*openness*): Indivíduos que apresentam altos níveis de abertura (abertura à mudança ou abertura à experiência) são os considerados artísticos, curiosos, engenhosos, espertos, imaginativos, inteligentes, originais, sofisticados. Por outro lado, baixos níveis de abertura indicam um indivíduo geralmente comum, simples, superficial, tolo, trivial, vulgar.

Para realizar pesquisas empíricas da abordagem do *Big Five*, foram criados alguns inventários, sendo um dos principais o NEO-PI-R (*Revised NEO Personality Inventory*), desenvolvido por Costa e McCrae (1992), com 240 questões de múltipla escolha. Por ser um inventário comercial (com direitos autorais), uma alternativa de custo livre ao NEO-PI-R é o NEO-IPIP (*NEO International Personality Item Pool*) com 300 itens (JOHNSON, 2000). Uma versão curta do NEO-IPIP é o TIPI (*Ten-Item Personality Inventory*), com 10 itens (GOSLING, RENTFROW e SWANN, 2003).

No TIPI, cada item possui uma afirmativa, a qual o respondente atribui um valor de concordância do quanto essa afirmativa o representa em uma escala tipo Likert de sete pontos. No Quadro 1 são apresentados os itens desse inventário.

01	Extrovertido, entusiasta.	06	Reservado, quieto.
02	Crítico, briguento.	07	Simpático, acolhedor.
03	Confiável, autodisciplinado.	08	Desorganizado, descuidado.
04	Ansioso, que se chateia facilmente.	09	Calmo, emocionalmente estável.
05	Aberto a novas experiências, complexo.	10	Convencional, sem criatividade.

Quadro 1. Itens do Inventário TIPI.

Apesar de tradicionalmente serem utilizados inventários para extrair os traços de personalidade, é importante citar que há abordagens que não exigem tanto esforço cognitivo por parte dos usuários, como, por exemplo, a utilização de mineração de dados de perfis de usuários em redes sociais.

4 | MÉTODO

Utilizando a teoria explanada na seção 3, foi escolhido o inventário TIPI para

identificar os traços de personalidade de alunos de 3 (três) turmas do Curso Técnico em Informática da Escola Técnica Redentorista de Campina Grande – PB. Uma vez que as turmas tinham número diferenciado de alunos, isso influenciou na quantidade de respostas obtidas. Serão denominadas neste capítulo como Turma A, Turma B e Turma C, sendo a quantidade de respondentes, respectivamente, 33, 8, e 20. Ao todo, portanto, foram 61 alunos que responderam ao inventário.

O inventário TIPI foi aplicado on-line, sendo elaborado e disponibilizado por meio do Google Drive (serviço da empresa Google, disponível por meio deste link: <https://drive.google.com>). Embora seja um instrumento de mensuração simplificado, a escolha do TIPI foi realizada especialmente por ser um inventário não comercial e que demanda pouco tempo do respondente.

Ao concluir a fase de respostas aos questionários, os dados foram avaliados, sendo, com base no TIPI, calculado para cada aluno o grau de intensidade dos traços, que varia de 1 a 7. Esse cálculo é feito ao realizar uma média dos valores definidos para os dois itens relativos a cada traço: os itens 1 e 6 referem-se ao traço Extroversão; os itens 2 e 7 ao traço Socialização; os itens 3 e 8 são relativos ao traço Realização; os itens 4 e 9 ao traço Neuroticismo; e, por fim, os itens 5 e 10 referem-se ao traço Abertura. Vale ressaltar que os itens 2, 4, 6, 8 e 10 possuem as características inversas ao nome caracterizador do traço em questão. Por exemplo: se for definida a avaliação “Concordo fortemente” para o item 1, isso indica que, para o traço Extroversão, se deve considerar o grau máximo (7); entretanto, para o item 2, isso indica que, para o traço Socialização, se deve considerar o grau mínimo (1). Dessa forma, um indivíduo que marcar “Concordo moderadamente” para o item 1 e o valor “Discordo fortemente” para o item 6, possui um grau de Extroversão igual a $(6+7)/2$, ou seja, um grau de **6,5** para o traço Extroversão.

Considerando os resultados possíveis com o TIPI, é possível definir que: (i) para graus de **1 a 3**, o indivíduo possui um nível baixo para aquele traço; (ii) de **5 a 7**, o indivíduo possui um nível alto para aquele traço; e (iii) de **3,5 a 4,5**, pode-se afirmar que tal traço é encontrado moderadamente no indivíduo. Por exemplo, para um aluno cujo resultado seja **2, 3,5, 5,5, 4,5 e 6,5**, respectivamente para os traços Extroversão, Socialização, Realização, Neuroticismo e Abertura, é possível dizer que ele se apresenta como um indivíduo com Extroversão baixa, Socialização e Neuroticismo moderados, e Realização e Abertura altas.

Para comparar por turma o nível de similaridade entre as personalidades dos alunos, considerou-se a formação de um vetor com os valores definidos para os graus relativos aos 5 (cinco) traços de personalidade. Uma vez que cada usuário foi representado por um vetor, utilizou-se a medida de similaridade dos cossenos (Quadro 2) para a comparação entre esses vetores. Quanto mais próximo de 0 (zero) o valor, isso indica a baixa similaridade. Quanto mais o valor seja próximo de 1 (um), mais similares são os vetores, que representam os indivíduos.

$$S_{a,b} = \frac{\sum_{i=1}^n (w_{a,i} * w_{b,i})}{\sqrt{\sum_{i=1}^n (w_{a,i})^2} \times \sqrt{\sum_{i=1}^n (w_{b,i})^2}}$$

Em que:

- $S_{a,b}$ representa o cálculo do cosseno entre o vetor/indivíduo a com um determinado vetor/indivíduo b ;
- $w_{a,i}$ indica o grau do indivíduo a para o traço i ;
- $w_{b,i}$ indica o grau do indivíduo b para o mesmo traço; e
- n indica a quantidade de traços existentes (no caso, $n=5$).

Quadro 2. Equação representativa da Medida dos Cossenos.

Fonte: Adaptado de COSTA, AGUIAR e MAGALHÃES, 2013, p. 59.

5 | RESULTADOS

Com base no exposto na seção 4, foram gerados os gráficos das Figuras 1 a 5 para apresentar os resultados obtidos.

A partir da Figura 1, percebe-se que a maioria apresenta uma Extroversão moderada, mas a quantidade de alunos com Extroversão alta é significativa ao considerar a totalidade de alunos. Entretanto, é interessante verificar que, na Turma B, não há alunos com esse perfil, o que poderia refletir na adoção de estratégias diferentes para uma turma sem características fortes de Extroversão, diferentemente, por exemplo, das Turmas A e C, que apresentam uma diversidade de perfis. É possível pensar que, como a turma B é bem menor, considerando as demais, se houvesse mais alunos, provavelmente se teria perfis mais extrovertidos. Mas isso reforça o fato de verificar os traços de personalidade tanto individualmente quanto por turma, para que o professor busque melhores estratégias para se relacionar e propor atividades a esses discentes.

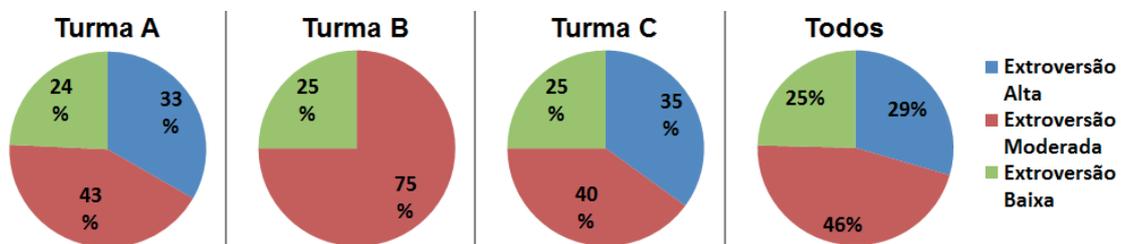


Figura 1. Comparação do resultado da aplicação do questionário TIPI em três turmas de um Curso Técnico em Informática — Traço Extroversão.

Fonte: Elaborado pelos autores

A partir da Figura 2, nota-se que poucos são os alunos que se apresentam com Socialização baixa. Entretanto, a existência de alguns perfis nesse sentido, em uma turma predominantemente com Socialização alta, como as Turmas A e B, pode fazer

com que o docente considere a turma de forma homogênea, podendo dificultar a aprendizagem desses indivíduos diferenciados, ao propor atividades que privilegiem o perfil de Socialização alta.

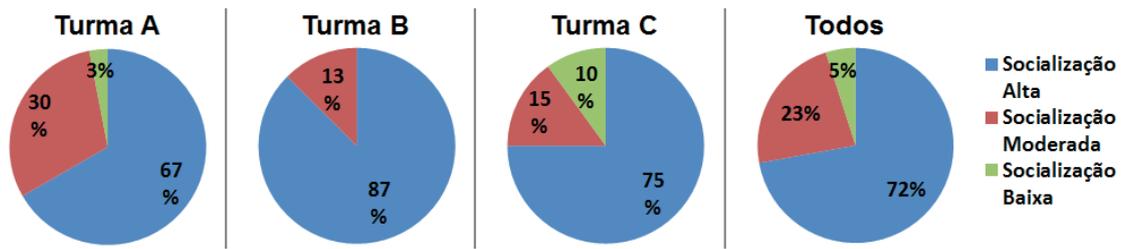


Figura 2. Comparação do resultado da aplicação do questionário TIPI em três turmas de um Curso Técnico em Informática — Traço Socialização.

Fonte: Elaborado pelos autores

Com os gráficos da Figura 3, é indicado que a maioria dos alunos apresenta uma Realização alta, ou pelo menos moderada. Entretanto, não inexistente esse tipo de perfil na turma B, o que pode requerer uma estratégia um pouco diferenciada, específica para tal aluno.

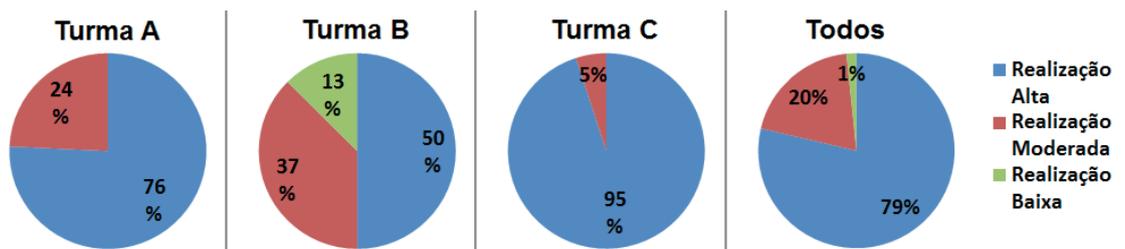


Figura 3. Comparação do resultado da aplicação do questionário TIPI em três turmas de um Curso Técnico em Informática — Traço Realização.

Fonte: Elaborado pelos autores

Considerando a totalidade de respostas, percebe-se, a partir da Figura 4, que a maioria dos alunos apresenta um nível moderado de Neuroticismo. Entretanto, ao analisar por turma, visivelmente percebe-se diferenças por turmas. Na turma C, por exemplo, o docente provavelmente teria que ter um cuidado menor do que em relação à turma B, uma vez que esta turma possui grande parte dos alunos com Neuroticismo alto, ou seja, alta instabilidade emocional.

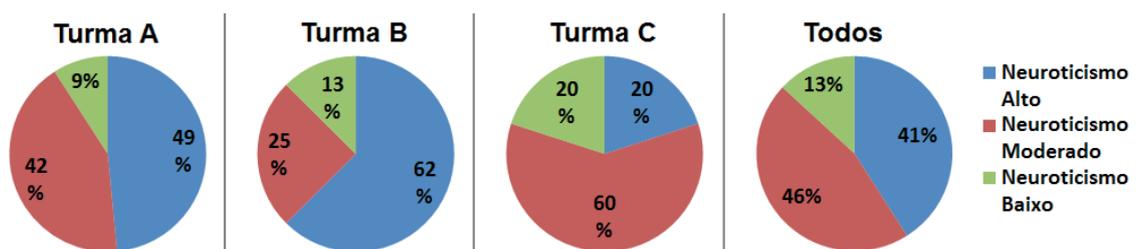


Figura 4. Comparação do resultado da aplicação do questionário TIPI em três turmas de um Curso Técnico em Informática — Traço Neuroticismo.

Fonte: Elaborado pelos autores

Similarmente ao traço Realização, na Figura 5 é apontado que os alunos apresentaram em geral um perfil com Abertura alta ou pelo menos moderada. Mas há uma especificidade em um perfil da turma C. Inovações metodológicas por parte do professor talvez prejudiquem o aprendizado deste aluno com Abertura baixa. Sem a identificação dessa característica por parte do professor, tal aluno poderia vir apresentando bom desempenho e, por mudanças ocorridas, poderia passar a ter um desempenho insuficiente.

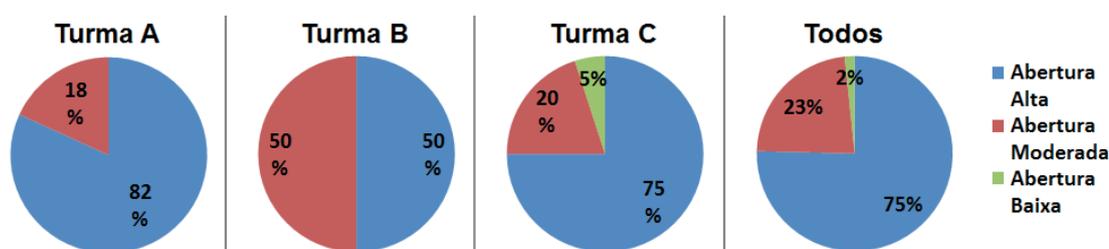


Figura 5. Comparação do resultado da aplicação do questionário TIPI em três turmas de um Curso Técnico em Informática — Traço Abertura.

Fonte: Elaborado pelos autores

Além da geração e análise dos gráficos, foi comparado por turma o nível de similaridade entre as personalidades dos alunos, para identificar as possíveis diferenças, considerando os 5 (cinco) traços de personalidade. Os maiores e menores valores de similaridades obtidos por meio da Medida dos Cossenos (Quadro 2), além das similaridades médias, podem ser vistos no Quadro 3.

	Turma A	Turma B	Turma C	Todos
Maior Similaridade	0,99955	0,99659	0,99900	0,99955
Menor Similaridade	0,71617	0,77981	0,63626	0,58286
Similaridade Média	0,93488	0,93527	0,92319	0,93130

Quadro 3. Similaridades entre alunos de um Curso Técnico em Informática.

Com base nos valores do Quadro 3, é possível observar que os alunos do curso técnico em computação, que participaram desta experiência, são similares, havendo perfis de alunos com personalidade quase 100% similares. Ao verificar as personalidades médias das turmas, também é evidenciado o quão parecidos podem ser os perfis de alunos nesses cursos. Entretanto, se forem analisadas as menores similaridades, apesar de serem valores acima de 50%, é importante considerar que há, na mesma turma, dois ou mais perfis que impossibilitam tratar a turma homogeneamente.

Percebe-se, com a realização desta experiência que, apesar de haver bastante similaridade no perfil dos alunos em um curso técnico em informática, relativo à sua personalidade, é possível encontrar alunos com características peculiares e que, se forem utilizadas estratégias considerando a homogeneidade da turma, estes alunos

tenderão a ser prejudicados.

A análise individual dos traços de personalidade e, por conseguinte, uma visão geral dos traços encontrados nas turmas, pode ajudar a prever que alguns aspectos metodológicos a serem adotados podem ser mais úteis em relação aos outros. Pessoas com características de pouca Extroversão e Socialização podem, por exemplo, obter resultados satisfatórios em provas escritas e/ou trabalhos individuais e, por outro lado, obter resultados insatisfatórios em trabalhos em equipes e/ou apresentações de seminários. Mesmo que a maioria da turma apresente perfis similares, é papel do professor buscar propiciar um adequado aprendizado também àqueles alunos diferenciados.

6 | CONSIDERAÇÕES FINAIS

O conceito de traços de personalidade é relativo à psicologia, mas sua aplicação pode abranger várias áreas, uma vez que a interação entre indivíduos é essencial em diversas tarefas, a exemplo de tarefas de transmissão e aquisição de conhecimento.

Segundo Litto (1996, apud KURI, 2004), deve-se haver uma educação individualizada, sendo inaceitável que se continue com o paradigma industrial de tratar todos os alunos iguais, como em uma linha de montagem.

Há estudos, por exemplo, sobre a correlação de personalidade e estilos de aprendizagem (KURI, 2004; SENRA, 2009; ZONASH e NAQVI, 2011; RAJU e VENUGOPAL, 2014), indicando correlações entre aspectos de personalidade e os estilos preferidos empregados pelos alunos para adquirir conhecimentos. O conceito de estilos de aprendizagem é comumente empregado em pesquisas em geral sobre educação e, por conseguinte, importante para a área de Educação em Computação.

Mesmo que ainda não sejam definidas nitidamente fortes correlações entre traços de personalidade e os estilos de aprendizagem dos alunos, a personalidade pode influenciar como o aprendiz se relaciona com os colegas, professores e até mesmo outros indivíduos envolvidos no processo ensino–aprendizagem. Esses relacionamentos podem impactar em desempenho e permanência dos alunos no curso. Portanto, é interessante que os professores reflitam a possibilidade de começarem a extrair tal informação no início de suas disciplinas, podendo até ser algo planejado e compartilhado com outros professores e coordenações de cursos.

Além de útil para a interação aluno–professor e aluno–aluno em sala de aula, por exemplo, os traços de personalidade podem ser pertinentes para a personalização de ambientes virtuais de aprendizagem, empregados como apoio em muitos cursos da área da computação.

Vale ressaltar que as tecnologias digitais de informação e comunicação podem facilmente ser empregadas para automatizar a aplicação e a geração de resultados dos inventários de personalidade, especialmente em cursos voltados à computação,

nos quais docentes e discentes geralmente possuem familiaridade de uso — além do fato especial da maioria dos docentes possuir conhecimento de linguagens de programação, facilitando ainda mais essa automatização. Por exemplo, os inventários para extrair os traços de personalidade podem ser facilmente elaborados por docentes via Google Drive, no qual é fácil criar, compartilhar, preencher e receber os resultados de formulários, com apenas conhecimentos básicos em informática — e, com conhecimento de programação, o uso do Google Drive pode ser expandido, gerando os resultados automáticos após o término do preenchimento. Além disso, pode-se ainda estudar e aplicar as iniciativas que utilizam mineração de dados de perfis de usuários em redes sociais para detectar sua personalidade.

Em relação a ameaças à validade do estudo, é importante ressaltar que a identificação de traços de personalidade é um tema complexo, voltado à área da psicologia, e que, segundo alguns autores, a aplicação de questionários nem sempre é eficaz, uma vez que os indivíduos podem oferecer respostas que não os representam realmente. Nunes (2012) defende que o uso do inventário TIPI é uma boa opção quando a extração da personalidade não for o tópico primário de interesse na pesquisa.

Todavia, ressalta-se que é importante tratar o aspecto da personalidade na educação em computação, uma vez que, por exemplo, baseando-se nos traços de personalidade dos alunos, podem ser formados grupos com perfis similares em busca de equipes de alto desempenho em engenharia de software.

Por fim, é possível dizer que a realização desta experiência, com alunos de informática em nível técnico, propiciou a visualização de aspectos heterogêneos nas turmas em relação à personalidade, baseando-se no inventário TIPI. Espera-se, com os resultados e discussões apresentados neste capítulo — publicados primeiramente nos Anais do XXIII Workshop sobre Educação em Computação (AGUIAR, FECHINE e COSTA, 2015) —, motivar mais estudos sobre essa temática.

Destaca-se que este trabalho faz parte de um projeto de pesquisa no qual estão sendo estudadas estratégias, considerando os TP dos estudantes, visando à melhoria das recomendações em Sistemas de Recomendação Educacionais (AGUIAR, ARAÚJO e COSTA, 2018; COSTA, AGUIAR E MAGALHÃES, 2013).

7 | AGRADECIMENTOS

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro. À Escola Técnica Redentorista de Campina Grande – PB, especialmente aos alunos do Curso Técnico em Informática que participaram desta pesquisa. À Sociedade Brasileira de Computação (SBC), por dar o direito dos autores publicarem este trabalho inicialmente publicado nos Anais do XXIII Workshop sobre Educação em Computação (WEI 2015).

REFERÊNCIAS

- AGUIAR, J. J. B.; ARAÚJO, J. M. F. R.; COSTA, E. B. **Estratégia baseada em Personalidade e Tendências para Recomendação de Objetos de Aprendizagem usando Algoritmo Genético**. In: Anais do I Workshop Latino-Americano de Trabalhos em Andamento em Computação (WLATAC – CLEI-LACLO 2018), São Paulo, 2018. No prelo.
- AGUIAR, J. J. B.; FECHINE, J. M.; COSTA, E. B. **Identificando os Traços de Personalidade de Estudantes de um Curso Técnico em Informática**. In: Anais do XXXV Congresso da Sociedade Brasileira de Computação (CSBC) – XXIII Workshop sobre Educação em Computação (WEI), Recife, 2015. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2015/036.pdf>>. Acesso em: 20 ago. 2018.
- ALLPORT, F. H.; ALLPORT, G. W. **Personality Traits: Their Classification And Measurement**. *Journal Of Abnormal And Social Psychology*, 16, p. 6–40, 1921.
- BRANCO, D.; CONTE, T.; PRIKLADNICKI, R. **Um estudo preliminar sobre Tipos de Personalidade em Equipes Scrum**. In: ClbSE 2012, Buenos Aires, Argentina, 2012.
- BURGER, J. M. **Personality**. Wadsworth, fifth edition. 2000.
- CAPRETZ, L. F.; AHMED, F. **Why Do We Need Personality Diversity in Software Engineering?**. *ACM SIGSOFT Software Engineering Notes*, v. 35, n. 2, p. 1–11, 2010.
- COSTA, E.; AGUIAR, J.; MAGALHÃES, J. **Sistemas de Recomendação de Recursos Educacionais: conceitos, técnicas e aplicações**. In: II Congresso Brasileiro de Informática na Educação – Jornada de Atualização em Informática na Educação (JAIE), p. 57–78, 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/pie/article/view/2589/2245>>. Acesso em: 25 ago. 2018.
- COSTA, P. T.; MCCRAE, R. R. **Revised neo personality inventory (neo-pi-r) and neo five-factor inventory (neo-ffi)**. Professional manual. 1992.
- CRUZ, S.; DA SILVA, F.; MONTEIRO, C.; SANTOS, P.; ROSSILEI, I. **Personality in Software Engineering: preliminar findings from a systematic literature review**. In: Proceedings of 15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011), p. 1–10, 2011.
- FARIAS, A. B.; DOBRÕES, J. A. L.; DA SILVA, R. Y. F. **Strategies for Teaching Based on Academic Personality Types**. In: XVIII Conferência Internacional sobre Informática na Educação (TISE), Porto Alegre–RS, p. 633–636, 2013. Disponível em: <<http://www.tise.cl/volumen9/TISE2013/633-636.pdf>>. Acesso em: 25 ago. 2018.
- FERREIRA, P. G.; SILVA, F. Q. B. **Fatores humanos que influenciam a utilização de processos de software**. In: Anais do VII Simpósio Brasileiro de Qualidade de Software (SBQS 2008), Florianópolis, Brasil, p. 123–138, 2008.
- GOSLING, S. D.; RENTFROW, P. J.; SWANN, W. B., Jr. **A very brief measure of the big-five personality domains**. *Journal of Research in Personality*. Elsevier, v. 37, p. 504–528, 2003. Disponível em: <<http://gosling.psy.utexas.edu/wp-content/uploads/2014/09/JRP-03-tipi.pdf>>. Acesso em: 25 ago. 2018.
- JOHNSON, J. A. **Web-based personality assessment**. In: 71st Annual Meeting of the Eastern Psychological Association, USA, 2000. Disponível em: <<http://www.personal.psu.edu/faculty/j/5/j5j/papers/ConferencePapers/2000EPA.pdf>>. Acesso em: 25 ago. 2018.
- KURI, N. P. **Tipos de personalidade e estilos de aprendizagem: proposições para o ensino de**

engenharia. 2004. Tese (Doutorado em Engenharia de Produção), Universidade Federal de São Carlos, São Carlos. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/3332>>. Acesso em: 25 ago. 2018.

NUNES, M. A. S. N. **Computação Afetiva personalizando interfaces, interações e recomendações de produtos, serviços e pessoas em Ambientes computacionais**. In: Nunes, M. A. S. N.; Oliveira, A. A.; Ordonez, E. D. M. (Org.). *Projetos e Pesquisas em Ciência da Computação no DCOMP/PROCC/UFS: São Cristóvão*, p. 115–151, 2012. Disponível em: <<http://almanaquesdacomputacao.com.br/gutanunes/publications/NunesDCOMP2012.pdf>>. Acesso em: 25 ago. 2018.

PAIXÃO, C. C.; FORTALEZA, L. L.; CONTE, T. **Um Estudo Preliminar sobre as Implicações de Tipos de Personalidade no Ensino de Computação**. In: *Anais do XXXII Congresso da Sociedade Brasileira de Computação (CSBC) – XX Workshop sobre Educação em Informática (WEI)*, Curitiba–PR, 2012. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2012/0026.pdf>>. Acesso em: 25 ago. 2018.

PAIXÃO, C. C.; FORTALEZA, L. L.; CONTE, T. **Desafios no Ensino de Computação: um estudo da relação entre perfil psicológico de alunos e evasão**. In: *Anais do XXXIII Congresso da Sociedade Brasileira de Computação (CSBC) – XXI Workshop sobre Educação em Informática (WEI)*, Maceió–AL, p. 720–729, 2013. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/0047.pdf>>. Acesso em: 25 ago. 2018.

RAJU, P. G.; VENUGOPAL, M. **Personality & learning styles – lessons for Indian corporate trainers**. *Indian Journal of Industrial Relations*, v. 49, n. 4, 2014.

SENRA, C. M. S. **Os Estilos de Aprendizagem de Felder a partir de Jung**. 2009. Dissertação (Mestrado em Educação Tecnológica), Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG, Belo Horizonte. Disponível em: <<http://www2.et.cefetmg.br/permalink/a2888022-14cd-11df-b95f-00188be4f822.pdf>>. Acesso em: 25 ago. 2018.

WHATTORENT. **What to rent?**. 2015. Disponível em: <<http://www.whattorent.com/top.php>>. Acesso em: 16 jun. 2015.

ZONASH, R.; NAQVI, I. **Personality Traits and Learning Styles among Students of Mathematics, Architecture, and Fine Arts**. *Journal of Behavioral science*, v. 21, p. 92–108, 2011.

UMA AVALIAÇÃO DA PERFORMANCE WEB DE E-COMMERCE NO BRASIL

Cristiano Politowski

Universidade Federal de Santa Maria, Programa de Pós-Graduação em Computação
Santa Maria – Rio Grande do Sul

Gabriel Freytag

Universidade Federal de Santa Maria, Programa de Pós-Graduação em Computação
Santa Maria – Rio Grande do Sul

Vinícius Maran

Universidade Federal de Santa Maria, Laboratory of Ubiquitous, Mobile and Applied Computing (LUMAC)
Cachoeira do Sul – Rio Grande do Sul

Lisandra Fontoura

Universidade Federal de Santa Maria, Programa de Pós-Graduação em Computação
Santa Maria – Rio Grande do Sul

RESUMO: A análise da performance de aplicações web é um tópico recente e pouco explorado no meio empresarial e acadêmico no Brasil. No entanto, a velocidade de acesso às páginas está diretamente relacionada com a experiência do usuário e, assim sendo, reflete diretamente na receita dos e-commerces. Nesta pesquisa é apresentada uma avaliação da performance web das 29 lojas virtuais mais acessadas no Brasil segundo o ranking da Alexa. Por meio do estudo foi possível ter um panorama da performance geral das páginas

web. Há uma larga diferença entre o melhor e o pior colocado. Além disso, os números mostram que as páginas dos e-commerces brasileiros, com poucas exceções, estão piores que os valores considerados adequados por especialistas e/ou médias mundiais.

PALAVRAS-CHAVE: aplicações web, análise de performance, e-commerce.

ABSTRACT: Performance of web applications is an underexplored topic by developers and academic work in Brazil. However, the access speed to pages is directly related to the user experience and, therefore, reflects directly the revenue from E-Commerces. This research presents an evaluation of performance of the 29 most accessed virtual stores in Brazil according to the Alexa ranking. There is a wide difference between the best and the worst placed. Moreover, the numbers show that the pages of Brazilian e-commerces, with few exceptions, are worse than the amounts considered appropriate by experts and/or global averages.

KEYWORDS: Web applications, performance analysis, e-commerce.

1 | INTRODUÇÃO

Comércios eletrônicos, lojas virtuais ou *e-commerces* fazem parte do cotidiano de

consumidores de todo o mundo. Grandes empresas como a *Amazon* e *AliExpress* veem sua receita aumentar a cada ano por meio da venda de grandes quantidades de produtos à preços baixos. Pequenas e médias empresas também têm espaço nesse mercado em constante crescimento. Segundo *eMarketer* [eMarketer 2016], em 2015, o total arrecadado por empresas desse tipo foi de aproximadamente U\$ 1.600.000.000.000 (um trilhão e seiscentos bilhões de dólares americanos) com previsão de aumento de mais de 110% até 2019. Os *e-commerces* no Brasil, apesar de apresentarem números mais modestos, tiveram uma receita em 2015 de cerca de U\$ 19.600.000.000 (dezenove bilhões e seiscentos milhões de dólares americanos) com uma previsão de aumento de cerca de 50% até 2019, ou seja, mais de trinta bilhões¹. Para efeito de comparação, este valor corresponde, hoje, ao Produto Interno Bruto (PIB) do Paraguai².

A competitividade do mercado obrigou donos de lojas virtuais à investir em diferenciais para seus negócios. Como consequência, a complexidade dos *e-commerces*, ou seja, o número de requisições e o tamanho total das páginas, aumentou. Segundo os dados de *HTTPArchive* [HTTPArchive 2016], no mês de março, o tamanho médio de uma página na Internet teve crescimento de 1311 KB em 2013 para 1703 KB em 2014, 2008 KB em 2015 e este ano chegou à 2281 KB, um aumento de mais de 74%.

Contrário ao crescimento das páginas, usuários não aceitam mais atrasos na compra de produtos via *e-commerces* [IPNews 2015]. Estudos realizados pela indústria e livros técnicos de profissionais especializados como Steve Souders, Ilya Grigorik e Patrick Meenan [Souders 2016, Grigorik 2016, Meenan 2016] comprovam esse fato. Segundo um estudo realizado pelo *Aberdeen Group* em 2008 [Aberdeen Group 2008], um segundo de atraso no tempo de resposta de aplicações *web* pode impactar na satisfação do usuário em 16%, nas páginas visualizadas em 11% e nas conversões (ato de compra do usuário) em 7%. De acordo com a empresa *Akamai* [Akamai 2002], a instabilidade e imprevisibilidade da performance de uma aplicação pode impactar negativamente nos clientes, ou seja, na escolha do *e-commerce* na qual realizarão suas compras.

Segundo Everts [Everts 2015], em 2006 usuários de *e-commerces* esperavam que uma página carregasse em 4 segundos ou menos. Atualmente, no entanto, cerca de metade dos usuários esperam um tempo de carregamento de 2 segundos ou menos [Forrester Consulting 2009] e 18% esperam que a página carregue instantaneamente. O mesmo autor ainda cita alguns casos onde a performance influenciou diretamente na receita das empresas:

- Cada 1 segundo retirado do tempo de carregamento da página equivaleu a um aumento de 2% na taxa de conversão de clientes em *Walmart.com*;
- *Staples.com* retirou 1 segundo do tempo de carregamento da página, au-

1 Dados retirados de Statista sendo a pesquisa realizada entre 2014 e 2015 por eMarketer.

2 Dados retirados dos sites: FMI, World Bank e Nações Unidas.

mentando a taxa de conversão em 10%;

- *AutoAnything.com* cortou o tempo de carregamento da página pela metade, aumentando a taxa de conversão em 9%;
- A plataforma de captação de fundos da *Mozilla* retirou 2.2 segundos do tempo de carregamento da página, aumentando o número de downloads em 15,4%;
- *Intuit.com* cortou o tempo de carregamento pela metade, aumentando a conversão em 14%.

Dada esta situação, como podemos otimizar um *website*? Souders [Souders 2007] apresenta a *Performance Golden Rule*, mostrando que é muito mais viável concentrar os esforços de otimização nos elementos “do lado do cliente” (*client-side* ou *front-end*) pois correspondem a 80–90% dos componentes da aplicação:

“Only 10–20% of the end user response time is spent downloading the HTML document. The other 80–90% is spent downloading all the components in the page.” — Steve Sauders.

Com a previsão de crescimento da receita dos *e-commerces* no Brasil, com o crescente aumento do tamanho e do número de requisições das páginas *web* e pelo fato do *front-end* corresponder à grande parte de uma página *web*, duas perguntas centrais devem ser respondidas para se ter um panorama da performance dos *e-commerces* no Brasil:

- 1) Qual o tamanho médio das páginas das lojas virtuais brasileiras? Como estão em comparação com a média mundial? Qual a média de *assets* por página?
- 2) Quais páginas mais se preocupam com o usuário, ou seja, qual o *ranking* das métricas relacionadas a experiência do usuário (*Time To Start Render*, *Speed Index*, etc)? Portanto, o objetivo deste trabalho é capturar e analisar dados de performance *web*

Portanto, o objetivo deste trabalho é capturar e analisar dados de performance *web* das lojas virtuais mais acessadas no Brasil. Um panorama dos principais *e-commerces* é importante e se faz necessário por ser um ponto de partida para uma estratégia de otimização. Além disso, novas ideias e *insights* surgem por meio da análise de performance e podem auxiliar pesquisadores de outras áreas como *User experience (UX)* e *Search Engine Optimization (SEO)*.

O artigo está estruturado da seguinte forma: na Seção 2 são apresentados os conceitos e termos necessários para compreender o trabalho. Na Seção 3 é mostrado o método utilizado na pesquisa. Na Seção 4 os dados obtidos por meio da pesquisa são analisados. Na Seção 5 são apresentados os trabalhos relacionados. Por fim, na Seção 6 são descritas as considerações finais.

2 | REFERENCIAL TEÓRICO

A visualização de uma página *web* é composta de uma requisição para cada *asset* (arquivo). Nos primórdios da internet, o arquivo HTML continha quase a totalidade dos bytes transferidos. Hoje, o cenário mudou, e o tamanho do arquivo HTML representa uma porção muito pequena do total de bytes transferidos.

Durante muitos anos a performance *web* foi — e continua sendo, porém, em menor escala — associada à otimização *backend*. No entanto, de acordo com Souders [Souders 2007], se formos capazes de cortar ao meio o tempo de resposta do *backend*, o tempo de resposta do usuário final seria reduzido apenas 5–10%. Porém, se nós reduzíssemos a performance do *front end* ao meio, o tempo de resposta total seria reduzido de 40–45%. Isso mostra a importância da otimização da performance do *front end*.

2.1. Métricas de Performance em Sistemas Web

Quando se fala em performance web, o primeiro detalhe a considerar são quais dados monitorar, ou seja, por meio de quais métricas a performance será analisada. Para tanto, existem várias métricas com diferentes propósitos que, inclusive, vão evoluindo ou sendo desconsideradas conforme mais estudos são realizados. De acordo com Viscomi [Rick Viscomi 2015], entre as métricas mais usadas encontram-se:

- *TTFB (Time To First Byte* — tempo até o primeiro byte): é o tempo do início da requisição até o navegador receber o primeiro byte da resposta;
- *TTSR (Time to Start Render* — tempo até o início da exibição na tela): é o tempo do início da requisição até o navegador exibir o primeiro pixel do conteúdo na tela;
- *Speed Index* (índice de velocidade): é uma representação do progresso visual de uma tela;
- *Load Time* (tempo de carregamento): é o tempo de carregamento total da página;
- *Requests* (requisições): o número de requisições HTTP antes do evento;
- *Connections* (conexões): é a quantidade de conexões realizada com servidores;
- *Bytes In* (bytes de entrada): é o número total de bytes recebidos no teste;
- *DOM Elements* (número de elementos contidos no DOM): é o número de elementos no documento.

2.2. Performance Percebida

Métricas por si só não garantem que um *website* pareça rápido para o usuário. Hoje, engenheiros de performance estão dando uma maior ênfase na performance percebida do que em números por si só. De acordo com a escala citada por Nielsen

[Nielsen 2009], ações que levam até 0.1 segundos para serem concretizadas passam a impressão de tempo real ao usuário. Entre 0.1 e 1 segundo, o usuário, caso esteja em plena atenção, percebe um atraso mínimo mas mantém o controle da aplicação navegando livremente. Após 1 segundo, o usuário começa a ficar impaciente, o que vai aumentando conforme o tempo passa. 10 segundos marca o tempo de quebra do fluxo da navegação, normalmente fazendo o usuário abandonar o página. Grigorik [Grigorik 2013] vai mais além e define uma “regra de ouro” (*rule of thumb*) para performance *web*: “Renderize suas páginas, ou pelo menos, forneça um feedback visual em menos de 250 milissegundos para manter o usuário engajado.” (tradução livre). Ou seja, a principal preocupação agora é dar uma resposta o mais cedo possível para o usuário, idealmente em até 250 milissegundos.

Como as métricas de rede não conseguem traduzir com precisão a performance percebida pelos usuários durante a navegação em *websites*, novas formas de medição foram criadas. Por exemplo, *Time to Start Render* é a métrica que mede o tempo decorrido da requisição até o momento que o primeiro pixel é mostrado na tela, ou seja, é a primeira indicação percebida pelo usuário de que a página está sendo renderizada, o que não significa que a página está totalmente carregada. *Above-The-Fold Time* é a métrica que mede o tempo entre o clique no *link* de um *website* e o carregamento da parte visível da página, ou seja, a parte inicial do *website* que não necessita de *scroll* para ser visualizada.

Speed Index é uma métrica criada pela ferramenta WebPagetest. De acordo com [Rick Viscomi 2015], o índice de velocidade de uma página é derivado do progresso visual de cada tela, até ser totalmente carregada, sendo que uma página que exhibe mais ao usuário e mais rapidamente tem um índice de velocidade inferior, ou melhor do que uma página que é mais lenta para exibir o conteúdo.

3 | METODOLOGIA DE PESQUISA E AVALIAÇÃO

Inicialmente definiu-se quais seriam os *e-commerces* analisados. Como base, utilizou-se o *ranking* dos 500 *websites* mais acessados no Brasil, segundo o site Alexa³ no mês de março. Desta lista, filtrou-se apenas os *websites* de comércio eletrônico, resultando em uma lista de 30 *e-commerces* com o melhor posicionamento nesse *ranking* (Tabela 1).

3 <http://www.alexa.com>

Loja	URL	Posição	Loja	URL	Posição
Americanas	www.americanas.com.br	23	Peixe Urbano	www.peixeurbano.com.br	248
Submarino	www.submarino.com.br	40	Livraria Cultura	www.livrariacultura.com.br	256
Walmart	www.walmart.com.br	54	Posthaus	www.posthaus.com.br	274
Extra	www.extra.com.br	55	Mobly	www.mobly.com.br	292
Casas Bahia	www.casasbahia.com.br	56	Kalunga	www.kalunga.com.br	319
Ponto Frio	www.pontofrio.com.br	63	Passarela	www.passarela.com.br	325
Saraiva	www.saraiva.com.br	83	Sou Barato	www.soubarato.com.br	339
Magazine Luiza	www.magazineluiza.com.br	87	Renner	www.lojasrenner.com.br	350
Dafiti	www.dafiti.com.br	89	Privalia	www.privalia.com.br	358
Ultrafarma	www.ultrafarma.com.br	92	oBoticario	www.boticario.com.br	386
Shoptime	www.shoptime.com.br	132	Marisa	www.marisa.com.br	421
Kabum	www.kabum.com.br	143	Polishop	www.polishop.com.br	423
Ricardo Eletro	www.ricardoeletr.com.br	173	Sephora	www.sephora.com.br	427
Kanui	www.kanui.com.br	217	Tricae	www.tricae.com.br	447
Natura	www.natura.com.br	232	OQVestir	www.oqvestir.com.br	482

Tabela 1. Lista dos *e-commerces* analisados.

Foram definidos quatro cenários de testes diferentes para cada amostra na lista. Todos os testes fizeram uso de um dispositivo *desktop* localizado no Brasil em uma instância da *Amazon EC2* com o navegador *Google Chrome*. Nos cenários 1 e 3 não houve a utilização de cache do navegador, simulando o primeiro acesso do usuário. Nos cenários 1 e 2 foi utilizado uma rede cabeada com velocidade de 5 megabits de download e 1 megabit de upload por segundo com RTT (*Round Trip Time* — demora para uma informação chegar ao destinatário e o remetente receber sua confirmação) de 28 milissegundos enquanto que nos cenários 3 e 4 a velocidade da rede simula um acesso utilizando uma rede 3G com 1.6 megabits de download e 768 megabits de upload por segundo respectivamente e RTT de 300 milissegundos. A velocidade média a cabo no Brasil é de 3,6 megabits⁴ e 3G é de 2 megabits⁵ (3G em São Paulo).

Cenário de teste	Dispositivo	Local do Teste	Navegador	Cache	Tipo de Conexão
1	Desktop	Brazil — EC2	Chrome	Não	Cable (5/1 Mbps 28ms RTT)
2	Desktop	Brazil — EC2	Chrome	Sim	Cable (5/1 Mbps 28ms RTT)
3	Desktop	Brazil — EC2	Chrome	Não	Mobile 3G (1.6 Mbps/768 Kbps 300ms RTT)
4	Desktop	Brazil — EC2	Chrome	Sim	Mobile 3G (1.6 Mbps/768 Kbps 300ms RTT)

Tabela 2. Lista da configuração dos testes realizados para cada amostra.

Os testes foram realizados no dia 08 de março de 2016. A página principal de cada *e-commerce* listado na Tabela 1 foi testado uma vez (sem descartes — *outliers*) nos quatro cenários descritos na Tabela 2, sempre com a mesma configuração de rede para cada *website* em cada cenário. Em ambos os testes nenhum usuário estava logado nas páginas web, visto que os logins normalmente expiram em um curto espaço de tempo e quando um usuário real acessa um *website* (principalmente no primeiro acesso) não está logado. Foram armazenados os *dashboards* referentes ao teste de cada *e-commerce* nos quatro cenários para posterior análise do desempenho.

A ferramenta escolhida para os testes de performance foi a WebPagetest⁶, uma

4 Velocidade média a cabo no Brasil: <http://goo.gl/B1CLWT>

5 Velocidade média 3G no Brasil: <http://goo.gl/8Kk4Mk>

6 <http://www.webpagetest.org>

ferramenta *online* para testes de desempenho de páginas *web*. WebPageTest permite gerar resultados e cenários de diversas formas, inclusive vídeos com o processo de renderização das páginas e *filmstrips* (linha do tempo do processo de renderização da página com *prints* da tela acima da borda a cada 0,5 segundos). A ferramenta ainda permite a execução de dois testes de um mesmo *website* em uma única execução por meio da seleção da opção *First View and Repeat View* nas configurações avançadas.

Dessa forma, os testes 1 e 2 assim como os testes 3 e 4 descritos na Tabela 2 foram realizados em duas execuções e, portanto, as métricas de cada teste estão presentes nos mesmos arquivos. WebPagetest também permite salvar um resumo das métricas dos testes, as métricas detalhadas de cada requisição de cada teste e também um arquivo com todas as informações das transações HTTP realizadas pela página *web*. Em cada teste foram armazenados os seguintes arquivos: o *Waterfall* (visão do carregamento sequencial do *website* em forma de requisições fluindo constantemente para baixo), o *Waterfall* das conexões realizadas com cada servidor de conteúdo, a captura de tela, o *filmstrip* e o vídeo de todo o carregamento dos *websites*.

4 | RESULTADOS E DISCUSSÃO

Nas Tabela 3 e 4 são apresentadas as métricas e resultados considerados mais importantes, retirados dos testes 1 e 3⁷. Na tabela, os dados estão separados por cores, sendo azul e vermelho os valores abaixo e acima da média da amostra respectivamente. Ainda, na tabela constam valores mínimos (MIN) e máximos (MAX) de cada métrica, ambos em destaque com fundo vermelho e verde. Os testes foram realizados em 29 dos *e-commerces* listados, isto porque os testes não puderam ser efetuados no domínio <http://www.ricardoletro.com.br/> pois o servidor da loja rejeitou o acesso efetuado pelo servidor do *WebPageTest*.

⁷ Todos os demais resultados podem ser visualizados na página do projeto em <http://inscale.github.io/perf-ecommerces-br/>.

Loja	TTFB	TTSR	SpeedIndex	LoadTime	Request	Connections	Bytes In	DOM Elem.
Americanas	156	2185	13104	14212	211	66	4430441	3271
Boticario	417	6784	17625	21510	496	193	4627767	6549
Casas Bahia	151	3194	3971	8509	221	110	2091314	3392
Dafiti	129	1895	4275	13612	297	154	2168881	1986
Extra	255	3789	9496	13005	272	128	2443574	1132
Kabum	364	6604	9426	11533	240	80	2565895	5809
Kalunga	586	7586	10303	10295	146	33	3782604	1609
Kanui	94	3100	5611	7822	208	92	3743222	1578
Livraria Cultura	276	3186	5570	10105	154	69	2561703	2216
Renner	1300	3888	8149	10006	287	140	2322744	1253
Magazine Luiza	554	6198	8974	16910	166	81	3452265	2994
Marisa	222	2986	6266	17945	357	179	4183569	2238
Mobly	256	2090	4156	2518	233	112	2031785	1807
Natura	255	4091	6286	21881	347	48	4818540	4373
OQVestir	354	7298	17459	19957	249	83	2760183	2937
Passarela	415	2294	4304	20734	367	107	3463009	1290
Peixe Urbano	2498	4789	10277	13844	158	98	2762172	1544
Polishop	381	8287	15681	25811	565	238	11167147	1675
Ponto Frio	163	7196	13059	11714	204	105	2316903	1821
Posthaus	211	3185	5629	9585	201	89	1520350	1422
Privalia	1065	3887	19512	12077	163	51	4115997	2304
Saraiva	147	5086	10074	13453	304	123	4539035	1232
Sephora	248	3880	9715	16223	430	144	3185218	1359
Shoptime	292	3100	18742	13823	352	64	12744964	1386
Sou Barato	139	2584	6801	10645	167	46	3535307	4253
Submarino	143	3485	17322	19111	271	87	6181333	2133
Tricae	286	2291	7328	7877	124	32	3434053	1343
Ultrafarma	1102	4081	13430	16866	112	25	3563155	4123
Walmart	571	2486	5937	11663	166	67	1642101	2213
MÉDIA	444,96	4250,68	10090,89	13985,11	260,79	99,18	3946897,50	2465,32
MIN	94	1895	3971	2518	112	25	1520350	1132
MAX	2498	8287	19512	25811	565	238	12744964	6549

Tabela 3. Resultados do teste 1 (Cable (5/1 Mbps 28ms RTT), no cache).

Entre os dados do teste 1, os melhores resultados estão bem distribuídos entre as amostras pois cada loja se sobressai em algum aspecto. No entanto, vale ressaltar que *Casas Bahia* apresentou o melhor índice de velocidade, seguido por *Mobly* e *Dafiti*. No entanto, a loja *Polishop* obteve os piores resultados em quatro métricas: *TTSR*, *Load-Time*, *Requests* e *Connections*. Se considerarmos os valores abaixo das médias, *Tricae* com 7, *Shoptime* com 6 e *Polishop* com 5 são as piores colocadas. No teste 3 os valores estão ainda mais distribuídos. O melhor índice de velocidade aqui fica com a *Loja Passarela*, seguida de *Marisa* e *Livraria Cultura*. *Boticario* e *Polishop* com 6 juntamente com *OQVestir* e *Privalia* com 5 são as lojas com maior número de valores abaixo das médias.

Também procuramos correlações entre os valores, ou seja, se o aumento/diminuição dos valores de uma métrica possuíam relação direta com outra métrica (Tabela5). Entre as correlações mais relevantes (consideramos valores entre -0,99 a -0,50 e 0,560 a 0,99) estão *SpeedIndex* com *TTSR*, visto que ambos se referem à renderização de conteúdo na tela e de *SpeedIndex* com *Bytes In*, pois a velocidade de exibição do conteúdo depende do download do conteúdo e conseqüentemente tem relação com o tamanho dos conteúdos. Há também correlações de *LoadTime* com *Requests* e de *LoadTime* com *Bytes In*, pois o tempo de carregamento depende da quantidade de requisições realizadas e do tamanho destas requisições. Além disso, há correlação de *Requests* com *Connections* e de *Requests* com *Bytes In*, visto que

o número de conexões varia de acordo com o número de requisições, assim como o tamanho total varia de acordo com o número de requisições.

Correlações	TTFB	TTSR	SpeedIndex	LoadTime	Request	Connections	Bytes In	DOM Elem.
TTFB	1,00	0,12	0,26	-0,09	-0,34	-0,22	-0,15	0,11
TTSR	0,12	1,00	0,55	0,23	-0,20	-0,29	0,01	0,16
SpeedIndex	0,26	0,55	1,00	0,47	0,05	-0,14	0,52	-0,05
LoadTime	-0,09	0,23	0,47	1,00	0,55	0,30	0,55	0,00
Request	-0,34	-0,20	0,05	0,55	1,00	0,80	0,55	0,01
Connections	-0,22	-0,29	-0,14	0,30	0,80	1,00	0,21	0,04
Bytes In	-0,15	0,01	0,52	0,55	0,55	0,21	1,00	0,09
DOM Elem.	0,11	0,16	-0,05	0,00	0,01	0,04	0,09	1,00

Tabela 5. Tabela de correlações entre as métricas.

Traçando um panorama geral dos resultados do teste 1, temos que, a página de uma loja virtual brasileira possui, em média: 260 requisições em sua página inicial, 103 conexões são realizadas com o servidor da aplicação ou CDNs, o número total de elementos carregados pelo navegador cliente é de cerca de 2392, totalizando 3,94 Megabytes de dados, sendo que o primeiro byte chega ao navegador em 444 milissegundos e a renderização começa depois de mais de 4 segundos de espera. Já no teste 3, com rede móvel 3G, os valores permanecem praticamente os mesmos, com exceção do *time to first byte* de 1383 milissegundos (aumento de três vezes) e o *time to start render* de mais de 8 segundos (duas vezes mais lento).

Se considerarmos os piores casos, a situação fica preocupante. Por exemplo, a página inicial da loja Kalunga leva mais de 22 segundos para mostrar algo na tela em uma conexão 3G. Isso significa que o usuário ficará todo esse tempo vendo uma tela em branco. Provavelmente neste intervalo já terá desistido da compra ou até mesmo pensado que o website está *offline*. Este caso pode ser melhor compreendido visualizando o *filmstrip* da Figura 1. Cada retângulo branco é meio segundo em que o usuário está esperando pelo conteúdo.

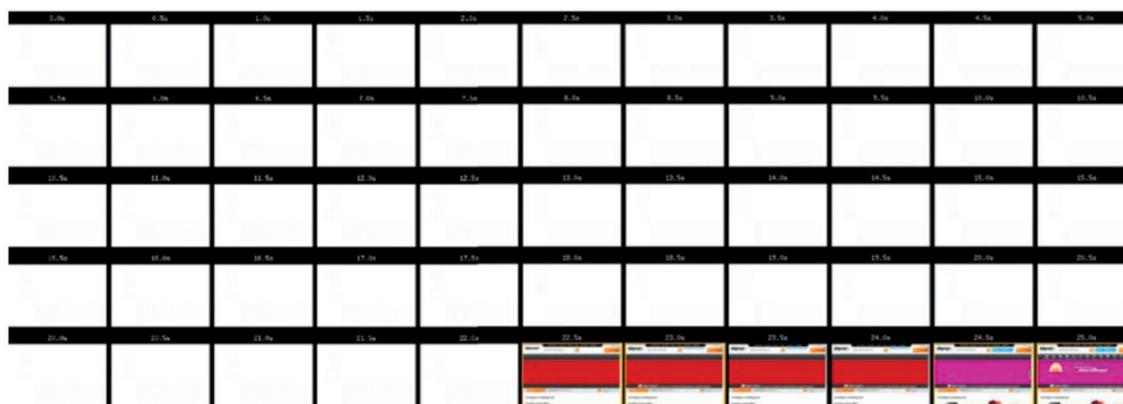


Figura 1. *Filmstrip* da página inicial da loja Kalunga testada em uma rede 3G.

5 | TRABALHOS RELACIONADOS

A empresa RADWARE realiza anualmente trabalhos de pesquisa sobre a performance de *websites* de várias áreas como lojas virtuais, sites de esportes, hotéis, etc. Por exemplo, em 2015 [RADWARE 2015] foram analisados os cem primeiros *e-commerces* listados no *ranking* do site Alexa.com por meio de algumas métricas como TTI e LoadTime e também outros detalhes como a composição das páginas em relação aos seus *assets*. Seus achados mostram que somente 12% dos sites analisados mostravam algum conteúdo abaixo da barreira dos 3 segundos; o tamanho e complexidade das páginas estão diretamente ligados com tempo de carregamento (*loadtime*); imagens não estão sendo devidamente otimizadas.

Recentemente a mesma empresa divulgou outro estudo [RADWARE 2016] mas dessa vez englobando também outras áreas. Seus resultados mostraram que lojas virtuais tiveram a melhor média de TTI (3.1 segundos, algo bem superior ao estudo prévio); o tamanho médio das páginas é 1,4 MBs, com 97 requisições e um tempo de carregamento total de 7,9 segundos; as páginas oscilaram entre 276 KB a 9 MB e 34% dos sites analisados não usavam uma compressão de imagens adequada;

No estudo apresentado neste trabalho, no entanto, o foco foi uma análise detalhada das lojas virtuais mais acessadas do Brasil, abrangendo um gama maior de métricas e uma análise da renderização das páginas considerando conexões comuns em *desktops* e redes móveis. Além disso, o trabalho também trata de detalhes relacionados a otimização *front end*.

6 | CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma avaliação da performance *web* dos 29 *e-commerces* mais acessados no Brasil segundo o ranking da Alexa. Analisando os melhores resultados de cada métrica dos *e-commerces* testados é possível perceber que poucos estão próximos das métricas ideais.

Por exemplo, de acordo com as especificações da ferramenta de medição de desempenho *web PageSpeed Insights* [GoogleDevelopers 2016], o TTFB ideal é igual ou menor que 200 ms. No Teste 1, apenas oito amostras (cerca de 27%) ficaram abaixo desse valor.

Outras métricas, no entanto, tiveram resultados bem inferiores dos desejados. Para a métrica TTSR, o valor ideal é abaixo de 1,2 segundos [WebSiteOptimization 2016], no entanto, nenhum *e-commerce* se aproximou desse valor, sendo o mais próximo 3,09 segundos da loja Dafiti no Teste 1. Outro exemplo é a métrica *SpeedIndex*, que segundo Paul Irish [Irish 2016], o valor ideal é abaixo de 1000, mas o melhor resultado alcançado foi de 3971 pela loja Americanas.

Comparando o tamanho médio das páginas brasileiras (3854 kilobytes) com o tamanho médio mundial no mesmo período (2281 kilobytes) [HTTPArchive 2016], é

possível observar que as páginas de comércio eletrônico brasileiras estão cerca de 68% maiores. Esta diferença é duas vezes maior quando comparada com a média mundial no setor de *e-commerces* [RADWARE 2015], cujo valor é de cerca de 1905 kilobytes por página. Além disso, a quantidade de elementos no DOM se mostraram altas — principalmente o *website* da loja *Shoptime* com 12 megabytes. A média mundial de elementos presentes no DOM, no mesmo período, era de 905, e o resultado médio do Teste 1 foi de 2465, quase três vezes maior.

Otimização da performance de aplicações *web* ainda é um tópico pouco discutido na indústria e ainda menos no meio acadêmico. Os dados apresentados aqui reforçam a necessidade de maior apreço pelo assunto. Mais pesquisas, técnicas e ferramentas precisam ser feitas para propiciar melhores experiências para usuários. Como trabalho futuro há previsão de extensão dos testes também para dispositivos móveis, abrindo possibilidade de outras análises como responsividade da página e comparação entre páginas *mobile* e páginas únicas. Além disso, a criação e execução de uma estratégia de otimização juntamente com a comparação do antes e depois são os próximos passos da pesquisa. Por fim, ainda é possível realizar um trabalho complementar, com viés mais qualitativo, visando capturar a percepção dos usuários finais.

REFERÊNCIAS

[Aberdeen Group 2008] Aberdeen Group (2008). **The Performance of Web Applications: Customers Are Won or Lost in One Second**. Technical Report November.

[Akamai 2002] Akamai (2002). **Why Performance Matters**. Technical report.[eMarketer 2016] eMarketer (2016). emarketer. Disponível online em <http://www.emarketer.com>.

[Everts 2015] Everts, T. (2015). **Time is Money The Business Value of Web Performance**.

[Forrester Consulting 2009] Forrester Consulting (2009). **eCommerce Web Site Performance Today**. Technical report.

[GoogleDevelopers 2016] GoogleDevelopers (2016). **Melhorar o tempo de resposta do servidor**. Disponível online em <https://goo.gl/1ZXawT>.

[Grigorik 2013] Grigorik, I. (2013). **High performance networking in google chrome**. Disponível online em <https://goo.gl/37RctB>.

[Grigorik 2016] Grigorik, I. (2016). **Ilya grigorik**. Disponível online em <https://www.igvita.com/>.

[HTTPArchive 2016] HTTPArchive (2016). **Http archive**. Disponível online em <http://goo.gl/82e14w>.

[IPNews 2015] IPNews (2015). **Paciência dos brasileiros ao utilizarem um aplicativo dura três segundos**. Disponível online em <http://goo.gl/sB6VcO>.

[Irish 2016] Irish, P. (2016). **Fast enough**. Disponível online em <https://goo.gl/4FBPZv>.

[Meenan 2016] Meenan, P. (2016). **Patrick meenan**. Disponível online em <http://blog.patrickmeenan.com/>.

[Nielsen 2009] Nielsen, J. (2009). **Powers of 10: Time scales in user experience**. Disponível online em <https://goo.gl/zJoUc9>.

[RADWARE 2015] RADWARE (2015). STATE OF THE UNION — SUMMER 2015 ECOMMERCE PAGE SPEED & WEB PERFORMANCE **Content and Speed: The Magic Formula**. Technical report.

[RADWARE 2016] RADWARE (2016). MULTI-INDUSTRY WEB PERFORMANCE 2016 STATE OF THE UNION (desktop edition) **Time is money when it comes to microwaves**. Technical report.

[Rick Viscomi 2015] Rick Viscomi, Andy Davies, M. D. (2015). **Using webpagetest: Web performance testing for novices and power users**. Technical report.

[Souders 2007] Souders, S. (2007). **High Performance Web Sites: Essential Knowledge for Frontend Engineers**.

[Souders 2016] Souders, S. (2016). **Steve souders**. Disponível online em <https://stevesouders.com/>.

[WebSiteOptimization 2016] WebSiteOptimization (2016). **Optimize start render time**. Disponível online em <http://goo.gl/s4RoFm>.

UMA ANÁLISE DOS PADRÕES DE TATUAGENS ASSOCIADOS À CRIMINALIDADE DO ESTADO DA BAHIA COM AUXÍLIO DA TEORIA DE REDES

Hernane Borges de Barros Pereira

Universidade do Estado da Bahia e Centro
Universitário SENAI CIMATEC
Salvador - Bahia

Antônio José Assunção Cordeiro

Centro Universitário SENAI CIMATEC
Salvador - Bahia

Carlos César Ribeiro Santos

Centro Universitário SENAI CIMATEC
Salvador - Bahia

Alden José Lázaro da Silva

Centro Universitário SENAI CIMATEC
Salvador - Bahia

RESUMO: Este artigo apresenta uma análise de padrões de Tatuagens associadas a ocorrências de crimes registrados pela Secretaria de Segurança Pública do Estado da Bahia utilizando a teoria de Redes Sociais e Complexas. As redes construídas consideram as relações existentes entre criminosos, crimes e tatuagens. Os resultados apresentados apontam redes com característica topológica Small World. Os resultados encontrados apontam dados relevantes para a compreensão dos padrões de criminalidade associados a tatuagens, podendo apoiar futuramente as autoridades de segurança pública na tomada de decisão.

PALAVRAS-CHAVE: Tatuagem. Segurança

Pública. Teoria de Redes

ABSTRACT: This paper presents an analysis of tattoos patterns associated with occurrences of crimes recorded by the Department of Public Safety State of Bahia, using the Networks theory. Networks built consider the relationship between criminals, crimes and tattoos. Networks present Small World phenomenon. The results show relevant data for the understanding of crime patterns associated with tattoos and may eventually support the public security authorities in decision.

KEYWORDS: Tattoos. Public Security. Network Theory.

1 | INTRODUÇÃO

O estudo da segurança pública no Brasil vem crescendo dentro do contexto acadêmico. Percebe-se uma aproximação das instituições de segurança no país com a comunidade científica. Estados e municípios buscam por soluções otimizadas para a realidade da violência e criminalidade em que estão inseridas. Tais demandas podem ser respondidas a partir de diversos estudos e métodos, dentre os quais se destaca a Teoria de Redes.

Uma rede social é uma estrutura capaz de

representar a ligação entre indivíduos, grupos de indivíduos, organizações, etc. (e.g. crimes, tatuagens, adesão a facções criminosas).

É uma das formas de representação dos relacionamentos efetivos ou profissionais de indivíduos entre si, ou entre seus agrupamentos de interesse mútuos (PEREIRA, 2013).

Pode-se afirmar que a estrutura da segurança pública de um estado é formada por diversos atores e relações que variam de acordo com entidades, áreas ou sub-áreas a que estão diretamente ou não relacionados. Policiais, órgãos, crimes cometidos, locais de crimes, criminosos, tatuagens de criminosos, entre outras características, podem formar uma extensa rede de relações. Notamos algumas redes sociais nesse âmbito.

Vale ressaltar que esforços têm sido feitos pelo Estado da Bahia na aproximação deste com a comunidade científica. Durante alguns anos da década passada, inseridos no Programa de Políticas Públicas, a Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) lançou os Editais Temáticos que tinham como foco a prevenção e o combate a violência e criminalidade, além de estimular a atividade de pesquisa em segurança pública nas universidades do Estado da Bahia.

Em 2014, o programa de Pós-Graduação em Modelagem Computacional e Tecnologia Industrial da Faculdade de Tecnologia Senai Cimatec-BA, obteve da Polícia Militar do Estado uma base de dados que contém uma pesquisa das relações existentes entre criminosos, tatuagens e crimes e um a cartilha com a síntese da proposta de análise e atuação (SILVA, 2014). Percebeu-se que importantes respostas sobre o comportamento e características dessas relações poderiam ser obtidas do uso da teoria de redes. A proposta de análise considera as relações existentes entre tatuagens e crimes, criminosos e tatuagem e crimes e criminosos.

Assim, a compreensão dos padrões de interação entre os atores analisados neste trabalho destaca o comportamento da criminalidade, sobre tudo, a questão central deste estudo: *Quais informações estratégicas, para a segurança pública do Estado da Bahia, podem ser obtidas com o uso da análise das redes sociais e complexas provenientes de uma base de dados de Tatuagens em Criminosos?*

O objetivo principal desse trabalho é modelar as relações entre criminosos e tatuagens, crimes e tatuagens e crimes e criminosos a partir da base de dados de Tatuagens da Polícia Militar da Bahia, utilizando técnicas e parâmetros da análise de redes sociais e da teoria de redes complexas.

A importância dessa pesquisa relaciona-se com uma sistematização de um método capaz de auxiliar pesquisadores, setores de inteligência da segurança pública e, principalmente, policiais, a compreender e explicar como se dão relações de interesse entre criminosos, tatuagens e crimes, contribuindo sensivelmente para futuras políticas públicas de segurança mais assertivas.

2 | FUNDAMENTAÇÃO TEÓRICA

Este artigo fundamenta-se em três importantes áreas teóricas: a teoria dos grafos, a análise de redes sociais e a teoria das redes complexas. Tais teorias têm sido aplicadas na resolução dos mais variados problemas, para diferentes áreas do conhecimento visto que têm obtido relevantes resultados para a tomada de decisão em empresas, pesquisas biológicas, tecnologia da informação, organizações de diferentes naturezas, universidades, entre outros.

O fundamento matemático dessa aplicação advém da teoria dos grafos, ramo da matemática que estuda as relações entre os objetos de um determinado conjunto. Para tal são empregadas estruturas chamadas de grafos, representado por $G = (V, E)$, onde V é um conjunto não vazio de objetos denominados vértices e finito e E é um conjunto de pares não ordenados de V , chamado de arestas.

Destaca-se ainda que uma rede é considerada complexa quando seu tamanho é suficientemente grande de modo que algumas propriedades emergem. Por sua vez, uma rede é considerada social quando a mesma lida com relações entre pessoas. As pesquisas, antes focadas nas pequenas redes e nas propriedades de vértices individuais ou arestas, passaram a considerar propriedades estatísticas em larga escala (Metz *et al.*, 2007).

Uma rede complexa pode apresentar topologias importantes, dentre as quais podemos destacar as seguintes: (1) Redes de Grafos Aleatórios (Erdős; Rényi, 1960); (2) Redes de Mundo Pequeno (*Small World*) (Watts; Strogatz, 1998); c) Redes Livres de Escala (*Scale Free*) (Barabasi; Albert: 1999).

Nas redes de grafos aleatórios, cada um dos vértices tem a mesma probabilidade de receber novas ligações. Nestas redes, a distribuição de conectividade decai exponencialmente. As redes mundo-pequeno (*Small World*) são caracterizadas por possuírem um alto grau de aglomeração e caminho mínimo médio similar ao de uma rede aleatória. A rede livre de escala possui uma distribuição de conectividade que segue uma lei de potência. Ou seja, vértices com muitas ligações possuem uma probabilidade maior de receber novas ligações.

As redes aqui apresentadas são redes de 1-modos. Com elas, analisamos o comportamento dos criminosos no que se refere aos crimes cometidos e tatuagens utilizadas por eles, bem como a relação existente entre tatuagens que podem representar determinados tipos de crimes.

3 | MÉTODO

Conforme já citado, a proposição deste artigo é construir e analisar redes de criminosos a partir de um banco de dados de tatuagens identificadas em criminosos oriundo da Polícia Militar do Estado da Bahia. A construção das redes iniciou-se com a delimitação de relevantes informações:

- Catálogo com mil criminosos: presos ou em processo. Identificados sequencialmente;
- Crimes Cometidos (tipos): sete tipos de crimes analisados que foram furto, roubo, associação ao crime, narcotráfico, homicídio, latrocínio e associação ao crime;
- Tatuagens: total de 60 tatuagens diferentes catalogadas na base de dados;
- Locais: denominados de Áreas Integradas de Segurança Pública (AISP), são os locais da região soteropolitana em que os crimes foram cometidos.

Com a base de dados e informações devidamente desnormalizadas, considerando-se ainda os fundamentos teóricos, construímos as redes sociais de indivíduos com base nos critérios Crimes-Tatuagens e nos critérios Crimes-Tatuagens-Localidade através da utilização dos softwares *Pajek* e *Gephi*. A próxima seção, apresentaremos a análise das redes de tatuagens.

4 | RESULTADOS E DISCUSSÃO

A base de dados utilizada é restrita ao município de Salvador, denotando uma lacuna para posterior análise de dados outros municípios do estado e do País.

Em um primeiro momento, esta pesquisa realizou o tratamento dos dados de modo a normalizá-los, viabilizando assim a concepção das redes. Essas foram construídas vinculando os indivíduos através da ocorrência de mesma tatuagem e crime praticado.

Desta forma, criminosos que possuíssem a mesma tatuagem estariam conectados entre si na rede. Porém, esta abordagem resultou em redes com uma densidade muito alta (mais de 29%) o que inviabiliza parte da análise, uma vez que as redes ficam altamente conectadas.

Por esse motivo, o foco da pesquisa configurou-se em construir redes com mais de um critério para estabelecer de vínculos entre os indivíduos. Assim, foram montadas redes de criminosos com vínculos definidos pela coexistência de pelo menos dois fatores em comum, a saber: mesma tatuagem e mesmo crime cometido.

Redes	Vértices v	Arestas E	Grau M. <k>	Densidade	C. M. M. L	Diâmetro	Cws
R1-CT	763	141.181	370,068	0,486	1,514	2	0,844
A1-CT	763	141.287	370,346	0,486	1,513	2	0,486
R2-CLT	666	78.975	237,162	0,357	1,653	3	0,843
A2-CLT	666	79.039	237,354	0,356	1,643	2	0,356

Tabela 1. Características das Redes de Tatuagens

A Tabela 1 mostra os dados das redes criadas e das redes aleatórias geradas a partir da quantidade de seus vértices e graus médios. A primeira rede vincula os

criminosos entre si que tenham cometido os mesmos crimes e que tenham as mesmas tatuagens. As informações dessa rede aparecem na primeira linha da tabela, com a referência R1-CT. A segunda linha da tabela contém as informações da rede aleatória (A1-CT) criada com base no número de vértices e grau médio da R1-CT.

A rede R1-CT (Figura 1), analisada através do maior componente, teve sua topologia identificada como *Small World*, visto que a distância entre qualquer par de vértices na rede é pequena. Além disso, seu coeficiente de aglomeração é maior do que o coeficiente de aglomeração da rede A1-CT, e seu caminho mínimo médio foi similar ao da mesma rede A1-CT. A caracterização da Rede R1-CT ilustra a ocorrência de dois eventos distintos, associando uma aresta entre indivíduos que possuem uma relação pelos dois eventos: (1) cometeram o mesmo crime; (2) possuem a mesma tatuagem. Esta relação possibilita a um gestor de segurança pública a identificação de grupos de tatuagens que se aglomeram por alguma prática dos crimes aqui analisados. Neste sentido, o gestor poderá perceber os padrões que explicam como os criminosos estão utilizando tatuagens para sua identificação criminosa. Destaca-se ainda a possibilidade de compreender a influência gerada pela distância entre criminosos que praticaram crimes distintos, e, até mesmo a distância entre indivíduos que cometeram crimes considerados mais brandos para crimes considerados mais graves, uma vez que muitos criminosos acabam cometendo vários tipos de crimes distintos.

A Rede R2-CLT (Figura 2) vincula os criminosos entre si, que tenham as mesmas tatuagens, cometido os mesmos crimes e atuado na mesma localidade. Essa rede é também *Small World*. A rede aleatória foi criada com base no número de vértices e grau médio da R2-CLT. Destaca-se que essa análise também utilizou a maior componente da rede R2-CLT. A relevância das informações geradas na Rede R2-CLT consiste na possibilidade do gestor de segurança pública poder identificar localidades onde os criminosos com padrões de associação de tatuagens se reúnem para realizar práticas criminosas, possibilitando o desenvolvimento de ações e políticas de policiamento ostensivo localizado, investigações de facções criminosas e a distância entre estes eventos (prática de Crimes, tatuagens localidades).

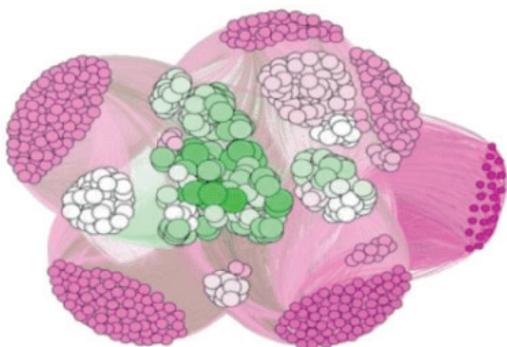


Figura 1. Rede R1-CT

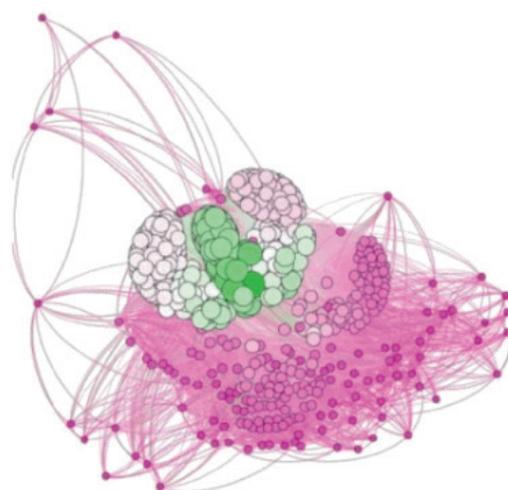


Figura 2. Rede R2-CLT

Considerando nosso modelo de análise em que um indivíduo pode ser ligado a outro, caso o mesmo tenha cometido o mesmo crime e que possua a mesma tatuagem, podemos dizer que um par de vértices é alcançável caso essa relação seja verdadeira, aumentando a chance de ser verdadeira a afirmação de que uma devida tatuagem represente um determinado crime. Estatisticamente, podemos afirmar que as tatuagens na Tabela 2 representam um crime específico, pois a tatuagem foi encontrada em 100% dos indivíduos que cometeram determinado crime.

Tatuagem	Crime	Qtde	%
Paz Justiça Liberdade	Associação ao Crime	241	100%
Bruxo	Furto	110	100%
121	Homicídio	217	100%
Coringa	Latrocínio	295	100%
Carpa	Narcotráfico	274	100%
Palhaço	Roubo	262	100%

Tabela 2. Vínculos Estatísticos de 100%

Tatuagem	Crime	Qtde	%
Coringa	Roubo	168	64%
Palhaço	Latrocínio	169	57%
Caveira	Associação ao Crime	130	54%
Coringa	Furto	57	52%

Tabela 3. Vínculos Relevantes de Menos de 100%

Estatisticamente, pode-se afirmar que há um grau elevado de possibilidade das tatuagens abaixo representarem um crime, pois foi possível observá-las em mais de 50% das ocorrências analisadas (Tabela 3). Com relação às demais tatuagens, não temos condições de afirmar que as mesmas estão associadas a um determinado tipo de crime, pois não foi constatado em mais de 50% das ocorrências (Tabela 4):

Tatuagem	Crime	Qtde	%
157	Latrocínio	146	49%
Palhaço	Furto	45	41%
Carpa	Associação ao Crime	91	38%
157	Roubo	98	37%
Arma Fogo	Latrocínio	106	36%
Arma Fogo	Roubo	90	34%
Coringa	Homicídio	74	34%
Caveira	Narcotráfico	93	34%
121	Furto	37	34%
Paz Justiça Liberdade	Narcotráfico	92	34%

Palhaço	Homicídio	69	32%
Jesus	Roubo	79	30%
PCC	Latrocínio	88	30%

Tabela 4. Vínculos de Baixa Relevância

5 | CONSIDERAÇÕES FINAIS

O modelo e a análise propostos permite-nos responder à questão de pesquisa: *Quais informações estratégicas, para a segurança pública do Estado da Bahia, podem ser obtidas com o uso da análise das redes sociais e complexas provenientes de uma base de dados de Tatuagens em Criminosos?*

Por um lado, observamos que é factível, por inspeção visual e por análise de agrupamento, identificar os membros de facções criminosas através de tatuagens. Por outro lado, a análise de influência a partir dos crimes cometidos e das tatuagens dos criminosos evidencia a distância entre indivíduos que cometeram crimes considerados mais brandos e crimes considerados mais graves. Esse tipo de constatação permite à segurança pública antecipar-se a situações criminosas, uma vez que muitos criminosos acabam cometendo vários tipos de crimes distintos ao longo do tempo.

A combinação das medidas de redes apresentadas nesse artigo, a partir da determinação dos atributos dos atores, fornecem diversos subsídios para a análise da estrutura dos arranjos e as respectivas relações entre os criminosos em questão. Os resultados encontrados também inferem novos questionamentos à pesquisa aqui apresentada.

REFERÊNCIAS

Barabási, A. L.; Albert, R. (1999) "Emergence of Scaling in Random Networks". *Science*, v. 286, p. 509-512.

Erdős, P.; Rényi, A. (1960) "On the evolution of random graphs". *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, n. 5, p. 17-61.

Metz, J. et al. (2007) "Redes Complexas: conceitos e aplicações", São Carlos: [s.n.]..

Plano Estadual da Segurança Pública - PLANESP: Governo da Bahia - 2012 a 2015 (2011). Secretaria da Segurança Pública. - Salvador: Secretaria da Segurança Pública.

Pereira, H. B. B. (2013) "Redes Sociais e Complexas: Aplicações em Difusão do Conhecimento". *Academia de Ciências da Bahia: Memória*, v. 3, p. 39-47.

Silva, Alden L. (2011) "Tatuagem: desvendando segredos", Salvador: Magic Gráfica, 74p.

Watts, D. J. & Strogatz, S. H. (1998) "Collective dynamics of 'small-world' networks", *Nature*, v. 393, n. 4, p. 440-442.

UM ESTUDO DE CASO DE AUTOMAÇÃO DE TESTE EM SISTEMAS LEGADOS SOBRE PLATAFORMA FLEX

Augusto Boehme Tepedino Martins

Universidade Federal de Santa Catarina
Florianópolis - SC

Jean Carlo Rossa Hauck

Universidade Federal de Santa Catarina
Florianópolis - SC

RESUMO: A realização de testes demonstra cada vez mais sua importância nas organizações que desenvolvem software. Ferramentas empregadas servem para trazer uma melhor confiabilidade e agilidade ao que está sendo produzido. Realizar testes, no entanto, é caro e demorado, e automatizar testes é uma alternativa interessante e tecnicamente viável. Sistemas legados comumente apresentam limitações para a automação de testes e isso ocorre também em sistemas Web desenvolvidos utilizando a plataforma Flex. Assim, este trabalho apresenta o desenvolvimento da automação de testes para plataforma Flex. Requisitos foram desenvolvidos para a escolha de uma ferramenta de automação de testes, a fim de escolher a que melhor atendesse as necessidades das aplicações, e que evitasse retrabalho. Um estudo de caso é então planejado e realizado objetivando a automação de testes de um sistema desenvolvido em plataforma Flex. Casos de teste são elaborados, uma ferramenta é selecionada, os scripts de teste

são desenvolvidos e executados em um estudo de caso em uma empresa de desenvolvimento de software. Os resultados observados apontam indícios de que a automação de testes pode aumentar inicialmente os esforços de elaboração dos casos de teste mas reduzem o tempo de execução, tendendo a gerar benefícios futuros.

PALAVRAS-CHAVE: Testes. Integração. Automação. Flex.

ABSTRACT: Software testing is increasing its importance in software development organizations. Used tools serve to bring better reliability and agility to what is being produced. Performing tests, however, is costly and time consuming, and automate testing is an interesting and technically viable alternative. Legacy systems often have limitations for test automation and it occurs on Web systems developed using the Flex platform. Thus, this work presents the development of test automation for Flex platform, test cases are elaborated, an automation tool is selected, test scripts are developed and executed in a case study over a software developer company. Test cases are developed, a tool is selected, test scripts are defined and implemented in a case study on a software development company. The observed results indicate initial evidence that test automation may initially increase the

development efforts of the test cases but reduce the runtime, tending to generate future benefits.

KEYWORDS: Tests. Integration. Automation. Flex.

1 | INTRODUÇÃO

Com a tecnologia cada vez mais avançada, muitas pessoas acabam não observando o quanto os softwares estão envolvidos em suas vidas, possivelmente porque suas ações são de tal forma transparentes aos seus usuários que se tornam algo trivial no seu dia-a-dia.

Com essa dependência atual de softwares em grande parte das atividades humanas, os produtos de software e o processo de criação de um software passaram a ser objeto de estudo, mesmo que um software gratuito venha a ser produzido, não será bem aceito se não for de fácil manuseio ou se possuir defeitos nas suas funcionalidades.

Especificamente falando de aplicativos que rodam na Web, algumas características esperadas de qualidade, como: disponibilidade de acesso, desempenho, evolução contínua e segurança [1]. Entretanto, a qualidade esperada nem sempre é alcançada. Algumas vezes, produtos de software têm sido entregues aos clientes cobrindo apenas os requisitos funcionais especificados, sem eliminar possíveis erros. Uma das maneiras de garantir que a qualidade de um produto de software seja atendida, e que as funcionalidades que o cliente necessita estejam presentes, é a aplicação de testes de software.

Entretanto, alguns testes são feitos por pessoas e, portanto, não são perfeitos, são suscetíveis a falhas. Muitas vezes as pessoas acabam deixando alguns pequenos problemas passarem despercebidos ou, também, possuem visões diferentes de como fazer os testes.

Nesse sentido, as padronizações e a automação de testes destacam-se na forma de garantir a qualidade de software. A padronização dos testes, como por exemplo, a documentação sobre casos de testes, com o passo-a-passo de como o teste deve ser feito, diminui a chance de variações na forma como os testes são realizados interferirem nos seus resultados. Já a automação de testes tende a diminuir a falha humana nos processos de testes. Assim, um grupo menor de pessoas poderia automatizar alguns tipos de testes que seriam repetidos toda vez que fosse necessário.

Ao se produzir um novo software, é necessária uma documentação dos testes para possuir um controle do que foi feito e do que será feito em relação ao software. Existem algumas ferramentas específicas para isso. Mas, infelizmente, nem todas as ferramentas de automatização de testes possuem uma integração com tais ferramentas. Assim, existe uma chance que esta documentação dos testes automatizados seja perdida, ou que saia de controle.

Assim, o presente trabalho apresenta um estudo de caso de desenvolvimento de automação de testes para a plataforma Flex com objetivo de reduzir a falha humana e a necessidade de repetir testes manuais no sistema legado.

As próximas seções são divididas da seguinte forma: na seção 2 é apresentado o estado da arte, seguindo da seção 3 onde a abordagem metodológica deste trabalho é brevemente apresentada; na seção 4 o estudo de caso é descrito e na seção 5 as conclusões são apresentadas.

2 | ESTADO DA ARTE

Como o presente trabalho trata da automação de testes de uma aplicação legada sobre a plataforma Flex, este tópico busca por soluções de automação já experimentadas para essa plataforma.

Assim, a busca por trabalhos similares foi realizada em consulta as seguintes ferramentas de pesquisa: Google (<http://www.google.com>), Google Scholar (<http://scholar.google.com>), Portal CAPES (http://www-periodicos-capes-gov-br.Ez46.periodicos.capes.gov.br/index.php?option=com_phome), pela facilidade de acesso e relevância como fontes de pesquisa, procurando envolver tanto referências da literatura quanto experiências da indústria.

Nessas ferramentas de busca, foram utilizadas as seguintes palavras de busca: “Software”, “Teste automatizado”, “Flex” e algumas variações/complementos: “Automação de testes de software”, “benefícios de testes automatizados”, “automatização de testes com Flex”, “Diferenças entre Flex e HTML”. Também foram utilizadas nas pesquisas as traduções na língua inglesa dos mesmos termos.

Sobre os resultados encontrados, para selecionar aqueles mais pertinentes a esta pesquisa, os seguintes critérios de inclusão foram utilizados:

- Ferramentas de automação de Testes com suporte à plataforma Flex: pois uma ferramenta para automatizar testes que não suporta a plataforma Flex, não é relevante ao trabalho;
- Plug-ins de automação de teste para Flex: tendo mesmo motivo de procura como foi relatado no critério acima;
- Relatos de experiência de automação de teste com Flex: para saber como ferramentas trabalham nesta plataforma e ter uma melhor decisão sobre o software escolhido

Para apoiar a busca, alguns requisitos de automação de testes para uma ferramenta legada sobre plataforma Flex foram identificados, com base: (i) nas características de ferramentas de automação de testes já previamente identificadas na literatura; (ii) nas necessidades para automação de testes coletadas por meio de entrevistas com analistas de testes da empresa desenvolvedora do software (vide

seção 4); e (iii) em entrevista com o gerente de projetos da empresa desenvolvedora do software; e, especialmente, (iv) nas limitações técnicas de automação de testes na plataforma Flex, que impossibilita a utilização de webdrivers por não manipular html diretamente:

- R1: Possuir alguma versão gratuita;
- R2: Suportar testes automatizados;
- R3: Estar atualizada e possuir comunidade ativa;
- R4: Suportar testes automatizados de aplicações que utilizam a plataforma Flex;
- R5: Suportar as versões mais recentes dos navegadores: Firefox, Chrome, Internet Explorer, e Opera;
- R6: Suportar testes automatizados em Aplicações Web;
- R7: Possuir documentação abrangente;
- R8: Ser capaz de utilizar um repositório central para que múltiplos usuários possam utilizar os scripts gerados

Após a execução das buscas nas ferramentas de pesquisa indicadas, foram encontradas as seguintes ferramentas que satisfazem, ao menos em parte, os requisitos propostos (vide Tabela 1).

Ferramenta	Descrição	R1	R2	R3	R4	R5	R6	R7	R8
Flex-ui-Selenium	Flex-ui-Selenium é uma extensão da ferramenta Selenium para a plataforma Flex, estendendo as possibilidades de uso do Selenium Webdriver, e pode gerar scripts nas linguagens Java, C#, Perl, Ruby, e PHP.	+	+	+	+	+	+	-	+
FlexMonkey	FlexMonkey é um aplicativo Adobe AIR gratuito, desenvolvida pela GorillaLogic que permite a criação de testes automatizados em plataformas AIR e Flex	+	+	-	+	+	+	+	+

Badboy	Badboy é uma ferramenta de automação de testes baseada na interface gráfica do usuário (GUI – Graphics User Interface), trabalhando com a movimentação do mouse e com um pequeno script, usando de modo Record/Playback para tornar ações do usuário em Scripts	+	+	+	-	+	+	+	+
Testmaker	Testmaker é uma ferramenta gratuita que consegue tornar um script de teste em um teste funcional, um teste de carga, ou um teste de performance, que cria scripts em Java, Ruby, Python, e PHP	+	+	+	?	-	+	-	+
Sikuli	Sikuli é uma ferramenta de automações de teste híbrida, na qual trabalha tanto com GUI, como com scripts de usuário. Interage com a tela a partir da captura de imagens, ou regiões, definidas no script	+	+	+	+	+	+	+	+
Bugbuster Test Record	Bugbuster Test Record é uma extensão feita para o Google Chrome, permitindo a criação de testes a partir de cliques feitos na janela do browser	+	+	+	?	-	+	+	-

Tabela 1. Ferramentas selecionadas

- +: atende completamente
- -: atende parcialmente ou não atende
- ?: não foi possível determinar

As ferramentas foram então instaladas e avaliadas uma a uma em relação aos requisitos propostos.

Conforme observado na Tabela 1, o Flex-ui-Selenium não cumpre o R7, pois não foi possível encontrar uma documentação abrangente, o que dificultou pesquisas e tentativas de utilização da ferramenta. O FlexMonkey não cumpre o R3, pois a ferramenta foi descontinuada, e não receberá mais atualizações.

A ferramenta Badboy falha no R4, pois não oferece suporte à plataforma Flex:

A ferramenta espera um retorno do comando para dar continuidade às ações, mas o Flex não responde este comando. Não foi possível descobrir se o Testmaker suporta a Plataforma Flex, pois não foi possível encontrar documentação para poder até mesmo instalar a ferramenta com facilidade, e o TestMaker funciona apenas sobre Firefox. Já a ferramenta Bugbuster foi possível de ser instalada, mas não foi possível determinar se a ferramenta suporta testes de aplicações na linguagem Flex, e a ferramenta apenas funciona na plataforma Google Chrome.

Após essa avaliação, a ferramenta que melhor atendeu os requisitos foi a ferramenta Sikuli, que cobriu todos os requisitos definidos.

3 | ABORDAGEM METODOLÓGICA

Nesta seção, são apresentadas as etapas da abordagem metodológica do presente trabalho.

Segundo Runeson [2], pode-se definir um estudo de caso como um estudo focado de um fenômeno em seu contexto, especialmente quando quando os limites entre o fenômeno e o contexto não podem ser bem definidos. Runeson [2] indica também que é compreensível aplicar estudos de caso às áreas relacionadas à engenharia de software tais como desenvolvimento, operação e manutenção de software. É necessário planejar o estudo de caso, com uma base bem definida, para se analisar as unidades, os métodos e técnicas para coletas de dados e para se definir o controle e os resultados da pesquisa.

- Cada uma das etapas é brevemente apresentada na sequência:
- Planejamento: objetivos são definidos e o estudo de caso é planejado;
- Preparação: procedimentos e protocolos para coleta de dados são definidos;
- Coleta: execução e coleta de dados do estudo de caso;
- Análise: onde os dados coletados são analisados e as conclusões são extraídas;

Relatório: onde os dados analisados são reportados.

A seção 4 a seguir detalha as principais etapas desta pesquisa no contexto do estudo de caso.

4 | ESTUDO DE CASO

De acordo com [2], é essencial para o sucesso de um estudo de caso que ele possua um bom planejamento. Assim, no contexto deste trabalho, o estudo de caso foi planejado onde foram definidos: a unidade de análise, os métodos utilizados para a coleta de dados, cronograma, participantes, etc. De forma resumida, o planejamento

do estudo de caso é apresentado na sequência, iniciando pela definição da unidade de análise.

4.1 Contextualização do Estudo de Caso

A empresa SPECTO Tecnologia é uma empresa que iniciou suas atividades no início da década de 90, e vem desenvolvendo produtos atualmente por meio de três linhas de produtos: CXM (gestão de atendimento), DCIM (gestão de datacenters), e BMS (gestão de prédios inteligentes).

O presente estudo de caso é realizado na divisão de gestão de datacenters (DCIM) da empresa. A DCIM tem como objetivo o desenvolvimento de produtos de software para monitorar e controlar o ambiente de datacenters, permitindo, por exemplo, alertar o usuário que um ambiente está com muita fumaça, ou que um dispositivo de monitoramento foi desconectado, por exemplo.

O sistema utilizado neste estudo de caso é o DataFaz Unity, um gerenciador de infraestrutura de datacenters, monitorando parâmetros físicos de ambiente, tais como: umidade, temperatura, energia, controle de acesso, etc. O sistema utiliza Flex para o desenvolvimento de sua interface com o usuário e o backend é desenvolvido em Java, com banco de dados tipicamente utilizado sendo PostgreSQL.

Neste estudo de caso são envolvidos os membros das equipes responsáveis pelos testes do sistema DataFaz Unity (membros da equipe de garantia de qualidade de produto), sendo três analistas de testes (dois formados em ciências da computação e um em automação) e dois testadores (graduandos de Ciências da Computação).

4.2 Preparação

- A coleta de dados corresponde ao conjunto das operações, através das quais o modelo de análise é submetido ao teste dos fatos e confrontado com dados observáveis[3]. O princípio da coleta de dados vem da utilização de várias fontes de evidência, criação de um banco de dados, de forma a encadear evidências de forma a buscar respostas para a pergunta de pesquisa deste trabalho: “Seria possível automatizar testes para um software Flex de forma a reduzir o esforço, tempo e retrabalho, tomando por base uma ferramenta já existente para modelagem e automatização de testes?”, os seguintes dados foram planejados para serem coletados:
- Quantidade de pontos de caso de uso;
- Quantidade de erros encontrados na execução dos testes automatizados;
- Avaliação subjetiva dos resultados observados pelos envolvidos na área de testes;

Quantidade de esforço realizado para a elaboração dos casos de testes automatizados.

4.3 Execução e coleta de Dados

O estudo de caso foi executado durante o período de quatro meses, após a análise de qual ferramenta de automação seria utilizada.

A ferramenta Sikuli, conforme já apresentada, é uma ferramenta de automação de testes de GUI que mescla Scripts com imagens, utilizadas como parâmetros, similares a constantes declaradas em qualquer linguagem de programação, que são procuradas quando o programa está rodando como. O Sikuli possui uma IDE própria que suporta linguagens Python e Ruby em uma plataforma Java: Jython e JRuby respectivamente, mas pode ser utilizado como um script de outras linguagens, como por exemplo Java, para utilização de outras IDEs, como o Eclipse e NetBeans. A IDE facilita o trabalho para capturar e adaptar imagens às necessidades de cada automação. Entretanto, sua utilização em outra IDE é mais complicada de se utilizar, porém pode ser integrada com outras linguagens e ferramentas.

Segundo Yeh et al. [4], a ideia de utilizar imagens para auxiliar na automação de testes vem da própria forma como ocorre interação na comunicação humana. Assim, na ferramenta Sikuli, as imagens são utilizadas como parâmetros, de maneira similar a constantes declaradas em qualquer linguagem de programação, onde o usuário pode definir o nome da imagem, o grau de similaridade que procura na tela, e aonde irá selecionar a tecla quando necessário.

Para utilizar a ferramenta, foi necessário instalar o Java JDK. Entretanto, infelizmente durante os testes não foi possível integrá-la ao Eclipse e utilizar a linguagem Java, pois não foi encontrada documentação que facilitasse a utilização do script com a ferramenta. Assim, os scripts de teste foram implementados em Jython.

Com a ferramenta instalada e pronta para executar os testes de forma automatizada, foi preparado o ambiente com as demais ferramentas necessárias e realizada a criação dos scripts de automação de testes. Para efetuar os testes manuais, configurou-se um computador para possuir o sistema DataFaz Unity instalado localmente, e também o software Enterprise Architect utilizado pela empresa para documentar a análise e modelagem do software (para organizar e documentar os casos de uso e os casos de teste com seus cenários), o PgAdmin para administração do banco de dados e recuperação do backup do banco de dados somente contendo a configuração inicial.

Com esse ambiente instalado, foram elaborados os casos de teste que seriam realizados tanto manualmente quanto de forma automatizada, de forma a possibilitar a comparação mais objetiva dos resultados da automação. Os casos de testes foram documentados na ferramenta Enterprise Architect, conforme já citado.

Foram elaborados, ao todo, dez casos de testes completos, cada um possuindo de três a cinco diferentes cenários, tomando-se por base as principais funcionalidades do sistema DataFaz Unity. Os casos de teste foram documentados seguindo as melhores práticas definidas na norma ISO/IEC/IEEE 29119 [5]. Para cada caso de teste foi realizada a implementação do seu correspondente em script de teste automatizado.

A Figura 1 mostra o extrato de um caso de teste para execução manual e a figura 2 mostra o mesmo caso de teste para execução automatizada e parte seu código de automação.

Identificador	TST-001.01
Item de teste	Login Padrão
Especificações de Entrada	1. Usuário: fazion; 2. Senha de acesso: 22; 3. Clicar no botão "Ok".
Procedimentos	1. Inserir o dado "Usuário" na área "Nome de Usuário"; 2. Inserir o dado "Senha de acesso" na área "Senha"; 3. Selecionar o botão "Enviar".
Especificações de Saída	Após o procedimento ser realizado, a tela deve ser alterada, apresentando a tela principal da aplicação.
Ambiente necessário	DataFaz Unity, banco de dados.
Exigências especiais	Não aplicável.
Interdependências	Fazer uso do caso de teste <u>específico</u> e PST - Proposta Técnica.

Figura 1 – Extrato de Caso de Teste para execução manual

TST-001.01	Código Sikuli IDE
Login Padrão	
1. Usuário: fazion; 2. Senha de acesso: 22; 3. Clicar no botão "Ok".	<pre># Especificações de entrada: login = "fazion" #Login do sistema password = "22" #Senha do sistema</pre>
1. Inserir o dado "Usuário" na área "Nome de Usuário"; 2. Inserir o dado "Senha de acesso" na área "Senha"; 3. Selecionar o botão "Enviar".	<pre># Definição para abrir uma nova Janela do Chrome defopenff(): ffLoc = r'C:\Program Files (x86)\Mozilla Firefox\firefox. Exe' App(ffLoc).focus() # opens FF if exists("HttpBar.png",10): keyDown(Key.CTRL + "l") paste("https://localhost:8443/datafaz/") type(Key.ENTER) # Definição para realizar Login defsystemlogin(lgin, pssw): # Procura o padrão da imagem e seleciona com o mouse um campo 47 pixels deslocados para baixo do eixo X. # Em código puro: # t = find(Pattern("LoginTab.png").targetOffset(0,47)) t = find(</pre>

Figura 2 – Extrato de Caso de Teste e código de automação

Para garantia da qualidade dos testes, um computador possuindo o sistema DataFaz Unity instalado localmente, o Enterprise Architect com a modelagem do DataFaz Unity para organizar e documentar os casos de uso e casos de teste com seus cenários, o PgAdmin como banco de dados, e um arquivo de backup, com dados limpos a ser utilizado como cenário inicial dos casos de teste.

Cada um dos casos de teste foi então executado manualmente por dois testadores diferentes e cada um dos scripts de teste automatizado foi executado dez vezes em um mesmo computador. Ao final dos testes, os analistas de teste e testadores foram submetidos a um questionário de avaliação, composto de quatro perguntas utilizando escala Likert de respostas, procurando identificar as percepções sobre o resultado da aplicação dos testes automatizados.

4.4 Análise dos Resultados

Durante toda a execução dos testes, os dados de duração foram coletados por meio do registro das tarefas no sistema de gerência de projetos utilizado pela empresa¹. Como resultado, conforme esperado, foi percebido que o tempo necessário para elaboração dos testes automatizados e manuais apresentou grande diferença, devido ao tempo utilizado na elaboração dos Scripts de testes automatizados, conforme mostra a Tabela 2.

Métricas	Teste Manual	Teste Automatizado
Elaboração dos casos de teste	05h00min	05h00min
Elaboração Scripts de Testes Automatizados	-	07h30min
Tempo médio de execução	00h08min	00h02min30seg
Total Duração	05h08min	12h32min30seg

Tabela 2 - Comparações entre teste manual e automatizado

Entretanto, conforme também mostra a Tabela 2, o tempo necessário para execução dos testes automatizados foi consideravelmente inferior. Espera-se que as futuras execuções dos testes automatizados venham a compensar o maior esforço aplicado na elaboração dos scripts de teste.

Além dos dados de tempo de execução, a partir da aplicação do questionário, foi possível coletar a impressão subjetiva dos participantes em relação aos resultados da automação de testes no sistema legado utilizando plataforma Flex.

Questões	Concordam
Facilidade na compreensão dos testes	80%
Cenários cobriram a funcionalidade proposta	80%
A infraestrutura disponibilizada para o ambiente de testes foi satisfatória para a execução dos testes	80%
O tempo gasto com teste automatizado diminui esforços de custo e tempo com execuções repetitivas	100%

Tabela 3 - Resultados do questionário

Conforme pode ser percebido pelas respostas dos participantes, a automação de testes utilizando a ferramenta Sikuli, atendeu às expectativas da maioria. Entretanto, foram percebidas oportunidades de melhoria pela equipe, como por exemplo, a possibilidade de integração direta com o Eclipse, possibilitando a implementação dos scripts em Java.

¹ <http://www.jexperts.com.br/category/pse/pse-pch/>

4.4.1 Ameaças à validade

Como tipicamente ocorre em estudos de caso [6][7], o presente trabalho somente foi aplicado em um único software de uma única empresa, não sendo possível desta forma, generalizar seus resultados a outras empresas e cenários, mesmo que positivos.

Da mesma forma, a empresa possui poucos analistas de testes na unidade organizacional do estudo de caso, fazendo com que os resultados obtidos fossem escassos, mais analistas trariam mais respostas, e o resultado da aplicação questionário seria mais abrangente. Além disso, o autor do trabalho teve participação ativa na execução do trabalho, e este envolvimento direto pode gerar viés de interpretação de dados coletados.

5 | CONCLUSÕES

Este trabalho apresenta uma experiência de automação de testes em um sistema legado desenvolvido sobre a plataforma Flex. Tendo em mente as limitações de automação de testes em Flex, foram analisadas diferentes abordagens de automação. Foi possível notar que nenhuma ferramenta iria poder automatizar a aplicação em sua totalidade, então foram criados requisitos para escolher qual ferramenta seria utilizada no trabalho. A ferramenta Sikuli, que cobriu a maior parte dos requisitos propostos, foi escolhida.

No contexto de um estudo de caso, casos de teste foram elaborados, sendo implementados scripts de teste na linguagem aceita pelo Sikuli. Os testes foram então executados de forma manual e automatizada, em um ambiente controlado.

Os dados coletados de esforço e tempo durante a elaboração e realização dos testes no estudo de caso levantam indícios de que o ganho de tempo na automação de testes é pequeno inicialmente. Espera-se, entretanto, que a organização envolvida no estudo de caso obtenha benefícios diretos com a redução de custos de retrabalho e aumento de confiabilidade da aplicação com a execução dos testes automatizados em futuros releases do sistema.

Como trabalhos futuros planeja-se a integração dos testes automatizados com a ferramenta de documentação e modelagem dos testes. Além disso, seria importante expandir os casos de teste automatizados para o restante das funcionalidades do sistema.

6 | AGRADECIMENTOS

Os autores agradecem aos membros da equipe da empresa Specto que participaram do estudo de caso. Também agradecem a cooperação e interesse da gerência da empresa no estudo de caso.

REFERÊNCIAS

Pressman, R. S. **Engenharia de software: uma abordagem profissional**; Tradução Ariovaldo Griesi e Mario Moro Feccio. 7ª ed. São Paulo: AMGH, 2011.

Runeson, P., and M. Höst. **Guidelines for conducting and reporting case study research in software engineering**. Empirical Software Engineering 14.2, 2009.

Quivy, R. et al. (1992). **Manual de Investigação em Ciências Sociais**. Lisboa: Gradiva.

Yeh, Tom, Chang, Tsung-Hsiang, Miller, Robert C. (2009): **Sikuli: using GUI screenshots for search and automation**. In: Proceedings of the ACM Symposium on User Interface Software and Technology, 2009, pp. 183-192. <http://doi.acm.org/10.1145/1622176.1622213>. International Organization for Standardization. ISO/IEC/IEEE 29119. Software Testing Standard. 2010.

Zelkowitz, M. V. **An update to experimental models for validating computer technology**, J. Syst. Softw., vol. 82, pp. 373–376, 2009.

Yin, R. K. **Applications of case study research**, ed. 3, 2011.

UM AMBIENTE VIRTUAL APLICADO AO ENSINO E PESQUISA EM ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

**Guilherme Álvaro Rodrigues Maia
Esmeraldo**

Instituto Federal de Educação, Ciência e
Tecnologia do Ceará campus Crato
Crato – Ceará

Edson Barbosa Lisboa

Instituto Federal de Educação, Ciência e
Tecnologia de Sergipe campus Aracaju
Aracaju – Sergipe

RESUMO: O estudo e pesquisa em arquitetura e organização de computadores têm ganhado ainda mais relevância em função da rápida evolução tecnológica, que têm disponibilizado uma variedade de plataformas computacionais cada vez mais complexas. O grande desafio é entender e dominar as peculiaridades das novas tecnologias. Para isso, o uso de ferramentas virtuais têm sido cada vez mais estimulado. Assim o propósito desse trabalho é apresentar um ambiente integrado de concepção e simulação de plataformas computacionais que possam ser facilmente configuradas e expansíveis, elevando o nível de abstração e facilitando o entendimento da integração entre hardware e software.

PALAVRAS-CHAVE: Arquitetura e Organização de Computadores, Simulador Virtual, Apoio ao Aprendizado.

ABSTRACT: Studies and research in architecture and computer organization has gained even more relevance due to the rapid technological evolution, which has made available a variety of increasingly complex computing platforms. The big challenge is to understand and master the peculiarities of new technologies. For this, the use of virtual tools have been increasingly stimulated. Thus the purpose of this work is to present an integrated environment for the design and simulation of computational platforms that can be easily configured and expandable, raising the level of abstraction and facilitating the understanding of the integration between hardware and software.

KEYWORDS: Computer Architecture and Organization, Virtual Simulator, Learning Support.

1 | INTRODUÇÃO

O estudo e projeto integrado de hardware e software tem se tornado ainda mais relevante devido aos grandes avanços tecnológicos, que têm disseminado uma vasta quantidade de plataformas computacionais para atender diferentes requisitos tecnológicos, tais como computação paralela, sistemas embarcados, internet das coisas (IoT) e automação (ECKER, 2009) (ATZORI, 2010). Esse contexto implica

em um grande desafio para o ensino e pesquisa em ambientes acadêmicos devido à complexidade inerente à alta escala de integração, o que dificulta a percepção de conceitos e funcionalidades presentes em arquiteturas e plataformas modernas.

Portanto, o uso de ambientes virtuais de simulação como abordagem de ensino e pesquisa tem sido fortemente recomendado, pois permitem simplificar e dinamizar a criação e a simulação de novos sistemas computacionais em cenários próximos aos reais. Assim, o objetivo desse trabalho é propor uma ferramenta de simulação para apoiar o projeto e a aprendizagem em arquitetura e organização de computadores. A ferramenta proposta, chamada de CompSim, suporta uma abordagem para projeto e estudo integrado dos conceitos relacionados aos componentes de hardware, suas funções e interações em si, bem como sua programação em baixo nível.

2 | TRABALHOS RELACIONADOS

Os ambientes de simulação como o HADES (Hamburg Design System) (HENDRICH, 2002) e o DEEDS (Digital Electronic Education Design Suite) (DONZELLINI; PONTA, 2013) possibilitam o estudo desde portas lógicas a sistemas microprocessados completos, em diferentes níveis de abstração. No entanto, a configuração e parametrização de componentes específicos não é algo tão simples, devido à complexidade do ambiente proposto. Abordagens como CPUSim (SKRIEN, 2001) e SimuS (SILVA; BORGES, 2016) incluem interfaces gráficas que permitem a configuração e interação com os componentes de hardware e o ambiente de simulação, e focam essencialmente na exploração da programação em baixo nível.

Esses simuladores, no entanto, por não incluírem suporte a alguns tipos de modelos de componentes ou ocultarem detalhes importantes para abstração da simulação, tornam-se ambientes de simulação distantes dos reais. Já os ambientes de simulação, como o VisiMips, são fortemente baseados no processo de animação da simulação em interface gráfica, vinculando a programação de baixo e médio nível com o comportamento dinâmico das unidades funcionais do processador (KABIR; BARI; HAQUE, 2011). No entanto, esse simulador só dá suporte a um único tipo de processador. Por outro lado, simuladores como o MPSoCBench (DUENHA; AZEVEDO, 2016), que apresentam componentes de hardware modelados seguindo todas as características de componentes reais, se tornam muito complexos de configurar e interagir.

Este trabalho apresenta o CompSim, um simulador gráfico que suporta o projeto de diferentes modelos de componentes de hardware, os quais podem ser facilmente configurados e simulados, além de trazer informações detalhadas e estatísticas de simulação sobre eles. A seção a seguir traz mais detalhes do simulador proposto.

3 | O AMBIENTE VIRTUAL COMPSIM

O simulador aqui proposto possui um ambiente gráfico integrado com diferentes recursos para apoio ao projeto e estudos de sistemas computacionais. Entre os recursos, pode-se citar:

- a) Editor de código - com ele é possível codificar uma nova aplicação em baixo nível para execução no simulador;
- b) CPU – exibe, em tempo de execução, os estados assumidos pelos registradores de um determinado processador;
- c) CACHE MEMORY - exibe os estados das linhas de memória cache;
- d) MAIN MEMORY - exibe os estados dos diferentes endereços da memória principal;
- e) DEVICES - exibe informações de comunicações de entrada e saída com os periféricos durante uma simulação;
- f) SIMULATION - esse componente possibilita configurar e controlar as simulações;
- g) LOG - neste componente são exibidas informações dos eventos e estatísticas de simulação gerados por cada um dos componentes de hardware;
- h) Assembler – valida um novo programa de máquina (análises léxica e sintática), e gera código binário executável.

O simulador CompSim foi codificado – componentes de simulação de hardware e interface gráfica – com a linguagem de programação Python 3.5 (PYTHON, 2018) e compilado com o framework Nuitka (NUITKA, 2018) para as plataformas MS Windows e GNU/Linux (32 e 64 bits).

4 | ESTUDO DE CASO: PROCESSADOR CARIRI

Para avaliação da abordagem de projeto utilizando o simulador proposto, criou-se um modelo de simulação de um novo processador simples de 16 bits, denominado Cariri, o qual possui um novo conjunto de instruções (*Instruction Set Architecture - ISA*). Suas principais unidades funcionais e o caminho de dados (datapath) podem ser vistos na Figura 1.

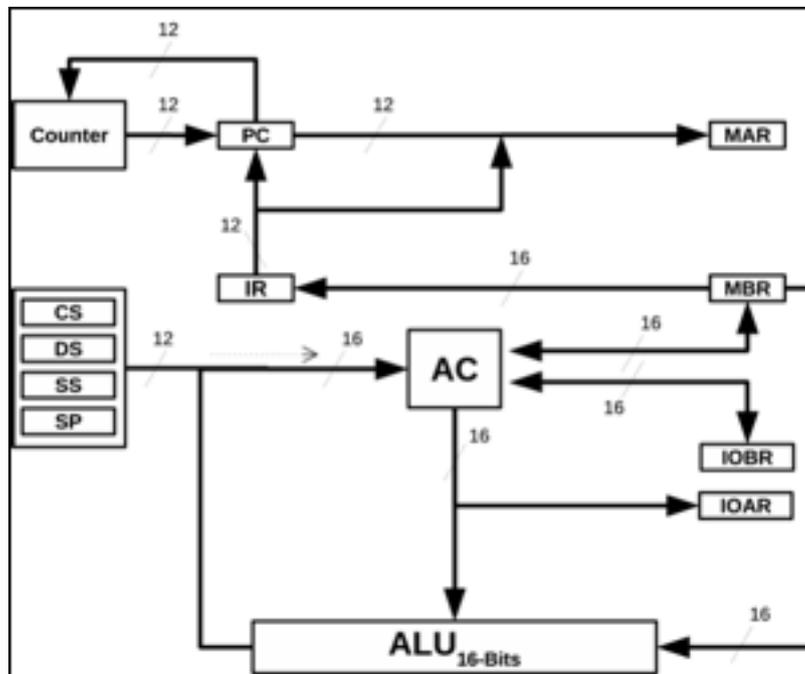


Figura 1. Processador Cariri: Unidades Funcionais e Datapath.

Entre as principais características desse processador, destacam-se: uma unidade lógica e aritmética (ULA), espaço de endereçamento diferenciado para entrada/saída e acesso à memória principal; suporte a diferentes modos de endereçamento; um banco de registradores para suportar 16 instruções de baixo nível; e suporte a operandos inteiros de 16-bits com sinalização (*signed int*) e a caracteres (*bytes*).

A plataforma computacional, criada para suportar a execução do novo modelo de processador, inclui memórias RAM e Cache, dispositivos de Entrada/Saída e um barramento. Para a simulação, utilizou-se uma aplicação Assembly, criada utilizando a ISA do processador Cariri, para cálculo iterativo do fatorial do número inteiro 7 (o resultado do cálculo do fatorial desse número está dentro do intervalo numérico que pode ser representado por um inteiro de 16-bits com sinal).

A Tabela 1 apresenta as estatísticas geradas pelos modelos de componentes da plataforma do estudo de caso, ao final da execução da aplicação de fatorial.

Componente			
CPU	Memória Cache	Memória Principal	Barramento
1. Instruções: 905	1. Acessos: 1707	1. Acessos: 3311	1. Transações: 1707
	Leituras.:1500	Leituras: 3104	Leituras:1500
	Escritas: 207	Escritas: 207	Escritas: 207
	2. Cache Hits: 1319	2. Blocos acessados: 388	2. Trans. Memória:1707
Arit./Lógicas: 252	Hits (leitura): 1189	Leituras: 388	Leituras: 1500
Transfer. Dados: 501	Hits (escrita): 130	Escritas: 0	Escritas: 207
Controle: 151	3. Cache Misses: 388	3.Bytes transf.: 6622	3. Transações de E/S: 0
Sistema: 1	Misses (leitura): 311	Leituras: 6208	Leituras: 0
	Misses (escrita): 77	Escritas: 414	Escritas: 0

Tabela 1 – Estatísticas dos componentes de hardware para a aplicação de fatorial.

5 | CONCLUSÃO E TRABALHOS FUTUROS

O uso de simuladores é uma prática recomendada no meio acadêmico, tanto para o aprendizado quanto para o projeto de novos sistemas computacionais. Acredita-se que o simulador proposto neste artigo permitirá otimizar a aprendizagem em arquitetura e organização de computadores, através de uma abordagem para estudo integrado dos conceitos relacionados aos componentes de hardware, bem como o projeto de novos sistemas computacionais.

Trabalhos futuros incluem suportar diferentes arquiteturas de processadores, plataformas de hardware especificadas em linguagem de descrição de hardware (HDL) e possibilitar a interação com outros ambientes por meio de cossimulação.

REFERÊNCIAS

ATZORI, L.; IERA, A.; MORABITO, G. **The Internet of Things: A survey**. Elsevier, 2010.

DONZELLINI, G.; PONTA, D. **From Gates to FPGA: Learning Digital Design with Deeds**. In: Proceedings of the Third Interdisciplinary Engineering Design Education Conference – IEDEC, 2013. p. 41-48.

DUENHA, L.; AZEVEDO, R. **Utilização dos Simuladores do MPSoCBench para o Ensino e Aprendizagem de Arquitetura de Computadores**. In: International Journal of Computer Architecture Education (IJCAE), V. 5, n. 1, 2016. p. 26-31.

ECKER, W.; MÜLLER, W., DÖMER, R. **Hardware-dependent Software Principles and Practice**. Springer, 2009.

HENDRICH, N. **From CMOS-Gates to Computer Architecture: Lessons Learned from Five Years of Java-Applets**. In: Proceedings of the 4th European Workshop on Microelectronics Education, EWME, 2002. p. 23-24.

KABIR, M. T.; BARI, M. T.; HAQUE, A. L. **VisiMips: Visual Simulator of MIPS32 Pipelined Processor**. In: International Conference on Computer Science & Education (ICCSE), 2011. p. 788–793.

NUITKA. **Nuitka Home**, 2018. Disponível em: <<http://nuitka.net/>>. Acesso em: 08 de ago. 2018.

PYTHON. **Welcome to Python**, 2018. Disponível em: <<https://www.python.org/>>. Acesso: 08 de ago. 2018.

SILVA, G. P.; BORGES, J. A. dos S. **SimuS: Um Simulador para o Ensino de Arquitetura e Organização de Computadores**. In: International Journal of Computer Architecture Education (IJCAE). V. 5, n. 1, 2016. p. 7-12.

SKRIEN, D. CPU Sim 3.1: **A tool for simulating computer architectures for computer organization classes**. In: Journal on Educational Resources in Computing (JERIC), 1(4), 2001. p. 46-59.

SISTEMA INTEGRADO DE GESTÃO ESPORTIVA: UMA FERRAMENTA DE APOIO AO PROGRAMA TALENTO OLÍMPICO DO PARANÁ¹

Robson Parmezan Bonidia

Universidade Tecnológica Federal do Paraná
(UTFPR) Cornélio Procópio – Paraná

Luiz Antonio Lima Rodrigues

Universidade Estadual de Londrina (UEL)
Londrina – Paraná

Rosangela Marques Busto

Universidade Estadual de Londrina (UEL)
Londrina – Paraná

Jacques Duílio Brancher

Universidade Estadual de Londrina (UEL)
Londrina – Paraná

RESUMO: Este capítulo tem como objetivo relatar o desenvolvimento de uma ferramenta de apoio a gestão esportiva do programa Talento Olímpico do Paraná, denominado SIGE—TOP (Sistema Integrado de Gestão Esportiva — Talento Olímpico do Paraná), que foi desenvolvida pela Universidade Estadual de Londrina em um trabalho conjunto do Departamento de Computação e da Secretaria de Estado do Esporte do Paraná (SEES). Esta ferramenta tem o propósito de realizar um acompanhamento e automatização das ações do programa, além de gerenciar e continuamente aprimorar as políticas, procedimentos e processos do projeto. Como resultado obteve-

se uma alternativa viável para integração entre as áreas envolvidas no esporte, garantindo aos jovens acadêmicos, atletas e técnicos, por meio da participação de atividades sistematizadas, a utilização do tempo de forma contributiva para o desenvolvimento de suas aptidões e o trabalho cooperativo.

PALAVRAS-CHAVE: Gestão Esportiva, TOP, SIGE, Governo do Estado do Paraná.

ABSTRACT: This work aims to demonstrate the development of a sports management support tool from the program Olympic Talent of Paraná, denominated SIGE—TOP (Integrated System Of Sports Management - Olympic Talent from Paraná), which was developed by the State University of Londrina in a joint work of the Computing Department and the State Department of Sport from Paraná (SEES). This tool has the purpose of monitoring and automating program actions, besides managing and continually improving its policies, procedures and its processes. As a result, it was obtained a feasible alternative to integration between the areas involved in the sport, guaranteeing to young academics, athletes, and coaches, through the participation of systematized activities, the use of their time in contributory form to the development of their

¹ O presente trabalho teve uma versão publicada em um evento da Sociedade Brasileira de Computação (SBC), especificamente no Simpósio Brasileiro de Sistemas de Informação, 2017.

skills and cooperative work.

KEYWORDS: Sports Management, TOP, SIGE, Government of the State of Paraná.

1 | INTRODUÇÃO

Com o rápido desenvolvimento e melhoria da vida pública e da economia social, toda a sociedade entrou na era da informação. Inerente a esse fator, o mundo dos esportes apresentou uma sistematização do desenvolvimento esportivo, a partir de políticas públicas ou privadas que tinham como elementos centrais a captação e formação de talentos esportivos mediante sistemas de treino (BASTOS,2003). Dada essa expansão, surgiu-se a necessidade de um termo conhecido como gestão esportiva.

De acordo com (NOLASCO et al., 2006), a gestão do esporte ou administração esportiva é responsável pelo controle, organização e direção, racional e sistemática, de atividades esportivas e físicas em geral. Especificamente, ela tem a finalidade de administrar atividades relacionadas a competições de alto nível (NOLASCO et al., 2006). As definições deste domínio de conhecimento mudam conforme países e continentes.

A Sociedade Norte-Americana define esta área como uma coleção de conhecimentos que se relacionam com a direção, liderança e organização do esporte (ROCHA; BASTOS, 2011; NOLASCO et al., 2006; XU, 2015). Na prática, a administração esportiva é uma tarefa de suporte ao esporte desde o século XIX, na Europa e nos Estados Unidos. No Brasil, as primeiras abordagens apresentaram-se no início do século XX e depois, mais amplamente, no final da década de 1920 e durante a década de 1930 nos meios militares (BOTELHO; MONTEIRO; VALLS, 2007; ROCHA; BASTOS, 2011; NOLASCO et al., 2006; XU, 2015). Neste período o tema em exame foi ligado à organização de competições e a gestão de informações esportivas.

Além disso, a gerência de dados esportivos começa a ser compreendida como estratégia para a gestão do esporte. A gestão da informação não é considerada um projeto ou uma ferramenta, mas sim um processo com objetivos e vantagens definidos. Segundo (BOTELHO; MONTEIRO; VALLS, 2007) os benefícios são: proteger e valorizar o capital intelectual, prover informação relevante, identificar quem sabe o quê, reutilizar soluções, ganhar produtividade, melhorar o processo de inovação, reduzir os silos de informação, obter e disseminar conhecimento externo e aumentar o grau de colaboração.

É correto afirmar que os benefícios descritos podem ser alcançados mediante um sistema integrado de gestão, que se apresenta como uma solução primordial para um controle eficaz e centralizado. Nesse contexto, uma ferramenta de gestão se mostra conveniente, pois abrange unicamente, em um sistema, informações de todos

os setores de uma administração (CARRAVETTA, 2006).

Todas essas informações são agrupadas, disponibilizando oportunidades de cruzamentos de dados, que podem gerar conhecimentos relevantes e apoio na tomada de decisão. Por intermédio desta solução, o fluxo de trabalho e atuação de cada profissional é relevantemente reduzido, além de apresentar baixos custos de implementação (CARRAVETTA, 2006).

Com base nessas atribuições, o presente capítulo tem o objetivo de apresentar o desenvolvimento de um sistema integrado de gestão esportiva, denominado SIGE—TOP, que proporciona o acompanhamento e automatização das ações do programa Talento Olímpico do Paraná (TOP). Este capítulo retrata a evolução do sistema, sua arquitetura, seus módulos, a organização de sua base de dados, suas etapas de uso e os resultados que a ferramenta proporcionou após sua implementação.

O restante deste documento está organizado da seguinte forma: A Seção 2 descreve trabalhos relacionados à gestão da informação. A Seção 3 apresenta as etapas executadas para o desenvolvimento do sistema de auxílio à gestão esportiva do programa TOP. Por fim, na seção 4, as considerações finais são retratadas.

2 | TRABALHOS RELACIONADOS

A literatura científica está repleta de trabalhos que demonstram a importância de sistemas de gestão em várias áreas do conhecimento. Deste modo, para a fundamentação do presente estudo, foram identificados trabalhos correlatos à sistemas gestores. Em comum, essas pesquisas demonstram que a implantação de um sistema gerenciador produz uma maior confiabilidade e confidencialidade dos dados, em que um ambiente monitorado em tempo real é estabelecido. Consequentemente, isso se converte na redução do retrabalho e dedução dos custos com o mesmo (ARANTES, 1998; BASTOS, 2003; CARVALHO et al., 2016).

Dentre as pesquisas selecionadas, os autores de (AVILA-SANTOS et al., 2015) abordam o desenvolvimento de um sistema integrado para controle de custos municipais, com a finalidade de possibilitar a visualização e comparação de despesas entre escolas de um município. Em (KALIKOSKI et al., 2009), os autores demonstraram a eficácia de um sistema de gerência de pesca compartilhada. Uma alternativa viável para a efetiva participação de governo e usuários, na administração da pesca, e como um caminho para a descentralização do gerenciamento de um pesqueiro.

Os trabalhos de (RAMOS; OCAÑA; OLIVEIRA, 2016; SUCUPIRA, 2015; TEIXEIRA et al., 2016) apresentaram sistemas de informação para coordenação de projetos científicos baseados em simulações computacionais, ferramentas web para controle e acompanhamento de uso de medicação em ambientes de vivência assistida, e fatores motivadores e inibidores que impactam a decisão de adotar um sistema de gestão para operadoras de planos de saúde.

Em uma linha de estudo similar a proposta deste trabalho, os autores de (ALBUQUERQUE, 2004; BOTELHO; MONTEIRO; VALLS, 2007; MOCSÁNYI; BASTOS, 2009; RITTER; RIGO, 2016) descreveram conceitos relativos à informação, conhecimento e gestão, relacionando as iniciativas de sistemas gerenciadores de informações esportivas.

Essas pesquisas relataram que é possível afirmar que um sistema de gestão é uma estrutura comprovada para gerenciar e, continuamente, melhorar as políticas, procedimentos e processos de uma organização. Mediante esta estrutura é possível controlar seus riscos sociais, ambientais e financeiros, melhorar a eficácia operacional e promover inovação (ARANTES, 1998; BASTOS, 2003).

3 | SISTEMA INTEGRADO DE GESTÃO ESPORTIVA (SIGE—TOP)

Tendo em vista as contribuições que um sistema de gestão pode exercer, como exposto na seção anterior, e a necessidade de suporte para as funções de planejamento, organização, direção e controle do programa TOP, formou-se uma equipe interdisciplinar com o objetivo de desenvolver um sistema para este fim. Assim, docentes e discentes, de cursos de graduação e pós-graduação, do departamento de Computação da Universidade Estadual de Londrina, em um trabalho conjunto com a Secretaria de Estado do Esporte do Paraná (SEES), desenvolveram um sistema integrado de gestão esportiva, para automatizar e apoiar as ações deste programa.

3.1 Programa Talento Olímpico do Paraná

O programa TOP é uma iniciativa do Governo do Estado do Paraná por meio da Secretaria de Estado do Esporte e do Turismo. Ele foi implementado em 2011, com a finalidade de incentivar e distribuir oportunidades aos atletas e técnicos paranaenses, que desejam construir e se dedicar a carreira esportiva de forma efetiva, por meio de bolsas. Este programa oferece auxílio financeiro para atletas, paratletas e técnicos (BUSTO et al., 2014a; BUSTO et al., 2014b). O eixo que conceitua o Programa TOP é o da oportunidade de revelar atletas que tenham a intenção de construir uma carreira esportiva, preocupação que surge da necessidade não somente de identificar, mas de oportunizar e permitir ao talento esportivo que se dedique mais efetivamente à sua modalidade e carreira esportiva, por meio do fornecimento de bolsa, com base em critérios de desempenho técnico.

O programa é executado com recursos oriundos do próprio Governo do Estado, de patrocínios diretos e incentivos fiscais autorizados pelo Ministério do Esporte por meio da Lei de Incentivo ao Esporte. O TOP abrange 34 modalidades e é dividido em sete categorias: Escolar, Formador, Nacional, Internacional, Técnico, Técnico formador e Olimpo (BUSTO et al., 2014a; BUSTO et al., 2014b). De acordo com o regulamento do programa, os principais objetivos do TOP são:

- Motivar os atletas e técnicos por meio de oferecimento de bolsa;
- Identificar e desenvolver potenciais talentos para as Olimpíadas e Paralimpíadas;
- Tornar o Paraná referência no esporte olímpico e paralímpico;
- Valorizar talentos esportivos e minimizar as possibilidades de evasão dos atletas paranaenses para outros centros;
- Analisar e medir o desempenho dos atletas e técnicos em competições por meio de avaliações;
- Acompanhar o desenvolvimento dos atletas em treinos, competições e contribuir para o desenvolvimento social.

3.2 Arquitetura

Para o projeto de arquitetura do SIGE—TOP, foi adotado um modelo que se baseou no trabalho (ANDRADE et al.,2016). Esse modelo divide o sistema em três camadas: apresentação, aplicação e dados. Essa arquitetura é ilustrada na Figura 1.

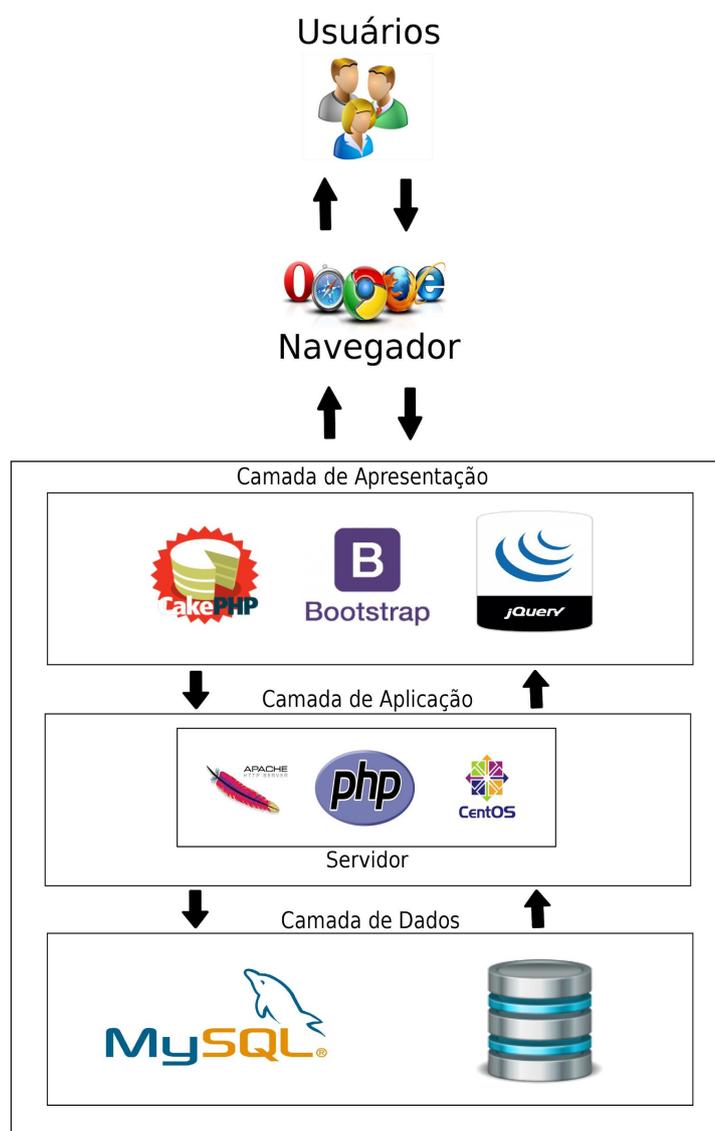


Figura 1 – Arquitetura do sistema proposto

Na figura acima, a camada de apresentação contém os elementos exibidos ao usuário final e permite a interação com o sistema. A versão web foi desenvolvida através do *Framework CakePHP*, *Framework bootstrap* e *Framework jQuery*. A camada de aplicação é responsável pelo processamento das operações requisitadas pelos usuários e contém toda a lógica do sistema, desenvolvido através da linguagem de programação *PHP*. Por fim, a camada de dados é responsável por armazenar todos os dados gerados e processados pela camada de aplicação e que são visualizados na camada de apresentação. Para gerenciamento do banco de dados foi utilizado o sistema de código aberto *MySQL*.

3.3 Módulos do Sistema

O sistema é composto por alguns módulos principais que estão divididos em:

- **Cadastro:** Este módulo é responsável pelo cadastro dos atletas e técnicos do programa. Mediante ele, é possível inserir dados pessoais e acompanhar a situação cadastral (aberta, fechada, homologada). Este foi desenvolvido de modo a ser configurável, sendo possível definir o período no qual os atletas e técnicos podem efetuar cadastros;
- **Controle de documentos:** Módulo responsável por gerar a documentação;
- **Relatórios:** Módulo pelo qual os atletas e técnicos reportam as atividades;
- **Administração:** Módulo em nível macro, disponível somente para os integrantes da Secretaria de Esporte e Turismo (SEET), dos Núcleos Regionais de Educação e das Federações. Ele pode ser dividido em diferentes submódulos, sendo eles: Gerenciamento de documentos, relatórios, controle financeiro, cadastro, pendências, seleção, visualização, homologação, mala direta, exportação de dados, controle de senhas e mapas.

3.4 Etapas de Uso

Iniciada às inscrições do programa TOP, o sistema gestor fornece a primeira etapa de cadastro. Os interessados em participar do programa fazem a criação de um usuário, o cadastro de dados básicos e confirmação de participação. Os inscritos são separados pelo sistema em 7 categorias:

1. **TOP Formador:** Atletas com idade entre 11 e 14 anos que estejam estudando em escolas públicas e participem de competições escolares em esportes Olímpicos e Paralímpicos;
2. **TOP Escolar:** Atletas com idade entre 11 e 18 anos (modalidade Olímpica) e 11 a 21 anos (modalidade Paralímpica), matriculados no ensino público ou privado com resultados expressivos em competições estaduais, nacionais ou internacionais;
3. **TOP Nacional:** Atletas com reconhecimento nacional e resultados expressivos em

competições dentro e fora do Estado;

4. **TOP Internacional:** Atletas que tenham defendido o Brasil em competições internacionais e mantenham vínculo federativo no estado do Paraná;
5. **TOP Olimpo:** Atletas que tenham participado da última edição dos Jogos Olímpicos ou Paralímpicos e que estejam vinculados a respectiva Federação Esportiva no Estado do Paraná;
6. **TOP Técnico:** Estejam residindo e treinando atletas ou equipes em esportes Olímpicos ou Paralímpicos no Estado do Paraná;
7. **TOP Técnico Formador:** Estejam residindo e treinando atletas ou equipes que participam de competições escolares, devendo estar vinculados a Estabelecimentos de Ensino da Rede Estadual do Paraná.

Após o término das inscrições, o sistema libera uma nova etapa para os coordenadores dos Núcleos Regionais de Educação e das Federações esportivas selecionarem os bolsistas. Uma vez selecionados, os bolsistas devem gerar os contratos no SIGE—TOP e enviá-los assinados para a SEET. Em seguida, com os contratos em mãos, a SEET em Curitiba valida as inscrições e homologa as bolsas. Por fim, os bolsistas selecionados e inscritos devem preencher relatórios de rendimento no sistema, informando detalhes dos treinos, do local e também como a bolsa está sendo investida. Todas as etapas acima expostas, são controladas automaticamente pelo sistema através de variáveis de início e fim.

3.5 Módulo Administrativo

O maior módulo do sistema é o Administrativo, que é composto por vários submódulos. Entre eles estão:

- **Mala Direta:** Responsável pelo envio de e-mails de aviso geral da equipe da SEET. Neste submódulo, pode-se filtrar o envio de e-mails para os Núcleos Regionais de Educação, Federações e também categorias de bolsistas. Sempre que um e-mail é enviado, uma cópia oculta é repassada para o correio eletrônico oficial do programa, com o objetivo de garantir a entrega dos e-mails enviados;
- **Visualização:** É possível realizar uma série de filtros dos bolsistas cadastrados, envolvendo Cidade, Nome, CPF, entre outros atributos;
- **Seleção:** Página de seleção dos candidatos a bolsa e visualização dos já selecionados. Atua como um acompanhamento do trabalho dos Núcleos Regionais de Educação e Federações por parte da SEET;
- **Pendências:** Permite aos gestores cadastrar novas pendências de docu-

mentos com aviso automático para os bolsistas. Esse mecanismo agiliza o processo de validação de cadastro. Anteriormente, esse método era feito somente via correspondências, o que aumentava consideravelmente o tempo para homologação;

- **Homologação:** Utilizado pela SEET para homologar as bolsas e bolsistas;
- **Controle de Relatórios:** Permite verificar a quantidade e a periodicidade em que os bolsistas estão fazendo o preenchimento dos relatórios de rendimento;
- **Pagamentos:** Controle dos pagamentos de bolsa que já foram feitos. Neste submódulo, pode-se verificar se o atleta já preencheu os relatórios (pré-requisito básico para recebimento de bolsa);
- **Exportação de Dados:** Geração de relatórios em formato *CSV*, *Xlsx* e *XML*;
- **Cadastrar Bolsistas:** Neste submódulo um administrador pode cadastrar um novo candidato a bolsa mesmo fora do período hábil;
- **Alterar senha:** Gerar novas senhas para bolsistas;
- **Mapa:** Responsável por anexar informações de cadastro em um mapa interativo. Onde é possível verificar a quantidade de bolsistas por cidade, além de informações básicas do IBGE, como população e área;
- **Gráficos:** Concede aos administradores a permissão para gerar gráficos distintos de resultados obtidos por bolsistas do programa.

3.6 Organização do Banco de Dados

Um bom sistema de banco de dados deve apresentar um projeto que visa a organização das informações, para que o futuro *software* obtenha boa performance e também facilite as manutenções que venham a acontecer (ABITEBOUL; HULL; VIANU, 1995; MATSUMOTO, 2006). Baseado nessas afirmações, o projeto SIGE-TOP elaborou um esquema lógico de uma base de dados. A Tabela 1, exemplifica o esquema e apresenta o tipo de dado armazenado em cada relação do banco.

Tabela	Descrição	Tabela	Descrição
Usuários	Credenciais dos usuários do sistema. Os usuário são divididos em grupos, de acordo com seu papel dentro do sistema.	Grupos	Armazena as opções de grupos disponíveis no sistema.
Categorias	Armazena as categorias existentes no TOP.	Federações	Informações das federações esportivas.

Modalidades	Informações das modalidades esportivas. Cada modalidade é relacionada a uma federação responsável.	Nres	Contém as informações dos Núcleos Regionais de Educação.
Escolas	Informação sobre as escolas do Paraná. Cada escola é relacionada ao Núcleo Regional de Educação.	Seed	Informações do SEED.
Municípios	Armazena o nome de todos os municípios do Paraná.	Pessoas	Informações básicas das pessoas cadastradas no sistema.
Atletas	Informações dos bolsistas que se cadastraram nas categorias que pertencem aos atletas.	Técnicos	Informações dos bolsistas que se cadastraram nas categorias que pertencem aos técnicos.
Atletas mod	Relação que armazena as modalidades dos atletas.	Técnicos mod	Relação que armazena as modalidades dos técnicos.
Eventos	Contém as informações dos eventos que os bolsistas participaram.	Imagens	Os bolsistas cadastram as imagens do evento, para a administrativo ter acesso.
Colocações	Contém as informações das colocações que um atleta conseguiu em um determinado evento.	Provas	Informações dos diferentes tipos de provas dentro de cada modalidade.
Lesões	Contém dados das lesões dos bolsistas.	Interrupções	Dados de interrupção por não lesão dos bolsistas. Problemas familiares, escolares, entre outros.
Relatórios	Relatórios dos bolsistas sobre a dedicação ao esporte.	Avaliações	Dados das avaliações feitas nos bolsista antes, durante ou ao fim do processo de bolsa.
Relat gastos	Gastos dos bolsistas com a bolsa.	Contratos	Código do contrato.
Pendências	Informações dos documentos que precisam ser entregues.	Homologados	Contém as informações dos bolsistas homologados.
Pagamentos	Contém os dados dos pagamentos das bolsas por bolsista.	Mapas	Informações de IBGE para apresentação do mapa estático.

Tabela 1 – Organização do Banco de Dados

Após o término do projeto lógico, ele foi transposto para um banco de dados físico. Essa implementação proporcionou ao programa TOP o controle centralizado de seus dados e uma atualização e recuperação de informações mais rápida e eficiente.

3.7 Apresentação do Sistema

Até o momento o SIGE—TOP¹ abrange o cadastro dos atletas e técnicos do Estado do Paraná, o controle de documentos, a inclusão e geração de relatórios por parte dos atletas, técnicos e administradores. Além disso, de maneira geral, o sistema é gerenciado pelos Núcleos Regionais de Ensino, Federações e integrantes da SEET do Paraná. A Figura 2 apresenta a página inicial do sistema. A página inicial apresenta informações e instruções para os usuários sobre o programa TOP, prazos de entrega de documentos, relatórios, atalhos para as principais funções da ferramenta e um módulo de ajuda.



Figura 2 – Página Inicial SIGE-TOP

Esse módulo do sistema possui uma página chamada Fale Conosco, que seria um canal de atendimento aos usuários. Essa função permite que os usuários enviem dúvidas, sugestões ou problemas encontrados ao utilizar a ferramenta de forma geral. As requisições são enviadas diretamente para o e-mail do suporte técnico.

No módulo administrativo, a ferramenta é utilizado por 32 Núcleos Regionais de Educação, que apresentam 1474 escolas cadastradas no sistema, sendo 883 estaduais, 21 federais, 162 municipais e 408 privadas. Integrantes da secretaria estadual de esporte e turismo e 37 federações do Estado do Paraná também fazem uso da ferramenta.

Como destacado no início deste capítulo, o SIGE—TOP possui usuários intitulados como administradores gerais, onde possuem uma página específica, que é apresentada na Figura 3. Mediante essa página, é possível: administrar, consultar e homologar,

1 O sistema pode ser acessado no link: <http://www.top2020.uel.br/top2020/>.

inscrições e cadastros, de atletas e técnicos; visualizar e controlar os relatórios dos bolsistas; enviar e-mails por mala direta; gerenciar informações financeiras; gerar gráficos; e solicitar ajuda, entre outras funções básicas de um sistema gestor.

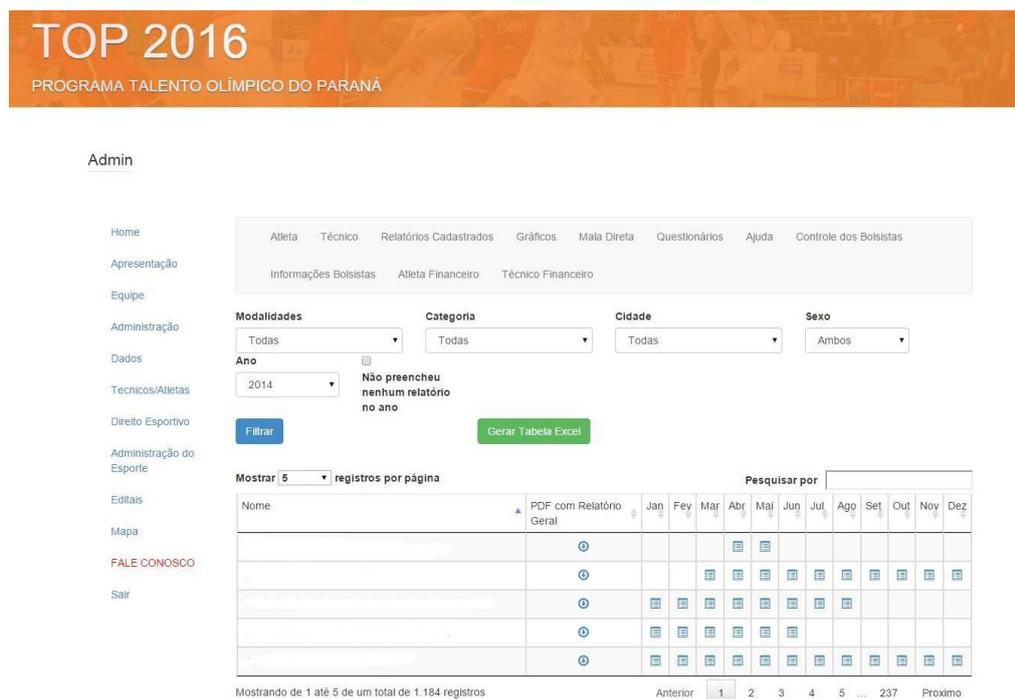


Figura 3 – Página administradores SIGE—TOP

A figura acima apresenta a página administrativa do sistema, demonstra especificamente o submódulo para controle de relatório dos bolsistas. Ele é responsável pela gerência da inclusão de atividades e resultados de atletas e técnicos. Os relatórios anexados ao sistema, indicam um total 7633 eventos cadastrados, como pode ser visto na Figura 4. Basicamente, a figura demonstra que 723 eventos são municipais, 3034 estaduais, 2377 nacionais, 983 internacionais e 516 olímpicos. São 60 modalidades esportivas incluídas nos relatórios de eventos, sendo 38 olímpicas e 22 paralímpicas. Por fim, a Figura 5 exemplifica o submódulo gerencial, responsável por anexar informações de cadastro em um mapa interativo.

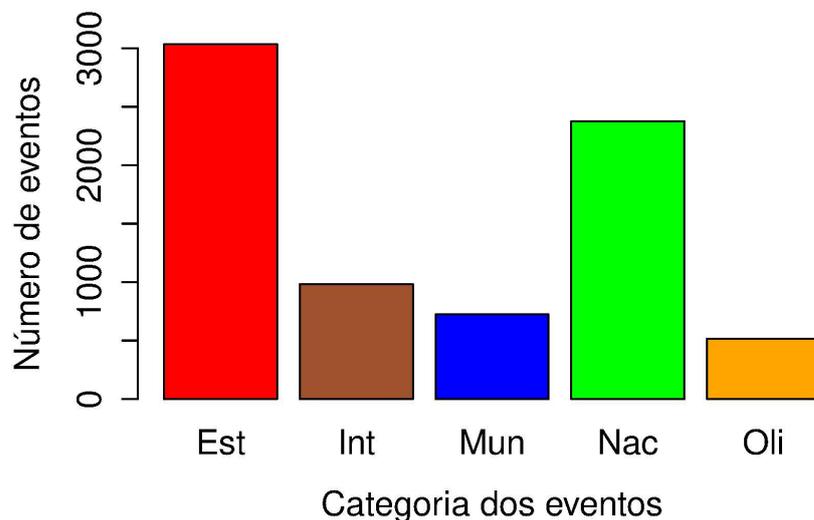


Figura 4 – Eventos Cadastrados no SIGE—TOP

Por intermédio dessa página, é possível verificar a quantidade de bolsistas por cidade, além de dados básicos do IBGE, como população e área. Com base nas atribuições retratadas nesta subseção e como evidenciado, apresentado e relatado neste documento, o SIGE—TOP está executando seu propósito que é coletar, armazenar e distribuir informações autênticas do programa TOP, com a finalidade de apoiar funções organizacionais e proporcionar integração, coordenação, controle, análise, acompanhamento, comunicação e visualização do ambiente de trabalho, transformando dados em informações úteis para tomada de decisões (LAUDON; LAUDON,2004).



Visitante

- Home
- Apresentação
- Equipe
- Administração
- Federações
- Núcleos Regionais
- Dados
- Técnicos/Atletas
- Recadastrar e-mail ou Reenviar Senha
- Direito Esportivo
- Administração do Esporte
- Editais
- Mapa
- Video-Tutorial
- FALE CONOSCO

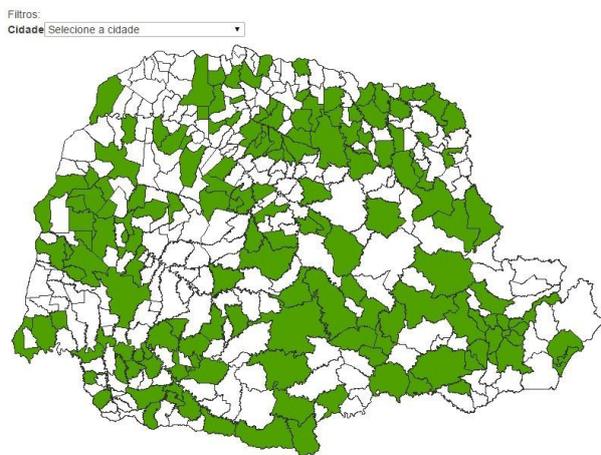


Figura 5 – Página mapa interativo SIGE—TOP

Todavia, o sistema ainda prossegue seu desenvolvimento, conseqüentemente, novos módulos estão sendo projetados com objetivo de aplicar continuamente melhorias na ferramenta.

4 | CONSIDERAÇÕES FINAIS

Este documento relatou o desenvolvimento de um sistema integrado para gestão esportiva do programa TOP, responsável por incentivar e distribuir oportunidades aos atletas e técnicos paranaenses que desejam construir e dedicarem-se a carreira esportiva de forma efetiva, por meio do oferecimento de bolsas de auxílio financeiro. O sistema, denominado SIGE—TOP, foi desenvolvido pela Universidade Estadual de Londrina em um trabalho conjunto do Departamento de Computação e da Secretaria de Estado do Esporte do Paraná. A ferramenta tem como finalidade realizar o acompanhamento e automatização das ações do programa TOP. Por meio desse capítulo, foi possível conhecer a evolução do sistema, sua arquitetura, seus módulos, a organização do projeto da base de dados e suas etapas de uso.

Pode-se concluir que, através da ferramenta desenvolvida, os gestores esportivos poderão ter uma visão mais ampla e fidedigna das ações realizadas no programa. Além disso, ele é uma alternativa viável para integração entre as áreas envolvidas no esporte, garantindo aos jovens acadêmicos, atletas e técnicos, por meio da participação de atividades sistematizadas, a utilização do tempo de forma contributiva para o desenvolvimento de suas aptidões e o trabalho cooperativo.

AGRADECIMENTOS

Este trabalho teve apoio financeiro da Secretaria do Esporte e do Turismo do Estado do Paraná. Os autores também agradecem à CAPES e a UTFPR pelo apoio financeiro no período em que este trabalho foi redigido.

REFERÊNCIAS

ABITEBOUL, S.; HULL, R.; VIANU, V. **Foundations of databases: the logical level**. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1995.

ALBUQUERQUE, A. F. **Gestão estratégica das informações internas na pequena empresa**. Tese (Doutorado) — Universidade de São Paulo, 2004.

ANDRADE, L. H. de et al. **Deuzikachico: o poder da agi no monitoramento e combate de epidemias como a de dengue, zika e chikungunya**. *XII Simpósio Brasileiro de Sistemas de Informação*, p. 377–384, 2016.

ARANTES, N. **Sistemas de gestão empresarial**. São Paulo: Atlas, 1998.

- AVILA-SANTOS, A. P. et al. **Sistema integrado de custo municipal: uma ferramenta de apoio à tomada de decisão pelo gestor público.** *XI Simpósio Brasileiro de Sistemas de Informação*, p. 651–658, 2015.
- BASTOS, F. da C. **Administração esportiva: área de estudo, pesquisa e perspectivas no brasil.** *Motrivivência*, n. 20-21, p. 295–306, 2003.
- BOTELHO, M. A.; MONTEIRO, A. M.; VALLS, V. **A gestão do conhecimento esportivo: a experiência da biblioteca da seme.** *Ciência e Informação, Brasília*, SciELO Brasil, v. 36, n. 1, p. 175–188, 2007.
- BUSTO, R. et al. **Iniciação e aperfeiçoamento ao esporte paralímpico.** 2014.
- BUSTO, R. et al. **Apoio às diversas ações definidas no projeto talento olímpico do paraná-top 2016.** 2014.
- CARRAVETTA, E. **Modernização da gestão no futebol brasileiro.** [S.l.]: Editora AGE Ltda, 2006.
- CARVALHO, L. F. de et al. **Desenvolvimento de guia eletrônico na forma de aplicativo móvel: Uma abordagem para a semana santa de são joão del-rei-mg.** *XII Simpósio Brasileiro de Sistemas de Informação*, p. 361–368, 2016.
- KALIKOSKI, D. C. et al. **Gestão compartilhada e comunitária da pesca no brasil: avanços e desafios.** *Ambiente & Sociedade*, SciELO Brasil, v. 12, n. 1, p. 151–172, 2009.
- LAUDON, K. C.; LAUDON, J. P. **Sistemas de informação gerenciais: administrando a empresa digital.** Tradução Arlete Simille Marques. [S.l.]: São Paulo: Pearson, 2004.
- MATSUMOTO, C. Y. **A importância do banco de dados em uma organização.** *Maringá Management: Revista de Ciências Empresariais: on-line*, v. 3, n. 1, 2006.
- MOCSÁNYI, V.; BASTOS, F. da C. **Gestão de pessoas na administração esportiva: considerações sobre os principais processos.** *Revista Mackenzie de Educação Física e Esporte*, v. 4, n. 4, 2009.
- NOLASCO, V. P. et al. **Administração/gestão esportiva.** *Atlas do esporte no Brasil. Rio de Janeiro: CONFEF*, 2006.
- RAMOS, L. S.; OCAÑA, K. A.; OLIVEIRA, D. de. **Um sistema de informação para gerência de projetos científicos baseados em simulações computacionais.** *XII Simpósio Brasileiro de Sistemas de Informação*, p. 216–223, 2016.
- RITTER, E. W.; RIGO, S. J. **Fitdata: Um sistema para monitoramento de atividade física baseado em dispositivos móveis.** *XII Simpósio Brasileiro de Sistemas de Informação*, p. 550–557, 2016.
- ROCHA, C. M. da; BASTOS, F. da C. **Gestão do esporte: definindo a área.** *Revista Brasileira de Educação Física e Esporte*, v. 25, n. spe, p. 91–103, 2011.
- SUCUPIRA, M. V. T. **Características de utilização de software as a service: Um estudo sobre as operadoras de planos de saúde.** *Projetos e Dissertações em Sistemas de Informação e Gestão do Conhecimento*, v. 4, n. 1, 2015.
- TEIXEIRA, M. S. et al. **Sistema web para gerenciamento e acompanhamento de uso de medicação em ambientes de vivência assistida.** *XII Simpósio Brasileiro de Sistemas de Informação*, p. 278–284, 2016.
- XU, H. **The research sports management monitoring system based on wsn.** *International Journal of Multimedia and Ubiquitous Engineering*, v. 10, n. 12, p. 79–88, 2015.

SISTEMA DE APOIO À AVALIAÇÃO DE ATIVIDADES DE PROGRAMAÇÃO POR RECONHECIMENTO AUTOMÁTICO DE MODELOS DE SOLUÇÕES

Márcia Gonçalves de Oliveira

Centro de Referência em Formação e EaD (Cefor)
Instituto Federal do Espírito Santo (IFES)
Vitória – ES

Leonardo Leal Reblin

Universidade Federal do Espírito Santo (Ufes)
Vitória - ES

Elias Silva de Oliveira

Universidade Federal do Espírito Santo (Ufes)
Vitória – ES

RESUMO: O desenvolvimento de um programa de computador é um processo de resolução de problemas que resulta em várias possibilidades de soluções. Dessa forma, a avaliação de exercícios de programação demanda muito esforço do professor tanto na avaliação manual, quando analisam-se várias possibilidades de soluções, quanto na avaliação automática, quando vários modelos de soluções devem ser fornecidos como entradas. Com o objetivo de auxiliar professores na identificação de modelos de soluções a partir de programas desenvolvidos por alunos, este trabalho propõe um sistema baseado em *clustering* para reconhecimento automático de modelos de soluções e para mapeamento dessas soluções em escores atribuídos por professores. Os primeiros experimentos de aplicação desse sistema em

duas bases de programas desenvolvidos por estudantes de programação apresentaram resultados promissores.

PALAVRAS-CHAVE: *Clustering*, Modelos de soluções, Programação, Rubricas

ABSTRACT: Developing a computer program is a process of solving problems that results in several possibilities of solutions. Thus, the assessment of programming exercises demands a lot of efforts both in manual evaluation when several potential solutions are analyzed, as the automatic assessment, when many solutions should be provided as inputs. In order to assist teachers to identify solution models from programs developed by students, this paper proposes a clustering-based system to recognize solution models and to map them in scores assigned by teachers. The first application of this system experiments on two real bases of programs developed by programming students showed promising results.

KEYWORDS: Solution Models, Programming, Clustering, Rubrics.

1 | INTRODUÇÃO

O desenvolvimento de programas de computador como parte do processo de aprendizagem de programação é uma prática

que resulta em várias possibilidades de soluções para cada problema a ser resolvido. Dessa forma, o processo de avaliação de exercícios de programação demanda de professores um grande esforço de análise de diversas soluções para descobrir modelos de soluções, no caso de avaliações manuais; e de composição de modelos de soluções, no caso de avaliações automáticas que requerem entradas de gabaritos. Nesse último caso, o problema se amplia, uma vez que nem sempre o professor conhece antecipadamente todas as possíveis soluções para um exercício de programação.

Problemas similares ao problema de reconhecer modelos de soluções ou representações da diversidade a partir de um conjunto de soluções de exercícios são tratados em muitos trabalhos científicos como um processo de seleção de amostras representativas de um conjunto de padrões como, por exemplo, textos, imagens e pessoas.

Para resolver essa classe de problemas, têm-se desenvolvido, várias estratégias tecnológicas de amostragem seletiva (LINDENBAUM *et al*, 2004), de aprendizagem ativa (TUIA *et al*, 2011; OLIVEIRA *et al*, 2014) e de rubricas (KWON; JO, 2005) para apoiar docentes na avaliação manual e automática de exercícios. Essas tecnologias, em geral, são desenvolvidas para sistemas de aprendizagem supervisionada, isto é, sistemas que aprendem através de exemplos para tomarem decisões de forma autônoma (KOTSIANTIS, 2007).

Para o domínio da aprendizagem de programação, para reconhecer a variedade de soluções, recomenda-se a aplicação de *clustering* (NAUDE *et al*, 2010). Seguindo essa ideia, com o objetivo de auxiliar professores na avaliação de exercícios de programação, propomos um sistema de reconhecimento automático de possíveis soluções com mapeamento destas em escores atribuídos por professores.

O reconhecimento de modelos de soluções e da diversidade destas é realizado a partir de uma coleção de programas em Linguagem C desenvolvidos por estudantes para um exercício de programação. Já o mapeamento em escores consiste em solicitar ao professor a atribuição de escores para os modelos de soluções reconhecidos. Esses modelos de soluções são, desse modo, utilizados como gabaritos de entrada de um avaliador automático ou como treino de avaliadores de aprendizagem supervisionada.

Este trabalho foca-se na seleção de modelos de soluções para composição de gabaritos de exercícios de programação. No entanto, esses modelos podem também ser utilizados por professores para gerar exemplos representativos de padrões de desempenhos informando critérios de avaliações no esquema de rubricas (MERTLER, 2001).

As contribuições do sistema proposto para a aprendizagem de programação são agilizar o processo de avaliação identificando automaticamente os potenciais gabaritos para cada exercício de programação e oferecer aos estudantes maior clareza do processo avaliativo fornecendo-lhes uma ampla variedade de soluções explicadas a partir dos escores atribuídos por professores.

Este trabalho está organizado conforme a ordem a seguir. Na Seção 2,

apresentamos os trabalhos relacionados que oferecem mecanismos de seleção automática de modelos de soluções de exercícios. Na Seção 3, descrevemos a nossa proposta metodológica para reconhecimento de soluções e o algoritmo de *clustering* utilizado. Na Seção 4, apresentamos o primeiro experimento realizado em duas bases de soluções de exercícios desenvolvidas por estudantes de programação e os resultados alcançados. Na Seção 5, concluímos com as considerações finais e trabalhos futuros.

2 | TRABALHOS RELACIONADOS

A estratégia tecnológica deste trabalho foi desenvolvida como uma extensão do sistema *PCodigo* (OLIVEIRA *et al*, 2015b), visando reduzir esforço humano (OLIVEIRA *et al*, 2014) através da seleção de amostras de treino para sistemas de aprendizagem supervisionada (LINDENBAUM *et al*, 2004; TUIA *et al*, 2011; OLIVEIRA *et al*, 2015a) e da apresentação de critérios de avaliação no esquema de rubricas (KWON; JO, 2005).

O *PCodigo* é um sistema de apoio à prática assistida de programação que, integrado ao *Moodle*, recebe soluções de atividades de programação submetidas por alunos, executa-as em massa, realiza análises e emite relatórios de avaliação para professores como, por exemplo, relatórios de plágio (OLIVEIRA *et al*, 2015b).

A redução de esforço humano no fornecimento de exemplos para treino de sistemas de aprendizagem supervisionada tem sido uma temática abordada em muitos trabalhos de pesquisa. O trabalho de Oliveira *et al*. (2014), por exemplo, propõe uma estratégia de aprendizagem ativa que visa selecionar uma quantidade mínima de exemplos de textos do *Twitter* para um especialista classificá-los e, a partir desses exemplos, classificar automaticamente uma grande quantidade de outros *tweets*.

Uma estratégia mais recente, que é aplicada para selecionar amostras representativas de exercícios de programação, é o método de seleção não-aleatória de amostras de treino para classificação de perfis proposto por Oliveira *et al*. (2015a).

Nesse último método, visa-se reduzir o esforço humano na atribuição de escores aos exemplos selecionados e, da máquina, na classificação automática, uma vez que realiza-se redução de dimensionalidade da matriz de sessenta dimensões que representa soluções de exercícios em Linguagem C. Para a redução de dimensionalidade, utiliza-se a análise fatorial e, para a seleção de amostras que representem a diversidade de uma coleção de exercícios de programação, utiliza-se o *Clustering em Grafo* (OLIVEIRA *et al*. 2015a).

A estratégia de seleção de amostras representativas para treino de sistemas de aprendizagem supervisionada também pode ser uma opção para definir critérios de avaliação de um professor no esquema de rubricas. Seguindo essa ideia, o trabalho de Kwon e Jo (2005) aplica uma ferramenta que possibilita a análise das preferências

dos professores e as características dos alunos para cada rubrica, explorando a classificação e as regras de associação utilizando técnicas de *data mining*.

O diferencial da nossa estratégia em relação às estratégias apresentadas é que utilizamos processos de *clustering* ajustados pelo número de *clusters* e pelos índices de similaridade interna e externa desses *clusters* para selecionar modelos de soluções e representações da diversidade de uma coleção de exercícios.

3 I RECONHECIMENTO AUTOMÁTICO DE MODELOS DE SOLUÇÕES

O sistema de reconhecimento de soluções proposto neste trabalho é uma extensão do *PCodigo*, que é um sistema de apoio à prática assistida de programação através da execução em massa e análise de exercícios de programação em Linguagem C (OLIVEIRA *et al*, 2015b).

O processamento do *PCodigo* se inicia após o professor disponibilizar via *Moodle*, na *Visão do Professor, Exercícios de Programação* e após os alunos realizarem *Submissões* de soluções para esses exercícios (OLIVEIRA *et al*, 2015b).

Em resumo, as etapas de processamento do *PCodigo* consistem em receber soluções de atividades de programação submetidas por alunos, executá-las, analisá-las para avaliação de indícios de plágios e emitir relatórios de avaliação para professores.

A Figura 1 apresenta a arquitetura do sistema de análise de soluções bem como a sua integração ao *Pcodigo*.

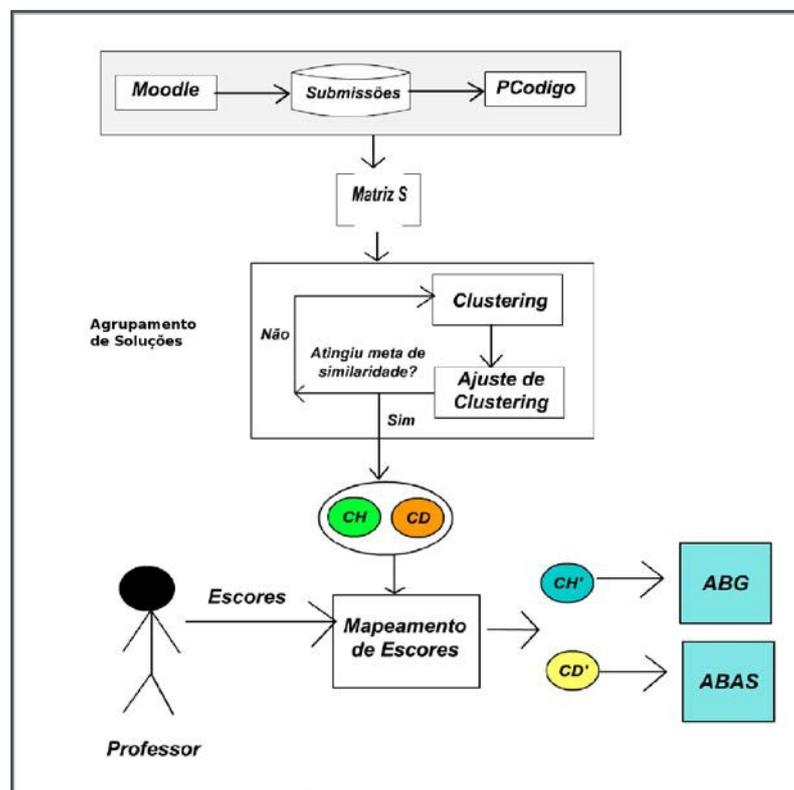


Figura 1. Sistema de análise de soluções

Conforme a Figura 1, o *Pcodigo* é integrado ao *Moodle* para receber as

Submissões de exercícios e fornece como entrada ao sistema uma *Matriz S* contendo as submissões vetorizadas em 60 dimensões. Nessa matriz, cada dimensão de um vetor representa a frequência de ocorrência de uma palavras-chave ou função da Linguagem C em código-fonte ou um valor lógico (1=Verdadeiro; 0= Falso) informando, por exemplo, se um programa compila e executa (OLIVEIRA *et al*, 2015b).

No processamento de *Agrupamento de Soluções* da Figura 1, a *Matriz S* é submetida ao algoritmo de *Clustering Bisecting K-means* do software *Cluto* (KARYPIS, 2003).

Sabendo que há mais possibilidades das melhores soluções de um exercício se assemelharem por aproximarem-se de um padrão de resolução ensinado por um professor (NAUDE *et al*, 2010), para obter um agrupamento com os melhores modelos de soluções, realizamos o *Ajuste de Clustering*. Para isso, repetimos os processos de *Formação de Agrupamentos* até que, pelo número de *clusters* escolhido, sejam obtidos *clusters* com maiores índices de similaridade interna (*ISIM*) e de similaridade externa (*ESIM*).

Após o processamento de *Agrupamento de Soluções*, são selecionados dois *clusters*: os *clusters CH* e *CD*. O *cluster CH* representa o agrupamento com as melhores soluções e o *cluster CD*, o agrupamento com maior representação da diversidade.

No módulo de *Mapeamento de Escores*, os *clusters CH* e *CD* são apresentados a um professor para que este atribua escores para cada um dos padrões desses *clusters*. Em seguida, os *clusters CH'* e *CD'* contendo padrões avaliados pelo professor são enviados como entradas para os algoritmos de avaliação automática de exercícios *ABG* (*Avaliação Baseada em Gabaritos*) e *ABAS* (*Avaliação Baseada em Aprendizagem Supervisionada*).

Os padrões de *CH'* enviados para o algoritmo de *ABG* podem ser utilizados como modelos de soluções para atribuição de escores a outras soluções muito semelhantes a estas. Já os padrões do *CD'* enviados para o algoritmo de *ABAS*, em um passo mais avançado, representando a diversidade de um conjunto de soluções composto de padrões com altos, baixos e médios escores, podem ser utilizados como treino do avaliador automático para que este aprenda o padrão de correção de um exercício e realize previsões de escores para outros exemplos de soluções desse mesmo exercício.

Uma outra aplicação da metodologia deste trabalho é utilizar os padrões avaliados dos *clusters CH'* e *CD'* como referências de avaliação para diferentes níveis de desempenhos em um esquema de rubricas. Nesse caso, os professores apresentariam de forma mais clara seus critérios de avaliação de cada exercício de programação aplicado, dando possibilidades ao aluno de refletir sobre sua própria aprendizagem.

4 | EXPERIMENTOS E RESULTADOS

Para a experimentação do sistema de reconhecimento de soluções, utilizamos

duas bases de soluções de um exercício de programação obtidas em turmas reais de programação de uma universidade e também utilizadas nos experimentos de Oliveira *et al.* (2015a).

A primeira base, que chamaremos de *Base-A*, reúne 100 amostras de soluções de um exercício coletadas em diferentes turmas de programação. Já a segunda base, que chamaremos de *Base-B*, é uma base formada de 40 amostras de soluções do mesmo exercício utilizado na *Base-A*. No entanto, essa base é considerada mais difícil porque foi obtida em condições controladas de prova. Desse modo, essa base apresenta uma maior diversidade de soluções e menor possibilidade de conter amostras plagiadas.

A *Base-A* e a *Base-B* foram mapeadas em vetores de 60 dimensões ou componentes de habilidades, onde cada dimensão, conforme já informamos, é quantificada pela frequência de ocorrência de palavras-chave, funções e indicadores de funcionamento (*compila*, *executa*) da Linguagem C (OLIVEIRA *et al.* 2015b; OLIVEIRA *et al.* 2015a). Essas bases compostas de amostras em formato vetorial são geradas pelo *PCódigo* nos processos de normalização e indexação (OLIVEIRA *et al.*, 2015b).

No sistema de reconhecimento de soluções, essas bases são submetidas no formato da *Matriz S* (Figura 1) ao algoritmo de *clustering Bisecting K-means* (KARYPIS, 2003). O *Ajuste de Clustering* é realizado até que se alcance o melhor número de *clusters* com maiores valores de similaridade interna e externa (*ISIM* e *ESIM*, respectivamente).

Após escolher o número de *clusters* mais adequado, selecionamos o *cluster* mais provável de ter as melhores soluções (*CH*) e o *cluster* com a maior representação da diversidade de soluções (*CD*).

Para análise dos *clusters* formados, um *script* gera um relatório com a soma das dimensões de cada vetor, o que nos fornece um indicativo de esforço de programação do estudante. Em geral, as melhores soluções apresentam menos instruções e menos linhas de código, o que pode ser observado nos relatórios das figuras 2 e 3.

4.1 Resultados

Ao aumentarmos o número de *clusters* no processamento de *Ajuste de Clustering*, alcançamos valores maiores de *ISIM* e de *ESIM* para os *clusters* formados. Isso nos possibilita obter *clusters* mais homogêneos e identificar soluções atípicas, isto é, soluções que são únicas em um *cluster*.

A Tabela 1, relativa à *Base-A* de 100 amostras, apresenta como os valores de *ISIM* máximo e mínimo aumentam e o *ESIM* mínimo diminui, indicando que, aumentando o número de *clusters*, formam-se *clusters* mais homogêneos e distintos de outros *clusters*.

Análise de Similaridade ISIM-ESIM

No. Clusters	ISim Máximo (%)	ISim Mínimo (%)	ESim Máximo (%)	ESim Mínimo (%)	(%)
1	69.8	69.8	100	100	

2	80.8	76	59.5	59.5	
3	87.6	76.1	67.9	59.5	
4	87.6	72.2	67.9	49.2	
5	92.6	72.2	70.2	49.2	
6	92.6	72.2	70.5	49.2	
7	92.6	72.2	70.5	49.2	
8	92.6	83.8	70.5	42.0	
9	92.6	79.1	72.6	42.0	
10	100	85.0	72.6	30.5	

Tabela 1. Análise de similaridade por número de clusters da Base-A

Para termos *clusters* mais homogêneos, conforme a Tabela 1, escolhemos dividir a *Base-A* em dez *clusters*. Na Tabela 2, apresentamos a organização desses *clusters* conforme os valores de *ISIM* e *ESIM*.

Análise de similaridade dos clusters

Cluster	Tamanho	ISim (%)	ESim (%)
0	1	100	30.5
1	2	85.4	42.0
2	4	86.8	53.1
3	5	92.2	63.4
4	21	85.2	61.8
5	22	92.6	70.3
6	8	89.9	67.7
7	10	89.1	70.2
8	14	91.3	72.6
9	13	85.0	70.5

Tabela 2. Análise de similaridade dos clusters da Base-A

Uma vez que os *clusters* são organizados de forma crescente de 0 a $N - 1$ ($N =$ Número de *Clusters*), os maiores valores de *ISIM-ESIM* estão nos *clusters* identificados com valores maiores. Da mesma forma, os *clusters* mais apertados e mais distantes dos outros, isto é, com menor *ESIM*, são identificados com valores menores (KARYPIS, 2003).

A Figura 2 apresenta os *clusters* 0, 1, 2, 3, 4 e 9 da *Base-A*. Nessas tabelas, *cluster* é o identificador do *cluster*, índice é o rótulo de cada amostra dentro dos *clusters*, *nota* é o escore atribuído pelo professor a uma solução e *somalinha* é a soma dos valores dos atributos, isto é, das dimensões de um vetor representando uma solução desenvolvida por um aluno. Os escores altos são marcados de verde, os médios, de amarelo e os escores baixos, de lilás. Observa-se que o *Cluster 4* apresenta uma maior diversidade de escores médios e altos. Por outro lado, o *Cluster 9* apresenta-se mais homogêneo com predominância das soluções de altos escores.

Para a *Base-A*, concluímos que o *Cluster 9* é o mais indicado para ser o *cluster CH*, isto é, o *cluster* homogêneo com mais possibilidades de representar as soluções de altos escores, isto é, os modelos de soluções.

Já o *Cluster 4* é o mais indicado para ser o *cluster CD*, uma vez que apresenta maior variedade de soluções com escores altos e médios, que são os escores predominantes na *Base-A*.

Desconhecendo os escores das amostras reunidas nos *clusters*, identificamos o *cluster CH* pelo maior valor de identificação do *cluster*, isto é, pelo maior valor de *ISIM-ESIM*. Já, para identificar o *cluster CD*, selecionamos o *cluster* mais cheio com menor *ISIM-ESIM*.

Cluster	Índice	Nota	Somalinha
0	al21	3.0	7
1	al89	4.1	10
1	al60	3.75	6
2	al79	2.5	128
2	al61	4.8	67
2	al48	3.0	151
2	al64	2.0	84

Cluster	Índice	Nota	Somalinha
4	al77	3.7	39
4	al96	5.0	51
4	al52	3.25	52
4	al78	0.5	99
4	al43	5.0	57
4	al38	5.0	57
4	al53	4.75	88
4	al100	4.0	78
4	al49	4.0	51
4	al46	5.0	52
4	al25	3.25	70
4	al28	3.75	99
4	al74	5.0	56
4	al47	3.5	33
4	al72	4.0	80
4	al1	4.25	75
4	al22	5.0	51

Cluster	Índice	Nota	Somalinha
3	al12	3.75	66
3	al37	3.25	45
3	al8	4.0	37
3	al73	3.6	63
3	al19	3.5	79

Cluster	Índice	Nota	Somalinha
9	al39	4.5	46
9	al26	5.0	50
9	al34	4.75	56
9	al2	4.0	84
9	al83	4.3	44
9	al50	4.5	39
9	al62	2.5	37
9	al32	4.75	79
9	al56	1.5	113
9	al95	3.7	99
9	al69	4.75	75
9	al98	1.2	41
9	al15	3.75	104

Figura 2. Análise de clusters da Base-A

A Tabela 3 apresenta como os valores de *ISIM* e *ESIM* dos *clusters* da *Base-B* variam à medida que o número de *clusters* aumenta. Para uma divisão de 10 *clusters* temos, portanto, maior *ISIM* máximo, maior *ISIM* mínimo, maior *ESIM* máximo e menor *ESIM* mínimo. Isso significa que a divisão em 10 *clusters* da *Base-A* oferece maior possibilidade de formar *clusters* mais homogêneos e bem distintos dos demais *clusters*.

Análise de Similaridade ISIM-ESIM

No. de clusters	ISim Máximo (%)	ISim Mínimo (%)	ESim Máximo (%)	ESim Mínimo (%)
1	82.8	82.8	--	--
2	90.0	78.9	73.0	73.0
3	94.7	78.9	79.4	73.0
4	100	85.1	79.4	51.7
5	100	87.3	79.4	51.7

6	100	87.3	79.7	51.7
7	100	91.6	79.7	51.7
8	100	91.6	81.8	51.7
9	100	92.9	81.8	51.7
10	100	95.0	82.8	51.7

Tabela 3. Análise de similaridade por número de clusters da Base-B

Da mesma forma que analisamos os *clusters* da *Base-A*, analisamos a *Base-B* identificando os *clusters CH* e *CD*. Mas vale destacar, conforme a Figura 3, que os *clusters* da *Base-B* apresentaram-se mais homogêneos e com as soluções inéditas isoladas nos primeiros *clusters*.

Cluster	Índice	Nota	Somalinha
0	al28	3.75	99
1	al3	3.25	101
2	al79	2.5	128
3	al22	5.0	51
4	al36	3.0	50
4	al84	3.0	82
4	al13	3.75	45
5	al87	3.2	80
5	al91	3.2	95
5	al80	3.7	98
5	al35	1.75	89
5	al63	1.5	108
6	al44	3.75	83
6	al82	2.7	99
7	al32	4.75	79
7	al33	4.5	55
7	al23	3.0	98
8	al2	4.0	84
8	al69	4.75	75
8	al40	3.5	133
8	al50	4.5	39
8	al20	3.5	88
8	al97	2.7	91
8	al26	5.0	50
8	al7	2.0	82
8	al67	3.75	78
8	al76	3.5	109
8	al94	2.7	88
8	al81	3.2	94
8	al14	3.25	112
8	al71	4.2	90
8	al9	2.75	85
8	al39	4.5	46
8	al60	3.75	6
8	al19	3.5	79
8	al29	3.5	93
9	al30	3.5	86
9	al92	5.0	46
9	al31	3.25	105

Figura 3. Análise de clusters da Base-B

Para a *Base-B*, selecionamos, conforme a Figura 3, o *Cluster 9* como o *cluster CH* e o *Cluster 8*, como o *cluster CD*, pelos mesmos critérios de seleção dos *clusters CH* e *CD* da *Base-A*.

Concluindo, os resultados da *Base-A* e da *Base-B* indicam que é possível selecionar modelos de soluções e amostras representando a diversidade de um conjunto de exercícios através de processos de *clustering*. No entanto, para gerar os conjuntos *CH'* e *CD'* com as amostras de *CH* e *CD*, respectivamente, pontuadas por um professor, demanda-se ainda esforço de correção de todas as amostras dos *clusters CH* e *CD*.

Para reduzir o esforço do professor de corrigir todas as amostras desses *clusters*, propomos, como próximos trabalhos, desenvolver estratégias para reduzir os conjuntos *CH'* e *CD'* através da seleção das amostras mais representativas dos modelos de soluções e da diversidade de exercícios presentes nos *clusters CH* e *CD*,

respectivamente.

5 | CONSIDERAÇÕES FINAIS

Este trabalho de pesquisa apresentou uma proposta inicial de sistema de seleção de modelos de soluções e da diversidade de soluções de exercícios de programação. Uma vantagem desse sistema é que modelos de soluções reconhecidos podem ser utilizados como entradas de sistemas baseados em gabaritos e as soluções de representação da diversidade como treino de sistemas de avaliação automática de aprendizagem supervisionada.

Uma outra vantagem do sistema proposto é auxiliar professores na identificação de modelos de soluções de um exercício para apresentação de critérios de avaliação no esquema de rubricas.

A principal limitação do sistema proposto é determinar o número mínimo de *clusters* que possibilita a melhor formação de agrupamentos dos modelos de soluções e *clusters* mais reduzidos.

Como trabalhos futuros a partir deste, propomos desenvolver pesquisas para determinar esse menor número máximo de *clusters* e realizar transformações nas bases de exercícios com técnicas de redução de dimensionalidade como a análise fatorial (OLIVEIRA et al, 2015a) de forma a melhorar o processo de *clustering*.

Com essas ideias de trabalhos futuros, poderíamos formar *clusters* menores reunindo os modelos de soluções e as amostras de representação da diversidade, o que implicaria em menor esforço de correção dos professores e mais precisão na escolha dos modelos de soluções.

Concluindo, as principais contribuições do sistema proposto para a aprendizagem de programação é agilizar o processo de envio de *feedbacks* de professores, principalmente se estes utilizarem sistemas de avaliação automática, e possibilitar esclarecer critérios de avaliações a partir de exemplos corrigidos dinamicamente.

REFERÊNCIAS

KARYPIS, George. **CLUTO-a clustering toolkit**. Minnesota Univ MINNEAPOLIS Dept of Computer Science, 2002.

KOTSIANTIS, Sotiris B.; ZAHARAKIS, I.; PINTELAS, P. **Supervised machine learning: A review of classification techniques**. Emerging artificial intelligence applications in computer engineering, v. 160, p. 3-24, 2007.

KWON, Hyeong-Gyu; JO, Mi-Heon; LEE, Eun-Jeong. **Design and implementation of the automatic rubric generation system for the neis based performance assessment using data mining technology**. *Journal Of the Korean Association of information Education*, v. 9, p. 113–126, 2005.

LINDENBAUM, Michael; MARKOVITCH, Shaul; RUSAKOV, Dmitry. **Selective sampling for nearest neighbor classifiers**. *Machine learning*, v. 54, n. 2, p. 125-152, 2004.

MERTLER, Craig A. **Designing scoring rubrics for your classroom**. Practical Assessment, Research & Evaluation, v. 7, n. 25, p. 1-10, 2001.

NAUDÉ, Kevin A.; GREYLING, Jean H.; VOGTS, Dieter. **Marking student programs using graph similarity**. Computers & Education, v. 54, n. 2, p. 545-561, 2010.

OLIVEIRA, Elias et al. **Combining clustering and classification approaches for reducing the effort of automatic tweets classification**. In: Proceedings of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management-Volume 1. SCITEPRESS-Science and Technology Publications, Lda, 2014. p. 465-472.

OLIVEIRA, Marcia et al. **Representação da diversidade de componentes latentes em exercícios de programação para classificação de perfis**. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2015. p. 1177.

OLIVEIRA, M. G.; NOGUEIRA, Matheus de Araújo; OLIVEIRA, Elias. **Sistema de apoio à prática assistida de programação por execução em massa e análise de programas**. In: XXIII Workshop sobre Educação em Computação (WEI)-CSBC. 2015.

TUIA, Devis; PASOLLI, E.; EMERY, William J. **Using active learning to adapt remote sensing image classifiers**. Remote Sensing of Environment, v. 115, n. 9, p. 2232-2242, 2011.

RASTREAMENTO DE OBJETOS EM VÍDEO COM APLICAÇÕES PRÁTICAS

Karla Melissa dos Santos Leandro

Instituto de Matemática e Tecnologia
Universidade Federal de Goiás
karlamelissaleandro@gmail.com

Sérgio Francisco da Silva

Instituto de Biotecnologia
Universidade Federal de Goiás
sergio@ufg.com

Marcos Napoleão Rabelo

Instituto de Matemática e Tecnologia
Universidade Federal de Goiás
rabelo@dmate.ufpe.br

RESUMO: Rastreamento de objetos em vídeo é fundamental no desenvolvimento de sistemas de visão computacional. Com foco em câmera estática, propomos uma metodologia para rastreamento de múltiplos objetos em vídeos usando subtração de plano de fundo via mistura de gaussianas, morfologia matemática e o filtro de Kalman. Após a subtração de plano de fundo aplicamos morfologia matemática para eliminação de ruídos e preenchimento de buracos nos objetos detectados. Então, aplicamos o filtro de Kalman para prever a posição atual dos objetos, com base em sua predição anterior e na evidência atual calculada com base na subtração de plano de fundo. Nossos resultados mostram que a predição do

centroide dos objetos feita com o uso do filtro de Kalman é mais precisa do que quando feita usando somente a subtração de plano de fundo. **PALAVRAS-CHAVE:** Subtração de Plano de fundo, Mistura de Gaussiana, Filtro de Kalman.

1 | INTRODUÇÃO

A detecção e o rastreamento de objetos em vídeo é uma tarefa primordial para muitos sistemas de visão computacional, tais como sistemas biométricos, identificação de atividades e em sistemas de monitoramento em geral. Contudo, para muitas aplicações, o processo de detecção e rastreamento não é trivial devido a fatores como ruídos, baixa resolução de câmera, mudanças de iluminação na cena, oclusão parcial ou total de objetos, entre outros. Assim, é de extrema importância a elaboração de métodos robustos a tais fenômenos.

Este trabalho foca no rastreamento de objetos a partir de câmera estática, tal como em sistemas de vigilância. Como consideramos câmera estática, aplicamos inicialmente um método de subtração de plano de fundo para a detecção dos objetos em movimento. Como lidamos com cenas onde há mudanças de iluminação, optamos pelo uso do método de mistura de gaussianas [1], que é bastante

robusto a tal fenômeno. Contudo, por conta de ruídos e baixo contraste de objetos com relação ao plano fundo, o resultado da subtração de plano de fundo apresenta falsos positivos e, em alguns casos, ocorrem buracos em partes dos objetos. Para tratar estes problemas é aplicado a operação morfológicas denominada fechamento [2], além da remoção de objetos pequenos (menores que um dado limiar). Mesmo assim não é possível rastrear satisfatoriamente os objetos uma vez que a segmentação dos objetos não é totalmente precisa, fazendo com que a delimitação dos objetos fique um pouco deslocada em alguns frames da cena, e por consequência, a trajetória capturada apresenta erros significativos em relação a trajetória exata dos objetos. Assim, usamos o filtro de Kalman [3] para filtrar a evidência dada pela subtração de plano de fundo juntamente com as operações morfológicas aplicadas posteriormente. Os resultados são avaliados através da medida de erro absoluto entre o centroide dos objetos preditos e os centroides reais dos objetos, dados pelo ground-truth. Nossos experimentos revelam que a filtragem de Kalman corrige a posição exata dos centroides dos objetos. Além disso, o erro entre o resultado da metodologia proposta em relação ao ground-truth é representado na Figura 2 onde é possível identificar o quanto os valores preditos pelo Filtro de Kalman se distanciou dos valores reais.

2 | CONCEITOS BÁSICOS

Esta seção apresenta os conceitos básicos acerca de das técnicas empregadas na metodologia proposta. As técnicas empregadas são: 1) subtração de plano de fundo por mistura de gaussianas; 2) operações morfológicas para eliminação de ruídos e preenchimento de buracos e o filtro de Kalman.

2.1 Mistura de Gaussianas

Há duas abordagens básicas para a detecção de objetos em movimento em vídeos estáticos: fluxo óptico (*optical flow*) e subtração de plano de fundo (*background subtraction*). Resumidamente, técnicas de fluxo óptico calculam vetores velocidade dos objetos em movimento.

Estes vetores velocidade, uma vez computados, podem ser usados para uma ampla variedade de tarefas indo de navegação até exploração de ambientes por agentes autônomos. Métodos de subtração de plano de fundo [6] estimam e mantêm um modelo de plano de fundo (*background*) que é subtraído do frame atual para obter o *foreground* (segmentação do(s) objeto(s) em movimento). Esta pesquisa foca em métodos de subtração de plano de fundo pois estes métodos permitem delimitar as fronteiras dos objetos que é uma característica fundamental para a identificação/reconhecimentos destes. Um dos métodos de subtração de plano de fundo de maior sucesso é a mistura de gaussianas.

Mistura de gaussianas é um método probabilístico paramétrico, proposto por [1], posteriormente melhorado por [5]. Atualmente Mistura de Gaussianas é um dos métodos de subtração de plano de fundo mais populares, principalmente devido a sua robustez com relação a mudanças de iluminação na cena. Neste método as distribuições para a cor de cada pixel são representadas por uma soma de distribuições de gaussianas ponderadas. A cada novo frame no tempo t , os parâmetros para todos os pixels são atualizados para determinar as variações de cores. De fato, no tempo t , é considerado que o modelo m_t , gerado para cada pixel a partir das medidas $\{z_0, z_1, \dots, z_{t-1}\}$ de um pixel, é correto. A probabilidade que um determinado pixel seja pertencente ao plano de fundo, é dada por:

$$P(z_t|m_t) = \sum_{n=1}^N \frac{\alpha_n}{(2\pi)^{d/2} |\Sigma_n|^{1/2}} e^{-1/2(z_t-\mu_t)^T \Sigma_n^{-1}(z_t-\mu_t)} \quad (1)$$

onde d é a dimensão do espaço de cor das medidas z_t e cada gaussiana n é descrita pela sua Média μ_n e matriz de covariância Σ_n . As gaussianas são ponderadas pelos fatores α_n onde $|\Sigma_n|$ denota a matriz determinante. Os canais (R,G,B) de cada pixel são considerados independentes. Para atualizar o modelo de mistura de gaussianas, somente agrega-se tais medidas a uma gaussiana n se:

$$\|z_t - \mu_n\| < k\sigma_n \quad (2)$$

onde k é 2 ou 3, e σ_n é um vetor representando a variância da distribuição gaussiana de índice n . Posteriormente, as gaussianas são atualizadas a cada novo frame conforme um coeficiente de aprendizado.

3 | OPERAÇÕES MORFOLÓGICAS

Operações morfológicas ou morfologia matemática consiste em ferramentas com variados propósitos em processamento de imagens, indo de pré-processamento e pós-processamento até ferramentas para descrição e caracterização de objetos através de forma, esqueletos, entre outras [2]. Neste trabalho usamos a operação morfológica denominada fechamento para corrigir pequenas falhas no resultado de subtração de plano de fundo. Como o resultado de subtração de plano de fundo é uma imagem binária, nós descrevemos a operação de fechamento para imagens binárias, embora este conceito possa ser estendido para imagens em níveis de cinza.

Conforme o nome, a operação de fechamento serve ao propósito de eliminar

pequenos buracos no objeto e preencher fendas no contorno. Na operação de fechamento é considerado um elemento estruturante que denominaremos de B . Assim o fechamento de uma imagem A , conforme o elemento estruturante B e dada por uma dilatação de A por B , seguida da erosão do resultado por B . Matematicamente a operação de fechamento é representada do seguinte modo:

$$A \bullet B = (A \oplus B) \ominus B \quad (3)$$

onde \oplus representa a operação de dilatação e \ominus representa a operação de erosão. A dilatação de A por B , denotada por $(A \oplus B)$ é definida como:

$$(A \oplus B) = \{z | (B)_z \cap A \neq \emptyset\} \quad (4)$$

O processo de dilatação começa pela obtenção da reflexão de B em torno de sua origem, seguido da translação dessa reflexão por z . A dilatação de A por B é então o conjunto de todos os deslocamentos z , tal que B e A sobreponham em pelo menos um elemento. Já a erosão A por B ($A \ominus B$), consiste no conjunto de todos os pontos z tais que B , quando transladado por z fique contido em A , isto é:

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (5)$$

3.1 Filtro de Kalman

Filtragem, no contexto de raciocínio probabilístico temporal, consiste em estimar variáveis de estado, tais como posição, velocidade, entre outras, de observações ruidosas ao longo do tempo. Para dar um exemplo claro de sua aplicação, nós repetimos as palavras de Russell e Norvig [3]: “Imagine observar um pássaro pequeno voar em uma floresta de folhagem densa na penumbra: você pisca brevemente e vê intermitentes *flashes* de movimento; você se esforça para adivinhar onde o pássaro está e ele irá aparecer de forma que você não perca ele”. Quando você tenta seguir a trajetória de voo do pássaro você está fazendo filtragem.

O filtro de Kalman é usado para estimar o estado de um sistema linear onde assume que o estado segue uma distribuição Gaussiana. O filtro de Kalman consiste de dois passos: predição e correção (também conhecida como passo de atualização). O primeiro passo usa as variáveis do estado anterior para prever as variáveis do estado corrente e a incerteza destas. O segundo passo usa a medição corrente, ou seja, os valores atuais captados pelos sensores, para corrigir a estimativa do estado corrente. Desta forma, o filtro de Kalman é um processo online onde a nova medição

(também chamada de observação) é processada assim que ele é recebida.

Vamos primeiro definir o modelo temporal geral usado pelo filtro de Kalman. Assume-se que ambos, o modelo de transição e o modelo de sensor seguem uma transformação linear com ruído gaussiano aditivo. Assim, temos que a probabilidade de um novo estado x_{t+1} dado o estado x_t e a probabilidade de uma dada observação z_t dado um estado x_t são dadas por:

$$P(x_{t+1}|x_t) = N(Fx_t, \Sigma_x)(x_{t+1}) \quad (6)$$

$$P(z_t|x_t) = N(Hx_t, \Sigma_z)(z_t) \quad (7)$$

onde F e Σ_x são matrizes que descrevem o modelo de transição linear e a covariância do ruído de transição, e H e Σ_z são as matrizes correspondentes ao modelo de sensor. As equações atualização para a média e covariância, na forma completa, são:

$$\mu_{t+1} = F\mu_t + K_{t+1}(z_{t+1} - HF\mu_t) \quad (8)$$

$$\Sigma_{t+1} = (I - K_{t+1}H)(F\Sigma_t F + \Sigma_x), \quad (9)$$

onde $k_{t+1} = (F\Sigma_t F^T + \Sigma_x)H^T (H(F\Sigma_t F^T + \Sigma_x)H^T + \Sigma_z)^{-1}$ é chamado de matriz de ganho de Kalman. Para melhor compreendermos estas equações, considere a atualização da estimativa de estado médio μ . O termo $F\mu_t$ é o estado predito em $t+1$, de forma que $HF\mu_t$ é a observação predita. Contudo, o termo $z_{t+1} - HF\mu_t$ representa o erro na observação predita. Esta é multiplicada por K_{t+1} para corrigir o estado predito; conseqüentemente, K_{t+1} corresponde à proporção em que confiamos na nova observação com relação à predição.

4 | METODOLOGIA PROPOSTA

A metodologia proposta para o rastreamento de objetos a partir de vídeos capturados por câmaras estáticas é ilustrada pela Figura 1. Inicialmente os objetos em movimento são extraídos de cada frame do vídeo usando subtração de plano de fundo por mistura de gaussianas. Como os objetos extraídos normalmente apresentam buracos e fendas em seu contorno devido à similaridade de regiões do objeto com o plano de fundo da cena, aplicamos a operação morfológica de fechamento na tentativa de corrigir tais irregularidades. Em seguida, calculamos o centroide de cada objeto, os quais são fornecidos como evidência das posições dos objetos para o filtro de Kalman. No decorrer do tempo, ou seja, à medida que é recebido um novo frame do vídeo, o filtro de Kalman corrige a posição do centroide do objeto com base na evidência atual, dada pelo centroide obtido após a subtração de plano de fundo, e na predição do filtro de Kalman conforme os centroides calculados para os frames anteriores.

5 | RESULTADOS EXPERIMENTAIS

Nesta pesquisa usamos a base de dados Background Models Challenge (BMC) [4]. A

base de dados BMC consiste de vídeos sintéticos contendo uma grande variedade de cenários, variação de iluminação, variados números de objetos com sobreposição, além de cenas com adição de ruído, neblina e vento. Outro motivo para o uso desta base e o fato desta conter informação de ground-truth, consistindo da delimitação do contorno dos objetos em cada frame o que permite estimar a precisão do método de rastreamento de objetos. Nos experimentos separamos a base de dados BMC em seis sub-bases de dados, conforme a trajetória dos veículos (retilínea e curva) e com as características físicas no cenário (normal, neblina e vento). A Figura 1 ilustra frames das sub-bases de dados consideradas.

Na configuração da metodologia proposta usamos os seguintes parâmetros: no método de subtração de plano de fundo por mistura de gaussianas usamos três gaussianas, quarenta frames de treinamento para inferência do plano de fundo inicial e taxa de aprendizagem de 0.001; na operação de fechamento usamos um quadrado com 15 pixels de lado como elemento estruturante.

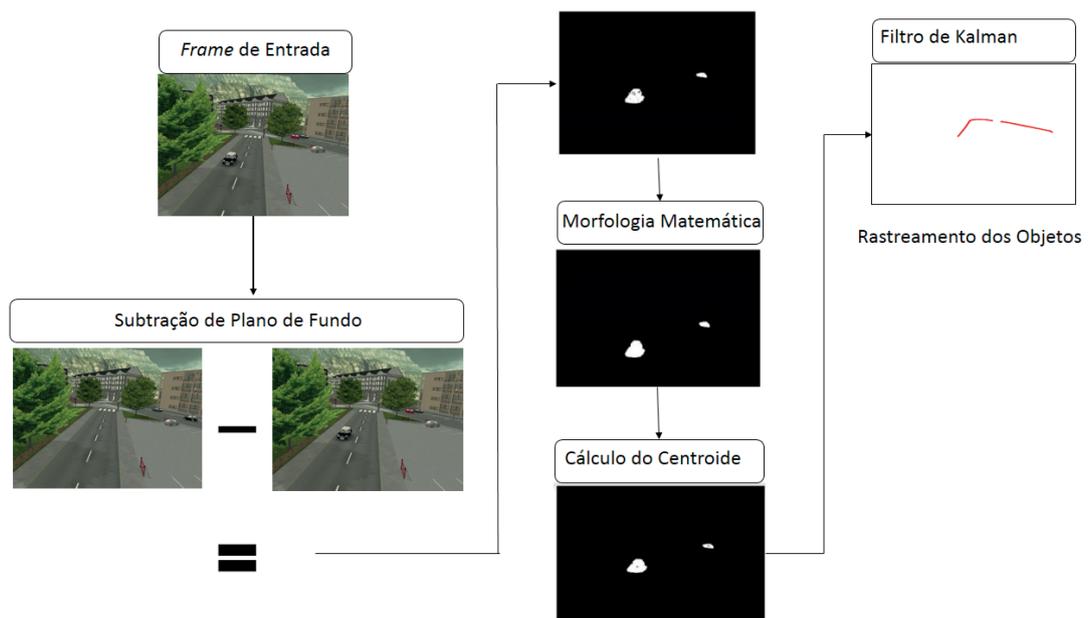


Figura 1 – Metodologia usada para rastreamento de objetos em movimento em vídeos capturados por câmera estática.

Fonte: Elaborada pelos Autores

6 | CONCLUSÃO

Ao longo do tempo foi observado que os valores correspondentes ao Filtro de Kalman se tornaram melhores, como o que acontece na Figura 2, pode ser visto que conforme o método vai sendo testado ele vai se adaptando ao cenário e torna-se

satisfatório os valores em relação a posição real do objeto, estes valores representados no gráfico corresponde a distância média dos valores preditos pelo Filtro de Kalman e valores detectados pelo método de Subtração de Plano de Fundo utilizando Mistura de Gaussianas. Desta forma é possível utilizar o Filtro de Kalman para acompanhar objetos ao longo de uma trajetória, isso é caracterizado quando na Figura 2 onde no começo do teste ele alcança valores altos e posteriormente conquista valores mais baixos em relação a Mistura de Gaussianas.

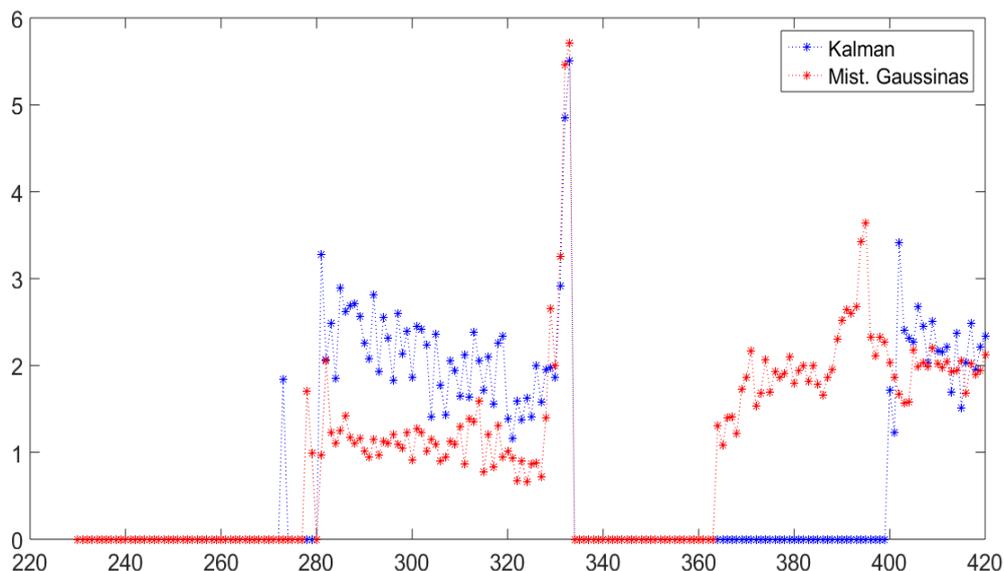


Figura 2 – Resultados referente ao rastreamento de objetos em movimento em vídeos capturados por câmara estática.

Fonte: Elaborada pelos Autores

REFERÊNCIAS

- C. Stauffer and W. E. L. Grimson IEEE Transactions on Pattern Analysis and Machine Intelligence **Learning patterns of activity using real-time tracking** 2000. 22. 8. 747- 757
- Gonzalez, Rafael C. and Woods, Richard. **Digital Image Processing** 2006. 54. 961-978.
- Russell, Stuart and Norvig, Peter. **Artificial Intelligence: A Modern Approach** 2009. 3. 8. 747-757
- Vacavant, Antoine and Chateau, Thierry and Wilhelm, Alexis and Lequi Laurent. Park, Jong-I and Kim, Junmo **A Benchmark Dataset for Outdoor Foreground/Background Extraction** 2013. Springer Berlin Heidelberg. 291–300
- Hayman, E. and Eklundh, J.-O. Proceedings of the Ninth IEEE International Conference on Computer Vision **Statistical background subtraction for a mobile observer** 2003.67- 74
- Bouwman, T. Computer Science Review. **Traditional and recent approaches in back- ground modeling for foreground detection: An overview** 2014. 11. 12. 31–66

PROPOSTA DE ESTRATÉGIA DE MATURIDADE E PRIORIZAÇÃO PARA SEGURANÇA DA INFORMAÇÃO BASEADA NA ISO/IEC 27001 E 27002 ADERENTE AOS PRINCÍPIOS DA GOVERNANÇA ÁGIL

Gliner Dias Alencar

Instituto Brasileiro de Geografia e Estatística
(IBGE)

Centro de Informática (CIn), Universidade Federal
de Pernambuco (UFPE)

Recife – PE

Hermano Perrelli de Moura

Centro de Informática (CIn), Universidade Federal
de Pernambuco (UFPE)

Recife – PE

RESUMO: A adoção de um modelo para gestão da segurança da informação, implementação de políticas e adequação a alguma norma de segurança da informação não é algo simples, conseqüentemente, tem-se dificuldades em sua implantação devido, muitas vezes, a complexidade das normas. Esses desafios demonstram a necessidade de mais investigações que abordem o problema. Para isso, este trabalho propõem uma estratégia de maturidade e priorização para a segurança da informação como base nos princípios expostos nas normas ISO/IEC 27001 e 27002 e na Governança Ágil. A pesquisa proposta realizou uma revisão sistemática da literatura e questionários para levantamento da situação atual da área de Segurança da Informação nas empresas e dos principais controles necessários, alcançando 157 empresas

distintas. Como resultado, foi possível classificar os controles ISO / IEC 27001 e 27002 em quatro estágios, de acordo com a importância dada pelas empresas. Também foram utilizados os níveis de maturidade do COBIT e uma matriz de análise de risco. Finalmente, a proposta foi testada com sucesso em uma empresa.

PALAVRAS-CHAVE: Segurança da Informação, Modelo de Maturidade, Governança de TIC, Governança Ágil.

ABSTRACT: The adoption of a model for information security management, along with the implementation of its policies and the required adjustments to some of its norms are not simple tasks. Therefore, the implementation of a model for information security management often implies in difficulties due to the complexity of the norms. Those challenges demonstrate a need for further investigations which address the problem. To achieve this goal, this work proposes a strategy to measure the maturity and prioritization of information security based on the principles exposed on the ISO/IEC 27001 and 27002 Standards and Agile Governance. The proposed research realized a systematic review of the literature and surveys regarding the current situation of information security in the industry and the main controls currently required, reaching 157 distinct companies. As a result, it was possible to classify the ISO/

IEC 27001 and 27002 controls in four stages according to the importance given by the companies. The COBIT maturity levels and a risk analysis matrix were also used. Finally, the adaptable strategy was successfully tested in a company.

KEYWORDS: Information Security, Maturity Model, IT Governance, Agile Governance.

1 | INTRODUÇÃO

Na sociedade atual, globalizada, competitiva e que necessita de ações e tomadas de decisões rápidas e alinhadas ao negócio, a obtenção e a guarda do conhecimento é de suma importância. Neste contexto, a informação tornou-se um dos mais valiosos ativos das empresas, visto que as informações manuseadas nas corporações podem gerar tanto lucro como grandes prejuízos, assim como o setor que a gerencia tornou-se, nas organizações, estratégico (CASTELLS, 2007).

Mesmo sabendo da importância das informações e da criticidade dos riscos atuais, diversas organizações não contam com planos adequados na área de segurança da informação e alinhamento dos mesmos ao negócio. Em alguns casos, as organizações adotam medidas de Segurança da Informação apenas para atender as forças externas, normalmente oriundas de obrigações legais e regulamentares (ALBUQUERQUE JUNIOR; SANTOS, 2014).

1.1 Motivação e Justificativa

O aumento dos incidentes de segurança cresce aceleradamente em todo o mundo. Os ataques atingem diversos tipos de organizações, tanto as governamentais quanto empresas privadas de diversos portes e segmentos. Além disso, vem se tornando cada vez maior a lista de empresas, países e instituições governamentais que estão em um verdadeiro duelo contra “hackerativistas” (PWC, 2016).

Por conta deste e de outros fatores, existem diversos padrões, frameworks, normas e regulamentos para a implementação de modelos de segurança. Eles fornecem diretrizes ou um conjunto de boas práticas visando a Gestão da Segurança da Informação que, em sua maioria, para incorporar todos os possíveis pontos inerentes à Segurança da Informação, torna-se grande e complexo, fazendo com que, de forma geral, as empresas não os apliquem adequadamente e não gerenciem as características de segurança da informação de forma adequada.

A complexidade e formalismo dos modelos tradicionais mais utilizados atualmente abre oportunidade para rever os processos de implantação de tais padrões, modelos, normas ou frameworks, adequando-os às necessidades específicas de cada organização, visto que mesmo não implantando todos os processos ou controles, a organização consegue obter uma grande mudança organizacional, melhoria em seus processos e maior alinhamento entre a área de TIC e as estratégias organizacionais (PRADO *et al.*, 2016; SILVA NETO; ALENCAR; QUEIROZ, 2015).

Almeida Neto *et al.* (2015a) também ressaltam esse problema ao apontar a necessidade de se ter um maior controle nas empresas. Porém, é necessário ter agilidade para tratar essas questões no cenário dinâmico atual.

É importante destacar, também, a escassez de estudos que investiguem a presente temática, comparando com outras áreas da computação. Por exemplo, tem-se constantemente visto nos tópicos de interesses dos últimos anos dos principais eventos da área (pode-se citar: SBSI – Simpósio Brasileiro de Sistemas de Informação; CONTECSI – Congresso Internacional de Gestão da Tecnologia e Sistemas de Informação; SBSEG – Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais; e o SBTI – Simpósio Brasileiro de Tecnologia da Informação) chamadas para a área de “Governança de TIC”, “Gestão da Segurança da Informação”, “Normatização da Segurança da Informação”, “Políticas de Segurança da Informação”, “Maturidade em Segurança da Informação”, ou temas semelhantes, porém, ainda são poucos os trabalhos que abordem tais áreas nos anais, como abordam Alencar *et al.* (2018a).

Neste contexto, acredita-se ser relevante para a área de segurança da informação realizar estudos que busquem produzir modelos para aferir a maturidade da área de segurança da informação. Bem como, dar subsídios para um melhor alinhamento da área à Governança Ágil de TIC e ao negócio buscando meios menos complexos ou burocráticos que os atuais. Com este pensamento, espera-se que uma visão sistêmica da área de segurança da informação com processos menos burocráticos e complexos modifique, de forma positiva, o ambiente corporativo de maneira geral.

1.2 Organização do Capítulo

O presente capítulo está organizado, a partir deste ponto, da seguinte forma: a Seção 2 apresenta a problemática estudada, sendo complementada pela Seção 3 que insere a proposta de solução. Na Seção 4 é detalhado o método da pesquisa. A seção 5 traz os pontos teóricos que embasam o projeto e alguns trabalhos correlatos. A seção 6 expõe as principais atividades realizadas e os resultados alcançados. Fechando o corpo do trabalho, a seção 7 insere as considerações finais. Por fim, o trabalho é complementado com as referências.

2 | APRESENTAÇÃO DO PROBLEMA

A ISO (International Organization for Standardization) criou a Família de normas 27000 que versam sobre a segurança da informação. Sendo esse um dos principais mecanismos na área de segurança da informação no que tange, especialmente, aos aspectos táticos e operacionais. Apesar destes modelos serem muito bem estruturados, o formalismo, por algumas vezes excessivo, tem tornado a adoção e melhoria contínua de seus processos uma tarefa complexa (PRADO *et al.*, 2016; SILVA NETO; ALENCAR;

QUEIROZ, 2015).

A família de normas ISO de segurança da informação apresenta um conjunto de controles. Porém, estudos que demonstrem se sua aplicação realmente afeta a maturidade de segurança da informação da corporação, assim como os alinhando à Governança Ágil de TIC ainda são escassos.

Formas de mensuração da governança de TIC de uma corporação têm sido exploradas (ALMEIDA NETO *et al.*, 2015a) como meios de se analisar a situação da instituição, assim como possibilitando a comparação dos níveis de governança entre corporações distintas. Tal aspecto pode ser útil para agregar valor à empresa, como pode ser visto em casos de análises para mercado de ações, vendas, fusões, etc. A área de engenharia de software é outro exemplo, pois utiliza com constância níveis de qualidade e maturidade para diferenciar empresas e produtos.

Diante do contexto citado, esta pesquisa pretende explorar a área respondendo o questionamento: “Como mensurar e priorizar a segurança da informação corporativa com base nos atuais arcabouços existentes na área?”

Para responder a este questionamento foi concebida uma proposta de solução, descrita na próxima seção, que é norteada pelos objetivos detalhados em seguida.

3 | PROPOSTA DE SOLUÇÃO

O presente trabalho está voltado a abordar os desafios de adoção e melhoria contínua da área de segurança da informação em organizações de natureza variadas, através da concepção, definição e avaliação de uma estratégia para a segurança da informação corporativa abordando as áreas de priorização e maturidade concebida, principalmente, segundo os princípios expostos na família de normas ISO/IEC 27000 e no COBIT, de forma a subsidiar melhorias na área de segurança da informação, sua gestão e governança.

A partir deste cenário, surge então a proposta de concepção de uma estratégia dedicada à avaliação de maturidade e priorização da segurança da informação no ambiente corporativo. Esta estratégia, através, principalmente, da ISO/IEC 27001 (ABNT, 2013a) e 27002 (ABNT, 2013b) que abordam a implantação e gestão da segurança da informação, bem como a ISO/IEC 27005 (ABNT, 2011), que versa sobre a área de gestão de risco, visa propor um arcabouço para que se possa, na área de segurança da informação:

- Mensurar a maturidade atual da empresa;
- Apontar áreas com maior desenvolvimento;
- Apontar áreas com maior carência;
- Comparar se as ações implantadas impactaram na corporação de forma a aumentar seu nível de maturidade;

- Comparar empresas variadas;
- Priorizar as ações de segurança da informação de acordo com a criticidade;

Além dos pontos expostos, percebe-se, como já apresentado na problemática em questão, a burocracia existente nos processos atuais. Neste sentido é possível observar um conflito entre o formalismo apresentado pela maioria destas iniciativas e a agilidade imposta por um mercado cada vez mais competitivo.

Em meados de 2001, pode-se observar, na área de desenvolvimento de software, uma dicotomia semelhante. Naquele período, metodologias como a Rational Unified Process (RUP), tidas como precursoras ao desenvolvimento de software (KRUCHTEN, 2004), também se depararam com um dilema parecido. Este problema motivou o surgimento do Manifesto for Agile Software Development (BECK *et al.*, 2001), manifesto que abordou um conjunto inovador de valores e princípios, promovendo uma quebra de paradigmas.

Com esta mesma perspectiva, a área de Governança de TIC vem sofrendo com os processos lentos e já surgem estudos de uma visão ágil e prática da mesma utilizando alguns princípios do Manifesto Ágil, entre os quais pode-se citar Almeida Neto (2015b) e Luna *et al.* (2016).

Desta forma, a estratégia proposta também se debruçará sobre o ramo da praticidade e agilidade de forma a se nortear em tais princípios e ideais como um possível meio de minimizar problemas referentes ao formalismo dos modelos e arcabouços atuais utilizados na área de segurança da informação.

Sintetizando a proposta de solução, o objetivo principal deste trabalho é propor uma estratégia para avaliação da maturidade e priorização da segurança da informação no ambiente corporativo através da utilização, em conjunto, dos principais arcabouços existentes na área.

4 | METÓDO DA PESQUISA

Para atender aos objetivos propostos, a pesquisa em questão é categorizada como Exploratória e Descritiva, utilizando os procedimentos técnicos de Pesquisa Bibliográfica, Revisão e Mapeamento Sistemático da Literatura, *survey* e Grupo Focal. Sendo Quanti-qualitativa e concentrada na área de Ciência da Computação: Sistemas de Informação e Segurança da Informação.

De acordo com WOHLIN e AURUM (2015), a presente pesquisa se classifica como exposto na Figura 1.

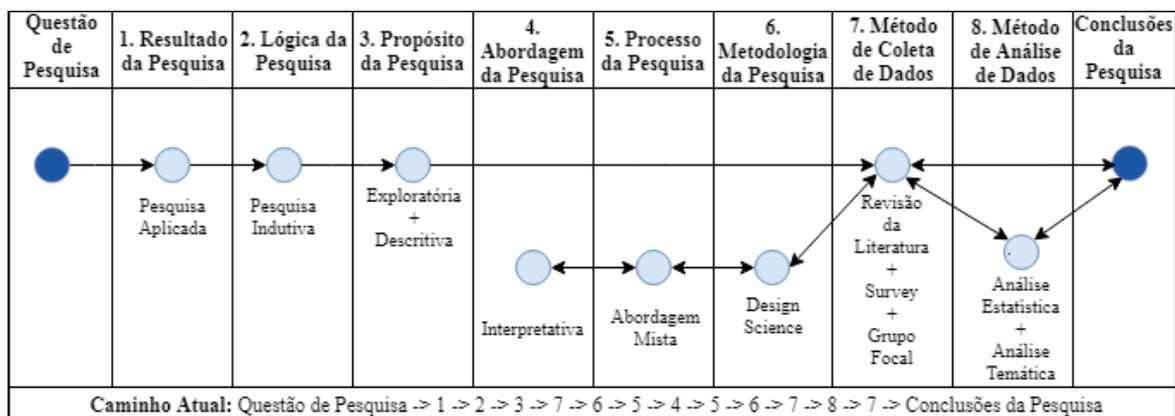


Figura 1. Estrutura de Tomada de Decisão Metodológica

A execução da pesquisa será realizada em atividades divididas em duas fases distintas, conforme Figura 2. A primeira etapa consiste em uma fase exploratória e tem por objetivo a construção de uma base teórica consistente para suportar a etapa seguinte. A etapa 2, apresenta uma característica mais descritiva e visa a real construção do modelo proposto.

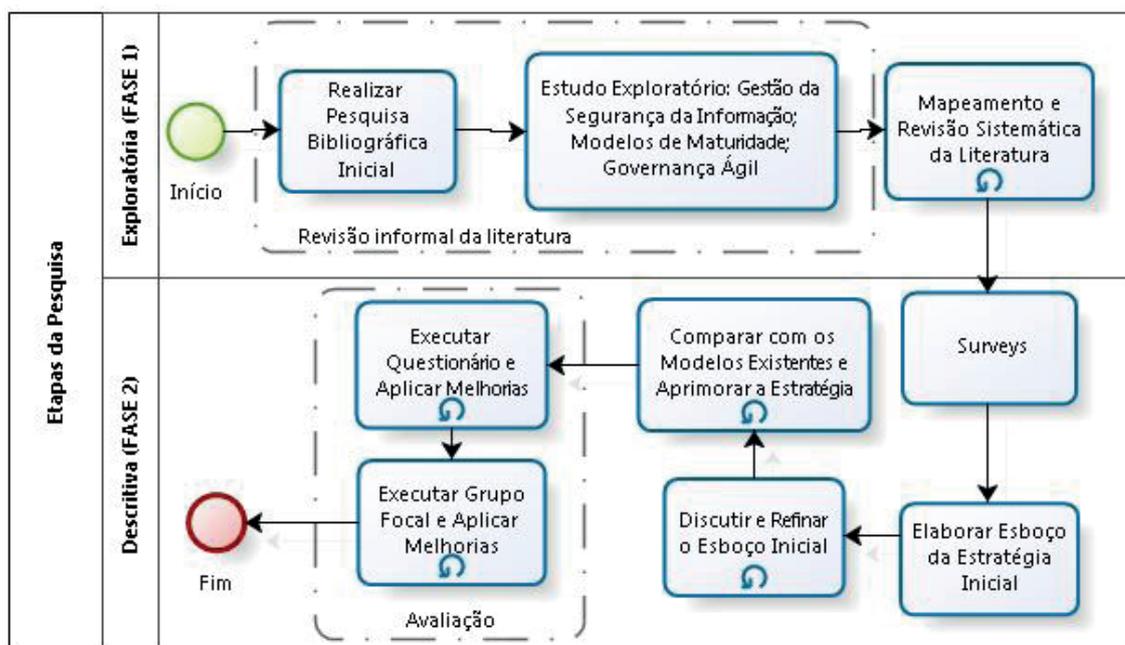


Figura 2. Etapas da Pesquisa

A Revisão Sistemática da Literatura foi baseada no método de Kitchenham (2004) em bases nacionais e internacionais.

A fase de *surveys* consiste no levantamento de características das empresas participantes e da sua visão quanto a importância de cada controle das ISO 27001 e 27002, para isso foram utilizados dois *surveys*.

O primeiro *survey* é uma aplicação do documento já utilizado outras vezes, entre elas Alencar, Queiroz e De Queiroz (2013a; 2013b) e que baseou o questionário utilizado por Silva Neto, Alencar e Queiroz (2015). Sendo composto por 43 questões

divididas em seis categorias, sendo elas: Dados da empresa; Dados do respondente; Importância estratégica da informação; Ferramentas de SI na empresa; Recursos humanos e estrutura organizacional; e Segurança da informação corporativa. Ao final foram adicionadas mais cinco questões inerentes a amplitude da pesquisa atual.

O segundo consta do nome da empresa (para correlacionar com o primeiro) e os 114 controles da versão de 2013 da ISO/IEC 27001 e 27002. Para os controles foram utilizados uma escala *likert* de cinco níveis em uma escala de 1 (nenhuma importância) a 5 (muito importante), sendo a nota 3 categorizada como neutro na escala. Nas questões o respondente marcará a importância de cada controle para o seu ambiente. Mesmo método utilizado por Silva Neto, Alencar e Queiroz (2015), porém os autores colocavam a versão anterior da ISO/IEC 27002 (de 2005) com 133 controles.

A amostra foi composta por empresas de todas as regiões do Brasil, bem como com abrangência de atuação local, regional, nacional e multinacional, sendo contabilizada apenas uma resposta por empresa. Os respondentes deverão ser da área de TIC e, preferencialmente, o responsável pela área de segurança da informação.

Após a análise, os resultados dos controles foram ordenados de acordo com sua média das notas de importância. Posteriormente verificado se existe algum pré-requisito entre os controles e, neste caso, os pré-requisitos foram inseridos antes. Essa fase gerou o Esboço do Modelo Inicial que, posteriormente, foi debatido e refinado. Após superada esta etapa, foi comparado com os modelos de maturidade existentes e ajustado. Estas etapas poderão ser repetidas.

Em diversas etapas, a versão atual da estratégia ou algum de seus componentes, foi enviado aos especialistas na área, procurando obter, ao menos, 5 respostas, e também enviado às empresas respondentes junto com o *survey* 3. Este terceiro *survey* questionou sobre a sua adequação, sendo a primeira etapa de validação. Após receber as respostas dos especialistas e empresas, é analisada e aplicada as melhorias propostas. Esta etapa poderá ser repetida.

Tendo o modelo pré-avaliado e melhorado pela indicação das empresas e especialistas, será realizado grupo focal para a validação final do mesmo (ainda em fase preparatória, não realizada). Esta etapa, semelhante a validação do modelo de maturidade de Almeida Neto *et al.* (2015b), poderá ser repetida.

Desta forma, acredita-se que objetivo principal seja atingido pela criação, ao final do trabalho, da estratégia para mensurar a maturidade e priorizar a segurança da informação atendendo aos princípios da governança ágil.

Esperar-se, também, que os demais objetivos específicos propostos, não atendidos com o modelo de maturidade formulado, sejam solucionados com as etapas intermediárias para a construção do trabalho e da estratégia: a análise dos dados coletados e da revisão da literatura. Como consequência dos passos realizados e da solução dos objetivos propostos, acredita-se que o problema de pesquisa seja resolvido a contento.

5 | BASES TEÓRICAS E TRABALHOS CORRELATOS

A presente pesquisa baseia-se, principalmente, nos conhecimentos das áreas de Governança de Segurança da Informação (ABNT, 2013c; MANOEL, 2014), Governança Ágil (LUNA *et al.*, 2014; LUNA *et al.*, 2016), Modelos de Maturidade (KAROKOLA; KOWALSKI; YNGSTRÖM, 2011; RIGON *et al.*, 2014) e normas ISO/IEC 27001 (ABNT, 2013a), 27002 (ABNT, 2013b) e 27005 (ABNT, 2011).

Na literatura percebe-se um conjunto de trabalhos que tratam de melhorias e maturidade para segurança da informação que podem ser inseridos como correlatos, entre os trabalhos pode-se citar: Rigon *et al.* (2014), Karokola; Kowalski e Yngström (2011) e Mahopo, Abdullah e Mujinga (2015). Porém eles ainda sofrem com os já citados problemas de burocracia e formalismo.

O modelo de maturidade de Almeida Neto *et al.* (ALMEIDA NETO *et al.*, 2015a, 2015b) tenta solucionar os problemas da burocracia e formalismo, baseando-se nos princípios da Governança Ágil, porém não tem foco em segurança da informação. Já Silva Neto, Alencar e Queiroz (2015), propõe uma simplificação da segurança, mas apresenta apenas uma visão inicial de uma política de segurança da informação simplificada, não formulando um modelo de maturidade. Assim, acredita-se que os trabalhos encontrados não atendem, por completo, todos os objetivos específicos propostos nesta presente pesquisa e problemas expostos.

6 | ATIVIDADES JÁ REALIZADAS E RESULTADOS ALCANÇADOS

Dentro das atividades propostas no método de pesquisa, expostas na Figura 2, apenas a última avaliação, através do Grupo Focal, não foi realizada. Os principais resultados das etapas já desenvolvidas são exibidos a seguir.

A estratégia consiste, inicialmente, em levantar a visão das empresas quanto a criticidade para cada um dos 114 controles da ISO/IEC 27001 (Anexo I) e 27002, pontuando em uma escala de 1 (pouco importante) até 5 (muito importante). Após a resposta, em uma segunda fase, os controles serão classificados em quartis. Para isto será calculada a média das notas de cada controle e ordenando-os. O primeiro quartil representa os 25% dos controles considerados mais importantes, enquanto o último quartil, apontará os 25% dos controles com menor nível de importância. Cada quartil é categorizado como “Estágio”. Em uma situação ideal, os controles estariam distribuídos, conforme Quadro 1.

Média do Controle	1º Quartil (maiores médias)	2º Quartil	3º Quartil	4º Quartil (menores médias)
Estágio	Essencial	Intermediário	Avançado	Completo
Controles	29	28	29	28

Quadro 1. Divisão dos controles por quartis

Porém a divisão dos quartis pode não ser tão exata quanto a proposta, visto que após a primeira divisão dos quartis deverá ser analisado se existe alguma dependência ou pré-requisito entre algum controle. Existindo, o controle pré-requisito deverá estar no mesmo quartil ou em um quartil anterior. Caso ele esteja em um quartil posterior, o controle pré-requisito será colocado no mesmo quartil do seu dependente. Por exemplo, uma situação que, inicialmente, o controle da ISO/IEC 27001 “A.5.1.1 - Um conjunto de políticas de segurança da informação deve ser definido, aprovado pela direção, publicado e comunicado para os funcionários e partes externas relevantes” está no 3º quartil, enquanto o controle ISO/IEC 27001 “A.5.1.2 - As políticas de segurança da informação devem ser analisadas criticamente a intervalos planejados ou quando mudanças significativas ocorrerem, para assegurar a sua contínua pertinência, adequação e eficácia” está no 2º quartil. Percebe-se que o controle A.5.1.1 (definição da política de segurança da informação) é pré-requisito para o controle A.5.1.2 (análise da política de segurança da informação), neste caso o controle A.5.1.1 será inserido no 2º quartil, mesmo do controle A.5.1.2. Esta análise de pré-requisitos é a terceira etapa.

Um quartil também pode ter a quantidade de controles aumentadas caso exista empate nas notas dos últimos controles, sendo todos incorporados ao quartil. Por exemplo, se os controles da posição 27, 28, 29 e 30 tiverem a mesma média, todos serão incorporados ao 1º quartil, tendo, neste caso, o 1º quartil com 31 controles no lugar dos 29 iniciais, exemplificados no Quadro 1.

A quarta etapa consiste em definir os níveis de maturidade. Esta proposta utiliza os níveis de maturidade e definições do COBIT (ISACA, 2007), são eles: Nível 0 (Inexistente), Nível 1 (Inicial), Nível 2 (Repetível), Nível 3 (Definido), Nível 4 (Gerenciado), Nível 5 (Otimizado).

Como quinto passo tem-se a definição do nível mínimo de maturidade para cada controle. Uma forma é elencar um padrão, por exemplo, todos os controles deverão alcançar, no mínimo, o nível 3 (definido). Uma outra opção, mais recomendada, é realizar uma análise de risco na empresa, como sugere a ISO/IEC 27005, e categorizar o nível mínimo de cada controle de acordo com sua probabilidade e impacto.

Para esta estratégia a probabilidade e o impacto serão categorizados como baixo, médio ou alto. Recebendo, respectivamente, o peso 1, 2 ou 3. Uma matriz é formada e o valor mínimo de maturidade a ser alcançado será a soma da nota da probabilidade e do impacto, conforme Figura 3.

Uma exceção é quando se atinge uma probabilidade e impacto altos, recebendo a nota 6 (3+3). Ciente que o modelo proposto aborda até o nível de maturidade 5 (otimizado), os controles categorizados com nota 6 deverão atingir o nível 5 (otimizado) e, por sua criticidade, deverão ser tratados prioritariamente pela empresa. Desta forma os controles que suas ausências geram riscos com maior probabilidade e maior impacto deverão ser tratados de forma diferenciada.

Probabilidade	Alta (3)	4 - Gerenciado	5 - Otimizado	6 - Otimizado
	Média (2)	3 - Definido	4 - Gerenciado	5 - Otimizado
	Baixa (1)	2 - Repetível	3 - Definido	4 - Gerenciado
		Baixo (1)	Médio (2)	Alto (3)
		Impacto		

Figura 3. Nível de maturidade mínimo de acordo com o impacto e probabilidade

Os controles não aplicáveis deverão ser devidamente justificados no relatório a ser apresentado no final da avaliação. Tendo como nível mínimo o 0 (inexistente) e não sendo contabilizados na estratégia.

A escolha do COBIT, bem como das ISO/IEC 27001, 27002 e 27005 se deu devido a sua consolidação na área.

Após as definições iniciais, tem-se a aplicação da estratégia como o sexto passo. As 6 etapas supracitadas são ilustradas na Figura 4.

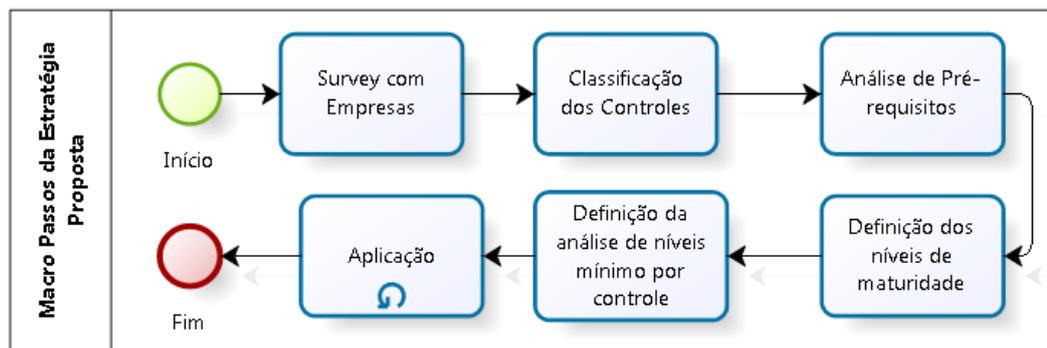


Figura 4. Macro passos da estratégia

A fase de aplicação da estratégia consiste em analisar e aplicar cada controle ordenados nos estágios até o estágio pretendido, conforme Figura 5.

Ressalta-se que o alcance de níveis e estágios de maturidade mais avançados geram custos e demandam tempo, podendo não ser do interesse de todas as empresas alcançar o Estágio Completo e o nível de maturidade 5 (Otimizado) em todos seus controles.

Também importante mencionar a etapa final da aplicação. Que consiste na geração do Relatório da Avaliação e da Comunicação aos Stakeholders das análises feitas, níveis a ser alcançados, pontos fortes e fracos da empresa.

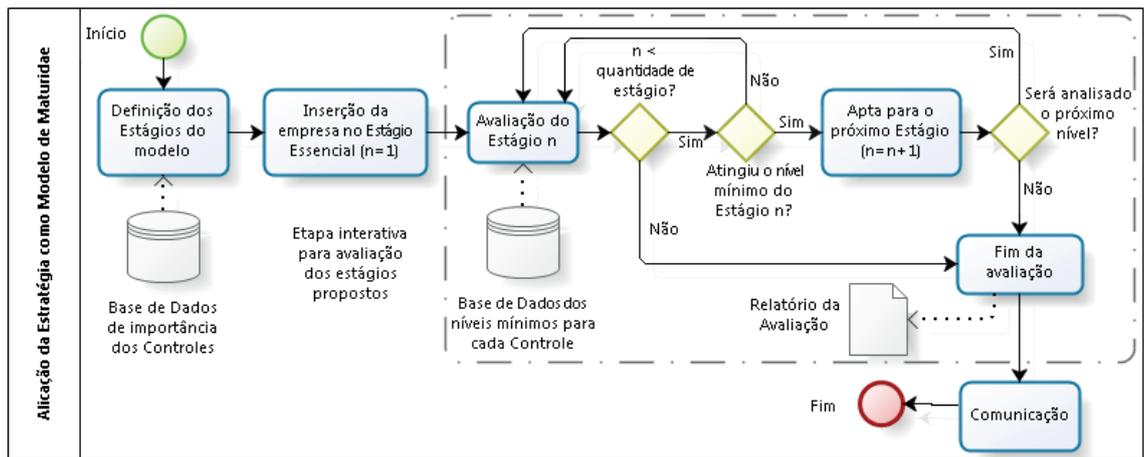


Figura 5. Fase de aplicação da Estratégia

6.1 Modelo de Maturidade

Os modelos de maturidade em segurança da informação, como os citados na Seção de trabalhos correlatos, utilizam, por exemplo, os controles da ISO/IEC 27001 ou 27002 medindo-os e classificando a maturidade, normalmente, com a média dos valores medidos. Tendo todos os controles o mesmo peso.

Como um diferencial da presente estratégia, o modelo de maturidade proposto trabalha com estágios e níveis de maturidades. Nos estágios estão classificados os controles por quartis, de acordo com a importância dada a eles pelas empresas. E a empresa só será avaliada no segundo quartil ao atender o nível mínimo dos controles do estágio anterior. Nesta nova configuração os controles mais importantes serão inseridos de forma prioritária. O modelo de maturidade proposto tem seus estágios e níveis apresentados na Figura 6.



Figura 6. Estágios e níveis do modelo de maturidade

No modelo proposto, para se passar de um estágio (Essencial, Intermediário, Avançado ou Completo) para o seguinte só é possível após atingir o nível de maturidade mínimo de 3 para todos os controles, sendo este o parâmetro da Base de Dados dos níveis mínimo para cada Controle (Figura 5). Ou seja, se a empresa é categorizada como Avançada Nível 2, significa que a média dos controles do grupo Avançado obteve a nível de maturidade 2 (Repetível) e que todos os controles aplicáveis do estágio Essencial e do Intermediário foram mensurados, no mínimo, como nível 3 (Definido). Sendo um diferencial do modelo proposto.

Para verificar quais controles da ISO/IEC 27001 e 27002 são os mais importantes, foi utilizado um *survey* com empresas brasileiras. Obtendo resposta de 157 empresas distintas, sendo 23 de atuação multinacional, conforme descrição no Gráfico 1.

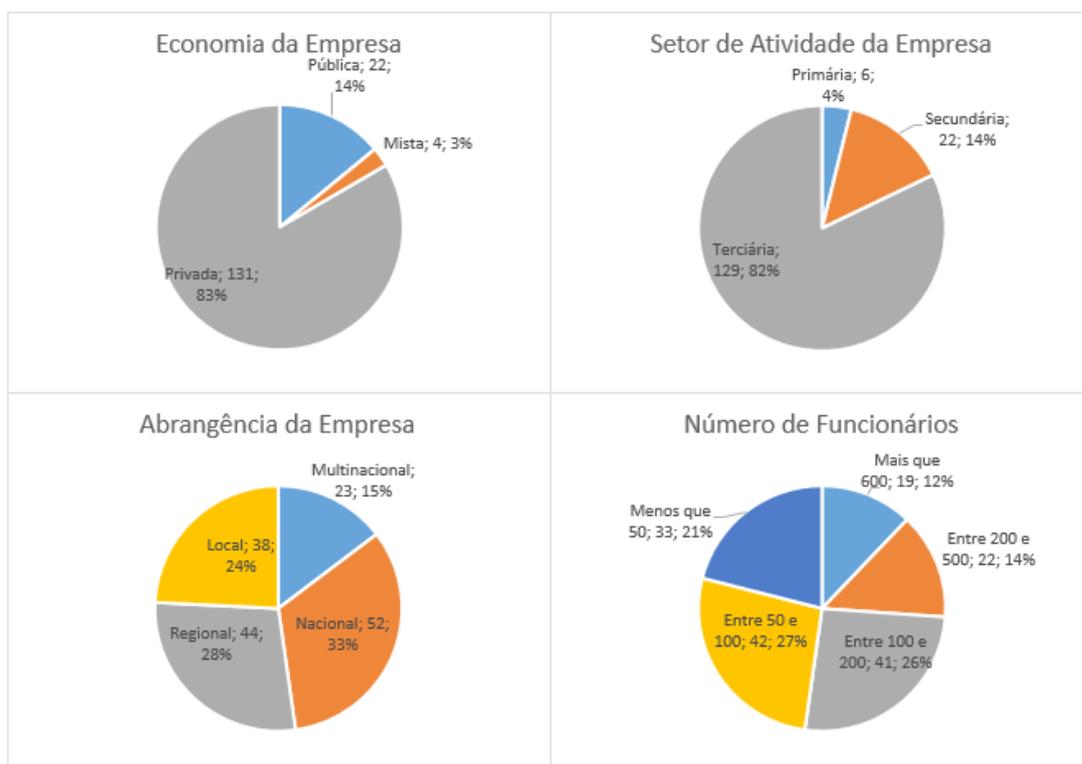


Gráfico 1. Classificação da Amostra

Todos os entrevistados eram funcionários ou sócios das empresas, sendo 69,43% era o responsável pela área de Segurança da Informação ou trabalhavam exclusivamente ou prioritariamente com a área de segurança da informação, os demais respondentes eram da área de TIC ou o responsável por ela. Dentre as empresas pesquisadas, 23,57% têm a TIC como área fim. Vale ressaltar que se alcançou empresas de todas as regiões do Brasil.

A quantidade de funcionários das empresas respondentes variou de 15 a 12 mil, enquanto a quantidade de computadores variou de 17 computadores a 7,7 mil máquinas.

Com base na média das respostas das empresas alcançadas, foi realizada a divisão dos controles entre os estágios, alcançado a configuração exposta no Quadro 2 que representa a Base de dados de importância dos controles (Figura 5). A numeração dos controles está de acordo com a exibida no Anexo A da ISO/IEC 27001.

Estágio	Quantidade de Controles	Controles
Essencial	31	A.5.1.1, A.6.1.1, A.6.1.5, A.6.2.2, A.7.1.1, A.7.2.1, A.8.1.2, A.8.1.3, A.8.2.1, A.8.2.3, A.9.1.2, A.9.2.1, A.9.2.3, A.9.2.4, A.9.2.5, A.9.4.2, A.9.4.4, A.11.1.5, A.11.2.4, A.11.2.5, A.11.2.6, A.11.2.7, A.12.5.1, A.12.6.2, A.13.1.3, A.15.1.3, A.18.1.1, A.18.1.2, A.18.1.3, A.18.1.4 e A.18.1.5
Intermediário	27	A.5.1.2, A.6.1.2, A.6.2.1, A.7.2.2, A.8.1.1, A.8.3.1, A.9.2.6, A.9.4.3, A.11.1.3, A.11.2.2, A.11.2.3, A.12.1.3, A.12.1.4, A.12.2.1, A.13.1.1, A.13.2.1, A.13.2.3, A.14.1.1, A.14.2.6, A.15.1.1, A.16.1.1, A.16.1.2, A.16.1.4, A.16.1.5, A.16.1.7, A.17.1.1 e A.18.2.2.
Avançado	29	A.7.2.3, A.8.1.4, A.8.2.2, A.9.1.1, A.9.3.1, A.9.4.1, A.9.4.5, A.11.1.1, A.11.1.2, A.11.2.1, A.11.2.9, A.12.1.1, A.12.1.2, A.12.3.1, A.12.4.1, A.12.6.1, A.12.7.1, A.13.1.2, A.13.2.4, A.14.1.2, A.14.1.3, A.14.2.5, A.14.2.9, A.15.1.2, A.15.2.1, A.16.1.3, A.17.2.1, A.18.2.1 e A.18.2.3.
Completo	27	A.6.1.3, A.6.1.4, A.7.1.2, A.7.3.1, A.8.3.2, A.8.3.3, A.9.2.2, A.10.1.1, A.10.1.2, A.11.1.4, A.11.1.6, A.11.2.8, A.12.4.2, A.12.4.3, A.12.4.4, A.13.2.2, A.14.2.1, A.14.2.2, A.14.2.3, A.14.2.4, A.14.2.7, A.14.2.8, A.14.3.1, 15.2.2, A.16.1.6, A.17.1.2 e A.17.1.3.

Quadro 2. Divisão dos controles por quartis

Periodicamente os controles de cada estágio são recalculados, devido a possibilidade de novas respostas ao modelo, aumentando a base de dados e representando a real importância de cada controle. Fato que confere um dinamismo ao modelo, sendo mais um diferencial aos já existentes.

Maiores detalhes sobre a estratégia apresentada, bem como seu modo de aplicação independente e um exemplo de aplicação real são exibidos em Alencar *et al.* (2018b).

O alinhamento da estratégia com o pensamento da Governança Ágil, em especial seus meta-valores apontados por Luna *et al.* (2016) como, por exemplo, o meta-valor “Comportamento e prática do que processos e procedimentos”, pode ser percebido ao se colocar a decisão das empresas sobre a criticidade de cada controle acima da necessidade de atendimento total dos normativos, bem como as diferentes formas de aplicação da estratégia (ALENCAR *et al.*, 2018b).

6.2 Evolução da Pesquisa e Publicações

Conforme exibido na Figura 1, a pesquisa utilizou-se do *Design Science Research* (DSR) como método. O DSR foi selecionado por ser propício para construção e avaliação de artefatos que visam atender aos requisitos de um problema real. Sendo o artefato de saída uma construção, modelo, método, instanciação ou estratégia nova.

Uma das características do método é a importância dada à divulgação e avaliação dos resultados encontrados. Visando o atendimento deste item, bem como analisar a aceitação deste trabalho por parte da comunidade científica e resultando como mais uma fase de avaliação, as etapas e evoluções deste projeto foram publicadas em congressos e periódicos nacionais e internacionais, totalizando nove publicações aceitas: Silva Neto, Alencar e Queiroz (2015), Alencar e Moura (2017a, 2017b), Alencar, Tenorio Junior e Moura (2017a, 2017b), Alencar e Moura (2018) e Alencar *et*

7 | CONSIDERAÇÕES FINAIS

O trabalho apresentou uma estratégia para a segurança da informação que pode ser aplicada como modelo de maturidade, bem como é possível sua aplicação independente focada em um alinhamento do negócio aos controles de segurança da informação, sendo testado com sucesso (ALENCAR *et al.*, 2018b).

A estratégia permite uma análise do nível de maturidade de forma mais aprimorada, fazendo com que se tenha uma maior visibilidade, através dos estágios, se os controles principais foram mensurados, não apenas analisando um nível de maturidade final da empresa.

A estratégia proposta consiste, basicamente, em um conjunto de módulos: Os 4 estágios propostos (Essencial, Intermediário, Avançado e Completo), os níveis de maturidade (baseado no COBIT), os aspectos e controles a analisar (utilizado os controles da ISO/IEC 27001 e 27002) e a definição do nível mínimo de cada controle, pré-definido ou baseado no risco inerente a cada um deles (ISO/IEC 27005). Acredita-se que essa estrutura seja um diferencial da estratégia proposta, visto que utiliza arcabouços consolidados, mas também possibilita, na aplicação independente, a troca de algum módulo por outro a critério da empresa. Por exemplo, um conjunto de níveis de maturidades diferentes já utilizados na empresa ou uma análise de risco baseado em outros parâmetros. Tal possibilidade, permite um maior ajuste da estratégia ao negócio, bem como pode reduzir tempo e recursos em sua aplicação ao utilizar algum módulo já conhecido e existente, como abordam Alencar *et al.* (2018b).

O fato da possibilidade de se avaliar como uma aplicação independente, utilizar módulos e permitir ajustes ao negócio apontam a convergência aos pensamentos da governança ágil.

Por fim, espera-se que a estratégia proposta promova apoio significativo à adoção e melhorias contínuas à Segurança da Informação mensurando a maturidade da empresa; apontando as áreas com maior desenvolvimento em segurança da informação e as áreas que precisam de maior investimento; ter, de forma palpável, o impacto na segurança da informação de alterações (sejam elas pessoais, procedimentais ou tecnológicas) ocorridas na corporação; possibilidade de comparação do “nível de maturidade de segurança” entre setores ou empresas; diminuição da burocracia e formalismo na área gerando maior agilidade na gestão da segurança da informação corporativa.

REFERÊNCIAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27001: Tecnologia da informação – Técnicas de segurança – Sistemas de gestão da segurança da informação – Requisitos**. Rio de Janeiro, p. 30. 2013a.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27002: Tecnologia da informação – Técnicas de segurança – Código de prática para controles de segurança da informação**. Rio de Janeiro, p. 99. 2013b.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27005: Tecnologia da informação – Técnicas de segurança – Gestão de riscos de segurança da informação**. Rio de Janeiro, p. 87. 2011.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 27014: Tecnologia da Informação – Técnicas de Segurança – Governança de segurança da informação**. Rio de Janeiro, p. 12. 2013c.
- ALBUQUERQUE JUNIOR, A. E.; DOS SANTOS, E. M. Adoção de medidas de segurança da informação: um modelo de análise para institutos de pesquisa públicos. **Revista Brasileira de Administração Científica**, v. 5, n. 2, p. 46-59, 2014.
- ALENCAR, G. D.; MENEZES, B. P.; AMORIM, E. S.; FARIAS JUNIOR, I. H.; MOURA, H. P. Governança, Gestão e Maturidade da Segurança da Informação: um mapeamento sistemático do cenário nacional. **Revista de Sistemas e Computação**, v. 8, n. 1, p. 153-173, 2018a.
- ALENCAR, G. D.; MOURA, H. P.; FARIAS JUNIOR, I. H.; TEIXEIRA FILHO, J. G. A. An Adaptable Maturity Strategy for Information Security. **Journal of Convergence Information Technology (JCIT)**, v. 13, n. 2, p. 1-12, 2018b.
- ALENCAR, G. D.; AMORIM, E. S.; MENEZES, B. P.; MOURA, H. P. Scientific Production about Governance, Management and Maturity of Information Security in the Main Computing-Related Brazilian Journals and Conferences. In: **15th International Conference on Information Systems & Technology Management (CONTECSI)**, São Paulo – SP, 2018c. *Anais...* 2018c, USP. p. 2756-2782.
- ALENCAR, G. D.; MOURA, H. P. Método Simplificado para Aplicação e Priorização da Segurança da Informação: Reflexões Teóricas e Soluções Futuras. In: **15th International Conference on Information Systems & Technology Management (CONTECSI)**, São Paulo – SP, 2018. *Anais...* 2018, USP. p. 2801-2816.
- ALENCAR, G. D.; TENORIO JUNIOR, A. J. A.; MOURA, H. P. Theoretical Guidelines for an Agile Model of Governance, Management and Maturity for Information Security. In: **14th International Conference on Information Systems & Technology Management (CONTECSI)**, São Paulo – SP, 2017a. *Anais...* 2017a, USP. p. 3661-3690.
- ALENCAR, G. D.; TENORIO JUNIOR, A. J. A.; MOURA, H. P. Information Security Policy: A Simplified Model Based on ISO 27002. In: **14th International Conference on Information Systems & Technology Management (CONTECSI)**, São Paulo – SP, 2017b. *Anais...* 2017b, USP. p. 4135-4156.
- ALENCAR, G. D.; MOURA, H. P. Maturity Model for Information Security: A Proposal Based on ISO/IEC 27001 and 27002 According to the Principles of Agile Governance. In: **14th International Conference on Information Systems & Technology Management (CONTECSI)**, São Paulo – SP, 2017a. *Anais...* 2017a, USP. p. 4817-4832.
- ALENCAR, G. D.; MOURA, H. P. Proposal of Information Security Maturity Model based on ISO/IEC 27001 and 27002 according to the Principles of Agile Governance. In: **XIII Simpósio Brasileiro de Sistemas de Informação (SBSI) / X Workshop de Teses e Dissertações em Sistemas de Informação (WTDSI)**, Lavras – MG, 2017b. *Anais...* 2017b, SBC. p. 80-84.

- ALENCAR, G. D.; QUEIROZ, A. A. L.; DE QUEIROZ, R. J. G. B. Insiders: Análise e Possibilidades de Mitigação de Ameaças Internas. **Revista Eletrônica de Sistemas de Informação**, v. 12, n. 3, p. 1-38, 2013.
- ALENCAR, G. D.; QUEIROZ, A. A. L.; DE QUEIROZ, R. J. G. B. Insiders: Um Fator Ativo na Segurança da Informação. In: **IX Simpósio Brasileiro de Sistemas de Informação (SBSI)**, João Pessoa – PB, 2013b. *Anais...* 2013b, SBC. p. 254-259.
- ALMEIDA NETO, H. R.; DE MAGALHÃES, E. M. C.; DE MOURA, H. P.; DE ALMEIDA TEIXEIRA FILHO, J. G.; CAPPELLI, C.; MARTINS, L. M. F. Avaliação de um Modelo de Maturidade para Governança Ágil em Tecnologia da Informação e Comunicação. **iSys - Revista Brasileira de Sistemas de Informação**, v. 8, n. 4, p. 44-79, 2015a.
- ALMEIDA NETO, H. R.; MAGALHÃES, E. M. C.; MOURA, H. P.; TEIXEIRA FILHO, J. G. A.; CAPELLI, C.; MARTINS, L. M. F. Avaliação de um Modelo de Maturidade para Governança Ágil em TIC usando Focus Group. In: **XI Simpósio Brasileiro de Sistemas de Informação (SBSI)**, Goiânia – GO, 2015b. *Anais...* 2015b, SBC. p. 15-22.
- BECK, K.; BEEDLE, M.; BENNEKUM, A. van; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R. C.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. Manifesto for Agile Software Development. 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em: 20 mar.2016.
- CASTELLS, M. **Era da Informação: A Sociedade em Rede**. Volume 1. 10ª Edição. São Paulo: Editora Paz e Terra, 2007. 698 p.
- ISACA. **COBIT 4.1: framework, control objectives, management guidelines and maturity models**. IT Governance Institute, 2007.
- KAROKOLA, G.; KOWALSKI, S.; YNGSTRÖM, L. Towards an Information Security Maturity Model for Secure e-Government Services: A Stakeholders View. In: **Fifth International Symposium on Human Aspects of Information Security and Assurance (HAISA)**, Londres – Inglaterra, 2011. *Proceedings...* 2011, p. 58-73.
- KITCHENHAM, B. **Procedures for performing systematic reviews**. Technical Report. Keele University, 2004, 33 p.
- KRUCHTEN, P. **The rational unified process: an introduction**. 3ª edição. Editora Addison-Wesley, 2004. 336 p.
- LUNA, A. J. D. O.; KRUCHTEN, P.; PEDROSA, M. L. D. E.; NETO, H. R.; DE MOURA, H. P. State of the art of agile governance: a systematic review. **International Journal of Computer Science and Information Technology**, v. 6, n. 5, p. 121–141, 2014.
- LUNA, A. J. H. O.; KRUCHTEN, P.; RICCIO, E. L.; DE MOURA, H. P. Foundations for an Agile Governance Manifesto: A Bridge for Business Agility. In: **13th International Conference on Information Systems and Technology Management (CONTECSI)**, São Paulo – SP, 2016. *Proceedings...* 2016, USP. p. 4391-4404.
- MAHOPO, B.; ABDULLAH, H.; MUJINGA, M. A formal qualitative risk management approach for IT security. In: **Information Security for South Africa (ISSA)**, Joanesburgo - África do Sul, 2015. *Proceedings...* 2015, IEEE. p. 1-8.
- MANOEL, S. S. **Governança de Segurança da Informação: Como criar oportunidades para o seu negócio**. Rio de Janeiro: Editora Brasport, 2014. 168 p.

PRADO, E. P. V.; MANCINI, M.; BARATA, A. M.; SUN, V. Governança de TI em Organizações do Setor de Saúde: um Estudo de Caso de Aplicação do COBIT. In: **XII Simpósio Brasileiro de Sistemas de Informação (SBSI)**, Florianópolis – SC, 2016. *Anais...*, SBC. p. 1-8.

PWC. PricewaterhouseCoopers. Pesquisa global de segurança da informação 2016. Disponível em: <<http://www.pwc.com.br/pt/publicacoes/servicos/consultoria-negocios/2016/pwc-pesquisa-global-seguranca-informacao-16.html>>. Acesso em: 04 jan. 2017

RIGON, E. A.; WESTPHALL, C. M.; SANTOS, D. R.; WESTPHALL, C. B. A cyclical evaluation model of information security maturity. **Information Management & Computer Security**, v. 22, n. 3, p. 265-278, 2014.

SILVA NETO, G. M.; ALENCAR, G. D.; QUEIROZ, A. A. L. Proposta de Modelo de Segurança Simplificado para Pequenas e Médias Empresas. In: **XI Simpósio Brasileiro de Sistemas de Informação (SBSI)**, Goiânia – GO, 2015. *Anais...* 2015, SBC. p. 299-306.

WOHLIN, C.; AURUM, A. Towards a decision-making structure for selecting a research design in empirical software engineering. **Empirical Software Engineering**, v. 20, n. 6, p. 1427–1455, 2015.

PROGRAMAÇÃO PARA TODOS: ANÁLISE COMPARATIVA DE FERRAMENTAS UTILIZADAS NO ENSINO DE PROGRAMAÇÃO

Silvino Marques da Silva Junior

Instituto Federal do Piauí (IFPI) – Campus
Floriano

Floriano – PI – Brasil

Sônia Virginia Alves França

Programa de Pós-Graduação em Tecnologia e
Gestão em Educação a Distância - Universidade
Federal Rural de Pernambuco (UFRPE)

Recife – PE – Brasil

RESUMO: A expansão do ensino de programação de computadores para diferentes públicos e áreas é uma realidade eminente, ao mesmo tempo que os assuntos relacionados a essa disciplina são considerados campeões em apresentar dificuldades de aprendizagem por parte dos estudantes. Este artigo tem por objetivo analisar e fazer um comparativo com algumas ferramentas que foram propostas em trabalhos anteriores e que são utilizadas no ensino dessa disciplina, buscando observar se essas ferramentas atendem a essa nova realidade do ensino inicial de programação para crianças e jovens. Observou-se que embora alguns dos projetos encontrados na literatura sejam extremamente exitosos, ainda se pode notar algumas deficiências.

PALAVRAS-CHAVE: ensino; programação; ferramentas;

ABSTRACT: The expansion of computer programming teaching to different audiences and areas is an eminent reality, and, at the same time, subjects related to Computer programming courses presents more learning difficulties by the students. This article aims to analyze and make a comparison with some tools that were proposed in previous papers and that are used in the Teaching of this subject to observe if these tools supply this new reality of the initial teaching of programming for children and young people. It was observed that although some of the projects found in the computer programming literature are extremely successful, some deficiencies can still be noticed.

KEYWORDS: teaching; programming; tools;

1 | INTRODUÇÃO

O ensino de programação tem se tornado cada vez mais importante no trajeto escolar das pessoas, não se limitando a acadêmicos dos cursos de computação, mas atingindo a todas as áreas e faixas etárias, em particular as crianças e jovens. Proporcionando um melhor entendimento e utilização das tecnologias computacionais atuais, e permitindo o desenvolvimento do raciocínio lógico e a capacidade de resolução de problemas.

Atualmente, em alguns países

desenvolvidos, como é o caso do Reino Unido, as disciplinas de programação já pertencem ao currículo escolar, abordando temas como pensamento computacional, lógica e algoritmos. (PAULA; VALENTE; BURN, 2014). Existem também várias iniciativas mundiais voltadas para disseminação do aprendizado de programação, como o *Code.org* e o *Khan Academy*.

No Brasil, apesar de ainda em fase inicial e apenas em alguns estados, já existem escolas adotando esse modelo. Recentemente, o prefeito da cidade de São Paulo anunciou que pretende implementar aulas de programação nas escolas administradas pelo município através de uma parceria entre a prefeitura e o Ministério da Educação para equipar as unidades escolares e promover o ensino de programação no currículo. (DORIA, 2017).

Entretanto, em vários trabalhos sobre aprendizagem de programação, foram encontrados exemplos da dificuldade de conhecer e se habituar aos ambientes e às linguagens, tarefa considerada difícil e que necessita de muito tempo e esforço por parte dos estudantes e, dos docentes.

As disciplinas de Lógica de Programação e Algoritmos e Programação podem ser ensinadas de diversas formas, cuja metodologia tradicional ainda prevalece, na qual o professor transmite conhecimentos e soluções de diversos problemas e o aluno recebe essas informações sem discutir e/ou criar as próprias soluções.

O uso de recursos como animações e vídeos torna-se facilitador do processo de aprendizagem, visto que a apresentação de conceitos abstratos é mais didática, melhorando a qualidade do material de apoio à aula (TAVARES et al., 2016).

Neste contexto, surgiram os seguintes questionamentos, que instigaram o desenvolvimento deste artigo: As ferramentas disponíveis e utilizadas atualmente estão aptas a atender essa nova demanda e público? Essas ferramentas possuem características necessárias para a aprendizagem inicial de programação?

O presente trabalho tem caráter qualitativo e realizou-se através de uma pesquisa bibliográfica sobre os softwares e ferramentas tecnológicas existentes que auxiliam os professores no ensino de algoritmos e programação. Dentre os softwares estudados, destacaram-se: *Feeper*, *WH-IDE*, *TutorICC*, *Portugol Studio*, *AAPW*, *TSTView*, *The Huxley*, *SOAP* e *Scratch*.

2 | FERRAMENTAS UTILIZADAS NO ENSINO DE PROGRAMAÇÃO

Esta seção aborda algumas ferramentas tecnológicas existentes que são utilizadas no apoio ao ensino das disciplinas de programação, abordando suas principais características, pontos positivos e negativos.

As ferramentas foram selecionadas com base em Revisões Sistemáticas da Literatura realizadas por Silva et al. (2015) e Ramos et al. (2015), que analisaram iniciativas brasileiras na criação de ambientes de aprendizagem de programação,

nos principais eventos brasileiros da área, que englobam o Congresso Brasileiro de Informática na Educação (CBIE) e a Revista Brasileira de Informática na Educação (RBIE). Foi excluída deste trabalho a avaliação de softwares criados antes de 2010.

2.1 Ambiente Virtual Feedback Personalizável (Feeper)

O Feeper é um ambiente virtual para apoiar o ensino/aprendizagem de programação no ensino superior. Utiliza um mecanismo para melhorar a comunicação entre aluno e professor, permitindo sinalizar partes do código fonte (ALVES; JAQUES, 2014).

A ferramenta funciona com o envio do código fonte produzido pelo estudante para validação de um Juiz On-line. Este então solicita ao banco de dados os testes existentes para o exercício que está sendo respondido. O Juiz On-line compara a resposta gerada pelo programa enviado com as respostas esperadas. Caso haja divergência, uma mensagem personalizada de alerta é emitida, contendo orientações para que este consiga resolver o problema gerado, do contrário, então o programa é aceito e a resposta de sucesso é emitida ao estudante.

O feedback fornecido pelo Juiz On-line foi personalizado para apresentar mensagens que indiquem onde aconteceu o erro, por que ele aconteceu e como corrigi-lo e para permitir a execução de diferentes tipos de teste no programa implementado (ALVES; JAQUES, 2014).

Através do *Feeper* o aluno pratica conceitos de programação orientada a objetos, utilizando a linguagem Java. Além da correção automática, a ferramenta emite dicas sobre os erros gerados pelo programa do aluno permitindo que o mesmo possa adicionar marcações em seu programa que serão exibidas ao professor. (ALVES; JAQUES, 2014).

2.2 Ambiente Personalizável para Aprendizado de Programação – WH-IDE

É um ambiente para ensino de programação de computadores, no qual os estudantes programam os algoritmos em português. O professor define todas as instruções que estarão à disposição dos aprendizes, personalizando, assim, a linguagem a ser utilizada e ignorando o máximo de detalhes técnicos. (FRANTZ; PONTES, 2014).

A ferramenta possui algumas limitações: ela não captura possíveis erros de sintaxe existentes no código, pelo fato de usar o compilador da linguagem Java para a geração do programa executável, além de não possuir depurador visual e área para exibição das variáveis criadas em memória (FRANTZ; PONTES, 2014).

2.3 Ambiente Interativo e Adaptável para Ensino de Programação – TUTORICC

O TutorICC é um ambiente interativo e adaptável utilizado para o ensino de

programação em Pascal. O conteúdo está dividido em níveis de dificuldade, permitindo ao estudante estabelecer o caminho mais conveniente para ele, dentro do conteúdo da disciplina (PÍCCOLO et al., 2010).

Há grande interatividade com o aprendiz, a cada passo é apresentado a ele um problema, para que construa um programa. Esses programas são corrigidos imediatamente pelo TutorICC, que recomenda ao estudante o melhor caminho a seguir dentro do conteúdo da disciplina.

A ferramenta foi desenvolvida utilizando a linguagem Java. A principal contribuição desta em relação aos demais trabalhos encontrados na literatura, é que permite que o professor personalize a apresentação visual em cada exemplo. Pode escolher posição, tamanho e cor de cada uma das variáveis, dispondo-as da forma que achar mais didática para captar a atenção dos alunos (PÍCCOLO et al., 2010).

2.4 Portugol Studio

O Portugol Studio é um Ambiente Integrado de Desenvolvimento (IDE) para auxiliar a aprendizagem de algoritmos para iniciantes. Foi desenvolvida no idioma português, facilitando o acesso ao público com baixa fluência no idioma inglês (NOSCHANG et al., 2014).

A IDE possibilita o desenvolvimento, a execução e a depuração de programas escritos em Portugol, sendo eficaz para iniciantes em programação, principalmente por utilizar o português como língua nativa (ANIDO, 2015).

O fato de ser um aplicativo para desktop dificulta um pouco para que o estudante possa praticar em casa, pois é necessário instalar as ferramentas do ambiente (editor, compilador, IDE, etc.), o que pode ser uma tarefa complexa (ANIDO, 2015).

2.5 Ambiente de Aprendizagem de Programação WEB – AAPW

O AAPW permite o aprendizado de conceitos envolvidos na programação *Web*, de forma simples e objetiva, podendo ser utilizada como recurso auxiliar em disciplinas de cursos que abordam desenvolvimento *Web*. (GOMES; D'EMERY; FILHO, 2014).

A ferramenta possui uma IDE de programação desenvolvida em Java, que é composta de barra de ferramentas; das telas de edição do programa fonte e do código compilado; de um browser para visualizar as páginas desenvolvidas; e de um console que exibe mensagens de erros presentes no código pertencente ao programa fonte.

A ferramenta trabalha com formulários, não necessita de *tags* no código e utiliza instruções simples com comandos em português. A IDE propicia ao usuário a visualização tanto do código construído em português quanto da codificação gerada, a qual pode ser interpretada em um browser. (GOMES; D'EMERY; FILHO, 2014).

2.6 Ferramenta de Acompanhamento de Exercícios de Programação – TSTVIEW

O TSTView permite acompanhar o desempenho dos estudantes em um certo período de tempo, além de monitorar em tempo real o desenvolvimento dos estudantes frente aos exercícios planejados possibilitando detectar, com mais facilidade e eficiência, dificuldades individuais dos estudantes, dificuldades comuns a grupos de estudantes e dificuldades gerais da turma (GAUDENCIO, 2013).

Os estudantes respondem aos exercícios propostos utilizando a linguagem de programação Python e submetem a um testador automático que aplica os testes criados pelos professores para as questões propostas. Os dados de controle e os resultados dos testes aplicados as submissões compõem a sua base de dados.

A ferramenta possui dois módulos principais, o do professor e o do aluno. O primeiro exibe uma lista dos aprendizes que estão realizando alguma atividade no momento da consulta. Já o segundo, exibe um relatório histórico individual para cada um deles (GAUDENCIO, 2013).

Ao submeter uma solução, o estudante recebe feedback na forma de resultados de testes sobre seu código, podendo visualizar que tipos de erros ele cometeu e o número de casos de teste em que falhou.

2.7 Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores - The Huxley

The Huxley é uma ferramenta *web* que permite aos estudantes responderem a exercícios de uma base de dados com centenas de problemas e submeterem às respostas através de código em diversas linguagens de programação.

O ambiente corrige os exercícios automaticamente através de análise sintática do código e testes de aceitação e emite feedback. O professor tem acesso à quantidade de problemas resolvidos, porcentagem de acertos/erros, tipos de problemas com mais erros, detecção de plágio e erros específicos de cada aluno (PAES, 2013).

Através do ambiente, o docente faz uma avaliação personalizada do aluno, baseada na identificação de erros específicos durante as avaliações. A definição e correção de exercícios e provas, verificação de plágio, definição de notas e publicação dos resultados pode ser feita automaticamente através do The Huxley (PAES, 2013).

2.8 Sistema On-Line de Atividades de Programação – SOAP

É um sistema *web* que possibilita ao professor disponibilizar tarefas para as turmas e, ao aluno, realizar submissões de exercícios. Os exercícios submetidos, que são programas de computador em Linguagem C, são compilados e executados em um servidor remoto, gerando relatórios que são disponibilizados aos professores.

As atividades são processadas pelos núcleos (Núcleo de Avaliação Diagnóstica NAD) e Núcleo de Avaliação Formativa (NAF). O NAD processa essas atividades,

gerando planilhas e mapas de perfis de aprendizagem, o NAF, com base no diagnóstico fornecido pelo NAD, recomenda atividades diretamente para que os estudantes com dificuldades possam melhorar os desempenhos (OLIVEIRA; OLIVEIRA, 2014).

O software combina técnicas de reconhecimento de padrões para classificar alunos e recomendar-lhes atividades conforme os seus perfis e possibilita ao docente realizar melhor gestão da aprendizagem individual de seus estudantes (OLIVEIRA; OLIVEIRA, 2014).

2.9 Scratch

O ambiente de programação *Scratch* traz linguagem que contribui para a aprendizagem de programação através da criação de histórias, animações, simulações e outros produtos multimídia, arrastando e agrupando diferentes conjuntos de blocos de comandos coloridos (MARQUES, 2013).

Pelo fato de os comandos *Scratch* se basearem em uma estrutura de blocos de montar, o aprendiz pode concentrar esforço na busca pela construção do algoritmo, não se preocupando com erros de sintaxe. Desta maneira, pode-se afirmar que o caráter mais didático do *Scratch* projeta no aluno a possibilidade de ele se concentrar no exercício do pensamento algorítmico e na criatividade durante a construção das soluções (SCAICO et al., 2013).

3 | COMPARATIVO DAS FERRAMENTAS

Os *softwares* apresentados compartilham objetivos, como desenvolver a criatividade, o raciocínio, a descoberta e interação social. Todos exploram as interfaces visuais como meio de motivar a aprendizagem dos conceitos de programação. O que os difere, entretanto, é que uns são apenas IDE, onde o aluno pode escrever e testar códigos, alguns são voltados para o ensino de lógica de programação e outros para conceitos mais avançados de programação, como orientação a objetos.

Para uma efetiva comparação das ferramentas analisadas, foram utilizados alguns critérios como a forma de interação do usuário com o ambiente, a plataforma de execução da ferramenta, as linguagens de programação suportadas e se está disponível no idioma Português.

No Quadro 1, pode-se observar a relação das ferramentas e os critérios que foram utilizados na análise.

Ferramenta	Forma de Interação	Plataforma	Linguagem de Programação	Disponível em Português
<i>Feeper</i>	Textual	<i>Web</i>	Java	Sim
<i>WH-IDE</i>	Textual	Java	Portugol	Sim
<i>TutorICC</i>	Visual e Textual	Java	Pascal	Sim
<i>Portugol Studio</i>	Visual e Textual	Java	Portugol	Sim
<i>AAPW</i>	Visual e Textual	Java	Portugol	Sim
<i>TSTView</i>	Visual e Textual	<i>Web</i>	<i>Phyton</i>	Sim
<i>The Huxley</i>	Textual	<i>Web</i>	Várias	Sim
<i>SOAP</i>	Visual e Textual	<i>Web</i>	C	Sim
<i>Scratch</i>	Visual e Textual	<i>Web</i>	Blocos	Sim

Quadro 1 – Características Técnicas das Ferramentas.

Fonte: Elaborado pelo autor, 2017.

Com exceção do *Scratch*, que utiliza blocos, todos os demais utilizam uma ou várias linguagens de programação. A maioria deles é composta de recursos visuais e textuais e podem ser executados na *web* através de um *browser*. Embora, todos os *softwares* apresentados sejam indicados para iniciantes, apenas três deles utilizam a linguagem *Portugol*, que é um fator considerado importante em disciplinas introdutórias.

Na literatura sobre aprendizagem de programação, encontram-se relatos de que o processo de conhecer e se habituar aos ambientes e às linguagens é uma tarefa árdua, com grande dispendiosidade de tempo por parte dos estudantes e, por vezes, dos docentes ou tutores (RIBEIRO, 2015).

No que se refere as funcionalidades implementadas, verificou-se acerca da possibilidade do usuário visualizar os valores das variáveis durante a execução de um algoritmo, o uso de animações e vídeos na explicação dos conceitos e técnicas utilizadas, a forma utilizada para emitir *feedback* e também se possuem ou não características de adaptatividade.

No Quadro 2, tem-se a comparação com relação às funcionalidades disponíveis:

Ferramenta	Visualização de variáveis	Animações	Vídeos	<i>Feedback</i>	Adaptatividade
<i>Feeper</i>	NÃO	NÃO	NÃO	Manual	NÃO
<i>WH-IDE</i>	NÃO	NÃO	NÃO	Manual	NÃO
<i>TutorICC</i>	SIM	SIM	NÃO	Automático	SIM
<i>Portugol Studio</i>	SIM	NÃO	NÃO	Manual	NÃO
<i>AAPW</i>	NÃO	NÃO	NÃO	Manual	NÃO
<i>TSTView</i>	NÃO	NÃO	NÃO	Automático	NÃO
<i>The Huxley</i>	NÃO	NÃO	NÃO	Automático	NÃO
<i>SOAP</i>	NÃO	NÃO	NÃO	Automático	NÃO
<i>Scratch</i>	NÃO	SIM	SIM	Automático	NÃO

Quadro 2 – Funcionalidades.

Fonte: Elaborado pelo autor, 2017.

Pode-se perceber que a maioria das ferramentas não utiliza recursos de animações e vídeos para apresentação de conteúdo, explicação dos códigos e funcionamento das variáveis.

A animação de algoritmos é importante, pois, muitas vezes, o professor tem a necessidade de usar representações visuais para ajudar os alunos a entenderem os algoritmos, o que na prática se concretiza, muitas vezes, mostrando o comportamento de programas (TAVARES et al., 2016).

Outra característica apresentada pela maioria dos *softwares* estudados é o *feedback* automático. Segundo Tavares et al. (2016), *feedback* imediato é crucial para a aquisição de conhecimento, independentemente da estratégia de aprendizagem adotada, pois motiva os alunos. Por outro lado, o *feedback* individual pode consumir muito tempo do professor com o risco de que os estudantes possam não se beneficiar dele no devido tempo (QUEIRÓS; LEAL, 2015).

Com relação à adaptatividade, apenas a ferramenta TutorICC apresentou essa funcionalidade, no qual o ritmo de aprendizagem se adapta ao perfil do aluno, podendo ser mais rápido ou mais lento, dependendo de sua maior ou menor facilidade em aprender programação.

A adaptatividade tem como propósito ajustar-se a diferentes níveis do educando, com diferentes conhecimentos prévios, facilitando o aprendizado e reforçando os conceitos apresentados nas aulas expositivas (BRAGA, 2015).

Segundo Gomes et al. (2008), para se atingir a capacidade de resolução de problemas são necessárias três fases: (1) resolução de problemas de diversos domínios, não utilizando algoritmos ou programação; em seguida, (2) uma amostra da utilidade da programação com aplicação dos conhecimentos adquiridos na fase anterior; e, finalmente, (3) passando para a construção dos algoritmos.

Na fase de resolução de problemas, a preocupação maior é com o tipo de problemas que serão apresentados. A utilidade da programação tem por objetivo mostrar aos estudantes para que serve e como se faz um programa. E na fase de construção os alunos teriam que analisar a resolução de problemas através de programas de computador e também propor e testar novas soluções.

O Quadro 3 analisa se as ferramentas estudadas possibilitam a aplicação dessas fases:

Ferramenta	Resolução de problemas de domínios diversos	Utilidade da programação	Construção dos algoritmos
<i>Feeper</i>	NÃO	NÃO	SIM
<i>WH-IDE</i>	NÃO	SIM	SIM
<i>TutorICC</i>	NÃO	NÃO	SIM
<i>Portugol Studio</i>	NÃO	SIM	SIM
<i>AAPW</i>	NÃO	SIM	SIM
<i>TSTView</i>	NÃO	NÃO	SIM
<i>The Huxley</i>	NÃO	NÃO	SIM

<i>SOAP</i>	NÃO	SIM	SIM
<i>Scratch</i>	SIM	SIM	NÃO

Quadro 3 – Competências trabalhadas (GOMES et al., 2008).

Fonte: Elaborado pelo autor, 2017.

Observa-se que a grande maioria das ferramentas estudadas prioriza a etapa de construção dos algoritmos, sem a devida importância às competências anteriores.

Em disciplinas iniciais de programação o foco principal deve ser a capacidade de resolução de problemas, e o uso da linguagem de programação fica para fases posteriores, como meio de concretizar essa resolução através de um algoritmo (BORGES, 2016).

Embora todos os *softwares* apresentados sejam indicados para iniciantes, dentre as ferramentas citadas, a que mais atende a critérios importantes para aprendizagem inicial de programação e discutidos neste trabalho é o *Portugol Studio*.

A ferramenta *Portugol Studio* prioriza o uso do idioma português, e possui diversos elementos presentes nas IDE profissionais, mas preservando sempre o enfoque na aprendizagem dos novatos em programação. Alguns elementos da interface auxiliam o docente a explicar processos como a depuração.

Embora alguns dos projetos encontrados na literatura sejam extremamente exitosos, ainda se pode notar algumas deficiências, como: (1) foco na capacidade de resolução de problemas, (2) uso de animações e vídeos para explicação de conteúdos complexos e (3) utilização do *Portugol*.

4 | CONCLUSÃO

O ensino de algoritmos e programação é um dos grandes desafios na área de ensino de computação, pois são grandes as dificuldades encontradas pelos alunos.

Considerando a evolução tecnológica e a visível necessidade de expansão desses conteúdos para além dos cursos de computação e uma possível inclusão dessa disciplina no currículo escolar brasileiro, é necessário verificarmos se as ferramentas utilizadas atendem as necessidades de aprendizagem dos estudantes.

Segundo Anido (2015), a primeira e mais importante barreira para estudantes brasileiros usarem as ferramentas existentes e que elas foram desenvolvidas para um público que tem inglês como língua oficial e, apesar dos meios de internacionalização, a maioria dos ambientes exige algum conhecimento da língua inglesa, e não apenas referente às palavras-chave da linguagem de programação.

Se adaptar a realidade desta nova geração, utilizando um ambiente mais lúdico e que considere as especificidades da aprendizagem dos estudantes são características indispensáveis as instituições de ensino. Portanto, é muito importante o

desenvolvimento de ferramentas tecnológicas que busquem despertar o interesse do aluno, assim como facilitar o entendimento dos conteúdos e da lógica de programação.

Espera-se que este trabalho incentive a reflexão acerca de metodologias inovadoras e utilização efetiva da tecnologia a favor da educação. Tornar o aluno mais participativo, motivado e autônomo da sua aprendizagem é um desafio diário a ser conquistado.

REFERÊNCIAS

ALVES, F. P.; JAQUES, P. **Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação.** In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2014. p. 51.

ANIDO, R. **Saci-ainda outro ambiente para o ensino de programação.** In: Anais do XXIII WEI - Workshop sobre Educação em Computação. 2015.

BORGES, M. J. M. **Ferramenta Multimédia de Apoio à Programação Algorítmica e Programação Pascal.** Nº de Folhas 217, Dissertação de Mestrado em Tecnologias de Informação e Comunicação. Universidade de Trás-os-Montes e Alto Douro, 2016.

BRAGA, O. R. M. **Objeto de aprendizagem adaptativo no ensino da lógica booleana.** 2015. Nº de Folhas 93, Dissertação de Mestrado em Engenharia Elétrica. Universidade Presbiteriana Mackenzie, São Paulo, 2015.

DORIA, J. [**comentário pessoal**]. Facebook. 10 mar 2017. Disponível em: <<https://www.facebook.com/jdoriajr>>. Acesso em: 11 mar. 2017.

FRANTZ, W. L. S.; PONTES, H. P. **Um Ambiente de Desenvolvimento Personalizável para o Ensino de Programação.** In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2014.

GAUDENCIO, M. et al. **Eu Sei o que Vocês Fizeram (Agora e) na Aula Passada: o TSTView no Acompanhamento de Exercícios de Programação.** In: Anais do Simpósio Brasileiro de Informática na Educação. 2013. p. 204

GOMES, A. et al. **Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores.** In: Educação, Formação & Tecnologias; vol.1(1), pp. 93-103, 2008.

GOMES, M. E. N.; D'EMERY, R. A.; FILHO, G. A. A. C. **AAPW: Uma Ferramenta para facilitar o Aprendizado de Programação Web.** In: Anais do XXII WEI - Workshop sobre Educação em Computação. 2014.

MARQUES, M. C. P. O. **O ensino da programação no desenvolvimento de jogos através do ambiente Scratch.** Nº de Folhas 177. Dissertação do Mestrado em Ensino de Informática. Universidade do Minho, 2013.

NOSCHANG, L. F. et al. **Portugol studio: Uma ide para iniciantes em programação.** In: Workshop sobre Educação em Computação. 2014. p. 1287-1296.

OLIVEIRA, M.; OLIVEIRA, E. **Metodologia de Diagnóstico e Regulação de Componentes de Habilidades da Aprendizagem de Programação.** In: Anais do XXII WEI - Workshop sobre Educação em Computação. 2014.

PAES, R. B. et al. **Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores**. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2013.

PAULA, B. H.; VALENTE, J. A.; BURN, A. **O uso de jogos digitais para o desenvolvimento do currículo para a Educação Computacional na Inglaterra**. Currículo sem Fronteiras, v. 14, n. 3, p. 46-71, 2014.

PÍCCOLO, H. L. et al. **Ambiente Interativo e Adaptável para ensino de Programação**. In: Anais do Simpósio Brasileiro de Informática na Educação. 2010.

QUEIRÓS, R.; LEAL, J. P. **Ensemble: An Innovative Approach to Practice Computer Programming**. In: QUEIRÓS, R., Innovative Teaching Strategies and New Learning Paradigms in Computer Programming. Hershey: Information Science, 2015. cap. 9.

RAMOS, V. et al. **A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos brasileiros**. In: Anais do Simpósio Brasileiro de Informática na Educação. 2015. p. 318.

RIBEIRO, R. S. **Construção e uso de ambiente visual para o ensino de programação introdutória**. Nº de Folhas 59. Dissertação do Mestrado em Ciência da Computação. Universidade de São Paulo, 2015.

SCAICO, P. D. et al. **Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a Linguagem Scratch**. Revista Brasileira de Informática na Educação, v. 21, n. 02, p. 92, 2013.

SILVA, P. et al. **Um Mapeamento Sistemático sobre Iniciativas Brasileiras em Ambientes de Ensino de Programação**. In: Anais do Simpósio Brasileiro de Informática na Educação. 2015. p. 367.

TAVARES, P. et al. **Técnicas para aumentar o envolvimento dos alunos na aprendizagem da programação**. In: VII Congresso Mundial de Estilos de Aprendizagem. Bragança, 2016. p. 1565-1577.

MODOS CONTEMPORÂNEOS DE APRENDIZADO E CONSTRUÇÃO DO CONHECIMENTO: REFLEXÕES SOBRE O ENSINO DE TEORIA DA COMPUTAÇÃO PARA SISTEMAS DE INFORMAÇÃO

Isabel Cafezeiro

Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brasil

Leonardo Cruz da Costa

Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói – RJ – Brasil

Ricardo Kubrusly

Programa de Pós-Graduação em História das Ciências e das Técnicas e Epistemologia – Universidade Federal do Rio de Janeiro
Rio de Janeiro – RJ – Brasil

RESUMO: Este artigo descreve a experiência de concepção e implantação de uma disciplina em curso de Sistemas de Informação que procura abordar o conteúdo de Teoria da Computação, um tema que vem sendo ao longo dos anos preterido nos cursos de Ciências da Computação e que não faz parte dos currículos usuais em Sistemas de Informação. Argumentamos que uma proposta ambientada na interdisciplinaridade através da abordagem sociotécnica pode reposicionar estes conteúdos colocando em evidência uma grande contribuição na compreensão da configuração atual dos sistemas computacionais através do seu processo histórico de construção.

PALAVRAS-CHAVE: TEORIA DA COMPUTAÇÃO, ABORDAGEM SOCIOTÉCNICA, ENSINO EM COMPUTAÇÃO

ABSTRACT: This article describes the experience of designing and implementing a discipline in the under graduation course in Information Systems that seeks to address the contents of the Theory of Computation, a topic that has been, over the years, deprecated in Computer Science courses and that is not part the usual curricula in Information Systems. We argue that a proposal based on interdisciplinarity through sociotechnical approach can reframe these contents bringing out a great contribution to the understanding of the current configuration of computer systems through its historical process of construction.

KEYWORDS: Theory of Computation, Sociotechnical Approach, Teaching in Computing

1 | AS DUAS CULTURAS

Nossas análises têm por base a observação dos cinco primeiros anos de operação do curso de Sistemas de Informação da Universidade Federal Fluminense (UFF), na cidade de Niterói, RJ. Arriscamos um diagnóstico ao problema do alto grau de retenção de estudantes nos primeiros três períodos do curso. Trata-se justamente do momento em que se concentram as disciplinas de base matemática, como

cálculo, lógica e matemática discreta, que apresentam um conteúdo não familiar tanto para o estudante recém-saído do ensino médio quanto para os que já possuem alguma experiência no mercado de trabalho em Sistemas de Informação e retornam à universidade em busca da titulação. Completam o quadro de retenção as disciplinas iniciais de programação, também ministradas nos primeiros períodos. Neste caso específico, precisamos notar que o mecanismo de raciocínio que esta disciplina exige difere da práxis do ensino médio, onde os conteúdos considerados exatos são apresentados em uma abordagem predominantemente teorematizada (isto é, que não deixa aparentes processo e motivação de construção das formulas, focando-se o modo de aplicação, e, quando muito, justificativas dedutivas). Diferentemente, a atividade de programação exige um pensamento algorítmico, centrado no “como fazer”, raciocínio aderente ao problema. Dessa forma, enquanto que uma abordagem teorematizada permite a abstração sobre o problema que está sendo tratado, uma abordagem problemática se mostra essencialmente conectada ao problema e ao mundo onde este problema é enunciado. Sobre a diferença entre as abordagens teorematizada e problemática, ver Deleuze & Guatarri (1997, p. 26).

A retenção não é uma característica exclusiva dos cursos de Sistemas de Informação. Ela também se faz presente nos cursos de Ciências da Computação, referente a este mesmo conjunto de disciplinas. De modo geral em computação, retenção, dificuldade de aprendizagem e evasão vem demandando atenção nos fóruns de discussão de currículos e metodologias de ensino. Como um exemplo recente, remetemos ao Curso de Qualidade da Sociedade Brasileira da Computação (CQ, 2015), cujo tema foi “Evasão e estratégias para permanência estudantil” (http://csbc2015.cin.ufpe.br/eventos_descricao/8). Percebemos também que o tema é recorrente nas diversas edições do WEI, um workshop anualmente promovido pela Sociedade Brasileira de Computação, dedicado ao ensino da Informática. No ano de 2015, por exemplo, podemos ver Souza, Morais, & Junior (2015). Assim, surpreende que este problema não tenha ainda encontrado solução satisfatória em cursos de longa tradição como os de Ciência da Computação (o curso de graduação em Ciências da Computação da UFF, em Niterói, foi implantado em 1986, mas os primeiros do Brasil datam de 1969). Porém é possível que os cursos de Sistemas de Informação, por fundamentarem-se em uma abordagem diferente e recente quando comparada ao tradicional enfoque em Ciências da Computação, sejam capazes de atrair atenção para uma nova dinâmica de compreensão do conhecimento, algo que pode ser elaborado de maneira a sugerir transformações e metodologias inovadoras no ensino tanto em Sistemas de Informação quanto nas Ciências da Computação. Trata-se da abordagem sociotécnica: a compreensão de que o conhecimento se constrói com a co-participação das diversas áreas de conhecimento (Boscarioli, Araujo, Maciel, 2016).

Desde os primeiros contatos com as apresentações dos cursos de Sistemas de Informação, o aluno percebe que se trata de um campo de conhecimento não completamente incluso na área de Ciências da Computação. As páginas de

apresentação dos cursos costumam indicar uma forte interdisciplinaridade, a computação situada como “atividade meio”, o que, no jargão da área, significa a intenção de capacitar o profissional para a utilização instrumental da tecnologia na solução criativa de problemas diversos. Esta maneira de situar a computação encontrou suporte nos documentos referenciais da área, como o currículo de referência da Sociedade Brasileira de Computação (SBC, 1999). A multiplicidade de campos de conhecimentos manifestou-se na grade curricular do Curso de Sistemas de Informação da UFF pela colaboração com diversos departamentos (direito, sociologia, psicologia, administração, ciências da informação) através do oferecimento de disciplinas em turmas mistas (não exclusivas a estudantes de Sistemas de Informação), o que veio a fortalecer o diálogo dos estudantes com as demais áreas. Fora das salas de aula, o curso contou com o apoio de um Núcleo Docente Estruturante, e com o Colegiado de Curso, que, atentos ao caráter híbrido requerido na formação do profissional de Sistemas de Informação, contribuíram para oferecer com sucesso a diversificação do corpo de conhecimentos presente na grade curricular.

O que não se verificou, entretanto, foi a efetivação da abordagem sociotécnica no interior de cada disciplina, e principalmente daquelas disciplinas consideradas “exatas”, como as matemáticas. O estudante se deparou com uma perceptível fronteira: as disciplinas ministradas pelos departamentos de computação e matemáticas são apresentadas como “técnicas” (ou “exatas”). O aprendizado é centrado em mecanismos abstratos ou metodologias validadas no cenário acadêmico mundial e as avaliações focalizam a aplicação destes mecanismos. As outras disciplinas, principalmente as ministradas nos departamentos das áreas humanas são apresentadas como disciplinas “humanísticas”, “não-técnicas”. Neste caso, o debate e a expressão oral e escrita são incentivados, e as avaliações não seguem necessariamente o formato das provas tradicionais e a abordagem demanda um posicionamento crítico pessoal que se afasta de um caráter supostamente neutro, usual no ensino das técnicas. Sem que se efetive o diálogo entre estas abordagens, modo de apresentação, conteúdos, avaliações, tudo se altera quando se passa de um universo para o outro. É uma diferença que acompanha a configuração dos saberes modernos, e que na UFF se faz visível também espacialmente, na separação dos campi universitários (as disciplinas humanísticas são oferecidas no Campus do Gragoatá, onde se concentram os departamentos e cursos correspondentes. Por outro lado, o Instituto de Computação se localiza no Campus da Praia Vermelha, e lá são ministradas as disciplinas da matemática e da computação). Latour (2014, tradução nossa) observou:

[P]arece que estamos sofrendo de uma divisão do trabalho intelectual: se você aprendeu história, ciências sociais, literatura, direito, arte ou qualquer ramo do que é chamado de “humanidades”, tenho certeza que você aprendeu um monte de coisas, mas não necessariamente sobre como a tecnologia e a ciência têm interferido em todos esses vários campos; por outro lado, se você assistiu aulas ou teve formação em engenharia, informática, ciências naturais ou negócios aplicados, não tenho a certeza que você vai ter aprendido algo sobre quando, onde e por

quem esses elementos de conhecimento foram produzidos e que eles relações entretêm com a história, literatura, arte, política ou ciências sociais. (...) Qualquer que seja o campo de onde você vem, há chances de que você tenha percebido de que existe uma divisão entre aqueles que vêm “das letras” e aqueles que vêm “das ciências”. E não apenas uma divisão, mas muitas vezes, infelizmente, alguma forma de condescendência, mesmo de rancor contra os do outro lado: você poderia ter sido tratado como um “nerd analfabeto” por algum estudioso literário brilhante que, por sua vez, poderia ter sido tratado como um “idiota romântico” por alguns colegas do sério departamento de matemática.

Diante desse quadro de cisão entre os conhecimentos, recusamos qualquer diagnóstico referente à retenção, evasão e dificuldades de aprendizagem que se situe na falta de preparo dos estudantes de Sistemas de Informação, e arriscamos dizer que o baixo desempenho nas disciplinas das ditas “ciências exatas” nasce no conflito entre a expectativa por um conhecimento híbrido e problematizador e o defrontamento com um conhecimento que se pretende puramente técnico. Este conflito é desmotivador, uma vez que não conduz à compreensão da utilidade destes conhecimentos na profissão. Traz para a reflexão o contraste com o alto desempenho nas ditas “humanísticas”, um fato que, ignorando-se a familiaridade destes conteúdos a estes estudantes desde o ensino médio, ainda desencadeia o argumento de que as chamadas “humanísticas” são “mais fáceis” do que as ditas “exatas”. A este impasse, referindo-se especificamente à Engenharia de Software, Cukierman et al., em 2007 já haviam proposto um caminho:

A saída para os impasses criados pela separação entre o técnico e o social consiste em mudar o ângulo de aproximação e, assim, percebê-los por um novo enquadramento. Resumindo brevemente, um enquadramento em que o técnico e o social constituem um movimento de “co-modificação”, somente percebido por uma abordagem concomitantemente social e técnica, por um olhar sociotécnico. Um olhar que busca apreender a ES sem fragmentá-la em “fatores ou aspectos técnicos” de um lado, e “fatores ou aspectos não técnicos” de outro, sem fatorá-la em quaisquer outras dualidades (“fatores técnicos” versus “fatores humanos, organizacionais, éticos, políticos, sociais, etc.”) que terminem por desfigurar o “pano sem costura” que imbrica na ES o técnico e o social em um mesmo e indivisível tecido.

2 | TEORIA DA COMPUTAÇÃO E OS CURRÍCULOS DE SISTEMAS DE INFORMAÇÃO

Se por um lado a saída apontada por Cukierman et al. encontra acolhimento em áreas como a Engenharia de Software, Interação Humano-Computador, Jogos, e outras especialidades que, de uma forma ou de outra, admitem algum diálogo com a sociedade, o mesmo não se verifica com a área da Teoria da Computação. Aqui, nos referimos precisamente aos conhecimentos relacionadas ao estudo da Computabilidade.

Muitas questões abordadas no campo da Computabilidade se estabeleceram antes da construção dos primeiros computadores, em um ambiente de busca pela compreensão da capacidade de expressão formal (sistemas formais). Naquele momento, início do século XX, percebiam-se certas idiossincrasias, como os paradoxos e enunciados não provados, que não eram bem-vindas numa ciência que buscava controle, exatidão e objetividade. Mas, em consonância com a concepção da ciência moderna de uma matemática completa e totalizante, buscavam-se no seio da própria matemática, explicações para essas esquisitices, dando margem aos programas de fundamentação da matemática do início do século XX (Cafezeiro et al, 2010). Resultados concebidos no contexto da afirmação da matemática enquanto disciplina autônoma migraram para a computação como fundamentação para o campo incipiente. Dali para a década de 1960, já com cerca de 15 anos de operação dos computadores, outros personagens, os fundadores da computação, reconheceram no estudo de sistemas formais e suas propriedades o coração da computação e a justificativa para a institucionalização do campo. Daí se compreende uma certa resistência para a abordagem híbrida: herança da tradição matemática como um campo purificado, que não se contamina com os demais, fortalecimento do caráter científico do novo campo a partir da aderência a um campo já estabelecido. Embora alguns cientistas, como Alan Perlis, insistissem na interação com saberes relacionados à gestão e à informação, outros pioneiros do campo da computação, como Edsger Dijkstra, Tony Hoare, Nicklaus Wirth, apostaram no caráter “exato” e se opuseram à institucionalização do campo em co-construção com outros saberes “não exatos”.

Argumentos em favor das abordagens formais podem ser lidos nos relatórios das famosas conferências da OTAN (Organização Militar do Atlântico Norte) de 1968 e 1969 (NATO 1969,1970), onde, sob o pretexto de discutir a “crise do *software*”, terminou-se por delimitar os contornos do que se passou a chamar de “Ciência da Computação”. O trecho a seguir mostra o quão frágil era, naquele momento a compreensão do campo que vinha então se configurando (NATO,1970, tradução nossa,p.49):

Strachey: (...). Acho que devemos nos perguntar seriamente a seguinte questão: é a computação uma ciência? (...) Até termos um corpo suficiente de tópicos que sejam importantes, relevantes e bem desenvolvidos, não podemos chamar o assunto de “ciência”.

Woodger: Eu me pergunto com o se parece que um princípio básico em informática, se eu me deparasse com um. Sério! Em que termos ele poderia ser expresso? Não é bom usar termos inexplicáveis como “língua” ou “compilador”; temos muito poucos termos claros em que para expressar qualquer princípio fundamental da ciência da computação. Será que algum professor aqui pode me dizer um único princípio básico?

Grande parte das discussões girava em torno de um impasse entre “teoria” e “prática”. Os defensores da “teoria” encontravam nos métodos matemáticos a garantia de um campo bem fundamentado e uma atividade de programação livre de erros.

Aqui Hoare e Dijkstra, em divergência com Perlis, ironizam os testes informais em programas e argumentam em favor das provas formais (NATO, 1970,p.16):

Hoare: Pode-se, muito facilmente, construir provas convincentes da futilidade de testes exaustivos em um programa e até mesmo de testes por amostragem. (...) Teste dos casos base poderiam, por vezes, serem automatizados. Atualmente, isto é basicamente teoria (...) Esta área de trabalho teórico parece mostrar uma possibilidade de resultados práticos, embora provar correção seja um processo trabalhoso e caro. Talvez não seja um luxo para certas áreas cruciais de um programa.

Perlis: Grande parte da complexidade do programa é espúria e uma série de casos de teste devidamente estudados esgota o problema dos testes. O problema consiste em isolar casos de teste adequados, e não provar o algoritmo, pois isto [a correção] se segue após a escolha dos casos de teste adequados.

Dijkstra: Teste mostra a presença, não a ausência de bugs.

Possivelmente em função do forte apelo dos pioneiros que argumentavam em favor de métodos formais, questões relacionadas à decidibilidade ganharam importância nos currículos da computação, no sentido da compreensão das capacidades e limitações das abordagens formais. No Brasil, a maioria dos cursos em computação que começaram a surgir na década de 1970 (as exceções são Ciências da Computação da UNICAMP e Processamento de Dados da UFBA que iniciaram em 1969) contava com disciplinas referentes tanto à parte de linguagens formais e autômatos, onde aparecem questões de decidibilidade, quanto aos conteúdos relacionados à enumerabilidade, Tese de Church, Máquina Universal. Mas, da década de 1970 aos dias de hoje, estes conhecimentos vêm perdendo espaço nos currículos da computação, e são praticamente inexistentes nos cursos de Sistemas de Informação, um campo que toma como periférica a importância destes saberes na formação profissional.

Ao mesmo tempo que verificamos que a abordagem usual da computabilidade é extremamente matematizada, centrada na apresentação de resultados e provas formais, e portanto, se afasta do enfoque sociotécnico buscado nos cursos de Sistemas de Informação, vemos também que a ausência destes assuntos causa um lapso nesta formação no que tange à compreensão das possibilidades de sistemas computacionais, a configuração atual de sistemas computacionais e suas perspectivas futuras e também a compreensão desenvolvimento da computação e do próprio computador. Acrescentamos ainda que a presença ubíqua da computação em nossas vidas vem demandando uma compreensão em termos teóricos (no suporte matemático), filosóficos e sociológicos que capacite os profissionais da computação a lidar com a nova dinâmica social fortemente amparada nas tecnologias. Por isso, convém refletir sobre a concepção de uma maneira sociotécnica de trabalhar estes conhecimentos. Longe de pretender desenvolver uma abordagem “suavizada” da abordagem formal, o que está aqui em questão é o resgate destes conteúdos devido à sua importância nas formações em computação. A estratégia é desfazer a

cisão entre os resultados matemáticos e as questões do mundo que lhes motivaram, buscar maneiras de apresentar conteúdos formais das disciplinas de Computabilidade inseridos na conjuntura que suscitou a sua formulação para que sejam compreendidos mais amplamente. Por exemplo: os números transfinitos de Cantor, suas propostas de compreensão do infinito, e suas percepções sobre cardinalidades nos parecem ainda hoje esquisitos porque fugiam do que era considerado “intuitivo”. No entanto, as formulações de Cantor não buscavam refletir diretamente o que a grande maioria considerava “intuitivo”, mas eram inspiradas nos dilemas religiosos de caráter muito intimista que Cantor vivia (Junior, 2006). Portanto, sob a luz destas inquietações as construções matemáticas de Cantor se deixam explicar mais claramente. Da mesma forma, se percebermos os movimentos artísticos que se desenrolavam na Europa dos anos 1930, que tem uma das primeiras expressões na revolta do dramaturgo Antonin Artaud com relação à compreensão da potência da representação (Cafezeiro, Gadelha & Chaitin, 2016), podemos verificar que as indagações formuladas pelos matemáticos com relação à representação matemática acompanhavam uma tendência daquela sociedade naquele momento. A Europa, em consequência da guerra recente, do autoritarismo dos governos e na iminência de uma nova guerra vivia um momento de crise nas representações, incluindo-se aí as representações políticas. Isto, embora seja claro em estudos sociológicos, artísticos e filosóficos, para os matemáticos é um ponto que ainda necessita ser debatido. Alguns cientistas matemáticos preferem considerar os resultados matemáticos como questões livres de qualquer influência do mundo em que viviam os matemáticos. Seriam questões limitadas ao escopo da própria disciplina: “Muitas referências ao teorema da incompletude fora do campo da lógica formal são obviamente absurdas e parecem basear-se em mal-entendidos graves ou algum processo de livre associação” (Franzen, 2005). Por fim, cabe mencionar a própria construção do computador, cuja compreensão não se dissocia da conjuntura da segunda guerra (Dyson, 2012). Quando acompanhados das circunstâncias complexas em que foram concebidos, aquilo que vem sendo usualmente apresentado como sendo “essencialmente matemático” ou “essencialmente técnico” passa a mostrar um caráter inevitavelmente interdisciplinar. Por esse caminho, abre-se espaço a novas traduções, interpretações e conexões com questões da contemporaneidade. Segue-se um exemplo.

Em meados do ano de 2013 começaram a aparecer na mídia carioca diversas notícias referentes a um novo campo da computação denominado “Computação Humana” (Setti, 2013). As notícias enfatizavam o trabalho do cientista guatemalteco Luis von Ahn, que concebeu as *captchas* (Ahn et al., 2004) e as *recaptchas* (Ahn et al., 2008), e consta como um pioneiro da Computação Humana. O termo se refere a sistemas computacionais cada vez mais populares entre nós, que se utilizam do processamento humano para efetuar pequenas tarefas componentes de um objetivo maior (Law & Ahn, 2011). O humano é convocado a participar em situações que a máquina não seria capaz de resolver sozinha por serem incomputáveis ou computacionalmente inviáveis.

Esta situação nos remete ao pioneirismo de Alan Turing, ao conceber em 1936, o modelo formal do computador. Poucos anos mais tarde, em 1938, em sua tese de doutorado (Turing, 1938), Turing vinha trabalhando numa hierarquia de sistemas formais. Partindo de um sistema (incompleto) ele descreveu um processo de “completação” de sistemas inicialmente sugerido por Gödel em seu artigo de 1931 (Gödel, 1965). Aqui, os termos “completo” e “incompleto” referem-se à sistemas formais: um sistema formal é completo se a qualquer enunciado formulado no sistema corresponda uma prova de sua veracidade ou falsidade. Remetendo-se aos Teoremas da Incompletude de Gödel, Turing argumentou que embora cada sistema de sua hierarquia fosse um pouco mais completo que o precedente, a hierarquia jamais alcançaria a completude. O processo consiste em gerar um novo sistema um pouco mais completo que seu precedente pela introdução de um enunciado que o fazia incompleto. Isso demandaria alguma forma de decidir a veracidade ou falsidade do enunciado em questão. Turing propôs um oráculo: alguma maneira não-maquínica que seria capaz de fornecer o passo não computável de modo a permitir o prosseguimento da tarefa de completção. Em sua tese de doutorado, Turing propôs: “algum meio não especificado de resolver problemas de teoria dos números; como se fosse uma espécie de oráculo. Não vamos mais longe na natureza deste oráculo além de dizer que ele não pode ser uma máquina” (Turing, 1938, p.18). Hoje, passados quase 80 anos, vemos a Computação Humana materializar a sugestão de Turing: o pequeno passo que ele delegou ao oráculo é o que hoje se delega ao humano. Como no caso da Computação Humana, é possível que muito do que vemos hoje na área de sistemas de informação, coisas que nos parecem essencialmente inovadoras, encontrem bases de fundamentação nos trabalhos matemáticos de um momento em que se dedicou grandes esforços criativos nos estudos relativos aos sistemas formais. Em outras palavras, fundamentos conceituais para o paradigma do século XXI denominado “computação humana” podem ser encontrados nas propostas de Alan Turing, em 1938. Assim, pode ser bastante interessante voltarmos os olhos estes estudos de modo a buscar uma melhor compreensão do nosso tempo presente.

Propomos aqui um enfoque sociotécnico que consiste em deixar aparente o percurso de construção dos conteúdos. Com isso, abrimos mão de uma concepção em que os resultados buscam explicar-se por si só, e, como bem colocou Cukierman et al., possibilitamos a verificação de que o que se diz “técnico” se conforma em co-construção com o que se diz “social”.

3 | MODOS CONTEMPORÂNEOS DE COMPREENSÃO E CONSTRUÇÃO DO CONHECIMENTO

A questão da retenção comentada na seção 1 junta-se com a necessidade de buscar uma dinâmica em sala de aula que melhor se adapte aos modos contemporâneo

de comunicação e entendimento. É possível que o recorte disciplinar em que os conteúdos estão tradicionalmente organizados já não mais dê conta de sensibilizar o estudante. As respostas aos anseios do nosso tempo não são encontradas nas disciplinas isoladas, mas no encontro entre elas, na conjunção de saberes, e por isso, o estudante está constantemente buscando, pesquisando em seus dispositivos móveis, questões relacionadas (ou não) ao que se fala em aula. Muitos dos conteúdos tradicionais já estão disponíveis em video-aulas na Internet, a qualquer tempo, em qualquer lugar e isto põe em cheque o papel do professor como enunciador de conteúdo. Os estudantes reivindicam diálogo, construções colaborativas, descentramento, e é neste espaço que o professor precisa se reposicionar para fazer circular a sua mensagem, não mais como centro catalizador, mas como parte da rede. Além disso, é preciso acompanhar o diálogo, e no fluxo dos interesses de cada turma, introduzir as questões que se quer comunicar. Por tudo isso, a sequência fixada de uma ementa ou programa passa a adquirir um papel orientador, apresentando questões, e não determinando o andamento das aulas.

Presenciamos hoje na sala de aula uma multiplicidade de recursos que disputam a atenção do aluno. Entretanto, o estudante parece mobilizar-se quando tem consciência da importância de um determinado assunto na sua formação profissional. Por isso, os conteúdos programáticos precisam ser justificados. O professor que ensina precisa saber para quem e para o quê ensina, e isto invalida o modelo de “aulões”, “provões” e disciplinas pré-formatadas, que colocam em primeiro plano o conteúdo e não o aluno. Invalida também a maneira como grande parte das disciplinas de conteúdo “exato” são ministradas. Uma ciência é considerada “exata” se seus conteúdos já estão de tal forma estabilizados que não admitem questionamentos. São fórmula, cálculos, que podem ser apresentados de modo a serem diretamente aplicados. Este conceito de “ciências exatas” amplifica o conflito entre um modo estático de transmissão, e a expectativa nos dias de hoje de uma construção em fluxo.

Problematizar a exatidão e rigor, ou seja, deixar aparente o processo de construção (a historicidade) dos conhecimentos quebra este conflito na medida em que coloca à vista os problemas e os embates que motivaram aquelas formulações. Destitui a imponência de um conhecimento que dispensa justificativas, abrindo possibilidades de traduções às situações inesperadas do tempo presente. Desta maneira torna-se possível ao professor situar a importância do assunto em questão à formação pretendida. No caso dos cursos de formação em computação, já foi dito, mas convém repetir que conhecer o processo de construção do computador e da computação ajuda a compreender o mundo em que vivemos hoje, onde a computação vem ocupando um espaço cada vez mais amplo, e assim verificar possibilidades futuras e conceber inovações.

Teoria da Computação, uma abordagem sociotécnica.

Durante este ano de 2016, o Curso de Sistemas de Informação da Universidade Federal Fluminense está passando por uma reforma curricular que visa, entre outros objetivos, minimizar a retenção nos primeiros períodos e adaptar os conteúdos às demandas locais. Dentre diversas mudanças elaboradas pela coordenação, em conjunto com o Núcleo Docente Estruturante, o Colegiado de Curso, e considerando as direções apontadas pelos estudantes, citamos a reorganização dos conteúdos de Matemática Discreta e Lógica Formal, em uma disciplina denominada Fundamentos Matemáticos para Computação, a ser ministrada no primeiro período do ingressante, concomitante com Teoria da Computação para Sistemas de Informação, outra disciplina recém-criada. Com base nas justificativas já comentadas nas seções anteriores, apresentamos o conteúdo programático de Teoria da Computação para Sistemas de Informação. A disciplina organiza-se em torno da linha do tempo, porém seu objetivo não é fazer um relato da sequência temporal purificada dos acontecimentos, mas trabalhar as construções matemáticas ao longo dessa diretriz. Ou seja, pretende-se uma abordagem sociotécnica, em que a matemática explica-se e explica a conjuntura em que foi concebida, e vice-versa. A elaboração desta proposta não teria sido possível sem as discussões realizadas no âmbito do Programa de Pós-Graduação em História das Ciências e das Técnicas e Epistemologia, onde as questões matemáticas são abordadas sob pontos de vistas diversos.

Objetivo: Percorrer a história do desenvolvimento matemático no decorrer do século XX, no que diz respeito às discussões sobre a fundamentação da matemática. Apresentar os conceitos fundamentais da Teoria da Computação imbricados nas situações que motivaram a sua formulação. Abordar não somente as maneiras como as construções teóricas deram suporte ao surgimento da tecnologia (computadores), mas também como as tecnologias deram suporte à novas construções teóricas. A partir da compreensão da construção histórica dos fundamentos da computação buscar a compreensão de situações contemporâneas envolvendo sistemas computacionais.

Embora o foco da disciplina esteja no decorrer do século XX, o curso inicia ambientando a concepção da ciência ao final do século XIX, para evidenciar como as questões emergentes no campo da matemática colocaram em cheque os anseios da ciência moderna que se queria completa e totalizante. O curso avança pelo século XXI para abranger questões da contemporaneidade em uma proposta de conteúdo que não pode ser completamente fixada. Embora os títulos se refiram a eventos históricos, pretende-se trabalhar em detalhe o conteúdo matemático relacionado a cada evento. Ressaltamos entre parênteses o conteúdo matemático a ser abordado, com a ressalva de que há interseções com a disciplina Fundamentos Matemáticos para Computação.

Conteúdo Programático:

Do século XIX ao século XX: As questões que intrigavam os matemáticos ao

final do século XIX e os embates com a Ciência Moderna (arimetização da análise, axiomatização da aritmética, formalização da geometria. Conjuntos indutivos e recursão primitiva a partir dos axiomas de Peano). Georg Cantor e os infinitos (coleções infinitas: enumerabilidade, cardinalidade, provas por diagonalização).

O século XX: Russell e o paradoxo (conjuntos, elementos, pertinência, conjuntos das partes). O nascimento dos programas de fundamentação da matemática: O programa Logicista, O programa Intuicionista (comparações com a lógica clássica), O programa Formalista (revisitar funções primitivas recursivas. Abordar também as parciais recursivas. Completude, Consistência, Decidibilidade de sistemas formais). O Teoremas de Gödel (numeração de Gödel, a prova da incompletude, contrapontos ao programa de Hilbert). Compreensões da matemática sob a percepção da incompletude. Propostas de formalização do conceito de computável. O cálculo lâmbda (Church/Kleene). As funções recursivas gerais (Gödel). A máquina de Turing como um programa (construção da máquina de turing). A máquina de Turing Universal (contraponto ao programa de Hilbert). Tese de Church. Evidências para a Tese de Church (equivalências entre os formalismos abordados). Compreensões filosóficas sob a percepção da incompletude. O Oráculo de Turing e a necessidade de operar com indecidíveis. A dicotomia homem-máquina e o nascimento da IA. O embate entre IA forte e IA fraca e a concepção de híbridos. Computadores, tecnologias e Segunda Guerra. Dilemas de Einstein, Russell, e Wiener. John von Neumann e a bomba atômica. John von Neumann e a implementação da máquina universal. A computação no pós-Guerra. O nascimento das CC, as Conferências da OTAN. O não-computável e os computadores. Número Ômega (a construção do número Ômega). A situação do Brasil no cenário da computação na década de 1970. Representação computacional dos números reais. As controvérsias sobre as abordagens matematizadas e os processos sociais.

O século XXI: A teoria da computação e o suporte às novas tendências da computação: Processos colaborativos, Abordagens sociotécnicas, Computação humana.

A construção desta proposta vem ocorrendo ao longo de três períodos consecutivos em que foi oferecida como disciplina optativa. Utilizou-se como bibliografia básica, os livros textos usuais da área para cobrir o conteúdo formal com uma notação atual. Além disso, artigos da época disponíveis na internet (cartas, discursos, etc) auxiliaram não propriamente no conteúdo formal, mas para apresentar pequenos trechos onde esses autores expõem motivações e intuições das suas propostas. Trechos de filmes e documentários também contribuíram para trazer um pouco do ambiente em que se vivia a cada época. Fotografias, cartas e documentos do arquivo digital de Turing, e por fim, alguns romances (Doxiadis,2001,2010) também contribuíram para ilustrar os dilemas a época.

REFERÊNCIAS

- AHN, L.; BLUM, M.; LANGFORT, J. Telling Human and Computers apart, **Com. of the ACM** Vol. 47, No. 2, 58-60. 2004.
- AHN, L.; MAURER, B.; MCMILLER, C.; ABRAHN, D.; BLUM, M. reCAPTCHA: Human-Based Character Recognition via Web Security Measure. **Science** v321, 2018.
- BOSCARIOLI, C.; ARAUJO, R. M.; MACIEL, R. S. P. I GranDSI-BR – Grand Research Challenges in Information Systems in Brazil 2016-2026. Special Committee on Information Systems (CE-SI). Brazilian Computer Society (SBC). ISBN: [978-85-7669-384-0]. 2017. 184p.
- CAFEZEIRO, I.; HAEUSLER, H.; MARQUES, I.; CUKIERMAN, H. Recontando a Computabilidad”. **Revista Brasileira de História das Ciências**, Rio de Janeiro, v. 3, n. 2, p. 231-251, 2010.
- CUKIERMAN, H.L.; TEIXEIRA, C.; PRIKLADNICKI, R. Um olhar sociotécnico sobre a engenharia de software. **Rev. de Inf. Teórica e Aplicada**, RS, UFRGS, 2007.
- CURSO DE QUALIDADE Evasão e estratégias para permanência estudantil In: XXXV CSBC. Bento Gonçalves, RS, 2015.
- DELEUZE, G. & GUATARRI, F. **Mil Platôs, vol 5**. São Paulo: Editora34,1997.
- Dyson, G. **Turing’s Cathedral: The Origins of the Digital Universe**. Vintage Books, 2010.
- DOXIADIS, A. **Tio Petrus e a conjectura de Goldbach**. São Paulo: Editora 34, 2010.
- DOXIADIS, A., PAPADIMITIRUS, C. **Logicomix, uma jornada épica em busca da verdade**. Martins Fontes,2010.
- FRANZEN, T., **Gödel’s Theorem: An incomplete guide to its use and abuse**. Wellesley, Mass.: A K Peters.
- GÖDEL, K. On formally undecidable propositions of Principia Mathematica and related systems. In: M. Davis (Ed), **The Undecidable: Basic Papers on Undecidable Propositions**, Unsolvability Problems and Computable Functions. NY: Dover Pubs, 1965.
- JUNIOR, W. G. N. O Infinito contado por Deus. Uma interpretação dedekindiana do conceito de número ordinal transfinito de Cantor. Tese Dout. Fil., PUC, RJ,2006.
- LATOUR, B. Scientific Humanities, <http://www.bruno-latour.fr/courses>, 2014.
- LAW, E. AHN, L. **Human Computation**. Morgan & Claypool Publishers, 2011.
- NATO Software Engineering: Report of a conference sponsored by the NATO Science Committee, Scientific Affairs Division,1969,1970.
- SETTI, R. (2013) Após digitalizar livros combatendo spams criador do duolingo quer traduzir web ensinando línguas. **O Globo**, 04/06/2013
- SBC Currículo de Referência da SBC para Cursos de Graduação em Computação. Sociedade Brasileira de Computação. Bento Gonçalves, RS, 1999.
- SOUZA, O. S., Morais, P. S., Junior, F. C. S., Reprovações e Trancamentos nas Disciplinas de Introdução à Programação da Universidade de São Paulo: Um Estudo Preliminar, In: **XXIII WEI – XXXV CSBC**. Bento Gonçalves, RS, 2015.

TURING, A. On computable numbers, with an application to the Entscheidungsproblem, **Proceedings of the London Mathematical Society**, Series 2, n.42, p 230-265, 1936.

TURING, A. Systems of Logic Based on Ordinals. Doctoral thesis, Princeton University, 1938.

MODELO DE FÁBRICA DE SOFTWARE ESCOLA

Edmilson Barbalho Campos Neto

Instituto Federal de Educação, Ciência e
Tecnologia do Rio Grande do Norte

Natal – RN

Alba Sandrya Bezerra Lopes

Instituto Federal de Educação, Ciência e
Tecnologia do Rio Grande do Norte

Natal – RN

Diego Silveira Costa Nascimento

Instituto Federal de Educação, Ciência e
Tecnologia do Rio Grande do Norte

Natal – RN

RESUMO: O presente artigo traz um relato de experiência da implantação de uma fábrica de software escola para os alunos dos cursos técnicos em Informática e Informática para Internet e curso superior em Licenciatura em Informática do Instituto Federal do Rio Grande do Norte, Campus Natal – Zona Norte. O objetivo foi criar um ambiente para que os alunos pudessem vivenciar de forma tangível as fases da Engenharia de Software e ao mesmo tempo atender a demandas reais. A metodologia foi aplicada tanto dentro de sala de aula como em projetos de pesquisa. Como resultado, 74,3% dos alunos indicaram ter internalizado melhor os conteúdos através da experiência de vivência na fábrica, ao comparar com a metodologia tradicional de aulas expositivas.

PALAVRAS-CHAVE: engenharia de software, fábrica, escola.

ABSTRACT: This paper presents an experience report of implantation of a software factory for education for students of technical and graduation course in computer at Instituto Federal do Rio Grande do Norte, – Campus Natal – Zona Norte. The goal was to create an environment where students could experience the stages of Software Engineering and still solve real demands. The methodology was applied inside classrooms and also in research projects. As results, 74.3% of students indicate that the learning process increased using the practical methods when compared with a theoretical approach.

KEYWORDS: software engineer, factory, school.

1 | INTRODUÇÃO

O conceito Fábrica de Software (FS) foi usada pela primeira vez na década de 60, mas popularizada apenas nos anos 90. Segundo Herbsleb e Grinter (1999), pode ser definido como “um processo estruturado, controlado e melhorado de forma contínua, considerando abordagens de engenharia industrial, orientado para o atendimento a múltiplas demandas de

natureza e escopo distintas, visando à geração de produtos de software, conforme os requerimentos documentados dos usuário e/ou clientes, da forma mais produtiva e economicamente possível”.

No âmbito acadêmico, a FS é uma proposta inovadora que busca a interação concomitante entre teoria e prática, subsidiando os alunos na aplicação real dos conceitos aprendidos em sala de aula, ao mesmo tempo que prepara mão de obra especializada para atuar no mercado de desenvolvimento de software. O modelo vem sendo seguido em instituições de educação do Brasil e no Mundo. Por exemplo, no trabalho de Brito et al. (2013), os autores apresentaram uma proposta de criação de uma metodologia para FS no Instituto Federal de Educação, Ciência e Tecnologia de Goiás – campus Inhumas de forma a atender demandas reprimidas de desenvolvimento de software, bem como, proporcionar aos alunos experiências práticas. Já no trabalho de Oliveira e Colenci (2003), os autores descrevem um relato de experiência da implantação de uma FS acadêmica implantada na Faculdade de Tecnologia de Jundiaí em São Paulo que visa estimular a capacitação tecnológica dos alunos e o incentivo a criação de empresas competitivas. No trabalho de Siqueira et al. (2008), os autores apresentaram uma abordagem metodológica para o ensino de Engenharia de Software (ES) na Escola Politécnica da Universidade de São Paulo, de forma que o ensino não ficasse restrito à sala de aula e o laboratório, mas que os alunos pudessem vivenciar um processo de desenvolvimento de software. Enquanto que no trabalho de Borges et al. (2012), os autores relatam as experiências alcançadas no projeto de extensão de uma FS desenvolvido no Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, que teve por objetivo propiciar aos alunos capacitação nas principais tecnologias de mercado e vivência no mundo do trabalho. E o trabalho de Ahmad et al. (2014), que os autores fizeram uma avaliação da percepção dos alunos através da utilização de quadro Kanban e aprendizado colaborativo que foi desenvolvido na Universidade de Oulu na Finlândia.

De acordo com as experiências anteriormente citadas no que se refere à aplicação do conceito de FS em âmbito formador, o presente artigo parte da hipótese de que tal abordagem é promissora na formação dos alunos na disciplina de ES, permitindo, principalmente, um amadurecimento técnico e profissional.

De forma a comprovar nossa hipótese, foi desenvolvido o projeto de extensão FaSEs, um acrônimo para Fábrica de Software Escola. O projeto tem como objetivo geral possibilitar aos estudantes dos cursos da área de tecnologia e afins do Instituto Federal do Rio Grande do Norte, Campus Natal – Zona Norte, aplicar os conhecimentos de ES trabalhados em sala de aula para criação de soluções reais demandadas pela própria comunidade escolar ou industrial. Com isso, almeja-se possibilitar aos alunos uma vivência de um ambiente mais próximo da realidade de uma organização de desenvolvimento de software, que permite (Herbsleb e Grinter 1999): capacitar os alunos para a compreensão e resolução de problemas relacionados à produção de software, seguindo processos de desenvolvimento de sistemas; pesquisar e aplicar

novas tecnologias e metodologias de desenvolvimento de software; divulgar o potencial dos alunos participantes junto ao mercado de trabalho, com vista a obtenção de colocações dentro das empresas de desenvolvimento de software; dar suporte a projetos da instituição, fornecendo soluções de software personalizadas; e promover outras ações (palestras, cursos de formação complementar, certificação profissional, eventos, entre outros).

Esse artigo está organizado da seguinte forma. A Seção 2 apresenta as metodologias de ensino e desenvolvimento utilizadas pela FaSEs. A Seção 3 apresenta e discute os resultados alcançados, A Seção 4 apresenta uma avaliação da metodologia adotada. Por fim, o artigo é concluído na Seção 5, onde são realizadas as considerações finais.

2 | METODOLOGIA DE ENSINO

Segundo Sommerville (2011), ES é uma disciplina cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até sua manutenção. Embora não exista um consenso entre autores quanto a terminologia adotada para as fases de desenvolvimento de um projeto de software, neste trabalho, adotamos as etapas categorizadas como: levantamento de requisitos, análise, projeto, implementação e implantação.

2.1. Seleção e Acompanhamento de Projetos

A metodologia adotada para o desenvolvimento deste projeto buscou envolver os alunos nos diversos anos de ensino dos cursos técnicos e em Informática e Informática para Internet e do curso superior em Licenciatura em Informática da instituição. Para tanto, a FaSEs foi organizada em dois grupos, aqui chamados de divisões da fábrica. A primeira, a Divisão Acadêmica (DA), abrange os projetos executados diretamente em sala de aula, pelos problemas e alunos envolvidos nas disciplinas integradas, sendo esses, portanto, projetos curriculares. O segundo grupo, intitulado Divisão de Inovação (DI) compreende os projetos executados por editais de pesquisa de fomento interno ou externo da instituição, seja esses por meio de bolsas de pesquisa ou não.

A Figura 1 ilustra uma visão geral da estrutura da FaSEs com alguns números correspondentes ao cenário atual de sua implantação na instituição. Além disso, é possível observar no lado esquerdo da figura, os grupos responsáveis por demandar soluções a serem atendidas por cada uma das divisões da fábrica, a depender da natureza da execução do projeto (acadêmica ou pesquisa). Independente da divisão responsável, todos os projetos da FaSEs atendem necessariamente às soluções que surgem desses grupos de demandantes (stakeholders), a seguir descritos:

- Comunidade Acadêmica (interno): Grupo formado pelos alunos, professores

e servidores da própria instituição; as soluções demandadas por esse grupo buscam otimizar/corriger problemas diretamente ligados ao dia a dia da instituição; e

- Indústria e Comércio (externo): Grupo formado por stakeholders diretamente ligados a indústria ou comerciantes locais, que demandam soluções específicas para o seu negócio.



Figura 1: Visão geral dos resultados.

Pela necessidade de gerenciar todos os projetos de forma unificada, adotou-se a ferramenta Trello (<http://www.trello.com>) que possibilita criar quadros de tarefas específicos (Kanban Board) para cada projeto e associar alunos e professores participantes. Além dessa ferramenta, foi adotado um repositório de código visando melhor versionamento dos sistemas e possibilitando que os projetos possam ser continuados por outras equipes no futuro. No primeiro ano de implantação, os projetos foram disponibilizados no GitHub (<http://www.github.com/fases>). No segundo ano, o repositório adotado foi o GitLab (<http://www.gitlab.com>).

2.2. Processo de Desenvolvimento de Software

O processo seguido pela FaSEs baseia-se no OpenUP (<http://epf.eclipse.org/wikis/openup/>), um modelo iterativo-incremental dividido em quatro fases: Concepção, Elaboração, Construção e Implantação. Cada fase foi subdividida em iterações, com duração média de 15 dias, e ao final de cada iteração são agendadas entregas para cada equipe apresentar os avanços obtidos. As entregas funcionavam como entregas simuladas de releases parciais da solução aos clientes/demandantes, que em alguns projetos participaram ativamente do processo, com avaliações das entregas. A escolha do processo OpenUP se deu pelo fato dessa metodologia ser capaz de fomentar a prática com profundidade de todas as atividades da ES, desde a análise à implantação, no ambiente acadêmico.

A Tabela 1 apresenta as principais atividades e tarefas gerados pela primeira fase

do processo, a Concepção. Extraída do OpenUP, o objetivo dessa fase é identificar e abordar os stakeholders do sistema para extrair/entender a relação de requisitos funcionais e não-funcionais que o projeto deverá atender, gerando, para tanto, alguns artefatos específicos para este fim.

Atividade	Tarefa	Artefato Gerado
Iniciar projeto	Iniciar projeto	- Documento de Visão - Glossário
Planejar e Gerenciar iteração	Planejar iteração	- Plano de Riscos - Plano de Iteração
Identificar requisitos	Identificar requisitos Detalhar cenários de UDC Realizar protótipos de telas	- Tabela de RNF - Casos de Uso (CDU) - Protótipos baixa fidelidade
Esboçar arquitetura	Definir arquitetura do sistema	- Documento de arquitetura

Tabela 1. Artefatos gerados na fase de Concepção do processo

A segunda fase do processo OpenUP, adaptada para o processo da FaSEs, tem o nome de Elaboração e seu objetivo é definir como as funcionalidades do sistema, identificadas na fase anterior, serão implementadas na fase de Construção seguinte. Para isso, é necessário identificar quais casos de uso apresentam maior risco para que possam ser implementados nessa fase, com o objetivo de validar a arquitetura projetada para o sistema. Considera-se casos de uso de maior risco (CDUMR) aqueles que, se não implementados/entregues aos clientes, desconfiguram a essência do sistema. A Tabela 2 reporta os artefatos gerados a fase de Elaboração.

As duas últimas fases do processo, a Construção e a Transição, têm como objetivo, respectivamente, implementar todos os casos de uso restantes e previstos para o sistema; e validar e implantar o sistema em ambiente real. Na Construção, os casos de uso especificados nas fases anteriores são em plenitude implementados seguindo a arquitetura validada na fase de Elaboração. Esses casos de uso são validados por meio de testes durante o desenvolvimento e, ao final, transferidos para um ambiente real de produção, para utilização e aprovação por parte dos stakeholders do sistema.

Atividade	Tarefa	Artefato Gerado
Gerenciar requisitos	Encontrar e descrever requisitos Detalhamento de requisitos Criar caso de teste	- Especificação de CDUMR - Plano de teste

Definir arquitetura	Desenvolver arquitetura Projetar banco de dados	- Diagrama de classes - Diagrama de pacotes - MER
Desenvolver um incremento da solução	Projetar a solução Implementar a solução Executar testes	- Protótipos alta fidelidade - Release com incremento - Relatórios de testes
Planejar e Gerenciar Iteração	Planejar interação	- Plano da Iteração

Tabela 2. Artefatos gerados na fase de Elaboração do processo

3 | RESULTADOS

Esta seção apresenta os resultados obtidos pela FaSEs após dois anos da sua implantação no campus da instituição de ensino, de acordo com a natureza da divisão da fábrica (acadêmica ou inovação).

3.1. Experiência de Interdisciplinaridade

Os projetos da Divisão Acadêmica (DA) foram executados englobando duas disciplinas da estrutura curricular do quarto ano do curso técnico integrado em Informática: Programação para Internet (PPI) e Projeto de Desenvolvimento de Software (PDS). O projeto pedagógico do curso considera que as duas disciplinas possam desenvolver projeto integrador, de forma a possibilitar melhor consolidação dos conteúdos e objetivos das disciplinas. A ementa da disciplina de PDS busca proporcionar ao aluno a compreensão do processo de desenvolvimento de software, as atividades técnicas e as iterações envolvidas neste processo. Já a disciplina de PPI propõe que os alunos possam desenvolver aplicações com programação do lado do cliente e do servidor e controlar o estado e o acesso em aplicações na internet.

Em 2015, buscando atingir esses objetivos, os professores dividiram a turma composta por 27 alunos em seis equipes e discutiram as demandas de sistemas existentes na fábrica. Das seis demandas apresentadas na turma, três eram da comunidade acadêmica e três da indústria e comércio. Cada equipe ficou responsável por uma demanda específica. Além disso, no que concerne a execução das atividades e da interdisciplinaridade, a disciplina de PDS foi a responsável por conduzir o processo de desenvolvimento, agregando as fases do processo e as atividades específicas de implementação trabalhadas com alunos pelo professor da disciplina de PPI. Em 2016, igual divisão foi realizada com uma nova turma formada por 31 alunos concluintes.

3.2. Experiência Entre Cursos de Níveis Distintos

Além da integração das disciplinas intracurso ocorrida no nível técnico em

2015, no segundo ano da vivência da FaSES da Divisão Acadêmica, em 2016, foi viabilizada a integração de disciplina de cursos diferentes e de níveis distintos. A experiência contou com a integração da disciplina de Engenharia de Software (ES) ofertada a uma turma de curso superior em licenciatura plena em informática com as duas disciplinas de PPI e PDS do curso técnico já integradas no ano anterior. Nessa nova formatação, os alunos de nível superior em licenciatura atuavam como mentores dos projetos, acompanhando a vivência do projeto e dando suporte pedagógico necessário ao entendimento dos conceitos envolvidos na execução de um processo de desenvolvimento de software, enquanto os alunos do curso técnico atuavam na implementação e geração dos artefatos previstos no processo.

3.3. Projeto da Divisão Acadêmica (DA)

A Tabela 3 apresenta a distribuição dos projetos. Para cada projeto é indicado o ano de início, o tipo do projeto, o objeto/área de concentração, a origem da demanda (comunidade acadêmica - CA ou indústria e comércio - IC) e a quantidade de alunos envolvidos em cada equipe. Além disso, a tabela também apresenta a quantidade de trabalhos que foram utilizados como trabalho de conclusão de curso pelos alunos do quarto ano, concluintes do curso técnico em Informática. A seguir, são apresentados dois projetos de naturezas distintas para exemplificar a diferença entre os tipos de demandantes da FaSEs na Divisão Acadêmica. Mais informações sobre os projetos desenvolvidos podem ser obtidas através do site do projeto (<http://fases.ifrn.edu.br/>).

Projeto	Ano	Tipo	Objeto	Demandante	#alunos	#TCC
Brainzer	2015	Jogo educativo	Saúde	IC	5	3
Comunica	2015	SI	Comunicação	CA	5	1
Emprenet	2015	Serviço online	Domésticas	IC	3	-
Filmões	2015	Serviço online	Filmes	IC	4	-
SUSI	2015	SI	Saúde	CA	5	-
WorldShare	2015	Rede Social	Acadêmico	CA	5	-
ACHA	2016	SI	Carga horária	IC	5	3
COLINFO	2016	Serviço online	Publicações	CA	5	3
BinChallenger	2016	Jogo Educativo	Binários	IC	5	3
FaSEs Web	2016	SI	FaSEs	IC	5	1
MudiFast	2016	SI	Resoluções	CA	1	3
SIGMIN	2016	SI	Minicursos	CA	6	6

Tabela 3. Sumário dos resultados da Divisão Acadêmica da FaSEs

(1) TCC - Trabalho de Conclusão de Curso; (2) IC - Indústria e Comércio; (3) SI - Sistema de Informação; (4) CA - Comunidade Acadêmica

O projeto Brainzer (IC) teve como objetivo o desenvolvimento de um portal de jogos online para auxiliar portadores da doença Alzheimer no estímulo de sua cognição. No processo de desenvolvimento, a equipe realizou entrevistas com médico especialista,

membros de associações de auxílio a familiares e portadores das doenças. Os jogos basearam-se em atividades endossadas por especialistas nesse tipo de doença neurodegenerativa, como jogos da memória, caça-palavras e diversos questionários intuitivos.

O projeto Worldshare (CA), por sua vez, foi desenvolvido para atender uma demanda institucional externada pelos docentes. A demanda foi por uma ferramenta onde os alunos pudessem compartilhar suas produções acadêmicas e culturais, tais como poesias, crônicas, além de resumos e artigos técnicos de disciplinas curriculares. Antes da publicação do conteúdo, a produção passa pela revisão de um professor da área, que aponta trechos carentes de melhorias que são corrigidos pelo aluno.

3.3.1. Utilização dos Projetos como Prática Profissional

Além da utilização dos projetos para a consolidação dos conteúdos das disciplinas de PDS e PPI, os alunos do curso técnico participantes dos projetos da Divisão Acadêmica ainda puderam aproveitar a vivência extracurricular possibilitada pelos projetos desenvolvidos para integralização da carga horária de prática profissional prevista no projeto pedagógico do curso técnico na instituição, pré-requisito para conclusão de curso, contribuindo assim para melhoria dos índices de êxito do curso.

Para avaliar esse impacto fomentado pela Divisão Acadêmica nos índices de conclusão, foram comparados os índices acadêmicos do ano letivo de 2015 e 2016 (após dois anos de implantação da FaSEs) com os de 2014 (ano anterior). A análise focou na avaliação da Taxa de conclusão (TC) de curso em cada ano letivo, que considera apenas os alunos que integralizaram todos os componentes curriculares, incluindo a prática profissional, no ano letivo analisado. A Tabela 4 apresenta um sumário dessa análise comparativa.

Ano letivo	#turmas	#alunos	#TC	#TC (FaSEs)
2014	2	49	4% (2)	-
2015	1	27	79% (23)	24% (7)
2016	1	21	100% (31)	61% (19)

Tabela 4. Resultado da análise comparativa

(1) Taxa de conclusão

Após essa análise, foi possível observar um avanço significativo na Taxa de Conclusão entre os anos de 2014 e 2015, que aumentou de 4% (apenas dois alunos em 2014) para 79% em 2015 (23 alunos) e, em 2016, atingiu 100% de conclusão. Dos 23 alunos que concluíram suas práticas profissionais em 2015, sete utilizaram projetos desenvolvidos pela FaSEs, o que implica uma Taxa de Conclusão relacionada a FaSEs de 24%. Esse número teve um crescimento ainda mais expressivo no segundo ano do projeto, atingindo 61%, o que equivale a 19 alunos.

3.4. Projetos da Divisão de Inovação (DI)

Os projetos desenvolvidos nessa divisão atuam numa outra frente de resultados, responsável por impulsionar a produção científica da FaSEs. Os projetos, diferente dos da divisão acadêmica, que são desenvolvidos em sala de aula, são fomentados por editais de pesquisa internos ou externos e todos com potencial de inovação. Nesse sentido, os resultados desses projetos podem ser mensurados nos números de publicações científicas e acadêmicas culminadas por cada um deles. A Tabela 5 a seguir apresenta um sumário desses resultados por projeto nesse biênio de implantação da metodologia proposta pela FaSEs, além de informações complementares sobre a origem de algumas das demandas de cada projeto e quantidade de alunos envolvidos por equipe.

O projeto Epuppy foi o projeto pioneiro da FaSEs e surgiu de uma demanda proveniente de uma clínica veterinária local, que externou, a partir da vivência diária, a necessidade de reunir em uma única rede as informações acerca dos animais domésticos e todos os envolvidos no cuidado desses animais, possibilitando a identificação do animal através de um QR-Code, que pode ser acoplado à coleira do animal.

Projeto	Demandante	#alunos	#feiras	#publicações	#premiações	#TCC
EPuppy	IC	5	1	1	0	3
Luzômetro	CA	3	0	1	0	0
SmartGás	IC	2	3	2	2	0
TED Saúde	IC	2	5	1	4	2
FreeAccess	IC	3	0	1	1	0
Pocket-Chef	IC	2	0	1	0	0

Tabela 5. Sumário dos resultados da Divisão de Inovação

(1) TCC - Trabalho de Conclusão de Curso; (2) IC - Indústria e Comércio; (3) CA - Comunidade Acadêmica

O projeto SmartGás, vide Figura 2, surgiu de um potencial de inovação da indústria e comércio de gás de cozinha doméstico. O projeto propõe o desenvolvimento e plataforma inteligente para o controle do consumo do gás. A plataforma fornece informações em tempo real da quantidade de gás restante no botijão, além de dados essenciais para controle da economia doméstica (Medeiros et al. 2016). O projeto, com publicações nacionais e internacionais, é financiado por edital interno e conta com dois alunos do terceiro ano do curso técnico em informática para internet.

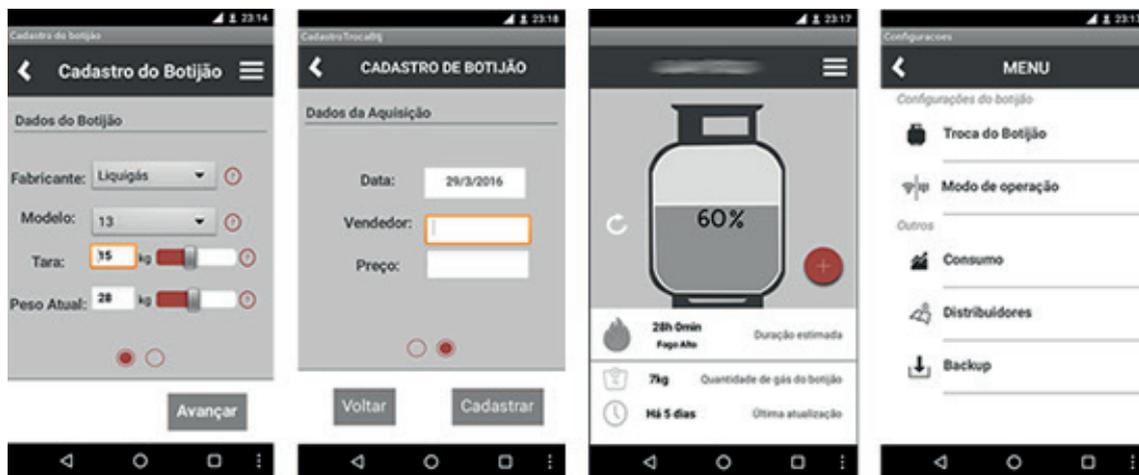


Figura 2. Protótipo do aplicativo desenvolvido no projeto SmartGás.

Outro projeto em destaque da FaSEs, o TEDSaúde, propõe o desenvolvimento de uma plataforma educativa digital, e também visa atender um potencial de inovação do setor de saúde (Silva et al. 2016). No projeto é desenvolvida uma plataforma interativa utilizando o dispositivo Kinect, a ser implantado nas salas de espera das unidades básicas de saúde, para a promoção da saúde. Os resultados do projeto foram disseminados por meio da publicação em eventos científicos e feiras tecnológicas, algumas das quais renderam premiações e credenciamentos para participação em outras feiras nacionais e internacionais.

4 | AVALIAÇÃO DOS RESULTADOS

Com o objetivo de avaliar a metodologia-modelo implantada pela FaSEs na Divisão Acadêmica, foi aplicada um questionário ao final da execução do segundo ano do projeto, que tinha como eixos principais: (i) Avaliar a compreensão/internalização dos conteúdos de engenharia de software; e (ii) Avaliar a perceptividade da metodologia adotada. A aplicação contou com 35 respondentes anônimos, entre alunos do curso técnico e do curso superior em licenciatura em informática, que participaram de projetos da Divisão Acadêmica durante o ano letivo de 2016.

Na primeira parte do questionário (primeiro eixo), os alunos foram perguntados sobre: (i) a nomenclatura e ordem das fases do processo adotado; e (ii) os objetivos de cada uma das fases. O objetivo era identificar se a adoção de uma metodologia prática para ensino de engenharia de software foi eficiente para absorção dos conhecimentos teóricos da disciplina. O índice de acerto dos respondentes sobre a primeira fase (Concepção) foi de 100% e de 94,3% para a segunda fase (Elaboração). Já em relação das duas últimas fases, os índices de acertos foram um pouco menores. Sobre a fase de Construção, 74,3% acertaram o seu objetivo, enquanto 77,1% responderam corretamente sobre o objetivo da fase de Transição.

Na segunda parte do questionário (segundo eixo), os alunos foram questionados

sobre a experiência da vivência de um processo de engenharia de software na prática. 74,3% (ou seja, 26 alunos), disseram acreditar que a vivência de um processo de engenharia de software na prática possibilita alta contribuição dos conceitos teóricos da disciplina, enquanto 14,3% (5), responderam moderada contribuição. Apenas quatro respondentes disseram acreditar que aprenderiam mais com aulas expositivas do que uma vivência prática. A resposta dessa questão pode ser vista na Figura 3 (a).



Figura 3. Nível de satisfação dos alunos com o estágio final dos seus projetos.

Também foi questionado a opinião dos alunos sobre a relação entre o esforço despendido e o aprendizado obtido na adoção de uma metodologia prática em relação a metodologias tradicionais de aulas expositivas-teóricas vide Figura 3 (b). Para 80% (28) dos respondentes, o esforço de uma metodologia prática é maior, mas em compensação o aprendizado obtido também é maior, enquanto apenas 17,1% (6) acreditam ser maior o esforço e menor o aprendizado. Um único respondente afirmou ainda acreditar que uma metodologia prática, além de possibilitar maior aprendizado, ainda requer um menor esforço que metodologias expositivas.

Sobre a satisfação geral com a metodologia prática adotada na disciplina de PDS e PPI, 68,5% (24) responderam excelente ou boa, enquanto 25,7% (9) indicaram como regular, apontando como necessidade de melhoria maior tempo de acompanhamento dos projetos em sala de aula, uma limitação da carga horária das disciplinas envolvidas. Por fim, foi solicitado que os alunos indicassem o nível de satisfação com o estágio final dos seus respectivos projetos, numa escala de 1 a 5, e mais de 80% responderam entre 4 e 5.

Além desses resultados, é possível avaliar o impacto da Divisão de Inovação pelo número de publicações (5), premiações (4) e projetos (4) já apresentados na Seção 4.2. Nos próximos anos será possível comparar também esses números para essa divisão de modo a observar a evolução desses indicadores em face aos demais projetos da instituição.

5 | CONSIDERAÇÕES FINAIS

Este trabalho apresentou um relato de experiência da implantação de uma Fábrica de Software Escola, denominada FaSEs, em uma instituição de ensino. A

execução do projeto culminou no desenvolvimento bem-sucedido de 18 projetos de software no ambiente acadêmico. Nesse período, os alunos puderam vivenciar as etapas de concepção, elaboração, construção e implantação inerentes a um processo de desenvolvimento de software. Como consequência, a implantação da fábrica contribuiu para a maturidade dos alunos no processo de desenvolvimento de sistemas. Ao serem questionados, 80% alunos consideraram que, apesar do esforço despendido na condução da disciplina de maneira prática ser maior, o aprendizado é também maior. E 80% dos alunos consideraram como satisfatório o estágio final dos seus projetos. Como trabalhos futuros, pretende-se incluir as etapas de testes e manutenção na metodologia de ensino para que alunos possam vivenciar também o processo de continuidade de um sistema já em desenvolvimento.

REFERÊNCIAS

- Ahmad, M. O., Liukkunen, K., and Markkula, J. (2014). **Student perceptions and attitudes towards the software factory as a learning environment**. In IEEE Global Engineering Education Conference, pages 422–428.
- Borges, K. S., de Carvalho, T. P., de Moraes, M. A. C. (2012). **Programa de extensão “fábrica de software acadêmica”: contribuindo para a formação profissional na área da informática**. Workshop sobre Educação em Computação.
- Brito, M. C. A., Silva, F. P., Cabral, E. P. (2013). **Elaboração de uma metodologia de desenvolvimento de software para a fábrica de software de uma instituição de ensino**. Revista Brasileira de Informática na Educação, 21(2):52–61.
- Herbsleb, J. D. and Grinter, R. E. (1999). **Splitting the organization and integrating the code: Conway’s law revisited**. In International Conference on Software Engineering, pages 85–95.
- Medeiros, G. V. S., Santos, M. R., Lopes, A. S. B., Campos Neto, E. B. (2016). **Smartgás: uma plataforma inteligente para monitoramento do consumo de gás de cozinha**. In VIII Computer on the Beach, Florianópolis, SC.
- Oliveira, D. H. e Colenci Neto, A. (2003). **Fábrica de software: Promovendo a criação de empresas competitivas em tecnologia da informação**. XXIII Encontro Nacional de Engenharia de Produção.
- Silva, F. M., Lopes, A. S. B., Campos Neto, E. B. (2016). **Uma aplicação de terminais interativos de interface natural em ambientes inteligentes**. In XV Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais (IHC), São Paulo, SP.
- Siqueira, F. L., Barbarán, G. M. C., Becerra, J. L. R. (2008). **A software factory for education in software engineering**. In IEEE 21st Conference on Software Engineering Education and Training, 2008, pages 215–222.
- Sommerville, I. (2011). **Engenharia de Software**. Pearson, 9 edition.

INTERDISCIPLINARIDADE NO IF FARROUPILHA - CAMPUS SANTO ÂNGELO ATRAVÉS DA PRÁTICA PROFISSIONAL INTEGRADA

Fábio Weber Albiero

Instituto Federal Farroupilha - *Campus* Santo Ângelo
Santo Ângelo - RS

Karlise Soares Nascimento

Instituto Federal Farroupilha - *Campus* Santo Ângelo
Santo Ângelo - RS

Andréa Pereira

Instituto Federal Farroupilha - *Campus* Santo Ângelo
Santo Ângelo - RS

Joice Machado

Instituto Federal Farroupilha - *Campus* Santo Ângelo
Santo Ângelo - RS

RESUMO: Este relato apresenta a Prática Profissional Integrada (PPI) desenvolvida no Instituto Federal Farroupilha - *Campus* Santo Ângelo, no ano de 2015, no eixo Informação e Comunicação, o qual abrange os seguintes cursos: Curso Técnico em Manutenção e Suporte em Informática - Integrado ao Ensino Médio, Curso Técnico em Informática para Internet - Subsequente (Pós Ensino Médio) e Curso Superior de Tecnologia em Sistemas para Internet. A PPI é uma série de atividades que visam a integração de diferentes disciplinas.

Como exemplo, no Curso Técnico em Manutenção e Suporte em Informática, há a integração de disciplinas da área básica (Arte, Biologia, etc.) com disciplinas da área técnica (Organização e Arquitetura de Computadores, etc.).

PALAVRAS-CHAVE: Interdisciplinaridade, Prática Profissional Integrada e Eixo de Informação e Comunicação.

ABSTRACT: This paper includes reports on Professional Practice Integrated (PPI) developed at the Federal Institute Farroupilha - *Campus* Santo Ângelo in the year 2015, on Information and Communication shaft, which covers the following courses: Technical Course on Maintenance and Support in Computer - Integrated to High School, Technical Course on Internet Information - Subsequent (post high school) and Technology on Internet Systems course. The PPI is a series of activities aimed the integration of different disciplines. As an example on the Technical Course on Maintenance and Support in Computer, there is the disciplines integration of basic area (Art, Biology, etc.) and disciplines of technical area (Computer Organization and Architecture, etc.).

KEYWORDS: Interdisciplinarity, Professional Practice Integrated and Information and Communication Shaft.

1 | INTRODUÇÃO

O Instituto Federal Farroupilha, em 2014, fundou na cidade de Santo Ângelo, um de seus onze campi no estado do Rio Grande do Sul. O *Campus* Santo Ângelo atua nos eixos tecnológicos de Informação e Comunicação, Ambiente e Saúde e Recursos Naturais, nos diferentes níveis de educação: Técnico Integrado ao Ensino Médio, Subsequente (Pós Ensino Médio), Superior, entre outras modalidades. Desta forma, faz-se necessário relatar sobre a interdisciplinaridade nos diferentes níveis, uma vez que os mesmos professores atuantes no curso superior também podem atuar nas demais modalidades.

O Corpo Docente, o Setor de Apoio Pedagógico e a Direção de Ensino, planejam conjuntamente, a interdisciplinaridade prevista na legislação, através de reuniões que discutem como será abordada e quais as disciplinas serão articuladas no ano ou semestre letivo e documentam as decisões no projeto da Prática Profissional Integrada (PPI). É por meio desta prática que o IF Farroupilha desenvolve a interdisciplinaridade entre os componentes curriculares de cada curso.

Este artigo apresenta, na seção 2, relatos das Práticas Profissionais Integradas desenvolvidas no ano de 2015, que ocorreram no Curso Técnico em Manutenção e Suporte em Informática - Integrado ao Ensino Médio, no Curso Técnico em Informática para Internet - Subsequente (Pós Ensino Médio) e no Curso Superior de Tecnologia em Sistemas para Internet. A seção 3 apresenta as considerações finais.

2 | INTERDISCIPLINARIDADE NO IF FARROUPILHA - CAMPUS SANTO ÂNGELO

A interdisciplinaridade tem como objetivo principal desenvolver um trabalho de integração dos conteúdos de uma disciplina com outras áreas de conhecimento, o que contribui para o aprendizado do aluno. Esta interação entre disciplinas aparentemente distintas, é uma maneira complementar que possibilita a formação de um saber crítico-reflexivo. É através dessa perspectiva que a interdisciplinaridade surge como uma alternativa para superar a fragmentação entre as disciplinas, proporcionando um diálogo entre estas e relacionando-as entre si para a compreensão da realidade (CARLOS, 2007).

No Instituto Federal Farroupilha, a interdisciplinaridade se faz presente por meio da Prática Profissional Integrada (PPI), que constitui parte do Projeto Pedagógico de Curso (PPC) dos três cursos já citados, do eixo Informação e Comunicação.

3 | PRÁTICA PROFISSIONAL INTEGRADA

De acordo com a instrução normativa nº 02, de 18 de fevereiro de 2013, do IF Farroupilha, entende-se por Prática Profissional Integrada a articulação da teoria no

processo de ensino e aprendizagem (INSTITUTO FEDERAL FARROUPILHA, 2013), visando a interdisciplinaridade assegurada no currículo e na prática pedagógica; e, superando a fragmentação de conhecimentos, imposta na organização curricular. A PPI tem por finalidade ser uma estratégia educacional favorável para a contextualização, flexibilização e integração entre o conhecimento teórico e profissional em prática.

3.1 PPI No Curso Técnico em Manutenção e Suporte em Informática - Integrado ao Ensino Médio

No Curso Técnico em Manutenção e Suporte em Informática - Integrado ao Ensino Médio, a Prática Profissional Integrada, teve como objetivo principal receber doações de equipamentos eletrônicos, reciclá-los ou descartá-los, de forma correta. Os objetivos específicos constituíram em: promover o reuso de equipamentos eletrônicos, aumentando seu tempo de vida e reduzindo a quantidade de lixo eletrônico; oportunizar aos alunos uma reflexão sobre o lixo eletrônico; incentivar os alunos na questão da cidadania solidária; promover a inclusão digital através do reuso desses equipamento; e, por fim, evitar a contaminação ambiental e prejuízos a saúde pública devido à disposição incorreta do lixo eletrônico.

Esta PPI integrou disciplinas da área básica, tais como: Arte, Biologia, Filosofia, Física, Geografia, Língua Portuguesa, Matemática, Química e Sociologia; com as disciplinas da área técnica: Introdução à Informática e Organização e Arquitetura de Computadores, por meio de uma série de atividades propostas.

A metodologia empregada consistiu, primeiramente, no recebimento de doações de componentes eletrônicos através dos alunos e dos servidores do *campus*. Após a etapa de coleta desses componentes, realizaram-se as atividades propostas em cada uma das disciplinas que integraram a PPI. Por exemplo, na disciplina de Química, a atividade proposta consistiu no estudo sobre os elementos químicos que compõe cada componente físico do dispositivo eletrônico. Por sua vez, na disciplina de Organização e Arquitetura de Computadores, a atividade proposta consistiu na realização de testes dos componentes físicos, verificando quais são as funções de cada componente e quais poderiam ser reaproveitados para a construção de novos dispositivos eletrônicos. Os componentes físicos que apresentaram defeitos ou falhas foram utilizados como matéria prima na disciplina de Artes para a elaboração de obras de arte.

3.2 PPI No Curso Técnico em Informática para Internet - Subsequente (Pós Ensino Médio)

No Curso Técnico em Informática para Internet - Subsequente (Pós Ensino Médio) foram realizadas duas Práticas Profissionais Integradas, uma em cada semestre letivo do ano de 2015. A primeira PPI teve como objetivo principal a elaboração de uma página web, tendo como conteúdo um glossário de palavras técnicas de origem Inglesa, que são utilizadas na área de Informática. Desta forma, esta PPI buscou integrar as

disciplinas de Aplicativos Web, Inglês I, Fundamentos da Computação e Metodologia Científica.

A metodologia utilizada para a primeira PPI consistiu na divisão da turma em pequenos grupos, os quais foram orientados a selecionar conceitos técnicos abordados na disciplina de Fundamentos da Computação. Após esta etapa, cada grupo de alunos produziu um glossário dos termos em Inglês com seus respectivos significados em Português. Posteriormente, cada um dos grupos construiu uma página web (utilizando conceitos aprendidos na disciplina de Aplicativos Web), aonde apresentaram os seus glossários. A conclusão desta atividade, ocorreu em forma de seminário, com a apresentação da página web criada e do respectivo glossário para os colegas e docentes do curso, além de um relatório integrando assim a disciplina de Metodologia Científica. Esta PPI proporcionou aos alunos do curso um aprofundamento nos conceitos técnicos trabalhados nas disciplinas de Fundamentos da Computação e Aplicativos Web, assim como um aprimoramento da capacidade de compreensão e escrita do vocabulário da Língua Inglesa e Portuguesa.

A segunda PPI teve como objetivo principal ampliar o conhecimento dos alunos sobre a construção de páginas web, levando em consideração a modelagem de um banco de dados e do sistema proposto. Esta PPI previu atividades interativas nos laboratórios de informática do *campus*, integrando as disciplinas de Banco de Dados, Análise e Modelagem de Sistemas, Programação Web Orientada a Objetos e Português Instrumental.

Quanto a metodologia, os discentes, de modo individual, construíram uma página web, implementada em PHP, e modelaram o banco de dados e as funcionalidades do sistema, apresentando os diagramas produzidos em forma de relatório. O tema para a construção da página web foi escolhido pelo próprio discente. A etapa final, consistiu na socialização do trabalho, em forma de seminário para os colegas e os docentes do curso.

3.3 PPI No Curso Superior de Tecnologia em Sistemas para Internet

No Curso Superior de Tecnologia em Sistemas para Internet foram realizadas duas Práticas Profissionais Integradas, uma em cada semestre letivo do ano de 2015. A primeira PPI teve como objetivo principal a elaboração de uma página web, tendo como conteúdo um glossário de palavras técnicas de origem Inglesa que são utilizadas na área de Informática. Diante disso, esta PPI integrou as disciplinas de Introdução à Informática, Fundamentos da Computação, Inglês Técnico e Leitura e Produção Textual.

A metodologia utilizada para o desenvolvimento desta PPI ocorreu de modo semelhante à metodologia empregada no curso subsequente: 1) divisão da turma em pequenos grupos, sendo orientados a selecionarem conceitos abordados na disciplina de Fundamentos da Computação; 2) produzir um glossário dos termos em Inglês com

seus respectivos significados em Português; 3) construir uma página web; e, por fim, 4) socializar, em forma de seminário, a página web criada para os colegas e docentes do curso, além de um relatório escrito.

Também de modo semelhante à PPI aplicada ao curso subsequente, a segunda PPI do curso superior teve como objetivo principal ampliar os conceitos dos alunos no que tange a construção de páginas web, levando em consideração a modelagem do banco de dados e a interação humano-computador (acessibilidade). Esta PPI também previu atividades interativas nos laboratórios de informática e a interdisciplinaridade entre os seguintes componentes curriculares: Construção de Páginas Web, Banco de Dados I, Interação Humano-Computador e Metodologia Científica.

4 | CONSIDERAÇÕES FINAIS

Este artigo apresentou relatos da Prática Profissional Integrada (PPI) desenvolvida no eixo Informação e Comunicação, do Instituto Federal Farroupilha - *Campus* Santo Ângelo, no ano de 2015. A PPI é uma forma encontrada pelo IF Farroupilha de pôr em prática a interdisciplinaridade, contribuindo para o aprendizado do aluno.

A execução dessas práticas em nossa instituição de ensino gerou como resultado a publicação de alguns trabalhos, sendo um deles intitulado: “Números do lixo eletrônico, criatividade, criação e conscientização”, desenvolvido no Curso Técnico em Manutenção e Suporte em Informática - Integrado ao Ensino Médio. Diante dos resultados positivos, pretende-se dar continuidade ao desenvolvimento da Prática Profissional Integrada visando sempre melhorá-la, a fim de que, em um dado momento, a interdisciplinaridade esteja fortemente dissolvida na estrutura curricular de cada um dos cursos da instituição.

Além disso, percebe-se um grande envolvimento por parte dos alunos, com a pesquisa e realização das atividades propostas, permitindo que visualizem de forma prática, os conceitos trabalhados em sala de aula.

REFERÊNCIAS

CARLOS, J. G. **Interdisciplinaridade no ensino médio: desafios e potencialidades.** [S.l.]: Repositório Universidade de Brasília, 2007. Disponível em: <<http://repositorio.unb.br/bitstream/10482/>>. Acesso em: 03 de mai. 2016.

INSTITUTO FEDERAL FARROUPILHA. Instrução normativa nº 2, de 28 de fevereiro de 2013. Disponível em: <http://w2.iffarroupilha.edu.br/site/midias/arquivos/2013119154234593instrucao_normativa_n%C2%B0_02_2013_-_pratica_profissional_integrada.pdf>. Acesso em: 29 de abr. 2016.

IDENTIFICAÇÃO DE ESTILOS DE APRENDIZAGEM EM TURMAS DE NÍVEL TÉCNICO, GRADUAÇÃO E PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Janderson Jason Barbosa Aguiar

Universidade Federal de Campina Grande
(UFCG),
Campina Grande – PB

Joseana Macêdo Fachine Régis de Araújo

Universidade Federal de Campina Grande
(UFCG),
Campina Grande – PB

Evandro de Barros Costa

Universidade Federal de Alagoas (UFAL),
Maceió – AL

RESUMO: Cada indivíduo possui Estilos de Aprendizagem (EA), que são preferências relativas à maneira como representa e organiza novas informações. Identificar os EA pode favorecer o processo de ensino e aprendizagem. Neste capítulo, é apresentado um estudo referente aos EA de alunos de computação de diferentes níveis de ensino (técnico, graduação e pós-graduação). Para a detecção dos EA, utilizou-se o Índice de Estilos de Aprendizagem (ILS — *Index of Learning Styles*), questionário baseado no Modelo de Felder–Silverman. Com este estudo, verificou-se o grau de heterogeneidade concernente às preferências no modo de aprender dos alunos em diferentes níveis de ensino, mesmo eles sendo de uma área específica — computação.

PALAVRAS-CHAVE: Estilos de Aprendizagem; Modelo de Felder–Silverman; ILS; Educação em Computação.

ABSTRACT: Each individual has Learning Styles (LS), which are preferences for the way he represents and organizes new information. The identification of LS can facilitate the teaching and learning process. In this chapter, we present a study on LS of computer students from different educational levels (technical, undergraduate and graduate). For the detection of LS, we used the Index of Learning Styles (ILS), a questionnaire based on the Felder–Silverman Model. With this study, we verified the heterogeneity concerning preferences in the way students learn at different levels of education, even if they are of a specific area — computing.

KEYWORDS: Learning Styles; Felder–Silverman Model; ILS; Computer Education.

1 | INTRODUÇÃO

O processo de aprendizagem não é vivenciado por todos os indivíduos da mesma forma, sendo desenvolvidos — como resultado da bagagem hereditária, das experiências pessoais e das exigências do ambiente —

estilos de aprendizagem, que enfatizam algumas habilidades sobre as outras (KOLB, 1984).

Mesmo os estudantes de cursos específicos, como computação, apesar de geralmente terem características em comum — se comparados a estudantes de cursos de outras áreas —, ainda apresentam estilos distintos, não devendo ser tratados da mesma forma pelos facilitadores da aprendizagem (docentes, coordenadores de curso etc.).

Ao tomar consciência que cada estudante tem seus EA, os docentes tornam-se capazes de promover um ensino utilizando estratégias que promovam um aprendizado mais eficaz (FERNANDES, BENITTI e CUNHA, 2013).

Sobre as formas com que os aprendizes lidam particularmente/individualmente com as informações, o conceito de Estilos de Aprendizagem (EA) é cada vez mais presente em pesquisas na educação. Segundo Riding e Rayner (2000), esses estilos indicam a maneira preferida, individual e habitual com que os aprendizes organizam e representam novas informações durante o processo de aprendizagem.

Nessa temática, Coffield et al. (2004) realizaram uma revisão da literatura (utilizando os seguintes termos de busca: *Learning style/s*; *Cognitive style/s*; *Conative style/s*; *Thinking style/s*; *Learning preference/s*, *strategy/ies*, *orientation/s*) e identificaram 71 modelos de EA, utilizados em áreas como educação, psicologia, sociologia, entre outras. Dentre esses modelos, é possível destacar o modelo de Felder–Silverman (FELDER e SILVERMAN, 1988), o mais utilizado no contexto brasileiro e internacional em pesquisas relativas à adaptação e personalização de materiais de aprendizagem (AGUIAR, FECHINE e COSTA, 2014; VALASKI, MALUCELLI e REINEHR, 2011).

Inicialmente, o modelo de Felder–Silverman foi usado por instrutores e estudantes de engenharia e ciências, sendo posteriormente aplicado em várias outras disciplinas. Para Felder e Silverman (1988), os EA referem-se aos modos pelos quais os indivíduos preferem receber e processar as informações. Ao receber, a informação externa (captada pelos sentidos) e a informação interna (que surge introspectivamente) ficam disponíveis para o indivíduo, que seleciona o material a ser processado. Ao processar, o indivíduo pode fazer uso de simples memorização ou raciocínio indutivo ou dedutivo, reflexão ou ação, introspecção ou interação com outros indivíduos. Como resultado, o conteúdo abordado no material selecionado é aprendido de uma forma ou de outra ou, então, não é aprendido.

O modelo de Felder–Silverman contempla 4 (quatro) dimensões de EA: (1) Processamento (estilos Ativo e Reflexivo); (2) Percepção (estilos Sensorial e Intuitivo); (3) Entrada (estilos Visual e Verbal); e (4) Compreensão (estilos Sequencial e Global). Inicialmente havia a dimensão Organização (estilos Indutivo e Dedutivo), removida em 2002 por Felder, defendendo que o modo indutivo obtém melhores resultados e, uma vez que os estudantes preferem o modo dedutivo, não queria que sua pesquisa servisse como justificativa para os professores continuarem a usar o modo dedutivo em suas aulas (FELDER, 2002).

Em geral, os indivíduos Ativos aprendem por meio da experimentação ativa, e compreendem as informações mais eficientemente discutindo e aplicando os conceitos; por outro lado, os Reflexivos necessitam de um tempo sozinhos para pensar/refletir sobre as informações obtidas. Os indivíduos Sensoriais preferem lidar com situações concretas, dados e experimentos; por outro lado, os Intuitivos são inovadores, gostam de lidar com abstrações, conceitos e teorias. Os indivíduos com estilo Visual aprendem mais facilmente por meio de demonstrações, diagramas, figuras, filmes, fluxogramas; por outro lado, aqueles com estilo Verbal compreendem melhor as informações transmitidas por meio de palavras. Os indivíduos Sequenciais aprendem melhor quando o conceito é expresso de maneira contínua de dificuldade e complexidade; por outro lado, os indivíduos Globais são multidisciplinares, aprendendo em grandes saltos e lidando com os conteúdos de forma aleatória (FELDER e SILVERMAN, 1988; DIAS, GASPARINI e KEMCZINSK, 2009).

Baseado nas 4 dimensões, foi desenvolvido o Índice de Estilos de Aprendizagem (*Index of Learning Styles* – ILS), instrumento para determinar as preferências de aprendizagem do modelo de Felder–Silverman (FELDER e SOLOMAN, 1999).

Neste capítulo é apresentada uma pesquisa realizada com variados estudantes da área de computação (no ensino técnico de nível médio, na graduação e na pós-graduação *stricto sensu*), com base no ILS, visando contribuir para a melhoria do processo de ensino–aprendizagem nessa área. Na seção 2, são comentados alguns estudos relacionados. Na seção 3, é descrita a metodologia empregada. Na seção 4, são apresentados os resultados obtidos. Na seção 5, são realizadas algumas considerações finais.

2 | TRABALHOS RELACIONADOS

Muitos trabalhos na literatura usam teorias relacionadas à forma com que os estudantes preferem aprender.

Cury (2000) aplicou o ILS a 44 estudantes de engenharia, demonstrando que eles se apresentaram com EA preferencialmente Ativos, Sensoriais, Visuais e Sequenciais. Belhot, Freitas e Dornellas (2005) coletaram dados de 123 estudantes do curso de Engenharia de Produção Mecânica, utilizando o ILS e o inventário de Keirse e Bates, e traçaram um perfil de EA dominantes desses estudantes.

Santos e Mognon (2010) realizaram um estudo buscando identificar os EA predominantes nos estudantes universitários. Aplicaram o ILS a 242 estudantes de diversos cursos e, dentre os resultados, indicaram que os estudantes apresentaram predominância pelos estilos Ativo, Sensorial, Visual e Sequencial.

Oliveira (2012) estudou o impacto dos EA no desempenho acadêmico do ensino de contabilidade, utilizando o ILS aplicado a 276 estudantes e 13 professores, e concluindo que é possível notar influência no desempenho dos discentes.

Fernandes, Benitti e Cunha (2013) apresentaram um levantamento feito junto a 118 estudantes da área de computação (Sistemas para Internet; Engenharia de Computação; Ciência da Computação; Tecnologia em Jogos Digitais), com base no Inventário de Estilos de Aprendizagem de Kolb, a fim de obter subsídios para melhorar o processo de ensino–aprendizagem.

Embora as pesquisas comentadas acima — e muitas outras encontradas na literatura sobre EA — sejam relacionadas à pesquisa descrita neste capítulo, não foram encontrados trabalhos cujo foco tenha sido a extração (utilizando o ILS) e análise dos EA de estudantes de computação em diferentes níveis de ensino.

3 | METODOLOGIA

Com o objetivo de analisar semelhanças/diferenças de perfis de estudantes na área da computação, relativos aos seus EA, foi aplicado o questionário ILS a estudantes de nível técnico, graduandos e pós-graduandos (mestrandos e doutorandos) em Ciência da Computação.

O ILS abrange 44 questões objetivas (alternativa “a” ou “b”), sendo 11 para cada uma das 4 dimensões. As questões 1, 5, 9, 13, 17, 21, 25, 29, 33, 37 e 41 são relativas à dimensão Processamento, sendo a primeira alternativa (letra “a”) representante do valor Ativo e a segunda (letra “b”) do valor Reflexivo. As questões 2, 6, 10, 14, 18, 22, 26, 30, 34, 38 e 42 referem-se à dimensão Percepção, sendo a primeira alternativa representante do valor Sensorial e a segunda alternativa representante do valor Intuitivo. As questões relativas à dimensão Entrada são 3, 7, 11, 15, 19, 23, 27, 31, 35, 39 e 43, sendo que a primeira alternativa representa o valor Visual e a segunda o valor Verbal. As demais questões (4, 8, 12, 16, 20, 24, 28, 32, 36, 40 e 44) são relacionadas à dimensão Entendimento, sendo o valor Sequencial representado pela primeira alternativa e o valor Global pela segunda.

As respostas às questões do ILS fornecem duas pontuações, para cada uma das quatro dimensões, correspondentes aos dois estilos da dimensão. A diferença entre as duas pontuações indica, dentre os dois estilos, qual é o predominante/preferido pelo respondente, além de permitir conhecer a intensidade dessa preferência — pode ser leve ou quase inexistente (pontuações 1 e 3), moderada (pontuações 5 e 7) ou forte (pontuações 9 e 11) para um ou outro estilo do par da dimensão.

Os estudos de Zywno (2003), Felder e Spurlin (2005) e Litzinger et al. (2007) concordam que o ILS é um instrumento confiável, válido e adequado para identificação dos EA, embora seja recomendado que as pesquisas com tal instrumento continuem a ser realizadas.

Ao todo, foram obtidas 118 respostas, sendo 61 de estudantes de nível técnico, 36 de estudantes graduandos e 21 de estudantes pós-graduandos (mestrandos ou doutorandos).

4 | RESULTADOS

Inicialmente, foram analisados os dados e gerados gráficos comparativos considerando a binaridade de estilos por dimensão (Figura 1).

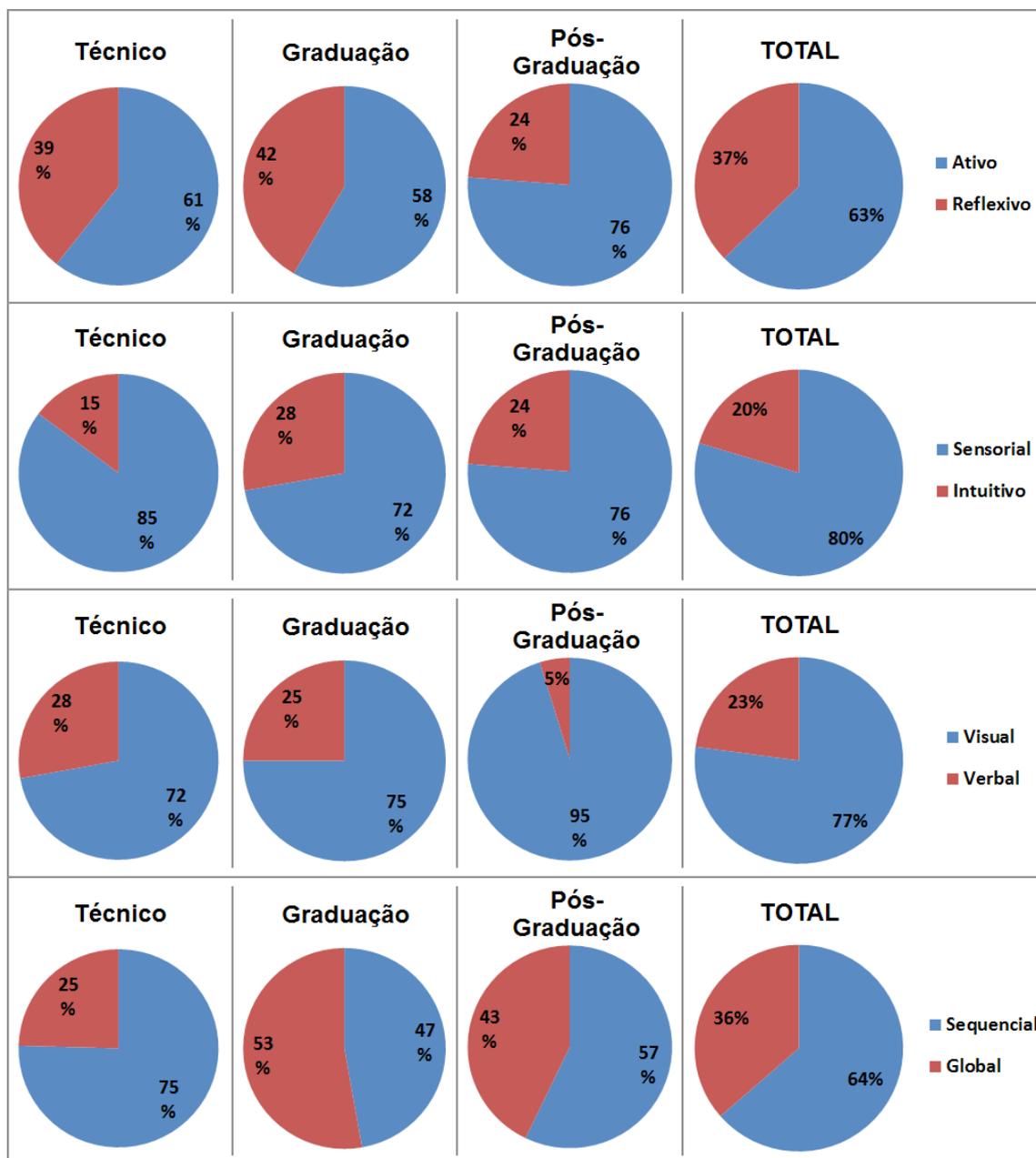


Figura 1. Comparação do resultado da aplicação do questionário ILS, em níveis diferentes de ensino de computação, considerando a binaridade dos estilos por dimensão.

Fonte: Elaborado pelos autores

De maneira geral, é possível afirmar, com base na Figura 1, que os estudantes de computação tendem a ser mais Ativos, Sensoriais, Visuais e Sequenciais. Esse resultado corrobora com os obtidos em pesquisas anteriormente citadas (CURY, 2000; SANTOS e MOGNON, 2010). Entretanto, a parcela de estudantes com os estilos opostos (Reflexivos, Intuitivos, Verbais e Globais) é significativa em muitos casos. Em

relação aos estudantes de graduação, por exemplo, há leve predominância do estilo Global em vez do Sequencial.

Já se percebe, com essa abordagem comparativa, a diferença entre os estudantes em todos os três níveis de ensino considerados neste estudo. Entretanto, julgou-se interessante também considerar a intensidade das preferências por estilo e, assim, os resultados foram categorizados em leve, moderado e forte (Figura 2).

Com base na Figura 2, é possível afirmar que é a minoria dos estudantes que apresenta algum estilo com o grau de intensidade forte. Em relação à dimensão Processamento, a maioria geralmente é levemente Ativa ou levemente Reflexiva. Em relação à dimensão Percepção, a maioria é levemente ou moderadamente Sensorial. Em relação à dimensão Entrada, a maioria é levemente ou moderadamente Visual. Em relação à dimensão Compreensão, embora haja predominância de ser levemente Sequencial, muitos também são levemente Globais ou moderadamente Sequenciais.

A partir da visualização gráfica dos dados relativos aos EA, considerando graus de intensidade, nota-se ainda mais como os estudantes podem ser diferentes.

Por fim, com base nos valores brutos — isto é, nem categorizados binariamente (Figura 1), nem por grau de intensidade (Figura 2) —, foi gerado o gráfico da Figura 3. No eixo horizontal, têm-se a variação para os 118 estudantes que fizeram parte desta pesquisa, respondendo o ILS.

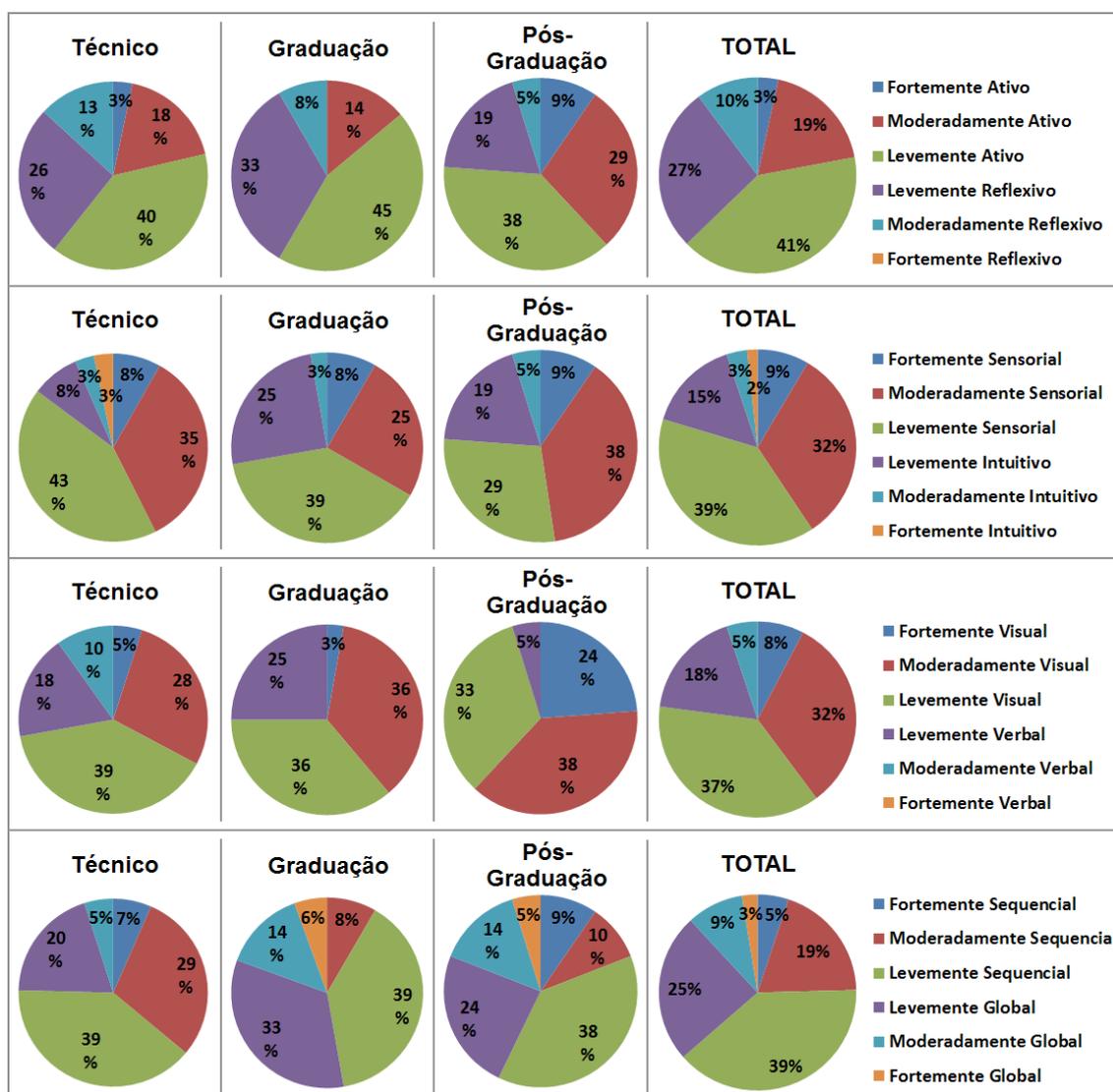


Figura 2. Comparação do resultado da aplicação do questionário ILS em níveis diferentes de ensino de computação, considerando três graus de intensidade (leve, moderado, forte) por estilo, em cada dimensão.

Fonte: Elaborado pelos autores

No eixo vertical do gráfico da Figura 3, os valores indicam a porcentagem (de 0% a 100%) para cada dimensão, sendo: para a dimensão Processamento, valores próximos a 0% representativos do estilo Ativo e valores próximos a 100% representativos do estilo Reflexivo; para a dimensão Processamento, valores próximos a 0% representativos do estilo Sensorial e valores próximos a 100% representativos do estilo Intuitivo; para a dimensão Processamento, valores próximos a 0% representativos do estilo Visual e valores próximos a 100% representativos do estilo Verbal; e para a dimensão Processamento, valores próximos a 0% representativos do estilo Sequencial e valores próximos a 100% representativos do estilo Global.

Em relação aos quatro ícones coloridos da Figura 3 (que representam as dimensões de EA), é visível grande variação. Este gráfico ilustra nitidamente a heterogeneidade relativa aos EA de estudantes da área de computação e possibilita ter uma ideia das quatro dimensões combinadas por indivíduo.

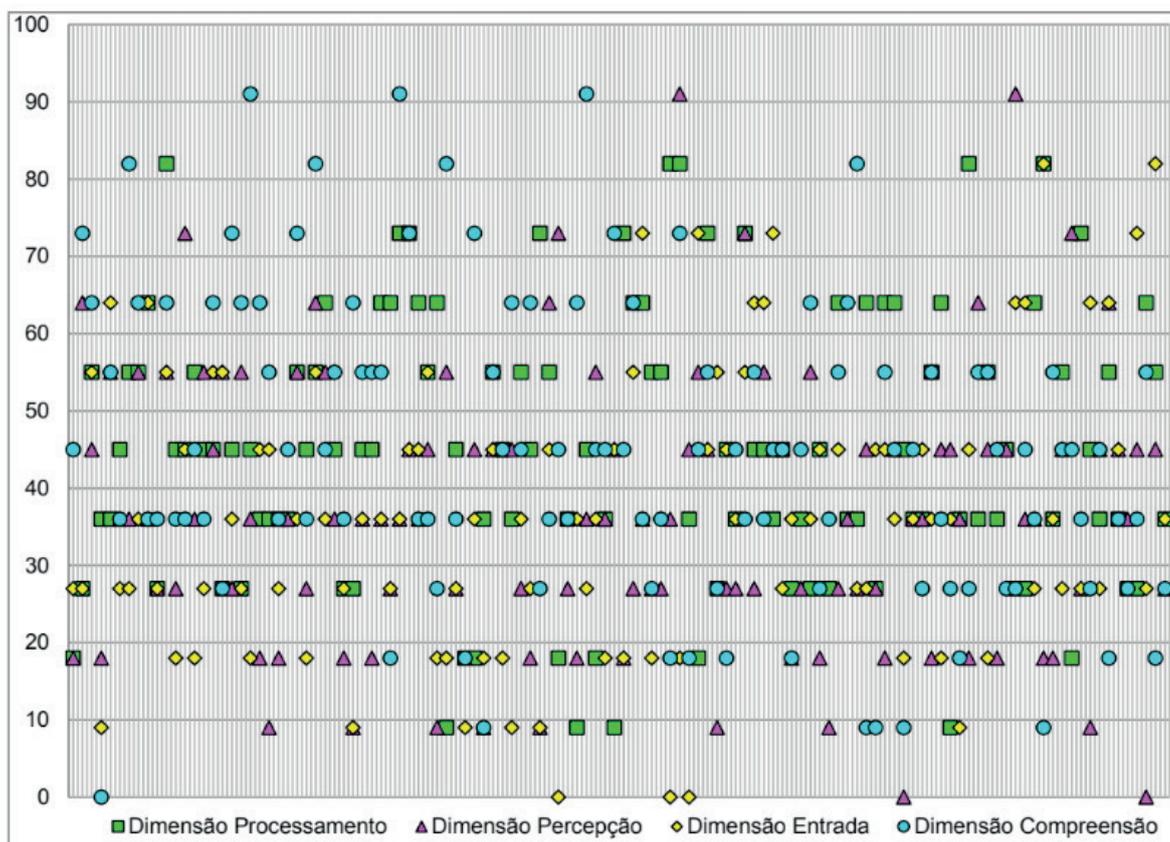


Figura 3. Ilustração da heterogeneidade dos Estilos de Aprendizagem dos 118 estudantes participantes desta pesquisa.

Fonte: Elaborado pelos autores

5 | CONSIDERAÇÕES FINAIS

As teorias relacionadas a EA são essenciais para dar suporte às diferenças intrínsecas ao modo particular de aprender dos estudantes de computação. Como abordado na Seção 4, essas diferenças abrangem desde turmas de estudantes em cursos técnicos de nível médio a estudantes de programas de pós-graduação.

Além da detecção desses EA ser bastante útil, por exemplo, na personalização de ambientes em sistemas de apoio à educação na modalidade a distância, o simples uso do questionário ILS, em cursos presenciais, é um meio interessante para o professor conhecer seus estudantes e, com base nos perfis encontrados, adaptar seus planos de aula, com abordagens metodológicas e avaliativas que facilitem o processo de ensino–aprendizagem. Em relação a pós-graduandos, por exemplo, os orientadores podem fazer uso dessa teoria para sugerir pesquisas/tarefas que se adaptem aos EA de seus orientandos — e não apenas indicando temas e direções que às vezes estes pouco se interessam.

Especialmente na Educação em Computação, a criação de formulários on-line por parte dos professores não requer muitos conhecimentos além dos quais estes já possuem. Tais docentes podem facilmente automatizar o processo de geração de resultados ao implementar, na linguagem de programação que lhe convier, o algoritmo para extrair os valores dos EA com base nas respostas das 44 questões do ILS (comentado na seção 3).

Além disso, é interessante comentar que há iniciativas de uso do modelo de Felder–Silverman que não utilizam o ILS, sendo possível citar as pesquisas de Popescu, Trigano e Badica (2007), Graf, Kinshuk e Liu (2008) e Dorça et al. (2013). Eles sugerem a utilização de métodos baseados na análise do comportamento implícito do estudante em sistemas de aprendizagem. Graf, Kinshuk e Liu (2008) considera, dentre outras relações, que, se um estudante frequentemente visitou exercícios, há preferência por um estilo Ativo de aprendizagem. Nas pesquisas de Dorça et al. (2013) e Popescu, Trigano e Badica (2007), além da obtenção dos estilos, são realizadas atualizações dinamicamente, considerando o desempenho na avaliação de uma seção de aprendizagem (DORÇA et al., 2013) e regras como, por exemplo, um valor alto no número de postagens em fórum indica um estilo Ativo e Verbal (POPESCU, TRIGANO e BADICA, 2007).

Como apresentado por Aguiar, Fachine e Costa (2014), há um aumento de pesquisas, nos últimos anos, considerando o conceito de EA. É importante cada vez mais a utilização desse conceito para que os estudantes não sejam tratados de forma igual em relação ao modo como adquirem e produzem conhecimento.

Ao considerar um universo de alunos relativamente pequeno, a apresentação dos resultados desta pesquisa (sumarizados nas Figuras 1 e 2) não tem o objetivo de generalizar que as turmas de computação possuem determinado perfil. Esta pesquisa visou evidenciar a diferença de perfis relativos a EA de alunos de computação,

mostrando a importância de considerar essas diferenças no processo de ensino–aprendizagem nessa área. Para fornecer uma generalização confiável do perfil do aluno de computação, seria necessário considerar várias instituições de ensino (de nível técnico, graduação e pós-graduação) em diferentes locais e regiões. A partir disso, poderia ser realizada, por exemplo, uma análise mais aprofundada, até mesmo com apoio de psicólogos, buscando entender como os alunos de computação, em determinado nível de ensino, possuem um estilo mais ou menos evidente em relação a alunos em outros níveis de ensino.

Almeja-se, com a divulgação desta pesquisa — que foi publicada primeiramente nos Anais do XXIII Workshop sobre Educação em Computação (AGUIAR, FECHINE e COSTA, 2015b) —, motivar pesquisadores e educadores da área da computação a fazerem uso e realizarem pesquisas sobre o conceito de EA, uma vez que, como apresentado neste capítulo, é válido e necessário considerar as particularidades dos estudantes dessa área independente do nível de ensino.

Destaca-se que este trabalho faz parte de um projeto de pesquisa no qual estão sendo estudadas estratégias, considerando os EA dos estudantes, visando à melhoria das recomendações em Sistemas de Recomendação Educacionais (AGUIAR, ARAÚJO e COSTA, 2018; AGUIAR et al., 2017; AGUIAR, FECHINE e COSTA, 2015a).

6 | AGRADECIMENTOS

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro. Aos discentes participantes da pesquisa, pelas respostas ao questionário ILS. À Sociedade Brasileira de Computação (SBC), por dar o direito aos autores de publicarem este trabalho inicialmente publicado nos Anais do XXIII Workshop sobre Educação em Computação (WEI 2015).

REFERÊNCIAS

AGUIAR, J. J. B.; ARAÚJO, J. M. F. R.; COSTA, E. B. **Recomendação de Objetos de Aprendizagem utilizando Filtragem Colaborativa baseada em Tendências e em Estilos de Aprendizagem**. In: Anais do XXIX Simpósio Brasileiro de Informática na Educação, Fortaleza, 2018. No prelo.

AGUIAR, J. J. B.; BARBOSA, A. F.; ARAÚJO, J. M. F. R.; COSTA, E. B. **Um Estudo sobre a Influência das Dimensões do Modelo Felder-Silverman na Recomendação de Recursos Educacionais baseada nos Estilos de Aprendizagem dos Alunos**. In: Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE), Recife, p. 1277–1286, 2017. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/7656>>. Acesso em: 21 ago. 2018.

AGUIAR, J. J. B.; FECHINE, J. M.; COSTA, E. B. **Estilos Cognitivos e Estilos de Aprendizagem em Informática na Educação: um mapeamento sistemático focado no SBIE, WIE e RBIE**. In: Anais do 25º Simpósio Brasileiro de Informática na Educação, Dourados, p. 441–450, 2014. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2972/2705>>. Acesso em: 20 ago. 2018.

AGUIAR, J. J. B.; FECHINE, J. M.; COSTA, E. B. **Recomendação de Objetos de Aprendizagem**

baseada na Popularidade dos Objetos e nos Estilos de Aprendizagem dos Alunos. In: Anais do XXVI Simpósio Brasileiro de Informática na Educação, Maceió, p. 1147-1156, 2015a. Disponível em: <<http://br-ie.org/pub/index.php/sbie/article/view/5438>>. Acesso em: 21 ago. 2018.

AGUIAR, J. J. B.; FECHINE, J. M.; COSTA, E. B. **Utilização do Índice de Estilos de Aprendizagem de Felder–Soloman em Turmas de Nível Técnico, Graduação e Pós-Graduação em Computação.** In: Anais do XXXV Congresso da Sociedade Brasileira de Computação (CSBC) — XXIII Workshop sobre Educação em Computação (WEI), Recife, 2015b. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2015/035.pdf>>. Acesso em: 20 ago. 2018.

BELHOT, R. V.; FREITAS, A. A.; DORNELLAS, D. V. **Benefícios do Conhecimento dos Estilos de Aprendizagem no Ensino de Engenharia de Produção.** In: XXXIII Congresso Brasileiro de Ensino de Engenharia (COBENGE), Campina Grande, 2005. Disponível em: <http://www2.eesc.usp.br/aprende/images/arquivos/Beneficios_Conhecimento_Estilos_Aprendizagem_no_Ensino_Engenharia_Producao.pdf>. Acesso em: 20 ago. 2018.

COFFIELD, F.; MOSELEY, D.; HALL, E.; ECCLESTONE, K. **Should we be using learning styles? What research has to say to practice.** London, Learning and Skills Research Centre, Learning and Skills Development Agency, 2004. Disponível em: <http://www.itlifejimbutnotasweknowit.org.uk/files/LSRC_LearningStyles.pdf>. Acesso em 20 ago. 2018.

CURY, H. N. **Estilos de Aprendizagem de Alunos de Engenharia.** In: XXVIII Congresso Brasileiro de Ensino de Engenharia (COBENGE), Ouro Preto, 2000. Disponível em: <<http://www.abenge.org.br/cobenge/arquivos/19/artigos/169.PDF>>. Acesso em: 20 ago. 2018.

DIAS, C. C. L.; GASPARINI, I.; KEMCZINSK, A. **Identificação dos estilos cognitivos de aprendizagem através da interação em um Ambiente EAD.** In: XXIX Congresso da Sociedade Brasileira de Computação — XVII Workshop sobre Educação em Informática (WEI), p. 489–498, 2009. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2009/011.pdf>>. Acesso em: 20 ago. 2018.

DORÇA, F. A.; LIMA, L. V.; FERNANDES, M. A.; LOPES, C. R. **Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis.** Expert Systems with Applications, v. 40, n. 6, p. 2092–2101, 2013. <https://doi.org/10.1016/j.eswa.2012.10.014>

FELDER, R. M. **Author’s Preface — June 2002.** In: FELDER, R. M.; SILVERMAN, L. K. Learning and Learning and Teaching Styles in Engineering Education, 2002. Disponível em: <<http://www.ncsu.edu/felder-public/Papers/LS-1988.pdf>>. Acesso em: 20 ago. 2018.

FELDER, R. M.; SILVERMAN, L. K. **Learning and Teaching Styles in Engineering Education.** Journal of Engineering Education, v. 78, n. 7, p. 674–681, 1988. Disponível em: <http://www.ncsu.edu/felder-public/Learning_Styles.html>. Acesso em: 20 ago. 2018.

FELDER, R. M.; SOLOMAN, B. A. **Index of Learning Styles (ILS).** 1999. Disponível em: <<http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSpage.html>>. Acesso em: 20 ago. 2018.

FELDER, R.; SPURLIN, J. **Applications, Reliability, and Validity of the Index of Learning Styles.** International Journal of Engineering Education, v. 21, p. 103–112, 2005. Disponível em: <[http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS_Validation\(IJEE\).pdf](http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS_Validation(IJEE).pdf)>. Acesso em: 20 ago. 2018.

FERNANDES, A. M. R.; BENITTI, F. B. V.; CUNHA, F. S. **Aplicando o Inventário de Estilos de Aprendizagem de Kolb como Ferramenta de Apoio ao Processo de Ensino Aprendizagem em Cursos de Computação.** In: XXXIII Congresso da Sociedade Brasileira de Computação — XXI Workshop sobre Educação em Informática (WEI), p. 420–425, 2013. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wei/2013/008.pdf>>. Acesso em: 20 ago. 2018.

GRAF, S.; KINSHUK; LIU, T-C. **Identifying Learning Styles in Learning Management Systems by Using Indications from Students’ Behaviour.** In: ICALT ’08 — Eighth IEEE International Conference

on Advanced Learning Technologies, p. 482–486, 2008. <https://doi.org/10.1109/ICALT.2008.84>

KOLB, D. A. **Experiential learning: experience as the source of learning and development**. Englewood Cliffs: Prentice-Hall. 1984.

LITZINGER, T. A.; LEE, S. H.; WISE, J. C.; FELDER, R. M. **A Psychometric Study of the Index of Learning Styles**. *Journal of Engineering Education*, v. 96, n. 4, p. 309–319, 2007. Disponível em: <[http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS_Validation\(JEE-2007\).pdf](http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ILS_Validation(JEE-2007).pdf)>. Acesso em: 20 ago. 2018.

OLIVEIRA, D. E. **Impacto dos estilos de aprendizagem no desempenho acadêmico do ensino de contabilidade: uma análise dos estudantes da Universidade Federal do Rio Grande do Norte**. Dissertação (Mestrado em Ciências Contábeis), Programa Multiinstitucional e Inter-Regional de Pós-Graduação em Ciências Contábeis, Universidade de Brasília, Universidade Federal da Paraíba, Universidade Federal do Rio Grande do Norte, Natal, 2012. Disponível em: <<http://repositorio.unb.br/handle/10482/12748>>. Acesso em: 20 ago. 2018.

POPESCU, E.; TRIGANO, P.; BADICA, C. **Adaptive Educational Hypermedia Systems: A Focus on Learning Styles**. In: EUROCON 2007 — The International Conference on “Computer as a Tool”, p. 2473–2478, 2007. <https://doi.org/10.1109/EURCON.2007.4400580>

RIDING, R.; RAYNER, S. **Cognitive styles and learning Strategies — understanding style differences in learning and behavior**. London: David Fulton Publishers, 2000.

SANTOS, A. A. A.; MOGNON, J. F. **Estilos de aprendizagem em estudantes universitários**. *Boletim de Psicologia*, São Paulo, v. 60, n. 133, 2010. Disponível em: <http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S0006-59432010000200009&lng=pt&nrm=iso>. Acesso em: 20 ago. 2018.

VALASKI, J.; MALUCELLI, A.; REINEHR, S. **Revisão dos Modelos de Estilos de Aprendizagem Aplicados à Adaptação e Personalização dos Materiais de Aprendizagem**. In: XXII Simpósio Brasileiro de Informática na Educação — SBIE, p. 844–847, 2011. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/1843/1605>>. Acesso em: 20 ago. 2018.

ZYWNO, M. S. **A Contribution to Validation of Score Meaning for Felder-Soloman’s Index of Learning Styles**. In: Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition. Nashville, Tennessee, 2003. Disponível em: <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/Zywno_Validation_Study.pdf>. Acesso em: 20 ago. 2018.

EXPLORAÇÃO EFICIENTE EM ESPAÇOS DE PROJETO DE COMUNICAÇÃO EM PLATAFORMAS MULTIPROCESSADORAS BASEADAS EM BARRAMENTOS

**Guilherme Álvaro Rodrigues Maia
Esmeraldo**

Instituto Federal de Educação, Ciência e
Tecnologia do Ceará *campus* Crato
Crato – Ceará

Edna Natividade da Silva Barros

Centro de Informática, Universidade Federal de
Pernambuco
Recife – Pernambuco

RESUMO: Sistemas embarcados se tornaram muito complexos, devido aos novos requisitos das aplicações, necessitando assim de mais de um processador, diferentes tipos de memória e dispositivos variados. Para tornar a comunicação mais eficiente entre esses dispositivos, deve-se customizar sua estrutura de comunicação. Quando se utiliza barramentos para a comunicação, muitos parâmetros têm que ser considerados, pois podem impactar diretamente no desempenho de todo o sistema. Este trabalho apresenta uma nova abordagem que permite explorar com grande desempenho, em relação às abordagens tradicionais de simulação, as opções de configuração de barramento de todo o espaço de projeto.

PALAVRAS-CHAVE: Análise de Comunicação, Exploração de Espaço de Projeto. Barramento.

ABSTRACT: Embedded systems have become

very complex due to the new applications requirements, thereby needing more than one processor, different types of memory and peripherals. To make more efficient the communication among these devices, the designer must customize the on-chip communication structure. When using buses for the communication, many parameters must be considered, since they can directly impact on the performance of the entire system. This paper presents a new approach that, compared to traditional simulation approaches, aims to increase the performance of the exploration task of the bus configuration options included in the design space.

KEYWORDS: Communication Analysis, Design Space Exploration, Bus.

1 | INTRODUÇÃO

Devido ao desenvolvimento das tecnologias de circuitos integrados (CIs), que hoje permitem integrar mais de um bilhão de transistores em um único chip (BAKLANOV et al., 2015), aos requisitos mais rígidos das aplicações e ciclos cada vez menores para lançamento de novos produtos, os sistemas embarcados, bem como seu projeto, passaram a ficar mais complexos. Então surgiram os

Systems-on-Chip (SoCs). Um *SoC* é um CI que implementa muitas ou todas as funções de um sistema eletrônico completo (JERRAYA; WOLF, 2004), podendo incluir memórias, microprocessador, interfaces com periféricos, lógica de controle de E/S, conversores de dados e outros componentes que abrangem sistemas computacionais completos (WILD; HEKERDOF; OHLENDORF, 2006). Com a redução do custo e do tamanho dos microprocessadores embarcados, dez ou mais microprocessadores podem ser integrados em um único chip para formar um *Multi-Processor System-on-Chip (MPSoC)*. Na prática, um *MPSoC* é um *SoC* que inclui vários processadores conectados por uma arquitetura de comunicação (JOO; KIM; HA, 2009).

Arquitetura de comunicação *on-chip* refere-se a uma estrutura física que integra os componentes de processamento e armazenamento de *SoCs* e disponibiliza um mecanismo para troca de dados e informações de controle entre eles (JERRAYA; WOLF, 2004). Arquiteturas baseadas em barramentos são atualmente bastante populares para implementação da estrutura de comunicação *on-chip* em projeto de *MPSoCs*, porque são simples de projetar e eficientes de implementar, com consequente redução do tempo de projeto, além de ocupar pequenas áreas no chip e ser extensível (PAUL et al., 2015).

Porém, por ser um recurso compartilhado, o barramento torna-se o fator limitante no desempenho da comunicação de um sistema embarcado (LAHIRI; RAGHUNATHAN, 2004). Além disso, selecionar e reconfigurar arquiteturas de comunicação baseadas em barramentos para atender a todos os requisitos de comunicação da aplicação, é um processo que consome muito tempo (PASRICHA et al., 2004). Isto deve-se ao grande conjunto de opções de configuração ao se considerar: diferentes topologias de barramentos customizáveis, protocolos de comunicação, tipos de transferências, larguras de barramentos de dados e endereço, frequências de operação, tamanho dos *buffers* e esquemas de arbitragem. Especialistas podem, através de experiência e intuição, acelerar o processo de escolha de uma configuração, porém, em sistemas com dezenas de parâmetros de configuração, essa tarefa pode se tornar muito difícil e, então, os especialistas podem ser levados a cometer erros e escolher uma configuração ótima local e não global (SHIN et al., 2004).

No projeto de *MPSoCs* há uma clara tendência em se utilizar a abordagem de Projeto Baseado em Plataformas (PBP) (NUZZO et al., 2015). Nesta abordagem, o sistema a ser desenvolvido é, inicialmente, especificado através de uma descrição em alto nível, como descrições textuais e/ou diagramas. As funções do sistema, contidas nessa especificação, são selecionadas para serem implementadas em *software* (que será mapeado para microprocessadores) ou em *hardware* (componentes de *hardware* de finalidade específica, implementados a partir de reuso de outros componentes, ou compilados a partir de ferramentas de síntese) (LAHIRI, RAGHUNATHAN; DEY, 2001). Estes componentes de *hardware* fazem parte de uma arquitetura predefinida, conhecida como plataforma, que pode ser modificada para que sua estrutura possa ser adaptada às necessidades da aplicação.

Desde que os componentes da plataforma compartilharão dados e se comunicar para implementar as funcionalidades do sistema, deve-se refinar os parâmetros de configuração da arquitetura de comunicação para atender às necessidades de comunicação dos componentes. Pesquisadores têm trabalhado no desenvolvimento de técnicas de análise considerando diferentes métricas, tais como desempenho, potência, custo do sistema, etc., para guiar os passos de particionamento/mapeamento. Nestes trabalhos, técnicas para particionamento automático podem: 1) ignorar completamente comunicação entre os componentes ou 2) utilizar modelos simples de comunicação para guiar o passo de particionamento/mapeamento (LAHIRI, RAGHUNATHAN; DEY, 2001).

Este trabalho apresenta uma nova abordagem para dar suporte ao projeto da estrutura de comunicação em plataformas multiprocessadoras. A abordagem inclui uma técnica de análise híbrida capaz de estimar o desempenho de comunicação de uma plataforma para todas as opções de configuração de barramento contidas no espaço de projeto. A técnica proposta suporta a análise da comunicação nos processos de seleção e refinamento das arquiteturas de comunicação após a aplicação ter sido particionada e mapeada para os componentes da plataforma.

2 | TRABALHOS RELACIONADOS

Dado o aumento da complexidade dos sistemas, o número de parâmetros das arquiteturas de comunicação das plataformas tem aumentado consideravelmente, levando à necessidade de examinar milhares de configurações diferentes para se obter a melhor arquitetura (LAHIRI, RAGHUNATHAN; DEY, 2001). Dependendo do tamanho do espaço de projeto, esse processo pode consumir muito tempo, dias ou meses, até se encontrar a melhor arquitetura ou apenas uma que atenda a todas as restrições de projeto.

Para acelerar a busca no espaço de projeto, muitos trabalhos têm focado no desenvolvimento de técnicas para suporte à análise de comunicação, disponibilizando estimativas do impacto da arquitetura de comunicação no desempenho e consumo de potência total do sistema (JERRAYA; WOLF, 2004). Neste trabalho, estas técnicas foram divididas, de um modo geral, em três categorias: (1) técnicas baseadas em simulação, (2) abordagens baseadas em estimação estática e (3) técnicas híbridas. São técnicas bastante distintas mas, em muitos casos, são complementares para avaliação de mecanismos de comunicação.

O primeiro grupo inclui as abordagens baseadas na simulação completa de sistema. Estas abordagens incluem modelos de simulação que podem incluir componentes de hardware e sua comunicação em diferentes níveis de abstração (PASRICHA; DUTT; BEN-ROMDHALE, 2005). O uso de abstrações de comunicação permite estabelecer um compromisso entre precisão e desempenho das estimativas.

Entretanto, essas técnicas requerem uma simulação completa do sistema para avaliar cada ponto do espaço de projeto, implicando em alto custo computacional.

Outros tipos de técnicas buscam, através de modelos matemático/estatísticos (KIM; IM; HA, 2003) (CHO; CHOI; CHO, 2006), estimar estaticamente as métricas do sistema, como, por exemplo, desempenho ou potência. O uso deste tipo de técnica só é possível em sistemas onde as computações e comunicações podem ser previamente agendadas (LAHIRI; RAGHUNATHAN; DEY, 2001). Além disso, podem ser demasiadamente otimistas, por não considerarem os efeitos dinâmicos da aplicação, como latências por contenção de barramento, ou pessimistas, assumindo o pior caso para contenção (GASTEIER; GLESNER, 1999). Por não incluir detalhes suficientes de comunicação, a utilização deste tipo de técnica pode resultar em estimativas imprecisas.

O último grupo inclui as abordagens híbridas (LAHIRI; RAGHUNATHAN; DEY, 2001) (SHIN et al., 2004) (WILD; HEKERDOF; OHLENDORF, 2006) (JOO; KIM; HA, 2009) (LUKASIEWYCZ et al., 2009), que agregam diversos tipos de técnicas, como por exemplo, heurísticas ou estimativas estáticas, às técnicas de simulação completa de sistema. Tais abordagens buscam aumento de desempenho das simulações e/ou exploração, enquanto procuram disponibilizar estimativas precisas da comunicação. Porém, na prática, algumas dessas abordagens não permitem explorar espaços de projeto maiores para refinamento de comunicação.

Este artigo apresenta uma abordagem híbrida que utiliza simulação completa de sistema para obter um perfil de comunicação da aplicação e, a partir dele, estimar o desempenho de comunicação para cada configuração de barramento. Com o uso da técnica proposta, a tarefa de exploração do espaço de projeto de comunicação se torna mais eficiente, pela redução do número de simulações necessárias para exploração do espaço de projeto, e é eficaz, por fazer uso de estimativas precisas de desempenho da comunicação da aplicação. Adicionalmente, a técnica proposta dispõe de suporte ao detalhamento das transações de barramento para que o projetista possa analisar todo o processo de comunicação e identificar os gargalos.

3 | ABORDAGEM PROPOSTA

O fluxo da abordagem proposta para o projeto da estrutura de comunicação de uma plataforma virtual inclui cinco fases, as quais são ilustradas na Figura 1.

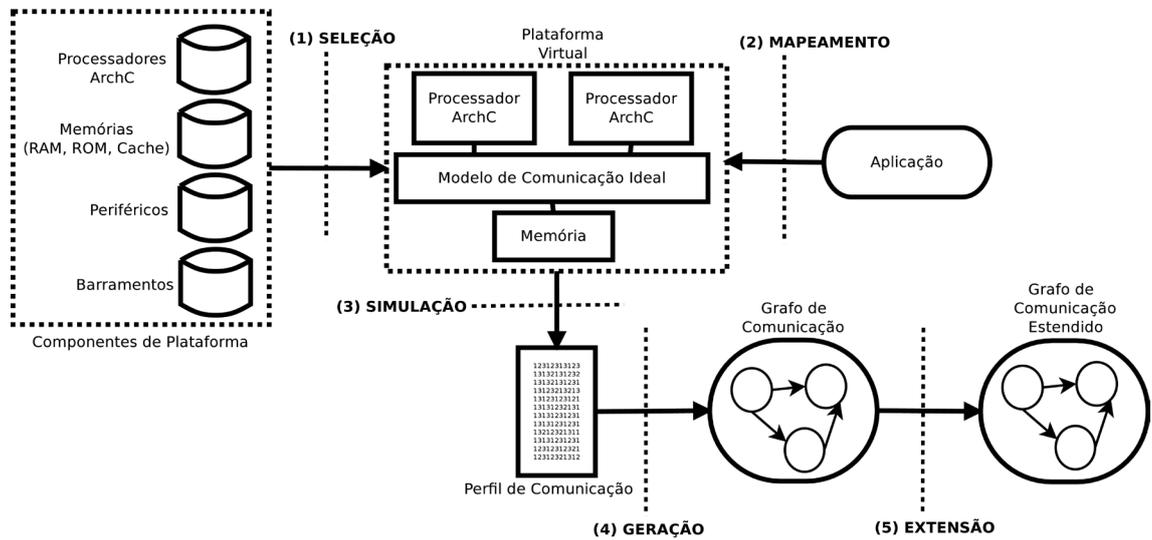


Figura 1. Fluxo proposto para o projeto da estrutura de comunicação.

A primeira e a segunda fase consistem da (1) *seleção* dos componentes de hardware que irão compor a plataforma virtual e do (2) *mapeamento* da aplicação, respectivamente. O modelo de plataforma adotado neste trabalho inclui processadores descritos em *ArchC* (ARCHC, 2018) e memórias RAM, descritas em *SystemC TLM* (IEEE, 2011), com capacidade de armazenamento, tamanho de palavra e latências parametrizáveis.

Para modelar a comunicação entre esses componentes, foi proposto um modelo de barramento genérico, também descrito em *SystemC*. Este modelo de barramento permite a simulação considerando um modelo de estrutura de comunicação ideal, o qual não introduz latências nos processos de comunicação, que permite transferências em paralelo, desde que os periféricos forneçam suporte para este tipo de acesso, e com diferentes tamanhos de dados. Adicionalmente, este modelo registra as informações de todas as transferências, possibilitando a extração de um perfil da comunicação da aplicação.

A terceira fase, consiste da (3) *simulação* da plataforma virtual com o modelo de barramento genérico, através da execução do programa gerado a partir do código *SystemC*. Nesta fase, é gerado um arquivo em disco, o perfil de comunicação, com o registro de todos os eventos de comunicação realizados no modelo de barramento genérico proposto. Formalmente, o perfil de comunicação é dado por:

Seja P um perfil de comunicação, que será dado por $P=\{r_k\}$, para $k=1,2,\dots,n$, onde r_k representa cada um dos registros de transações de barramento contidas no perfil, que será dada pela 3-upla $r_k=(m, e, t)$, onde: m será o identificador do mestre do barramento que originou a transação; e será o evento de transação, que pode ser de (q) requisição de uso do barramento, (c) envio de sinais de controle, (d) envio de dados e (l) liberação do barramento; t será número de sequência da transação.

O perfil de comunicação inclui registros de todos os eventos de comunicação entre os componentes da plataforma, porém sem incluir as características específicas

do barramento, tais como tamanho de palavra ou latências de transferências e contenção. Assim, é proposto que esse perfil seja modificado para incluir características de barramentos reais, e que seja utilizado para estimar o desempenho da aplicação na plataforma com barramentos reais. A modificação da estrutura que representa o perfil de comunicação é obtida através de: 1) alteração da sequência de eventos das transações, impondo assim um modelo de prioridades de acesso (escalonamento de transferências por arbitragem) e 2) adição de características específicas de modelos reais de barramentos, tais como largura de barramentos de dados, latências inerentes aos protocolos de transferências e frequência de operação.

O modelo adotado para representar em memória os perfis de comunicação, e com isso otimizar o acesso aos dados, é uma extensão do modelo proposto em (LAHIRI; RAGHUNATHAN; DEY, 2001), que utilizou grafos – chamado de Grafos de Comunicação (GC). Neles, os vértices são utilizados para representar eventos de computação e comunicação. Tais eventos de comunicação consideram apenas transações atômicas de barramento, indivisíveis, descaracterizando cenários reais de comunicação embarcada, como por exemplo, interrupção de uma transferência por preempção de arbitragem. Assim, a primeira contribuição do trabalho aqui apresentado é a extensão desse modelo de GC, para suportar a modelagem das fases internas de uma transação (requisição de uso do barramento, envio de sinais de controle, transferência de dados e liberação) no perfil de comunicação. Com isso, busca-se tornar as estimativas mais precisas. Formalmente, um GC, neste trabalho, é dado por:

Seja G um grafo de comunicação, dado por $G=(V,E)$, onde V é o conjunto de vértices, que representam as transações, e E o conjunto de arcos, os quais representam a sequência entre as transações em V . Seja a transação v , tal que $v \in V$, v será dada por $v=(m, e, t)$, onde m é o identificador do mestre que iniciou a transação, e e é uma 4-upla (q, c, d, l) , onde $q, c, d, e l$ representam as quantidades de eventos de cada fase da transação (requisição de uso, sinais de controle, envio de dados e liberação do barramento) e t o número de ordem na sequência das transações. Sejam os vértices $u=(m_u, e_u, t_u)$ e $v=(m_v, e_v, t_v)$ transações do barramento, então o arco $(u, v) \in E$, estabelece ordem de sequência entre as transações u e v , de forma que após a transferência u haverá a transferência v , sendo que $m_u=m_v$ (pertencerão ao mesmo mestre).

A fase seguinte inclui a (5) extensão de um GC para embutir características de barramentos reais e com isso obter uma estimativa de desempenho da comunicação sob aquelas configurações. As características do barramento, que terão seus efeitos adicionados ao grafo de comunicação, são definidas no *Perfil de Barramento*. O perfil de barramento é uma descrição dos parâmetros de um modelo incluindo possíveis larguras do barramento de dados, frequências, suporte a transferências com *pipeline*, latências das transferências (quantidade de ciclos necessários para completar as fases de uma transferência), tipos de transferência (simples, rajada, com e sem preempção), além do mecanismo de arbitragem (e prioridades de mestres, quando for o caso). Caso

o espaço de projeto inclua diferentes modelos de barramento, podem ser utilizados vários perfis de barramento. Em termos gerais, o processo de extensão de um grafo de comunicação envolve quatro etapas:

a. Fixação de largura do barramento de dados: para incorporar uma largura fixa do barramento de dados, o GC deve ser modificado para suportar a largura de barramento definida no perfil do barramento. Para um mestre com tamanho de palavra maior que a largura do barramento de dados, em seu respectivo ramo no grafo, serão adicionados novos vértices para cada vértice já existente, de forma a simular a divisão de uma transferência em transferências com menor tamanho de palavra. Assim, para um mestre com tamanho de palavra de 64 *bits* e um barramento de dados de 8 *bits*, para cada vértice em seu ramo do grafo, seriam criados sete novos vértices, totalizando assim oito transferências de 8 *bits*. Mestres com tamanho de palavra inferior à largura do barramento continuarão com um vértice apenas para representar a transferência de uma palavra, implicando assim em não alteração no grafo. Para esse mestre seria possível ainda agrupar vários vértices (várias transferências) em apenas um único vértice, simulando assim uma transferência com múltiplas palavras. Assim, teríamos, por exemplo, para um mestre com palavras de 8 *bits*, um vértice que representa uma comunicação de um barramento de 64 *bits* poderia agrupar até oito vértices de comunicação deste mestre simulando, assim, oito transferências de 8 *bits*.

b. Inclusão de política de arbitragem: quando há mais de um mestre no barramento, um grafo de comunicação terá um ramo para cada mestre, representando as respectivas sequências de transferências. Mencionou-se anteriormente que a disposição dos ramos pode representar transferências em paralelo e, como o barramento é um recurso compartilhado, deve-se considerar o uso de um mecanismo de arbitragem para escalonar o uso do barramento entre os mestres. Para tanto é necessário definir inicialmente a política de arbitragem a ser adotada e distribuir as prioridades de uso do barramento a cada um dos mestres. Em seguida, o grafo deverá convergir seus ramos de forma que, ao fim, haja apenas um único ramo. O ramo final conterá todos os vértices dispostos em ordem seguindo a política de arbitragem e prioridades definidas anteriormente. Conseqüentemente, o ramo final incluirá, em sua sequência de transferências, as contenções provenientes da arbitragem, assim como ocorre em um modelo de barramento real. Formalmente, o processo de convergência dos ramos de um grafo de comunicação é descrito a seguir:

Dado $G=(V,E)$, um grafo de comunicação de uma dada plataforma T , e M seja o conjunto de mestres de barramento da plataforma T . M será dado por $M=\{M_k\}$, para $k=1,2,\dots,n$, e M_k representa o conjunto de transferências de um mestre k , tal que $M_k = \{v_{k1}, v_{k2}, \dots, v_{km}\}$ e $v_{kt} \in V$, para $t=1, 2, \dots, m$ e t representa a ordem de execução das transações de cada mestre. O processo de convergência será dado por:

i) Cria-se um novo grafo de comunicação $G'=(V', E')$, com os conjuntos V' e E' vazios.

ii) Define-se um conjunto $P=(p_1, p_2, \dots, p_n)$, onde p_1, p_2, \dots, p_n representam as

prioridades de uso do barramento dos mestres M_1, M_2, \dots, M_n , respectivamente.

iii) Cria-se um conjunto M' que conterá os conjuntos M_1, M_2, \dots, M_n , dispostos ordenadamente de acordo com as prioridades definidas por P .

iv) Remove-se o próximo vértice $v_{jz} \in M_p$, onde $M_j \in M'$ e M_j é o conjunto de vértices do mestre de maior prioridade, e adiciona-se v_{jz} a V' . Caso não existam vértices em M_p , este conjunto será removido de M' e M_j passará a ser o próximo elemento de M' .

v) Em seguida, remove-se o próximo vértice $v_{iu} \in M_p$, onde a prioridade p_i de M_i será a segunda maior entre as prioridades de P , e adiciona-se v_{iu} a V' , desde a aresta $(v_{jz}, v_{iu}) \in E'$. Caso não existam mais vértices em M_p , este conjunto será removido de M' e o próximo elemento de M' passará a ser o M_i .

vi) Repetem-se os passos (iv) e (v) até que não restem conjuntos a serem removidos de M' .

Outro aspecto importante considerado neste trabalho, relacionado à arbitragem, é o suporte a transferências com bloqueio (sem preempção) e transferências sem bloqueio (com preempção). Quando um mestre utiliza transferência sem bloqueio, ele pode ser desalocado do barramento, durante uma transferência, para que um mestre de maior prioridade possa utilizar o barramento. Ao receber novamente a concessão de uso do barramento, o mestre deverá reiniciar a transferência interrompida.

c. Agrupamento de transferências: ao convergir os ramos de um grafo de comunicação, os vértices referentes às transferências dos mestres do barramento farão parte de um único ramo, o qual representará toda a sequência de transferências no barramento. Neste ramo final é possível que haja sequências de vértices representando transferências seguidas de um mesmo mestre. De acordo com (LAHIRI; RAGHUNATHAN; DEY, 2001), uma das formas de aumentar o desempenho do processo de estimação, utilizando GC, se dá através do agrupamento dos vértices, reduzindo-se assim o tamanho do grafo. Neste trabalho, o processo de agrupamento será aplicado em conjuntos de vértices que seguem determinados padrões, dados por: 1) um conjunto de transferências realizadas sequencialmente e 2) transferências pertencentes a um mesmo mestre. Se a transferência for do tipo rajada, o agrupamento acontece em duas fases. Na primeira, será dado o agrupamento das transferências que compõem uma rajada. Por exemplo, uma transferência de rajada composta por quatro transferências simples, terá agrupamento de quatro vértices em um único. Este último representará uma rajada completa. Na segunda fase, realiza-se o agrupamento dos vértices que representam sequências de rajadas completas. Os agrupamentos de transferências simples e a primeira etapa do agrupamento de rajadas são bastante semelhantes e podem ser dadas da seguinte forma:

Dado um grafo de comunicação $G=(V,E)$ e $v_1, v_2, v_3, \dots, v_n$ as transações tais que $v_1, v_2, \dots, v_n \in V$, $(v_1, v_2), (v_2, v_3), (v_3, \dots), (\dots, v_n) \in E$ e $v_k=(m, e, t)$, para $k=1, 2, 3, \dots, n$, onde n é o tamanho de uma sequência de vértices analisada, m representa o mestre que iniciou a transferência, e define o conjunto de eventos da transferência e t o número de ordem na sequência de transações, o processo de agrupamento será dado por:

i) Inicialmente, deve-se verificar se os valores de m em v_k , para $k=1,2,3,\dots,n$, são iguais, ou seja, se as transações da sequência analisada pertencem ao mesmo mestre.

ii) Caso pertençam ao mesmo mestre, deve-se verificar, em seguida, se a sequência de eventos de comunicação dado por e dos vértices v_k , para $k=1,2,3,\dots,n$, contém as mesmas quantidades de eventos q , c , d , e l , ou seja, se as transações são do mesmo tipo.

iii) Caso o tipo de transferência configurada para o mestre seja do tipo simples, será adicionado um novo campo T ao vértice v_1 . O novo campo T receberá valor n , que é a quantidade de vértices da sequência analisada e que contabilizará a quantidade de transferências que o vértice v_1 representará. Os demais vértices v_2, v_3, \dots, v_n serão descartados.

iv) Caso o tipo de transferência configurada para o mestre seja do tipo rajada, na primeira etapa do agrupamento será adicionado um novo campo T ao vértice v_1 . Esse novo campo receberá valor referente ao tamanho da rajada, que é predeterminado no perfil do barramento. Os campos d e c , de e , em v_1 , serão também modificados para incluir a quantidade de eventos de endereçamento e envio de dados para as transferências internas da rajada. Os demais vértices da sequência, v_2, v_3, \dots, v_n , serão descartados.

Já a segunda etapa do processo de agrupamento para transferências em rajada pode ser formalizado de acordo com a definição a seguir:

Dado $G'=(V',E')$ um grafo de comunicação e $v_1', v_2', v_3' \dots, v_n'$ transações do tipo rajada tais que $v_1', v_2', \dots, v_n' \in V'$, (v_1', v_2') , (v_2', v_3') , (v_3', \dots) , $(\dots, v_n') \in E'$ e $v_k'=(m, e, t, T)$, para $k=1, 2, 3, \dots, n$, onde n é tamanho de uma sequência de vértices analisada, m representa o mestre que iniciou a transferência, e consiste no conjunto de eventos da transferência, t representa o número de ordem na sequência de transações e T define o tamanho da rajada, a segunda etapa do processo de agrupamento será dada por:

i) Verifica-se se os valores dos campos m dos vértices v_k' , para $k=1,2,3,\dots,n$, são iguais, ou seja, se as transações da sequência analisada pertencem ao mesmo mestre.

ii) Caso pertençam ao mesmo mestre, deve-se verificar, em seguida, se a sequência de eventos de comunicação contida no evento e dos vértices v_k' , para $k=1,2,3,\dots,n$, contém as mesmas quantidades de eventos q , c , d e l , ou seja, se as transações são do mesmo tipo, e se são do tipo rajada, ou seja, se os campos $c>1$ e $d>1$, de e , bem como campo $T>1$.

iii) Caso a sequência de transferências seja do tipo rajada, será atribuído ao campo T do vértice v_1' valor referente ao número de transferências por rajada que irão compor o agrupamento. Os demais vértices v_2', v_3', \dots, v_n' serão descartados.

d. Estimação de latências: o cálculo das latências das transferências é dado através da contagem de ciclos necessários para conclusão das transferências nos

vértices. Nesta contagem, são considerados: frequência de operação do barramento e número de ciclos necessários para cada fase de uma transação de barramento.

O processo de estimação é iniciado com o cálculo do tempo necessário para a ocorrência de um ciclo de relógio do barramento. Isto pode ser realizado a partir da frequência de operação do barramento, obtida no perfil de barramento, e é dado pela equação a seguir:

$$P = \frac{1}{F} \quad (1)$$

onde P é o tempo necessário para a ocorrência de cada ciclo de barramento (período) e F é a frequência de operação informada no perfil de barramento.

Com essa informação, e as quantidades de ciclos necessários para completar as fases de uma transação, calcula-se o tempo necessário para que as transferências representadas pelos vértices possam ser executadas. O cálculo para um vértice é dado pela equação:

$$T_{(\text{vértice})} = (q+c+d+l) \cdot T \cdot P \quad (2)$$

onde q , c , d e l são as quantidades de ciclos para que os eventos de transferência de barramento possam ser executados, T é o total de transferências representadas pelo vértice e P é o período entre ciclos.

Em grafos de comunicação com mais de um vértice, mesmo após realizado o agrupamento de transferências, a estimativa de desempenho da aplicação é realizada através do cálculo das latências de cada vértice. Ao fim, contabiliza-se o total, o qual representará o tempo total da comunicação da aplicação.

Para grafos com vértices representando transferências por rajada, do tipo simples, no cálculo do tempo podemos considerar o suporte a *pipeline*, caso este parâmetro de configuração esteja presente no perfil do barramento. Em modo de *pipeline*, as fases de configurações da transação (controle) e envio de dados são executadas em paralelo.

Para cálculo do tempo de uma transferência em rajada, utiliza-se a equação a seguir, que inclui as fases livres e paralelas:

$$T_{(\text{vértice})} = ((q+c+d+l) + (n-1) \cdot c) \cdot T \cdot P \quad (3)$$

onde q , c , d e l são as quantidades de ciclos para que os eventos de transferência de barramento possam ser executados, n é o tamanho da rajada, T é o número total de transferências representadas pelo vértice e P é o período entre ciclos. O termo

$(q+c+d+l)$ contabiliza as fases não paralelas, e $(n-1).c$ as fases de dados e controle em paralelo. Para o cálculo dos ciclos alinhados, a equação considera que a quantidade de ciclos necessária para a fase de controle (c) é igual à quantidade de ciclos da fase de dados (d), desde que são executadas em paralelo dentro de um *pipeline*.

Assim como no trabalho de (LAHIRI; RAGHUNATHAN; DEY, 2001), onde é possível estimar os tempos da comunicação da aplicação para cada configuração de barramento do espaço de projeto, a abordagem proposta neste artigo obtém também maior desempenho em relação às técnicas de simulação, contudo, por incluir as fases internas inerentes às transações de barramentos, busca atingir uma maior precisão nas estimativas.

4 | RESULTADOS EXPERIMENTAIS

Para avaliação da abordagem proposta, foram escolhidas duas aplicações paralelas que geram bastante tráfego de comunicação no barramento: uma de ordenação de um conjunto de números por *radix* (CORMEN et al., 2009) e outra de multiplicação de matrizes de números inteiros. A primeira aplicação, cujo código fonte utilizado faz parte do pacote de *benchmarks SPLASH* (WOO et al., 1995) – conjunto de aplicações multiprocessadas utilizadas para estudo das seguintes propriedades: balanceamento de carga computacional, taxas de computação e requisitos de tráfego na comunicação; além de questões relacionadas à localidade espacial e como estas propriedades podem ser escaláveis com o tamanho dos problemas e a quantidade de processadores –, consiste de dois processos, onde o primeiro aloca, em uma memória compartilhada, uma estrutura de dados, composta por um conjunto de números inteiros escolhidos aleatoriamente e algumas *flags* de controle. Em seguida, os inteiros na estrutura de dados serão ordenados pelos dois processos paralelamente. A segunda aplicação possui quatro processos, sendo que o primeiro aloca em memória compartilhada duas matrizes de números inteiros, escolhidos de forma aleatória, bem como respectivos *flags* de controle, e, acompanhado dos demais processos, seleciona determinadas linhas e colunas das matrizes para serem multiplicadas. Para sincronizar o acesso às estruturas de dados em memória compartilhada, foi utilizado o algoritmo de *Lamport* (também conhecido como *Lamport's Bakery Algorithm*) (CORMEN et al., 2009).

Para analisar a eficiência e precisão da abordagem proposta para exploração de espaços de projeto de comunicação, foram utilizadas plataformas multiprocessadoras baseadas no processador *MIPS* (descritos em *ArchC*): uma plataforma com dois processadores para a aplicação *radix* e uma plataforma com quatro processadores para a aplicação de multiplicação de matrizes. Os demais componentes das plataformas consistiram de uma memória compartilhada, para armazenar programas

e dados das aplicações, bem como dados compartilhados, e um modelo em alto nível, com precisão de ciclo, do barramento *ARM AMBA AHB* (ARM AMBA, 1999). Os parâmetros de configuração utilizados no barramento para definir o espaço de projeto, são apresentados na tabela a seguir.

Parâmetro de Configuração	Ordenação por <i>Radix</i>	Multiplicação de Matrizes
Largura do barramento de dados	8, 16 e 32 <i>bits</i>	
Mecanismo de arbitragem	Prioridade Fixa	Prioridade Fixa, sendo que o primeiro mestre sempre terá maior prioridade em relação aos demais mestres.
Frequência	100, 166 e 200 (MHz)	
Tipo de transferência	Simples com preempção, Simples sem preempção.	

Tabela 1: Perfil do barramento AMBA AHB utilizado nas aplicações do estudo de caso.

Combinando os parâmetros apresentados na Tabela 1, foi possível construir espaços de projeto totalizando 72 e 486 configurações distintas de barramento para a aplicação de ordenação por *radix* e multiplicação de matrizes, respectivamente. A diferença do tamanho dos espaços de projeto se dá pela quantidade de mestres em cada plataforma, sendo necessário combinar prioridades para dois mestres para a aplicação de ordenação por *radix* e para três mestres para a de multiplicação de matrizes.

Para as simulações, utilizou-se um microcomputador com processador Intel Core 2 Duo de 2,4 GHz com 4 GB de RAM. Os tempos de simulação do espaço de projeto para as duas aplicações são mostrados na tabela a seguir.

Métrica	Ordenação por <i>Radix</i>	Multiplicação de Matrizes
Tempo médio de simulação para cada configuração	32 segundos*	45 segundos*
Tempo total de simulação de espaço de projeto	36 minutos e 10 segundos*	6 horas e 31 minutos*

Tabela 2. Resultados de desempenho para a simulação dos espaços de projetos das aplicações do estudo de caso.

* Os tempos de simulação para cada configuração do espaço de projeto incluem: tempo para reconfiguração dos parâmetros do barramento, compilação e execução do simulador.

Para a geração dos grafos de comunicação, substituiu-se o modelo de alto nível do barramento ARM AMBA das plataformas de simulação pelo barramento com modelo de comunicação genérico proposto neste trabalho. A Tabela 3 apresenta os resultados das simulações para cada aplicação, integrando informações de desempenho e dos perfis de comunicação gerados.

Métrica	Ordenação por <i>Radix</i>	Multiplicação de Matrizes
Transferências de leitura	441.346	3.114.601

Transferências de escrita	182.422		504.980			
Total de transferências	623.768		3.619.581			
Transferências por mestre	Mestre		Mestre			
	1	2	1	2	3	4
	311.866	311.902	96.411	837.370	1.089.817	1.595.983
Tamanho do perfil de comunicação em disco	5,1 MB		62 MB			
Tempo médio de simulação para extração do perfil de comunicação	9 segundos		25 segundos			

Tabela 3. Resultados das simulações e respectivos perfis de comunicação para as aplicações do estudo de caso.

Da análise da Tabela 3, observa-se que a aplicação de multiplicação de matrizes inclui um maior número de transferências, implicando assim em um maior tempo de simulação e um maior espaço em disco para o perfil de comunicação.

Após o grafo de comunicação ser construído, foram selecionadas as configurações do perfil de barramento, apresentado na Tabela 1, para serem incorporadas ao grafo. A Tabela 4 apresenta os resultados de desempenho para estimação dos espaços de projeto com a abordagem de grafos de comunicação proposta.

Métrica	Ordenação por <i>Radix</i>	Multiplicação de Matrizes
Tempo médio para geração de grafo de comunicação	0,9 segundos	14,7 segundos
Tempo médio para estimação de desempenho para uma configuração de barramento	0,6 segundos	6,7 segundos
Ganho médio de desempenho sobre a simulação de uma configuração de barramento	98,2%	85,1%
Tempo médio para estimação do desempenho para o espaço de projeto	41 segundos	54 minutos
Ganho médio de desempenho sobre a simulação do espaço de projeto.	98,2%	86,2%

Tabela 4. Desempenhos para estimação de espaços de projeto por grafos de comunicação para as aplicações do estudo de caso.

Observa-se na Tabela 4 que os tempos para estimação de cada configuração são muito menores que os tempos obtidos por simulação do sistema. Consequentemente, os desempenhos para estimação dos espaços de projeto também serão maiores, com ganhos de 98,2% e 86,2% para as aplicações de ordenação por *radix* e multiplicação de matrizes, respectivamente.

Já a Tabela 5 apresenta os erros máximos, mínimos e médios entre as estimativas obtidas pela abordagem de grafos de comunicação e simulação completa do sistema

para os espaços de projeto para as duas aplicações do estudo de caso. De acordo com os resultados expostos na Tabela 5, percebe-se que as estimativas de desempenho obtidas pela abordagem de grafos de comunicação são bastante próximas às obtidas por simulação do sistema, com erro máximo, entre as duas aplicações, de 3,3% para a multiplicação de matrizes. Os gráficos de erro acumulado mostrados na Figura 2, reforçam a precisão das estimativas obtidas com a abordagem proposta, onde o erro acumulado de todo o espaço de projeto da aplicação de ordenação por *radix* fica próximo de zero (total de quase $5,0e-06$) e da aplicação de multiplicação de matrizes fica em torno de 3%.

Métrica	Ordenação por <i>Radix</i>	Multiplicação de Matrizes
Erro máximo	4,2e-08 %	3,3 %
Erro mínimo	2,8e-08 %	3,1e-07 %
Erro médio	3,5e-08 %	0,6 %

Tabela 5. Erros médio, máximo e mínimo das estimativas de desempenho obtidas por grafos de comunicação para as aplicações do estudo de caso.

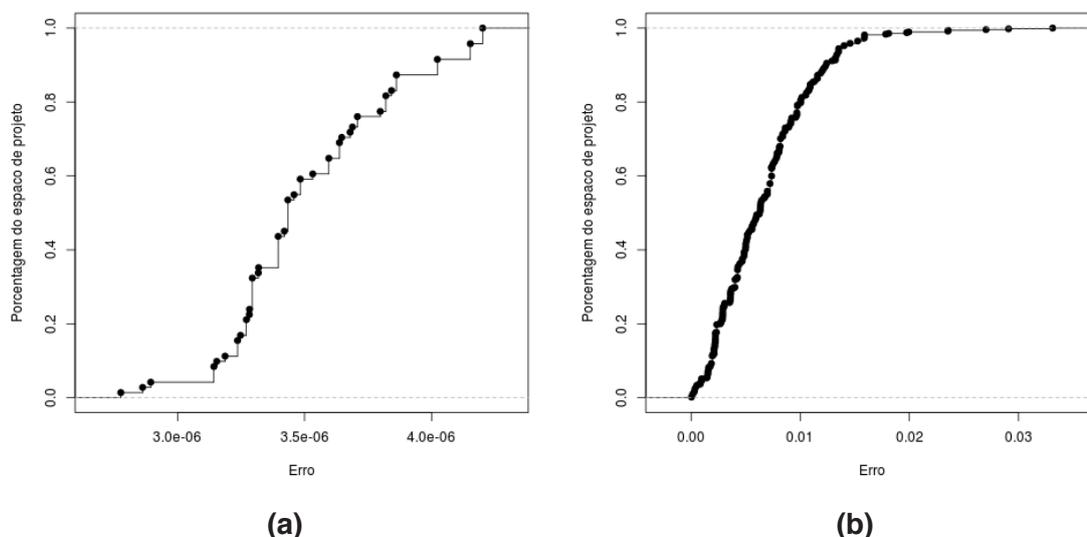


Figura 2. Gráficos de erros acumulados para as aplicações de (a) Ordenação por *Radix* e (b) Multiplicação de Matrizes.

5 | CONCLUSÕES

Projetar a estrutura de comunicação de plataformas multiprocessadoras é uma tarefa difícil e que pode consumir muito tempo e esforço. Quando se considera o uso de barramentos compartilhados, muitos parâmetros devem ser considerados para que se possa atender aos requisitos de comunicação dos componentes da plataforma.

Este artigo apresentou uma nova abordagem que alia uma única simulação do sistema com uma abordagem de estimativas de desempenho de comunicação por

grafos de comunicação para exploração de espaços de projeto de configurações de barramento.

Por considerar apenas uma simulação de sistema e os eventos internos às transações de barramento, a abordagem aqui proposta apresentou, nos experimentos, um ganho máximo de 98,2% no desempenho da tarefa de exploração e um erro máximo de 3,3 % nas estimativas de desempenho do sistema, sendo assim bastante otimistas em relação à abordagem de simulação completa de sistema.

REFERÊNCIAS

ARCHC. Architecture Description Language. Disponível em: <<http://www.archc.org/>>. Acesso em: 08 de ago. 2018.

ARM AMBA. **AMBA Specification rev. 2.0**, 1999. Disponível em: <<http://soc.eecs.yuntech.edu.tw/Course/SoC/doc/amba.pdf>>. Acesso em: 08 de ago. 2018.

BAKLANOV, M. R.; ADELMANN, C.; ZHAO, L.; DE GENDT, S. **Advanced interconnects: Materials, processing, and reliability**. In: ECS Journal of Solid State Science and Technology, v. 4, n. 1, p. Y1-Y4, 2015.

CHO, Y.-S. ; CHOI, E.-J.; CHO, K.-R. **Modeling and analysis of the system bus latency on the SoC platform**. In: Proceedings of the International Workshop on System- Level Interconnect Prediction, 2006. p. 67–74.

CORMEN, T.H., LEISERSON, C.E. RIVEST, R.L.; STEIN, C. **Introduction to Algorithms, 3rd. Ed.**”, The Mit Press, 2009.

GASTEIER, M.; GLESNER, M. **Bus-based communication synthesis on system level**. In: ACM Trans. Design Automation Electronic Systems, 1999. p. 1–11.

IEEE Standards Association. **Standard SystemC Language Reference Manual**. IEEE Std. 1666-2011, 2011.

JERRAYA, A. A.; WOLF, W. **Multiprocessor systems-on-chips**”, Morgan Kaufmann, 2004.

JOO, Y.-P.; KIM, S.; HA, S. **On-chip communication architecture exploration for processor-pool-based MPSoC**. In: Proc. of Design, Automation and Test in Europe, 2009.

KIM, S.; IM, C.; HA, S. **Schedule-aware Performance Estimation of Communication Architecture for Efficient Design Space Exploration**. In: Proc. of the Intl. Conf. on Hardware/Software Codesign and System Synthesis, 2003. p. 195–200.

LAHIRI, K.; RAGHUNATHAN, A. **Power Analysis of System-level On-chip Communication Architectures**. In: Proc. of the Conference on Hardware/Software Codesign and System Synthesis, 2004. p. 236–241.

LAHIRI, K., RAGHUNATHAN, A., DEY, S. **System-Level Performance Analysis for Designing On-Chip Communication Architectures**. In: IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 20, no.6, 2001. p.768-783.

LUKASIEWYCZ, M.; STREUBÜHR, M.; GLASS, M.; HAUBELT, C.; TEICH, J. **Combined System Synthesis and Communication Architecture Exploration for MPSoCs**. In: Proceedings of Design,

Automation and Test in Europe, 2009. p. 472-477.

PASRICHA, S.; DUTT, N.; BOZORGZADEH, E.; BEN-ROMDHANE, M. **Floorplan-aware Bus Architecture Synthesis**. CECS Technical Report 04-27, 2004.

PASRICHA, S.; DUTT, N.; BEN-ROMDHANE, M. **Using TLM for Exploring Bus-based SoC Communication Architectures**. In: Int. Conference on Application-specific Systems, Architectures and Processors, 2005.

NUZZO, P.; SANGIOVANNI-VINCENTELLI, A.; BRESOLIN, D.; GERETTI, L.; VILLA, T. **A platform-based design methodology with contracts and related tools for the design of cyber-physical systems**. In: Proceedings of the IEEE, v. 103, n. 11, p. 2104-2132, 2015.

PAUL, J.; OECHSLEIN, B.; ERHARDT, C.; SCHEDEL, J.; KRÖHNERT, M.; LOHMANN, D.; STECHELE, W.; ASFOUR, T.; SCHRÖDER-PREIKSCHAT, W. **Self-adaptive corner detection on MPSoC through resource-aware programming**. In: Journal of Systems Architecture, v. 61, n. 10, p. 520-530, 2015.

SHIN, C.; KIM, Y.-T.; CHUNG, E.-Y.; CHOI, K.-M.; KONG, J.-T.; EO, S.-K. **“Fast Exploration of Parameterized Bus Architecture for Communication-Centric SoC design.”**, In: Proc. of the conference on Design, automation and test in Europe, 2004.

WILD, T.; HEKERDOF, A.; OHLENDORF, R. **Performance evaluation for system on-chip architectures using trace-based transaction level simulation**. In: Proceedings of the conference on Design, automation and test in Europe, 2006. p. 248–253.

WOO, S. C.; OHARA, M.; TORRIE, E.; SINGH, J. P.; GUPTA, A. **The SPLASH-2 Programs: Characterization and Methodological Considerations**. In: Proc. of the 22nd International Symposium on Computer Architecture, 1995. p. 24-36.

SOBRE O ORGANIZADOR

Ernane Rosa Martins - Doutorado em andamento em Ciência da Informação com ênfase em Sistemas, Tecnologias e Gestão da Informação, na Universidade Fernando Pessoa, em Porto/Portugal. Mestre em Engenharia de Produção e Sistemas pela PUC-Goiás, possui Pós-Graduação em Tecnologia em Gestão da Informação pela Anhanguera, Graduação em Ciência da Computação pela Anhanguera e Graduação em Sistemas de Informação pela Uni Evangélica. Atualmente é Professor de Informática do Instituto Federal de Educação, Ciência e Tecnologia de Goiás - IFG (Câmpus Luziânia), ministrando disciplinas nas áreas de Engenharia de Software, Desenvolvimento de Sistemas, Linguagens de Programação, Banco de Dados e Gestão em Tecnologia da Informação. Pesquisador do Núcleo de Inovação, Tecnologia e Educação (NITE).

Agência Brasileira do ISBN
ISBN 978-85-7247-046-9

