

: ernane_rosa_martins(organizador)

~# ciência da:computação

/ tecnologias_emergentes

> em_computação

editora científica



: ernane_rosa_martins(organizador)

~# ciência da:computação

/ tecnologias_emergentes

> em_computação

1ª Edição
2020

editora científica

Copyright© 2020 por Editora Científica Digital

Copyright da Edição © 2020 Editora Científica Digital
Copyright do Texto © 2020 Os Autores



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

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

C569 Ciência da computação [recurso eletrônico] : tecnologias emergentes em computação / Organizador Ernane Rosa Martins. – Guarujá, SP: Científica Digital, 2020.

Formato: PDF
Requisitos de sistema: Adobe Acrobat Reader
Modo de acesso: World Wide Web
Inclui bibliografia
ISBN 978-65-87196-55-8
DOI 10.37885/978-65-87196-55-8

1. Computação – Pesquisa – Brasil. 2. Tecnologias emergentes. I. Martins, Ernane Rosa.

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. Permitido o download e compartilhamento desde que os créditos sejam atribuídos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.



editora científica

EDITORA CIENTÍFICA DIGITAL LTDA
Guarujá - São Paulo - Brasil
www.editoracientifica.org - contato@editoracientifica.org

editora científica

CORPO EDITORIAL

Editor Chefe
Reinaldo Cardoso

Editor Executivo
João Batista Quintela

Editor Científico
Prof. Dr. Robson José de Oliveira

Assistentes Editoriais
Elielson Ramos Jr.
Érica Braga Freire
Erick Braga Freire
Bianca Moreira
Sandra Cardoso

Arte e Diagramação
Andrewick França
Bruno Gogolla

Bibliotecário
Maurício Amormino Júnior - CRB6/2422

Jurídico
Dr. Alandelon Cardoso Lima - OAB/SP-307852

editora científica

CONSELHO EDITORIAL

Prof. Dr. Robson José de Oliveira - Universidade Federal do Piauí

Prof. Dr. Carlos Alberto Martins Cordeiro - Universidade Federal do Pará

Prof. Me. Ernane Rosa Martins - Instituto Federal de Educação, Ciência e Tecnologia de Goiás

Prof. Dr. Rossano Sartori Dal Molin - FSG Centro Universitário

Prof. Dr. Carlos Alexandre Oelke - Universidade Federal do Pampa

Prof. Me. Domingos Bombo Damião - Universidade Agostinho Neto, Angola

Prof. Dr. Edilson Coelho Sampaio - Universidade da Amazônia

Prof. Dr. Elson Ferreira Costa - Universidade do Estado Do Pará

Prof. Me. Reinaldo Eduardo da Silva Sale - Instituto Federal de Educação, Ciência e Tecnologia do Pará

Prof. Me. Patrício Francisco da Silva - Faculdade Pitágoras de Imperatriz

Prof. Me. Hudson Wallença Oliveira e Sousa - Instituto Nordeste de Educação Superior e Pós-Graduação

Profª. Ma. Auristela Correa Castro - Universidade Federal do Oeste do Pará

Profª. Drª. Dalízia Amaral Cruz - Universidade Federal do Pará

Profª. Ma. Susana Martins Jorge - Ferreira - Universidade de Évora, Portugal

Prof. Dr. Fabricio Gomes Gonçalves - Universidade Federal do Espírito Santo

Prof. Me. Erival Gonçalves Prata - Universidade Federal do Pará

Prof. Me. Gevair Campos - Faculdade CNEC Unaí

Prof. Esp. Flávio Aparecido de Almeida - Faculdade Unida de Vitória

Prof. Me. Mauro Vinicius Dutra Girão - Centro Universitário Inta

Prof. Esp. Clóvis Luciano Giacomet - Universidade Federal do Amapá

Profª. Drª. Giovanna Faria de Moraes - Universidade Federal de Uberlândia

Profª. Drª. Jocasta Lerner - Universidade Estadual do Centro-Oeste

Prof. Dr. André Cutrim Carvalho - Universidade Federal do Pará

Prof. Esp. Dennis Soares Leite - Universidade de São Paulo

Profª. Drª. Silvani Verruck - Universidade Federal de Santa Catarina

Prof. Me. Osvaldo Contador Junior - Faculdade de Tecnologia de Jahu

Profª. Drª. Claudia Maria Rinhel Silva - Universidade Paulista

Profª. Drª. Silvana Lima Vieira - Universidade do Estado da Bahia

Profª. Drª. Cristina Berger Fadel - Universidade Estadual de Ponta Grossa

Profª. Ma. Graciete Barros Silva - Universidade Estadual de Roraima

Prof. Dr. Carlos Roberto de Lima - Universidade Federal de Campina Grande

editora científica

CONSELHO EDITORIAL

Prof. Dr. Júlio Ribeiro - Universidade Federal Rural do Rio de Janeiro

Prof. Dr. Wescley Viana Evangelista - Universidade do Estado de Mato Grosso Carlos Alberto Reyes Maldonado

Prof. Dr. Cristiano Souza Marins - Universidade Federal Fluminense

Prof. Me. Marcelo da Fonseca Ferreira da Silva - Escola Superior de Ciências da Santa Casa de Misericórdia de Vitória

Prof. Dr. Daniel Luciano Gevehr - Faculdades Integradas de Taquara

Prof. Me. Silvio Almeida Junior - Universidade de Franca

Profª. Ma. Juliana Campos Pinheiro - Universidade Federal do Rio Grande do Norte

Prof. Dr. Raimundo Nonato F. do Nascimento - Universidade Federal do Piauí

Prof. Dr. Iramirton Figuerêdo Moreira - Universidade federal de Alagoas

Profª. Dra. Maria Cristina Zago - Faculdade de Ciências Administrativas e Contábeis de Atibaia

Profª. Dra. Gracielle Teodora da Costa Pinto Coelho - Centro Universitário de Sete Lagoas

Profª. Ma. Vera Lúcia Ferreira - Centro Universitário de Sete Lagoas

Profª. Ma. Glória Maria de França - Universidade Federal do Rio Grande do Norte

Profª. Dra. Carla da Silva Sousa - Instituto Federal de Educação, Ciência e Tecnologia Baiano

SUMÁRIO

CAPÍTULO 01 10

AN EFFICIENT ALGORITHM FOR CLOSEST STRING PROBLEM

Omar Latorre Vilca; Mário Salvatierra Júnior

DOI: 10.37885/201102014

CAPÍTULO 02 23

AVALIAÇÃO DE SOFTWARE NA EDUCAÇÃO ESPECIAL: UM OLHAR PARA EDUCANDOS COM AUTISMO

Uilliam P. Oliveira; Wesley B. Thereza; Darley D. de Almeida; Elton R. F. de Rezende; Paula L. S. Lima; Áurea V. P. da Silva; André F. Pereira

DOI: 10.37885/201102096

CAPÍTULO 03 44

***BENCHMARK* PARA ANÁLISE COMPORTAMENTAL DO SISTEMA DE MEMÓRIA VIRTUAL DO LINUX**

Rúbens Antônio Rodrigues; Lourenço Alves Pereira Júnior

DOI: 10.37885/201102053

CAPÍTULO 04 62

DESAFIO DE APLICATIVOS: UMA EXPERIÊNCIA NO INSTITUTO FEDERAL DE GOIÁS (IFG) CÂMPUS LUZIÂNIA

Ernane Rosa Martins

DOI: 10.37885/201101964

CAPÍTULO 05 73

ESTUDO DE CASO DE SEMIÓTICA – O CASO DE " OVER THE GARDEN WALL"

Agnaldo Gomes; Hélio Parente

DOI: 10.37885/200700638

CAPÍTULO 06 87

***FRAMEWORK* DE GERENCIAMENTO ÁGIL DE PROJETOS DO GRUPO DE PESQUISA EM ENGENHARIA DE SOFTWARE DO IFPB**

Alexandre Dias Sales de Morais; Carlos Diego Quirino Lima; Helder Jerônimo Leite. Rangel; Heremita Brasileiro Lira; Nadja da Nóbrega Rodrigues; Samyra Lara Ferreira de. Almeida; Tiago Cesário Barbosa

DOI: 10.37885/201001809

CAPÍTULO 07 100

IFLIVROS – SISTEMA DE GERENCIAMENTO DE LIVROS PARA O IFG CÂMPUS LUZIÂNIA

Jorge Tawnan Leles Vilela; Reinoldo Oliveira Sommer; Ernane Rosa Martins

SUMÁRIO

DOI: 10.37885/201101963

CAPÍTULO 08 118

PROPOSTA DE IDENTIFICAÇÃO DOS USUÁRIOS WIRELESS EM CUMPRIMENTO COM A LEI DO MARCO CIVIL: O CASO DA UNIVERSIDADE DO ESTADO DO PARÁ – CAMPUS XX

Ramon Pinheiro da Silva; Mario Afonso Gomes da Silva; Manoel Fernandes Casimiro Damasceno Costa; Carlos Benedito Barreiros Gutierrez

DOI: 10.37885/201101948

CAPÍTULO 09 136

PRÁTICAS INTERDISCIPLINARES APOIADAS POR PROBLEMAS NA FORMAÇÃO DO TÉCNICO EM INFORMÁTICA

Luiz Fernando Delboni Lomba; Olavo José Luiz Junior

DOI: 10.37885/201102023

CAPÍTULO 10 149

SELEÇÃO DE RECURSOS DE INFRAESTRUTURA CONSIDERANDO DIFERENTES OPÇÕES DE IMPLANTAÇÃO

Raphael de Aquino Gomes; Geovany Santa Cruz Victor Rodrigues; Gilberto Vampré Guimarães Lobo; Jonas Almeida dos Santos

DOI: 10.37885/201101967

CAPÍTULO 11 163

SISTEMA DE APOIO À PRÁTICA ASSISTIDA DE PROGRAMAÇÃO POR EXECUÇÃO EM MASSA E ANÁLISE DE PROGRAMAS

Márcia Gonçalves de Oliveira; Matheus de Araújo Nogueira; Elias Silva de Oliveira

DOI: 10.37885/201102098

CAPÍTULO 12 176

SOFTWARES EM METATRANSCRIPTOMA: O EXPONENCIAL IMPACTO DO BIG DATA NA SAÚDE HUMANA

Diego Lisboa Rios; Silvia de Siqueira Costa; Thiago Araújo Andrade; Paula Margarita Salazar Torres; Lucas Roberto da Silva; Pedro Gontijo Carneiro; Cássio Siqueira Souza Cassiano; Welington Ribeiro Aquino Marques; João Batista Matos Júnior; Fabyola Antunes Gonçalves Souza

DOI: 10.37885/201101943

CAPÍTULO 13 191

TECNOLOGIA VESTÍVEL PARA DETECÇÃO DE GLP E MONÓXIDO DE CARBONO DO AMBIENTE

Mateus Emanuel Mamani Araujo; Magnu Windell Araújo Santos; Leticia Teixeira Nascimento; Pedro Dalvi Boina; Jefferson Oliveira Andrade; Karin Satie Komati; Richard Junior Manuel Godinez Tello

DOI: 10.37885/201102092

SUMÁRIO

CAPÍTULO 14 ..... **201**

UMA FERRAMENTA BASEADA EM SEMÂNTICA PARA O APOIO NA NAVEGAÇÃO EM AMBIENTES INTELIGENTES INCLUSIVOS

José de Jesus Pérez-Alcázar; Fábio Nakano; Joedson Santos de Santana; Henrique Nakaema Simões; Felipe Morine Magami; André Luis Meneses Silva

DOI: 10.37885/201102089

CAPÍTULO 15 ..... **225**

WIKIPOLIS: CIBERCIDADES ENTRE O ATUAL E O VIRTUAL

Reydi Kamimura

DOI: 10.37885/201102151

SOBRE O ORGANIZADOR **237**

ÍNDICE REMISSIVO **238**

01

“An efficient algorithm for closest string problem

- Omar Latorre **Vilca**
  ICOMP/UFAM
- Mário **Salvatierra Júnior**
  ICOMP/UFAM

doi 10.37885/201102014

RESUMO

Neste artigo abordamos o Problema da Cadeia de Caracteres mais Próxima (PCCP) que surge na pesquisa web, teoria da codificação e biologia molecular computacional. Para solucioná-lo deve-se encontrar uma string que minimize a máxima distância de Hamming de um determinado conjunto de strings. PCCP é um problema NP-difícil. Neste trabalho, propomos um algoritmo de tempo linear denominado Algoritmo da Primeira Minimização (APM) que resolve instâncias do PCCP com três strings (3-PCCP). Sua ideia principal é identificar tipos coluna-posição na instância de entrada, o que nos permite decompor em cinco tuplas diferentes, correspondendo à posição de cada string no conjunto de strings para, finalmente, determinar todas as colunas-posições não fixadas por simples avaliação através dos diferentes casos. Uma prova formal de corretude e complexidade computacional do algoritmo proposto serão fornecidas. O algoritmo proposto é comparado com uma formulação de programação inteira para o 3-PCCP. Além disso, experimentos computacionais em tabelas de comparação mostrarão a eficácia do algoritmo proposto.

Palavras-chave: Algoritmo Polinomial, Otimização Combinatória, Problema da Cadeia de Caracteres Mais Próxima.

INTRODUÇÃO

Seja Γ um alfabeto com q símbolos. O Problema da Cadeia de Caracteres mais Próxima (PCCP) ou problema da String Central [7] exige encontrar uma string $x \Gamma^m$ que melhor se aproxime a um determinado conjunto de strings $s^1, s^2, \ldots, s^k \in \Gamma^m$. A aproximação é medida pela distância de Hamming $d(s, s')$, que conta o número de caracteres diferentes em s, e s'. Uma solução ótima do PCCP é uma string x que, entre todas as strings $x \in \Gamma^m$, minimiza a distância máxima $d(x, s^i)$ para i, $1 \leq i \leq k$. 3-PCCP é uma instância de PCCP com 3-sequências. A versão de decisão do PCCP pertence a NP-completo [5].

A ideia central da complexidade parametrizada é estudar o quão difícil um problema permanece após alguns de seus parâmetros serem fixados. Nesse contexto, o PCCP é tratável com parâmetros fixos (FPT), pois existem algoritmos de parâmetros fixos para o PCCP parametrizado por k, número de strings [7], também parametrizado por d, a Distância de Hamming [10]. Além disso, Gramm *et al.* Propôs um algoritmo combinatório direto para encontrar uma string central x para três strings [6], entretanto, este caso é apenas de interesse teórico [6]. Liu et al. apresentou um algoritmo de tempo linear para três strings com alfabeto binário [9]. Boucher *et al.* Propôs um algoritmo linear para encontrar uma string x tal que $max_{i=1}, \ldots, {}_4 d(x, s^i) \leq d$ para quatro strings binárias [3], Amir *et al.* apresentou um algoritmo de tempo quadrático para 5 strings com alfabeto binário [1].

Em biologia computacional, frequentemente surgem problemas relacionados com strings: strings fornecidas são comparadas entre si e sua parte comum é pesquisada. Na criptografia para compressão de dados enfrenta um desafio semelhante [12]. Um dos desafios na pesquisa web é a ocorrência múltipla dos mesmos dados, seja em duplicatas exatas ou com pequenas alterações. Por essas razões, há um maior interesse prático em encontrar métodos para resolvê-lo.

Tabela 1. Uma lista de artigos que usa métodos exatos para resolver o PCCP.

Ano	#Sequências	Alfabeto	Tempo de execução	Referências
2003	k	q	$O(km + kd^{d+1})$	[7]
2008	k	q	$O(km + kd(16\|\Gamma\|)d)$	[10]
2009	3	q	O(m)	[6, 9]
2009	4	2	$O(m)$	[3]
2016	5	2	$O(m^2)$	[1]

Notação. Ao longo deste artigo, estaremos considerando a instância de entrada como uma matriz. Se o conjunto S tem k strings, $s^1, \ldots, s^k$, cada um de comprimento m, vemos S como uma matriz de dimensões k m. Assim, podemos nos referir a colunas e linhas.

Diremos coluna para denotar uma coluna desta matriz e linha para denotar uma linha da matriz. Assim, por exemplo, o elemento é a segunda linha da coluna *j*, é o j-ésimo símbolo de s^2. A distância de uma string *s* de *S* é $max_{s} \in {}_{S}d(s', s)$. Dada uma string *s*, denotamos com $\sigma_j(s)$ o caractere na j-ésima posição de *s*. Referimo-nos a duas colunas idênticas como tendo o mesmo tipo de coluna, também conhecido como tupla. Denotamos com $T(S)$ os tipos coluna de *S*. Para um tipo coluna $t \in T(S)$ denotamos com $\#t$ o número de tipos coluna *t* em *S*. Um conjunto de coluna posições *I* é denotado como $I = \{p_1, ..., p_m\} \subseteq \{1, ..., m\}$ onde $p \in I$ é uma coluna posição específica. Dada uma string *s*, denotamos com *sP* uma substring de *s* restrita a um conjunto de índices *P*.

O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta nossa contribuição, a Seção 3 apresenta uma formulação em programação linear inteira para o 3-PCCP; também apresenta uma análise teórica do Algoritmo da primeira minimização APM. Os resultados experimentais são relatados na Seção 4. Uma discussão em cima dos resultados é dado na Seção 5. Finalmente, a Seção 6 conclui o artigo.

OBJETIVO

Encontrar soluções ótimas para uma instância de entrada do PCCP com três sequências. Com este objetivo, este trabalho descreve a prova de corretude para um algoritmo de tempo linear para resolver o 3-PCCP com alfabeto arbitrário, ele sempre calcula o valor ótimo da solução.

MÉTODOS

Uma formulação em ILP para o 3-PCCP (Gramm *et al.*, [6]) resolveu a versão de decisão deste ILP diretamente usando o algoritmo de (Lenstra, [8]) que possui uma explosão exponencial no número de variáveis. Assim, eles não foram capazes de resolver o ILP por mais de quatro strings. Os autores sugeriram uma formulação ILP baseada em tipos coluna. Isso produz tratabilidade de parâmetro fixo para o PCCP em relação ao parâmetro *k* (número de strings).

Defina n_0 a n_4 como o número de tipos coluna v_0 a v_4. No tipo coluna v_0, você claramente sempre escolhe α. Caso contrário, você precisa atribuir $x_{\alpha,1}$ do tipo coluna v_1 a α (e o resto, $x_{\beta,1} = n_1\, x_{\alpha,1}$, para β), $x_{\alpha,2}$ tipo coluna v_2 para α (e o resto, $x_{\beta,2} = n_2\, x_{\alpha,2}$ para β), $x_{\alpha,3}$ tipo coluna v_3 para α (e o resto, $x_{\beta,3} = n_3\, x_{\alpha,3}$ para β), e $x_{\alpha,4}$, $x_{\beta,4}$, $x_{\gamma,4}$ tipo coluna v_4 para α, β, γ respectivamente (com $x_{\alpha,4} + x_{\beta,4} + x_{\gamma,4} = n_4$). Claramente, dado $x_{\alpha,1}$ a $x_{\gamma,4}$, o processo de "normalizar as colunas, reordenar as strings se necessário, fazer as seleções, des-normalizar

as colunas" leva tempo O(m) (Gramm *et al.*, [7]). Os números $x_{\alpha,1}$ até $x_{\gamma,4}$ são a solução para o seguinte ILP:

$$
\begin{aligned}
\min \quad & d \\
s.t.: \quad & x_{\alpha,1}+x_{\beta,2}+x_{\beta,3}+x_{\beta,4}+x_{\gamma,4} \le d \\
& x_{\beta,1}+x_{\alpha,2}+x_{\beta,3}+x_{\alpha,4}+x_{\gamma,4} \le d \qquad (1) \\
& x_{\beta,1}+x_{\beta,2}+x_{\alpha,3}+x_{\alpha,4}+x_{\beta,4} \le d \\
& \sum_{\sigma \in v_j} x_{\sigma,j} = n_j \quad j = 1,\ldots,4 \qquad (2) \\
& x_{\sigma,j} \in \{0,1,\ldots,n_j\} \qquad (3) \\
& d \in \mathbb{Z}_+. \qquad (4)
\end{aligned}
$$

Nesta formulação, para cada coluna posição considere $v_j \in V(S)$ onde $j = 1, ..., 4$ e σ v_j, a variável $x_{\sigma, j}$ é o número de ocorrências que σ tem na string mais próxima nas posições que correspondem ao tipo coluna v_j. Seja $s^i \in S$ onde $i = 1, ..., 3$ seja a instância de entrada e seja σ o caractere da string s^i no tipo coluna posição v_j. As restrições (1) calculam a distância de Hamming de s^i para a string central, ou seja, para cada tipo coluna v_j somamos os caracteres da string central em posições que correspondem a v_j que são diferentes do que o caractere s^i tem nessas posições. As restrições (2) impõem o número de tipos coluna existentes em cada tupla, e (3) faz $x_{\sigma, j}$ ter valores inteiros. Por fim, o objetivo é minimizar o valor d.

Algoritmo de tempo linear para o 3-PCCP. No Algoritmo da Primeira Minimização (APM) (Vilca, [13]), a identificação dos tipos coluna é necessária. O algoritmo se divide em cinco tipos coluna. Finalmente, ele decide um caractere para cada tipo coluna por avaliação simples através dos diferentes casos de acordo com o número de tipos coluna dados na instância de entrada. O APM pode encontrar a solução ótima percorrendo todas as posições das strings de entrada apenas uma vez. Com tal algoritmo obtemos, o valor ótimo $\lceil max_{i, j = 1,2,3} d(s^i, s^j) / 2 \rceil$ e quando o número de todas as tuplas com caracteres diferentes é maior do que as outras, seu valor ótimo é $\lceil (n_1 + n_2 + n_3 + 2n_4) / 3 \rceil$.

Neste algoritmo sem perda da generalidade considere as distâncias entre pares de strings s^1, s^2 e s^3 satisfazem $d(s^1, s^2) \ge d(s^1, s^3)$ $d(s^2, s^3)$, isto é, a string s^1 é a mais distante e s^3 é a mais próxima das outras duas strings. No algoritmo 1, se o número de tuplas "todas os caracteres são diferentes" for zero, temos o caso binário, então *count é* $\lfloor (d(s^{u1}, s^{u3}) - d(s^{u2}, s^{u3})) / 2 \rfloor$. *Caso contrário, se* $k_1 + k_2 > n_4$ *e* $k_2 > n_4$, *então* k_1 *é* 0, k_2 *é* n_4, *e count é* $\lfloor (k_1 - k_2 + n_4) / 2 \rfloor$; *se* $k_1 + k_2 > n_4$ *e* $k \le n_4$, *então* k_1 *é* $n_4 - k_2$, *e count é* $\lfloor (k_1 + k_2 - n_4) / 2 \rfloor$; *finalmente, se* $k_1 + k_2 < n_4$, *então count é zero.*

O algoritmo 1 atribui os caracteres de s^1 *à solução x entre as posições onde (todos os caracteres diferem com* $k_1 > 0$*) ou (*s^1 *emparelha a* s^2 *ou* s^1 *não emparelha a* s^3 *com count* > 0*), e uma vez que count é reduzida a zero, ele fixa os caracteres de* s^3 *na solução x. Ele*

atribui os caracteres de s^2 à solução x entre as posições onde todos os caracteres diferem, com $k_2 > 0$. Finalmente, uma vez que k_1 e k_2 são reduzidos a zero, ele fixa os caracteres do conjunto de strings alternando-se uma de cada vez na solução x entre as posições onde todos os caracteres diferem.

Algorithm 1: APM pseudo-código.

Input: $S = \{s^1, s^2, s^3\}$: *uma instância de 3-strings de tamanho m.*
Output: *x: solução ótima tal que* $d(x,s) \leq d_{opt}\ \forall s \in \mathscr{S}$.

1 **if** $|S| = 3$ **then**
2 *Ordena as distâncias par a par em* s^1, s^2 *e* s^3.
3 *Seja* u_1, u_2 *e* u_3 *satisfazem* $d(s^{u_1}, s^{u_2}) \geq d(s^{u_1}, s^{u_3}) \geq d(s^{u_2}, s^{u_3})$
4 $k_1 \leftarrow d(s^{u_1}, s^{u_2}) - d(s^{u_2}, s^{u_3})$
5 $k_2 \leftarrow d(s^{u_1}, s^{u_2}) - d(s^{u_1}, s^{u_3})$
6 $k_3 \leftarrow 0$; `// itera sobre as 3-strings`
7 *Seja* n_4 *o número de colunas posição quando todos os caracteres diferem*
8 **if** $n_4 = 0$ **then**
9 $count \leftarrow \lfloor (d(s^{u_1}, s^{u_3}) - d(s^{u_2}, s^{u_3}))/2 \rfloor$; `// Caso 1`
10 **else if** $k_1 + k_2 > n_4$ **then**
11 **if** $k_2 > n_4$ **then**
12 $count \leftarrow \lfloor (k_1 - k_2 + n_4)/2 \rfloor$; $k_1 \leftarrow 0$; $k_2 \leftarrow n_4$; `// Caso 3`
13 **else**
14 $count \leftarrow \lfloor (k_1 + k_2 - n_4)/2 \rfloor$; $k_1 \leftarrow n_4 - k_2$; `// Caso 4`
15 **else**
16 $count \leftarrow 0$; `// Caso 2`
17 **for** *i=1* **to** *m* **do**
18 **if** $(s_i^{u_1} \neq s_i^{u_3}$ *and* $s_i^{u_1} \neq s_i^{u_2}$ *and* $s_i^{u_2} \neq s_i^{u_3})$ **then**
19 **if** $(k_1 > 0$ *or* $k_2 > 0)$ **then**
20 **if** $(k_1 > 0)$ **then**
21 $x_i \leftarrow s_i^{u_1}$; $k_1 \leftarrow k_1 - 1$
22 **else if** $(k_2 > 0)$ **then**
23 $x_i \leftarrow s_i^{u_2}$; $k_2 \leftarrow k_2 - 1$
24 **else**
25 $x_i \leftarrow s_i^{u_{k_3+1}}$; $k_3 \leftarrow (k_3 + 1)\%3$
26 **else if** $(s_i^{u_1} = s_i^{u_2}$ *or* $(s_i^{u_1} \neq s_i^{u_3}$ *and* $count > 0))$ **then**
27 $x_i \leftarrow s_i^{u_1}$
28 **if** $(s_i^{u_1} \neq s_i^{u_3}$ *and* $s_i^{u_2} = s_i^{u_3})$ **then**
29 $count \leftarrow count - 1$
30 **else**
31 $x_i \leftarrow s_i^{u_3}$

Tempo de execução. *A condicional if-then das linhas 1-31 é executada quando a instância de entrada tem 3-sequências; este bloco condicional requer 1 passo. As linhas 2 e 3 ordenam as distâncias par a par em s^1, s^2 e s^3, são necessárias 4m iterações para obter as distâncias de Hamming e c_1 passos para ordená-las em ordem não decrescente por suas distâncias de Hamming o qual o determina em tempo constante. As linhas 4 e 5 fazem operações aritméticas sobre as distâncias de Hamming; essas operações levam um tempo*

constante c_2. Na linha 7, a variável n_4 conta o número de vezes que o tipo coluna-posição v_4 (todos diferem) são apresentadas na instância de entrada, leva m passos. A condicional if-then-else das linhas 8-16 faz operações aritméticas, leva um tempo constante c_3. O laço for das linhas 17-31 itera em m passos, portanto, requer 3m iterações. A condicional if-then-else das linhas 18-31 faz operações aritméticas, leva um tempo constante c_4. Portanto, o tempo de execução do APM é comprovado ser $O(8m)$.

***Análise teórica.** Primeiro, introduzimos três lemas e, em seguida, os usamos para provar que APM pode encontrar uma solução ótima para uma instância de 3-strings com alfabeto arbitrário.*

***Lema 1** Seja $S = s^1, s^2, s^3$ uma instância do PCCP com 3-strings e comprimento m. Se x uma string é uma solução ótima da instância do PCCP e d_{opt} é a distância correspondente, então $\lceil (n_1 + n_2 + n_3 + 2n_4) / 3 \rceil \leq d_{opt}$.*

Demostração: Independentemente de o caractere aparecer na coluna posição j na string solução x, ele se diferencia com no mínimo 1 caractere para v_1, v_2, v_3 e com mínimo de 2 caracteres para v_4. A soma desses valores é igual à distância de Hamming entre x e $s^i \in S$; dividindo-o por 3, obtemos a distância de Hamming média.

***Lema 2 (Liu** et al.**, [9])** Seja S uma instância do PCCP. Se x uma string é uma solução ótima do PCCP e d_{opt} é a distância correspondente, então $\lceil d_{opt} \geq \max_{i,j=1}, ..., kd(s^i, s^j) / 2 \rceil$.*

***Lema 3** Seja $S = \{s^1, s^2, s^3\}$ uma instância do PCCP com 3-strings e comprimento m. Os números das tipos colunas satisfazem a restrição $n_1 \geq n_2 \geq n_3$.*

Demostração: De (5)-(9), temos $d(s^1, s^2) = n_1 + n_2 + n_4$, $d(s^1, s^3) = n_1 + n_3 + n_4$, e $d(s^2, s^3) = n_2 + n_3 + n_4$. Suponha, sem perda de generalidade, que $d(s^1, s^2) \geq d(s^1, s^3) \geq d(s^2, s^3)$. Depois de fazer operações aritméticas, temos, $n_2 \geq n_3$ e $n_1 \geq n_2$. Assim, obtemos, $n_1 \geq n_2 \geq n_3$.

***Teorema 1** Dada uma instância do PCCP com 3-strings e um alfabeto arbitrário. O APM sempre encontra uma solução ótima.*

Demostração: A prova é composta por quatro casos, é feita pelo método direto.

v_0	$s_j^1 = s_j^2 = s_j^3$	*todos os caracteres são iguais*	(5)
v_1	$s_j^1 \neq s_j^2 = s_j^3$	s_j^1 *é a minoria*	(6)
v_2	$s_j^2 \neq s_j^1 = s_j^3$	s_j^2 *é a minoria*	(7)
v_3	$s_j^3 \neq s_j^1 = s_j^2$	s_j^3 *é a minoria*	(8)
v_4	$s_j^1 \neq s_j^2$, $s_j^1 \neq s_j^3$, *e* $s_j^2 \neq s_j^3$	*todos os caracteres são diferentes*	(9)

Considere o alinhamento das três strings s^1, s^2, e s^3. Observe que, de (5)-(9), temos, no geral, para qualquer instância de entrada, existem cinco tipos coluna diferentes. De acordo com o Lema 3, obtemos $n_1 \geq n_2 \geq n_3$. Disso segue que:

Caso 1. *Se* $n_4 = 0$*, temos o caso binário [Liu et al., [9] foi probado]. Então no Algoritmo 1, o valor inicial da variável* **count** *é igual a* $r(n_1 - n_2)/2|$*. Note que* $|\Gamma| = 2$ *e então também* $s_j^3 = s_j^1$ *ou* $s_j^3 = s_j^2$ *entre as posições onde* s^1 *não se emparelha com* s^2*. Portanto* $d(s^1, s^2) = n_1 + n_2$*. A solução x do APM é decidido por:*

(1.1) Entre as posições onde s^1 *se emparelham com* s^2*,* $x\,j = s_j^1 = s_j^2$

(1.2) Entre as posições onde s^1 *não se emparelham com* s^2*:*

$d(x, s^1) \le n_1 - r(n_1 - n_2)/2|$ $d(x, s^2) \le n_2 + r(n_1 - n_2)/2|$

$$\begin{aligned} d(x,s^1) &\le n_1 - \lceil (n_1 - n_2)/2 \rceil \\ d(x,s^2) &\le n_2 + \lceil (n_1 - n_2)/2 \rceil \\ d(x,s^3) &\le n_3 + \lceil (n_1 - n_2)/2 \rceil \le n_3 + \lceil (d(s^1,s^2) - 2n_2)/2 \rceil \\ &\le n_3 - n_2 + \lceil d(s^1,s^2)/2 \rceil \le \lceil d(s^1,s^2)/2 \rceil. \end{aligned} \tag{10}$$

Tomados em conjunto, os sub-casos (1.1) e (1.2) dão um limite inferior (Veja Figura 1(a)), isto é,

$$max_{i=1,2,3} d(x,s^i) \le d(s^1,s^2)/2. \tag{11}$$

Caso 2. *Se* $n_1 - n_3 + n_2 - n_3 \le n_4$*. Então no Algoritmo 1, o valor inicial da variável* **count** *é igual a* 0*,* $k_1 = n_1 - n_3$*,* $k_2 = n_2 - n_3$*. Note que* $|\Gamma| = 3$*. A solução x do APM é decidido assim:*

(2.1) Entre as posições onde existem pelo menos dois caracteres diferentes, temos

$$x_j = argmax_{\sigma \in \Gamma} \sum_{i=1}^{3} |s_j^i = \sigma| \tag{12}$$

(2.2) Entre as posições em que todos os caracteres são diferentes:

$$\begin{aligned} d(x,s^1) &\le n_3 + k_1 + k_2 + \lceil 2(n_4 - k_1 - k_2)/3 \rceil \\ d(x,s^2) &\le n_3 + k_1 + k_2 + \lceil 2(n_4 - k_1 - k_2)/3 \rceil \\ d(x,s^3) &\le n_3 + k_1 + k_2 + \lceil 2(n_4 - k_1 - k_2)/3 \rceil \\ &\le n_1 + n_2 - n_3 + \lceil 2(n_4 - n_1 - n_2 + 2n_3)/3 \rceil \\ &\le \lceil (n_1 + n_2 + n_3 + 2n_4)/3 \rceil. \end{aligned} \tag{13}$$

De (2.1) e (2.2), e pelo Lema 1 (Veja Figura 1(b)), obtemos,

$$max_{i=1,2,3} d(x,s^i) \le \lceil (n_1 + n_2 + n_3 + 2n_4)/3 \rceil \le d_{opt}. \tag{14}$$

Caso 3. *Se* $n_1 - n_3 + n_2 - n_3 > n_4$ *e* $n_2 - n_3 > n_4$*. Então no Algoritmo 1, o valor inicial da variável* **count** *é igual a* $\lceil (n_1 - n_3 - (n_2 - n_3 - n_4)) / 2 \rceil$*,* $k_1 = 0$*,* $k_2 = n_4$*. Note que* $|\Gamma| = 3$ *e então ambos* $s_j^3 = s_j^1$ *ou* $s_j^3 = s_j^2$ *entre as posições onde* s^1 *não emparelha* s^2*,* $d(s^1, s^2) = n_1 + n_2 + n_4$*. A solução x do APM é decidido assim:*

(3.1) Entre as posições onde s^1 *emparelham* s^2*,* $x_j = s_j^1 = s_j^2$

(3.2) Entre as posições onde s^1 não-emparelham s^2:

$$\begin{array}{ll} d(x,s^1) & \leq n_1 - \lceil (n_1 - n_3 - (n_2 - n_3 - n_4))/2 \rceil + n_4 \\ d(x,s^2) & \leq n_2 + \lceil (n_1 - n_3 - (n_2 - n_3 - n_4))/2 \rceil \\ d(x,s^3) & \leq n_3 + \lceil (n_1 - n_3 - (n_2 - n_3 - n_4))/2 \rceil + n_4 \\ & \leq n_3 + \lceil (n_1 - n_2 + n_4)/2 \rceil + n_4 \\ & \leq \lceil (n_1 + n_2 + n_4)/2 \rceil \\ \textit{Since } k_2 & = n_2 - n_3 = n_4 \textit{ we get } n_3 = n_2 - n_4. \end{array} \qquad (15)$$

De (3.1) e (3.2), e pelo Lema 2 (Veja a Figura 2(a)) obtemos,

$$max_{i=1,2,3} d(x,s^i) \leq d(s^1,s^2)/2 \leq d_{opt}. \qquad (16)$$

Caso 4. *Se $n_1 - n_3 + n_2 - n_3 > n_4$ e $n_2 - n_3 \leq n_4$. Então no Algoritmo 1, o valor inicial da variável* **count** *é igual a $\lceil (n_1 - n_3 - (n_4 - n_2 + n_3)) / 2 \rceil$, $k_1 = n_4 - n_2 + n_3$, $k_2 = n_2 - n_3$. Note que $|\Gamma| = 3$ e então ambos $s_p^3 = s_j^1$ ou $s_j^3 = s_j^2$ entre as posições onde s^1 não emparelha s^2, $d(s^1, s^2) = n_1 + n_2 + n_4$. A solução x é decidido assim:*

(4.1) Entre as posições onde s^1 emparelham s^2, $x_j = s_j^1 = s_j^2$

(4.2) Entre as posições onde s^1 não-emparelham s^2:

$$\begin{array}{ll} d(x,s^1) & \leq n_1 - \lceil (n_1 - n_3 - (n_4 - n_2 + n_3))/2 \rceil + n_2 - n_3 \\ d(x,s^2) & \leq n_2 + \lceil (n_1 - n_3 - (n_4 - n_2 + n_3))/2 \rceil + n_4 - n_2 + n_3 \\ d(x,s^3) & \leq n_3 + \lceil (n_1 - n_3 - (n_4 - n_2 + n_3))/2 \rceil + n_4 \\ & \leq \lceil (n_1 + n_2 + n_4)/2 \rceil. \end{array} \qquad (17)$$

Ambos (4.1) e (4.2), e pelo Lema 2 (Veja Figura 2(b)), obtemos,

$$max_{i=1,2,3} d(x,s^i) \leq d(s^1,s^2)/2 \leq d_{opt}. \qquad (18)$$

Ao todo os Casos 1, 3, e 4 provam que a solução x do APM é uma ótima e a solução ótima é $\lceil max_{i,l=1,2,3} d(s^i, s^l) / 2 \rceil$. Também, a solução ótima para o Caso 2 é $\lceil (n_1 + n_2 + n_3 + 2n_4) / 3 \rceil$. Assim, o teorema segue.

Exemplo 1 *Seja S uma instância do PCCP com 3-strings e cada string de tamanho* 10*, S é obtido apartir de S onde o conjunto de strings foi ordenado par a par pelas distâncias de Hamming, assim obtemos:*

$$S = \begin{cases} AGTATTGGTG \\ CCCTTTGAGA \\ TAGTGGGTCT \end{cases} \qquad S' = \begin{cases} TAGTGGGTCT \\ AGTATTGGTG \\ CCCTTTGAGA \end{cases}$$

$$S' = \left\{ \begin{array}{ccc|ccc|c|ccc} v_4 & v_4 & v_4 & v_2 & v_1 & v_1 & v_0 & v_4 & v_4 & v_4 \\ \hline \mathbf{T} & \mathbf{A} & G & \mathbf{T} & G & G & \mathbf{G} & \mathbf{T} & C & T \\ A & G & \mathbf{T} & A & \mathbf{T} & \mathbf{T} & \mathbf{G} & G & \mathbf{T} & G \\ C & C & C & \mathbf{T} & \mathbf{T} & \mathbf{T} & \mathbf{G} & A & G & \mathbf{A} \\ \hline T & A & T & T & T & T & G & T & T & A \end{array} \right.$$

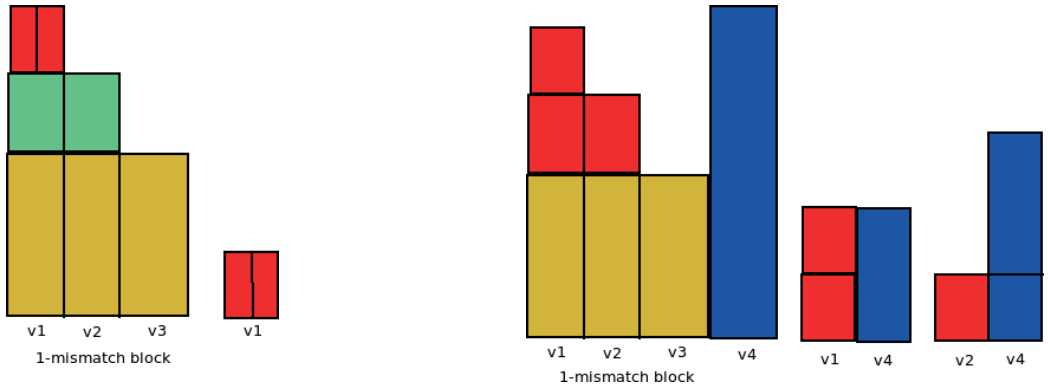
O número de tipo colunas quando "todos os caracteres são diferentes" é $n_4 = 6$*. Os três contadores têm os seguintes valores* $k_1 = n_1\ n_3 = 2$*,* $k_2 = n_2\ n_3 = 1$*, e count* $= 0$*. Com esses valores, APM identifica o Caso 2, na verdade a solução ótima para S ou S é x = TATTTTGTTA com distância de Hamming* $d(x, s) \leq 5, \forall s \in S$*. Note que o valor da solução ótima para o Caso 2 é* $\lceil (n_1 + n_2 + n_3 + 2n_4) / 3 \rceil = \lceil (2 + 1 + 0 + 2(6))/3 \rceil = 5$*.*

RESULTADOS

Apresentamos agora os experimentos computacionais realizados com os algoritmos descritos na Seção 3. Inicialmente, descrevemos o conjunto de instâncias usadas nos testes. Em seguida, na Seção 5 os resultados obtidos pela formulação IP e o algoritmo proposto são discutidos.

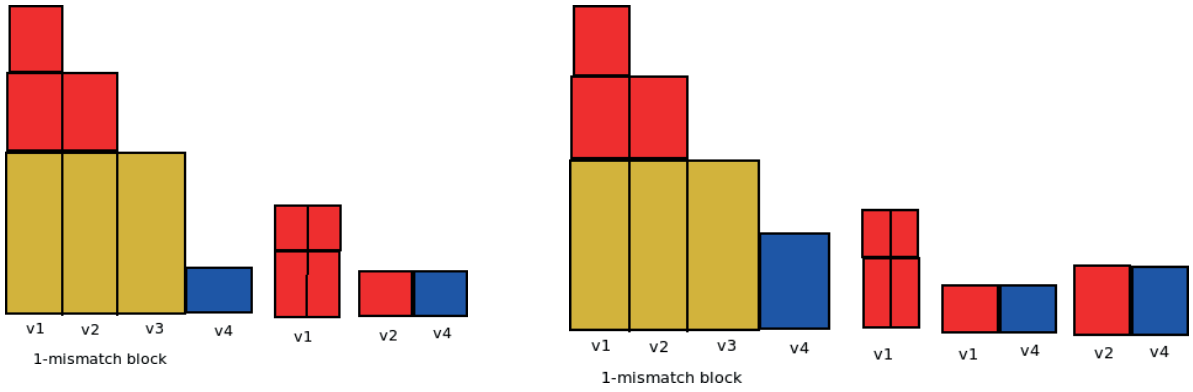
Instâncias e ambiente de teste. *Alguns experimentos computacionais foram realizados envolvendo instâncias do 3-PCCP, o algoritmo usado para geração de números aleatórios é uma implementação do Well equidistributed long-period linear, um gerador de números aleatórios (Panneton et al., [11]); este algoritmo foi preferido devido ao seu longo período de* $2^{44497} - 1$*. Essas instâncias consideram alfabetos com dois, quatro e vinte caracteres, que obtiveram soluções ótimas usando o APM e a formulação IP.*

Figura 1. Ilustração da aplicação do Caso 1, para uma instância do PCCP com 3-strings e um alfabeto binário, existem três tipos coluna a serem considerados v_1, v_2, e v_3. O algoritmo fixa blocos 1-mismatch, em cores laranja dourada e verde, com seu valor maioritário; em seguida, atribui o tipo coluna v_1 (em cor vermelha), metade delas com seu valor maioritário e a outra com seu valor minoritário. Ilustração da aplicação do Caso 2, existem quatro tipos coluna a serem considerados v_1, v_2, v_3, e v_4.O algoritmo fixa blocos 1-mismatch, em cor laranja dourada, com seu valor maioritário, depois disso, ele atribui os tipos coluna v_1 e v_4 com seu valor maioritário e o caractere de s^1; ele fixa os tipos colunas v_2 e v_4 com seu valor maioritário e pelo caractere de s^2; finalmente, ele atribui o resto dos tipos coluna v_4 em uma ordem todos contra todos.



(a) Caso 1 **(b)** Caso 2

Figura 2. Ilustração da aplicação do Caso 3. Existem quatro tipos coluna a serem considerados v_1, v_2, v_3, e v_4. O algoritmo fixa blocos 1-mismatch, em cor laranja dourada, com seu valor maioritário; depois disso, ele atribui às tipo colunas v_2 e v_4 com seu valor maioritário e os caracteres de s^2; ele fixa o tipo coluna v_1 (em cor vermelha), metade delas com seu valor maioritário e a outra com seu valor minoritário. Ilustração da aplicação do Caso 4, Existem quatro tipos coluna a serem considerados v_1, v_2, v_3, e v_4. O algoritmo fixa blocos 1-mismatch, em cor laranja dourada, com seu valor maioritário, depois disso, ele atribui as tipos colunas v_1 e v_4 com seu valor maioritário e pelo caractere de s^1; ele fixa os tipos colunas v_2 e v_4 com seu valor maioritário e pelo caractere de s^2; finalmente ele fixa o tipo coluna v_1 (em cor vermelha), metade delas com seu valor maioritário e a outra com seu valor minoritário.



(a) Caso 3 **(b)** Caso 4

Tabela 2.Número de instâncias para os quais se obtiveram valores ótimos.

	Valor	IP	APM
Instâncias binárias (140)	4421.00	0.106	0.008
Instâncias ADN (140)	7690.73	0.362	0.009
Instâncias proteínas (140)	10850.14	1.047	0.010

Tabela 3. Resumo dos resultados para alfabetos com dois, quatro e vinte caracteres.

Instância	2 Caracteres					4 Caracteres					20 Caracteres				
m	Min	Média	Max	IP	APM	Min	Média	Max	IP	APM	Min	Média	Max	IP	APM
1000	250	258.9	264	0.009	0.001	429	441.1	451	0.065	0.003	611	616.9	622	0.051	0.002
2500	628	637.2	653	0.017	0.002	1080	1095.0	1113	0.048	0.002	1537	1545.1	1555	0.125	0.002
5000	1252	1266.0	1285	0.031	0.003	2175	2190.3	2213	0.094	0.003	3071	3088.1	3100	0.253	0.003
7500	1876	1900.7	1927	0.044	0.004	3252	3277.9	3308	0.143	0.004	4608	4629.1	4645	0.389	0.005
10000	2456	2523.6	2575	0.059	0.005	4324	4382.9	4436	0.191	0.006	6169	6183.1	6193	0.536	0.006
12500	3126	3150.8	3204	0.075	0.006	5429	5465.1	5509	0.252	0.007	7698	7716.5	7746	0.674	0.007
15000	3740	3780.8	3818	0.089	0.007	6522	6562.9	6597	0.297	0.008	9234	9261.5	9282	0.818	0.008
17500	4371	4406.3	4470	0.104	0.008	7615	7658.6	7719	0.347	0.009	10791	10808.9	10828	0.971	0.009
20000	5006	5038.6	5086	0.119	0.009	8701	8749.1	8794	0.399	0.010	12280	12347.3	12394	1.177	0.011
22500	5642	5660.8	5681	0.134	0.010	9820	9846.5	9893	0.450	0.012	13863	13888.0	13921	1.357	0.012
25000	6239	6281.1	6329	0.149	0.011	10894	10943.9	10979	0.503	0.013	15414	15440.8	15462	1.520	0.013
27500	6879	6917.9	6972	0.165	0.013	12007	12047.0	12100	0.551	0.014	16935	16981.4	17006	1.694	0.014
30000	7504	7545.2	7586	0.180	0.013	13079	13143.7	13206	0.604	0.015	18495	18535.9	18567	1.860	0.016
50000	12479	12526.1	12595	0.306	0.022	21803	21866.2	21928	1.129	0.024	30819	30859.3	30900	3.227	0.025

Os programas foram desenvolvidos em linguagem C++ e utilizaram o compilador Gnu C++ com otimizações. Todos os testes foram executados em um computador desktop com a seguinte configuração: processador Dell Intel Core I7 com velocidade de 3,4 Ghz e 7,7 GB de RAM em Linux Ubuntu 18.04 LTS funciona em 64 bits. O solver IBM ILOG CPLEX versão 12.7 foi usado para resolver a formulação IP.

DISCUSSÃO

O APM foi executado em um conjunto de 420 instâncias, com 14 instâncias para cada um dos alfabetos. Dez instâncias foram geradas para cada entrada. Na Tabela 3 os resultados representam a média dos valores obtidos.

Os cabeçalhos da Tabela 3 possuem os seguintes significados: a primeira coluna indica a instância testada, indicando o parâmetro (m) tamanho da string; as colunas (2,4 e 20 caracteres) indicam os (Min, Média, Max) valores mínimo, médio e máximo de soluções ótimas para três sequências com alfabeto binário, de ADN e de proteínas. As colunas rotuladas "IP" fornecem o tempo médio de execução (em milissegundos) para a formulação IP. As colunas rotuladas "APM" fornecem o tempo médio de execução (em milissegundos) para o algoritmo proposto.

Na Tabela 2, vemos que em média o APM produz valores de solução dentro de 100% dos valores de solução obtidos por formulação IP para instâncias com alfabetos de tamanhos dois, quatro e vinte, respectivamente. O tempo médio de CPU para instâncias de proteína usando APM: 0,010 (milissegundos), que é menor do que a formulação IP: 1,047 (milissegundos).

CONSIDERAÇÕES FINAIS

Neste trabalho, mostramos um algoritmo de tempo linear, sua prova de corretude e o valor da solução ótima para o PCCP com no máximo três strings para um alfabeto arbitrário. É principalmente uma caracterização do tipo coluna posição. Na literatura, permanece um problema em aberto para encontrar uma string central para cadeias de $n \geq 4$ *com um alfabeto arbitrário. Esse problema não parece fácil mesmo para quatro strings.*

REFERÊNCIAS

1. Amir A., Paryenty H., and Roditty L., Configurations and minority in the string consensus problem. Algorithmica, 74(4), 1267-1292, 2016.
2. Basavaraju M., Panolan F., Rai A., Ramanujan M. S., Saurabh S., On the kernelization complexity of string problems. Theoretical Computer Science, 730, 21–31 (2018).
3. Boucher C., Brown D. G., and Durocher S., On the Structure of Small Motif Recognition Instances. Springer Berlin Heidelberg, Berlin, Heidelberg, (2009) 269-281.
4. Fomin F. V., Lokshtanov D., Saurabh S., Zehavi M., Kernelization: Theory of Parameterized Preprocessing. chapter 1. Cambridge University Press, Great Britain, UK, 2019.
5. Frances M. and Litman A., On covering problems of codes. Theor. Comput. Syst., 30 (1997) 113-119.
6. Gramm J., Niedermeier R., and Rossmanith. P. , Exact solutions for closest string and related problems. In Proceedings of the 12th International Symposium on Algorithms and Computation, ISAAC ’01 (2001) 441-453.
7. Gramm J., Niedermeier R., and Rossmanith P., Fixed-parameter algorithms for closest string and related problems. Algorithmica 37(1) (2003) 25-42.
8. Lenstra Jr. H. W., Integer programming with a fixed number of variables. Mathematics of Operations Research 8(4) (1983) 538-548.
9. Liu X., Liu S., Hao Z., and Mauch H., Exact algorithm and heuristic for the closest string problem. Computers & Operations Research, 38(11) (2011) 1513-1520.
10. Ma B. and Sun X., More Efficient Algorithms for Closest String and Substring Problems.
11. Springer Berlin Heidelberg, Berlin, Heidelberg, (2008) 396-409.
12. Panneton F., L’Ecuyer P., Matsumoto M., Improved long-period generators based on linear recurrences modulo 2, ACM Transactions on Mathematical Software, 32 (1) (2006) 1–16.
13. Storer J. A., Data compression: Methods and theory. chapter 1. Computer Science Press, Inc., New York, NY, USA, 1988.
14. Vilca O. L., Combinatorial Approaches for the Closest String Problem. Doctoral thesis, Federal university of Amazonas, Computing institute. Recovered from https://tede.ufam.edu.br/handle/tede/7449

02

“Avaliação de software na educação especial: um olhar para educandos com autismo

- Uilliam P. **Oliveira**
  UNEMAT
- Wesley B. **Thereza**
  UNEMAT
- Darley D. de **Almeida**
  UNEMAT
- Elton R. F. de **Rezende**
  UNEMAT
- Paula L. S. **Lima**
  UNEMAT
- Áurea V. P. da **Silva**
  UNEMAT
- André F. **Pereira**
  UNEMAT

doi 10.37885/201102096

RESUMO

Este trabalho procurou entender o Transtorno do Espectro Autista (TEA) e suas técnicas de intervenção, verificamos se as abordagens de avaliação de softwares educacionais existentes na literatura permitem avaliar um produto para alunos com Necessidade Educativas Especiais (NEE) e se podem ser usadas pelos profissionais da educação no processo de ensino-aprendizagem. Após um mapeamento sistemático, exploração e análise de 13 abordagens, atentando-se nos critérios pedagógicos, de interface e de software, recomendou-se a adaptação de uma das abordagens, para torná-la mais inclusiva, de fácil acesso e entendimento para os profissionais da educação.

Palavras-chave: Autismo, Software Educacional, Educação Especial, Avaliação de Software.

INTRODUÇÃO

Os Transtornos Invasivos do Desenvolvimento (TID) referem-se a uma categoria do Manual Estatístico e Diagnostico da Associação Americana de Psiquiatria (DSM-V), que descreve condições de neurodesenvolvimento infantil definida por manifestações comportamentais, nela está incluso o Transtorno do Espectro Autista (TEA) (Brasil, 2015; Fuentes, 2014; Gadia et al., 2004), um distúrbio de desenvolvimento complexo com variados graus de severidade, que trazem déficits nas áreas da socialização, comunicação (verbal e não-verbal) e linguagem (podendo existir atraso ou até mesmo ausência da mesma). Além do pobre contato visual, dificuldades em participar de atividades que requerem interação social, bem como padrões singulares e repetitivos de comportamentos e interesses ou atividades específicas (Bosa 2006; Kwee, 2006).

No Autismo muitas técnicas de intervenção e teorias são estudadas no intuito de promover a inclusão dessas crianças, com destaque recentemente no tratamento baseado em tecnologia (Hungate, 2019), as mais eficazes ou eficientes são: a abordagem TEACCH (*Treatment and Education of Autistic and related Communication-handicapped Clhildren*) (Kwee, 2006; Macedo, 2011) é um programa educacional que combina diferentes materiais visuais e estratégias para facilitar a compreensão da criança em relação ao ambiente de trabalho e o que esperar dele, ou seja, adequando o ambiente e atuando no cuidado com a previsibilidade e sem sobrecarga de estímulos externos, bastante usados nas esferas de assistência educacional e clínica; A ABA (*Applied Behaviour Analysis*) (Camargo e Rispoli, 2013; Gomes, 2016a) trata-se de um abordagem científica e intervenção comportamental bastante utilizada na psicologia, seguindo a mesma perspectiva do condicionamento operante apresentadas por Skinner (Macedo, 2011), estuda as variáveis que afetam comportamentos e age no intuito de diminuir atitudes e/ou ensinar novas habilidades, realizando por meio de observações de seus principais conceitos: estímulo antecedente, reposta e consequência; e o PECS (*Picture Exchange Communication System*) (Bosa, 2006; Oliveira et al., 2015) apresenta-se como instrumento estímulo visual e de Comunicação Alternativa e/ou Suplementar (CAS) por meio da troca de estímulos visuais por objetos ou atividades de interesse.

A inclusão escolar, social e digital de crianças com Necessidades Educativas Especiais (NEE) nas escolas está dependendo de recursos da computação que, apesar da sua exorbitante utilidade no dia a dia, atualmente as novas tecnologias possuem potencial para serem utilizadas como recurso pedagógico, favorecendo o nível de qualidade do ensino-aprendizagem (Vieira, 1998). O problema está tanto na adoção das ferramentas pelos profissionais da educação, visto que há - por parte deles – uma insegurança ocasionada pela falta de treinamento para com as novas tecnologias (Buckingham, 2010), quanto na avaliação de tais softwares para serem aplicados em sala de aula (Silva et al., 2016).

No cenário escolar, a utilização das Tecnologias Digitais da Informação e Comunicação (TDICs) como ferramenta pedagógica na educação especial cresceu de forma exponencial nos últimos anos, principalmente a partir de 2012 após a lei nº 12.764 (Brasil, 2012) que visa garantir os direitos legais a pessoas com alguma deficiência (Ferreira et al., 2014), uma vez que as TDICS conduzem a novas propostas de ensino e oferecem maiores possibilidades de inclusão e acessibilidade. Consequentemente, é relevante a questão dos Softwares Educacionais (SEs) serem elaborados seguindo os métodos ou as teorias pela qual o sujeito aprende, como ele se apropria e constrói o conhecimento, para que assim estimulem os alunos a interagir com esses recursos provenientes do avanço tecnológico e do mundo globalizado (Jucá, 2006; Valente, 1998).

O objetivo deste trabalho é verificar as ferramentas de avaliação de SEs, se estas, permitem a adoção de produtos informatizados para alunos dentro do espectro autista e, se são da fácil aplicação e entendimento para profissionais da educação que venham utilizá-la. Vale ressaltar que o foco da pesquisa não está em destacar pontos positivos ou negativos de quaisquer ferramentas ou softwares selecionados e sim: (i) guiar para produção de produtos com melhores qualidade de aprendizagem e acessibilidade; (ii) contribuir para o acesso do educador á ferramentas de avaliação de software para a educação especial.

Este trabalho está dividido em nove seções, incluindo essa. Logo na parte 2 é apresentado referencial teórico. Posteriormente na 3º, a metodologia utilizada. Os trabalhos selecionados estão na parte 4º. Na 5º temos a verificação da facilidade de aplicação das abordagens exploradas. A seção 6º é composta pela exploração dos métodos. Na 7º temos a conclusão da análise dos modelos de avaliação. 8º, apontamos as considerações do estudo e trabalhos futuros. Por último temos as referências.

REFERENCIAL TEÓRICO

Silva et al. (2016) acentuam que os SEs devem estar perfeitamente adequados para o público alvo destinado, tal como seu conteúdo, interface, necessidades, e limitações dos usuários, pois quaisquer problemas em um destes níveis podem comprometer o processo de construção do conhecimento. A partir desse pensamento Jucá (2006) e Vieira (1999) ressaltam que uma das principais características em um SE é seu desenvolvimento fundamentado em uma teoria de aprendizagem pela qual o aprendiz melhor se apropria das informações e constrói seu conhecimento. Analisamos as principais teorias de aprendizagem – Construtivismo e Behaviorismo a fim verificar qual se encaixa melhor no contexto de SEs para educandos com TID, especificamente para alunos com Autismo.

Na psicologia educacional e do desenvolvimento infantil temos Piaget com seus estudos baseado na psicologia genética, ressaltando a aprendizagem como um processo de

reorganização cognitiva que acontece por meio de processos interpessoais, onde o aluno deve descobrir por si mesmo as propriedades dos objetos que interage e de suas próprias ações (Almeida, 2005; Castorina, 2006). Além de deixar claro que as crianças constroem o conhecimento a partir de estruturas cognitivas existentes, o autor ressalta um importante conceito em sua teoria, o conflito sociocognitivo, que por sua vez, na situação em que o indivíduo está diante de uma nova informação que contrapõe o que ele sabia, acontece ali um conflito entre essa informação e o conceito que o indivíduo carregava, havendo um processo de "equilibração" mental – entre assimilação e acomodação - levando o indivíduo ao conhecimento. Logo, o conflito sociocognitivo não permite que o aluno produza automaticamente processos de conhecimento, é necessário que o sujeito deva possuir instrumentos intelectuais para que o conflito funcione (Castorina, 2006; Cunha, 2004; Taile et al., 1993).

Na concepção da teoria sócio-histórica de Vygotsky, a aprendizagem interage com o desenvolvimento, devido a consequência de processos interpessoais e posteriormente intrapessoais, isto é, a interação - com uso dos signos - entre a criança e um adulto permite que a criança incorpore intrinsecamente sistemas semióticos de representação e suas significações (Castorina, 2006; Taile et al., 1993). Quando ele postula que a aprendizagem deve estar coerente com o nível de desenvolvimento da criança, descreve um novo e excepcional conceito chamado de "zona de desenvolvimento proximal" (ZDP) (Vygotsky, 1998), no qual descreve dois níveis de desenvolvimento: o real, engloba aquilo que elas conseguem fazer por si mesmas; e o proximal, aquilo que não sabem fazer sozinhas e necessitam de assistência. Dessa maneira, a relação entre ensino-aprendizagem é bem trabalhada na ZDP, pois:

> *"[...[o aprendizado adequadamente organizado resulta em desenvolvimento mental e põe em movimento vários processos de desenvolvimento que, de outra forma, seriam impossíveis de acontecer" (Vygotsky, 1998. p. 61).*

Mas, qual das duas teorias seria melhor aproveitada com os alunos dentro do espectro autista?

Foram dissecadas duas linhas de desenvolvimento cognitivo diferentes que trabalham: "de dentro para fora", permitindo que o aluno construa por si só o conhecimento, desta forma, levando- o a novos desafios intelectuais e que, a aprendizagem é um processo distinto e dependente do desenvolvimento; e outra que acontece "de fora para dentro", por meio da mediação e uso dos signos culturais, ressaltando que a aprendizagem influencia e impulsiona o desenvolvimento fazendo com que o aluno tenha novas conquistas. Um preza a construção do conhecimento, o outro enfatiza o desenvolvimento cognitivo do aluno.

É notório que tanto Piaget quanto Vygotsky valorizam as relações interpessoais e intrapessoais, pois ambas são importantes para o desenvolvimento humano (Taile et al., 1993), particularmente no Autismo essas relações são essenciais no processo de internalização.

Deste modo, por meio de um referencial teórico construtivista, construcionista, contextualizado e significativo pode conduzir a um processo pedagógico inclusivo (Almeida, 2005), no entanto, para nós, um método de pesquisa sócio-histórica é a mais indicada para trabalhar com esse público alvo, uma vez que estamos interessados em analisar cada estágio de desenvolvimento dessas crianças, o que torna importante a ser estudado os processos e não os objetos (Moraes, 2008; Passerino, 2005).

MÉTODOS

Essa pesquisa segue uma abordagem exploratória com elementos de um mapeamento sistemático, por realizar o levantamento bibliográfico das técnicas e proporcionar uma visão geral sobre as abordagens e sua relação com SEs e seus públicos alvos. Identificamos as abordagens, analisamo-las dentro do contexto da educação especial.

Primeiramente foi feito um mapeamento sistemático nos anais de eventos e revistas cientificas que possuem trilha ou são destinados ao tema de informática na educação: TISE (Conferência Internacional sobre Informática na Educação); SBIE (Simpósio Brasileiro sobre Informática na Educação); RBIE (Revista Brasileira de Informática na Educação) e RENOTE (Revista Novas Tecnologias na Educação), que compreendeu publicações de 2016 a 2019 – este período foi escolhido de acordo com o cronograma do projeto, ampliar a base de dados e período de interesse acarretaria em atrasos em outras fases de desenvolvimento do mesmo. Admitimos artigos resumidos (*short papers*) e completos (*full papers*), utilizando palavras chaves (*strings* de busca) em português: (“avaliação” OR “educação especial” OR “software” OR “*checklist*”) AND (“ ” OR “autismo” OR “autista”), em inglês: (“*evaluation*” OR “*special education*” OR “*educational software*” OR “*checklist*”) AND (“ ” OR “*autism*” OR “*autistic*”), e em espanhol: (“*evaluación*” OR “*educación especial*” OR “*software educativo*” OR “*lista de verificación*”) AND (“*autismo*” OR “*autista*”).

A seleção dos artigos se deu via busca manual, pelo título dos mesmos. Devido ao excesso de trabalhos com análises bibliográficas dos mesmos métodos foi necessário fazer um refinamento, de acordo com esses critérios de exclusão:

a. Estudos de revisão sistemática da literatura;
b. Estudos direcionados a desenvolvedores;
c. Estudos duplicados – quando o mesmo trabalho, dos mesmos autores fora encontrado em diferentes locais pesquisados, foi dada prioridade a publicação mais recente;
d. Estudos que apresentam soluções de intervenção sem uma avaliação;
e. Estudos sobre avaliação de software de outras deficiências intelectuais;

f. Estudos sobre o TEA que fogem da temática da educação.

A partir desse ponto – Tabela 1, iniciamos a abordagem exploratória dos trabalhos selecionados e adaptamos uma tabela comparativa dos mesmos – Tabela 2, a fim de verificar se as abordagens de avaliação abrangem qualidades pedagógicas, de interface e de software em um SE para educandos dentro de espectro.

Tabela 1. Relação de trabalhos encontrados.

Eventos	Trabalhos encontrados	Trabalhos selecionados
TISE	29	03
SBIE	68	05
RBIE	02	0
RENOTE	02	0

A seção a seguir apresenta uma visão geral sobre os trabalhos selecionados, com o propósito de conhecermos como funciona a aplicação de tal abordagem.

TRABALHOS SELECIONADOS

A organização dos trabalhos selecionados segue uma ordem cronológica de publicação.

D'Carlo et al. (2016) em seu trabalho apontou os requisitos de usabilidade de um sistema móvel, apresentou a avaliação Heurística (AH), que consiste em um método empírico que julga a interface de acordo com o conjunto de heurísticas. Após realizado o levantamento bibliográfico e detectado 12 critérios denominados "heurísticas genéricas" (HG) e foi proposto um conjunto de 14 heurísticas específica (HE) visando a potencialização da interface e iteração. Os resultados obtidos permitiram constatar que as HEs foram capazes de concretizar melhor os potenciais problemas de usabilidade de um aplicativo. No quesito de avaliação, o participante deveria indicar o grau de relevância dos critérios de acordo com a seguinte escala: Fortemente Relevante; Relevante; Pouco Relevante e Irrelevante.

Pereira et al. (2016) com o objetivo de definirem um conjunto mínimo de diretrizes para uma avaliação de SE, realizaram uma análise comparativa entre 08 abordagens propostas na literatura, que foram: Método Reeves, Técnica TICESE, Técnica de Mucchielli, Método LORI, Abordagem MAQSEI, Abordagem FASE, Método de SAVI e Abordagem PATESE. Relatou os pontos comuns e específicos de cada abordagem, algumas das mesmas – apesar de terem nomes diferentes – analisavam um mesmo tipo de SE, tal como foi ressaltado a pluralidade e falta de padronização desses instrumentos.

Nunes e Santos (2018) constataram que os métodos analisados são antigos e direcionados a softwares para desktops, fazendo com que alguns critérios não se apliquem

a aprendizagem móvel. A pesquisa enfatizou a carência de um padrão ergonômico para aplicativos educacionais de dispositivos móveis, e considerou em analisar as heurísticas mais comuns nos trabalhos, resultando especificamente em duas propriedades numa avaliação de SEs: 07 aspectos ergonômicos (usabilidade) e 07 requisitos de caráter pedagógico (aprendizagem). Para avaliação foi atribuído conceitos referentes ao grau de satisfação aos requisitos avaliados, em que E significa excelente; B significa bom; R representa regular; e P quer dizer que é péssimo.

Oliveira e Thereza (2018) utilizaram a *checklist* de Silva (2014) para analisar SEs direcionados a educação especial, apesar da proposta ter encaixado bem com o objetivo, poderia ser melhor explorado se houvessem heurísticas mais específicas para o público-alvo estudado. Silva (2014) baseando-se na *checklist* proposta pelo PROINFO (Programa Nacional de Informática na Educação), atualizou e estabeleceu um novo paradigma mais eficiente e detalhado para avaliação de software educacional em diferentes categorias, tais como: exercícios e pratica, tutorial, jogos, software de autoria, modelagem, navegadores, hipertexto e hipermídia, eletrônico e aplicativo. Com heurísticas específicas para características pedagógicas, interface, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

Aguiar et al. (2018) fizeram uma pesquisa bibliográfica acerca das metodologias de ensino, mecânicas dinâmicas e estéticas usadas em jogos digitais para educandos com TEA. O trabalho resultou numa abordagem que visa a avaliação de jogos digitais educativos para indivíduos dentro do espectro, a ferramenta em questão trata-se de um questionário com 07 questionamentos base para avaliação de jogos digitais para este público com NEE, utilizando uma escala de concordância e discordância como forma de avaliação do SE.

Diante da diversidade de abordagens para avaliação de SEs, Junior e Aguiar (2018) assumiram a responsabilidade de unificar os instrumentos de avaliação, considerando que eles eram complementares e eliminando as duplicidades existentes a fim de favorecer um uso homogêneo de termos e de suas respectivas definições, assim foi o processo de concessão do TaCASE (Taxonomia de Critérios para Avaliação de Software Educativo). Concebida a partir da combinação de 14 abordagens - inclusive as analisadas por Pereira et al. (2016) -, considerou a análise de 413 critérios de avaliação de software educativo, resultando em duas classes que agrupam 21 critérios de qualidade de software e 12 critérios de qualidade de uso.

Neves et al. (2018) direcionaram seu estudo na usabilidade de jogos para alunos com deficiências intelectuais e, elaborou um questionário especifico e simples abordando questões do jogo educativo tal como os benefícios trazidos pelo mesmo, a avaliação do SE apresentando se deu por meio de um estudo experimental de observação, na qual realizou os devidos testes de usabilidade, não empregando as *checklist* ou técnicas de avaliação de SEs a serem exploradas neste trabalho.

Almeida et al. (2018) aplicaram os 10 passos do guia GDSM (*Global-Driven Software Measurement*), incorporando os resultados do Quali-EDU para melhorar e elaborar um modelo mais completo, coerente com as necessidades de seleção e avaliação do SE. Para atingir a meta do trabalho foram elencadas entidades – objetos de interesse, atributos, características ou propriedades de tais entidades -, e de acordo com o contexto educacional analisado gerou-se uma *checklist* com métricas de medição singulares. Por meio dos cinco indicadores: (i) Totalmente incoerente; (ii) Parcialmente incoerente; (iii) Indiferente; (iv) Parcialmente coerente e (v) Totalmente coerente, se a maiorias das notas forem entre o item 4 ou 5, significa que a ferramenta deve ser adotada no plano pedagógico.

O método (GDSM) utilizado na abordagem de Almeida et al. (2018) mostrou-se extremamente relevante por ser moldável para a finalidade de qualquer estudo de caso, sendo possível utiliza-la também na educação especial, no entanto, pela a finalidade deste estudo essa abordagem não será explorada, uma vez que: a *checklist* foi específica para a finalidade do estudo de caso, compará-la na presente pesquisa seria despropositado e acarretaria um resultado final indesejado.

Organizamos todas as abordagens encontradas nos trabalhos selecionados - Tabela 2. Cada técnica de avaliação recebeu um identificador (ID) – “A01” Abordagem 01- seguindo a mesma ordem da presente seção.

Mediante análise da Tabela 2 é possível observar quais abordagens contemplam aspectos de qualidade pedagógica, de interface e de software (A03, A05, A06, A09 e A10), ainda podemos averiguar que A07 não marca em nenhum dos critérios por tratar-se de um banco de dados – sistema especialista. A12 possui heurísticas especificas, baseada em seu público alvo, seus questionamentos atuam como heurísticas para avaliação do SE. 07/13 abordagens (A0, A02, A03, A08, A10, A112) declaram a quais tipos de SE avaliam. A09 examina SEs de conteúdo especifico - matemática.

FACILIDADE NA APLICAÇÃO DOS INSTRUMENTOS

Sendo um dos objetivos do presente trabalho: verificar se tais instrumentos são da fácil aplicação e entendimento para profissionais da educação que venham utilizá-la. Nesta seção será explorada - especificamente - a última coluna da Tabela 2 e relatada quais instrumentos de avaliação de objetos de aprendizagem são passiveis de aplicação por educadores, seja ela *checklist*, questionário, metodologia, exploração cognitiva, entrevistas, inspeções formais, ferramenta web ou outro tipo de sistema que avalie tal produto educacional.

Para melhor analise, agrupamos as abordagens de acordo com o modelo que a mesma é: *checklist*, metodologia, questionário e sistema.

No grupo das *checklists* temos: 05/13 abordagens, abaixo segue algumas observações marcantes:

I. A02 apoia-se em duas abordagens, sendo uma com 14 critérios pedagógicos e outra com 10 critérios relacionados à interface, isto faz com que ela misture elementos de *checklist*, avaliação heurística e ensaio de interação. Apesar de possuir descrição dos parâmetros, a escala de avaliação é demasiada difícil de estabelecer um ponto, devido a mesma não ser numerada;
II. A03 integra aspectos pedagógicos, de interface e usabilidade, permite a classificação do tipo de software, avalia desde a qualidade da informação do manual/documentação que acompanha o produto á adequabilidade do mesmo no contexto pedagógico ou situação especifica. Apresenta formulário de inspeção e sua avaliação é obtida por meio da ponderação de pontos dos critérios e sub-criterios. O autor sugere a leitura da técnica, um prévio conhecimento e exploração do software para aplicar a avaliação;
III. A05 apresenta um manual de utilização bem fácil, como forma de medida apresenta dispõe de uma escala de avaliação numerada de 01 (um) á 05 (cinco), há formulários e explicação de cada critério tal como a justificativa dos valores atribuídos na escala avaliativa;
IV. A11 traz heurísticas pertinentes e de fácil compreensão, mas não há explicação dos mesmos, nem formulário extra para impressão.

Das *checklists* apresentadas, todas pedem ser utilizadas por profissionais da educação. Apenas A03 e A05 dispõem de formulário para inspeção e manual de utilização desagregados – ambos podem ser impressos. Encontramos formulários das outras *checklist* – A02, A09 e A11 – apenas nos trabalhos originais dos respectivos autores, este fato retrata o quão limitado e "escondido" estão tais ferramentais, dificultando o trabalho da escola e do educador.

Como metodologia temos 02/13 abordagens – A04 e A06. Ambas possuem determinado número de etapas bem descritas a serem seguidas para avaliação de um SE. A04 e A06 necessitam de uma equipe com especialistas e IHC e educadores, um não pode prescindir o outro para a garantia da qualidade de aspectos pedagógicos, de interface e usabilidade. A06 pode ser utilizada para softwares em desenvolvimento ou em softwares prontos, podendo ser personalizada ou completada a depender do estudo de caso, da formação dos avaliadores e/ou aspectos do produto.

A08 e A10 enquadram-se como questionário. A proposta de A08 é um modelo de avaliação de jogos educacionais, ainda não foi testada nem validada, seus critérios são claros e dispensam descrições dos mesmos. A10 apresenta uma tabela com aspectos

ergonômicos e pedagógicos, há explicação de cada parâmetro para facilitar a compreensão e aplicação por profissionais da educação. Tais questionários também estão confinados nos trabalhos originais.

Concluindo esta seção, as abordagens: A01 contribuição para área de IHC, pois atualiza e apresenta heurísticas para SEs para aplicativos móveis; A07 trata-se de uma ferramenta web - sistema especialista - para criação de instrumentos de avaliação de SE, retornando um feedback quando preenchida pelo usuário juntamente com banco de dados e alternativas com pontuações; e A13 foi validada por especialistas Engenharia de software e de Usabilidade, como trabalho futuro, comprometeu-se ao desenvolvimento de um recurso digital para apresentar a taxonomia e amplia-la com critérios pedagógicos.

Com o propósito de verificar se tais abordagens estão disponíveis para educadores e se são de fácil aplicação e entendimento pelos mesmos, foi notório que a facilidade de aplicação está em apenas 02 (duas) das 13 (treze) abordagens exploradas, somente A03 e A05 possuem formulário e/ou manual de utilização desprendido dos estudos que a originaram, estão também disponíveis na internet para serem impressos, os demais métodos de avaliação não compartilham deste quesito. O recomendado seria que essas ferramentas avaliativas estivessem retiradas/separadas dos trabalhos originais dos autores e disponibilizadas numa outra versão juntamente com seu manual e/ou descrição dos critérios/heurísticas, em consequência, a facilidade e possibilidade de o educador adotar um SE no processo de ensino-aprendizagem seria exponencialmente maior.

EXPLORAÇÃO DAS ABORDAGENS NO CONTEXTO DO TEA

Como já fora explanado nos estudos de Junior e Aguiar (2018), Nunes e Santos (2018) e Pereira et al. (2016), a diversidade e convergências entre os critérios dos diferentes modelos de avaliação, focaremos em analisar as abordagens descritas nos trabalhos relacionados com nosso estudo sobre o TEA e referencial teórico apresentado em seções anteriores, dessa forma, explorando-as, espera-se ver qual paradigma pode ser utilizado ou adaptado para avaliar SEs direcionados a alunos com TEA.

Nesta ocasião iremos explorar as abordagens apresentadas na seção de trabalhos selecionados e organizados na Tabela 02 - adaptado de Pereira et al. (2016) -, salientamos pontos primordiais que devem ser levados em consideração em um SE para educandos com Autismo, que são: base pedagógica, utilização de mecanismos/abordagens utilizadas nas esferas clínicas e/ou educacionais; adaptabilidade, acessibilidade, portabilidade e avaliação.

Tabela 2. Quadro comparativo das abordagens.

Id	Abordagem	Critérios				Tipos de SEs avaliados	Conteúdos	Avaliador
		Qualidades pedagógicas	Qualidade de interface	Qualidade de software	Específicos			
A01	D' Carlo	-	-	14	-	Aplicativos móveis	-	Especialistas em IHC
A02	Reeves	14	10	-	-	Informatizados	-	Professores
A03	TICESE	01	07	09	-	Informatizados	-	Especialistas em IHC e professores
A04	Muchielli	-	-	10	-	-	-	Especialistas em IHC e professores
A05	LORI	01	04	04	-	-	-	Professores
A06	MAQSEI	01	04	04	-	-	-	Especialistas em IHC e professores
A07	FASE	-	-	-	-	-	-	Especialistas em IHC e professores
A08	Savi	03	24	-	-	Jogos	-	Professores
A09	PATESE	07	08	04	-	-	Matemática	Professores
A10	Nunes	07	03	04	-	Aplicativos móveis	-	Desenvolvedores e professores
A11	Silva	03-08*	-	20-32*	-	Informatizados	-	Desenvolvedores e professores
A12	Aguiar	-	-	-	07	Jogos	-	Desenvolvedores e professores
A13	TaCASE	-	12	21	-	-	-	Especialistas em IHC

*Os números de critérios podem variar de acordo com o tipo de software analisado.

Base Pedagógica

Nos estudos de Vieira (1999) e Silva et al (2016), ambos destacam a importância de uma base pedagógica correspondente às necessidades do seu público, a vista disso, dentre às 13 ferramentas de avaliação de software, apenas 05 questionam base pedagógica na qual se apoia o SE (Reeves, Muchielli, TICESE MAQSEI e Aguiar), contudo, 11 (Reeves, Muchielli, TICESE, Maqsei, LORI, Savi, Silva, D'Ccarlo, TaCASE, Aguiar e Nunes) apresentam parâmetros para averiguar as características pedagógicas presentes nos SEs.

Utilização de Mecanismos

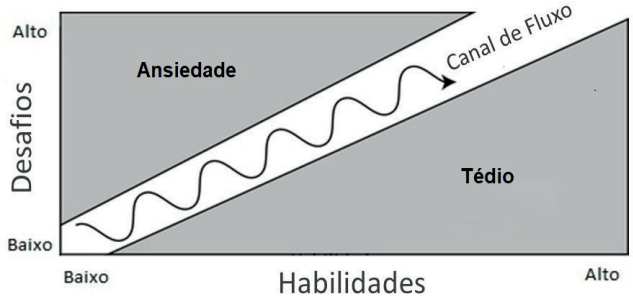
A utilização de mecanismos refere-se à atenção dos métodos de avaliação de SEs - independente se estes forem direcionados a um público alvo ou não - estão verificando meios ou técnicas para potencializar o processo de ensino-aprendizagem.

Considerando as abordagens exploradas apenas a PATESE e Savi explicitam pra qual público alvo é aplicada. Exceto a *checklist* de Silva - que utiliza métricas específicas

para cada tipo de SE -, as outras abordagens como Reeves e TICESE utilizam as mesmas métricas para avaliar distintos tipos de SEs, o que implica em aspectos importantes de um determinado tipo de software passará despercebido e na possibilidade de haver critérios irrelevantes/não aplicáveis, tornando-a uma abordagem genérica.

Experiências, diferentes estímulos, novas sensações é o que o psicólogo Mark Csikszentmihalyi (1999) desenvolve em sua obra, na qual define o termo “fluxo” como uma fonte de energia psíquica que concentra a atenção e motiva a ação. Por exemplo, quando estamos realizando uma atividade, que, os desafios são extremamente altos, ficamos frustrados, posteriormente preocupados e mais tarde angustiados, e quando há desafios baixos em relação á nossas habilidades, ficamos relaxados e seguidamente entediados. A Figura 1 ilustra este conceito sobre a onda de fluxo, que, ocorre justamente no limite entre tédio e a ansiedade, dando origem a uma sensação de ação sem esforço, permitindo ao indivíduo um constante entretenimento, técnica bastante utilizada na indústria de jogos eletrônicos.

Figura 1. Conceito da onda de fluxo.



Como já fora explanado nos estudos de Junior e Aguiar (2018), Nunes e Santos (2018) e Pereira et al. (2016), a diversidade e convergências entre os critérios dos diferentes modelos de avaliação, focaremos em analisar as abordagens descritas nos trabalhos relacionados com nosso estudo sobre o TEA e referencial teórico apresentado em seções anteriores, dessa forma, explorando-as, espera-se ver qual paradigma pode ser utilizado ou adaptado para avaliar SEs direcionados a alunos com TEA.

O recomendado em um SE direcionado a crianças dentro do espectro, é que, além de permitir um aprendizado totalmente adaptativo, utilize estratégias baseada nos modelos de intervenções já citadas anteriormente, como por exemplo organização, antecipação e previsibilidade do TEACCH, o estímulo visual do PECS, a dinamicidade de um aprendizado agradável da ABA, etc. Como resultado, permitiremos uma melhor experiência de aprendizagem e desenvolvimento para esses educandos, favorecendo sua inclusão digital, social e escolar.

Acessibilidade

Esta subseção está intimamente relacionada as Tecnologias Assistivas para crianças com algum tipo de deficiência e para qualquer pessoa (mediador) venha a utilizar o software sem dificuldades. Neste ponto, o critério mais partilhando entre as abordagens é o de acessibilidade - 10/13 frisam essa questão -, visto que a interface permite uma melhor comunicabilidade dos produtores do sistema aos usuários (Neves et al., 2016), desta forma, permitindo que o software seja utilizado e entendido por qualquer pessoa, seja pelo o usuário em busca do conhecimento ou um programador, professor/mediador que venha – respectivamente - a elaborar ou adota-la como ferramenta de ensino.

Adaptabilidade

Um SE adaptável ao usuário permite o mesmo trabalhar em seu próprio ritmo e nível de habilidade, maximalizando seus pontos fortes e minimizando os pontos de dificuldades, de forma que ajude na progressão da aprendizagem (Zakari et al., 2014). Este é um requisito demasiadamente importante para trabalhar com crianças com TEA, visto que, além déficits na interação social, comportamental e de fala, eles possuem dificuldades em reter a atenção em certos estímulos devido a reações opostas vindas de estímulos sensoriais (tais como soma altos, supersensitividade táctil, fascínio por determinados estímulos visuais, dentre outros.) (Almeida, 2005; Bosa, 2006; Gadia et al., 2004; Passerino, 2005). As abordagens Reeves, Muchielli, TICESE, LORI, Silva e Tacase permitem avaliar a carga informacional apresentada nas atividades, carga de estímulos e grau de conhecimento do indivíduo.

Portabilidade

Educandos com TEA possuem limitada habilidade de generalização e carecem de estímulos constantes (Gomes, 2016b; Kwee, 2006), um SE que permita que a criança continue aprendendo o conteúdo – seja conteúdo acadêmico ou atividades da vida diária - de maneira variada e lúdica fora do contexto escolar e/ou clínica pode favorecer uma melhor seu desenvolvimento. A viabilidade de um SE multiplataforma pode conduzir o educando a melhores formas entender o mundo que o cerca. Adequem as TDICs de acordo com as limitações e potencialidades do público alvo em questão, como por exemplo, a utilização da TV digital como instrumento pedagógico para alfabetização de crianças com deficiência intelectual (Neves et al., 2016).

Dentre as 13 abordagens estudadas, apenas 03 - Reeves, Silva e Aguiar - apresentam heurísticas aplicáveis a SEs em dispositivos móveis. Exclusivamente 02 - D'Carlo e

Nunes - apresentam critérios específicos para a plataforma mobile, visto que a maioria das abordagens se concentram em SEs para desktop.

Avaliação

É evidente que as TDICs trazem um novo paradigma educacional e, com novas formas de ensinar, vem também novas formas de avaliar. Essas novas ferramentas devem permitir o registro e extração de dados das atividades executadas nele, a fim de análises posteriormente pelos educadores e outros profissionais que acompanham o desenvolvimento do educando. Das abordagens exploradas, apenas 03 - TICESE, MAQSEI e Silva - possuem métricas a fim de verificação da avaliação do aprendizado.

Em nosso contexto – alunos com TEA- vale questionar como ocorre a avaliação no período de escolarização e apresentar novas concepções.

Culturalmente temos um modelo de avaliação estática (Moraes, 2008) na qual, analisa o produto ao invés do processo, isto é, uma pratica avaliativa que se concentra no que a criança já sabe, retrospectiva, que considera dados quantitativos e não permite a exploração mais ampla das funções psicológicas superiores. Com embasamento teórico nas obras de Vygotsky, Moraes (2008) defende a adoção de uma avaliação dinâmica – prospectiva -, que permite acompanhar melhor o desenvolvimento das funções psicológicas implícitas nas atividades propostas pelo educador, dessa forma, é possível explorar além das funções já desenvolvidas e as que estão em processo de desenvolvimento, compreendendo que a forma de avaliar o processo de ensino-aprendizagem como forma de acompanhar/orientar o desenvolvimento do educando pode conduzir a novas experiencias (Valentim e Oliveira, 2013; Oliveira et al., 2015).

CONCLUSÃO DA EXPLORAÇÃO

Sendo um dos objetivos do presente trabalho: verificar as ferramentas de avaliação de SEs, se estas, permitem a adoção de produtos informatizados para alunos dentro do espectro autista e, se são da fácil aplicação e entendimento para profissionais da educação que venham utilizá-la como instrumento pedagógico. Após análise da Figura 2, que expõe uma visão panorâmica das nossas pontuações na seção 6, sobre os requisitos essências para alunos com Autismo: base pedagógica, utilização de mecanismos/abordagens, adaptabilidade, acessibilidade, portabilidade e avaliação, a conclusão da exploração foi que:

Figura 2. Panorama das abordagens em relação aos requisitos. Fonte: Autores.

Id	Base pedagógica	Utilização de mecanismos	Adaptabilidade	Acessibilidade	Portabilidade	Avaliação
A01	✓			✓		
A02	✓		✓	✓	✓	
A03	✓		✓	✓		✓
A04	✓		✓	✓		
A05	✓		✓	✓		
A06	✓			✓		✓
A07						
A08	✓	✓				
A09		✓		✓		
A10	✓			✓		
A11	✓	✓	✓	✓	✓	✓
A12	✓					
A13	✓		✓	✓	✓	

A abordagem A11 mesmo não sendo direcionada para educandos com necessidades educativas especiais, apresentou-se bem tanto no trabalho de Oliveira e Thereza (2018) quanto no presente estudo, visto que:

I. Seus critérios são claros e de fácil entendimento, podendo ser aplicada por professores e por pessoas mais leigas, não necessitando de conhecimento específico em IHC;
II. Abrange tanto aspectos pedagógicos, quanto de qualidade de interface e de software – de forma implícita;
III. Possui métricas específicas para cada categoria de software, valorizando as propriedades distintas de cada um.

Portanto, para deixá-la mais inclusiva, recomendamos a criação ou adaptação de heurísticas com aspectos dos métodos de intervenção, da teoria sócio histórica e pontos essenciais apresentados nas seções 1, 2 e 6 – respectivamente - a fim de valorizar os processos de desenvolvimento da criança com TEA. Outra observação está relacionada a tabulação de resultados, a mesma possui questões objetivas (sim ou não), o que pode dificultar uma conclusão de avaliação, por isso, recomendamos ao adapta-la, colocar questões ponderadas - atribuição de valores para tais -, deste modo, ela será melhor explorada e possibilitará uma análise fidedigna da qualidade do SE.

Outras abordagens como A07 não marcou quaisquer pontos na tabela da Figura 2 por tratar- se de uma ferramenta web direcionada a especialistas e avaliadores de características de softwares. No tópico da acessibilidade, foi notório que A08 e A12 não frisam tal requisito por serem técnicas específicas focadas nas experiências dos usuários em jogos digitais educacionais.

CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

É incontestável a pluralidade das abordagens de avaliação de softwares educacionais. Todas as analisadas possuem pontos semelhantes e distintos, algumas são simples de aplicar e outras necessitam de conhecimentos específicos, no entanto, cabe ao usuário verificar qual se adequa melhor a sua situação. Diante disso realizamos a presente pesquisa a fim de verificar se havia abordagens que possibilitasse a análise de softwares educacionais para educandos com Autismo e qual seria o modelo de avaliação mais indicado para classificar os SEs a serem utilizados junto a alunos com o Transtorno do Espectro Autista.

É extremamente importante a teoria de aprendizagem estar alinhada com o processo de aprendizagem do indivíduo. Foi possível observar em nosso estudo bibliográfico sobre o TEA e as práticas de intervenção que, a abordagem ABA tem proporcionado melhores resultados (Foxx, 2008), isso se dá pelos princípios Behaviorista (comportamentalista) utilizados pela mesma, trabalhando a memorização - com o condicionamento operante. Desta forma, foi intencional abordarmos a teoria Behaviorista na seção 2, para enfatizar a importância dessa base teórica quando forem avaliar um SE e em novas soluções tecnológicas que venham ser desenvolvidas para educandos dentro do espectro.

No tocante a facilidade de aplicação, este ponto está intimamente atrelado a: (i) manuais de utilização disponíveis; (ii) descrição dos critérios e, (iii) se não encontramos, não dá para aplica-la. Analisamos e concluímos em seções anteriores que as abordagens de avaliação de SEs estão escondidas, trabalhos de extrema qualidade se perdendo nos bancos de dados de pesquisa acadêmica por não levarem adiante a divulgação da mesma. Como sugestão para desenvolvedores, não é necessário criar um sistema que unifique todas as abordagens – cada uma se adequa a um contexto/estudo de caso -, mas um sistema que seja possível encontra-las facilmente com um suporte a esclarecer dúvidas dos utilizadores.

Para trabalhos futuros, progredindo em nosso projeto de elaborarmos uma intervenção baseada em tecnologia multiplataforma e gratuita para educandos dentro do espectro, iremos: (i) adaptar a *checklist* de Silva (2014); (ii) elaborar o manual de instrução da mesma; (iii) valida-la empiricamente e quantitativamente com a ajuda das famílias, profissionais que trabalham com esse público e dos mais recentes softwares educacionais disponíveis para download. e (iv) desenvolver um web site, para a divulgação da *checklist* com detalhes sobre cada heurística nela presente e versão imprimível, para que não fique limitada apenas a pesquisadores, visto que a pesquisa em si se preocupou com a facilitação da adoção de tais ferramentas digitais pelos educadores.

REFERÊNCIAS

1. Aguiar, E. C. et al.. (2018). **Avaliando Jogos Digitais Educativos para Indiv´ıduos Portadores do Transtorno do Espectro Autista**. Simpósio Brasileiro de Informática na Educação, [S. l.]. Disponível em: http://www.br-ie.org/pub/index.php/sbie/article/view/8164. Acesso em: 4 mar. 2019

2. Almeida, A. L. de. (2005**). Interação de Crianças Autistas com o Mundo Digital: Uma Travessia de Emoção e Prazer**. 260 p. Tese (Doutorado) - Engenharia de Produção, Programa de PósGraduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis- SC.

3. Almeida, A. et al. (2018**). Indicadores para Avaliação de Software Educacional com base no guia GDSM (Goal Driven Software Measurement**). Simpósio Brasileiro de Informática na Educação, [S. l.]. Disponível em: http://www.br-ie.org/pub/index.php/sbie/article/view/7950. Acesso em: 10 mar. 2019.

4. Bosa, C. A. (2006). **Autismo – Intervenções psicoeducacionais**. 7 p. Artigo (Instituto de Psicologia) - Universidade Federal do Rio Grande do Sul, Porto Alegre/RS.

5. Brasil. (2012). **LEI Nº 12.764, de 27 de dezembro de 2012.** [S. l.]. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12764.htm. Acesso em: 11 fev. 2019.

6. Brasil, (2015). Ministério da Saúde. **Linha de cuidado para a atenção às pessoas com transtornos do espectro do autismo e suas famílias na Rede de Atenção Psicossocial do Sistema Único de Saúde**. Brasília-DF: Editora MS. 156 p. ISBN 978-85-334-2108-0. Disponível em: https://pesquisa.bvsalud.org/bvsms/resource/pt/mis-36971. Acesso em: 1 mar. 2020.

7. Buckingham, D. (2010)**. Cultura Digital, Educação Midiática e o Lugar da Escolarização**. Educação & Realidade, Porto Alegre.Disponível em: https://seer.ufrgs.br/educacaoerealidade/article/view/13077/10270. Acesso em: 3 mar. 2019.

8. Camargo, S. P. H.; rispoli, M. (2013). **Análise do comportamento aplicada como intervenção para o autismo: definição, características e pressupostos filosóficos**. Revista Educação especial, Santa Maria, v. 26, n. 47, p. 639-650, dez. Disponível em: <https://periodicos.ufsm.br/educacaoespecial/article/view/6994>. Acesso em: 12 dez. 2019.

9. Castorina, J. A. et al. (2006). **Piaget - Vygotsky: Novas contribuições para o debate.** São Paulo, SP: Editora ática.. Csikszentmihalyi, M. (1999). **A descoberta do Fluxo**. [S. l.]: Rocco.

10. Cunha, C. A. da. (2004). **O papel do conflito sociocognitivo na aquisição da conservação de comprimento.** Psicol. estud. [online]. vol.9, n.1, pp.111-118. ISSN 1413-7372. http://dx.doi.org/10.1590/S1413-73722004000100014.

11. Cunha, R. M. (2011). **"Desenvolvimento e avaliação de um jogo de computador para ensino de vocabulário para crianças com autismo,"** Proceedings of Games for Change.

12. D' Carlo, D., Barbosa, G. A. R.; Oliveira, E. R. (2018). **Usabilidade em Aplicativos Móveis Educacionais: Um Conjunto de Heurísticas para Avaliação**. XXVII Simpósio Brasileiro de Informática na Educação. [S. l.]. Disponível em: http://www.br- ie.org/pub/index.php/sbie/article/view/6763. Acesso em: 8 mar. 2019.

13. Farias, E. B. (2017). **Validação empírica de uma abordagem para alfabetização de autistas utilizando aplicativos para dispositivos móveis.** Dissertação (Mestre em Informática) - Universidade Federal de Alagoas, Maceió -AL, 2017. Disponível em: http://www.repositorio.ufal.br/handle/riufal/2687. Acesso em: 31 mar. 2019.

14. Ferreira, W. et al. (2018). **Panorama das Publicações Nacionais sobre Autismo, Educação e Tecnologia**. Simpósio Brasileiro de Informática na Educação (SBIE), [S. l.].

15. Fuentes, J. et al (2014**). Autism Spectrum Disorder. Developmental Disorders,** [S. l.]. Disponível em: http://iacapap.org/wp-content/uploads/C.2-ASD-2014-v1.1.pdf. Acesso em: 16 mar. 2019.

16. Foxx, R. M. **Applied Behavior Analysis Treatment of Autism: The State of the Art. Child and Adolescent** Psychiatric Clinics of North America, [s. l.], v. 17, ed. 4, p. 821-834, 2008. DOI https://doi.org/10.1016/j.chc.2008.06.007. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S1056499308000412. Acesso em: 30 mar. 2019.

17. Gadia, C. A.; Tuchman, R.; Rotta, N. T. (2004). **Autismo e doenças invasivas de desenvolvimento.** Disponível em: <http://www.scielo.br/scielo.php?pid=S0021->. Acesso em: 14 mar. 2019.

18. Gomes, C. G. S.; Silvera, A. D. (2016a). **Ensino de habilidades básicas para crianças com autismo: Manual para intervenção comportamental intensiva**. 1º. ed. Curitiba/PR: Appris. 133 p.

19. Gomes, C. G. S.; Souza, D. das G. de. (2016b)**. Ensino de Sílabas simples, Leitura Combinatória e Leitura com Compreensão para aprendizes com autism.** Revista Brasileira de Educação Especial, [S. l.]. Disponível em: http://www.scielo.br/scielo.php?pid=S1413-65382016000200233&script=sci_abstract&tlng=pt. Acesso em: 24 fev. 2019

20. Hungate, M., Gardner, A. W., Fackett, S., & Spencer, T. D. (2019**). A convergent review of interventions for school-age children with autism spectrum disorder.** Behavior Analysis: Research and Practice. Disponível em :http://dx.doi.org/10.1037/bar0000090 . Acesso em: 14 mar. 2019.

21. Jucá, S. C. S. (2006). **A relevância dos softwares educativos na educação profissional.** Ciências & Cognição, V8: p 22-28 [S. l.]. Revista interdisciplinar de estudos de cognição. Disponível em: http://www.cienciasecognicao.org/revista/index.php/cec/article/view/571/359. Acesso em: 02 jan. 2019.

22. Junior, O. de O. B.; Aguiar, Y P. C. (2018). **Taxonomia de Critérios para Avaliação de Software Educativo – TaCASE**. Simpósio Brasileiro de Informática na Educação, [S. l.]. Disponível em: http://www.br-ie.org/pub/index.php/sbie/article/view/7984. Acesso em: 4 mar. 2019.

23. Kwee, C S. (2006**). Abordagem Transdisciplinar no Autismo: O Programa TEACCH**. 110 p. Dissertação (Mestrado Profissionalizante em Fonoaudiologia) - Universidade Veiga de Almeida, Rio de Janeiro.

24. Macedo, E. D. N. (2011). **O docente e o mundo do autismo.** 72 p. Dissertação (Mestrado em Ciências da Educação na área de Educação Especial) - Escola Superior de Educação Almeida Garrett, Lisboa – Portugal.

25. Machado, E. M; Cândido, C. C. (2017). **Software Luz do Saber Infantil: Recurso Didático Alfabetizador.** Congresso Internacional sobre Informática na Educação (TISE), Santiago, Chile.

26. Moraes, S. P. C. de. (2008**). A concepção de aprendizagem e desenvolvimento em Vigotski e a avaliação escolar.**. [S. l.], Disponível em: http://www.histedbr.fe.unicamp.br/acer_histedbr/jornada/jornada11/artigos/9/artigo_simposio_9_10 08_silvia.moraes@uol.com.br.pdf. Acesso em: 1 fev. 2019.

27. Neves, L. A.; Seixas, P.; Ferraz, F. (2018). **Utilização de um aplicativo para Smart TV como Ferramenta de Apoio para Alfabetização de Crianças com Síndrome de Down**. Conferência

28. Internacional sobre Informática na Educação (TISE), Brasilia, Brasil. Disponível em: http://www.tise.cl/Volumen14/TISE2018/585.pdf. Acesso em: 10 mar. 2019.

29. Neves, L. A.; Kanda, J. Y. (2016). **Desenvolvimento e Avaliação de Jogos Educativos para Deficientes Intelectuais.** Conferência Internacional sobre Informática na Educação (TISE), Santiago, Chile. Disponível em: http://www.tise.cl/volumen12/TISE2016/612-617.pdf. Acesso em: 10 mar. 2019.

30. Nunes, R. P.; Santos, I. M. (2018). **A importância da avaliação ergonômico-pedagógica de aplicativos educacionais e os desafios encontrados no campo da aprendizagem móvel.** Simpósio Brasileiro de Informática na Educação, [S. l.]. Disponível em: http://br- ie.org/pub/index.php/sbie/article/view/7952. Acesso em: 4 mar. 2019.

31. Oliveira, G. C. de et al. (2015**). Considerações da aplicação do método PECs em indivíduos com TEA.** Estudos, Goiania, v. 42, n. 3, p. 303-314, mar. Disponível em: <http://tede2.pucgoias.edu.br/index.php/estudos/article/view/4129/2359>. Acesso em: 27 Jan. 2019.

32. Oliveira, U. P. de; Thereza, W. B. (2018). **O Uso de TICs para Inclusão Social de Crianças com Transtorno do Espectro Autista.** Conferência Internacional sobre Informática na Educação (TISE), Brasília, Brasil. Disponível em: http://www.tise.cl/Volumen14/TISE2018/569.pdf. Acesso em: 3 abr. 2019.

33. Passerino, L. M. (2005**). Pessoas com Autismo em Ambientes Digitais de Aprendizagem: estudo dos processos de Interação Social e Mediação**. 317 p. Tese (Doutor pelo Programa de Pós-Graduação em Informática na Educação) - Universidade Federal do Rio Grande do Sul, Porto Alegre/RS.

34. Pereira, W. S. et al. (2016). **Avaliação de Software Educativo: Análise de Abordagens para Definição de Diretrizes.** Conferência Internacional sobre Informática na Educação (TISE), Santiago, Chile. Disponível em: http://www.tise.cl/volumen12/TISE2016/557-562.pdf. Acesso em: 8 mar. 2019.

35. Silva, R. S. da et al. (2016)**. Avaliação de Software Educativo: a complexidade de escolher uma abordagem adequada.** In: Congresso Regional sobre Tecnologias na Educação (CTrl+e) Natal - RN. Disponível em: http://ceur-ws.org/Vol-1667/CtrlE_2016_AC_paper_26.pdf. Acesso em: 24 fev. 2019.

36. Silva, Á. V. P. da. (2014). **Criação e validação de checklists especificas para categorias de softwares educacionais.** 64 p. Monografia (Licenciatura em Ciência da Computação) - Universidade do Estado de Mato Grosso, UNEMAT, Alto Araguaia/MT.

37. Taile, Y. de L.; Oliveira, M. K. de; Dantas, H. (1993**). PIAGET, VYGOTSKY, WALLON: Teorias Psicogenéticas em Discussão**. São Paulo, SP: Summus.

38. Valente, J. A. (Org.). (1998)**. Computadores e Conhecimento: Repensando a educação.** 2. ed. Campinas/SP: Unicamp/NIED. 513 p. v. 1.

39. Valentim, F. O.; Oliveira, A. A. S. de. (2013). **Avaliação da aprendizagem e deficiência intelectual na perspectiva de professores do ensino comum**. Revista Dialogo Educacional, [S. l.], Disponível em: https://periodicos.pucpr.br/index.php/dialogoeducacional/article/view/2569. Acesso em: 12 fev. 2019.

40. Vieira, F. M. S. (1999). **Avaliação de software educativo: reflexões para uma análise criteriosa**. Disponível em: <https://pt.scribd.com/document/222436005/Avaliacao-de-Software--Educativo- Reflexoes-Para-Uma-Analise-Criteriosa>. Acesso em: 01 mar. 2019.

41. Vygotsky, L. S. (1998). **Formação Social da Mente**. 4º Edição. - São Paulo: Livrarias Martins Fontes. 90 p.

42. Zakari, H. M.; Ma, M.; Simmons, D. (2014). **A Review of Serious Games for Children with Autism Spectrum Disorders (ASD). International Conference on Serious Games Development and Applications.** Springer, Cham, Berlin, Germany. Disponível em: https://www researchgate.net/publication/291345119_A_Review_of_Serious_Games_for_Children _with_Autism_Spectrum_Disorders_ASD>.. Acesso em: 01 mar. 2019.

03

“ *Benchmark* para análise comportamental do sistema de memória virtual do linux

- Rúbens Antônio **Rodrigues**
  IFSP
- Lourenço Alves **Pereira Júnior**
  ITA

doi 10.37885/201102053

RESUMO

Conhecer o desempenho do sistema operacional é importante para o projeto e especificação do ambiente de implantação de serviços computacionais. O desempenho pode variar quando exposto a diferentes cargas de trabalho: em um ambiente controlado pode-se excitar o sistema em estudo em diferentes níveis de exigência computacional. Especificamente, no caso do sistema de memória virtual do Linux, pode haver degradação do desempenho e até mesmo sua inutilização no caso de *thrashing*. Muito embora, seja possível perceber a ocorrência desses fenômenos, a existência de uma ferramenta que permita a condução de experimentos a fim de produzir dados empíricos para a estimação de modelos matemáticos que representam o sistema de memória virtual do Linux é muito escassa. Uma ferramenta como esta auxiliaria no entendimento do comportamento e na predição de ações em situações adversas. Este trabalho apresenta 1) a proposta de um *benchmark* para este tipo de aplicação e 2) resultados que demonstram seu funcionamento.

Palavras-chave: Análise de Desempenho do Linux, Avaliação do Comportamento Dinâmico do Sistema de Memória, Ferramenta de Benchmark.

INTRODUÇÃO

O desempenho de um sistema operacional é influenciado pelo tipo da carga de trabalho. Dependendo da característica desta, todo o sistema ou partes distintas dele podem ficar com o desempenho comprometido. Dentre estas partes há o sistema de memória, e, no caso de sua degradação de desempenho, também conhecida como *thrashing* (Denning, 1968), o sistema fica inutilizável. A identificação de sistema possibilita que projetistas, engenheiros e administradores façam sua adequação para que ele possa oferecer o melhor desempenho de acordo com uma carga recebida. Uma vez identificado o modelo do sistema, é possível desenvolvê-lo de forma que seja auto ciente, ou seja, que conheça a sua capacidade e se adeque para atender aos diferentes tipos de cargas (Aguirre, 2007; Hellerstein *et al.*, 2004; Ljung, 1999).

A identificação de sistemas é feita em um ambiente controlado (Tedre and Moisseinen, 2014), no qual é imposta uma carga ao sistema e seu desempenho é observado. Para essa observação, elenca-se um conjunto de variáveis, sejam elas de entrada (variáveis independentes) ou de saída (métricas de desempenho) e, com isso, executa-se o sistema com diferentes configurações para obter-se os valores impostos e os que foram produzidos (Aguirre, 2007). Por meio de algoritmos matemáticos é viável fazer uma relação de causa e efeito entre eles, obtendo-se, assim, uma representação do sistema. A Identificação de Sistemas Computacionais tem apresentado ganhos consideráveis em Sistemas auto cientes de desempenho (Bae *et al.*, 2011; Ermon, 2008; Yang and Liu, 2010; Klein *et al.*, 2013; Klein *et al.*, 2014). Atualmente, um grande destaque pode ser dado à Computação em Nuvem. Projetar um sistema que saiba de seus limites e que seja capaz de se adaptar a partir das alterações em sua carga, possibilita que níveis de serviço sejam acordados (*Service Level Agreement* — SLA) eficientemente. A identificação em sistemas computacionais geralmente é feita em formato de caixa preta. Em termos práticos, faz parte do processo: excitar o sistema a fim de expor seu comportamento dinâmico; os dados gerados nessa etapa são passados para algoritmos de estimação que fixam parâmetros do(s) modelo(s); a partir do modelo, é feita uma análise de como uma determinada variável de entrada influencia no resultado de uma métrica; e, por fim, o sistema é modificado de tal maneira que seja possível observar alterações nas variáveis de entrada e reagir de forma adequada para que a variável de saída esteja nos patamares de qualidade desejados (Quality of Service— QoS) (Hellerstein *et al.*, 2004).

O Linux é um sistema operacional de código aberto amplamente utilizado. Ele possui um sistema de memória virtual baseado em páginas, no qual é possível alocar uma quantidade de memória maior do que a quantidade fisicamente disponível e, para isso é utilizado uma

extensão do disco. Assim como outros sistemas operacionais, o seu sistema de memória pode ficar comprometido quando submetido à determinada carga.

Para a identificação de um sistema ou de uma subparte dele, utilizam-se ferramentas de *benchmark*, que se constituem como dispositivos de software capazes de impor uma demanda de trabalho a um sistema e coletar dados que permitem avaliar seu custo computacional na realização de uma determinada tarefa. Tais dispositivos podem tanto ser compostos por softwares pré-existentes quanto ser confeccionados para atender tarefas mais específicas. O Linux, por ter seu código fonte aberto, possibilita a sua recodificação para inserção de peças de software para a análise de seu comportamento.

Este trabalho faz a abordagem de uma proposta de *benchmark* para o estudo do comportamento dinâmico do sistema de memória do Linux. Ele parte de uma investigação de ferramentas e técnicas de *benchmark* existentes; faz análise de uma arquitetura para a construção de uma ferramenta junto com o levantamento de parâmetros para execução da mesma; executa experimentos para sua avaliação; e finaliza com a demonstração e discussão dos resultados obtidos.

OBJETIVO

Identificar o comportamento dinâmico do sistema de memória de um sistema operacional se faz importante para adequá-lo, ou permitir que se adeque de forma automatizada, a diferentes cargas de trabalho. Este trabalho tem por objetivo apresentar uma proposta de um *benchmark*, cujas funcionalidades auxiliam na análise comportamental do sistema de memória do Linux de acordo com uma carga sintética em um ambiente controlado, analisando e identificando as responsabilidades dos componentes que o compõem. E também objetiva-se a apresentar resultados empíricos obtidos através da execução de um protótipo em ambiente controlado. Outra contribuição diz respeito à reprodutibilidade dos experimentos conduzidos com o *benchmark* proposto.

Diferentes versões do *kernel* do Linux podem apresentar comportamentos distintos, o que justifica o uso deste *benchmark* para um estudo direcionado à versão desejada. A memória virtual utiliza uma área do disco e, por isso, ele pode ser utilizado para a análise do comportamento do sistema em um ambiente com a presença de *Redundant Array of Independent Disks* — RAID, bem como de *Solid-state Driver* — SSD. Outro uso para este *benchmark* é a identificação do sistema em infraestruturas com concorrência por recursos, como é o caso, por exemplo, de um ambiente virtualizado.

TRABALHOS RELACIONADOS

Carneiro *et al.*, (2009) apresentam uma proposta de política de gerenciamento de páginas baseada em uma teoria de controle retroalimentada. Em linhas gerais, é feito um estudo do comportamento de um sistema operacional com base na dinâmica existente entre o tamanho do conjunto de páginas residentes e a taxa efetiva de falta. Trata-se da implementação de uma versão adaptada do algoritmo Page-Fault-Frequency — PFF (Chu and Opderbeck, 1976) para o controle. O maior problema destacado, bem como a motivação do trabalho foi a questão do comportamento estático do algoritmo e a possibilidade de torná-lo dinâmico.

Outra abordagem dinâmica é apresentada por Terraneo and Leva (2013), em que é proposto um sistema ativo de *swap-in* (ou *page-in*). O sistema controla a quantidade de memória principal que pode ser liberada. Dessa forma, traz da área de troca, as páginas que em implementações tradicionais continuariam no disco (memória secundária) e que poderiam gerar ausência de página (*page fault*). A grande contribuição a ser destacada deste trabalho é o formalismo com que o problema é tratado: é feita uma análise das variáveis que influenciam o estado de alocação de memória (principal e secundária) e das respectivas restrições que os valores podem assumir, definindo estados consistentes e inconsistentes. Além disso, destaca-se a malha de controle utilizada.

Conforme destacado na literatura técnica específica para a modelagem dinâmica de sistemas gerais (Aguirre, 2007; Ljung, 1999) e de computação (Hellerstein *et al.*, 2004), a identificação do modelo capaz de representar a relação entre duas variáveis de entrada e saída deve ser feita conhecendo de antemão, que uma influencia na outra.

Os resultados apresentados por Santos and Maziero (2009) e Maziero *et al.* (2013) fundamentam o objeto de estudo deste trabalho na medida em que detalham como é possível observar o comportamento dinâmico (variáveis tratadas como sinais) de quatro sistemas operacionais (Linux, FreeBSD, OpenSolaris e Windows). Nos artigos mencionados, apresenta-se o conjunto de experimentos que foram conduzidos, nos quais se pode apreciar o comportamento daqueles sistemas em situações de carga extrema (*thrashing*).

O diferencial desta proposta em relação aos trabalhos relacionados é que esta é direcionada a experimentação em um ambiente controlado, permitindo a reprodução de experimentos e, também, tendo o cuidado de produzir dados para geração de modelos e análise matemática.

PROJETO DO *BENCHMARK*

Os experimentos para a geração de um modelo que represente o sistema de memória do Linux pode ser feita através do método caixa preta. Para isso, elenca-se um conjunto de

variáveis de entrada e um conjunto de variáveis de saída, altera-se os valores das variáveis de entrada com base em faixas determinadas e observa-se os valores das variáveis de saída. A identificação das variáveis e das faixas nem sempre se dá de forma clara, algumas variáveis são dadas de forma empírica, como quantidade de memória alocada, tempo de acesso a memória (leitura/escrita). Outras variáveis são encontradas somente depois de experimentações para o preenchimento de lacunas até então desconhecidas. As faixas de valores para as variáveis podem ser definidas tendo como base um conhecimento prévio do sistema: um sistema com 2 GB (Gigabytes) de memória, por exemplo, só pode operar com valores entre 0 GB e 2 GB. Elas podem, também, ser definidas por meio de trabalhos como este, que fornecem as faixas que foram utilizadas na execução de experimentos. Ou, ainda, podem ser definidas por meio de experimentação. Assim, estipula-se um valor para uma variável, executa-se um experimento por um curto intervalo de tempo para observar se o valor estipulado é compatível com a execução e, caso não seja, altera-se esse valor até que seja possível identificar um valor que seja adequado.

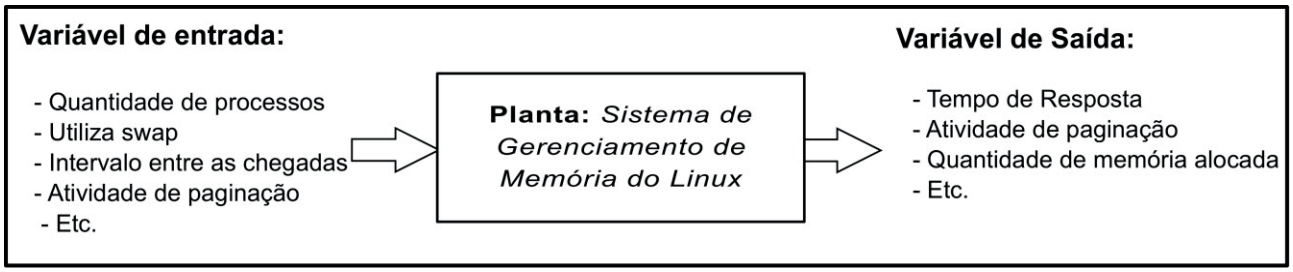
A Figura 1 ilustra um modelo conceitual do sistema de memória do Linux, apontando algumas variáveis que possuem potenciais relacionamentos de causa—efeito. À esquerda estão as variáveis de entrada (independentes). Por meio delas é possível excitar de forma controlada o sistema de memória, que, por consequência, produz as variáveis de saída (dependentes), representadas à direita, as quais podem ser utilizadas para observar o comportamento do sistema. Como entrada pode definir-se a quantidade de processos que serão criados, a quantidade de memória alocada por cada processo, e o intervalo de chegada dos processos, dentre outras. Como saída, podem ser elencados o tempo de resposta, a quantidade de memória alocada, e a atividade de paginação ou qualquer outra variável que seja dependente de uma variável de entrada.

Solução arquitetural

Para excitar de forma controlada o sistema de memória, foi desenvolvido um modelo de *benchmark* que tem como base o algoritmo utilizado por Santos and Maziero (2009). Nesse modelo, são criados três programas: 1) um gerenciador responsável por receber os parâmetros do usuário e orquestrar a execução controlada dos demais programas; 2) um monitor que faz leituras das variáveis de saída e as grava em uma base de dados que permite a manipulação estatística e 3) um gerador de carga que aloca uma área de memória e efetua, dentro dela, operações de leitura e escrita. O intuito do presente *benchmark* é impor uma carga controlada ao sistema de memória do Linux, executando operações de leitura e escrita em endereços de memória pseudoaleatórios, como também obter os estados das variáveis de saída. A execução de um experimento exige um grande período de tempo para

ser concluída e, por isso, para automatizar as execuções, foi idealizado o modelo de um programa que fica em um computador (*Manager*) externo ao que está rodando o experimento (*System Under Test*— SUT), gerenciando, assim, a execução dos experimentos.

Figura 1. Modelo conceitual de causa e efeito para o sistema de memória do Linux. À esquerda estão as variáveis de entrada que podem ser submetidas ao bloco que representa o sistema de gerenciamento de memória do Linux. Este poderá produzir as variáveis à direita utilizadas para observação.



Os componentes do benchmark estão ilustrados na Figura 2. O bloco principal, denominado Benchmark, é composto por dois blocos que representam duas máquinas. Uma delas, denominada Manager, é a responsável por gerenciar a execução de experimentos. A outra, denominada SUT, contém o sistema a ser testado. Dentro do manager estão: o Gerenciador de Experimentos, que é um programa responsável por carregar um conjunto de parâmetros (fatores) e conduzir o experimento de acordo com eles e, uma base de dados, Parâmetros CSV, constituída com os possíveis valores das variáveis de entrada. No SUT está contido o Gerenciador, que tem a função de: 1) receber e interpretar os parâmetros; 2) iniciar o Monitor, para gravar em uma base de dados, Resultados CSV, os valores amostrados periodicamente das variáveis de saída e 3) iniciar de forma parametrizada um conjunto de *n* geradores de carga que alocam uma região de memória que será utilizada para operações de leitura/escrita.

As setas coloridas relacionam-se com a legenda e demonstram o fluxo de execução de experimentos: na seta azul são carregados os parâmetros para execução do experimento; na de cor laranja, a máquina com o SUT é reiniciada; logo que ela retorna a normalidade, na seta marrom os parâmetros são passados para o Gerenciador no SUT, o qual os processa; na seta verde o Monitor é iniciado pelo Gerenciador; na dourada, é parametrizado e iniciado o gerador de carga; pela seta preta, os dados coletados pelo monitor são gravados na base de dados, Resultados CSV; finalmente após o término do experimento, na seta vermelha, os dados são copiados do SUT para o Manager.

Parâmetros CSV

Os parâmetros têm a finalidade de regular a carga imposta ao sistema de gerência de memória e é de acordo com as configurações dos seus valores que se faz possível aumentar a concorrência por tempo de CPU, a quantidade de memória alocada, ou ainda a alteração

do tempo de leitura/escrita em regiões da memória. A Tabela 1 apresenta a relação de parâmetros necessária para este modelo. A sua nomenclatura segue a definição de Santos and Maziero (2009). *N* é um valor inteiro que especifica a quantidade de processos consumidores que será criada a partir do programa gerador de carga; *S* é a área de memória alocada por cada processo; *W* é a quantidade de operações de leitura/escrita durante um ciclo; *tc* é o tempo de entrada utilizado para cada processo consumidor e *ts* é o tempo de repouso entre cada ciclo de operações de leitura e escrita.

Figura 2. Distribuição dos programas do *benchmark*. Os dois blocos principais do *Benchmark* (Manager e SUT) representam duas máquinas, uma para gerenciar o experimento e outra com o sistema em teste. O Manager é constituído pelo Gerenciador de Experimentos e por uma base de dados, Parâmetros CSV. Já o SUT é composto pelo Gerenciador, pelo Gerador de Carga, pelo Monitor e Resultado CSV, uma base de dados com os resultados. As setas coloridas representam o fluxo da execução de experimentos, conforme legenda.

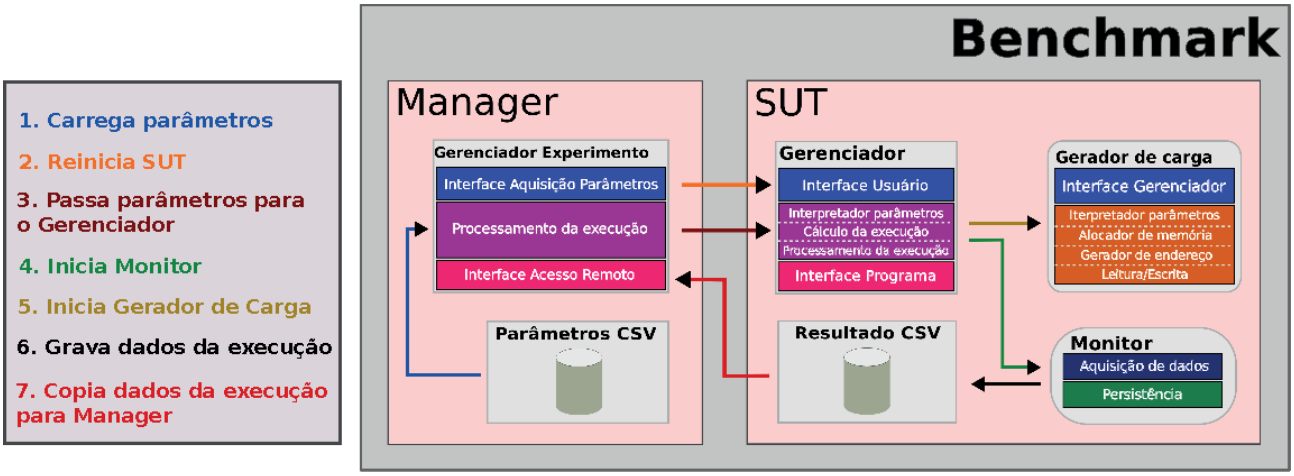


Tabela 1. Relação de parâmetros.

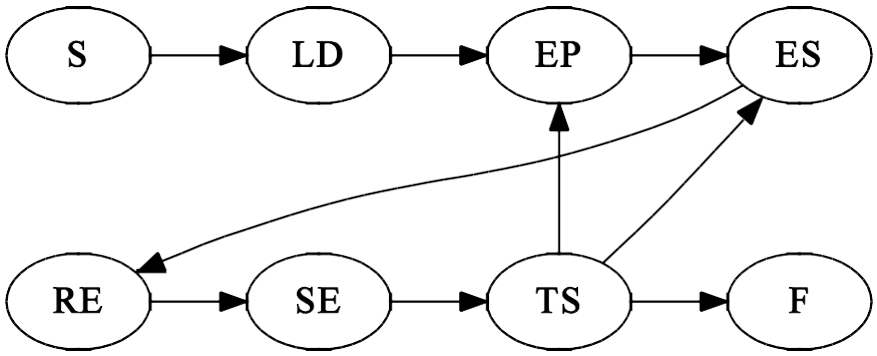
Parâmetro	Significado
N	Quantidade de processos.
S	Área de memória alocada.
W	Quantidade Leitura/Escrita por ciclo.
tc	Tempo de entrada de cada processo consumidor.
ts	Tempo de repouso entre cada ciclo de Leitura/Escrita.

O Gerenciador de Experimentos

Para melhor compreensão, o programa Gerenciador de Experimentos é apresentado em três blocos. O primeiro bloco, de cima para baixo, é a Interface Aquisição Parâmetros. Através dela os parâmetros são carregados para a memória do programa. O próximo bloco, Processamento da execução, é responsável pelo processamento e execução do experimento e, nele, cada conjunto de parâmetros é interpretado e passado em formato texto para o SUT. O último bloco do Gerenciador de Experimentos é a Interface Acesso Remoto, a qual permite que o programa conecte-se ao SUT através do protocolo *Secure Shell* – SSH.

Para seu funcionamento, deve-se criar uma base de dados no formato CSV (*Comma-separated Values*) de acordo com o planejamento do experimento. Seu funcionamento é ilustrado na Figura 3. O programa é iniciado (*S*), carregam-se os dados da execução (*LD*), então entra em um laço (*EP*) onde para cada parâmetro entra em outro laço (*ES*) em que para cada semente o executor é reiniciado (*RE*). Os parâmetros de execução são passados para o executor (*SE*), quando o mesmo finaliza (*TS*), se houver mais sementes, repete (*ES*), caso contrário se houver parâmetros, repete (*EP*) e senão houver nem semente e nem parâmetro, sua execução é finalizada (*F*).

Figura 3. Estados do programa Gerenciador de Experimentos.



O Gerenciador

O programa Gerenciador é divido três partes. De cima para baixo, a primeira é a Interface Usuário, por meio dela é possível a interação de um sistema com o programa, passando os parâmetros da execução do experimento em formato texto. A segunda parte, ainda de cima para baixo, é a responsável pelo processamento e execução do experimento. Ela está subdividida em três partes, que são as principais rotinas do programa. Na Interpretador parâmetros, ocorre a conversão dos parâmetros do formato texto para os tipos usados pelo programa. Após a conversão, ocorre a organização dos dados a serem utilizados pela rotina que calcula os tempos da execução, Cálculo da execução. Por fim, na Processamento da execução, são criados processos Geradores de Carga de acordo com os parâmetros e com os dados calculados pela rotina de cálculo da execução. A última parte do Gerenciador é a Interface Programa. Nela, por meio de chamada de sistema para execução de programas, é possível executar o monitor e o gerador de carga.

O cálculo dos tempos da execução do experimento é importante para que se tenha controle sobre a carga do sistema de memória do Linux. Os tempos calculados são o tempo de entrada (*tci*) de cada processo, criado a partir do programa gerador de carga, e também os tempos de saída (*tfi*). Para o tempo de entrada, tem-se:

$$tc_i = tc.k, \quad (1)$$

em que *tc* é um parâmetro fornecido pelo usuário e *k* é uma constante utilizada como base de tempo na ordem de *ms* (milissegundos). Para o tempo de saída, tem-se:

$$tf_i = tc \,.N\,.k \tag{2}$$

Considera-se k > 0, para as Equações (1) e (2).

Gerador de Carga

O Gerador de Carga é responsável por receber parâmetros do gerenciador, interpretá-los e excitar de forma controlada o sistema de memória de acordo com os valores passados por eles. Ele é constituído pela Interface Gerenciador e, através dela, é possível que o programa Gerenciador passe parâmetros durante o carregamento para que, dessa forma, se possa especificar a quantidade memória que será alocada e os tempos entre os ciclos de leitura/escrita. Os parâmetros passados são a área de memória alocada (*S*), a quantidade de leitura/escrita por ciclo (*W*) e o tempo de repouso entre os ciclos de escrita (*ts*).

Ainda acerca de sua constituição, há um bloco dividido em quatro partes que são as principais rotinas desse programa. De cima para baixo, a primeira delas é o Interpretador de Parâmetros, é onde ocorre a conversão dos parâmetros, de texto para formatos adequados ao programa. Na sequência, tem-se a rotina de alocação de memória, Alocador de memória, a qual aloca uma área de memória de acordo com o parâmetro *S*. A próxima rotina, Gerador de endereço, gera aleatoriamente endereços que sofrem operações de leitura/escrita. A última rotina, Leitura/Escrita, acessa cada um desses endereços gerados para efetuar, neles, uma operação de leitura e uma de escrita. As duas últimas rotinas citadas são organizadas dentro de um laço e são executadas de acordo com o parâmetro *W*, o qual especifica a quantidade de iteração do laço. Após o termino das iterações, ocorre um tempo de repouso de acordo com o parâmetro *ts* e, após esse período, a execução inicia-se novamente no laço gerando, assim, o ciclo de leitura/escrita que se repete até o término do experimento.

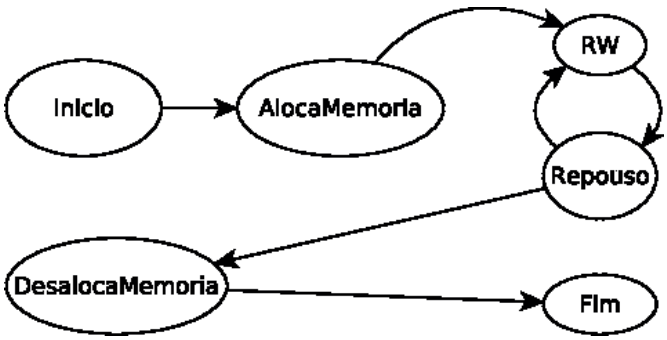
Em um experimento, o gerador de carga poderá criar de 1..*n* processos geradores de carga. O valor de *n* é especificado por meio de parâmetro fornecido, que deve ser calculado para ficar em conformidade com a memória do sistema (*mt*). Dessa forma, realiza-se um cálculo para identificar os valores dos parâmetros *S* e *N*, visando que se adeque dentro dos limites da memória, Equação (3). Por exemplo, em um sistema com 512 MB de memória livre, não pode existir dois processos alocando mais que 256 MB de memória cada um, pois, dessa forma, a soma da quantidade de memória especificada é maior que a quantidade disponível no sistema. Formalmente,

$$N = \frac{mt}{S} \tag{3}$$

Portanto, para que haja sucesso na execução do experimento: $1 \leq N$.

Com base em parâmetros adequados recebidos do gerenciador, o gerador de carga entra em funcionamento conforme demonstrado na Figura 4. Ele é iniciado, aloca memória conforme parâmetro, efetua operações de leitura/escrita, entra em repouso por um tempo fornecido e, após isso, desaloca a memória e é finalizado.

Figura 4. Estados do processo gerador de carga. Pode estar iniciando, alocando memória, executando operações de leitura e escrita, desalocando memória ou finalizado.



A rotina para a geração de endereço utiliza uma função pseudoaleatória alimentada por semente (*seed*), o que garante a reprodutibilidade da execução. Com isso, uma vez que a mesma semente for utilizada, a sequência gerada será a mesma em todas as execuções. A função empregada na geração de endereços não deve apresentar periodicidade, caso isso aconteça, o sistema de memória poderá se beneficiar de algum mecanismo de *cache*. Para validar os endereços gerados, executa-se a rotina de geração de endereços durante um período de tempo e, assim como demonstra Santos and Maziero (2009), converte-se a série gerada em coordenadas (*x*, *y*) fazendo uso da técnica de plotagem do espaço de fase (Fraser and Swinney, 1986). Para que uma rotina não favoreça o sistema de memória, esta deve apresentar um gráfico conforme a Figura 5, gerada a partir dos endereços gerados com a função random nativa do Linux. Ao todo, foram gerados 187.750 endereços. Como pode ser observado, os pontos estão distribuídos de maneira uniforme, com formação similar a de uma "nuvem".

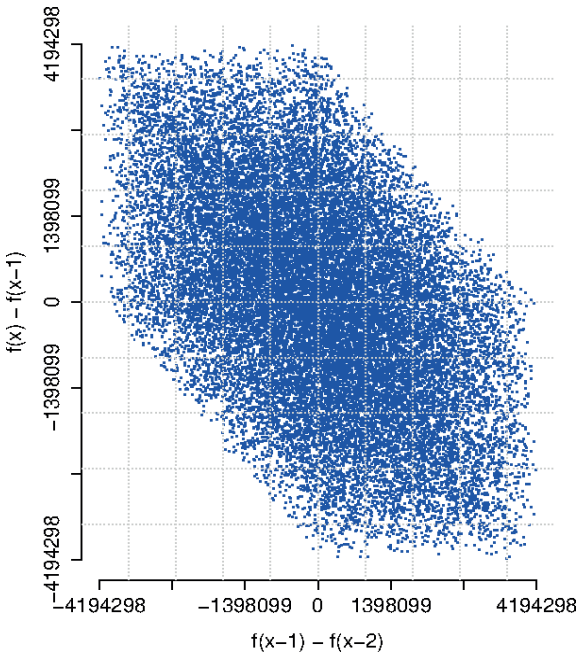
O Monitor

O Monitor faz leituras periódicas dos estados das variáveis de saída e os grava em uma base de dados, a qual permite a manipulação por meio de ferramentas estatísticas. Ele pode ser dividido em duas partes conforme o diagrama de blocos, sendo que o bloco superior, Aquisição de dados, representa a interface responsável por ler as variáveis de saída e adequá-las ao programa, enquanto que o bloco inferior, Persistência, é o responsável por gravar os valores das variáveis de saída em uma base de dados.

As variáveis de saída definidas são: o uso da CPU, a quantidade de memória alocada, a taxa de paginação (quantidade de páginas de memória transferida entre a memória física

e o disco), o tempo decorrido desde o início da execução do experimento e jiffies (tempo contabilizado no *kernel* desde o *boot* do sistema). No Linux, os dados relativos à CPU, memória e paginação podem ser obtidos através do sistema de arquivos proc.

Figura 5. Plotagem dos endereços gerados no espaço de fase. Há ausência de atratores, significando que não há uma localidade forte dos endereços gerados. Portanto, como há uma formação dispersa, conclui-se que o impacto da memória *cache* pode ser negligenciada.



jiffies é uma variável do tipo unsigned long e seu valor é incrementado a cada tick (interrupção de timer). Como o valor dessa variável não está disponível para o usuário, é preciso fazer uma instrumentação do Linux para a extração do seu valor. Para tal, uma possibilidade é programar um *kernel module* que crie uma entrada no sistema de arquivos proc, faça a leitura do valor de jiffies e o exporte para o usuário.

Resultado CSV

A base de dados de resultados é organizada em um arquivo no formato CSV. Os dados são organizados e gravados, respectivamente, como: tempo decorrido desde o início da execução do experimento, jiffies, taxa de utilização da CPU em modo usuário, taxa de uso da CPU em modo *kernel*, taxa total de uso da CPU, total de memória utilizada, taxa de páginas de memória transferidas do disco para a memória principal (*page-in*) e taxa de páginas de memória transferida da memória principal para o disco (*page-out*).

MÉTODOS

Com o intuito de testar o *benchmark*, dois experimentos foram planejados. A Tabela 2 apresenta os parâmetros utilizados na execução deles. Para ambos os experimentos, com

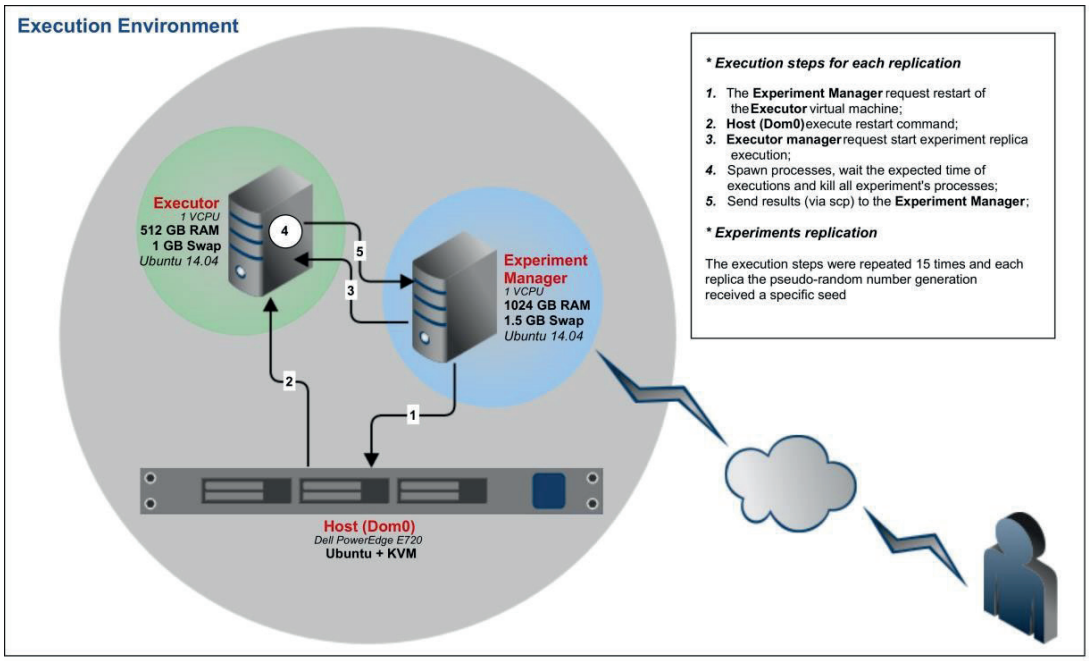
exceção do tempo de entrada dos processos, que é igual a 2 segundos no experimento número 1 (entrada como rampa) e 2 milissegundos no experimento 2 (entrada como degrau), os demais parâmetros têm valores iguais para ambos os experimentos. Estes são: 150 processos Geradores de Carga, 8 MB de memória alocada por cada processo, 5 segundos de tempo de repouso e 1.000 operações de leitura/escritas por ciclo. Para cada configuração do experimento foram executadas 15 (quinze) réplicas, sendo que, para cada réplica, foi utilizada uma semente diferente para alimentar o gerador de endereços.

Tabela 2. Parâmetros utilizados na execução dos experimentos 1 e 2.

Variável	Experimento 1	Experimento 2
Quantidade de processos.	150	150
Quantidade de memória alocada.	8 MB	8 MB
Tempo de repouso.	5s	5s
Leitura/escrita por turno.	1000	1000
Tempo entre as entradas	Rampa (2 s)	Degrau (2 ms)

Para a execução do experimento, foi utilizado um ambiente virtualizado conforme ilustrado na a Figura 6.

Figura 6. Ambiente de execução. Uma máquina física (Host (Dom0)) com Ubuntu + KVM que hospeda duas VMs: Experiment Manager e Executor, ambas com o sistema operacional Ubuntu 14.04. As setas indicam o fluxo do processamento do experimento. A Experiment Manager atua sobre Host (Dom0) e Executor para a execução do experimento. É necessária a intervenção humana apenas para iniciar o experimento, e após sua finalização, para a coleta dos dados.



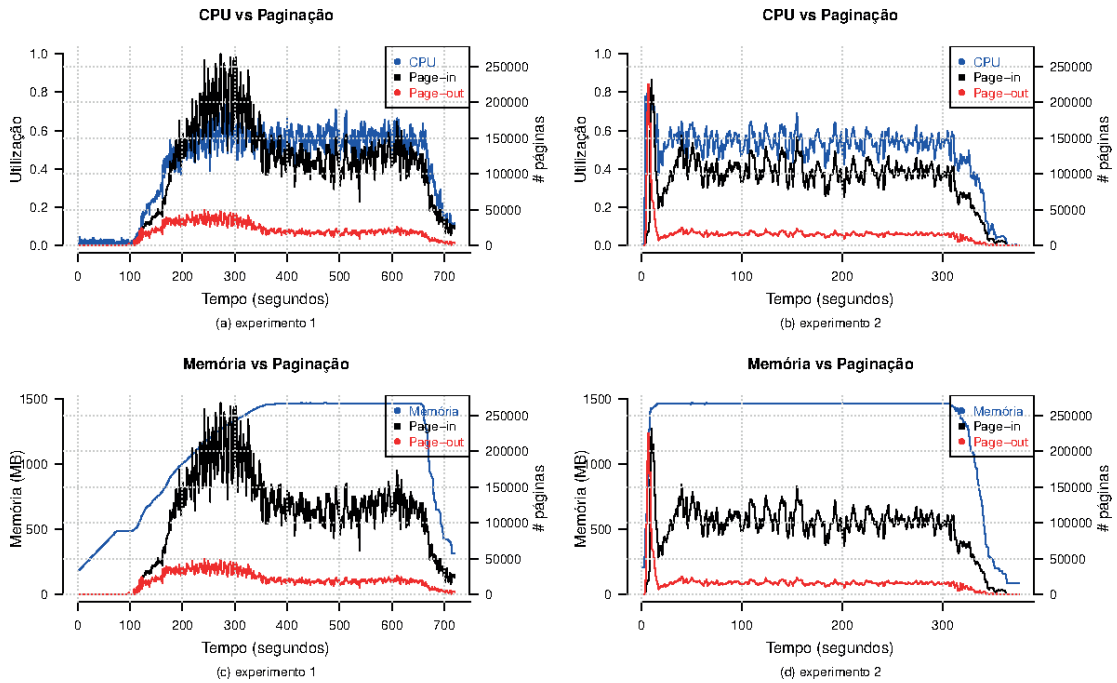
É possível visualizar duas VMs (*Virtual Machine* — Máquinas Virtuais), a Executor e a Experiment Manager, sendo executadas na máquina física Host (Dom0), bem como um usuário. Por meio de acesso remoto, o usuário configura o experimento na Experiment

Manager e invoca o comando para iniciá-lo; a Experiment Manager solicita ao Host (Dom0) para reiniciar a Executor; os parâmetros são passados para a Executor e, assim que o experimento é finalizado, os dados são copiados para a Experiment Manager através do comando scp (*Secure Copy*). É necessária a intervenção humana apenas duas vezes, uma para iniciar o experimento e outra, ao final, para a coleta dos dados resultantes.

RESULTADOS

Os resultados da execução são apresentados nos gráficos da Figura 7 (os valores representam as médias dos valores obtidos com as execuções das réplicas). Dispostos em quatro gráficos, os resultados do experimento relacionam a demanda por CPU e a alocação de memória com a movimentação de páginas na memória virtual (*page-in*, *page-out*). À direita estão os gráficos do primeiro experimento e, à esquerda, os do segundo. Em cima estão os gráficos da utilização de CPU e, em baixo, os de alocação de memória. Em cada gráfico há três curvas. A curva em azul se relaciona com o eixo vertical mais à esquerda, que é a métrica em análise relacionada com a movimentação das páginas. As curvas de cor preta e vermelha representam a movimentação das páginas – *page-in* e *page-out* respectivamente –, e se relacionam com o eixo vertical mais à direita. Ambos são analisados em função da variável independente, Tempo, sendo seu valor dado em segundos no eixo horizontal.

Figura 7. Resultados



DISCUSSÃO

Foram relacionadas a utilização da CPU e a alocação de memória com a movimentação de páginas entre a memória e o disco nas operações de *page-in* e *page-out*. Como pode ser visualizado nos gráficos, as taxas de entrada e de saída de páginas são assimétricas, ou seja, a taxa de entrada é maior que a de saída. Esse comportamento pode ser produto da política de substituição de páginas do Linux, no qual são reescritas no disco apenas as páginas que foram modificadas.

A CPU é influenciada pela entrada de páginas (Figura 7a), uma vez que ocorre um aumento da sua utilização logo após o sistema começar a fazer paginação. Tal comportamento também pode ser atribuído ao algoritmo de substituição de páginas, porém, para evidenciar este fato é necessário realizar um estudo direcionado. Na Figura 7b, ocorre um pico seguido de uma oscilação e, por volta dos 50 segundos, volta-se ao repouso. Quando relacionadas as Figura 7 (a) e (b), nota-se que a taxa de entrada de páginas apresenta uma diferença de aproximadamente 11, 1%, 270.000 para excitação como rampa, enquanto 240.000 como degrau. Essa diferença pode ocorrer por três diferentes motivos, a saber: 1) pelo menor grau de paralelismo, pois não havia operações de leitura/escrita durante a criação dos processos; 2) pelo mecanismo de carregamento de programas do Linux que apenas mapeia as páginas de memória para o arquivo no disco, deixando para fazer sua carga para a memória quando o processo demandar; 3) na Figura 7b o algoritmo já sabe, de antemão, todas as páginas a serem alocadas, fazendo com que a escolha seja mais certeira de que na Figura 7a, caso em que uma página que seria a melhor escolha no instante de tempo *t* acontecerá no futuro ($t + \Delta t$), portanto indisponível.

Analisando a Figura 7c, uma grande diferença entre as atividades de *page-in* e *page-out* é visível, aproximadamente 90%, a qual pode ser atribuída, novamente, à política de substituição de páginas. Na atividade de alocação de memória, entre os instantes 75 e 105, ocorre uma parada que pode ter sido ocasionada pelo escalonador de entrada e saída, que tem por finalidade minimizar a movimentação das cabeças do disco. A atividade de paginação que se iniciou no instante 105, dá suporte a essa afirmação. Na Figura 7d, entre os instantes 2 e 15, ocorre um pico de *page-out* provocado pela alta demanda de páginas gerada a partir do instante de tempo 1, momento em que chega uma rajada de processos provocando muitas faltas de páginas. Devido a isso, a quantidade de páginas transferidas para o disco nesse momento é alta e, aproximadamente no instante 35, o comportamento fica similar ao dos outros ensaios.

CONCLUSÃO

Foi proposto o projeto e a implementação de um protótipo de um *benchmark* com a finalidade de auxiliar na produção de dados para a análise comportamental do sistema de memória do Linux em um ambiente controlado. Cada componente tem suas responsabilidades bem definidas: gerenciamento da execução do experimento, tratamento de parâmetros e gerenciamento da carga, gravação de *log* e geração da carga imposta ao sistema. Idealmente, eles devem ser distribuídos em máquinas distintas, uma com o sistema em teste e outra para gerenciar a execução do experimento. A execução de experimentos constitui-se na definição de um conjunto de fatores, bem como na alimentação de uma base de dados CSV e na iniciação do conjunto de ferramentas para execução do experimento. Ao término da execução, as bases de dados no formato CSV são disponibilizadas para o processamento por meio de ferramentas estatísticas. O fato de o gerenciador de experimentos ser executado em um ambiente externo ao do sistema em teste possibilitou a automatização da execução de uma série de experimentos, os quais tendem a ser longos — tempo de execução e replicação. Esse mecanismo possibilitou que toda a execução recebesse intervenção humana apenas duas vezes, para iniciar o experimento e, após o seu término, para a coleta dos dados.

Foram definidos dois experimentos com a finalidade de testar o *benchmark*. Nos resultados produzidos pela execução, pode ser visualizada a existência de diferenças entre as taxas de *page-in* e *page-out*. A movimentação de páginas entre a memória e o disco influencia na taxa de utilização da CPU. A alocação da memória é uniforme enquanto o sistema não entra em *thrashing*. A taxa de paginação durante a criação de processos é menor quando o sistema é excitado em forma de degrau se relacionado com a excitação em forma de rampa. Com base nesses resultados, sugere-se o planejamento de experimentos do tipo fatorial completo de dois níveis (Jain, 1991; Montgomery, 2006), contendo as variáveis descritas na Tabela 3. Ao final da execução dos experimentos, será possível realizar uma regressão linear capaz de relacionar as influências das variáveis de entrada em relação às variáveis de saída. A variável "tempo de resposta" corresponderá ao tempo gasto para realizar as escritas por vez. Durante o intervalo de amostragem (1 s), será calculada a média dos tempos de respostas daquele intervalo. Cada processo escreverá seu log de tempos de resposta, contendo o instante de tempo (jiffies) e o valor médio do intervalo. A média dos processos será extraída de acordo com o arquivo de aferições (que também contém o instante de tempo).

Tabela 3. Planejamento para experimentos futuros

Tipo da variável	Fator 1	Níveis
Entrada	Número de processo	150, 300
Entrada	Quantidade de memória alocada por processo	4 MB, 8 MB
Entrada	Tempo de entrada	2 ms, 10 ms

Tipo da variável	Fator 1	Níveis
Entrada	Tempo de espera entre as escritas	5 s, 2 s
Entrada	Escritas por vez	1000, 4096
Saída	Pag-in e Page-out	–
Saída	Tempo de resposta	–

Trabalhos futuros devem tratar da execução dos experimentos sugeridos. Seus resultados possibilitarão que o modelo da Figura 1 seja completado. Tanto os valores de entrada quanto os de saída podem ser escalares das variáveis, mas também podem ser compostos. Uma vez que os experimentos propostos produzirão influência de fatores, uma possibilidade é que uma métrica para o modelo seja um índice. Outra leva de experimentos que explorarão os valores das métricas envolvidas serão conduzidos para que os algoritmos de estimação de parâmetros sejam executados a partir dos dados empíricos produzidos. Espera-se a criação de um modelo que seja útil para a implantação em sistemas de execução online, possibilitando, assim, a criação de políticas de admissão, bem como de políticas para o gerenciamento de memória em sistemas operacionais. O *benchmark* pode ser estendido para diferentes funções de excitação (PRBS (Aguirre, 2007), senóide (Hellerstein *et al.*, 2004) etc.). Além disso, ele pode ser implementado para outros sistemas operacionais como, por exemplo, o FreeBSD, o OpenBSD, o Solaris, o MacOS etc.

AGRADECIMENTOS

Os autores agradecem ao IFSP, câmpus Araraquara, pelo apoio durante a execução deste trabalho.

REFERÊNCIAS

1. Aguirre, L. (2007). *Introdução à Identificação de Sistemas – Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*. Editora UFMG, 3rd edition.

2. Bae, C., Lange, J. R., and Dinda, P. A. (2011). Enhancing virtualized application performance through dynamic adaptive paging mode selection. In ICAC, pages 255–264.

3. Carneiro, I. S. C., Barreto, L. P., and Sá, A. S. d. (2009). Aplicação e análise de teoria de controle realimentado no tratamento de faltas de páginas em sistemas de gerenciamento de memória. *In VI Workshop de Sistemas Operacionais*. Bento Gonçalves, RS.

4. Chu, W. W. and Opderbeck, H. (1976). Program behavior and the page-fault-frequency replacement algorithm. *Computer*, 9(11):29–38.

5. Denning, P. J. (1968). Thrashing: Its causes and prevention. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, AFIPS '68 (Fall, part I),pages 915–922, New York, NY, USA. ACM.

6. Ermon, S. (2008). Optimal contol techniques for virtual memory management. Master's thesis, Università degli studi di Padova.

7. Fraser, A. M. and Swinney, H. L. (1986). Independent coordinates for strange attractors from mutual information. *Phys. Rev.* A, 33:1134–1140.

8. Hellerstein, J. L., Diao, Y., Parekh, S., and Tilbury, D. M. (2004). *Feedback Control of Computing Systems.* John Wiley & Sons.

9. Jain, R. K. (1991). *The Art of Computer Systems Performance Evaluation.* John Wiley & Sons.

10. Klein, C., Maggio, M., Årzén, K.-E., and Hernández-Rodriguez, F. (2013). Introducing service-level awareness in the cloud. In *2013 ACM Symposium on Cloud Computing,* Santa Clara, CA.

11. Klein, C., Maggio, M., Årzén, K.-E., and Hernández-Rodriguez, F. (2014). Brownout: Building more robust cloud applications. In *36th International Conference on Software Engineering (ICSE)*, Hyderabad, India. Accepted for publication.

12. Ljung, L., editor (1999). System Identification (2Nd Ed.): *Theory for the User.* Prentice Hall PTR, Upper Saddle River, NJ, USA.

13. Maziero, C., dos Santos, D., and Santin, A. (2013). Evaluation of desktop operating systems under thrashing conditions. *Journal of the Brazilian Computer Society,* 19(1):29–42.

14. Montgomery, D. C. (2006). *Design and Analysis of Experiments.* John Wiley & Sons.

15. Santos, D. and Maziero, C. (2009). Avaliação do comportamento de sistemas operacionais de mercado em situação de thrashing. In *VI Workshop de Sistemas Operacionais.* Bento Gonçalves, RS.

16. Tedre, M. and Moisseinen, N. (2014). Experiments in computing: A survey. *The Scientific World Journal,* 2014:1–11.

17. Terraneo, F. and Leva, A. (2013). Feedback-based memory management with active swap-in. In *Control Conference (ECC), 2013 European,* pages 620–625.

18. Yang, F. and Liu, J. (2010). Transient analysis of general queueing systems via simulation--based transfer function modeling. In *Simulation Conference (WSC), Proceedings of the 2010 Winter,* pages 1110–1122.

04

“Desafio de aplicativos: uma experiência no Instituto Federal de Goiás (IFG) Câmpus Luziânia

Ernane Rosa **Martins**
IFG

doi 10.37885/201101964

RESUMO

Os desafios encontrados no processo de ensino-aprendizagem resultam na busca por métodos, técnicas e ferramentas que tornem este processo mais intuitivo e divertido. Neste contexto, o presente trabalho tem como objetivo incentivar a produção e a promoção de novos aplicativos reais, com utilidade prática. A proposta metodológica envolve as possibilidades trazidas pela linguagem de programação visual APP Inventor, em uma atividade de ensino aplica em alunos do curso técnico em informática para internet do Instituto Federal de Goiás Câmpus Luziânia. Como resultados obteve-se a ampliação dos conhecimentos, o estimulo a criatividade, a participação em trabalhos em equipe e a aceitação de novos desafios.

Palavras-chave: Aplicativos, Sistemas.

INTRODUÇÃO

Um dos grandes desafios impostos aos docentes atualmente é como aproveitar as ferramentas tecnológicas a favor do ensino. Compreender como os recursos computacionais podem contribuir para o desempenho das ações humanas e ser capaz de explorar programas de computadores e seus componentes é requisito mínimo de domínio dos alunos, conforme as exigências do atual mundo do trabalho . Para despertar mais o interesse por pesquisa nos alunos, é importante a divulgação do desenvolvimento de projetos de pesquisa, pela importância da exposição dos resultados obtidos pelos alunos nos estudos realizados, assim como pela valorização do trabalho discente (COX, 2003).

Os dispositivos móveis estão cada vez mais presentes no cotidiano da sociedade, estes enriquecem a experiência de aprendizado ao agregar à metodologia alguns dos seus principais recursos: a mobilidade, a portabilidade, a interatividade social, a conectividade, a individualidade, dentre outros (VALENTIM, 2009). Sendo assim o presente estudo busca apresentar os resultados do projeto “Desafio de aplicativos”, executado no Campus Luziânia do Instituto Federal de Goiás (IFG). Que visou inserir os discentes em atividades de pesquisa e explorar as potencialidades do software livre APP Inventor.

Este artigo está estruturado nas seguintes seções. Revisão da literatura, com apresentação das características da ferramenta de desenvolvimento APP Inventor e sua utilização em contextos educacionais. Metodologia, onde são apresentados os procedimentos metodológicos para realização da atividade proposta e investigação, utilizados na pesquisa. Resultados obtidos, onde são apresentados os aplicativos desenvolvidos pelos alunos participantes e suas funcionalidades. Por fim, estão as considerações finais e referências.

REVISÃO BIBLIOGRÁFICA

Nesta seção são apresentados os conceitos básicos da ferramenta APP Inventor e sua utilização no contexto educacional.

APP Inventor

O APP Inventor, objeto de estudo do presente trabalho, foi desenvolvido pelo Google e é atualmente mantido pelo MIT Center for Mobile Learning, foi inicialmente projetado para auxiliar o aprendizado de programação, mas tem sido amplamente utilizado em diversas aplicações, inclusive em contextos didático-pedagógicos variados.

O APP Inventor é um ambiente de programação visual online que possibilita a criação de aplicações para dispositivos móveis Android através de blocos de código, sem

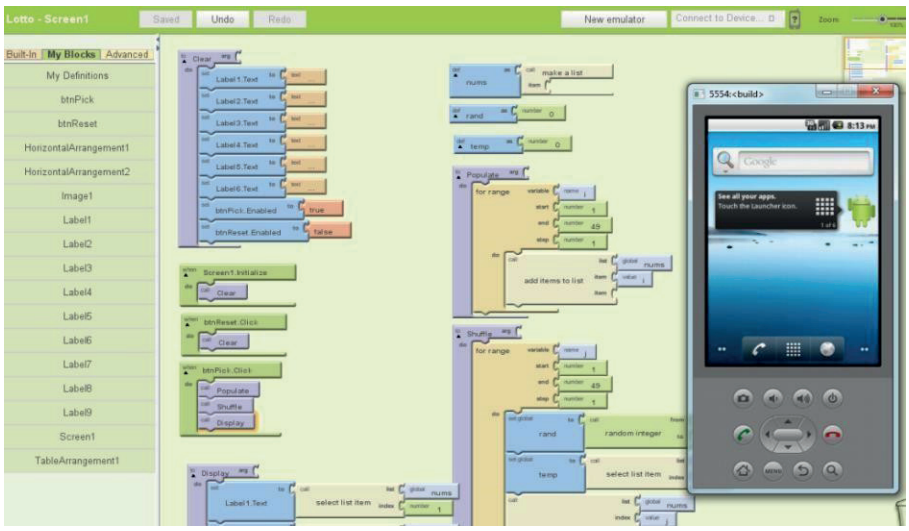
requerer, portanto, conhecimentos avançados em programação (MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2015). O ambiente de desenvolvimento do APP Inventor permite a criação de aplicativos, mesmo que o usuário tenha somente noções mínimas de programação de computadores. Por fornecer uma interface gráfica que permite os desenvolvedores utilizarem a técnica drag and drop (arrastar e soltar) para os objetos visuais.

Para utilizar o APP Inventor, não é necessário instalá-lo em seu computador, o aplicativo funciona online diretamente em uma janela de seu navegador de internet padrão. Para utilizá-lo, é necessário possuir uma conta de usuário da Google. (APP INVENTOR, 2015). Os aplicativos desenvolvidos no APP Inventor para telefones com o sistema operacional Android utilizam um navegador da web, bastando ter uma conta do Google GMAIL para ter o acesso à plataforma. Os projetos desenvolvidos no APP Inventor são armazenados em servidores web do APP Inventor.

As aplicações são construídas pelo APP Inventor Designer, onde o usuário escolhe quais os componentes serão utilizados em sua aplicação. No APP Inventor Blocks Editor, o usuário monta os blocos de programa, onde estes especificam como os componentes irão se comportar. Os blocos são montados encaixando as peças como se fosse um brinquedo de quebra-cabeça.

Quando o aplicativo (de extensão apk) estiver concluído, o desenvolvedor empacota-o, resultando numa aplicação pronta para ser instalada no smartphone. A Figura 1 apresenta o APP Inventor onde o desenvolvedor projeta o seu aplicativo, emula e empacota o apk para o smartphone.

Figura 1. APP Inventor: Janela Blocks Editor. Exemplo de Aplicativo e Emulador em Execução.



O ambiente de desenvolvimento do APP Inventor roda em diversos sistemas operacionais como o Mac OSX, GNU/Linux e Windows, e uma enorme variedade de modelos de simulações para o smartphone contendo o Sistema Operacional Android. O APP Inventor foi

projetado com a missão de provocar mudanças positivas nas experiências introdutórias de programação. Permite que usuários sem conhecimento avançado em programação possam desenvolver aplicativos para a plataforma Android.

Um dos principais diferenciais do APP Inventor consiste em permitir ao usuário, mesmo que estando em um estágio inicial de aprendizado de lógica de programação, desenvolver aplicações interessantes e atraentes para dispositivos móveis Android. As aplicações podem incorporar serviços baseados na web, leitura de códigos de barra, interação com sensores de orientação e geolocalização, dentre diversas outras funcionalidades de maneira simplificada, o que não seria possível em linguagens de programação tradicionais, visto que, exigiria um avançado conhecimento em programação.

- A ferramenta APP Inventor possui uma interface amigável, permitindo assim que despontem algumas vantagens na sua utilização em processos educacionais (Smutny, 2011):
- Sem código. A linguagem baseada em blocos elimina a necessidade de linhas de código;
- Orientada a eventos. Uma aplicação construída com o APP Inventor é um conjunto de blocos que respondem a eventos. O modelo conceitual é baseado na ideia do que acontece quando um componente realiza uma determinada ação;
- Mitiga a possibilidade de cometer erros. Nem todos os blocos do APP Inventor são conectáveis. Quando um usuário tenta conectar blocos de tipos não compatíveis, o Blocks emite uma mensagem de aviso e não permite a ação, minimizando, assim, a possibilidade do usuário cometer erros;
- Menos abstração. Quando um componente é selecionado e arrastado para o Viewer, um objeto e os blocos de funções associadas a este são criados no Blocks. Embora isto limite a possibilidade de reuso de código e torne o programa um pouco maior, este é um benefício em potencial para usuários finais, uma vez que diminui a abstração envolvida no processo de criação do aplicativo.

O APP Inventor possui um componente chamado TinyDB que é um banco de dados, que permite armazenar dados que estarão disponíveis cada vez que o aplicativo for executado. TinyDB é um componente não-visível.

A utilização do APP Inventor em contextos educacionais

Diversas aplicações do APP Inventor em contextos educacionais são encontradas na literatura, sobretudo em experiências introdutórias de ensino de programação, tais como:

- Gestwicki e Ahmad (2011) defendem que o software é um “ambiente de programação que permite que os estudantes não precisem de conhecimento de Ciência da Computação para desenvolver aplicativos de smartphones e, assim, explorar conceitos fundamentais da ciência da computação”;
- Lake (2011) relata o componente de verão de um projeto em que visava abordar se o APP Inventor seria uma plataforma computacional adequada para os alunos. O projeto reuniu uma equipe composta por dois professores do ensino médio, os estudantes eram facilmente capazes de desenvolver aplicativos móveis complexos completamente por conta própria. No geral, a equipe encontrou no APP Inventor uma plataforma acessível e poderosa que pode muito bem apoiar disciplinas de nível introdutório na faculdade;
- Spertus et al., 2010 utilizaram o APP Inventor para criação de aplicativos para celulares em cursos introdutórios de informática, por permitir que os usuários criem aplicativos móveis que contenham as redes sociais, a conscientização de local, e serviços baseados na Web para o Android. Doze faculdades e universidades diferentes usaram o APP Inventor em ampla gama de cursos de ciência da computação no outono de 2009.
- Wolber (2010) utiliza o APP Inventor para ensinar aos estudantes universitários de um curso na University of San Francisco, com foco em demostrar como a linguagem facilita as interações com o mundo fora da sala de aula;
- Wolber et al. (2011) apresenta uma visão geral da linguagem e aborda conceitos da Ciência da Computação por meio de design de jogos e de aplicações gráficas 2D.
- Gomes e Melo (2013, p. 224) definem que: “A aplicação do APP Inventor em contextos educativos tem demonstrado o seu potencial didático-pedagógico, principalmente em cursos de programação voltado a iniciantes”.
- Em outras áreas do saber, também são encontradas experiências de ensino envolvendo o APP Inventor, tais como:
- Smutny (2011) que apresenta aplicações criadas no APP Inventor que permitem controlar, através de dispositivos móveis, robôs dos kits Lego Mindstorms, permitindo que os alunos aprendam conceitos fundamentais de robótica e de programação.
- Brás (2013) que apresenta no ensino de física, uma aplicação educacional que se apropria da funcionalidade do acelerômetro, permitindo aos alunos visualizar os vetores de velocidade e aceleração sobre uma partícula virtual mostrada na tela.

PROCEDIMENTOS METODOLÓGICOS

Nesta seção são apresentados os aspectos metodológicos utilizados na condução do estudo.

O Desafio de Aplicativos do Instituto Federal de Goiás campus Luziânia teve como objetivo fomentar dentro do IFG a produção e a promoção de novos aplicativos, com a exploração das potencialidades do APP Inventor, estimulando a criatividade e a participação dos alunos em trabalhos em grupo, ampliando também os conceitos aplicados ao técnico em informática, colocando em prática o que é estudado e abordado dentro das salas de aula, aplicando conhecimentos básicos de programação, estimulando a criatividade e aceitação de novos desafios, explorando trabalho em equipe e competitividade.

Para que um aplicativo fosse elegível a participar da atividade, ele deveria ser um aplicativo (mobile) em versões final. Participaram alunos matriculados nas disciplinas de programação para web do curso técnico em informática para internet do Instituto Federal de Goiás campus Luziânia. Os aplicativos submetidos deveriam: concordar com regulamento e regras estabelecidas no regulamento; deter os direitos de uso de todos os recursos criado pelos participantes; executar corretamente na plataforma indicada; as equipes não poderiam ultrapassar 3 (três) integrantes; as equipe deveriam utilizar, obrigatoriamente, o APPInventor para o desenvolvimento do aplicativo. Quanto à desclassificação ficou estabelecido que: quaisquer aplicativos que utilizassem obras de terceiros seriam desclassificados; não seriam aceitos aplicativos submetidos com teor pornográfico, racismo, preconceito, questões éticas ou morais, sexo explícito, violência, intimidação, incitação ao ódio, falsificação de identidade ou comportamento enganoso, informações pessoais e confidenciais sem autorização prévia, atividades ilegais, jogos de azar, produtos maliciosos e a não utilização do APP Inventor.

Os aplicativos submetidos considerados elegíveis foram avaliados por uma banca avaliadora definida pela comissão organizadora. Os membros desta banca avaliadora foram formados por professores do Instituto Federal de Goiás campus Luziânia. A banca avaliadora analisou os aplicativos submetidos e atribuiu uma nota de 1 a 5, onde 1 significa “Muito fraco” e 5 significa “Excelente”, para cada um dos critérios listados: Inovação e Criatividade. Apresentação de soluções inovadoras e não convencionais que fundamentem o aplicativo. Qualidade de uso. Planejamento da interface de forma a propiciar a efetividade (eficácia e eficiência) e satisfação da interação. Projeto. Organização, qualidade da apresentação, profundidade e clareza da metodologia de execução do aplicativo. Descrição das etapas necessárias para o desenvolvimento e os elementos utilizados na estrutura.

Para cada aplicativo, foi calculada a média aritmética simples das notas atribuídas por cada avaliador em cada um dos critérios. A pontuação final para cada um dos aplicativos foi dada pela média aritmética simples das avaliações médias atribuídas pelos avaliadores aos

aplicativos. O intuito desta atividade não foi eleger o melhor aplicativo, e sim fomentar dentro do Instituto Federal de Goiás campus Luziânia a produção e a promoção de novos aplicativos.

Optou-se pelo uso de Tecnologias da Informação e Comunicação (TIC), por poderem ser aplicadas para diferentes finalidades, como fonte de informação, auxiliar no processo de construção de conhecimentos ou como meio para desenvolver autonomia pelo uso de softwares que possibilitem pensar e criar soluções. Foram classificados 2 (dois) aplicativos, pertencentes a 6 (seis) alunos da disciplina de programação para web I, do segundo ano do curso técnico em informática para internet, do ano de 2015. A atividade para criação de aplicativos foi estruturada de forma que os discentes participantes fossem sujeitos ativos em sua aprendizagem, tornando-se independentes na realização das pesquisas, ampliando seu conhecimento acerca das temáticas abordadas e compartilhando com os demais colegas o conhecimento adquirido.

A criação dos aplicativos no APP Inventor não necessita de conhecimento prévio sobre linguagem de programação, pois as funções dos elementos necessários para os aplicativos são pré-dispostos na forma de blocos lógicos, que podem ser justapostos para gerar as funções necessárias. As atividades foram estruturadas em 4 (quatro) etapas interligadas: etapa 1 – Pesquisa: os alunos devem elaborar pesquisas sobre um tema, culminando em uma síntese contendo os pontos chave acerca do objeto estudado; etapa 2 – Arquitetura: os temas pesquisados são aplicados na estruturação dos aplicativos, resultando em um software para execução; etapa 3 – Revisão: os alunos verificam possíveis falhas ou erros na execução do aplicativo, apontando onde e como o aplicativo pode ser corrigido ou adaptado; etapa 4 – Consolidação: os aplicativos são validados com relação à sua estruturação e funcionalidades.

RESULTADOS OBTIDOS

Nesta seção são apresentados os 2 (dois) aplicativos desenvolvidos no projeto:

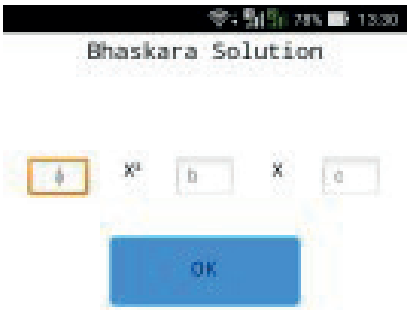
Bhaskara

Este aplicativo tem como objetivo ajudar as pessoas na resolução do cálculo de Bhaskara, dando a resposta rapidamente dos valores de X’ e X”, onde é necessário informar os valores e o aplicativo apresenta a resposta final, pois muitas pessoas têm dificuldades em cálculos assim, e o aplicativo pode ajudá-las na confirmação dos valores.

Para se utilizar o aplicativo, é necessário apenas que a pessoa informe os valores de A, B e C, e apertar em OK. Rapidamente o aplicativo irá informar os valores de X’ e X’’. O aplicativo possui uma interface simples onde é apresentado ao usuário apenas os

locais onde ele irá informar os valores necessários para o calculo de Bhaskara e assim que for apertado ok, serão apresentados os valores encontrados, conforme ilustra a Figura 2.

Figura 2. Aplicativo Bhaskara



Calc

O aplicativo consiste em um modelo de calculadora que realiza todas as funcionalidades padrão, como multiplicar, dividir, somar, e subtrair, e, além disso, calcular a raiz quadrada e potência. O aplicativo foi criado através de iniciativa própria, devido ao grande contato com a criação de programas que tivessem as funcionalidades de calculadora.

O aplicativo possui um teclado, enumerado de 0 a 9, e botões com o ponto, sinal de igual, soma, subtração, multiplicação, divisão, raiz quadrada, potência e o botão de limpar. Seu funcionamento se assemelha ao de uma calculadora normal, são selecionados os números e as operações que se deseja realizar com os devidos números. A estrutura de organização do aplicativo é simples e objetiva, apresentando todas as suas funcionalidades, conforme ilustra a Figura 3.

Figura 3. Aplicativo Calc



CONSIDERAÇÕES FINAIS

O APP Inventor, ferramenta utilizada no desenvolvimento do projeto utiliza programação orientada a eventos para celulares com sistemas operacionais Android e é de fácil utilização,

apresenta inúmeras funcionalidades e já vem sendo empregada em salas de aulas para estimular o aprendizado no desenvolvimento de aplicativos para dispositivos móveis. Ficou evidenciado na realização deste projeto, o potencial que o desenvolvimento de aplicativos pode representar no processo de ensino e aprendizagem. Mediante a qualidade dos aplicativos produzidos, percebeu-se a preocupação de alguns alunos em apresentar não apenas o aplicativo funcionando, mas também apresentar o melhor resultado possível utilizando o tempo e os recursos disponíveis.

Outro ponto relevante observado é o compartilhamento de ideias entre os membros do grupo e o desenvolvimento da autonomia dos alunos com relação à busca por informações para auxiliar no processo de elaboração dos aplicativos, resultando, consequentemente, no domínio acerca dos temas abordados na atividade. O projeto permitiu a utilização do conhecimento adquirido durante o curso técnico em informática para internet na construção de uma aplicação real, com utilidade prática. Os resultados obtidos na finalização deste projeto não encerram as pesquisas relacionadas aos aplicativos desenvolvidos, abrindo oportunidade para futuras melhorias nos aplicativos. Como sugestão para trabalhos futuros sugere-se ampliação do projeto em alunos de diferentes disciplinas, cursos, campus ou instituições e o uso de mais ferramentas livres como o APP Inventor.

REFERÊNCIAS

1. App Inventor. Site do APP Inventor. Disponível em: <http://beta.appinventor.mit.edu>. Acesso em 10 outubro. 2015.

2. Brás, C. M. Plataforma Android: Os Sensores e a Física. Anais do I Encontro Internacional da Casa das Ciências, 2013.

3. Cox, K. K. Informática na educação escolar. Campinas: Autores Associados, Coleção Polêmicas do nosso tempo, 2003.

4. Gestwicki, P., & Ahmad, K. APP Inventor for Android with Studio-Based Learning. Journal of Computing Sciences in Colleges , pp. 55-63, 2011.

5. Gomes, T. C.S.; Melo, J. C. B. APP Inventor for Android: Uma Nova Possibilidade para o Ensino de Lógica de Programação. Anais do II Congresso Brasileiro de Informática na Educação, p.620-629, 2013.

6. Lake, P., Lanerolle, T., Limardo, N., Morelli, R., & Uche. Can Android APP Inventor Bring Computational Thinking for K-12? Technical Symposium On Computer Science (SIGCSE), 2011.

7. Massachusetts Institute of Technology. Massachusetts institute of technology. Fonte: MIT APP Inventor: appinventor.mit.edu/ Acesso em 10 de novembro. 2015.

8. Smutny, P. Visual programming for smartphones. Proceedings of 12th International Carpathian Control Conference - ICCC, p. 358-361, 2011.

9. Spertus, E., Chang, M. L., Gestwicki, P., & Wolber, D. Novel approaches to CS 0 with app inventor for android. Proceedings of the 41st ACM Technical Symposium on Computer Science Education, 2010.

10. Wolber, D. APP Inventor and Real-World Motivation. Fonte: http://cs.usfca.edu/~wolber/appinventor/wolberSigCse11.pdf, 2010.

11. Wolber, D., Abelson, H., Spertus, E., & Looney, L. APP Inventor: Create your own Android Apps. Sebastopol: O'Reilly Media, 2011.

05

“ Estudo de Caso de Semiótica – O caso de “ Over The Garden Wall”

- Agnaldo **Gomes**
  UNIFANOR
- Hélio **Parente**
  UNIFANOR

doi 10.37885/200700638

RESUMO

O artigo a seguir tem como objetivo a análise da obra de animação gráfica " *Over The Garden Wall*" sob o ponto de vista da Semiótica Aplicada, levando em conta os conceitos de gramática especulativa e fenomenologia, elaborados por Charles Peirce, relacionando-os com a obra acima. Analisaremos quais elementos presentes na obra se caracterizam como signos semióticos e como estes auxiliam no complemento da narrativa. Nosso principal objeto de estudo será o plano de fundo e suas cores durante o decorrer da obra e como o mesmo aprofundam a narrativa segundo a semiótica de Peirce.

Palavras-chave: Over The Garden Wall, Semiótica, Narrativa, Plano de Fundo, Cor, Estações.

INTRODUÇÃO

Ao assistirmos a obra de Mchale (2014), notamos que se faz necessário uma maior atenção à mensagens e elementos presentes na obra assim como uma análise mais profunda dos mesmos, pois, sem isto a obra perderia grande parte de seu sentido, não sendo possível entende-la em todos os níveis propostos. Sendo assim, percebemos que os estudos sobre a análise de imagem abordados pela Semiótica se fazem necessário, trazendo assim estes estudos como a metodologia utilizada para validar os argumentos apresentados neste artigo.

Um exemplo de um desses elementos na obra é a presença de signos inspirados em períodos históricos distintos, interagindo uns com os outros, tornando-se assim difícil datar o tempo exato dos acontecimentos experienciados pelos personagens. Esse anacronismo utilizado como ferramenta narrativa dá um maior mistério ao enredo da obra. No entanto, essa ferramenta não pode ser classificada como um ruído, pois não há a distorção da interpretação do signo e sim a criação de uma nova camada interpretativa que irá dar maior qualidade à obra. Para validar este argumento trazemos a seguinte definição de ruído:

> "O ruído se dá por deficiência de emissão dos sinais do código, de recepção desses mesmo sinais, sejam decorrentes do canal utilizado ou da situação de emissão ou de recepção" (NIEMEYER, 2003, p. 33).

Um outro exemplo é a relação de signos com o folclore presente em diversas obras literárias infantis do século 19, fazendo também referências à grandes obras da literatura como A Divina Comédia de Dante Alighieri. Esta referência ao livro de Dante se faz melhor explícita ao analisarmos a possível interpretação da obra de Mchale como uma metáfora a um momento entre vida e morte, elemento presente em ambas as obras.

Estes signos tratam-se então de elementos que muitas vezes são, não apenas complementares, mas essenciais para a narrativa da obra.

Neste artigo, analisaremos semioticamente como se representa a tensão e o medo na obra através do plano de fundo no qual os personagens se encontram. O plano de fundo nesta obra, recebe atenção especial da direção de arte pois suas cores e sombras fazem uma ligação tanto à estação do ano em que a obra se passa, como aos sentimentos dos personagens principais.

Dividiremos o artigo em três tópicos essenciais, uma macroanálise, uma microanálise e uma análise semiótica. Resolvemos fazer esta divisão para que o texto se torne mais dinâmico e para que o leitor seja primeiro expostos aos signos de tal forma que estão presentes na obra, em seguida para que sejam oferecidos à interpretação dos autores e finalmente, no último tópico, analisaremos o que foi posto à amostra sob um olhar semiótico, explicando as conexões e o tipo de signo utilizado.

MICRO ANÁLISE

Primeiramente, analisaremos esta tensão ocorrida durante um único episódio, caracterizando uma microanálise. Para isto escolhemos o episódio número 02 – "*Hard Times at The Husking Bee"*. Para melhor dividir esta análise, a faremos da seguinte forma: trabalharemos com o início, explicando parte da história, o contexto em que se encontram os personagens, suas emoções e *frames* tirados da obra.

No início deste episódio os protagonistas se encontram vagando pela floresta. Aqui, ainda no começo da série e do capítulo, os protagonistas estão despreocupados, somos até mesmo mostrados a cenas pacatas e pacíficas de uma vida no campo enquanto os personagens conversam entre si.

Neste período o plano de fundo apresenta cores bem vivas e vibrantes, em geral tons de amarelo e laranja que fazem referência ao outono, visto que a obra em si se passa durante o Halloween, que ocorre no mês de outubro, onde, geralmente, nos países nortenhos, temos a estação de outono. Podemos ver a relação entre cores na paleta de cores ao lado.

Imagens 01 e 02: *frames* retirados da obra



O vermelho, laranja e o amarelo são cores recorrentes em toda a obra, justamente, como mencionado antes, pela estação do ano em que se passa. No entanto, notamos durante

esta análise, que estas cores vão gradativamente empalidecendo, tornando-se mais frias ou até mesmo, tornando-se quase preto e branco.

Isto ocorre por dois motivos, o primeiro é que essa transição serve para indicar o estado emocional dos personagens principais, ou seja, quanto mais frias e pálidas as cores do plano de fundo, mais tensão e medo estão presente naquele momento, isto geralmente ocorre quando Wirt e Greg, os principais, se encontram com criaturas sobrenaturais ou participam de algum conflito. O segundo motivo, que será mais abordado no próximo tópico, é que essa transição de cores ocorre gradativamente na obra como um todo, demonstrando assim a passagem do outono até o inverno.

Aqui, há também o elemento da sombra, que cerca o *frame* do filme como uma moldura e fica cada vez mais espessa à medida que temos o aumento de tensão. Como demonstrado nas imagens abaixo.

Imagens 03 e 04: *frames* retirados da obra



Com o decorrer do episódio, vemos que os personagens, Wirt e Greg, se encontram numa vila. Esta vila, no início, os causa uma sensação de alívio, pois notam que não estão mais sozinhos, porém, aos poucos percebem que não estão em uma vila normal, seus habitantes, como descobrem são mortos-vivos e os querem ter na vila para sempre.

Assim, ao se encontrarem nesta situação de desespero, podemos ver muito claramente nesse episódio como as cores do plano de fundo se alteram para enfatizar o sentimento das personagens.

Imagens 5 à 7: *frames* retirados da obra

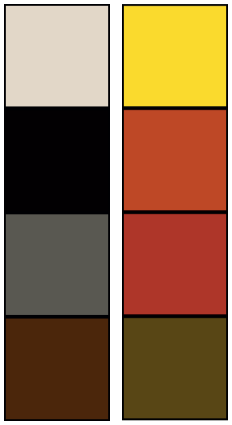






É importante salientar que o ambiente em que se encontram é o mesmo e que, segundo a história, não passam mais do que algumas horas na curiosa vila. No entanto, a mudança climática é drástica, a cores, como vemos na comparação abaixo se tornam muito mais escuras e o contraste diminui bastante, isso, é um efeito proposital da direção de arte da obra, que se utiliza dessa variação como técnica para enfatizar o medo e o estranhamento.

Imagem 08: Contraste entre paletas de cores.



Fonte: acervo dos autores.

Encontro Com Entidades Sobrenaturais

Durante a obra, Wirt e Greg, se encontram com inúmeras criaturas sobrenaturais, algumas delas são companheiras e os ajudam em sua jornada, no entanto algumas são vilões de certo episódios ou até mesmo da obra toda, como é o caso da criatura chamada " A Besta". Durante o encontro com esses seres, é esperado que qualquer personagem que não está acostumado a ver coisas fantásticas sinta certo estranhamento, desespero ou até mesmo medo e é, exatamente isto que ocorre com os personagens principais.

Sempre que têm estes encontros, as cores ficam mais pálidas e se dá ênfase no azul, preto e cinza, cores geralmente associadas também ao inverno.

Isto ocorre tanto para mostrar a transição das estações, como abordaremos num tópico mais a frente, como para causar um bruto contraste entre os momentos com estes vilões e momentos sem estes vilões.

Imagens 09 e 10: *Frames* de vilões, retirados da obra.





Vemos aqui que a moldura da sombra, que foi citada anteriormente, se mescla com o plano de fundo, e as vezes toma para si suas cores, deixando tudo o que está atrás do personagem em destaque, desfocado e negro aos arredores, isto causa um contraste que nos prende a atenção no personagem. As cores do fundo neste momento são em geral cinza ou pretas, com ocasionais tinturas de azul, justamente para representar o medo e o conflito.

MACRO ANÁLISE

Durante os episódios da obra as cores do outono, presentes no início da série de forma clara e vibrante, vão aos poucos perdendo as suas cores e força, dando lugar à cores características do inverno, como o azul e o cinza.

Isto se dá sutilmente ao logo de toda a obra, tanto para demonstrar o tempo que se passa como também, para se utilizar de uma técnica narrativa na qual o próprio ambiente em que se encontram os personagens refletem os sentimentos internos.

Por isso, é notável que na hora, com o decorrer da história, vemos a mesma vai tomando um rumo mais sombrio e frio, que causa mais medo que os episódios iniciais e vemos que o plano de fundo acompanha essa narrativa.

Imagem 11: *Frame* episódio 02, retirado da obra



Imagem 12: *Frame* episódio 06, retirado da obra



Imagem 13: *Frame* episódio 09 (penúltimo), retirado da obra



No último episódio (número 10), quando ocorre o clímax de toda a série, o medo e o desespero chegam aos níveis máximos enquanto os personagens lutam contra a ameaça de morte e contra o vilão principal, " A Besta".

Aqui, Wirt se encontra completamente desamparado, enquanto luta para salvar a vida do irmão, Greg, que estão quase por se esvair por completo, a cor some, a borda de sombras se engloba quase todo o frame, intensificando o medo e o desespero. Aqui também, neste episódio e no penúltimo, a chegada do inverno, que está sempre atrelado ao frio e ao medo, simbolizando a desistência do personagem Wirt, aceitando seu cruel destino como dita a história, apenas para acordar e ter seu irmão quase morto.

Imagem 14: *Frame* episódio 10, retirado da obra



Comparativo das paletas de cores ao logo da série.

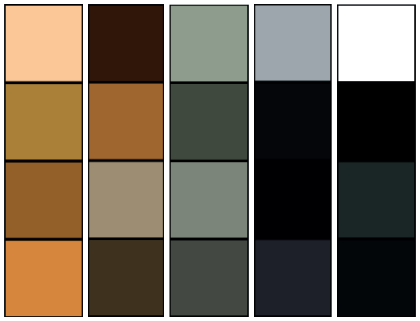


Imagem 15: *Frame* episódio 10, retirado da obra



REFERENCIAL TEÓRICO

Como referencial teórico deste artigo, nos utilizamos dos estudos da Semiótica realizados por Peirce, nos baseando também no livro " Elementos de Semiótica Aplicados ao Design" de Lucy Niemeyer. Durante o desenvolvimento da análise de signos presentes na obra nos utilizaremos de diversos conceitos presentes no estudo semiótico como a Primeiridade, Secundidade, Terceiridade, relações sígnicas dentre outros.

Quando falamos de signo entendemos isto como algo presente no universo que esta presentando um outro algo para alguém, este alguém sendo o sujeito que produz mensagens, sendo chamado de Interpretador. No entanto, ainda que represente algo o signo nunca se

tornará aquilo em si, caso isso aconteça perdemos o valor de signo. Sobre isso tiramos o seguinte trecho:

> "Por mais extenso que seja o inventário sígnico, ele jamais dará conta de seu Objeto de origem, pois caso tal se dê, ele perde a característica de ser signo e passa a ser o Objeto em si. Neste quadro teórico, é Objeto aquela coisa que passa pela esfera do conhecimento (Pinto, 1995, p.37 apud Niemeyer, 2003, p.37).

Aqui é necessário também nos atentarmos às relações sígnicas, especificamente à primeira tricotomia de Peirce, na qual estudamos a decomposição da estruturação do signo, também chamado de representâmen. Aqui dividimos em três os aspectos do representâmen, sendo estes o Qualisigno, que diz respeitos a características do signo que são mais comuns, não o particularizam, O Sinsigno, que seria a contrapartida ao Qualisigno, visto que diz respeito às características que o individualizam e, por último, o Legisigno, que diz respeito à convenções e leis que se apresentam no signo. Assim define Niemeyer:

> "O Qualisigno (uma qualidade que é signo) é o aspecto do Representâmen que diz respeitos às suas características que menos o particularizam, como as cores, os materiais, a textura, o acabamento. O Sinsigno é o aspecto do signo que já o particulariza e o individualiza como ocorrência: sua forma, suas dimensões. O Legisigno é como as conversões e as regras, os padrões se manifestam no Representâmen: as aplicações de perspectiva, o atendimento a normas. " (Niemeyer, 2003, p. 39)

Em seguida é preciso também definir as categorias da experiência sígnica, entendo essa experiência como a relação entre o signo e o Interpretador, gerando um Interpretante (aquilo que um signo pode gerar na mente de um sujeito), pois esta relação será abordada no desenvolvimento do artigo. Essas categorias são divididas em três momentos: a Primeiridade, a secundidade e a Terceiridade.

A Primeiridade seria o caráter que define a primeira característica percebida, o que se sobressai no signo fazendo-o mais próximo de seu objeto, aquilo cuja mensagem absorvida é a mais ampla. A secundidade são as características que torna o signo específico e a Terceiridade são as características presentes no signo as quais podemos generalizar e formar leis a partir. Assim define Niemeyer sobre as categorias:

> "Um momento primeiro, ou Primeiridade, em que predomina o caráter qualitativo, pré- reflexivo, sensível, signo em que o Objeto faz parecer estar no Representâmen, cujo Interpretante é o mais amplo possível; Peirce chama secundidade à categoria da experiência, da manifestação específica. Nela, a relação de causa acidental, fortuita, experimentada, constrói uma dimensão segunda, que se apoia na primeira; o âmbito terceiro, ou Terceiridade, é o lugar da regra, da lei, da convenção, da ciência, da previsão, do controle. " (Niemeyer, 2003, p.45)

Ao utilizar a Lógica Crítica de Peirce, devemos ter em mente a definição do raciocínio em torno de análises empíricas de qualquer signo. O raciocínio é o conhecimento que exige provas e demonstrações e se realiza igualmente por estas. São várias ligações intelectuais que se dão origem de interações com o ambiente em volta feitas rudimentarmente e cotidianamente, sem uma ciência em específico, fazendo ligações internas destes atos intelectuais. O raciocínio, segundo a Lógica de Peirce, está separado em três aspectos: dedução, indução e abdução.

A dedução parte do pressuposto de um conhecimento, uma verdade já conhecida pelo indivíduo, adquirido em uma demonstração anterior. Podemos afirmar que a dedução vai de uma verdade coletiva para uma verdade do indivíduo. O ponto inicial da dedução se dá em uma teoria verdadeira ou idéia verdadeira. A indução traça o caminho contrário, ela vem de uma idéia individual, que busca elementos de uma lei geral, e o torna coletivo. A abdução tem semelhanças com a indução, porém necessita de comprovações de que os elementos da lei geral são verdadeiras. Para Peirce, o raciocínio abdutivo tem como catalisador um *insight*, que apesar de necessitar de um conhecimento prévio (semelhante a dedução), a abdução tende a gerar novos tipos de pensamentos, fugindo do "óbvio".

ANÁLISE SEMIÓTICA

Análise Da Paleta De Cores Em Relação Às Estações

Agora que fomos devidamente introduzidos ao objeto de estudo e ao referencial teórico, cabe a nós analisar como estes significados se formam em nossa mente, como essa técnica narrativa de alterar o plano de fundo e suas cores nos levar a fazer analogias e tirar conclusões, como e quais são os signos que se formam.

Podemos considerar como *Ícone* as cores presentes na transição mencionada. Pois, quando a representação de um objeto é feita por meio de analogias temos a presença de um ícone, então quais seriam as analogias presentes na obra¿ Os ícones seriam as paletas de cores frias e quentes, ambas irão representar a estação a qual dizem respeito, inverno para as cores frias e outono para as cores quentes, esta analogia se dá quando comparamos fotos ou imagens que representam as estações mencionadas, nas quais, estas cores estão sempre presentes. Como comparamos na imagem abaixo:

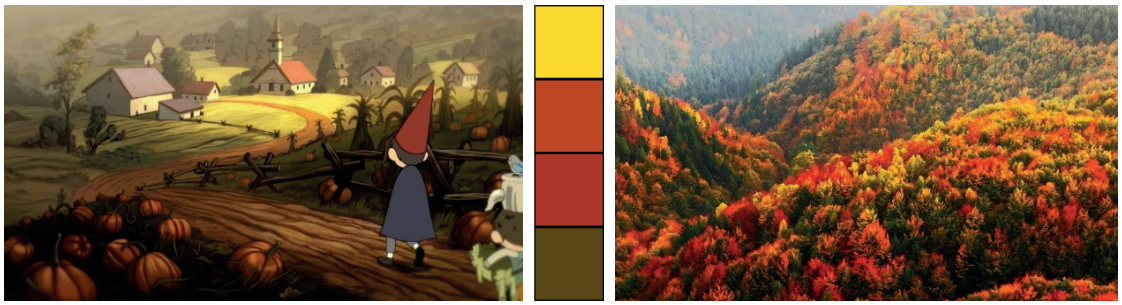
Imagens 15 e 17: 17: foto espaço no período de inverno



Fonte: https://images.unsplash.com/photo-1510798831971-661eb04b3739?ixlib=rb-0.3.5&ixid=eyJhcHBfaWQiOjEyMDd9&s=9f96bdaf999d3644b1157b27d214517a&w=1000&q=8 0

Podemos perceber que as cores escolhidas na animação se assemelham muito à cores vistas na estação invernal, o mesmo ocorre com as cores escolhidas para a estação do outono.

Imagens 02 e 18: 18: Foto da estação outono.



Fonte: https://d1w9csuen3k837.cloudfront.net/Pictures/1280x720/6/3/3/108633_autumn_forest.jpg

Assim, o processo de formação do *Ícone* se dá através da analogia entre as cores das estações as que estão presentes no plano de fundo.

Porém o próprio uso e escolha dessas cores são uma convenção da sociedade que atribui as cores azul, cinza e branco ao inverno e as cores amarelo, laranja e vermelho ao outono. Este tipo de processo de estabelecimento de relação caracteriza um *Símbolo*, aonde o representâmen é percebido no signo através de um processo de convenção social, ou seja, entende-se e aceita que estas cores remetem às estações por meio de uma convenção social. No entanto por se tratar de um caráter mais icônico podemos dizer que a relação entre as cores e as estações estão presentes num *Símbolo Icônico.*

Temos ainda um procedimento de Índice, mais especificamente de um Índice de Identificação, pois sua origem pode ser traçada. Um Índice se forma quando há uma relação de causalidade entre o signo e o representâmen, por tanto, de que forma isto está presente na paleta de cores¿

Como mencionado anteriormente, temos uma mudança de uma paleta quente para uma paleta fria gradativamente em toda a obra. Esta mudança de temperatura é causada pela narrativa presente na obra que o inverno e coisas mais sombrias e amedrontadoras

estão por vir, por tanto o *Índice* é a mudança de cor, sua causa a mudança de estação e humor das personagens.

Análise Da Paleta De Cores Em Relação Ao Encontro Com O Sobrenatural

Há ainda um outro signo que analisaremos semioticamente, que é a mudança para uma paleta de cores mais fria toda vez que os personagens encontram uma entidade sobrenatural que os causa medo.

O fato do medo ser associado a cores frias e escuras constitui uma forma de convenção social, por tanto a relação entre o representâmen, no caso, o medo ou o sobrenatural, e o signo, as cores, se dá por meio de *Símbolo*, mais especificamente um *Símbolo Índicial*.

Neste caso o processo de *Índice* também está presente, pois um índice é causado a partir de algo, e serve justamente para representar esta origem que pode ser conhecida (*Índice de Identificação)* ou não (*Índice de Indicação)*. No caso que analisamos temos a representação por meio de um *Índice de Identificação*, pois, está claro na narrativa, que as cores só se escurecem e esfriam no momento em que as personagens se encontram com medo, sendo então o medo a origem da causa.

Dividimos esta analise semiótica por tópicos e dividimos os signos por categorias separadas apenas em prol da didática da explicação, pois, na verdade, estes processos ocorrem de forma simultânea e se interseccionam, invés de permanecerem em paralelo.

CONCLUSÃO

Concluímos que uma análise mais profunda dos elementos estéticos do filme se faz necessário para que possamos entender a obra em todo os níveis de profundidade propostos. Os conhecimentos oriundos da semiótica nos ajudaram a entender os processos que nos levam a atribuir significados aos mais diversos objetos e assim, analisar mais profundamente a obra tendo como base um material teórico e científica.

REFERÊNCIAS

1. NIEMEYER, Lucy. **Elementos de semiótica aplicados ao design** . Rio de Janeiro: 2AB, 2003.
2. NIEMEYER, Lucy; HERMIDA, Stella. **Análise Semiótica da Comunicação de Marcas: O Caso Melissa** . 2011.
3. LENA FARIAS, Priscila. **O conceito de diagrama na semiótica de Charles S. Peirce** . Rio de Janeiro, 2008.
4. ROMANINI, Vinicius. **Design como comunicação: uma abordagem semiótica** . São Paulo.

06

“*Framework* de gerenciamento ágil de projetos do grupo de pesquisa em engenharia de software do IFPB

- Alexandre Dias Sales de **Morais**
  IFPB
- Carlos Diego Quirino **Lima**
  IFPB
- Helder Jerônimo Leite. **Rangel**
  IFPB
- Heremita Brasileiro **Lira**
  IFPB
- Nadja da Nóbrega **Rodrigues**
  IFPB
- Samyra Lara Ferreira de. **Almeida**
  IFPB
- Tiago Cesário **Barbosa**
  IFPB

doi 10.37885/201001809

RESUMO

As atividades realizadas no período retratam a criação de um *framework* ágil de gerenciamento de projetos baseado no *Scrum* que foi aplicado e validado no gerenciamento de um projeto do Grupo de Pesquisa de Engenharia de *Software* do IFPB (GPES-IFPB). Durante o período de aplicação do framework o GPES-IFPB estava desenvolvendo três projetos distintos, requerendo assim um gerenciamento e integração ágil das equipes envolvidas. Nesse contexto, foi criado um framework para o gerenciamento de projetos para o GPES utilizando as boas práticas pesquisadas pelo grupo. As atividades foram iniciadas com a criação de um *framework* para gerenciar as atividades de um dos projetos do GPES-IFPB, com o propósito de direcionar cada ação dentro do projeto, apoiando a equipe na aprendizagem e desenvolvimento de atividades com prazo, qualidade e em grupo. Após a criação do *framework*, o mesmo foi aplicado e avaliado em algumas *sprints,* tendo sido melhorado com a inclusão de boas práticas de gerenciamento ágil de projeto, e a mudança de ferramentas de gerenciamento e de comunicação, o que poderá ser observado nos resultados apresentados.

Palavras-chave: Engenharia de *Software*, Gerenciamento de Projetos, Metodologia Ágil, *Scrum*.

INTRODUÇÃO

Há no mercado inúmeras propostas de métodos, técnicas e ferramentas para o Gerenciamento de Projetos de Software (GPS), entretanto, um dos grandes problemas ou desafios é identificar e integrar as boas práticas do GPS em um *framework* de Gerenciamento Ágil de Projetos (FGAP) de forma a atender às necessidades de um cenário específico de produção de software. No do Grupo de Pesquisa de Engenharia de Software do IFPB[1](GPES-IFPB) estavam em andamento três projetos de pesquisa: Repositório de Práticas em Gerenciamento de Projeto de Software (RPGPS) (Morais et al., 2016); Repositório de Práticas em Processos de Desenvolvimento de Software (RPPDS) (Silva et al., 2016); e Ambiente de Boas Práticas Reconfiguráveis em Engenharia de Software (ABPRES) (Almeida et al., 2016), neste desenvolve-se uma plataforma web para disponibilizar todas as boas práticas identificadas nos dois primeiros projetos. De acordo com o *Project Management Institute* (PMI, 2016) o nível estratégico das organizações vêm percebendo os benefícios do uso de boas práticas/metodologias para gerir seus projetos.

Levando em consideração que as diversas práticas de gerenciamento de projeto de *software* podem influenciar o resultado de um projeto, caso aplicadas da forma correta, o desafio a ser enfrentado é “Qual a melhor maneira de gerenciar o projeto ABPRES do GPES-IFPB?” Portanto, a finalidade desse trabalho é definir, aplicar e avaliar um *framework* de gerenciamento ágil de projetos de *software*. Este trabalho foi desenvolvido de maneira iterativa e incremental, analisando como melhorar um *framework* adaptável para gerenciar os projetos de *software* do GPES-IFPB.

Os resultados até então observadas na aplicação do *framework* no projeto ABPRES, indicam que a definição e o uso do *framework* proposto foi aceito de modo positivo pelos envolvidos, tendo como principal resultado a conclusão de cinco *sprints* de acordo com o planejado.

OBJETIVO

Objetivo Geral

O objetivo geral é definir um *framework* de Gerenciamento Ágil de Projetos de *Software* para o GPES-IFPB, que será aplicado, analisado e refinado a cada sprint executada no Projeto ABPRES, adaptando o mesmo às necessidades do contexto do projeto.

Objetivos Específicos

1 "GPES-IFPB Grupo de Pesquisa, Desenvolvimento e Extensão em Engenharia de Software" http://dgp.cnpq.br/dgp/espelhogrupo/0686408453733553.

- Identificar as necessidades de gerenciamento dos projetos do GPES-IFPB;
- Identificar boas práticas e ferramentas de gerenciamento ágil de projetos de software que atendam a necessidade da equipe para gerenciar os projetos do GPES-IFPB;
- Definir o *framework* de gerenciamento ágil de projetos para o GPES-IFPB utilizando boas práticas de GPS;
- Aplicar o *framework* no gerenciamento do projeto ABPRES.
- Analisar comparativamente os resultados das *sprints* executadas.

METODOLOGIA

De acordo com Prodanov e Freitas (2013), considerando a metodologia de pesquisa científica, este projeto é de Natureza Aplicada, pois pruduz conhecimentos para aplicação prática dirigidos à solução de problemas específicos. O Método Científico utilizado é o Hipotético-Dedutivo, onde são formuladas hipóteses para expressar as dificuldades do problema, deduzindo consequências que devem ser testadas ou falseadas. Considerando o Objetivo do Estudo, podemos classificar este projeto de pesquisa como Exploratório, pois visa proporcionar maior familiaridade com o problema, explicitando-o ou construindo hipóteses sobre ele. A abordagem de pesquisa adotada é Quali-quantitativa e os principais Procedimentos Técnicos utilizados são: Pesquisa Bibliográfica, Levantamento de Dados, Estudo de Caso e Pesquisa Participante.

Do ponto de vista de execução, o projeto é desenvolvido de forma iterativa e incremental, com entregas parciais trimestrais, nos seguintes passos: Levantamento e Estratificação das Informações; Disponibilização das Informações no Repositório de Boas Práticas; Validação; Qualificação Continuada dos Pesquisadores; e, Elaboração de Relatórios e Artigos dos resultados para registro e publicação (Morais et al., 2016).

O *FRAMEWORK*

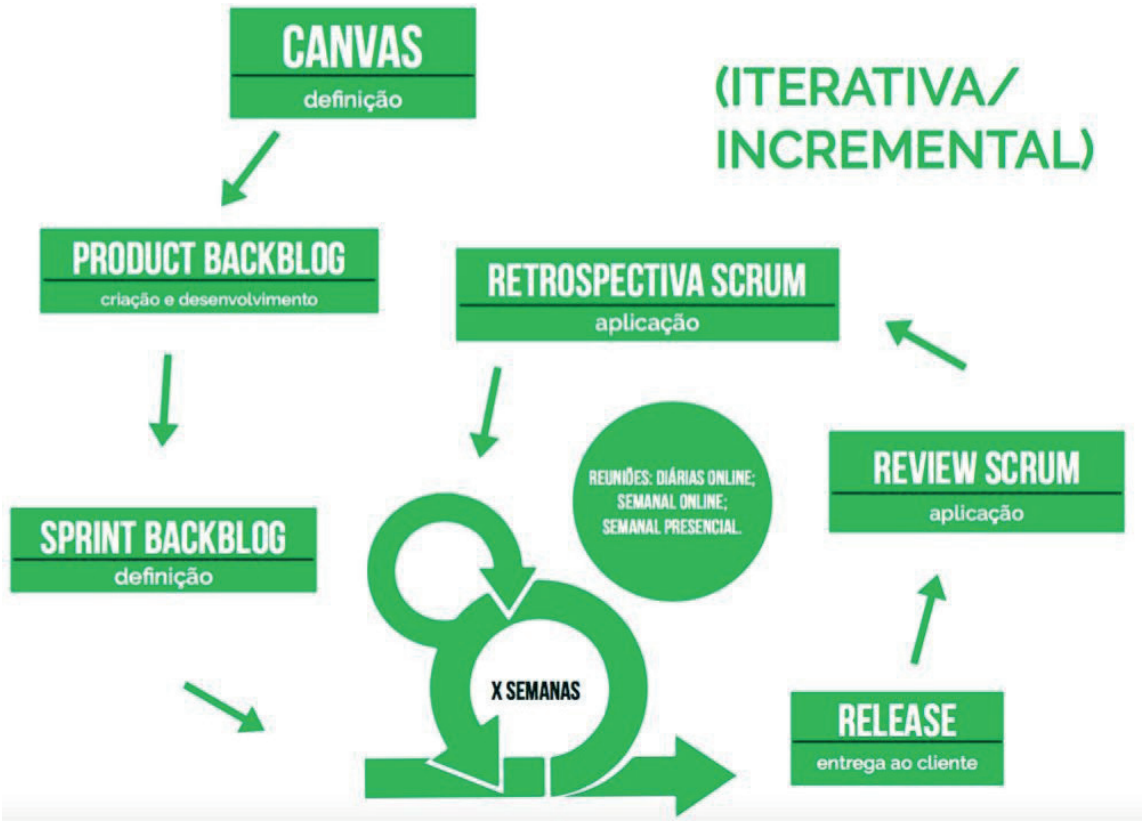
O *Framework* de Gerenciamento Ágil de Projetos (FGAP) foi desenvolvido utilizando como base o Scrum e as boas práticas identificadas nos projetos RPGPS e RPPDS, bem como considerando a participação dos pesquisadores no desenvolvimento no projeto do ABPRES. O *framework* foi dividido em etapas onde se define cada parte individualmente para facilitar o entendimento e a ordem das atividades para o gerenciamento ágil de projetos. O *framework* foi criado baseado no Scrum e nas metodologias ágeis *Dynamic Systems Development Method* e *Feature-Driven Development*.

As etapas do *framework* de gerenciamento do projeto proposto, mostrado na Figura 1, representa essa visão geral e ajuda o gerente de projetos atual, e os futuros, a organizar e seguir passo a passo o gerenciamento do projeto e de cada *sprint*.

As reuniões do projeto são feitas a partir de três distintos tipos de encontros, sendo eles: Reunião Diária Informal; Reunião Semanal Virtual e Reunião Semanal Presencial, onde uma vez durante a semana toda a equipe se reúne na instituição de ensino para dúvidas e decisões sobre o andamento do projeto. A reunião é cronometrada e possui pautas e atas digitais, essas mesmas são feitas anteriormente e previamente avisadas aos demais da equipe e são disponibilizadas no repositório do Google Drive.

Conforme apresentado na Figura 1, o FGAP se inicia com a criação de um *Project Model Canvas* (PMC), pois é necessário definir, entender e visualizar melhor o projeto para começar a gerenciá-lo. O *Product Backlog* é o segundo passo e também o mais longo. Nele são definidas as estórias de usuário, suas estimativas e respectivos graus de importância, através de reuniões presenciais, onde participam o Gerente de Projetos, o *Scrum Master*, a equipe e o *Product Owner*. Para criação do *Product Backlog* foi utilizada a ferramenta de gerenciamento de projetos Taiga. Na etapa da Sprint Backlog são feitas as seguintes atividades: estória de usuário, divisão das estórias de usuário em sprints e a transformação de cada estória do usuário em atividades.

Figura 1. Visão Geral do FGAP do GPES-IFPB.



Ao final de cada *sprint* é feita a *Review Scrum* que apresenta os resultados da *sprint* para o *Product Owner* em uma demonstração envolvendo todos os envolvidos. E por fim, a *Retrospectiva Scrum*, que fornece novamente à equipe a oportunidade de analisar a *sprint* finalizada e sugerir melhorias de acordo com as boas práticas do GPS. O FGAP ajuda o gerente de projetos e demais envolvidos a se organizar e seguir passo a passo o andamento do projeto e de cada *sprint*.

RESULTADOS

Primeira *Sprint*

Durante a primeira retrospectiva do projeto foi usada a boa prática de GPS chamada *WWW* (Campos, 2012), onde a equipe avalia o que deu certo (*What Went Well*) e o que deu errado (*What Went Wrong*), com o uso de post-it, identificando o porquê dos erros e registrando as lições aprendidas para não errar nas próximas *sprints*.

Em uma reunião presencial, todos os membros da equipe se reúnem e iniciam a aplicação da prática com o gerente de projetos comandando a reunião. Em um círculo, a cada membro da equipe, de forma individual, começou a escrever em post-it primeiramente todos os fatos positivos daquela *sprint*, com um tempo cronometrado e determinado pelo gerente de projetos previamente. Seguindo a lógica da prática, novamente o time possui um novo tempo, mas agora para escrever tudo que aconteceu de errado durante aquela *sprint*. Logo após o tempo acabar o gerente de projetos inicia a discussão, onde por vez, cada um começa a expor o que foi escrito, primeiramente sobre os fatos positivos e assim que cada um ler o seu post-it a equipe como um todo pode opinar sobre o que foi escrito, concordando ou discordando sobre o que foi dito. Quando todos terminam de apresentar suas opiniões sobre o que houve de positivo, o gerente começa a explicar como funcionará a próxima rodada da prática, onde agora cada um irá apresentar o que houve de errado naquela *sprint*, só que dessa vez a equipe deverá surgir com uma solução sobre o que deve ser feito para evitar esse mesmo erro na próxima *sprint*.

Na primeira *sprint* foram identificados pelo time através das boas práticas aplicadas os seguintes resultados: Bom trabalho em equipe; Reuniões presenciais produtivas; Realização de 90% (Taiga) das atividades. Algumas melhorias sugeridas para a segunda *sprint* foram: Proatividade dos envolvidos em atividades individuais; Reuniões objetivas e cronometradas com pautas; Novas estratégias para reunião diária online; Compreender melhor os requisitos do ambiente web do ABPRES e tirar dúvidas com o *Product* Owner.

Segunda *Sprint*

A segunda *sprint* teve uma metodologia de gerência muito similar a primeira sprint, sem grandes alterações. A principal diferença obtida pela equipe foi o melhor entendimento sobre o problema que deveria ser solucionado pela equipe, algo que acabou não ficando muito claro para todos na *sprint* anterior, afetando os resultados da *sprint*. Na segunda *sprint* os principais resultados observados foram: Melhor conhecimento sobre tempo de execução das tarefas; Com a realização de todas as atividades previstas na *Sprint*, a equipe conseguiu enxergar um progresso dentro do projeto; Melhor comunicação da equipe de desenvolvimento e independência com relação aos orientadores; Pensamento unificado dos requisitos do projeto ABPRES. Melhorias sugeridas para as próximas *sprints*: Decisões tomadas em reuniões online devem ser registradas em ata; Ampliar uso das ferramentas de GPS para melhorar a comunicação; Ampliar reuniões semanais presenciais.

Terceira *Sprint*

Durante a terceira *sprint* foi utilizada uma diferente ferramenta de gerenciamento, o Trello, devido a reclamações dos membros com a ferramenta utilizada nas *sprints* anteriores, os mesmos achavam complicada e queriam encontrar uma ferramenta mais intuitiva e que se adapta-se com as regras de nosso *framework*. Por conta do tempo, o Trello, que já era conhecido por todos, foi a opção escolhida, porém um estudo pela busca de uma nova ferramenta para a próxima *sprint* foi feita durante esse tempo.

Com os requisitos reformulados e a partir das *User Stories* criadas anteriormente iniciou-se a terceira *sprint*, onde foi obtido um protótipo funcional da ferramenta. Durante a retrospectiva foi aplicado um formulário com a equipe para medir a satisfação de cada membro individualmente com os resultados obtidos durante aquela release. Grande parte dos desafios durante a terceira *sprint* se deu devido ao fato de poucas pessoas na equipe não possuíam capacitação necessária para lidar com a tecnologia do desenvolvimento, fazendo com que muitas atividades acabassem ficando concentradas naqueles que tinham maior conhecimento.

Quarta *Sprint*

A *sprint* quatro começou após a semana de *review* e retrospectiva da *sprint* anterior. A reunião de planejamento se baseou bastante nas atividades não finalizadas da *sprint* três e nas adições as quais o *Product* Owner achou necessário. Durante a *sprint* tivemos algumas mudanças no gerenciamento, com a mesma duração de duas semanas, a sprint dessa vez contava com um novo gerente e dessa vez assume um diferente papel na *Sprint,*

o de *scrum master*, o qual iria trabalhar junto com o gerente para o desenvolvimento das atividades previstas, além de mais um especificador para adiantar as atividades.

Outra novidade durante esse tempo foi a substituição da ferramenta de gerenciamento de projetos Trello, onde dessa vez todas as atividades da *sprint* ficaram descritas dentro do Jira - apresentada na Figura 18 - como também o uso do Slack para as reuniões diárias. Sobre as atividades concluídas da *sprint*, tivemos o total de oito das doze atividades programas para serem entregues dentro da quarta *sprint*.

A reunião de *review* foi feita seguindo um roteiro de apresentação da *sprint*, definido pelo *scrum master*, com o objetivo de apresentar de forma concisa e prática os resultados da *sprint*, dando uma chance para que todos os participantes ficassem de alguma forma envolvidos na apresentação, onde cada membro da equipe falava sobre sua participação naquela entrega e ouvia um feedback de um dos *Product* Owner. Foram apresentados os gráficos gerado automaticamente pela ferramenta de gerenciamento utilizada das atividades previstas, para melhor entendimento e visualização de como foram realizadas as atividades do grupo durante o tempo determinado da *sprint*.

Quinta *Sprint*

Com algumas atividades pendentes da *sprint* anterior, uma migração da ferramenta de gerenciamento de projetos foi feita devido a quantidade de arquivos que estavam sendo colocados, fazendo com que o servidor ficasse com sobrecarga e deixando a equipe impossibilitada de utilizar a ferramenta por determinado tempo. O Asana, foi a nova escolhida a ferramenta também acabou substituindo o Slack, já que a mesma oferece um chat que acabava servindo para o uso da reunião diária.

Sobre a equipe, tivemos uma pequena diferença, onde mais um membro foi adicionado a equipe para lidar com as atividades de especificação previstas dentro do processo de desenvolvimento de software do projeto e um membro foi retirado das atividades da *sprint*. Diferente das diversas *sprints* anteriores, não tivemos reuniões presenciais, além da *review scrum*, e todo o trabalho foi feito de forma remota, tirando exceções de pequenas equipes - dupla de especificação, dupla de testes - que se reuniram a parte para desenvolver as atividades previstas para aquela *sprint*. As tarefas, surgidas a partir das *User Stories*, foram concluídas no dia determinado do fim da *sprint*, porém das quarenta atividades apenas vinte e quatro foram concluídas e dezesseis ficaram de fora, porém serão realocadas para a próxima *sprint*.

DISCUSSÃO

A partir da terceira *Sprint* foi realizado uma pesquisa entre a equipe com o objetivo de entender como a equipe estava lidando com cada entrega. A seguir os principais resultados de acordo com o formulário, *Feedback* da *Sprint* (FS) feito com os todos os membros da equipe mostrado nas Figuras 2, 3 e 4. Levando em conta que respostas com valores entre 1 - 5 foram ditas como insatisfatórias para a gerência e que respostas entre 6 - 10 são as satisfatórias, foi percebido, na Figura 2, a satisfação dos membros sobre as reuniões presenciais acabaram agradando a equipe, na qual 85.7% (soma das notas entre 6 e 10) achou que as reuniões presenciais ajudaram no desenvolvimento das atividades daquela *sprint*.

Um ponto importante que foi discutido é sobre a participação individual de cada membro na *sprint*, encontrada na Figura 3, já que 42.9% dos membros não sentiram que participaram o suficiente da *sprint*, levando a gerência a discutir uma realocação de papéis da equipe, fazendo com que cada membro possa contribuir com a *sprint* de acordo com suas habilidades.

Figura 2. Satisfação com as reuniões presenciais na Terceira *Sprint*

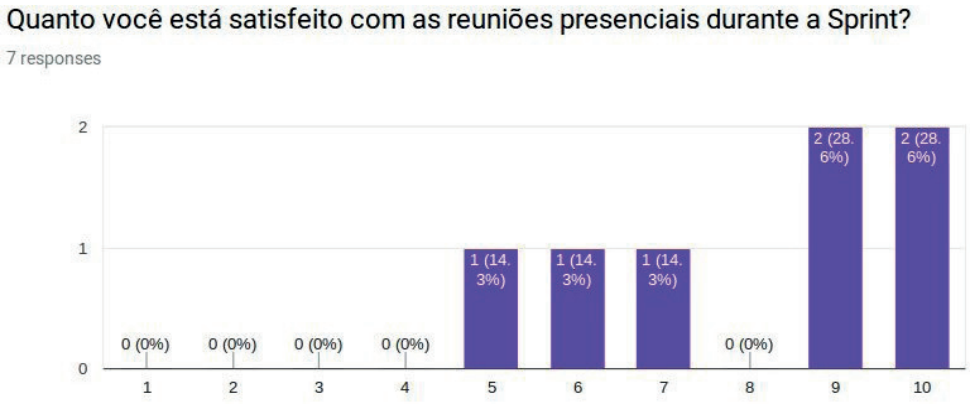
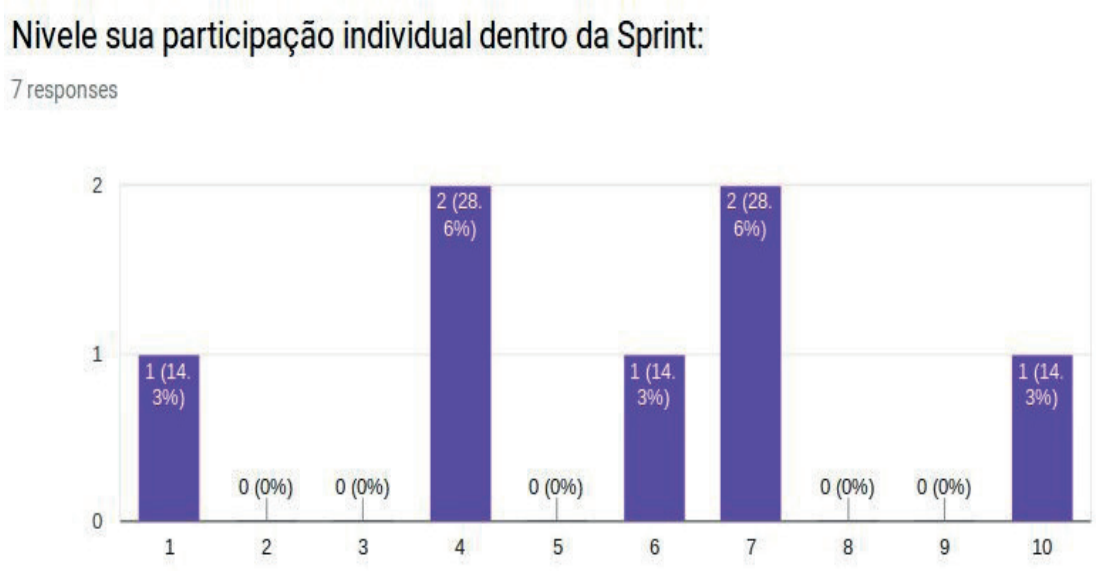
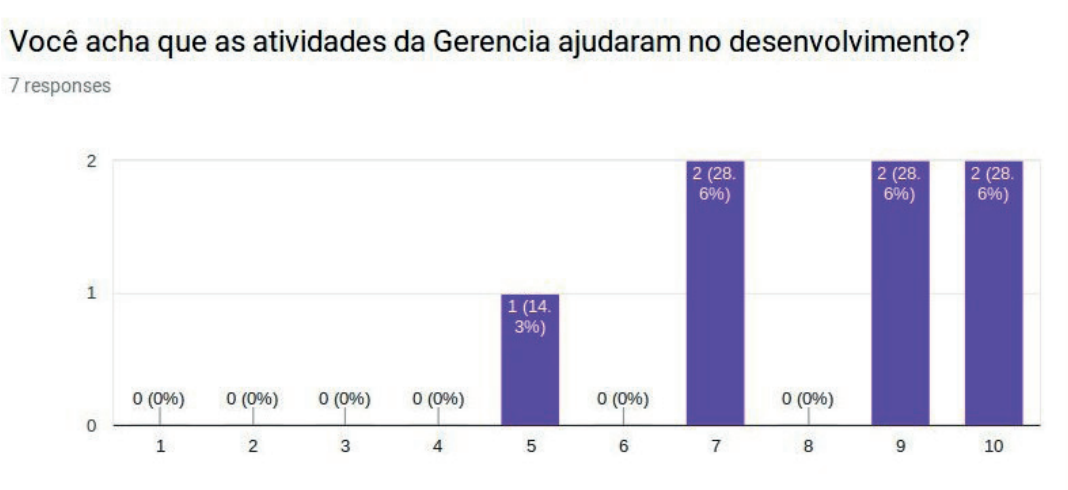


Figura 3. Participação individual na Terceira Sprint.



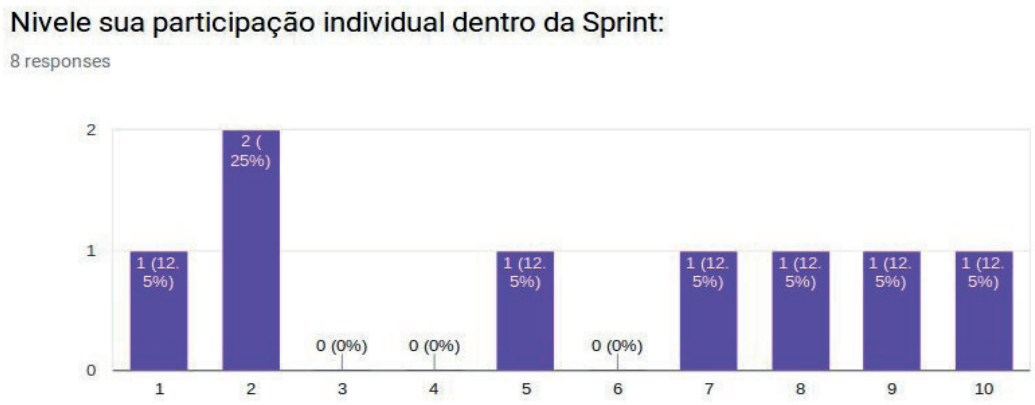
Outro ponto positivo dado através do FS pode ser encontrado na Figura 4, que durante a terceira *sprint*, 85.7% dos membros da equipe conseguiram assimilar a importância do papel da gerência e como ela através de reuniões, ferramentas de gerenciamento de projetos, distribuição e acompanhamento de atividades, foram essenciais para o cumprimento do que foi planejado para as duas semanas da *sprint*.

Figura 4. Percepção da importância do FGAP na Terceira *Sprint*.



O *Feedback* da *Sprint* (FS) foi aplicado a quarta *sprint* para avaliar em números como foi o andamento da *sprint* para os membros. Conforme mostra a Figura 5, durante essa *sprint*, 50% dos membros não ficaram satisfeitos com sua própria participação nas atividades da *sprint*, número que acabou diminuindo comparado com a *sprint* anterior.

Figura 5. Participação Individual na Quarta Sprint

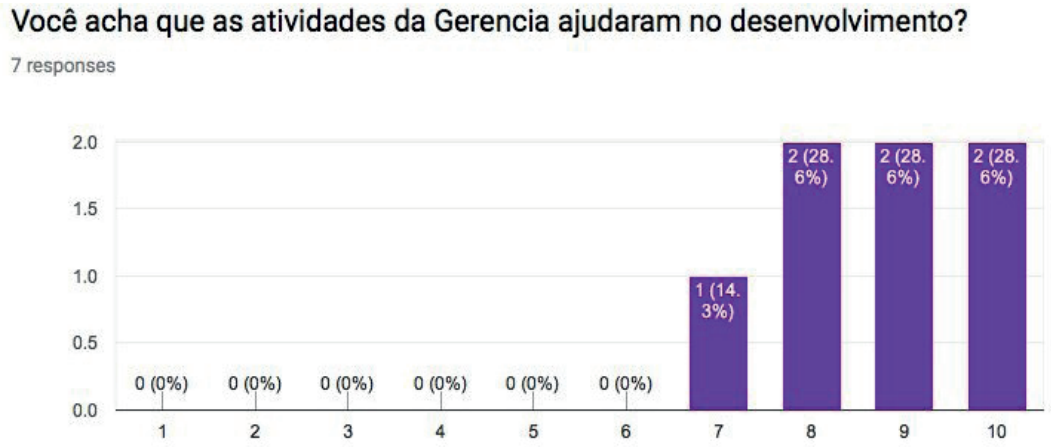


Algumas alternativas discutidas pelos membros durante a retrospectiva da quarta *sprint* foram: (1) Comunicar por uma reunião ao P.O. dos artefatos da *sprint* antes de começar a rodar, deixando o mesmo ciente do que será entregue durante a *review*; (2) Atender as necessidades do processo de desenvolvimento de *software*, já que nem tudo foi seguido como foi definido no PDS-GPES; (3) Melhorar as atividades da *review* apresentando códigos de

forma explícita na reunião e uma visão geral da *sprint* apresentando o andamento do *product backlog*; (4) Melhorar o relato das atividades realizadas na *sprint* entre a equipe através do cumprimento das reuniões; (5) Melhorar comunicação entre gerente e *scrum master* através do uso das ferramentas de GP sugeridas.

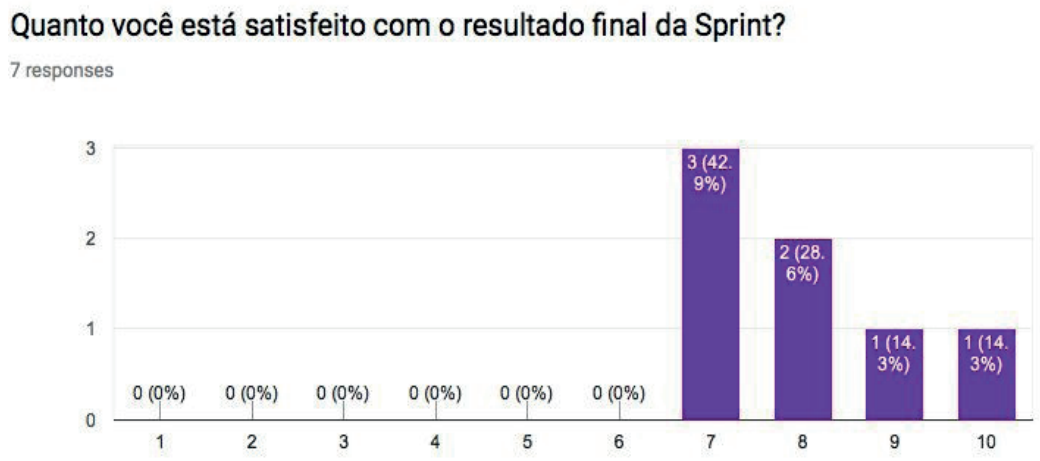
Ao finalizar a quinta *sprint* foi visto pelo time que diferente da quarta *sprint*, a equipe conseguiu enxergar o papel da gerência dando 100% de aceitação sobre as atividades exercidas pelo gerente de projetos, lembrando que os dados considerados como satisfeitos pela equipe foram as notas geradas entre seis e dez e podem ser visualizados na Figura 6.

Figura 6. Atividades da gerencia na Quinta *Sprint*



Apesar de nem todas as atividades concluídas, a equipe se sentiu também satisfeita com a entrega feita e que apesar da comunicação entre os mesmos não ter ficado muito clara, o que foi entregue durante a *review scrum* foi satisfatório por parte da equipe, fazendo com que 100% ficasse satisfeito com o que havia sido entregue, como pode ser percebido através da Figura 7.

Figura 7. Satisfação com os Resultados da Quinta *Sprint*.



CONCLUSÃO

A definição do *framework* foi iniciada com o intuito de gerenciar as atividades de um grupo de projeto de pesquisa em criar um ambiente reconfigurável de boas práticas de engenharia de *software*. O framework proposto foi aplicado e validado ao longo de cinco *sprints* no projeto ABPRES. O *framework* evolui a partir dos resultados da avaliação em cada *sprint,* tendo sido incluído, por exemplo: utilização de novas práticas estudadas nos outros projetos do GPES-IFPB; modificação de alguns papéis dentro da equipe; alteração das ferramentas de gerenciamento de projetos; e o uso de formulários de avaliação durante as retrospectivas.

Como resultado de avaliação do *framework,* pode ser citado os formulários aplicados nas *sprints* 3, 4 e 5 que trouxeram em forma de números a satisfação, seja positiva ou negativa, da equipe em cada *sprint* a qual ela foi aplicada. Os mesmos acabam podendo ser utilizados também como base de análise comparativa entre as *sprints*, fazendo com que o gerente possa acompanhar o aumento ou diminuição das atividades da gerência e do *framework,* graças as respostas da equipe.

A mudança de ferramenta de gerenciamento que houve durante a segunda a quinta *sprint*, deu a equipe a oportunidade de conhecer, aplicar, validar e comparar quatro ferramentas de gerenciamento podendo disponibilizar críticas em trabalhos futuros dos pontos positivos e negativos de cada ferramenta utilizada. Fazer com que a equipe pense sobre as ferramentas de gerenciamento trouxe uma melhor comunicação, tendo em vista que no final de cada *sprint* o time apresentou sua avaliação sobre aquela ferramenta e se seria viável a utilização dela na próxima *sprint*.

As atividades desenvolvidas no âmbito do GPES-IFPB têm uma diversidade de riscos relacionados à alta rotatividade dos envolvidos no projeto. Nesse contexto, foi extremamente importante a utilização do framework proposto que explica passo-a-passo todo o processo do gerenciamento de projetos. Além do *framework*, treinamentos são necessários para fazer com que a equipe esteja sempre atualizada e preparadas caso mudanças sejam feitas na equipe.

O framework resultante deste trabalho poderá e deverá ser adaptado para melhor aplicação do mesmo. Todos os dados e resultados até então estão disponíveis para melhor entendimento e revisão caso necessário. O *framework* criado é algo totalmente adaptado, onde o gerente de projetos terá total controle juntamente com a equipe de qualidade para alterar e modificar o mesmo de acordo com suas necessidades, pensando sempre em aumentar a produtividade da equipe e a satisfação do cliente. Com o intuito de melhorar a gerência, é de grande importância a curiosidade dos gerentes posteriores em descobrir novas maneiras e técnicas de como deixar a equipe cada vez mais interessada nas atividades que devem ser desenvolvidas.

REFERÊNCIAS

1. ALMEIDA, S. L. F.; MORAIS, A. D. S.; SILVA, S. C; ALENCAR, F. P.; LIRA, H. B. “Ambiente de Boas Práticas Reconfiguráveis em Engenharia de Software.” XI Semana de Ciência e Tecnologia do IFPB, <http://editora.ifpb.edu.br/index.php/ifpb/catalog/view/72/40/114-1>, 2016.

2. CAMPOS, E. L. “Técnica para fazer uma reunião de retrospectiva scrum.” Blog ScrumHalf. 2012. Disponível em: <https://shar.es/1U4IEq>. Acessado em Fev/17.

3. MORAIS, A. D. S.; RANGEL, H. J. L.; LIRA, H. B.; RODRIGUES, N.N.; ALMEIDA, S.L.F.; SILVA, S.C.;MEDEIROS, F.P.A.”Repositório de Práticas em Gerenciamento de Projetos de Software.” XI Semana de Ciência e Tecnologia IFPB, <http://editora.ifpb.edu.br/index.php/ifpb/catalog/view/72/40/114-1> 2016.

4. PMI, Project Management Institute. “Um Guia do Conhecimento em Gerenciamento de Projetos” (PMBOK®)–5a. Edição, PMI-Project Management Institute, 2016.

5. PRESSMAN, R. S.; MAXIM, B. R. “Engenharia de Software - Uma Abordagem Profissional” - 8 Ed. Argh Editora, 2016.

6. PRODANOV, C. C.; FREITAS, E. C. “Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico.” 2. ed. Novo Hamburgo: Feevale, 2013. Disponível em <goo.gl/ydRNhH>. Acessado em: Fev/2017.

7. SILVA, S. C.; RANGEL, H. J. L.; LIRA, H. B.; RODRIGUES, N. N.; MEDEIROS, F. P. A.”Repositório de Práticas em Processos de Desenvolvimento de Software.” XI Semana de Ciência e Tecnologia do IFPB, <http://editora.ifpb.edu.br/index.php/ifpb/catalog/view/72/40/114-1>, 2016.

07

IFLivros – Sistema de Gerenciamento de Livros para o IFG Câmpus Luziânia

- Jorge Tawnan Leles **Vilela**
  IFG
- Reinoldo Oliveira **Sommer**
  IFG
- Ernane Rosa **Martins**
  IFG

10.37885/201101963

RESUMO

O livro faz parte da base educacional em qualquer instituição de ensino, por esse motivo torná-lo mais acessível aos alunos é uma excelente contribuição para a comunidade acadêmica. Existem grupos em redes sociais que reúnem o pessoal interessado em trocar livros, mas um sistema facilitaria identificar quais livros uma pessoa precisa e quais ela tem e quer se desfazer. Neste trabalho é desenvolvido um sistema web com a finalidade de gerenciar os livros pessoais dos docentes e discentes do Instituto Federal de Goiás (IFG). A metodologia aplicada foi uma abordagem qualitativa caracterizada, sobretudo, pela sua natureza descritiva, com pesquisa bibliográfica e desenvolvimento de projeto de software.

Palavras-chave: Livros, Sistema.

INTRODUÇÃO

O livro é uma das formas mais antigas de se passar e adquirir conhecimento, querendo ou não ele faz parte de toda a vida acadêmica, do jardim de infância com os livros de colorir ao nível superior com os complexos livros de engenharia de *software*. A cidade de Luziânia conta com apenas uma livraria e uma biblioteca pública que não tem um acervo atualizado constantemente e nem tantos livros como deveria, para uma cidade que segundo os dados de 2015 do Instituto Brasileiro de Geografia e Estatística (IBGE) têm 1.823 docentes e 43.778 alunos matriculados na rede escolar.

O Instituto Federal de Goiás (IFG) Câmpus Luziânia têm disponível para os 536 alunos matriculados no primeiro semestre de 2017, 44 servidores do administrativo, 68 professores e aproximadamente 3.700 livros. A instituição possui um sistema de empréstimo desse acervo bibliográfico, no entanto, é notória a escassez de exemplares em todos os cursos que o instituto oferece, sendo o aluno obrigado a recorrer à compra de livros pessoais ou a buscar o recurso da pirataria. Para isso, será implantado um sistema web exclusivo para o IFG – Luziânia que gerenciará os livros dos alunos e servidores, proporcionando alguns benefícios como: ganho de tempo, venda de material inutilizado e conforto.

A vida acadêmica exige muito estudo e dedicação, e os estudantes sempre estão à procura de bons livros, seja para adquirir conhecimento ou apenas para passar o tempo. Tendo isso em vista, muitos livros são adquiridos no decorrer dos anos e depois não se sabe o que fazer com eles. Assim, vê se a necessidade de criação de um sistema único capaz de gerenciar e unificar o acervo pessoal de todos os envolvidos na instituição. Atualmente, não existe um serviço de automação do acervo pessoal de docentes e discentes do IFG. Os livros (digitais e físicos) são armazenados da forma que lhes convêm e muitas vezes inutilizados com o passar do tempo. Além do mais, livros são caros (em sua maioria) para alunos, fazendo com que o mesmo opte por adquirir cópias pirateadas.

O objetivo deste trabalho é desenvolver um sistema *web* para gerenciar o acervo individual de livros dos docentes e discentes do Instituto Federal de Goiás Câmpus Luziânia. Automatizar as atividades acadêmicas que envolva estoque pessoal de livros proporcionaria o gerenciamento e integração de estoque e desejos de cada usuário. Um sistema possibilitaria a unificação de todos os usuários, assim existirá uma forma de empréstimo, troca ou venda do material. Com todos estes recursos disponíveis, a procura por falsificações na Internet diminuiria, possibilitando a oportunidade de vender, trocar ou emprestar seus livros.

Nesta introdução foi apresentada além da definição da problemática de pesquisa, o objetivo, a justificativa, a importância do estudo e a estrutura da presente pesquisa. A sessão 2 traz o referencial teórico, com a formação de uma base conceitual e teórica que fornecem subsídios para o desenvolvimento deste estudo, bem como seu desenvolvimento

prático. Na sessão 3 são apresentados os materiais, os métodos empregados e as técnicas e procedimentos metodológicos utilizados para a realização do estudo. Na sessão 4 são mostrados os artefatos do projeto necessários para a criação do mesmo, como: regras de negócio, modelos e diagramas. Na sessão 5 são apresentadas as análises e discussões dos resultados obtidos na pesquisa sobre o desenvolvimento do sistema *web*. Finalmente na A sessão 6, é realizada uma abordagem conclusiva do projeto e apresentadas às recomendações de trabalhos futuros.

REFERENCIAL TEÓRICO

O mundo atual é altamente dependente de software. No século atual, a maioria dos produtos elétricos fabricados incluem um *hardware* e um *software* embutido. Uma grande parte da indústria e manufatura já é informatizada, assim como o sistema financeiro. Os sistemas de *software* são abstratos e intangíveis, isso significa que eles não são restringidos pelas propriedades dos materiais, nem governados pelas leis da física, mas devido a essa falta de restrições físicas, os sistemas de *software* podem se tornar extremamente complexos de modo muito rápido, difíceis de entender e caros para alterar (SERVILHEIRO, 2011).

As aplicações para web são projetadas para serem executados em um navegador através da internet. Essas aplicações se conectam com um servidor (pode ser uma conexão local, ou seja, na própria máquina ou em um servidor na internet) e seu principal objetivo é interagir com esse servidor através de requisições para obter dados/respostas. As tecnologias disponíveis, como *Hyper Text Markup Language* (HTML), *JavaScript* e uma linguagem de programação do *server-side* eram suficientes para atender a demanda. Ao longo dos anos a *internet* passou por grandes avanços e se tornou um ambiente também de negócios. Com isso o usuário além de ler suas notícias, passou a consultar preços de produtos, reservar passagens aéreas, solicitar orçamentos para serviços e inúmeras outras coisas (SANCHEZ & ALTHMANN, 2013). Uma aplicação para *Web* ("*WebApp*") é uma categoria de *software* centralizada em redes que está em constante evolução. Possui ambientes computacionais sofisticados que podem estar integrados a bancos corporativos e aplicações comerciais. Uma WebApp além de oferecer recursos especializados, fornece funções computacionais e conteúdo para o usuário final (PRESSMAN, 2011).

A arquitetura web é projetada em camadas e existem vários modelos disponíveis. Os mais conhecidos são em duas camadas, três camadas e quatro camadas. Os mais utilizados são em três e quatro camadas, pois separam melhor o comportamento da aplicação (PRESSMAN, 2011). Neste projeto faz-se a utilização do conceito *Thin Client* ("Cliente Magro"). Fundamentalmente significa que o cliente não precisa instalar nada, além de um navegador para utilizar os recursos de uma aplicação web, e o processamento dos dados

é feito pelos servidores. O padrão de projeto arquitetônico Model, View e Controller (MVC) separa a aplicação em três camadas trazendo várias vantagens na infraestrutura do projeto, a principal vantagem é justamente desacoplar a interface do usuário (*view*) da funcionalidade (*controller*) e do conteúdo (*model*) da aplicação *web (PRESSMAN, 2011).*

O objetivo da engenharia de *software* visa à construção de um *software*, em todos os aspectos. Os requisitos, projeto, desenvolvimento, gerenciamento, teste (SOMMERVILLE, 2007). A grande parte dos desenvolvedores comete o erro de achar que uma análise mais detalhada dos requisitos seria perda de tempo e começam o *software* pelo desenvolvimento. A engenharia de requisitos constrói uma ponte para o projeto, permitindo uma apreciação mais detalhada do *software* a ser realizado (PRESSMAN, 2011). As necessidades específicas do projeto e a construção devem atender as prioridades que orientam a ordem na qual o trabalho deve ser completado e as informações, funções e comportamentos que terão um impacto profundo no projeto resultante (PRESSMAN, 2011).

Os requisitos funcionais de um sistema descrevem o que ele deve fazer, como deve reagir a entradas específicas e de como deve se comportar em determinadas situações. Dependendo do caso, os requisitos funcionais podem dizer o que o sistema não deve fazer (SOMMERVILLE, 2011). Segundo Sommerville (2011), os requisitos funcionais podem ser divididos em duas categorias: Requisitos de usuário: são os requisitos elaborados de uma forma mais abstrata para que o usuário possa entender. Requisitos de sistema: são requisitos mais específicos que descrevem com detalhes as funcionalidades do sistema. Os requisitos não funcionais descrevem os requisitos que não estão diretamente relacionados com os serviços específicos do sistema. Os requisitos surgem por meio das necessidades dos usuários, devido a restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de *software* ou *hardware*, ou a partir de fatores externos, como regulamentos de segurança ou legislações de privacidade (SOMMERVILLE, 2011).

Modelagem de sistemas é um processo de desenvolvimento de modelos abstratos, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema. Esses modelos são utilizados na engenharia para ajudar a extrair os requisitos do sistema. Na fase de projeto, os modelos são utilizados para auxiliar os engenheiros na criação do *software*, e servem de documentação para a estrutura do sistema (SOMMERVILLE, 2011).

A *Unified Modeling Language* (UML) é um conjunto de notações gráficas, apoiada por modelos que ajudam na descrição e no projeto de *software*. A UML é um padrão relativamente aberto, nasceu da unificação das muitas linguagens gráficas de modelagem orientadas a objetos que floresceram no final dos anos oitenta, início dos noventa (FOWLER, 2005). No presente trabalho, a linguagem UML foi aplicada no desenvolvimento dos modelos de caso de uso e diagrama de classes. Os diagramas de casos de uso representam as funcionalidades

e elementos externos que podem ser observadas em um sistema. Modelo de análise que deve ser feito com base nos requisitos funcionais. Cada caso de uso é definido por uma descrição narrativa das interações que ocorrem entre os elementos externos e o sistema (BEZERRA, 2007). O diagrama de classes descreve as variações de objetos presentes no sistema e os tipos de relacionamento entre eles. Mostram as propriedades e as operações, mais conhecidas como métodos, de uma classe e a maneira que os objetos se aplicam as restrições conectadas (FOWLER, 2005).

MATERIAIS E MÉTODOS

O projeto está na categoria de Aplicações *Web*. Foram utilizadas as metodologias da engenharia de *software* descritas a seguir, para atender as necessidades do projeto. É retratado o processo de engenharia de requisitos utilizado no sistema. Na fase de estudo da viabilidade, foi feito uma análise com os professores e servidores da biblioteca do IFG Luziânia sobre a viabilidade do projeto e se poderia agregar valor à instituição. Na fase de elicitação e análise de requisitos, utilizando dados do IBGE, realizou-se o levantamento de quantos professores e alunos existem em Luziânia. Utilizando dados do IFG Luziânia, foi feito o levantamento da quantidade de livros na biblioteca, de alunos e de servidores. Assim, foi definido o problema e a necessidade do sistema. Na fase de especificação de requisitos, foi feito um documento de requisitos com base no problema e nas necessidades encontradas na fase de elicitação e de viabilidade. O documento é composto do escopo, requisitos funcionais, requisitos não funcionais e regras de negócio do sistema. Na fase de validação de requisitos foi realizada uma análise dos requisitos para garantir que todos os requisitos foram declarados de forma não ambígua. Procuramos detectar inconsistências, erros e se os requisitos estavam dentro do padrão estabelecido para o projeto. Em seguida foi feita uma análise dos requisitos junto com o orientador e entramos em comum acordo com as mudanças sugeridas.

Para nivelar a análise de requisitos a ser tomada prestou-se atenção nos seguintes fatores: Tamanho e complexidade do aplicativo; Tamanho da equipe de desenvolvedores; Nível em que os membros da equipe de desenvolvimento trabalham juntos; Nível de êxito da organização é diretamente dependente do êxito da aplicação. Analisando o tamanho da equipe e a complexidade do projeto, o mesmo foi mensurado apenas para atender a demanda do IFG. A pesquisa envolveu uma abordagem qualitativa caracterizada, sobretudo, pela sua natureza descritiva, com pesquisa bibliográfica, desenvolvimento de projeto de software e realização de questionário de usabilidade com discentes e docentes do Instituto Federal de Goiás Câmpus Luziânia, de modo a avaliar os resultados do desenvolvimento do sistema *web*.

Foram utilizadas as seguintes tecnologias na construção do projeto:

1. ASP.NET MVC (*Microsoft*, 2017b): *framework* para criação de aplicativos *web* que utiliza o padrão arquitetônico MVC. Foi optada essa tecnologia pela boa integração com as ferramentas da *Microsoft* utilizadas nesse projeto.
2. *Visual Studio Community (Microsoft, 2015):* uma IDE gratuita, completa e extensível para a criação de aplicativos modernos para *Windows*, *Android* e iOS, bem como aplicativos *web* e serviços de nuvem. O *Visual Studio* é o ambiente de desenvolvimento mais indicado para trabalhar com a linguagem de programação C# e possui integração com as ferramentas da *Microsoft.*
3. Linguagem C# versão 7.0: O C# (leia-se C-Sharp) é uma linguagem de programação bastante popular desenvolvida pela *Microsoft*. É uma tecnologia gratuita e disponível para download no site da *Microsoft*. Foi utilizada por ser a linguagem padrão do *Visual Studio*.
4. *Entity Framework Code First* 6.1.3 (Dykstra, 2015): ferramenta de mapeamento objeto relacional (ORM – *Object Relational Management*). Utilizada para fazer a persistência com o banco de dados. Foi utilizada por ser uma ferramenta já integrada no projeto ASP.NET MVC, isso garante o seu bom funcionamento e suporte.
5. *Code First Migrations (dos Santos, 2012):* permite gerar atualizações gerenciáveis no banco de dados ou deixar que o próprio *Migrations* cuide de tudo de forma automática, mantendo o banco de dados sempre atualizado com suas classes. Utilizado para manter o banco atualizado com as classes de modelo.
6. ASP.NET Identity (Rastogi, Anderson, Dykstra, & Galloway, 2013): é responsável por toda a autenticação e facilmente integrado com as redes sociais. O Identity é um componente indispensável neste trabalho. Tornou-se *open-source* (código aberto) a partir da segunda versão.
7. MySql versão 5.7 (Oracle, 2017): Sistema gerenciador de banco de dados. Foi utilizado por ser gratuito.
8. Linguagem HTML, CSS, JavaScript: Utilizado na camada de visão. Essas linguagens são padrões universais, HTML por ser de marcação na web, CSS por ser de estilização das páginas e JavaScript por ser a linguagem de programação utilizada nos navegadores.
9. Bootstrap versão 3.3.7 (Otto & Thornton, 2017): *framework front-end* para desenvolvimento de projetos web. Foi utilizado neste trabalho por reunir de forma simples e fácil as linguagens utilizadas.
10. JQuery versão 1.12.1 (The jQuery Foundation, 2017): conjunto de interações de interface de usuário, efeitos, *widgets* e temas construídos em cima da biblioteca jQuery *JavaScript*. Utilizado na camada de visão.

11. Astah *Community (Astah, 2017):* ferramenta para criação de modelos, como casos de uso e diagramas de classe. Foi utilizada por ser uma ferramenta gratuita e que atende as necessidades desse projeto.
12. *Team Foundation Server* – TFS (Microsoft, 2017a): ferramenta para controle de versão. Integrada no *Visual Studio*, é bastante eficiente e atende as necessidades desse projeto.

IFLIVROS

IFLivros surgiu a partir da ideia de reunir todo material obtido pelas pessoas durante a fase acadêmica, sejam livros, revistas, *e-books,* e colocá-los num sistema onde essas possam trocar, vender e emprestar seu acervo. Esse sistema foi desenvolvido inicialmente para o IFG, mas pretende ser expandido a diversas instituições a nível mundial.

A Tabela 1 apresenta a especificação dos requisitos funcionais com um identificador, descrição e os requisitos relacionados.

Tabela 1. Requisitos funcionais.

Identifica dor	Descrição	Requisitos Relacionados
RF01	O sistema deve permitir o cadastramento e manutenção de tipos de grupo.	
RF02	O sistema deve permitir o cadastramento e manutenção de tipos de integrantes dos grupos.	
RF03	O sistema deve permitir o cadastramento e manutenção de tipos de livros.	
RF04	O sistema deve permitir o cadastramento e manutenção de tipos de oferta.	
RF05	O sistema deve permitir o cadastramento e manutenção de tipos de pessoa.	
RF06	O sistema deve permitir o cadastramento e manutenção de pessoas.	RF05, RF19
RF07	O sistema deve permitir o cadastramento e manutenção de livros.	
RF08	O sistema deve permitir o cadastramento e manutenção de desejos.	RF07, RF06, RF04
RF09	O sistema deve permitir o cadastramento e manutenção de grupos.	RF01
RF10	O sistema deve permitir o cadastramento e manutenção de estoque.	RF03, RF06, RF07
RF11	O sistema deve permitir o cadastramento e manutenção de gênero dos livros.	
RF12	O sistema deve permitir o cadastramento e manutenção de sugestão de livros.	RF06, RF11
RF13	O sistema deve permitir o cadastramento e manutenção de ligação entre livro e gênero.	RF07, RF11
RF14	O sistema deve permitir o cadastramento e manutenção de ofertas.	RF10, RF04
RF15	O sistema deve permitir o cadastramento e manutenção de transações.	RF06, RF10, RF14
RF16	O sistema deve permitir o cadastramento e manutenção de entradas no grupo.	RF06, RF09
RF17	O sistema deve permitir o cadastramento e manutenção de integrantes do grupo	RF16, RF02
RF18	O sistema deve permitir o cadastramento e manutenção de ofertas para grupo.	RF04, RF10, RF09
RF19	O sistema deve permitir o cadastramento e manutenção de usuários.	
RF20	O sistema deve conter tela de login.	RF19
RF21	O sistema deve ter uma tela onde o usuário posso pesquisar ofertas.	

A Tabela 2 apresenta a especificação dos requisitos não funcionais com um identificador, descrição e os requisitos relacionados.

Tabela 2. Requisitos não funcionais.

Identifica dor	Descrição	Requisitos Relacionados
RNF01	O sistema deve ser desenvolvido para *web*.	
RNF02	O sistema deve ser exclusivo para pessoas com algum vínculo ao IFG – Luziânia.	
RFN03	O sistema deve estar online 24 h/ 7 dias por semana.	
RFN04	O sistema deve ter uma interface de fácil utilização.	RNF01
RFN05	O sistema deve ser responsivo, para que rode em mobile.	RFN04

A Tabela 3 apresenta a especificação das regras de negócio com um identificador, descrição e os requisitos relacionados.

Tabela 3. Regras de negócio.

Identifica dor	Descrição	Requisitos Relacionados
RN01	A tela de cadastro, visualização, edição e exclusão de tipos de grupo deve conter o campo de: Nome do tipo (Será obrigatório, do tipo string e com tamanho de até 80 caracteres); Quantidade máxima de participantes do grupo (Será obrigatório, do tipo inteiro).	RF01
RN02	A tela de cadastro, visualização, edição e exclusão de tipos de integrantes dos grupos deve conter o campo de: Nome do tipo (Será obrigatório, do tipo string e com tamanho de até 80 caracteres).	RF02
RN03	A tela de cadastro, visualização, edição e exclusão de tipos de livros deve conter o campo de: Nome do tipo (Será obrigatório, do tipo string e com tamanho de até 80 caracteres).	RF03
RN04	A tela de cadastro, visualização, edição e exclusão de tipos de oferta deve conter o campo de: Nome do tipo (Será obrigatório, do tipo string e com tamanho de até 80 caracteres).	RF04
RN05	A tela de cadastro, visualização, edição e exclusão de tipos de pessoas deve conter o campo de: Nome do tipo (Será obrigatório, do tipo string e com tamanho de até 80 caracteres).	RF05
RN06	A tela de cadastro, visualização, edição e exclusão de gêneros deve conter o campo de: Nome (Será obrigatório, do tipo string e com tamanho de até 150 caracteres).	RF11
RN07	A tela de cadastro de pessoas deve conter os seguintes campos: Nome (Será obrigatório, do tipo string e com tamanho de até 80 caracteres). Matrícula (Será obrigatório, do tipo string e com tamanho de até 50 caracteres). CPF (Será obrigatório, do tipo string e com tamanho fixo de 15 caracteres). Sexo (Será obrigatório, do tipo string e com tamanho de 1 caractere). RG (Será obrigatório, do tipo string e com tamanho de até 20 caracteres). Órgão expedidor (Será obrigatório, do tipo string e com tamanho de até 50 caracteres). Cidade (Será obrigatório, do tipo string e com tamanho de até 150 caracteres). UF (Será obrigatório, do tipo string e com tamanho fixo de 2 caracteres). E-mail (Será obrigatório, do tipo string e com tamanho de até 150 caracteres). Endereço (Será obrigatório, do tipo string e com tamanho de até 150 caracteres). Telefone (Será obrigatório, do tipo string e com tamanho entre 13 e 14 caracteres). Celular (Será obrigatório, do tipo string e com tamanho entre 13 e 14 caracteres). Tipo Pessoa, da pessoa por padrão será um usuário comum, deve fazer referência com a tabela de tipopessoa. Usuário, recebera o código do usuário da pessoa cadastrada	RF06
RN08	O cadastramento de usuário deve conter os seguintes campos: E-mail (Será obrigatório, do tipo string e com tamanho de até 150 caracteres). Senha (Será obrigatório, do tipo string e com tamanho de até 50 caracteres).	RF19
RN09	O cadastro de pessoas e usuário devem ser feitos na mesma tela e sincronizados.	RN07, RN08
RN10	A tela de cadastro, visualização, edição e exclusão de livros deve conter o campo de: Autor (Será obrigatório, do tipo string e com tamanho entre 2 e 254 caracteres). Autor (Será obrigatório, do tipo string e com tamanho entre 2 e 50 caracteres). Número da edição (Será obrigatório e do tipo inteiro) Páginas (Será obrigatório e do tipo inteiro) Título (Será obrigatório, do tipo string e com tamanho entre 2 e 100 caracteres). Subtitulo (Será obrigatório, do tipo string e com tamanho entre 2 e 100 caracteres). Editora (Será obrigatório, do tipo string e com tamanho entre 2 e 100 caracteres). Ano (Será obrigatório, do tipo string e com tamanho fixo de 4 caracteres). Url (Não obrigatório do tipo string com tamanho máximo permitido pelo banco).	RF07
RN11	O cadastramento de Ligar Livro Gênero deve conter os seguintes campos: IdDoLivro que deve fazer referência ao livro. IdDoGenero que deve fazer referência ao gênero. Ativo, deve ser do tipo boolean e obrigatório.	RF13
RN12	O cadastramento de ligar livro gênero deve ser feito junto com o cadastro de livros, dando ao usuário a opção de selecionar quais gêneros ele quiser para o livro.	RN11, RN10

RN13	A tela de cadastro, visualização, edição e exclusão de grupos deve conter os campos de: IdTipoGrupo (Faz referência ao tipo do grupo). Nome (Será obrigatório, do tipo string com tamanho 100). DataCriacao (Será obrigatório, do tipo DateTime). Ativo (Será obrigatório, do tipo boolean).	RF09
RN14	O cadastro de entrada grupo deve conter os seguintes campos: IdPessoaConvite (Faz referência a pessoa que fez o convite). IdPessoaConvidado (Faz referência a pessoa que foi convidada). IdDogrupo (Faz referência ao grupo). FoiConvidado (Será obrigatório, do tipo boolean). Cadastrado (Será obrigatório, do tipo boolean).	RF16
RN15	O cadastro de integrantes do grupo deve conter os seguintes campos: IdTipoIntegranteGrupo (Faz referência ao tipo de integrante grupo). IdEntradaGrupo (Faz referência a entrada grupo). DataEntrada (Será obrigatório, do tipo DateTime). Bloqueado (Será obrigatório, do tipo boolean).	RF17
RN16	O cadastramento de entrada no grupo deve ser feito assim que o usuário confirma sua entrada no grupo.	RN14
RN17	Caso o usuário insira o mesmo livro o sistema deve contabilizar a quantidade inserida no livro que já existe em estoque.	RN10
RN18	A tela de cadastro, visualização, edição e exclusão de estoque deve conter o campo de: IdDapessoa (Faz referência a pessoa). IdLivro (Faz referência ao livro). IdTipoLivro (Faz referência ao tipo livro). Quantidade (Será obrigatório, do tipo inteiro).	RF10
RN19	Caso a quantidade do estoque seja zero o livro não deve aparecer no estoque do usuário, porém deve estar cadastrado.	RN18
RN20	A tela de cadastro, visualização, edição e exclusão de oferta deve conter o campo de: IdEstoque (Referente ao estoque). IdTipoOferta (Referente ao tipo oferta). Valor (Não será obrigatório, do tipo decimal). Quantidade (Será obrigatório, do tipo inteiro).	RF14
RN21	Caso a quantidade da oferta seja zero o livro não deve aparecer no estoque do usuário, porém deve estar cadastrado.	RN20
RN22	Caso o usuário insira o mesmo estoque e mesmo tipo de oferta o sistema deve contabilizar a quantidade inserida no livro que já existe em estoque.	RN18
RN23	A tela de cadastro, visualização, edição e exclusão de desejo deve conter o campo de: IdLivro (Referente ao livro). Quantidade (Será obrigatório, do tipo inteiro). IdDapessoa (Faz referência a pessoa). DataCriacao (Será obrigatório, do tipo DateTime).	RF08
RN24	Caso a quantidade do desejo seja zero o livro não deve aparecer no estoque do usuário, porém deve estar cadastrado.	RN23
RN25	Caso o usuário insira o mesmo desejo o sistema deve contabilizar a quantidade inserida no livro que já existe em estoque.	RN23
RN26	A tela de cadastro, visualização, edição e exclusão de oferta do grupo deve conter o campo de: IdEstoque (Referente ao estoque). IdTipoOferta (Referente ao tipo oferta). Valor (Não será obrigatório, do tipo decimal). Quantidade (Será obrigatório, do tipo inteiro). IdGrupo (Referente ao grupo).	RF18
RN27	A tela de pesquisa deve ser feita por nome do livro e deve conter um filtro por tipo do livro.	RF21
RN28	O cadastro de sugestão de livros deve conter os campos de: IdPessoa (Referente a pessoa). IdGenero (Referente ao gênero). Quantidade (Será obrigatório, do tipo inteiro).	RF12
RN29	O cadastro de sugestão de livro deve ser feito quando o usuário cadastra um livro em estoque, o sistema soma a quantidade mais um de acordo com o gênero do livro. Deve ser feito também na pesquisa, o sistema soma a quantidade de acordo com o gênero dos 5 primeiros registros encontrados na busca.	RN28
RN30	O Sistema deve permitir login com facebook	RF20
RN31	O Sistema deve conter uma tela de recuperação de senha.	RF19

O modelo relacional (Figura 2) é utilizado no BD, contendo informações importantes que são traduzidas em coleção de relações (tabelas). E o diagrama de classes (Figura 1) é um documento imprescindível para o desenvolvedor de código, pois esse artefato ilustra quais atributos e métodos devem conter nas classes, além de mostrar o relacionamento entre elas.

Figura 1. Diagrama de classes do IFLivros.

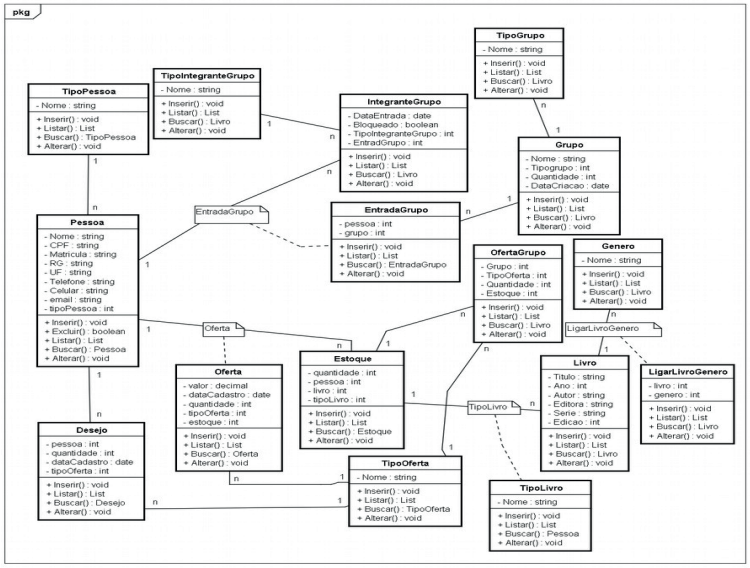
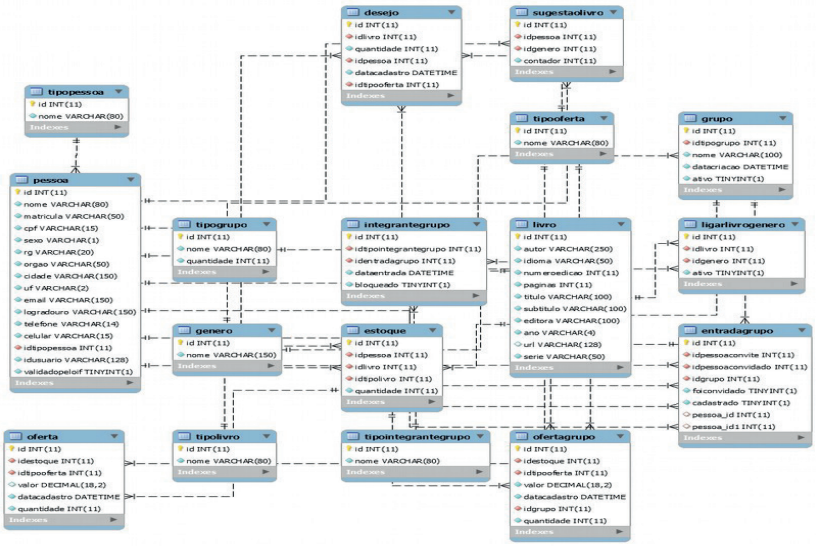
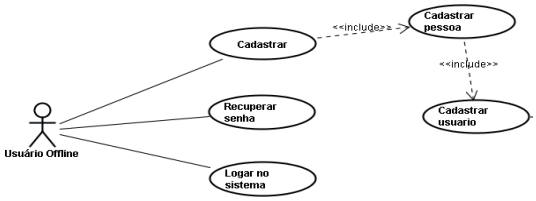


Figura 2. Modelo relacional do IFLivros.



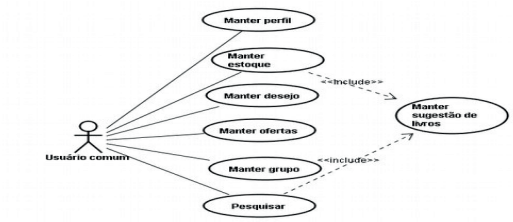
Diagramas de caso uso documentam a interação do sistema com o usuário do modo geral, sem se preocupar muito com a parte técnica. Um caso de uso requer um cenário, onde os eventos acontecem, um usuário (pode ser um administrador ou um usuário comum) e a comunicação entre os eventos e o usuário. O caso de uso referente aos usuários *off-line* é apresentado na Figura 3. As funções disponíveis no momento em que o usuário acessa o sistema são: cadastrar, recuperar senha e *logar* no sistema. Quando se cadastra uma pessoa, o sistema inclui um usuário no banco de dados.

Figura 3. Caso de uso *off-line*



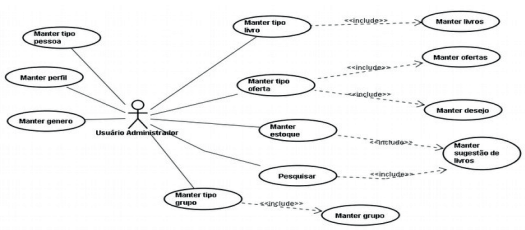
A Figura 4 mostra os casos de uso referente aos usuários cadastrados. Depois de cadastrado, o usuário tem alguns recursos disponíveis: manter perfil, manter estoque, manter desejo, manter ofertas, manter grupo e pesquisar. Manter significa alterar, criar, excluir e adicionar algo, igualmente conhecido como CRUD (*Create, Read, Update, Delete*).

Figura 4. Caso de uso: usuário comum



Por fim, a Figura 5 mostra os casos de uso referentes aos usuários administradores do sistema. Um administrador tem as permissões de usuário comum com algumas vantagens que são: manter tipo pessoa, manter tipo livro, manter gênero e manter tipo grupo. Isso significa que somente um administrador poderá cadastrar um livro ou o gênero de um, por exemplo, essa regra de negócio traz mais organização e segurança para o sistema, pois impossibilita que aja gêneros e livros inexistentes.

Figura 5. Caso de uso: Usuário Administrador

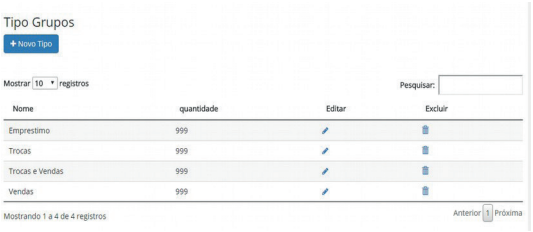


RESULTADOS

Baseado no conhecimento adquirido no presente estudo foi possível construir o sistema proposto. Sendo assim, este capítulo apresenta às telas do sistema seguindo as regras de negócio. O sistema encontra-se *online* no endereço www.iflivros.com.br.

A Figura 6 representa o resultado da regra de negócio RN01 e do requisito funcional RF01.

Figura 6. Tipo dos Grupos.



A Figura 7 representa o resultado da regra de negócio RN02 e do requisito funcional RF02.

Figura 7. Tipo de integrantes do grupo.



A Figura 8 representa o resultado da regra de negócio RN03 e do requisito funcional RF03.

Figura 8. Tipo dos livros.



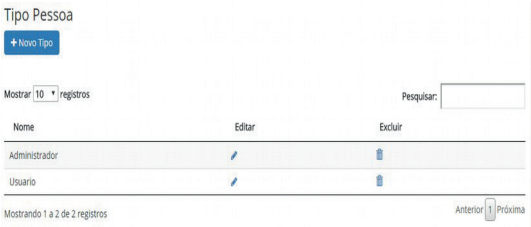
A Figura 9 representa o resultado da regra de negócio RN04 e do requisito funcional RF04.

Figura 9. Tipos de oferta.



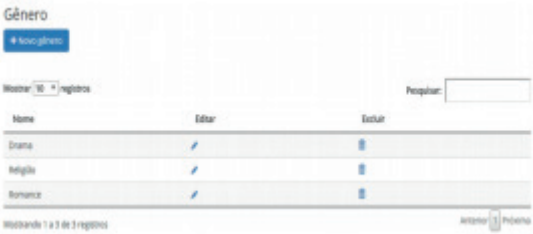
A Figura 10 representa o resultado da regra de negócio RN05 e do requisito funcional RF05.

Figura 10. Tipos de pessoa.



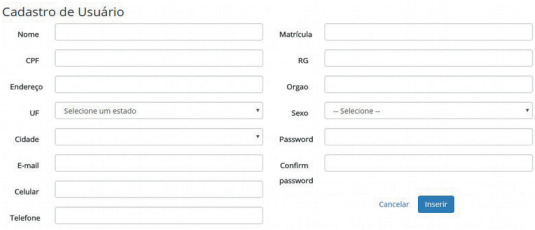
A Figura 11 representa o resultado da regra de negócio RN06 e do requisito funcional RF11.

Figura 11. Gêneros.



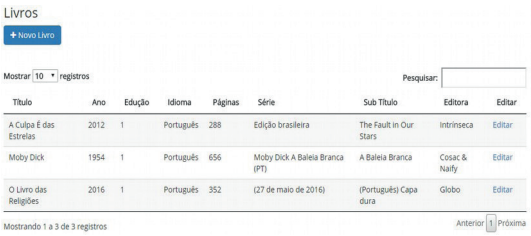
A Figura 12 representa o resultado das regras de negócio RN07, RN08, RN9 e dos requisitos funcionais RF06, RF07, RF08 e RF19.

Figura 12. Pessoas e usuário.



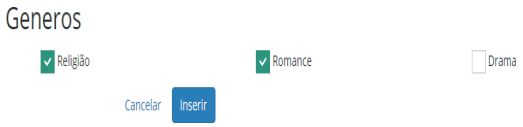
A Figura 13 representa o resultado da regra de negócio RF10 e do requisito funcional RF07.

Figura 13. Livros.



A Figura 14 representa o resultado das regras de negócio RN11 es RN12.

Figura 14. Ligar livro gênero.



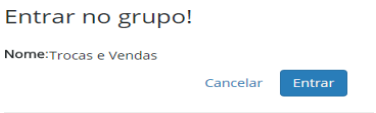
A Figura 15 representa o resultado da regra de negócio RN13 e requisito funcional RF09.

Figura 15. Grupos.



A Figura 16 representa o resultado das regras de negócio RN14, RN16 e do requisito funcional RF16.

Figura 16. Entrada no grupo.



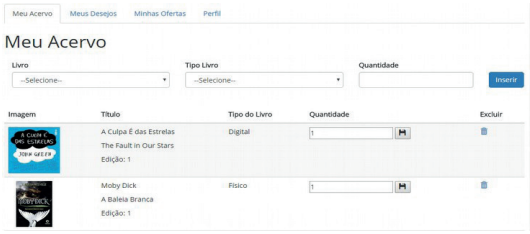
A Figura 17 representa o resultado da regra de negócio RN15 e do requisito funcional RF17.

Figura 17. Integrantes do grupo.



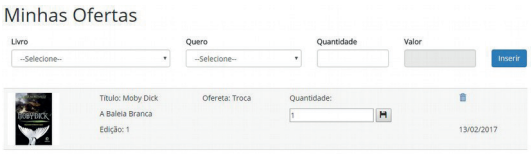
A Figura 18 representa o resultado da regra de negócio RN17, RN18, RN19 e do requisito funcional RF10.

Figura 18. Estoque.



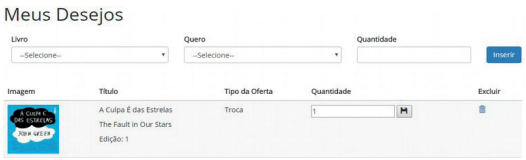
A Figura 19 representa o resultado das regras de negócio RN20, RN21, RN22 e do requisito funcional RF14.

Figura 19. Ofertas.



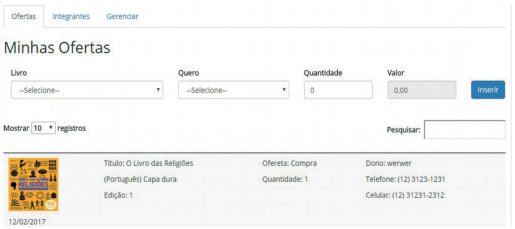
A Figura 20 representa o resultado das regras de negócio RN23, RN24, RN25 e do requisito funcional RF08.

Figura 20. Desejos.



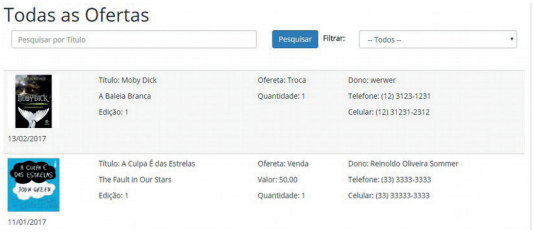
A Figura 21 é referente ao resultado da regra de negócio RN26 e do requisito funcional RF18.

Figura 21. Ofertas grupos.



A Figura 22 é referente ao resultado da regra de negócio RN27 e do requisito funcional RF21.

Figura 22. Pesquisa.



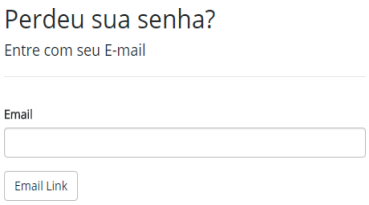
A Figura 23 é referente ao resultado da regra de negócio RN30 e RF20.

Figura 23. Login.



A Figura 24 é referente ao resultado da regra de negócio RN31 e RF19.

Figura 24. Recuperação de senha.



CONCLUSÃO

Neste trabalho foi desenvolvido um sistema *web* que visa atender os alunos e professores do Instituto Federal de Goiás Câmpus Luziânia. O sistema se chama IFLivros e desempenha o papel de gerenciar a coletânea de livros dos usuários. O Sistema de Gerenciamento de Livros – IFLivros atendeu aos objetivos propostos, que foram: ser disponibilizado *online* para os discentes e docentes da instituição, ter as funcionalidades de cadastramento, catalogação e consultas a livros, além de facilitar a venda, troca ou empréstimo de livros por docentes e discentes, diminuindo a busca por cópia ilegal de material didático.

Com a criação desse sistema, a acumulação de livros desnecessária pode ser resolvida, tendo em vista que todos os alunos e servidores do IFG Câmpus Luziânia dispõem de uma plataforma *online* gratuita capaz de facilitar a venda, troca ou empréstimo de livros pessoais, e consequentemente diminuir a pirataria. O sistema foi pensado para atender as necessidades do IFG Câmpus Luziânia, porém ele é ampliável para outras instituições e até mesmo para o público em geral, no entanto, seria inviável no momento, pois o projeto não possui investimento financeiro para expandi-lo a nível mundial.

Para trabalhos futuros sugerimos a ampliação do sistema para uso em outras instituições a nível global. Sendo necessárias algumas modificações, como adaptações de linguagem caso seja expandido para fora do Brasil (e nesse caso, a adição de exemplares pelo administrador do sistema seria muito mais vasta), contratação de servidor mais robusto para atender um publico maior sem perder desempenho, oferecer opções de pagamento e de frete e funcionalidades novas como *chat online,* adicionar relatórios estatísticos para

utilização dos dados armazenados pelo sistema, como por exemplo, qual o livro ou gênero de livro mais procurado e compartilhado.

REFERÊNCIAS

1. ASTAH. (2017). *Astah*. Disponível em: http://astah.net/editions/community. Acesso em: 10 dez. 2016

2. DOS SANTOS, C. (2012). *Entity Framework Code First–Migrations.* Disponível em: https://msdn.*Microsoft*.com/pt-br/library/jj591330.aspx. Acesso em: 03 fev 2017

3. DYKSTRA, T. (2015). *Getting Started with Entity Framework 6 Code First using MVC 5*. Fonte: asp.net: https://www.asp.net/mvc/overview/getting-started/getting- started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp- net-mvc-application. Acesso em: 10 out. 2016

4. FOWLER, M. (2005). *UML Essencial – Um breve guia para a linguagem-padrão de modelagem de objetos 3a Edição.*

5. *MICROSOFT*. (2015). *Microsoft*. VisualStudio. Disponível em: https://www.visualstudio.com/pt-br/vs/community/. Acesso em: 15 dez. 2016

6. *MICROSOFT*. (2017a). *TFS*. Fonte: TFS. Disponível em: https://www.visualstudio.com/pt-br/tfs/. Acesso em: 10 jan. 2017

7. MICROSOFT. (2017b). Visão geral do ASP.NET MVC. Disponível em: https://msdn.microsoft.com/pt-br/library/dd381412(v=vs.108). Acesso em: 10 fev. 2017.

8. ORACLE. (2017).*MySQL 5.7 Reference Manual*. Disponível em: https://dev.mysql.com/doc/refman/5.7/en/. Acesso em: 10 out. 2016

9. OTTO, M., & THORNTON, J. (2017). *Bootstrap*. Disponível em: http://getbootstrap.com.br/. Acesso em: 10 out. 2016

10. PRESSMAN. (2011). *Engenharia de Software uma abordagem profissional 7ª Edição.*

11. RASTOGI, P., ANDERSON, R., DYKSTRA, T., & GALLOWAY, J. (2013).

12. *Introduction to ASP.NET Identity*. Disponível em: https://www.asp.net/identity/overview/getting-started/introduction-to-aspnet- identity. Acesso em: 15 dez. 2016

13. SANCHEZ, F., & ALTHMANN, M. (2013). *Desenvolvimento Web com Asp.Net MVC.*

14. SOMMERVILLE. (2007). *Engenharia de Software 8ª Edição.* São Paulo: Pearson. SOMMERVILLE. (2011). *Engenharia de Software – 9ª Edição.*

15. THE JQUERY FOUNDATION. (2017). *jqueryui*. Disponível em: https://jqueryui.com/ Acesso em: 10 jan. 20

08

Proposta de Identificação dos usuários wireless em cumprimento com a lei do marco civil: o caso da Universidade do Estado do Pará – Campus XX

- Ramon Pinheiro da **Silva**
- Mario Afonso Gomes da **Silva**
- Manoel Fernandes Casimiro Damasceno **Costa**
- Carlos Benedito Barreiros **Gutierrez**

doi 10.37885/201101948

RESUMO

A UEPA – Campus XX enquadra-se no Marco Civil da Internet enquanto Administrador de Sistema Autônomo. No entanto, não emprega método para manter registros das conexões à Internet e de acesso a aplicações de Internet conforme exigido nos artigos 13 e 22 dessa lei. Este estudo tem como objetivo apresentar uma proposta para identificação de usuários e armazenamento de logs dos acessos à Internet dessa rede. Utilizou-se as tecnologias FreeRadius, MySQL e Syslog-NG para atender às obrigações de manutenção de registro dos acessos à Internet e às aplicações de Internet além do framework PHP Laravel para desenvolvimento de aplicação web para gestão de usuários e relatórios dos logs gerados. Constatou-se o cumprimento dos objetivos deste estudo ao atender às requisições do Marco Civil da Internet abordadas neste estudo por meio da identificação dos usuários e registro das respectivas conexões à rede. Some-se a isso a aplicação web para gerenciamento do sistema proposto.

Palavras-chave: Marco Civil da Internet, Autenticação de Usuário, Logs de Acesso, Segurança de WLAN.

INTRODUÇÃO

A utilização de redes locais de transmissão de dados sem fio, padrão IEEE 802.11 b/g/n ou Wi-Fi (WLAN), vem tomando proporções cada vez maiores em diferentes ambientes, como no uso doméstico, empresarial, para setores de pesquisa, no uso militar, entre outros, especialmente pelo rápido crescimento na utilização de dispositivos móveis (JINDAL e SINGH, 2017). O maior benefício dessa tecnologia de rede e um fator popularizante é o fato de que os usuários a acessam sem o uso de cabos de rede, havendo autonomia quanto à mobilidade do host, eliminando a necessidade de se planejar a infraestrutura de cabeamento da rede e dos respectivos dispositivos que a acessam, diminuindo a complexidade e o custo da sua implantação (WALIULLAH, MONIRUZZAMAN e RAHMAN, 2015).

Instituições de Ensino Superior (IES) vêm adotando essa tecnologia para facilitar e melhorar o ambiente para estudos e pesquisas, porém, há o risco de ocorrerem ações ilícitas por meio dessas conexões.

É de grande importância disponibilizar esses recursos, havendo uma extrema necessidade de sigilo das informações transitadas que são passíveis de uma grande variedade de ameaças, por isso, o zelo pela integridade e a definição de políticas de segurança para esses ativos são primordiais (CASTILHO e FONTE, 2013). A demanda por redes Wi-Fi em IES é significativa, sendo de suma importância resguardar os ativos destas, pois as informações transportadas são sensíveis tanto à comunidade acadêmica quanto aos sistemas dessas instituições (FIGUEIREDO, 2016).

Em relação a esse constante crescimento no uso dessas redes em IES, somado à facilidade e ao anonimato nesse acesso, há o aumento do risco de crimes serem cometidos nesse meio combinando-se técnicas de invasão com ferramentas já conhecidas e o desenvolvimento de novas ferramentas (MORETTI e BELLEZI, 2014). Muitos ataques às redes wireless são causados por pessoas sem sentimento de culpa, pois elas agem em ambiente “online” como nunca agiriam “off- line”, isto é, na “vida real” (GONÇALVES, 2014). Assim, a criminalidade digital se tornou uma ameaça no uso de redes Wi-Fi.

O campus XX da Universidade do Estado do Pará (UEPA), localizado no município de Castanhal, Pará, Brasil, disponibiliza acesso à Internet por meio de uma rede Wi-Fi para o público geral, alunos, funcionários e visitantes. Para utilizá-la, basta autenticar-se por meio da senha correspondente ao respectivo Service Set Identifier (SSID) da rede. Essa senha é disponibilizada em quadros de avisos em espaços de livre acesso do campus, assim sendo, qualquer pessoa pode acessar a rede.

Esse fato gera um cenário de insegurança, pois descarta-se qualquer proteção advinda da criptografia utilizada na sua senha de acesso à rede Wi-Fi ao divulgá-la indiscriminadamente (NOH, KIM e CHO, 2018). Nesse caso, é possível interceptar toda transmissão

na rede através de métodos e ferramentas tais como os ataques Man-In-The-Middle, De-Authentication e De-Association, Key- Recovery, Denial-Of-Service, entre outros (WALIULLAH, MONIRUZZAMAN e RAHMAN, 2015), além de possibilitar o acesso ilegítimo a chaves restritas bem como facilitar a invasão da privacidade dos usuários da rede, pondo em cheque informações sensíveis contidas nas conexões (NOH, KIM e CHO, 2018).

Essa fragilidade das redes wireless se dá pela própria natureza de broadcast dessa tecnologia, pois todos os dispositivos próximos na mesma faixa de frequência a enxergam, o que não ocorre em redes cabeadas onde os dispositivos devem estar interligados via cabos de rede. Assim, as camadas da tecnologia Wi-Fi que incrementam o conjunto de vulnerabilidades é a Física e a MAC, pois, em ambas, basta que o host malicioso entre na área da rede para "farejar" as transmissões ou adquirir acesso ilegítimo com a clonagem de endereços MAC (ZOU et al., 2016).

Além do exposto acima sobre as vulnerabilidades próprias da tecnologia Wi-Fi, há outro fator preocupante à UEPA, que é a legislação brasileira vigente sobre crimes cometidos na Internet, sendo inevitável atendê-lo para minimizar ou evitar complicações legais e possíveis punições à instituição.

A principal regulamentação que estabelece os direitos e deveres ao uso da Internet no Brasil é a lei nº 12.965/14, conhecida como o Marco Civil da Internet (MCI). Essa lei exige a manutenção dos registros de conexões à Internet e dos registros de acesso a aplicações de Internet em seus artigos 13 e 22, que devem ser mantidos pelos provedores de acesso a conexão. Esses registros devem permanecer armazenados pelo prazo de um ano, prorrogável por 60 dias (BRASIL, 2017).

Ainda segundo o MCI, em seu art. 10, a preservação da intimidade, da vida privada, da honra e da imagem dos envolvidos na provisão de conexão e de aplicações de Internet devem ser atendidas na guarda e disponibilização dos registros. Caso cometa-se um crime com uso da Internet, de acordo com o art. 23 do MCI, garantir o sigilo das informações e preservar a privacidade é dever do juiz encarregado após a disponibilização dos dados decorrente de requisição judicial.

Ao consultar por Número de Sistema Autônomo (ASN) através da Interface de Programação de Aplicações (API) do site IPINFO.IO, consta que a UEPA possui um ASN com o domínio UEPA.BR. Essa identificação é atribuída a Provedores de Serviços de Internet (ISP) e organizações que controlam blocos de endereços de Internet Protocol (IP) (IPINFO, 2018). Com isso, constata-se que a UEPA se enquadra na lei supracitada enquanto um Administrador de Sistema Autônomo que concede acesso à Internet, conforme o inciso IV do art. 5º do MCI.

Essa lei (artigos 24 e 25 – que tratam da atuação estatal na promoção da Internet e aplicações de Internet) prevê a adoção de tecnologias abertas e livres e foca a acessibilidade a todos os interessados, que devem ser atendidos independentemente de capacidade físico-motora, perceptiva, sensorial, entre outras, além da compatibilidade tanto com a leitura humana quanto com o tratamento automatizado das informações (Brasil, 2017).

O não cumprimento dessas diretrizes pode resultar desde advertências e multas até a suspensão temporária das atividades ou proibição de exercício das atividades. Assim, o cumprimento do exposto acima não é um fator trivial na disponibilização de conexões à Internet no Brasil.

Levando em consideração a prevenção de incidentes quanto ao uso de redes Wi-Fi e o exposto sobre o MCI, há o problema de que a UEPA não realiza a identificação dos usuários nem armazena os registros das conexões à sua rede. Caso ocorra um ato ilícito, não há como garantir informações detalhadas sobre esse ato e o usuário autor a serem disponibilizadas às autoridades competentes para aplicação das medidas cabíveis.

Tendo conhecimento sobre essa falta de segurança e os possíveis empecilhos legais que surgem, é indispensável que haja uma maneira de identificar os usuários e suas ações individualmente, garantindo essas informações pelo prazo determinado de forma segura, íntegra e confiável.

OBJETIVO

Este estudo objetiva atender aos critérios dos artigos 13 e 22 do MCI, propondo uma solução para o registro e identificação dos usuários, bem como o armazenamento de logs das conexões da rede Wi-Fi da UEPA. Por outro lado, não se pretende solucionar vulnerabilidades da rede Wi-Fi do campus, nem apresentar contramedidas a possíveis ataques e/ou invasões.

Definiu-se três objetivos específicos para atingir o objetivo geral deste estudo: definir a infraestrutura de rede utilizada na instituição, criar ambiente de rede virtual para aplicação da solução desenvolvida e desenvolver aplicação web de gerenciamento dos usuários e logs de acesso.

MÉTODOS

Verificou-se previamente a infraestrutura de rede da UEPA Campus XX, junto ao setor de Tecnologia da Informação (TI). A partir da análise das informações obtidas e de pesquisas na literatura, definiu-se a seguinte proposta de identificação dos usuários e registro dos logs da rede. Utilizou-se ambiente virtualizado para simular a solução desenvolvida configurando-se sistemas de roteamento, controle de usuários de rede e armazenamento de logs de

conexões de Internet. Além do desenvolvimento de aplicação web para gerenciamento de usuários e logs da rede.

Criação do Ambiente de Rede Virtual

Através de entrevista e aplicação de questionário, definiu-se a infraestrutura utilizada na instituição e, com essas informações, desenvolveu-se um novo cenário com recursos disponíveis para propor uma solução viável. A Fig. 1 apresenta um cenário com a utilização do recurso de virtualização de sistemas computacionais por meio do programa Oracle VM VirtualBox, versão 5.1.18.

Figura 1. Ambiente virtualizado e conexão em modo “Bridge”



Na Fig. 1, para validação da proposta, o ambiente foi adaptado para simular a topologia de rede foco deste estudo, por meio da criação de uma conexão “Bridge” e outra denominada “intnet”, que representam o link de saída para a Internet (WAN – Wide Area Network) e a rede local (LAN – Local Area Network) de acesso público, respectivamente. Nesse cenário, foram implementados todos os recursos (roteador, servidores e clientes) necessários ao desenvolvimento da solução.

Instalação do Sistema Roteador Mikrotik RouterOS

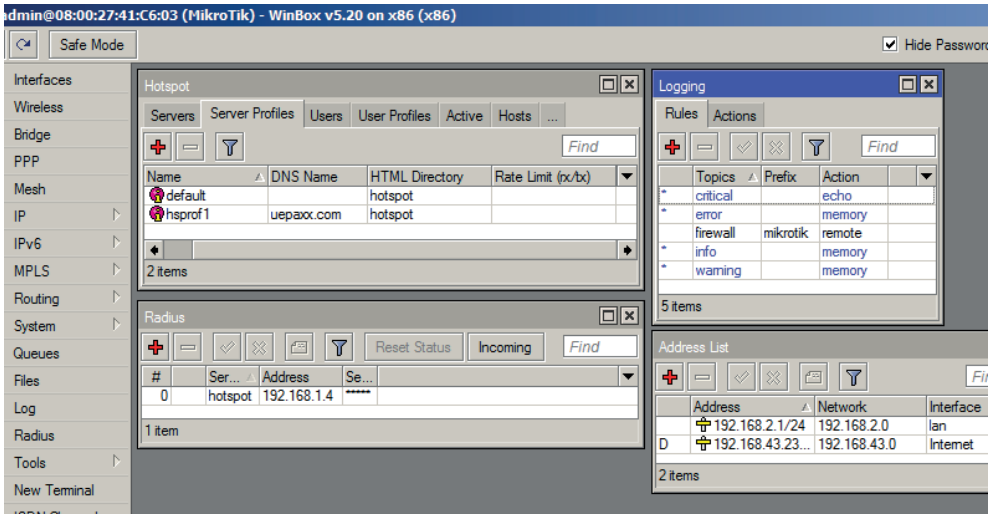
A rede do campus conta com um roteador Mikrotik RouterBoard, modelo 951UI-2HND, que atua gerenciando a rede e por onde os links de Internet são conectados para serem distribuídos pelo campus. Os Pontos de Acesso (AP) à rede wireless estão fisicamente interligados na rede cabeada, por meio de cabos Ethernet par trançado CAT-5e.

Esse roteador pode ser utilizado tanto em conjunto com dispositivo dedicado fornecido pela própria Mikrotik, quanto em máquinas de terceiros, pois o importante é seu sistema operacional, o Mikrotik RouterOS. Adequando ao contexto de virtualização, foi utilizada a segunda opção e foi configurada uma máquina virtual para receber a versão 5.20 desse sistema.

A Fig. 2 apresenta o WinBox, versão 5.20, que é o software utilitário para configuração do RouterOS. Por meio desse utilitário, foram configuradas as interfaces de rede “Internet”

e “Lan” no roteador, as regras para envio dos logs, e habilitados os serviços de autenticação por Radius e Hotspot no roteador, para, então, permitir acesso ao login no serviço de hotspot aos usuários.

Figura 2. Configurações de interfaces, serviço de Hotspot e Logs



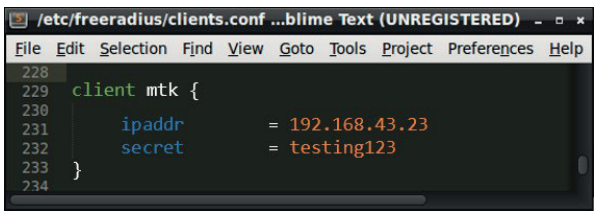
Identificação dos Usuários e Logs de Acesso da Rede

Para a identificação dos usuários na rede virtualizada, foi implementado um servidor utilizando a distribuição Linux Ubuntu 14.04, que trabalha em conjunto com o roteador e recebe os parâmetros de Login, para autenticar e registrar a sessão de cada usuário. Esse servidor utiliza o protocolo RADIUS, IETF/RFC 2865, com sistema gratuito e de código aberto chamado FreeRadius, versão 2.1.12.

Para que acontecesse a troca de mensagens entre o roteador e o servidor, foi necessário cadastrar o servidor de Radius no Mikrotik (Fig. 2), definindo o IP da máquina e a porta usada pelo serviço, além de modificar o arquivo “clients.conf” no FreeRadius.

A Fig. 3 apresenta o arquivo “clients.conf” do FreeRadius, onde foram definidas as informações de IP fixo e senha referentes ao cliente, neste caso o roteador, que está apto a enviar solicitações e obter resposta do serviço.

Figura 3. Alterações feitas no arquivo “clientes.conf”



Esse sistema permite armazenar contas de usuários em uma base de dados local, e, para isso, foi integrado a um banco de dados utilizando-se o MySQL, versão 5.5.60, disponibilizado gratuitamente pela empresa Oracle. Para possibilitar essa integração, foi necessário

executar o script do arquivos “admin.sql” e “schema.sql” na base de dados, além de configurar os arquivos “sql.conf” e “radiusd.conf”, disponíveis nos diretórios do FreeRadius.

A Fig. 4 apresenta os scripts contidos no arquivo “admin.sql”, responsáveis pela criação do banco de dados “radius” e do usuário “usuario_radius”, com suas devidas permissões, ambos utilizados pelo FreeRadius.

Figura 4. Scripts do arquivo “admin.sql”

```
# -*- text -*-
##
## admin.sql -- MySQL commands for creating the RADIUS user.
##
##      WARNING: You should change 'localhost' and 'radpass'
##               to something else.  Also update raddb/sql.conf
##               with the new RADIUS password.
##
##      $Id$
CREATE DATABASE IF NOT EXISTS 'radius';
#
#  Create default administrator for RADIUS
#
CREATE USER 'radius'@'localhost';
SET PASSWORD FOR 'radius'@'localhost' = PASSWORD('radpass');

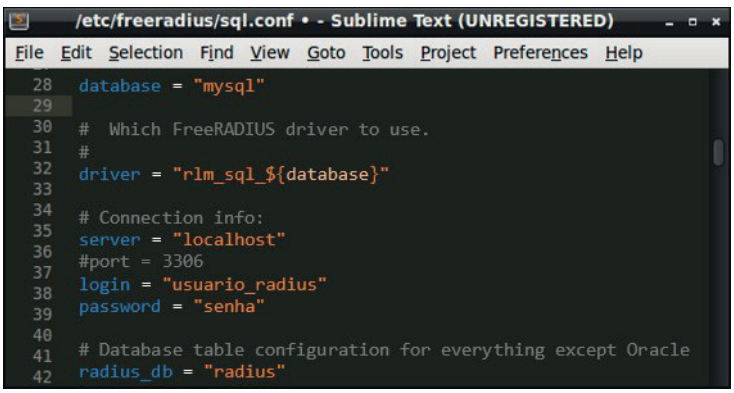
# The server can read any table in SQL
GRANT SELECT ON radius.* TO 'radius'@'localhost';

# The server can write to the accounting and post-auth logging table.
#
#  i.e.
GRANT ALL on radius.radacct TO 'radius'@'localhost';
GRANT ALL on radius.radpostauth TO 'radius'@'localhost';
```

Na Fig. 4, são exibidos os parâmetros de configuração padrão do arquivo “admin.sql” e foram definidas outras credencias de acesso durante a execução do script neste estudo por questões de segurança.

Após criado o banco, foi necessário modificar o arquivo “sql.conf”. A Fig. 5 apresenta esse arquivo e nele são cadastradas as informações referentes ao nome do banco, campo “radius_db”, endereço do servidor, campo “server”, porta de serviço do MySQL, campo “port” e nome de usuário e senha, campos “login” e “password”, respectivamente.

Figura 5. Configurações do arquivo “sql.conf”

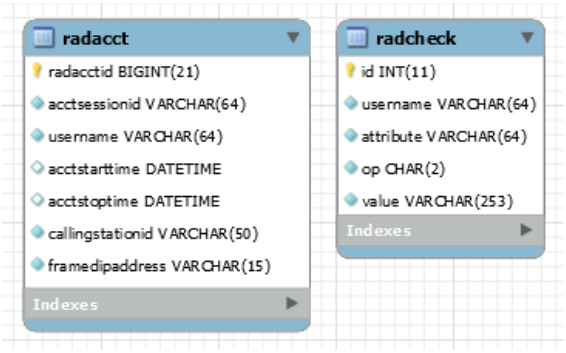


Ainda no arquivo “sql.conf”, foi necessário descomentar a linha “#readclients = yes”, removendo o caractere “#”, para possibilitar o retorno de informações do banco aos clientes, neste caso o roteador Mikrotik.

Com o intuito de habilitar o módulo SQL no FreeRadius, foi necessário descomentar também a linha “#$INCLUDE sql.conf” do arquivo “radiusd.conf” e todas as linhas que constam “#sql” no arquivo “default”, em “sites-available” do diretório “freradius”.

Dentro do MySQL, foram executados os scripts do arquivo “schema.sql”, necessários à geração das tabelas de registro de usuários e logs, utilizadas pelo FreeRadius. A Fig. 6 apresenta as tabelas que são criadas a partir dessa execução.

Figura 6. Tabelas usadas pelo FreeRadius

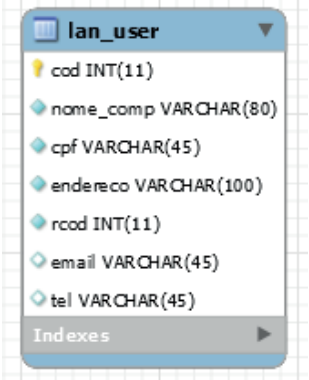


Baseando-se na Fig. 6, no banco de dados MySQL por meio de scripts, os usuários são registrados na tabela “radcheck” e são associados aos atributos “username” e “value”, que representam o nome de usuário (login) e senha de cada utilizador da rede, respectivamente.

A tabela “radacct” registra o histórico de sessões de cada usuário. O campo “acctsessionid” armazena o identificador da sessão, o campo “username” grava o nome do usuário que iniciou a sessão, “acctstarttime” e “acctstoptime” guardam data e hora de início e fim da sessão, respectivamente; o campo “callingstationid” armazena o endereço MAC do dispositivo conectado e “framedipaddress” guarda o endereço IP atribuído ao dispositivo do usuário que se conectou à rede.

Para atender às solicitações de identificação do MCI foi criada uma tabela para registro das informações pessoais de cada usuário. A Fig. 7 apresenta essa tabela, nomeada “lan_user”.

Figura 7. Estrutura da tabela “lan_user” no banco de dados



Na Fig. 7, a tabela “lan_user” se relaciona com a tabela “radcheck” através da chave estrangeira “rcod” e permite armazenar dados como nome completo, CPF, data, telefones e e-mail para contato.

Para o cadastro dos usuários da rede, foi construída uma aplicação utilizando as tecnologias PHP, versão 5.6.36, HTML5 e Javascript. Quanto ao acesso dos usuários, o Mikrotik já dispõe de uma tela de login padrão que foi modificada para se adaptar às necessidades deste estudo. Ambas são apresentadas no tópico 3. deste estudo.

Como o campus XX utiliza o mecanismo de Network Address Translation (NAT) para compartilhar o endereço IP e garantir a conexão à Internet aos usuários, foi necessário identificar os usuários individualmente. De acordo com Tebaldi and Guardia (2016), essa identificação se dá por meio dos endereços IP locais e portas TCP ou UDP de origem, além dos endereços IP requisitados e portas de destino de todas as conexões.

Em algum momento, dada a necessidade de levantar relatórios, esses registros podem ser cruzados com os dados das tabelas de sessão do FreeRadius, por meio dos endereços de IP, data e hora para identificação de um usuário que realizou determinado acesso.

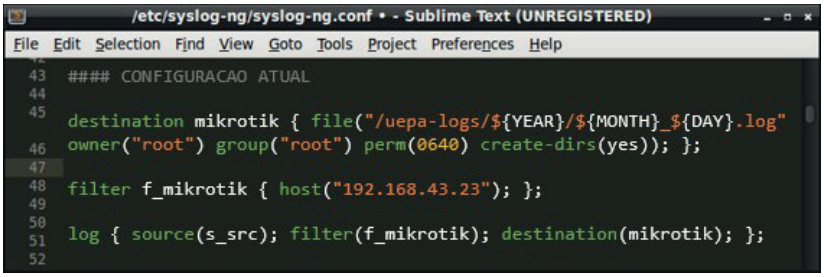
O sistema roteador da MikroTik e o servidor FreeRadius não realizam o armazenamento dos registros de acesso à Internet, sendo assim, é indispensável o uso de outra solução para esse fim.

O sistema básico que foi utilizado para o registro de logs de acesso foi o Syslog-NG, em sua versão 3.5.3, pois é compatível com o sistema RouterOS e de código aberto. Por questões de praticidade foi instalado no servidor do FreeRadius, visto que é uma aplicação direcionada a distribuições Linux.

Para que os acessos fossem gravados, foi necessário configurar o firewall do MikroTik para que ele permitisse o redirecionamento dos pacotes para o servidor de Log (Fig. 2), além de efetuar modificações no Syslog-NG.

A Fig. 8 apresenta o arquivo “syslog-ng.conf”, onde foram definidas as informações de IP do cliente, o roteador MikroTik, responsável por enviar os logs das conexões TCP e o fluxo UDP ao servidor. Também foi definido o diretório de armazenamento desses registros, definido por meio dos parâmetros ano, para a pasta, e mês e dia atuais, para o arquivo de log gerado.

Figura 8. Configurações feitas no arquivo “syslog-ng.conf”



Aplicação de Gestão de Usuários e Relatórios

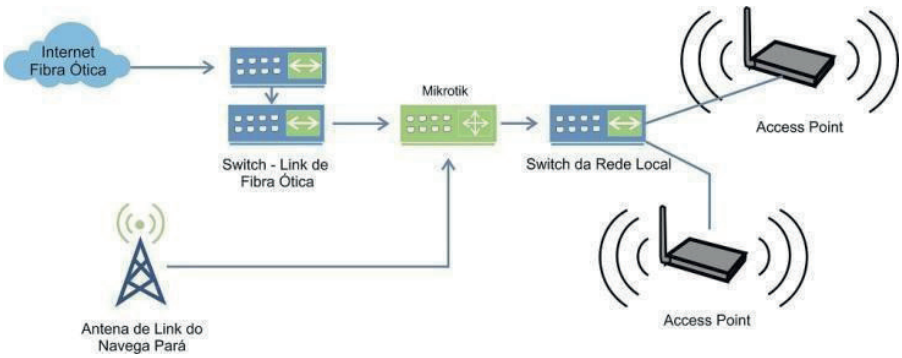
Como diferencial deste estudo, foi desenvolvida uma aplicação para ambiente web, construída utilizando o framework PHP Laravel, versão 5.4.36, que emprega a arquitetura MVC (Model View Controller) e incentiva o uso de boas práticas de Programação Orientada a Objetos (POO), em conjunto com as linguagens Javascript, HTML5 e CSS3.

Essa aplicação é responsável por gerenciar os registros recolhidos pelos sistemas FreeRadius e Syslog-NG, e a partir dessas informações emitir relatórios sobre as sessões e logs de acesso dos usuários da rede, tendo seu acesso restrito aos funcionários da instituição.

RESULTADOS

A Fig. 9 apresenta o resultado da análise junto ao corpo de TI do campus, onde foi possível compreender o design da topologia física e equipamentos da rede local da instituição.

Figura 9. Topologia de Rede do Campus XX da UEPA

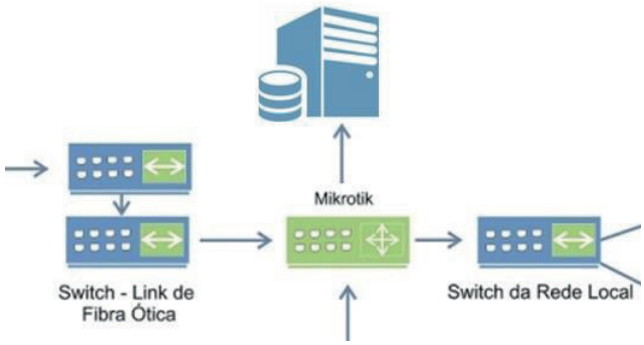


Como observa-se na Fig. 9, o campus conta com um link de fibra ótica sem limitação de banda e que faz parte da rede metropolitana de castanhal. Esse link é a porta de entrada e saída de todo o tráfego da rede local, e tem sua velocidade de transmissão reduzida ao padrão ethernet de 100 Mbps ao chegar ao switch da instituição. E como link reserva, em substituição ao de fibra ótica, é utilizado um canal de rádio de 4 Mbps do programa Navega Pará, fornecido pela Empresa de Tecnologia da Informação do Estado do Pará (PRODEPA).

Ambos os links são redirecionados ao Mikrotik, que distribui o sinal de Internet para todos os usuários da rede Wi-Fi da instituição em conjunto com os switches e AP locais.

Para atender as necessidades deste estudo, na estrutura de rede apresentada (Fig. 9), foi adicionada uma máquina servidora responsável pelo armazenamento dos logs e da aplicação de gestão desses arquivos. A Fig. 10 destaca a mudança ocorrida, onde é estabelecida a comunicação do servidor de logs com o roteador Mikrotik por cabo de rede ethernet.

Figura 10. Topologia da rede após a integração do servidor de logs



A Fig. 11 apresenta a tela de login que é disponibilizada pelo roteador Mikrotik para a qual o usuário é redirecionado ao conectar-se aos AP do campus. Essa página foi modificada para se adaptar à instituição e possui os campos “usuario” e “senha”, que representam as colunas da tabela “radcheck”, necessários à autenticação na rede.

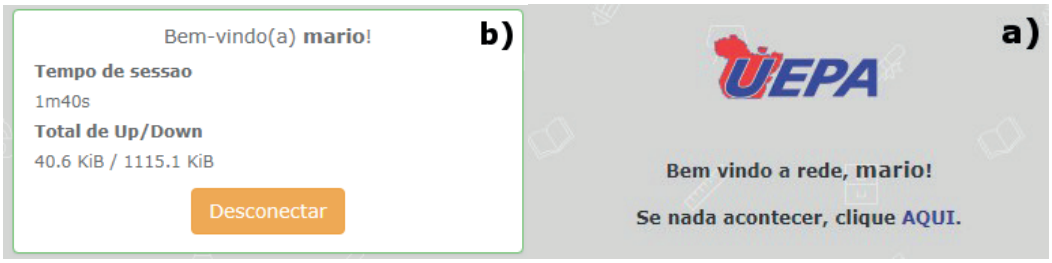
Figura 11. Tela de acesso à rede pública do campus



Por questões de segurança, para evitar o uso não autorizado dos recursos da rede, o usuário não cadastrado deverá solicitar à coordenação da instituição que o encaminhará ao setor de TI para o cadastro no banco de usuários para ter acesso à rede (CASTILHO e FONTE, 2013).

O usuário devidamente cadastrado, ao realizar login, tem seu acesso liberado e receberá uma mensagem de boas-vindas, conforme figura 12a. Ao mesmo tempo, é aberta uma janela pop-up que possibilita ao usuário desconectar-se da rede e mostra informações relativas ao tempo e tráfego da sessão, ilustrado na figura 12b.

Figura 12. Telas após login



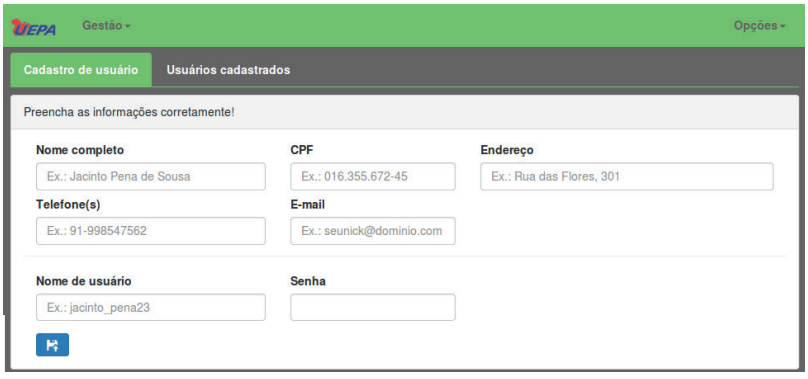
A Fig. 13 apresenta a tela de acesso à aplicação de gestão de relatórios e usuários da rede, que é restrita aos funcionários da administração. Nessa tela há os campos “usuario” e “senha”, que são os parâmetros para acessar o sistema.

Figura 13. Tela de acesso ao sistema de gestão de usuários e relatórios



O funcionário já cadastrado, ao realizar login, é redirecionado para a tela de gestão de usuários, conforme ilustra a figura 14. As opções apresentadas são “Cadastro de usuário” e “Usuários cadastrados”.

Figura 14. Tela de gestão de usuários

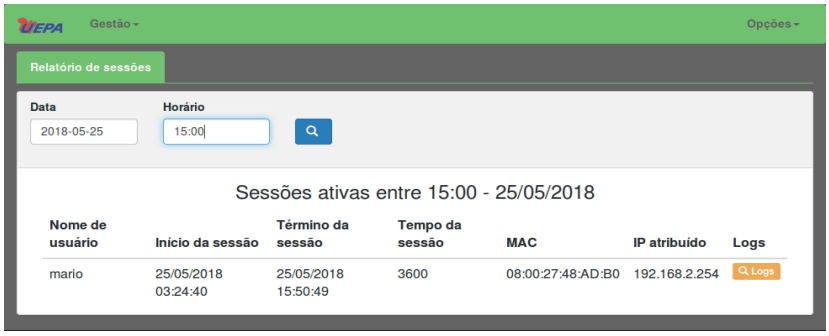


Na primeira opção, “Cadastro de usuário”, são exigidas informações pessoais do solicitante para criar uma conta de acesso à rede. O ato de cadastrar o usuário faz com que as informações como Nome, CPF e informações para contato sejam armazenadas na tabela “lan_user”, apresentada no tópico 2.3. deste estudo. O nome e o CPF são informações que têm validade jurídica e servem para atender ao requisito de identificação do usuário do MCI.

Já na segunda opção, “Usuários cadastrados”, é possível realizar pesquisas de usuários castrados na base de dados, com o intuito de atualizar os dados ou desativar uma conta específica.

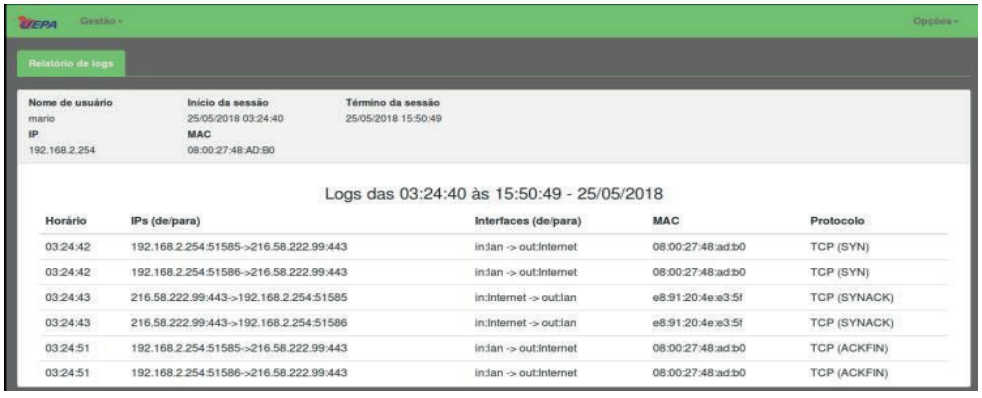
Caso o usuário entre no menu “Gestão” e selecione a opção “Relatórios”, ele será redirecionado à tela de “Relatório de sessões”, conforme a figura 15. Nessa área, por meio dos parâmetros “Data” e “Horário”, é possível saber quais usuários estavam ativos na rede em um momento específico, além de dados como início e término de cada sessão, endereço MAC do usuário e IP atribuído ao seu dispositivo durante a conexão. Caso não sejam informados os parâmetros, a página retorna com todas as sessões abertas no momento da pesquisa.

Figura 15. Tela de relatórios de sessões



Para cada uma das sessões listadas na Fig. 15, há um ícone ao lado denominado “Logs”. Ao clicar nesse ícone, uma janela é aberta e são apresentados todos os logs de acesso da sessão selecionada, apresentados na figura 16. Para isso, são cruzadas as informações de IP, data e hora dos logs com os registros de sessão (figura 15).

Figura 16. Tela de relatórios de Logs



Esses logs não guardam as URLs que são acessadas pelo usuário nem qualquer outro tipo de informação que possa ferir a sua privacidade, além disso, atende ao requisito de identificação solicitado pela lei: somente é guardado o IP de origem e destino das requisições.

Verificou-se em ambiente de testes que o acesso a uma página web aumenta 4kB o tamanho do arquivo de log em média. Usando como base de cálculo 50 usuários ativos na

rede, cada usuário acessando 50 páginas por dia, dentro do prazo mínimo previsto em lei, de um ano para manter os registros armazenados, o consumo em armazenamento seria de 3,6 GB, aproximadamente, desprezando o consumo acrescentado aos logs de outras tarefas e aplicações. Sendo assim, propõe-se o desenvolvimento de um método para compressão dos arquivos de log, visando otimizar o consumo de espaço em disco, além de incrementar o recurso de "Relatório de sessões" (Fig. 14) com outras possibilidades de busca, como, por exemplo, filtragem de sessões por usuário específico.

DISCUSSÃO

Correlacionando este a outros estudos similares, em CARISANI e GUARDIA (2016), objetivou-se a utilização das tecnologias de protocolo 802.1X, serviço Radius e servidor de diretórios Lightweight Directory Access Protocol (LDAP), IETF/RFC 1487, para autenticação e registro dos usuários em uma rede corporativa sem fio. Na proposta dos autores, a autenticação inicia-se a nível de porta de comunicação com os AP e utiliza como base usuários pré-cadastrados em um repositório local. Diferentemente, este estudo propõe um sistema que permite o acesso à rede por meio de portal web, exigindo como pré-requisitos estar conectado a um AP e cadastrado em base de dados SQL criada pelo FreeRadius. A solução deste estudo, além de registrar os logs das conexões da rede da instituição, armazena todos os logs de acesso à Internet feitos pelos usuários, o que não é abordado no estudo supracitado.

KUPTSOV, KHURRI e GURTOV (2009) apresentam um esquema de autenticação distribuída para hosts baseado em Host Indentity Protocol (HIP), IETF/RFC 4423, implementado em uma WLAN pública na cidade de Oulu, na Finlândia. Em cada AP da rede, foi configurado um firewall compatível que se comunica com um servidor de proxy HIP central. Essa abordagem possibilitou apenas o tráfego de clientes pré-estabelecidos em uma lista de controle de acesso Host Indentity Tag (HIT) compartilhada e sincronizada entre os firewalls, além de prover autenticação, garantindo por meio do IP Security Protocol (IPSec) um canal de comunicação seguro com os usuários. A autenticação na rede se dá de forma automática e transparente ao usuário, sem interação manual, e não são apresentados mecanismos para registro do histórico de logs de conexão e acesso dos clientes, o que distingue deste estudo, pois não possibilita a verificação de cada acesso à rede.

No estudo de FARIAS e SALES (2016), apresenta-se um mecanismo para rastreabilidade dos acessos na rede do Campus XX da UEPA. Nessa proposta, é utilizado um banco de dados SQL contendo as credenciais de cada cliente apto a acessar a rede para a autenticação dos usuários. Além disso, são registradas todas as sessões iniciadas, porém não se armazenam registros das conexões à Internet de cada usuário como é requisitado no MCI e abordado no presente estudo.

Este estudo desenvolveu uma aplicação exclusiva para gestão dos usuários e de relatórios dos logs das conexões à rede, o que o diferencia dos estudos correlatos apresentados.

CONSIDERAÇÕES FINAIS

O sistema proposto garantiu o cumprimento dos artigos 13 e 22 do MCI, desta forma, atingindo o objetivo proposto pelo estudo. Quanto ao art. 13, o sistema garante a manutenção de logs dos registros das conexões da rede Wi-Fi da UEPA para o prazo mínimo de um ano, além de não transferir a responsabilidade pela manutenção desses registros a terceiros, a solução é mantida pelo setor de TI da própria instituição. No que concerne ainda ao artigo 13, os registros de conexão podem permanecer armazenados por prazo superior a um ano, possibilitando atendimento de requisição judicial. O sistema proposto pelo estudo também atende ao art. 22 quanto ao fornecimento dos registros de acesso a aplicações de internet das respectivas conexões, os quais podem ser garantidos por soluções em nível de software.

Além disso, esse sistema possui um diferencial relevante que é o módulo de gestão dos usuários e logs da rede acessível a funcionários autorizados pela instituição, facilitando a verificação e disponibilização desses registros em relatórios. Almeja-se que com a implementação do sistema de autenticação e registro de logs de conexões e acessos, os usuários da rede Wi-Fi da instituição passem a ter conscientização sobre os riscos decorrentes do uso inadequado da rede, o que leva a uma mudança positiva de comportamento quanto ao uso dos recursos da rede do campus, possibilitando a geração de benefício mútuo, pois com a adoção dessa nova política de segurança, a UEPA passa a não ser alvo constante e totalmente conivente com ações maliciosas e os usuários passam a ser mais responsáveis em seus acessos.

REFERÊNCIAS

1. BRASIL. Lei n° 12.965/2014, de 23 de abril de 2014. Lei do Marco Civil da Internet. Diário Oficial da União: seção 1, Brasília, DF, ano 157, p. 1-3, 23 abr. 2014. (https://www.jusbrasil.com.br/diarios/69354783/dou-secao-1-24-04-2014-pg-1?ref=goto).

2. CARISANI, Rafael Vicente; GUARDIA, Helio Crestana. Identificação dos Usuários em Rede Corporativa. **Revista TIS**, v. 4, n. 2, p. 148-154, mai-ago, 2016. (http://www.revistatis.dc.ufscar.br/index.php/revista/article/view/311/110).

3. CASTILHO, Sérgio Duque; FONTE, Miguel Feitoza da. Política de segurança da informação aplicada em uma instituição de ensino mediante análise de Risco. **RETEC-Revista de Tecnologias**, v. 5, n. 2, p. 51- 66, jul-dez, 2012. (https://www.fatecourinhos.edu.br/retec/index.php/retec/article/view/99/144).

4. FARIAS, Ana Paula Pereira; SALES, Luiz Augusto Viana. **Implementação de Métodos de Rastreabilidade e Controle de Acesso à Rede Interna da UEPA - Campus XX de Castanhal**. 2016. Trabalho de Conclusão de Curso - Universidade do Estado do Pará (UEPA), Castanhal, 2016.

5. FIGUEIREDO, Davis Anderson. **Análise de vulnerabilidades e ameaças presentes em redes Wi-Fi (IEEE 802.11) de instituições de ensino superior de Minas Gerais**. 2016. 126. Dissertação (Mestrado Profissional em Sistemas de Informação e Gestão do Conhecimento) – Universidade FUMEC Faculdade De Ciências Empresariais, Belo Horizonte, 2016. (http://www.fumec.br/revistas/sigc/article/view/4051/2642).

6. GONÇALVES, Wilson José (organizador). **Termos Técnicos Fundamentais** – teoria e prática. Campo Grande - MS: UFMS, 2014. (https://d1wqtxts1xzle7.cloudfront.net/34613700/Termos_Tecnicos_Fundam entais_-_2014-A.pdf?1409724417=&response-content-disposition=inline%3B+filename%3DTermos_Te cnicos_Fundamentais_2014_A.pdf&Expires=1603336206&Signature=DqWvqFwrPHHwok8poOdXCP Y2Y1oI-2VSxRn6nWjJMDhfS2q--c4t91Gx7n9fzg~jDPTFJCqhniVtNdjBfPdyDSk4ZyJJCz3krA1-Q5eP 5h9mkNCRV1cwcAAf7Yka7leYJHgvi~wfuYZEGr2XiJRf3kT8klsyldDYKJPaDIPiz33wRQAVd4Q1V 3btjy~c0Ah7P4~8Nip~ZKzRQwPamY8Xsie39YIqe0tGSDacVPxzydYgGpnB1EWOB5aEqkwFBBtdqlOQ2zeAPqSmmp-aq~cv44uIViDodbcOWa6gbzmuUD7eH8rdOQRLj5r-mkn~EAW~VEc0mzQU9aOF VmQpywp6WQ &Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA).

7. IPINFO. **Network Universidade do Estado do Pará**. 2018. Disponível em: <https://ipinfo.io/AS262533>. Acesso em: 16/06/2018.

8. JINDAL, Poonam; SINGH, Brahmjit. Quantitative analysis of the security performance in wireless LANs. **Journal of King Saud University-Computer and Information Sciences**, v. 29, n. 3, p. 246-268, jul, 2017. (http://dx.doi.org/10.1016/j.jksuci.2014.12.012).

9. KUPTSOV, Dmitriy; KHURRI, Andrey; GURTOV, Andrei. Distributed user authentication in Wireless LANs. **World of Wireless, Mobile and Multimedia Networks & Workshops**, 2009. WoWMoM 2009. IEEE International Symposium on a. IEEE, 2009. p 1-9. (DOI: https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1109%2FWOWMOM.2009.5282469).

10. MORETTI, Cleber; BELLEZI, Marcos Augusto. Segurança em Redes Sem Fio 802.11. **Revista TIS**, v. 3, n. 1, p. 24-33, jan-abr, 2014. (http://revistatis.dc.ufscar.br/index.php/revista/article/view/73/67).

11. NOH, Jaewon; KIM, Jeehyeong; CHO, Sunghyun. Secure Authentication and Four-Way Handshake Scheme for Protected Individual Communication in Public Wi-Fi Networks. **IEEE Access**, v. 6, p. 16539- 16548, mar, 2018. (https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8314146).

12. TEBALDI, Lucas; GUARDIA, Helio Crestana. Serviço de autenticação, identificação e registro de usuários para redes sem fio públicas usando infraestrutura em nuvem. **Revista TIS**, v. 4, n. 2, p. 155-164, mai-ago, 2015. (http://revistatis.dc.ufscar.br/index.php/revista/article/view/312/111).

13. WALIULLAH, Md; MONIRUZZAMAN, A. B. M.; RAHMAN, Md Sadekur. An Experimental Study Analysis of Security Attacks at IEEE 802. 11 Wireless Local Area Network. **International Journal of Future Generation Communication and Networking**, v. 8, n. 1, p. 9-18, 2015. (https://pdfs.semanticscholar.org/125c/39d9131bd7a3c73c94ba7f8dd503f0e2c76a.pdf).

14. ZOU, Yulong; ZHU, Jia; WANG, Xianbin; HANZO, Lajos. A survey on wireless security: Technical challenges, recent advances, and future trends. **Proceedings of the IEEE**, v. 104, n. 9, p. 1727-1765, 2016. (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7467419).

09

“Práticas interdisciplinares apoiadas por problemas na formação do técnico em informática

- Luiz Fernando Delboni **Lomba**
  Instituto Federal de Mato Grosso do Sul - IFMS
- Olavo José **Luiz Junior**
  Instituto Federal do Paraná - IFPR

doi 10.37885/201102023

RESUMO

Com a implantação dos Institutos Federais, foi construída no Brasil uma organização pedagógica que contempla cursos que vão da educação básica à superior, oportunizando que os conceitos da computação sejam introduzidos já na educação básica. Os cursos técnicos da área da computação não devem se prender apenas a ensinar o "fazer", mas o ensinar para "saber fazer" e "saber buscar o que fazer". O objetivo da atividade relatada foi ampliar o escopo de aplicação prática de disciplinas técnicas do currículo do curso Técnico em Informática Integrado ao Ensino Médio. A abordagem foi desenvolver as competências de diferentes disciplinas em uma atividade conjunta, realizada sob a ótica da atuação em um ambiente profissional, baseada na metodologia da problematização. Os resultados apresentam as percepções dos professores quanto ao processo de avaliação e quanto à operacionalização de atividades desta natureza, concluindo-se que este tipo de abordagem infere mais significado aos aprendizes, superando a divisão do trabalho entre as ações de executar e de pensar.

Palavras-chave: Interdisciplinariedade, Educação Profissional, Problematização, PBL, Avaliação.

INTRODUÇÃO

O governo brasileiro propôs um modelo de proposta político-pedagógica na educação profissional, com uma organização verticalizada, que vai da educação básica ao ensino superior. É missão das escolas técnicas federais o compromisso de contribuir no desenvolvimento do país, através da criação de soluções para o desenvolvimento técnico e tecnológico de cada região, de forma sustentável e buscando a inclusão social (PACHECO, 2011).

Um dos desafios que este modelo oferece é levar aos diferentes níveis de ensino conteúdos que tradicionalmente são ofertados apenas em cursos de nível superior. Para tanto, apresentamos a seguir os conceitos que envolvem o currículo integrado e o ensino dos conteúdo de computação.

Os conteúdos de computação e o ensino médio integrado à educação profissional:

O currículo da educação básica no Brasil não contempla conteúdos da área de Ciência da Computação, exceto nos cursos técnicos de nível médio do eixo tecnológico de Informação e Comunicação. Na abordagem tecnicista, ensinar os conteúdos de computação nos cursos técnicos é ensinar o uso de ferramentas para resolução de problemas do mercado. Há o equívoco de equiparar o ensino de computação ao ensino de informática, sendo este último normalmente direcionado ao ensino do uso do equipamento e das suas ferramentas. Quando se fala em ensino de computação, busca-se um nível em que as técnicas possam ser aplicadas na resolução de problemas, nas diversas áreas do conhecimento.

Nos cursos técnicos de informática há o registro de alunos que desistem por considerarem desnecessária a conclusão do mesmo, uma vez que não desejam seguir ou atuar nessa área técnica de formação. Nestes casos o objetivo do estudante é tão somente obter o título, através do aprender a “fazer”, enquanto deveria contemplar o conhecimento obtido, ou seja, o “saber fazer” e “saber buscar o que fazer”.

Outro aspecto a ser considerado na formação dos profissionais da computação, é a integração entre as disciplinas, valorizando o trabalho desenvolvido em cada unidade componente do currículo e mostrando a importância de cada conteúdo no contexto geral da área. Currículo é definido pelo Ministério da Educação (2007b, p. 49) como “um desenho pedagógico e sua correspondente organização institucional à qual articula dinamicamente experiências, trabalho, valores, ensino, prática, teoria, comunidade, concepções e saberes observando as características históricas, econômicas e socioculturais do meio em que o processo se desenvolve”.

Nesse contexto, Frigotto, Ciavatta, & Ramos (2005) esclarecem que o currículo integrado é aquele que tem como base a compreensão do real, colocando o currículo como a organização do conhecimento, no sentido de compreender a realidade e a própria ciência na sua historicidade.

Integrar o currículo é organizar o conhecimento de maneira que os conceitos sejam compreendidos em um contexto que considere o todo. Não é apenas formar para o exercício de um ofício, mas também para a compreensão do mundo do trabalho, para uma atuação crítica e autônoma, de forma que os componentes curriculares, bem como os conteúdos, devem se complementar. Assim, busca- se desenvolver os fundamentos das diferentes ciências que propiciem aos jovens a capacidade analítica tanto dos processos técnicos quanto das relações sociais.

A ideia de formação integrada sugere superar o ser humano dividido historicamente pela divisão social do trabalho entre a ação de executar e a ação de pensar, dirigir ou planejar. De acordo com Ciavatta *et al*. (2005, p. 85): "Trata-se de superar a redução da preparação para o trabalho ao seu aspecto operacional, simplificado, escoimado dos conhecimentos que estão na sua gênese científico-tecnológica e na sua apropriação histórico-social".

Conforme o Ministério da Educação (2007b, p. 41) "o que se pretende é uma integração epistemológica, de conteúdos, de metodologias e de práticas educativas. Refere-se a uma integração entre teoria e prática, entre o saber e o saber-fazer". Em complemento, não se deve descartar as demandas oriundas do mercado de trabalho, que é dinâmico. A formação integrada deve apresentar elementos que contextualizem a realidade a ser vivenciada no exercício profissional.

Segundo Pacheco (2011, p. 15), a proposta pedagógica da educação profissional deve recusar o conhecimento exclusivamente voltado a formação da mão de obra e deve valorizar a relação do homem com o mundo: "Sua orientação pedagógica deve recusar o conhecimento exclusivamente enciclopédico, assentando-se no pensamento analítico, buscando uma formação profissional mais abrangente e flexível, com menos ênfase na formação para ofícios e mais na compreensão do mundo do trabalho e em uma participação qualitativamente superior nele".

A escola técnica onde se ambienta esse artigo foi inaugurada no ano de 2010. O início de suas atividades não trouxe novidades apenas à comunidade local, mas também aos profissionais que assumiam os cargos de professores nos primeiros cursos ofertados pela instituição, considerando a estrutura de formação verticalizada que permite ao docente atuar em diferentes níveis de ensino, que podem ir do nível médio ao doutorado. Em geral, os docentes vinham de experiências apenas com docência na graduação em cursos de engenharias e de computação, sem experiências no ensino técnico de nível médio.

Os autores iniciaram suas atividades no curso técnico de nível médio integrado em informática com o desafio de lecionar em um nível de ensino em que nunca haviam atuado. Para superar o paradigma da fragmentação, segundo Ferreira, Carpim, & Behrens (2010, p. 52), é necessário que o docente tenha "uma visão sistêmica, atuando de maneira significativa,

onde seu papel de pesquisador e mediador seja evidente e colabore para o desenvolvimento de um ser integral e pleno". Os professores procuraram pautar sua atuação nessa premissa.

Os autores consideram a avaliação como um evento importante para a aprendizagem, sobretudo na educação profissional. Como diz Moretto (2010, p. 18), a avaliação é um momento privilegiado de estudo, no qual "o ensino proporciona o desenvolvimento de habilidades e a aquisição de conhecimentos, que conduzem às competências almejadas".

A atividade prática interdisciplinar oportuniza ao estudante um *feedback* imediato sobre seu real aprendizado, bem como, de acordo com a condução da atividade, permite a reflexão sobre a melhor alternativa para resolução dos problemas, ajudando na aprendizagem significativa.

O ensino apoiado por problemas:

Com o objetivo de quebrar a abordagem tradicional das aulas, centrada no professor como transmissor dos conteúdos, buscou-se uma metodologia que coloque o aluno como o centro do processo de ensino-aprendizagem. Segundo Almeida e Moran (2005, p. 64) "na perspectiva da interatividade, o professor pode deixar de ser um transmissor de saberes para converter-se em formulador de problemas, provocador de interrogações, coordenador de equipes de trabalho, sistematizador de experiências e memória viva de uma educação que, em lugar de prender-se à transmissão, valoriza e possibilita o diálogo e a colaboração". Sob esta óptica adotou-se uma metodologia baseada em problemas para realizar algumas das atividades de um curso técnico em informática.

A Metodologia da Problematização, proposta por Charlez Maguerez (BORDENAVE, PEREIRA, 1977, apud BERBEL, 1995) e baseada no "Método do Arco", utiliza-se dos problemas como base para o processo de ensino-aprendizagem, em um circuito de etapas: observação da realidade, para identificar o problema; definição de pontos-chave, elencando os tópicos a serem investigados; teorização, para construir respostas concretas para o problema; hipóteses de solução, para alinhar os conceitos teóricos à realidade; e por fim, a aplicação prática, em que efetivamente há a intervenção sobre o problema. Segundo Berbel (1995) ela apoia-se na realidade do contexto, que observada sob diversos ângulos "manifesta-se para alunos e professor, através dos fatos concretos e daí são extraídos os problemas."

Outra abordagem é o Método de Aprendizagem Baseada em Problemas (*Problem Based Learning* – PBL), que de acordo com Sakai e Lima (1996 apud BERBEL, 1998) "é o eixo principal do aprendizado teórico do currículo de algumas escolas de Medicina, cuja filosofia pedagógica é o aprendizado centrado no aluno". O PBL é uma "metodologia formativa à medida que estimula uma atitude ativa do aluno em busca do conhecimento e não meramente informativa como é o caso da prática pedagógica tradicional."

Apesar da semelhança entre as duas metodologias há diferenças na concepção e na execução delas. Dentre as diferenças, apontadas por Berbel (1998), destaca-se o fato de que a Metodologia da Problematização não requer grandes alterações materiais ou físicas na escola: "as mudanças são mais na programação da disciplina", enquanto na PBL há necessidade de adequações nos espaços físicos (que devem ser suficientemente equipados e espaçosos), da equipe de trabalho, da organização dos horários, na distribuição de temas *versus* tempo, entre outras.

Os trabalhos na literatura sobre a abordagem apoiada por problemas na área de computação, em geral, apontam para o uso do PBL. O trabalho de Araújo, Rodrigues, Silva, Soares (2015) relata a experiência de um minicurso realizado por estudantes do curso de Licenciatura em Computação, utilizando a abordagem por problemas para ensinar lógica de programação aos alunos do ensino fundamental, a partir do uso de atividades da computação desplugada[1] e do software *scratch*[2].

Nos cursos de computação as disciplinas de algoritmos e programação, em geral, são classificadas pelos alunos como difíceis, justamente pela natureza voltada à resolução de problemas. Espelhado na experiência de Araújo, Rodrigues, Silva, Soares (2015), entende-se que utilizar esta abordagem em outras áreas do curso é uma oportunidade de retirar destas disciplinas a responsabilidade exclusiva de levar o aluno a raciocinar, já que "a computação, enquanto ciência tem como objeto de estudo a resolução de problemas". Temos no bojo dos cursos técnicos a oportunidade de introduzir os conceitos da Ciência da Computação já na educação básica, disseminando o pensamento computacional.

Outra experiência é apresentada por Notari, Bohn e Boff (2009), que utilizou a Aprendizagem Baseada em Problemas, em uma disciplina do curso de Licenciatura em Computação. O objetivo era ensinar o processo de instalação e configuração de Sistemas Operacionais, com enfoque na aprendizagem colaborativa, despertando a capacidade dos alunos em "trabalhar em conjunto para identificar e analisar problemas e/ou gerar soluções".

O trabalho de Oliveira, Arruda e Bittencourt (2007) apresenta a experiência de um curso de Engenharia da Computação com o PBL, destacando o trabalho desenvolvido na disciplina de Arquitetura de Computadores. Adaptando Venturelli (1997), eles apontam três adaptações para a aplicação da metodologia em cursos diferentes da Medicina e que se encaixam no contexto de um curso de ensino médio integrado: integrar as disciplinas em eixos temáticos; desenvolver estudos baseados em problemas e fundamentados na realidade; e

1 A Computação Desplugada (Computer Science Unplugged) é um método que defende o ensino de computação sem o uso do computador. Ele permite que o aluno use o pensamento computacional para resolver problemas reais, onde ele é o ator, e realiza todas as ações necessárias.

2 Scratch é uma plataforma de programação, com uma sintaxe simples, que permite a criação de histórias, jogos e animações por meio da manipulação de blocos dinâmicos, comumente utilizada para introduzir à programação quem não tem experiência no assunto.

utilizar um mecanismo de avaliação diferenciada, que seja contextualizada e permita uma reflexão do aluno sobre sua prática.

Segundo Berbel (1998), a PBL impacta na reorganização curricular do curso, se deu baseado na Metodologia da Problematização "pode ser utilizada para o ensino de determinados temas de uma disciplina". Assim, o presente trabalho se deu baseado Metodologia da Problematização, mas apoiado na experiência de Oliveira, Arruda e Bittencourt (2007), a fim de aplicar as adaptações sugeridas do PBL para introduzir uma nova concepção no contexto do curso técnico em informática, sem a necessidade de grandes mudanças na estrutura do curso e da escola.

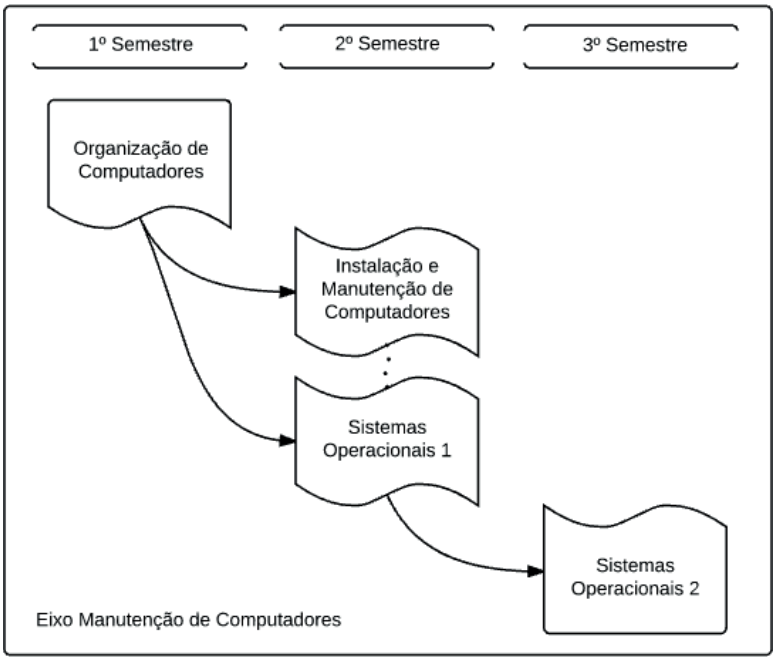
OBJETIVO

Relatar as experiências de uma proposta de intervenção realizada para o ensino de disciplinas de computação em um curso técnico de nível médio integrado em informática, utilizando-se as premissas do ensino baseado em problemas.

MÉTODOS

A primeira etapa foi identificar e relacionar as disciplinas que se encaixam no eixo temático definido como Manutenção de Computadores, considerando a estrutura curricular do curso técnico de nível médio integrado em Informática em que esta experiência foi realizada. Quatro disciplinas foram agrupadas neste eixo e são apresentadas na Figura 1, que também apresenta em qual semestre do curso cada uma delas é oferecida.

Figura 1. Organização das disciplinas no Eixo Manutenção de Computadores.



Em um segundo passo foi elaborado um plano de ação para o eixo temático, para garantir que os conteúdos de uma disciplina pudessem complementar o de outra e de que não haveria sobreposição entre eles.

A disciplina de Organização de Computadores é a responsável por introduzir os conceitos fundamentais do sistema computacional, com ênfase nos componentes que compõem um computador de mesa (*desktop*) e os seus detalhes técnicos. Diferente da abordagem tradicionalmente abordada nesta disciplina, aqui o foco é apresentar o contexto prático de como o técnico em informática utilizará estes conceitos. Por exemplo, na apresentação da hierarquia de memórias, o foco é permitir que o aluno identifique as características delas nos componentes do computador: nos conceitos de memória *cache*, por exemplo, o aluno deve identificar que se trata de um detalhe técnico do processador e que através dos manuais dos processadores é possível identificar esta característica.

Nas disciplinas de Instalação e Manutenção de Computadores e Sistemas Operacionais 1, o objetivo é permitir que o estudante faça a instalação e configuração de um computador: configuração da BIOS, instalação do sistema operacional, aplicativos e *drivers*, identificação de falhas físicas e de *software* (*troubleshooting*).

A Tabela 1 mostra a ementa das disciplinas que compõem o eixo.

Tabela 1. Ementa das disciplinas do Eixo Manutenção de Computadores

EIXO: Manutenção de Computadores	
Disciplina	Ementa
Organização de Computadores	Estrutura básica e conceitos fundamentais da informática. Tipos de componentes e periféricos. Critérios para seleção de hardware.
Instalação e Manutenção de Computadores	Configuração e instalação de microcomputadores e seus periféricos. Técnicas de Manutenção Preventiva e Corretiva. Realização de cópias de segurança, restauração de dados e atividades de prevenção, detecção e remoção de vírus. Seleção de soluções adequadas para corrigir as falhas de funcionamento de computadores, periféricos e softwares.
Sistemas Operacionais 1	Evolução histórica dos Sistemas Operacionais. Arquiteturas computacionais e os Sistemas Operacionais. Tipos de Sistemas Operacionais. Instalação e configuração de Sistemas Operacionais. Utilização de programas utilitários em modo texto e modo gráfico.
Sistemas Operacionais 2	Processamento. Processos, Sincronização, comunicação inter- processos. Gerenciamento de CPU, Memória e Entrada/Saída. Sistemas de Arquivos.

Já na disciplina de Sistemas Operacionais 2 são abordados os temas mais clássicos da disciplina, porém com o foco em mostrar a aplicação de cada um delas, através do uso de utilitários disponíveis nos sistemas proprietários e livres do mercado.

A etapa seguinte foi planejar as atividades a serem realizadas em cada uma das unidades curriculares, baseado em problemas. Apesar das unidades curriculares serem ministradas por docentes diversos e em dias diferentes, o plano de ação permitiu a identificação de problemas que poderiam ser abordados em conjunto, cabendo ao professor relacionar os conteúdos com os das outras disciplinas. Assim, as disciplinas de Instalação e Manutenção de Computadores e Sistemas Operacionais 1, ofertadas simultaneamente no 2º semestre do curso, foram planejadas para que ocorressem de forma integrada.

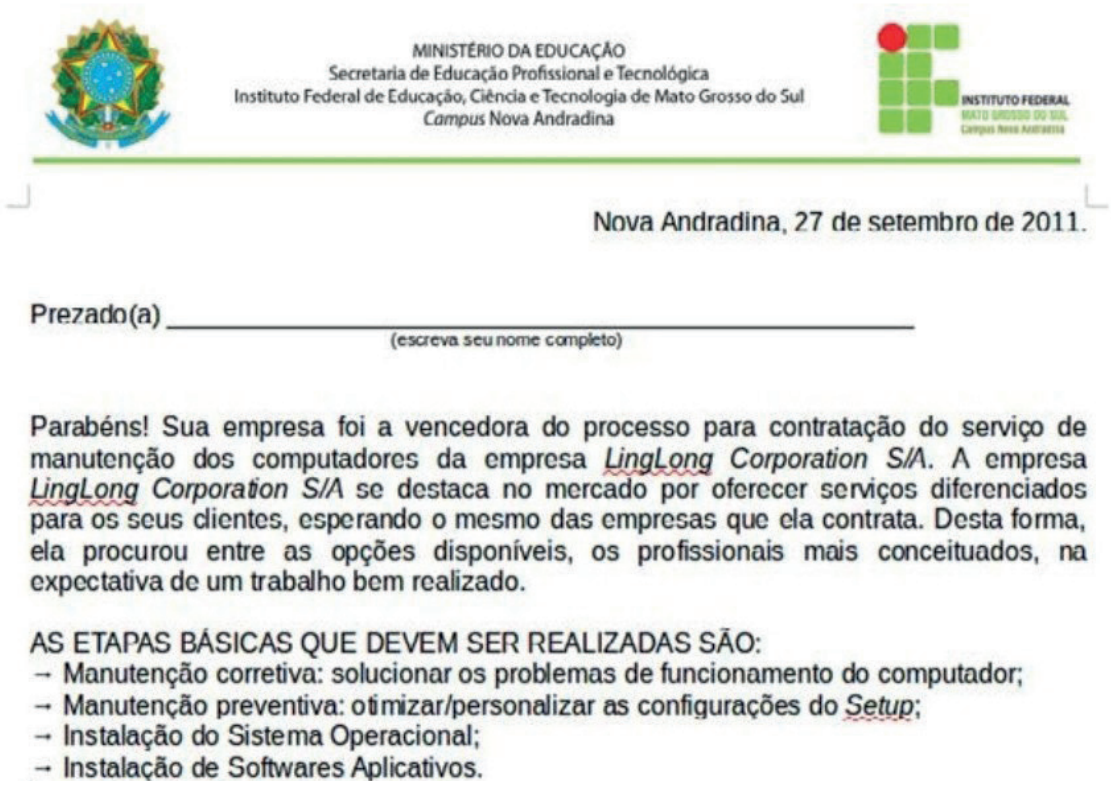
Posto esta organização, relata-se a seguir a realização de uma avaliação prática conjunta entre os dois componentes curriculares. Além de integrar os conceitos abordados das disciplinas, procurou-se avaliar as competências de ambas através de um problema conjunto, contextualizando o problema em um cenário do mercado, que simulava o ambiente de uma empresa prestadora de serviços.

O problema proposto exigia que o aluno aplicasse técnicas de manutenção corretiva (para solucionar os problemas de funcionamento do computador), manutenção preventiva (para otimizar/ personalizar as configurações do computador) e de instalação de *softwares* (sistema operacional, aplicativos e *drivers*).

Cada aluno recebeu um modelo simulado de contrato e ordem de serviços com uma série de trabalhos a serem efetuados em um computador, sendo estes complementares e interdependentes. O laboratório foi preparado pelos professores de forma a gerar problemas técnicos no funcionamento dos computadores a serem utilizados na atividade.

Para contextualização, apresentamos a atividade na Figura 2.

Figura 2. Fragmento do simulacro utilizado na atividade.



MINISTÉRIO DA EDUCAÇÃO
Secretaria de Educação Profissional e Tecnológica
Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul
Campus Nova Andradina

INSTITUTO FEDERAL

Nova Andradina, 27 de setembro de 2011.

Prezado(a) _____
(escreva seu nome completo)

Parabéns! Sua empresa foi a vencedora do processo para contratação do serviço de manutenção dos computadores da empresa *LingLong Corporation S/A*. A empresa *LingLong Corporation S/A* se destaca no mercado por oferecer serviços diferenciados para os seus clientes, esperando o mesmo das empresas que ela contrata. Desta forma, ela procurou entre as opções disponíveis, os profissionais mais conceituados, na expectativa de um trabalho bem realizado.

AS ETAPAS BÁSICAS QUE DEVEM SER REALIZADAS SÃO:
→ Manutenção corretiva: solucionar os problemas de funcionamento do computador;
→ Manutenção preventiva: otimizar/personalizar as configurações do *Setup*;
→ Instalação do Sistema Operacional;
→ Instalação de Softwares Aplicativos.

Além da realização e finalização correta das tarefas técnicas propostas, a atividade deveria ser finalizada com o preenchimento de relatório técnico que procedesse prestação de contas ao cliente do que foi efetivamente realizado e qual a razão de cada intervenção técnica. Com o intuito de considerar outros aspectos da formação integral do indivíduo e do profissional, foram introduzidos parâmetros como compromisso e profissionalismo na atividade.

Outro objetivo focava em que o estudante pudesse mensurar o resultado financeiro do seu trabalho, além de identificar a importância de informar às pessoas que confiaram em seu trabalho, o que e o porquê do serviço que foi realizado, de forma transparente e adequada. O enunciado do exercício sugeria: “Lembrando que conforme combinado na proposta do contrato, o serviço será ‘pago’ somente após a apresentação do relatório dos serviços prestados e a conclusão dos mesmos”, de forma a traçar um paralelo entre o pagamento do serviço na vida real, com a atribuição do conceito escolar referente à atividade.

Após a realização da primeira parte da avaliação, que os alunos consideravam que era única e estaria finalizada, os professores aplicaram um novo desafio, exigindo que novas

instalações/configurações fossem realizadas, fazendo um paralelo com solicitações de serviço de pós-venda muitas vezes existentes em ocorrências reais da vida profissional. Essa situação levaria à uma necessária reflexão sobre as consequências de sua intervenção técnica no médio prazo, procurando sempre executar da melhor forma as instalações solicitadas.

RESULTADOS

Quando a atividade foi revelada aos estudantes, a reação dos mesmos foi apresentar a preocupação que um possível mau desempenho na avaliação resultaria em uma "nota ruim" nas duas disciplinas. Alguns alunos questionaram se a nota seria zero caso não conseguissem finalizar a atividade, Muitos também perguntaram se a prova de recuperação seria feita em um formato similar, demonstrando contrariedade, o que evidencia a cultura ainda existente entre os estudantes de que o que esperam em uma avaliação é a nota à ela atribuída e não o processo e a aprendizagem que a envolve. Mesmo assim, ao final da atividade a percepção dos alunos foi considerá-la uma avaliação mais interessante do que o usual, haja vista que os motiva a utilizar conteúdos de componentes curriculares diversos.

Quanto aos professores, verificou-se que o formato das atividades permitiu uma maior facilidade por parte dos estudantes em demonstrar de fato a aprendizagem e isso se refletiu naturalmente em uma melhoria no resultado, em comparação com avaliações realizadas anteriormente.

Para realizar esta atividade, foram necessárias algumas alterações na organização e logística das aulas. Para este dia, excepcionalmente, as aulas das duas disciplinas foram agrupadas, a fim de permitir tempo suficiente para o estudante identificar os problemas, solucioná-los e para que escrevesse o relatório. Esta situação exigiu que outros professores colaborassem, cedendo o tempo de suas aulas. Este aspecto mostra a necessidade dos professores refletirem sobre sua prática e mudarem as características habituais de preparação das aulas, que tendem a uma rotina de repetição e reprodução (mais do mesmo...).

DISCUSSÃO

Verificou-se que, em geral, o professor até quer integrar a sua disciplina, mas não quer modificar sua forma de atuar, esperando que a outra área de conhecimento se integre à sua. Uma atividade integrada exige que o professor saia de sua área de atuação e passe a encontrar pontos convergentes em outras áreas do conhecimento. A partir da abordagem por problemas, foi possível criar um ponto C, que agregou as áreas A e B, não apenas somando os conteúdos, mas criando um novo ponto a partir da convergência deles.

Esta experiência envolveu professores que ministram disciplinas diferentes mas atuam em área comum, o que acabou facilitando a execução da experiência. Destarte, a organização do eixo de Manutenção de Computadores possibilitou que todos os professores envolvidos tivessem uma visão macro da área, indo além da sua unidade curricular. É provável que a agregação de conteúdos de outras áreas do conhecimento (através do trabalho dos respectivos docentes) ampliasse fortemente a necessidade de planejamento e preparação mais sofisticados da atividade, bem como demandasse um acompanhamento mais apurado dos resultados.

CONCLUSÃO

Os professores perceberam que o argumento utilizado, que coloca o aluno no papel de um profissional perante um problema, mostrou ser um formato que representou mais significado aos aprendizes sobre os conteúdos trabalhados. Com a problematização foi possível simular um ambiente mais próximo do encontrado na vida profissional, exigindo que os alunos identificassem os problemas e teorizassem hipóteses de acordo com o conhecimento obtido nas disciplinas, e aplicassem à realidade.

Esta experiência ficou restrita à integração das disciplinas de apenas uma área, sendo que novas experiências deverão ser realizadas para atingir as outras áreas do conhecimento, que envolvem a formação do aluno para uma atuação crítica e autônoma.

Quanto à metodologia utilizada, conclui-se que é necessário quebrar a abordagem tradicional utilizada, que está centrada no conteúdo e avaliada de acordo com a nota obtida na prova. O aluno precisa ser agente ativo do processo e não mero consumidor do conteúdo.

Aplicar este tipo de abordagem em momentos pontuais pode deixar o aluno desorientado. Faz-se necessário que esteja presente em outros momentos e não pontualmente em uma ou outra atividade, para que o aluno se aproprie da mesma e passe a ser o agente principal do processo.

Iniciar o trabalho com um grupo de disciplinas é uma oportunidade para buscar a desmistificação do trabalho interdisciplinar, apresentando-se gradativamente como alternativa factível e viável entre todos os docentes.

Esta abordagem mostrou-se como uma oportunidade para superar a divisão do trabalho entre as ações de executar e de pensar, a partir de um problema conjunto, próximo à realidade que será encontrada pelo estudante em sua atuação profissional. Colocar o aluno como agente ativo e crítico do processo, proporcionará a sua aproximação do mundo real, sobrepondo o papel de mero executor de tarefas.

REFERÊNCIAS

1. Almeida, M. E. B. and Moran, J. M. Integração das Tecnologias na Educação. Ministério da Educação, Seed, Brasília, DF. 2005.

2. Araújo, D. C., Rodrigues, A. N., Silva, C. V. A., and Soares, L. S. O ensino de computação na educação básica apoiado por problemas: Práticas de licenciandos em computação. In XXIII Workshop sobre Educação em Informática - Anais do XXXV Congresso da Sociedade Brasileira de Computação, Recife. SBC. 2015.

3. Berbel, N. A. N. Metodologia da problematização: uma alternativa metodológica apropriada para o ensino superior. Semina: Ciências Sociais e Humanas, 16(2):9–19. 1995.

4. Berbel, N. A. N. A problematização e a aprendizagem baseada em problemas: diferentes termos ou diferentes caminhos? Interface — Comunicação, Saúde, Educação, 2(2):139–154. 1998.

5. Bordenave, J. D. and Pereira, A. M. Estratégias de ensino aprendizagem. Vozes, Petrópolis. 1982.

6. Ferreira, J. L., Carpim, L., and Behrens, M. A. Do paradigma tradicional ao paradigma da complexidade: um novo caminhar na educação profissional. Boletim Técnico do Senac: a Revista da Educação Profissional, 36(1):51–59. 2010.

7. Frigotto, G., Ciavatta, M., and Ramos, M. Ensino médio integrado: concepção e contradições. Cortez, São Paulo. 2005.

8. Ministério da Educação. Programa Nacional de Integração da Educação Profissional com a Educação Básica na Modalidade de Educação de Jovens e Adultos. Ministério da Educação, Brasília. 2007.

9. Moretto, V. P. Prova: Um Momento Privilegiado de Estudo, Não Um Acerto de Contas. 9 ed. Lamparina, Rio de Janeiro. 2010.

10. Notari, D. L., Bohn, J., and Boff, E. Uma abordagem baseada em problemas para aprendizagem colaborativa de sistemas operacionais. In XVII Workshop sobre Educação em Informática - Anais do XXIX Congresso da Sociedade Brasileira de Computação, Bento Gonçalves. SBC. 2009.

11. Oliveira, W. L. A., Arruda, G. H. M., and Bittencourt, R. A. Uso do método pbl no ensino de arquitetura de computadores. In Proceedings of the 2007 International Conference on Engineering and Computer Education, Santos. International Conference On Engineering and Computer Education. 2007.

12. Pacheco, E. Institutos Federais: uma revolução na educação profissional e tecnológica. Moderna, Brasília. 2011.

13. Sakai, M. H. and Lima, G. Z. Pbl: uma visão geral do método. Olho Mágico, 2(5/6):51–59. 1996.

14. Venturelli, J. Educación médica. Serie PALTEX salud y sociedad 2000. Organización Panamericana de la Salud, Washington. 1997.

10

“Seleção de recursos de infraestrutura considerando diferentes opções de implantação

- Raphael de Aquino **Gomes**
  IFG
- Geovany Santa Cruz Victor **Rodrigues**
  IFG
- Gilberto Vampré Guimarães **Lobo**
  IFG
- Jonas Almeida dos **Santos**
  IFG

doi 10.37885/201101967

RESUMO

A multiplicidade de tipos de serviços disponíveis em plataformas de nuvem, derivada da grande variedade de provedores e regiões em que estes atuam, torna o processo de seleção de recursos de infraestrutura em tais ambientes uma tarefa desafiadora. Este desafio é ainda mais evidente ao se considerar modelos de implantação distintos em cenários em que o uso de alternativas a nuvens públicas é uma estratégia mais viável ou em que o uso de ambientes *on-premises* é uma restrição imposta. Diante disso, neste capítulo é apresentado o projeto, implementação e validação de uma ferramenta que permite auxiliar na seleção de recursos através da utilização de diferentes modelos de implantação e da exploração da multiplicidade de tipos de recurso em plataformas de nuvem. O objetivo principal é oferecer uma solução que auxilie na seleção de recursos de infraestrutura tendo como foco a otimização de custos. Experimentos realizados demonstram que a ferramenta proposta possui maior nível de produtividade e acurácia que soluções existentes.

Palavras-chave: Recurso, Infraestrutura, Computação em Nuvem, TCO.

INTRODUÇÃO

No cenário atual o uso de ambientes de Computação em Nuvem como alternativa para desenvolvimento de infraestrutura é uma estratégia cada vez mais adotada. Neste contexto, estimativas indicam que 2/3 de gastos com Infraestrutura como Serviço (*Infraestructure as a Service* – IaaS) são focados em recursos de computação (STALCUP, 2019). Para selecionar estes recursos, a descoberta de serviços em nuvem, isto é, o processo de encontrar um provedor que possa satisfazer de maneira adequada as necessidades do cliente (VAKILI; JAHANGIRI; SHARIFI, 2019), é realizado tendo como um dos focos principais a otimização de custo, dado que este é um dos parâmetros mais vitais para selecionar um provedor adequado (ACHAR; THILAGAM, 2014). Contudo, a adoção de uma única alternativa para esta tarefa é uma estratégia limitante. Como exemplo, para algumas aplicações a localidade de implantação não é importante. Dentre estas, no treinamento de modelos de redes neurais a geo-distribuição dos datacenters pode ser explorada como estratégia de alocação a um custo mais reduzido. A consideração de outro modelo de implantação também pode ser uma estratégia interessante visto que em cenários específicos pode ser economicamente vantajoso criar uma infraestrutura própria em vez de adotar nuvens públicas. Outro cenário que requer uma solução *on-premises* é aquele em que restrições legais impedem o uso de nuvens públicas.

A descoberta de serviços em nuvem considerando apenas nuvens públicas já é por si só uma tarefa desafiadora para os clientes, uma vez que o mercado de serviços em nuvem está sobrecarregado com um grande número de ofertas de nuvem heterogêneas, derivada da quantidade de provedores e as múltiplas regiões em que eles atuam. Além da grande variedade de serviços equivalentes, há também o fato de cada provedor adotar uma nomenclatura específica sobre os serviços por ele oferecidos, dado que em virtude da concorrência comercial não há iniciativa por parte dos mesmos em criar uma padronização. Como consequência, serviços de nuvem equivalentes podem ser representados por expressões diferentes a depender do provedor, restando para o cliente a tarefa de distinguir e associar um serviço a outro, usando as especificações e descrições dos mesmos. O uso de ferramentas de busca de propósito geral não é uma estratégia eficaz para lidar com este desafio pois as mesmas são baseadas somente no uso de palavras-chave e casamento entre termos. Todo o espectro semântico e de domínio da consulta é perdido ao ponderar e selecionar os resultados.

O uso de soluções de *Cloud Service Broker* (CSB) são comumente propostas como alternativa mais promissora na descoberta de serviços em nuvem (ELHABBASH et al., 2019). Um CSB pode ser definido como um serviço que atua em nome de um cliente para fornecer serviços, o que inclui o provisionamento automático de recursos em várias nuvens

(GROZEV; BUYYA, 2014). O uso de um CSB possibilita lidar com aspectos de heterogeneidade e solucionar problemas relacionados à descoberta de serviços. Contudo, grande parte das soluções existentes focam mais em aspectos funcionais e não-funcionais (CASTAÑÉ et al., 2018) em vez de econômicos, sendo que o que torna um CSB eminente entre outros é um mecanismo para otimizar o valor do investimento para o cliente. Não obstante, parte destas soluções não oferecem abstração suficiente para lidar com múltiplos provedores (PATINIOTAKIS; VERGINADIS; MENTZAS, 2014) ou são específicas para nuvem pública, não permitindo a análise com outros modelos de implantação.

Diante disso, nesse capítulo é descrito o projeto e implementação de um CSB para descoberta de serviços em nuvem visando estimativa de custo de recursos de computação, considerando diferentes alternativas de implantação. A ferramenta permite explorar a multiplicidade de tipos de recurso e região de implantação, além de possibilitar a análise de diferentes modelos de implantação. Experimentos demonstram que a ferramenta aqui apresentada oferece maior nível de usabilidade e acurácia que alternativas existentes, permitindo resultados mais significativos em um menor tempo.

OBJETIVO

O objetivo geral deste trabalho consiste em projetar, implantar e avaliar uma ferramenta que auxilie na busca de recursos computacionais considerando múltiplos provedores de nuvem e diferentes alternativas de implantação. O escopo desta proposta não é otimizar a descoberta de serviços em nuvem durante o tempo de execução mas sim oferecer uma ferramenta que sirva de apoio na busca de recursos com determinadas propriedades de hardware ou ainda oferecer um componente que possa ser utilizado na construção de soluções mais complexas. Este objetivo geral se desdobra nos seguintes objetivos específicos:

- Investigar as principais soluções existentes relacionadas a descoberta de recursos de compu- tação tendo como foco otimização de custos.
- Propor uma estratégia para padronização de termos relacionados à infraestrutura, que permita a agregação de diferentes alternativas de implantação de aplicação.
- Desenvolver uma técnica de estimativa de custo *on-premises*.
- Definir uma arquitetura que permita a descoberta de recursos de computação, tendo como base os mecanismos propostos.
- Construir um protótipo funcional da arquitetura definida, o qual servirá de base para avaliação dos mecanismos implantados.
- Avaliar, com base no protótipo, propriedades da solução proposta.

MÉTODOS

Para o desenvolvimento da proposta, inicialmente foi realizada uma fase inicial exploratória com o levantamento bibliográfico sobre ferramentas de seleção de recursos em nuvem e estimativa de custos de infraestrutura. Este estudo foi realizado mediante a análise da literatura científica publicada nesta área e análise de soluções desenvolvidas na indústria.

As abordagens de implementação de descoberta de serviço exigem que os clientes especifiquem requisitos funcionais e não-funcionais para permitir que o sistema recomende provedores de nuvem e serviços que melhor atendam às suas necessidades. Para tal, várias ferramentas de descoberta de serviço foram desenvolvidas na indústria, como Cloudorado Cloud Server Comparison (CLOUDORADO, 2020) e RankCloudz (RIGHTCLOUDZ, 2020). Contudo, estas ferramentas não suportam múltiplos modelos de implantação. Além disso, o processo de seleção de RankCloudz exige que os clientes determinem o nível de importância de um conjunto de atributos que servem apenas para comparar provedores, sem realizar a descoberta de serviços. Os próprios provedores de nu- vem oferecem ferramentas que auxiliam nesta tarefa visando otimização de custos, o que inclui a comparação com a implantação *on-premises*. Exemplos são AWS Pricing Calculator (AMAZON, 2020), Azure TCO Calculator (MICROSOFT, 2020), e Google Pricing Calculator (GOOGLE, 2020). Contudo, as consultas nestas alternativas consideram apenas recursos do próprio provedor.

Um dos aspectos a serem tratados na ferramenta aqui proposta é o desafio de prover interoperabilidade em nuvem. Pouco foi feito por outros trabalhos para enfrentar este desafio, o que é evidenciado pela adoção esparsa de bibliotecas de mapeamento e tradução para abstrair as diferenças entre as Interfaces de Programação de Aplicação (*Application Programming Interface* – API) dos provedores de serviço. Além disso, muitas estruturas pressupõem que os provedores adotarão uma API comum para que possam colaborar entre si. Na realidade, o mercado de nuvem demonstrou sua resistência em permitir que os clientes se movimentem livremente entre os concorrentes e, corroborando com esta vertente, a grande maioria da literatura assume que os dados publicados por provedores são comparáveis (ELHABBASH et al., 2019). No entanto, é evidente que os serviços oferecidos são fundamentalmente heterogêneos; cada provedor tem sua própria descrição das ofertas e não é realístico assumir que os provedores adotarão descrições de oferta padrão. Dessa formal, a solução mais promissora e amplamente discutida na literatura é o uso de ontologias.

Como primeiro passo na construção de uma solução baseada em ontologia foram analisadas as principais contribuições nesta vertente com relação a IaaS em nuvem, tendo como foco a otimização de custos. Para tal, foi utilizado como base os estudos de Tankelevicienea e Damaseviciusb (2009), onde foi apresentado um arcabouço de avaliação para ser utilizado durante o processo de construção de ontologias. No processo conduzido houve a

reavaliação das contribuições apresenta- das por Al-Sayed, Hassan e Omara (2019) e a expansão destes resultados considerando outras ontologias. A avaliação teve como escopo apenas os aspectos relacionados aos objetivos do CSB aqui proposto, de forma que demais características das ontologias foram desconsideradas.

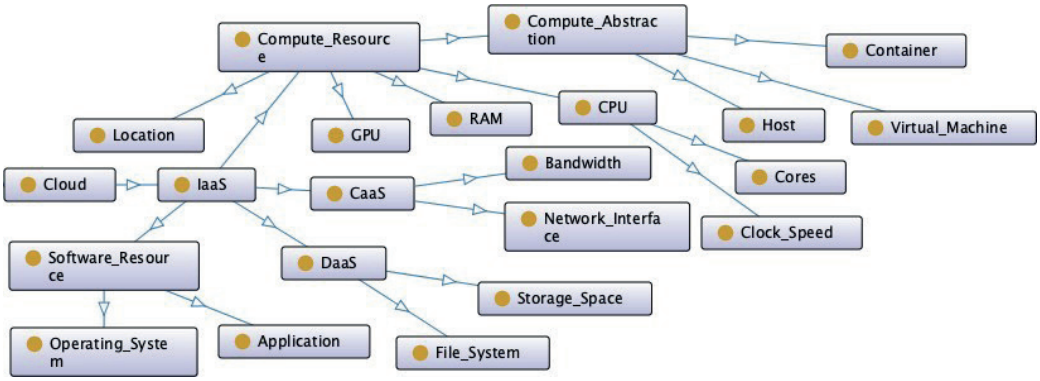
Continuando o processo de elaboração da ferramenta foi assumido que para que a estimativa de custos de infraestrutura seja completa a mesma deve considerar também a alocação de recursos usando outros modelos de implantação. Uma estratégia nessa direção é a análise do custo de implantação considerando uma infraestrutura *on-premises*. Para tal, foi proposto o uso da técnica de análise de Custo Total de Propriedade (*Total Cost of Ownership* – TCO), amplamente usada como um meio de abordar os custos reais atribuídos à propriedade e gerenciamento de uma infraestrutura de Tecnologia da Informação em uma empresa. Foi elaborada uma metodologia própria para obter esta estimativa uma vez que as principais estratégias existentes apresentam suposições específicas dos cenários considerados, dificultando seu uso no cenário aqui tratado. Ademais, apesar de alguns provedores de nuvem oferecerem ferramentas para estimativa de TCO, as mesmas são enviesadas de forma a privilegiar resultados em seus ambientes.

A metodologia de estimativa de TCO proposta se baseia nos trabalhos de Li et al. (2009) e Cui et al. (2017). A utilização destas duas abordagens são complementares, uma vez que a primeira oferece uma metodologia ampla para recursos em um datacenter, ao passo que a segunda é mais focada na estimativa do custo de energia elétrica neste ambiente.

RESULTADOS

Com relação ao desenvolvimento da ontologia proposta, apesar da grande aplicabilidade das ontologias existentes em diferentes cenários, a abstração proposta por elas não é suficiente para representar de maneira adequada recursos de computação. Sobretudo ao se considerar o critério de completude, apenas algumas consideram componentes de hardware na granularidade necessária para o desenvolvimento do CSB proposto. Outro critério importante, precisão, relacionado ao detalhamento necessário não é contemplado de maneira adequada. Na realidade, considerando o cenário tratado, nenhuma ontologia satisfaz de maneira completa todos os critérios. Isso acontece possivelmente por serem soluções que visam cenários mais amplos do que aqui considerado. Diante disso, foi proposta uma nova ontologia, denominada *Cloud Infrastructure Service Ontology* (CISO), visando contornar as limitações identificadas nas ontologias existentes.

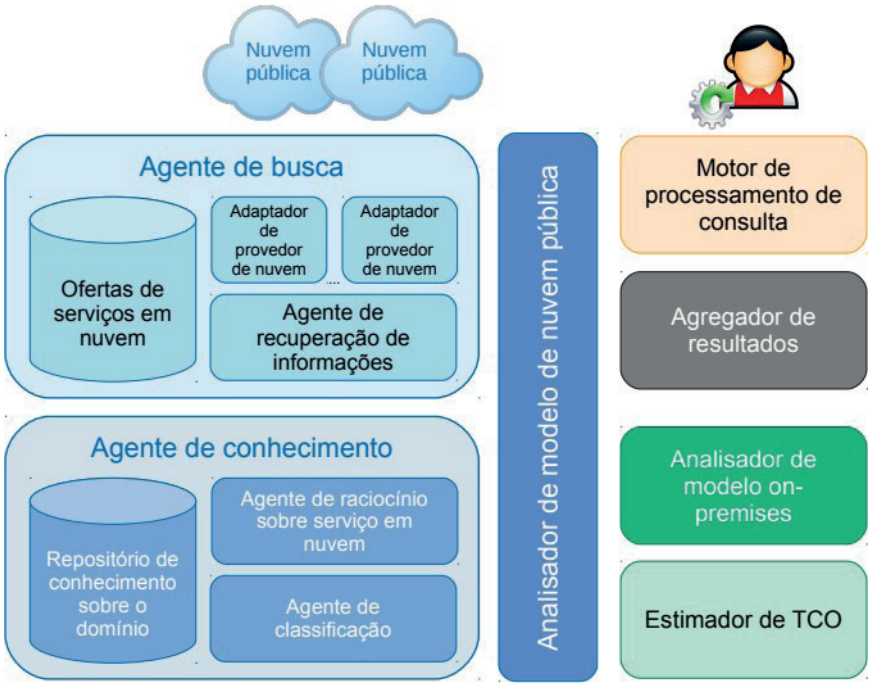
Figura 1. Estrutura da ontologia CISO.



A ontologia CISO foi desenvolvida na linguagem OWL. Como pode ser visualizado na Figura 1, sua estrutura inclui classes que representam os conceitos de mais alto nível como *Communication as a Service* (CaaS), *Data as a Service* (DaaS), *Compute Resource*, etc. O significado de cada classe é intuitivo e pode ser extraído da própria figura. Mas é importante ressaltar que a ontologia inclui os elementos principais necessários para a alocação de recursos de computação, os quais não estavam presentes de maneira adequada nas outras ontologias existentes analisadas.

A arquitetura proposta, ilustrada na Figura 2, é baseada nesta ontologia e a estratégia desenvolvida para estimativa de custos *on-premises*. Por simplicidade foi adotado como recurso apenas máquina virtual (*Virtual Machine* – VM). A interação com o cliente é realizada através do ***motor de processamento de consulta***. Por meio dele é fornecida a descrição dos recursos necessários, o que compreende características de hardware e quantidade de VMs solicitadas. As características desejadas são fornecidas através do uso de operadores lógicos e relacionais, o que possibilita atender a um conjunto mais amplo de necessidades. Com base nesta descrição, o componente gera a consulta numa linguagem interna que é repassada ao ***agregador de resultados***, que tem por finalidade consolidar os resultados obtidos considerando os diferentes modelos de implantação de nuvem.

Figura 2. Arquitetura da ferramenta proposta.



Atualmente, a ferramenta considera apenas nuvens públicas e ambientes *on-premises* mas a arquitetura foi pensada visando permitir sua futura expansão para outros modelos. A avaliação de custos considerando os dois modelos atualmente suportados é realizada pelo ***analisador de modelo de nuvem pública*** e pelo ***analisador de modelo on-premises***, com base na consulta gerada. A avaliação do custo de implantação *on-premise*s dos recursos pesquisados é realizada tendo como base a estratégia de estimativa de TCO. Por outro lado, devido à complexidade de se considerar múltiplos provedores de nuvem pública, o desenvolvimento da análise de custo para estes ambientes é mais complexo. Para isso, o componente responsável por esta tarefa utiliza dois componentes auxiliares. O primeiro componente, denominado ***agente de busca***, é responsável por descobrir os dados relacionados aos tipos de recurso nos diferentes provedores. Esta tarefa é realizada de maneira específica para cada provedor, através de ***adaptadores de provedor de nuvem***. Os resultados fornecidos por estes adaptadores são agregados pelo ***agente de recuperação de informações***, que os armazena na ***base de ofertas de serviços em nuvem***. A correspondência de termos e formação de uma nomenclatura comum é de responsabilidade do ***agente de conhecimento***. Seu principal sub-componente, ***agente de raciocínio sobre serviço em nuvem***, utiliza a ontologia proposta para raciocinar sobre as relações entre os serviços em nuvem. Para tal, ele faz uso do ***agente de classificação***, que auxilia na identificação de inconsistências e na preleção no caso de resultados equivalentes. Os resultados sobre o conhecimento são mantidos no ***repositório de conhecimento sobre o domínio***, a qual pode ser atualizado em face de novas consultas.

Como forma de demonstrar a ferramenta proposta foi desenvolvido um protótipo da arquitetura. O protótipo constitui uma aplicação web desenvolvida na linguagem Python. A implementação do agente de conhecimento se baseia na ontologia CISO. O desenvolvimento da ontologia, por sua vez, se baseia nas especificações *Resource Description Framework* (RDF) que faz parte de uma família de especificações da W3C. Dentre os formatos disponíveis para descrição de uma ontologia foi adotado o formato RDF/XML com utilização de propriedades da *Ontology Web Language* (OWL) (MCGUINNESS; HARMELEN et al., 2004).

Com relação ao agente de busca, uma vez que apenas um conjunto restrito de provedores de nuvem oferecem APIs de consulta aos serviços oferecidos, foram utilizadas técnicas de *web scraping* como forma de extrair as informações sobre recursos. Essa alternativa tem a vantagem de permitir a inclusão na ferramenta de qualquer provedor de nuvem. O protótipo atual da ferramenta considera dados dos provedores Amazon Web Services (AWS), Microsoft Azure, Google, Alibaba Cloud e Oracle. Foram utilizados estes provedores por serem os principais relatados pela Gartner em seu relatório Magic Quadrant for Cloud Infrastructure as a Service (BALA et al., 2020). No protótipo foi considerado apenas o modelo de cobrança sob demanda.

Para validar a ferramenta proposta, foi realizado uma avaliação experimental através da comparação com a consulta de tipos de VM e preços usando outras ferramentas. Mais precisamente, a ferramenta desenvolvida foi comparada com as ferramentas AWS Pricing Calculator, Azure TCO Calculator e Cloudorado Cloud Server Comparison. A escolha das duas primeiras alternativas se deve ao fato delas realizaram análise de TCO em conjunto a consultas de VM, ao passo que a ter- ceira foi escolhida por permitir consulta a múltiplos provedores de nuvem.

Foi adotado uma abordagem de experimento controlado, em que os participantes recebem uma única especificação de VM com características de hardware, de forma que eles possam utilizar as ferramentas analisadas dentro de um tempo experimental razoável. Este projeto de estratégia de experimento controlado é aproveitado para avaliar a interação dos usuários com a ferramenta proposta. A análise dessa interação possibilita a identificação de vantagens e limites da mesma, além de melhorias que podem ser introduzidas. O procedimento do experimento durou no máximo duas horas por participante e usaram o mesmo computador.

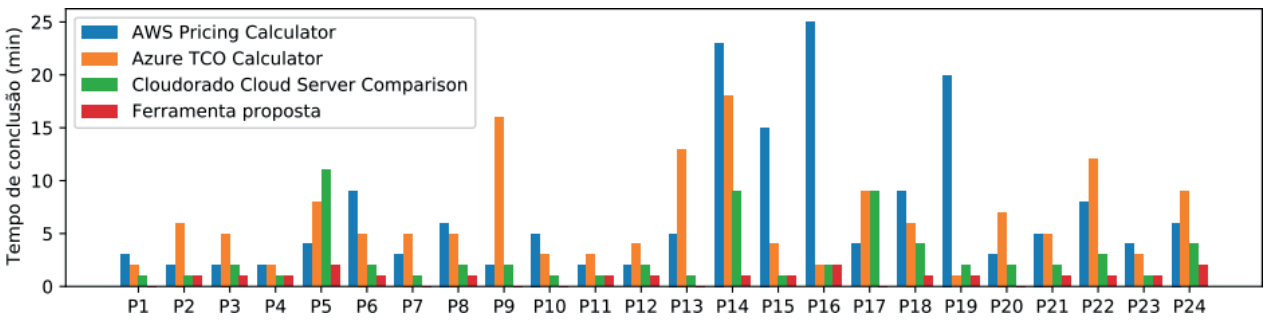
A definição das consultas utilizadas no experimento buscou considerar consultas típicas de um cenário real. Para tanto foram consideradas as necessidades de aplicações que são tipicamente implantadas em nuvem. Além disso, com base em Simmons (2017), foi adotado um critério de eficiência de Pareto utilizando como preferência o critério de eficiência de processamento, medido como “núcleo de CPU/$-hr”, eficiência de memória, medida como

"GB de RAM/$-hr" e eficiência de armazenamento, medida como "GB de disco/$-hr". Como resultado foi estabelecido um conjunto de especificações de hardware distintas, variando de 1 núcleo CPU, 1GB de RAM, sem disco, a 32 núcleos de CPU, 64GB de RAM, 1TB de disco.

Além dos resultados das buscas os participantes deveriam coletar o tempo gasto para obter os resultados. No final do experimento, o participante preenche um questionário sobre sua experiência em conceitos de computação em nuvem, seleção de serviços em nuvem, análise de TCO, linguagens e ferramentas de modelagem de seleção de recursos, além de *feedback* sobre a usabilidade e produtividade da ferramenta proposta e aspectos a melhorar. Os participantes foram recrutados entre alunos de Bacharelado em Sistemas de Informação do Instituto Federal de Goiás. Foi oferecido um incentivo à participação na forma de pontos extras na nota de uma disciplina, tendo sido recrutados 24 participantes com diversos níveis de especialização em programação e sistemas em nuvem, sendo 17 sem nenhum conhecimento sobre computação em nuvem, 6 com conhecimento básico e apenas 1 com conhecimento prático. Além disso, 10 deles relataram conhecimento sobre a análise de TCO e 14 nenhum conhecimento. Nenhum dos participantes havia usado as ferramentas antes.

A produtividade dos participantes foi avaliada calculando o tempo gasto para obter resulta- dos sobre a consulta de recursos. Este tempo é coletado individualmente para cada ferramenta e in- clui o período gasto na especificação do tipo de VM atribuído ao participante. A Figura 3 compara a produtividade de cada participante nas quatro abordagens. Para todos os participantes, a ferramenta proposta reduz significativamente o tempo de consulta. O que pode ser visto claramente nesta figura é que a ferramenta proposta possibilita que consultas sejam realizadas de maneira fácil, independente do participante. Enquanto o tempo para as outras ferramentas avaliadas possui uma distribuição com alta variação, para a ferramenta proposta a variação é pequena, com um máximo de 2 minutos para completar a consulta.

Figura 3. Tempo gasto pelos participantes para concluir a consulta usando as ferramentas.



A acurácia da seleção é avaliada calculando a precisão, recall e medida *F* considerando os resultados retornados por ferramenta, com relação à satisfação das consultas realizadas.

Como serviço de nuvem relevante foram considerados todos os tipos de VM que atendiam de maneira exata às consultas especificadas. Alguns tipos retornados pelas ferramentas AWS Pricing Calculator e Azure TCO Calculator foram classificados como irrelevantes por não coincidir com a configuração de hardware especificada na consulta. É importante ressaltar também que além dos provedores considerados foram incluídos os resultados de outros provedores (10 no total), suportados na ferramenta Cloudorado Cloud Server Comparison. Os resultados usando as métricas consideradas são apre- sentados na Tabela 2.

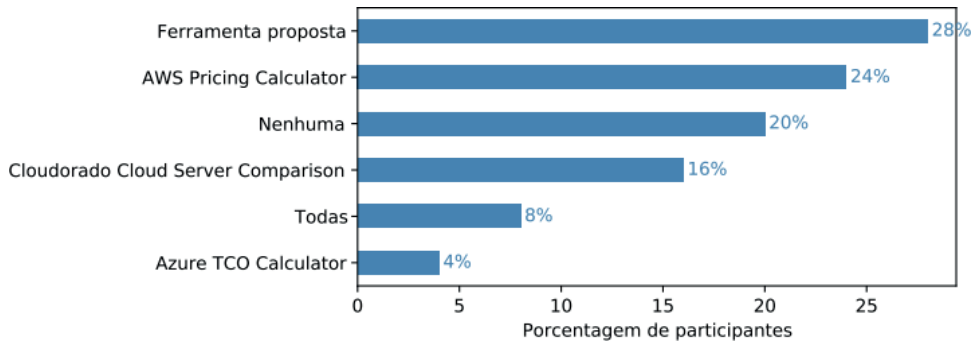
Tabela 2. Acurácia de seleção de recursos usando as ferramentas.

Ferramenta	Precisão	Recall	Medida *F*
AWS Pricing Calculator	0.375	0.001	0.001
Azure TCO Calculator	0.625	0.001	0.002
Cloudorado Cloud Server Comparison	1	0.019	0.037
Ferramenta proposta	1	0.986	0.993

Por considerar um único provedor, as duas primeiras ferramentas possuem acurácia muito baixa. Um motivo para isso é que grande parte dos resultados retornados foram considerados irrelevantes. Por outro lado, a ferramenta Cloudorado Cloud Server Comparison apresenta máxima precisão mas o recall e medida *F* são baixos, mesmo ela considerando mais provedores. Isso acontece porque é retornado um único resultado por provedor. Diferentemente, a ferramenta proposta retorna todos os tipos que satisfazem a busca, considerando as várias regiões de um mesmo provedor.

Apesar das similaridades com outros trabalhos, a solução aqui proposta visa preencher algumas lacunas no processo de consulta oferecido por estas soluções. O principal diferencial está no fato de permitir uma análise mais precisa sobre o custo de recursos ao se considerar múltiplos provedores (e suas diferentes regiões de atuação) em conjunto com uma avaliação de outros modelos de implantação. Com isso, espera-se que a ferramenta proposta permita oferecer indicadores mais precisos sobre os custos relacionados à alocação de máquinas virtuais, motivando sua adoção. Contudo, para verificar este aspecto, após a realização do experimento descrito na seção anterior, os participantes foram solicitados a indicar qual ferramenta ou estratégia usariam caso necessitasse encontrar o menor custo para implantar uma VM em nuvem. O resultado é apresentado na Figura 4. A ferramenta proposta foi indicada como a que seria adotada pela maioria (28%) dos participantes, ficando AWS Pricing Calculator como segunda opção. Curiosamente, grande parte dos participantes (20%) optou por não usar nenhuma das ferramentas diretamente mas sim realizar a busca em ferramentas de busca de propósito geral, provavelmente pela maior familiaridade com esta estratégia.

Figura 4. Feedback dos participantes sobre adoção da ferramenta.



Como última questão, foi solicitado que os participantes apresentassem em um formato livre críticas e sugestões sobre a ferramenta proposta. De maneira geral, as críticas foram positivas, indicando que a ferramenta possui interface intuitiva, é simples, clara e objetiva. Como pontos negativos, os participantes destacaram a ausência de detalhamento das informações retornadas. Como exemplo, indicaram a necessidade de explicitação da unidade monetária, assim como da unidade de tempo considerada na cobrança, além de descrição de quais provedores são considerados na busca.

Como é comum em projetos de estudos de casos experimentais, a validade externa (ou seja, a capacidade de generalizar os resultados) é naturalmente prejudicada em uma extensão, a fim de atingir maior validade interna (ou seja, validar a inferência de causa-efeito). No entanto, uma vez que um dos principais objetivos é facilitar a consulta por usuários sem conhecimentos prévios, e dado que as consultas foram realizadas considerando uma variedade grande de tipos, é possível afirmar que os resultados indicam uma validação da ferramenta proposta. É importante ressaltar porém alguns ameaças de validade como o fato dos próprios participantes coletarem o tempo, sem o uso de ferramentas automatizadas.

CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado o projeto e implementação de um CSB para consulta de custos de recursos de computação. Os resultados de um estudo de caso experimental sugerem que o nível elevado de usabilidade fornecido pela ferramenta resulta em melhorias significativas na produtividade da pesquisa e precisão na seleção de serviços. Como trabalhos futuros é sugerida a inclusão de novos provedores na ferramenta proposta, assim como a implementação de novos mecanismos de cobrança e modelos de nuvem. Em cima disso, uma análise mais aprofundada sobre os resulta- dos obtidos com a ferramenta também deve ser realizada.

A próxima etapa é colocar a ferramenta de análise de custos na prática dos ambientes de nuvem do cliente e tentar traçar regras de orientação da análise para realizar busca de recursos com alta eficiência econômica.

REFERÊNCIAS

1. ACHAR, R.; THILAGAM, P. S. **A broker based approach for cloud provider selection**. In: IEEE. *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.], 2014. p. 1252–1257.

2. AL-SAYED, M. M.; HASSAN, H. A.; OMARA, F. A. **Towards evaluation of cloud ontologies**. *Jour- nal of Parallel and Distributed Computing*, Elsevier, v. 126, p. 82–106, 2019.

3. ALFAZI, A.; SHENG, Q. Z., QIN, Y.; NOOR, T. H. **Ontology-based automatic cloud service catego- rization for enhancing cloud service discovery**. In 2015 IEEE *19th International Enterprise Distribu- ted Object Computing Conference*. p. 151-158), 2015.

4. AMAZON. **AWS Pricing Calculator**. [S.l.]: Disponível em <https://calculator.aws>. Acesso em 11 Set 2020, 2020.

5. BALA, R.; GILL, B.; SMITH, D.; WRIGHT, D.; JI, K. **Magic Quadrant for Cloud Infrastructure and Platform Services**. [S.l.]: Disponível em <https://www.gartner.com/doc/reprints?id=1-242R58F3&ct=200902&st=sb>. Acesso em 08 Set 2020, 2020.

6. BEHRENDT, M.; GLASNER, B.; KOPP, P.; DIECKMANN, R.; BREITER, G.; PAPPE, S.; KREGER, H.; ARSANJANI, A. **Introduction and architecture overview IBM cloud computing reference ar- chitecture 2.0**. *Draft Version V*, v. 1, n. 0, 2011.

7. CASTAÑÉ, G. G.; XIONG, H.; DONG, D; MORRISON, J. P. **An ontology for heterogeneous resour- ces management interoperability and HPC in the cloud**. *Future Generation Computer Systems*, Else- vier, v. 88, p. 373–384, 2018.

8. CLOUDORADO. **Cloud Server Comparison**. [S.l.]: Disponível em <https://www.cloudorado.com/ cloud_server_comparison.jsp>. Acesso em 11 Set 2020, 2020.

9. CUI, Y.; INGALZ, C.; GAO, T.; HEYDARI, A. **Total cost of ownership model for data center tech- nology evaluation**. In: IEEE. *2017 16th IEEE Intersociety Conference on Thermal and Thermomechani- cal Phenomena in Electronic Systems (ITherm)*. [S.l.], 2017. p. 936–942.

10. ELHABBASH, A.; SAMREEN, F.; HADLEY, J.; ELKHATIB, Y. **Cloud brokerage: A systematic survey**. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 51, n. 6, p. 1–28, 2019.

11. GOOGLE. **Google Cloud Pricing Calculator**. [S.l.]: Disponível em <https://cloud.google.com/pro- ducts/calculator>, Acesso em 18 Set 2020, 2020.

12. GROZEV, N.; BUYYA, R. **Inter-Cloud architectures and application brokering: taxonomy and survey**. *Software: Practice and Experience*, Wiley Online Library, v. 44, n. 3, p. 369–390, 2014.

13. HAN, T.; SIM, K. M. **An ontology-enhanced cloud service discovery system**. In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*. [S.l.: s.n.], 2010. v. 1, n. 2010, p. 17–19.

14. HOGAN, M.; LIU, F.; SOKOL, A.; TONG, J. **NIST cloud computing standards roadmap**. *NIST Spe- cial Publication*, v. 35, p. 6–11, 2011.

15. LENK, A.; KLEMS, M.; NIMIS, J.; TAI, S.; SANDHOLM, T. **What's inside the Cloud? An architec- tural map of the Cloud landscape**. In: IEEE. *2009 ICSE workshop on software engineering challenges of cloud computing*. [S.l.], 2009. p. 23–31.

16. LI, X.; LI, Y.; LIU, T.; QIU, J.; WANG, F. **The method and tool of cost analysis for cloud computing**. In: IEEE. *2009 IEEE International Conference on Cloud Computing*. [S.l.], 2009. p. 93–100.

17. MCGUINNESS, D. L.; HARMELEN, F. V. **OWL web ontology language overview**. *W3C recommen- dation*, v. 10, n. 10, p. 2004, 2004.

18. MICROSOFT. **Azure TCO Calculator**. [S.l.]: Disponível em <https://azure.microsoft.com/en-us/pri- cing/tco/calculator>. Acesso em 11 Set 2020, 2020.

19. MOSCATO, F.; AVERSA, R.; DI MARTINO, B.; FORTIŞ, T. F.; MUNTEANU, V. **An analysis of mosaic ontology for cloud resources annotation**. In: IEEE. *2011 federated conference on computer science and information systems (FedCSIS)*. [S.l.], 2011. p. 973–980.

20. PATINIOTAKIS, I.; VERGINADIS, Y.; MENTZAS, G. **Preference-based cloud service recommen- dation as a brokerage service**. In: *Proceedings of the 2nd International Workshop on CrossCloud Sys- tems*. New York, NY, USA: Association for Computing Machinery, 2014. (CCB '14). ISBN 9781450332330. Disponível em: <https://doi.org/10.1145/2676662.2676677>.

21. RIGHTCLOUDZ. **Cloud Service Providers Evaluation**. [S.l.]: Disponível em <https://rightcloudz.com/ RankCloudzOnline>. Acesso em 18 Set 2020, 2020.

22. SIMMONS, W. **How to select the most efficient AWS EC2 instance types using Pareto front analy- sis**. [S.l.]: Disponível em <https://read.acloud.guru/selecting-the-most-efficient-aws-ec-2-instance-types- using-pareto-front-analysis-3a5c81bae3a2>. Acesso em 08 Set 2020, 2017.

23. STALCUP, K. **$14.1 Billion in Cloud Spending to be Wasted in 2019**. [S.l.]: Disponível em <https:// www.parkmycloud.com/blog/cloud-spending/>. Acesso em 17 Set 2020, 2019.

24. TAHAMTAN, A.; BEHESHTI, S. A.; ANJOMSHOAA, A; TJOA, A. M. **A cloud repository and dis- covery framework based on a unified business and cloud service ontology**. In: IEEE. *2012 IEEE Eighth World Congress on Services*. [S.l.], 2012. p. 203–210.

25. TANKELEVICIENEA, L.; DAMASEVICIUSB, R. **Characteristics of domain ontologies for web ba- sed learning and their application for quality evaluation**. *Informatics in Education*, Institute of Ma- thematics and Informatics, v. 8, n. 1, p. 131, 2009.

26. VAKILI, M.; JAHANGIRI, N.; SHARIFI, M. **Cloud service selection using cloud service brokers: approaches and challenges**. *Frontiers of Computer Science*, Springer, v. 13, n. 3, p. 599–617, 2019.

11

“Sistema de apoio à prática assistida de programação por execução em massa e análise de programas

- Márcia Gonçalves de **Oliveira**
  UFES
- Matheus de Araújo **Nogueira**
  FAESA
- Elias Silva de **Oliveira**
  UFES

doi 10.37885/201102098

RESUMO

Assistir a prática de exercícios em um curso de programação, especialmente em turmas numerosas, demanda tempo e esforço de professores. Com o objetivo de auxiliar o professor na avaliação de exercícios de programação, desenvolvemos o *PCodigo*, um sistema integrado ao *Moodle* que oferece recursos para executar e analisar programas em Linguagem C. As principais contribuições do *PCodigo* para apoiar o trabalho docente e favorecer a aprendizagem de programação são as seguintes: executar programas em massa diariamente e oferecer recursos de análise de programas como o reconhecimento de classes de soluções, a identificação de soluções divergentes e a detecção de indícios de plágios. Os resultados mostram que o *PCodigo* se apresenta como um instrumento relevante de apoio ao trabalho de avaliação de programação.

Palavras-chave: Análise de Programas, Execução em Massa, *PCodigo*, Programação.

INTRODUÇÃO

A prática da programação de computadores como parte de um processo de aprendizagem tem sido considerada complexa tanto para alunos quanto para professores. Para os alunos, a programação é um conhecimento de difícil aprendizagem porque o desenvolvimento de programas é um processo que envolve a combinação de várias habilidades cognitivas. Para o professor, assistir e avaliar individualmente seus alunos na prática da programação dispendem muito tempo e esforço, principalmente em turmas numerosas. Dessa forma, não conseguindo fornecer *feedbacks* imediatos para muitos alunos, a tendência é que professores reduzam cada vez mais o número de exercícios aplicados. No entanto, isso pode afetar a aprendizagem dos alunos, uma vez que a programação, para ser bem aprendida, requer extensa prática de exercícios (INHATOLA, *et al.* 2010).

Atentando para a complexidade da aprendizagem de programação, nos últimos anos, muitas tecnologias têm sido desenvolvidas para apoiar o ensino, a aprendizagem e a avaliação de programação. Hoje temos, por exemplo, tecnologias para executar programas *online* (CAMPOS e FERREIRA, 2004), recomendar exercícios personalizados (OLIVEIRA*, et al.* 2013) e até para avaliação automática de exercícios de programação (MOREIRA e FAVERO, 2009; SOUZA *et al.* 2011). No entanto, ainda carecemos de tecnologias que de fato favoreçam a prática assistida de programação atendendo às demandas de avaliação em massa de exercícios de programação e de *feedback* imediato.

Há, porém, alguns sistemas de apoio à prática da programação que foram desenvolvidos com as finalidades de submissão, execução e avaliação de exercícios. Entre esses sistemas destacamos o sistema de submissão de exercícios BOCA (CAMPOS e FERREIRA, 2004; FRANÇA *et al*, 2011), o sistema *Pdetect* para análise e clusterização de programas suspeitos de plágios (MOUSSIADES e VAKALI, 2005) e um sistema de mapeamento de programas em vetores (OLIVEIRA*, et al.* 2014).

O BOCA é um sistema de internet para submissão de exercícios com autenticação, controle de tempo e disponibilização de resultados em tempo real (CAMPOS e FERREIRA, 2004). O BOCA tem sido aplicado no Brasil como apoio às competições de programação promovidas pela Sociedade Brasileira de Computação (SBC).

O processo de correção automática do BOCA consiste dos seguintes passos (CAMPOS e FERREIRA, 2004; FRANÇA *et al*, 2011): receber o programa-fonte, compilá-lo, gerar o programa executável, testar as entradas fornecidas, comparar a saída gerada com a saída do gabarito, avaliar e enviar um relatório de submissão.

Uma adaptação do sistema BOCA é o BOCA-LAB que oferece envio múltiplo de arquivos para compilação, a integração ao *Moodle* e uma infraestrutura para prover o balanceamento de carga entre diversos servidores (FRANÇA *et al*, 2011).

O *PDetect* é um sistema que faz análise e clusterização de plágios em exercícios de programação (MOUSSIADES e VAKALI, 2005). Para isso, recebe códigos-fontes, mapeia-os em um conjunto de *tokens* de uma linguagem de programação, identifica os pares suspeitos de plágio e os reúne em *clusters*, formados a partir de um parâmetro de corte do índice de similaridade (MOUSSIADES e VAKALI, 2005).

O *Núcleo de Avaliação Diagnóstica* (NAD) de (OLIVEIRA e OLIVEIRA, 2014) recebe programas e os transforma em vetores cujas dimensões são chamadas de componentes de habilidades (*Ci*). Cada componente de habilidade representa a frequência de ocorrência de palavras reservadas, símbolos, operadores, funções e indicadores de funcionamento.

Estendendo as ideias desses sistemas, desenvolvemos o *PCodigo*, um sistema integrado ao *Moodle* com recursos de execução em massa e de análise de programas que tem como finalidade auxiliar o professor na avaliação de exercícios. A execução em massa do *PCodigo* é aplicável a diferentes linguagens de programação. Já os recursos de análise de códigos foram desenvolvidos para a Linguagem C, embora possam ser adaptados para outras linguagens.

As principais contribuições do *PCodigo* para apoiar o trabalho docente e favorecer a aprendizagem de programação são, portanto: executar programas em massa diariamente de forma flexível e oferecer recursos para análise de programas como a identificação de classes de soluções, de soluções divergentes e de indícios de plágios.

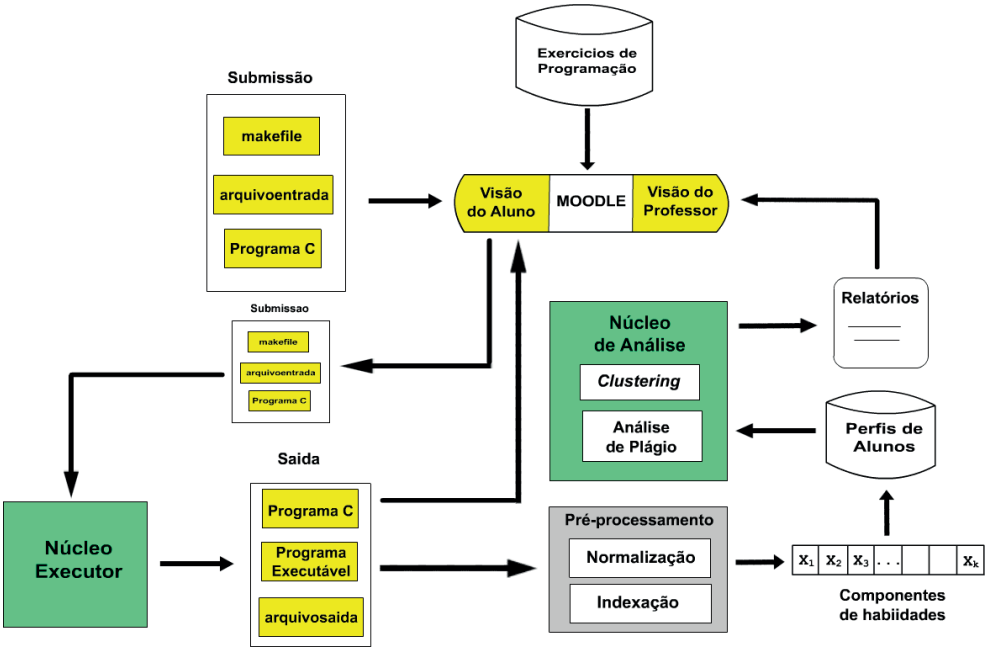
OBJETIVO

Auxiliar professores na avaliação de exercícios de programação por execução em massa de programas e análise de aprendizagem em códigos-fontes.

MÉTODO

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 e emite relatórios de avaliação para professores.

Figura 1. Arquitetura do *PCodigo*



A arquitetura do *PCodigo*, ilustrada na Figura 1, é formada por uma interface web do *Moodle*, um *Núcleo Executor*, um módulo de *Pré-processamento* e um *Núcleo de Análise*. O processamento do *PCodigo* se inicia após o professor disponibilizar via *Moodle*, na *Visão do Professor*, *Exercícios de Programação* e após os alunos iniciarem a *Submissão* de soluções para esses exercícios na *Visão do aluno*.

A *Submissão*, conforme a Figura 1, deve conter pelo menos um *Programa C*, um *Arquivoentrada* e um arquivo *Makefile* com as instruções de entrada, processamento e saída para execução de um programa. Uma vantagem do modelo de submissão com *makefile* é a flexibilização do formato de submissão, que permite o aluno gerenciar a execução do seu programa desde o recebimento da entrada até a geração da saída.

Quando a *Submissão* de um aluno chega ao *Núcleo Executor* do *PCodigo*, é gerada uma *Saída* contendo o *Programa C*, o programa *Executável* e o *Arquivosaida* com os resultados do *Executável*. O *Arquivosaida* pode ser visualizado pelo aluno na *Visão do Aluno* e pelo professor, nos *Relatórios* emitidos pelo *PCodigo* na *Visão do Professor*.

O *Núcleo Executor*, a cada três horas, executa de uma só vez todas as submissões encaminhadas pelo *Moodle* ou por outras interfaces *web* ao diretório de execução do *PCodigo*. As submissões que já foram executadas e as submissões que não seguem o padrão *PCodigo* de submissão não são processadas.

No módulo *Pré-Processamento* da Figura 1, os arquivos da *Saída* são submetidos a um processo de *Normalização* que gera em arquivo uma representação compacta de uma submissão. Na *Normalização*, retiramos os comentários e os textos entre aspas (*strings*) do programa-fonte substituindo-os pelos *tokens* *@coment* e *@textstr*, respectivamente.

Para retirar ambiguidades do código, substituimos os símbolos ambíguos (como o operador "*++*", que pode ser lido como dois operadores "+" na Linguagem C) por *tokens* precedidos por *@* (Exemplo: *@inc*, representando o operador *++*). Além disso, por conter a palavra reservada *int* da Linguagem C, substituimos o comando *printf* pelo *token @imprimir*. Finalizando a normalização, acrescentamos ao arquivo normalizado o *token @compila*, se o programa compilou e o *token @executa*, se o programa funcionou. Os programas da Figura 3 são exemplos de programas normalizados.

O arquivo gerado na *Normalização* é submetido à *Indexação*, que gera uma representação desse arquivo em vetor, onde cada dimensão é a frequência de ocorrência de cada *token* no arquivo normalizado. Na Figura 1, chamamos o vetor gerado de *Perfil de Aluno* e as suas dimensões, de *Componentes de habilidades* (OLIVEIRA e OLIVEIRA 2014).

É importante destacar que a retirada de *strings* e comentários na *Normalização* e a não contagem das variáveis na *Indexação* facilitam o reconhecimento de plágios, já que essas partes, em geral, são as partes mais alteradas por alunos para disfarçarem seus plágios.

No *Núcleo de Análise* da Figura 1, os *Perfis de Alunos* associados a um mesmo exercício de programação são reunidos em uma Matriz *A* para serem analisados e comparados. Através de algoritmos de *Clustering*, os vetores representantes das soluções dos alunos são reunidos em agrupamentos conforme as similaridades entre eles. Um exemplo de gráfico de *Clustering* gerado é apresentado na Figura 4.

No módulo de *Análise de Plágio* do *Núcleo de Análise* da Figura 1, os vetores da *Matriz A* são comparados dois a dois e é gerada uma Matriz *As*, formada pelos índices de similaridade entre cada par de vetores. Esses índices são calculados pela medida de similaridade *cosseno*, que é obtido pelo produto interno entre dois vetores. Os índices de similaridade variam de 0 (dissimilaridade total) a 1 (similaridade total).

Para análise do professor, o módulo de *Análise de Plágio* retorna em um arquivo apenas os pares de vetores com índices de similaridade acima de 0.9, isto é, com semelhanças entre si acima de 90%. Um exemplo desse relatório de plágio pode ser visualizado na Figura 2.

Após os processos de execução e análise, o *PCodigo* emite, na *Visão do Professor*, para cada exercício, *Relatórios* de todas as submissões. Através desses relatórios, o professor poderá tomar decisões de avaliações, bem como ter uma visão mais ampla de como seus alunos progridem individualmente na prática da programação.

RESULTADOS

O *PCodigo* foi desenvolvido em 2014 para turmas de programação dos cursos de Engenharia e de Computação da Universidade Federal do Espírito Santo.

Através do *Moodle* e de outras interfaces *web*, o *PCodigo* já recebeu cerca de 16.390 submissões de arquivos para 665 exercícios de programação.

Na fase inicial dos cursos de programação, os alunos tiveram algumas dificuldades com o formato de submissão do *PCodigo*, pois eles deveriam submeter, para cada exercício, pelo menos três arquivos: o *makefile*, o *arquivoentrada* e o programa em Linguagem C. Mas, à medida que os programas aumentaram em complexidade, os alunos compreenderam melhor as vantagens desse formato para testagem, especialmente, quando há entradas com grande quantidade de dados e vários casos de testes. Os relatórios do *PCodigo* são compostos pelos arquivos de saída gerados pela execução das submissões, por gráficos de clusterização (Figura 4) e por arquivos contendo os pares de soluções suspeitas de plágio na forma vetorial (Figura 2).

Para os professores, esses relatórios são muito úteis para o diagnóstico, comparação e classificação das soluções de programação desenvolvidas pelos alunos.

Execução em massa de exercícios de programação

Inicialmente, os programas submetidos pelos alunos ao *PCodigo* eram processados em um único servidor no próprio sistema operacional desse servidor. Todavia, logo percebemos as desvantagens dessa abordagem. Várias vezes tivemos de reiniciar o servidor por falta de espaço em disco em decorrência de códigos que geravam enormes arquivos de saída. Outro problema com o qual tivemos que lidar foi o de programas em *loop*. Nesse caso, os demais programas, na sequência, não eram processados. Como consequência desses e de muitos outros problemas na compilação e execução dos códigos submetidos de uma turma, adotamos o uso de máquinas virtuais distribuídas para o processamento em massa desses programas.

Em nossa estrutura, utilizamos um servidor, isto é, uma máquina *mestre*, para coordenar o particionamento das submissões entre as várias máquinas virtuais denominadas *escravas*. Dessa forma, uma máquina virtual pôde receber um ou mais conjuntos dos arquivos submetidos através do *Moodle* pelos alunos.

A máquina *mestre* possui um processador *AMD Phenom II X4 B95* com 3.0 Ghz e 8GB de memória RAM. Cada máquina *escrava* possui um processador Intel *Core-I7* com 2.80Ghz e 12GB de memória RAM.

A distribuição dos arquivos entre as máquinas *escravas* é realizada para se alcançar um balanceamento de carga. Com isso, tivemos como diminuir o tempo de execução do conjunto de códigos submetidos. Por outro lado, um código problemático, com essa nova arquitetura, passou a ficar retido em uma dada máquina virtual. Assim, quando é esgotado o tempo máximo para processamento do código, que é de cinco segundos, desativamos essa

máquina e o conjunto de arquivos enviados para essa máquina é marcado como encerrado de forma anormal.

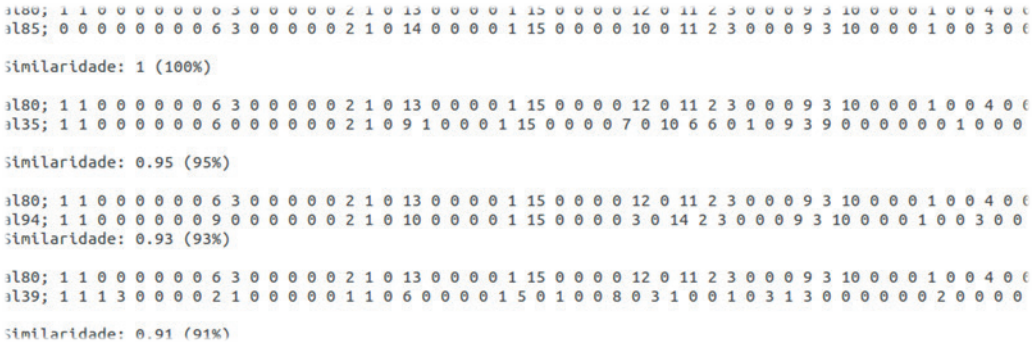
Essa nova forma de apoiar a prática de programação vem abrindo oportunidades para que alguns professores possam demandar dos alunos exercícios mais complexos. Além disso, os professores estão também desenvolvendo exercícios com tempo de processamento mais longo. Isso porque o tempo de execução de todos os códigos submetidos não está mais relacionado à soma de cada um deles, mas ao tempo máximo utilizado por uma das máquinas virtuais.

Análise de indícios de plágios

No relatório de plágio emitido pelo *Pcodigo*, conforme a Figura 2, são mostradas em representação de vetores as soluções de programação, que são as submissões dos alunos de programação. Essas soluções são identificadas, respectivamente, como *al80*, *al85*, *al35*, *al94* e *al39 na Figura 2*. Em relação à solução *a80, as soluções al85*, *al35*, *al94* e *al39 possuem* índices de similaridade acima de 90%.

Observamos também que a solução *al85*, com similaridade de 100% em relação à *al80*, conforme indicam os vetores de ambas na Figura 2, são praticamente idênticas. No entanto, elas se diferenciam principalmente em relação aos dois primeiros valores de seus vetores, que representam, respectivamente as componentes de habilidades *compila* e *executa*. Nessas componentes, o valor 1 indica que o programa funciona e 0, que não funciona. Nesse caso, embora as soluções sejam muito semelhantes, *al80* funciona e *al85* não funciona.

Figura 2. Análise de programas



Analisando a Figura 2, um professor de programação, considerando que as variáveis e *strings,* que são as partes que os alunos mais alteram para disfarçar plágios, não foram contadas na indexação e que a solução *al85* não funciona, suspeitaria que *al85* é um plágio de *al80*. Dessa forma, além de evidenciar fortes indícios de plágio entre as soluções, o *PCodigo* sinaliza o possível plagiador.

Ao comparar os códigos normalizados das soluções *al80* e *al85* na Figura 3, as suspeitas de plágio se evidenciam. Nesses códigos o professor avaliou o uso de estruturas de repetição e se o aluno resolveria, com apenas duas expressões lógicas em uma estrutura *if aninhada*, por número de pontos, o problema de identificar o campeão e o vice-campeão de um campeonato de futebol. A pontuação do professor variou de 0 a 5, sendo que seria descontado um ponto se o programa não compilasse.

Nessa análise de soluções da Figura 3, observa-se que *al85* (linhas 10-18) contém a mesma estrutura condicional de *al80* (linhas 12-21). As duas soluções utilizaram mais de duas expressões lógicas conforme os trechos de códigos da Figura 3. Para o professor, essa construção com excesso de expressões lógicas não está correta. Dessa forma, considerando que, em programação, é mais comum as soluções corretas serem mais semelhantes entre si do que as incorretas, as similaridades entre trechos incorretos trazem mais evidências de ocorrência de plágio.

Figura 3. Programas suspeitos de plágio

Solução al80

```
1.@compila   @funciona
2.#include <stdio.h>
3.#include <stdlib.h>

4. int main(int argc, char *argv[])
5. {
6.    int x=0, y1=0, y2=0, P, GP, GN,
7.        VF, VC, E;
8.    int T1=0, T2=0, T3=0;

9. @imprimir(@textostr, &GP, &GN, &VF, &VC, &E);
10. P = 5*GP - GN + 3*VF + 3*VC + E;
11.    T1 = P;
                ...

12.     if((T1>T2) @e (T1>T3))
13.     {
14.         @imprimir( @textostr );
15.         if(T2>T3){
16.                     @imprimir( @textostr );
17.         }
18.         else{
19.                     @imprimir( @textostr );
20.             }
21.     }              ...

22.  system( @textostr );
23.  return 0;
24. }
```

Solução al85

```
1. #include <stdio.h>
2. #include <stdlib.h>

3. int main(int argc, char *argv[])
4. {
5. int time1, time2, time3, P=0, GP, GN,
6.        VF, VC, E, T1=0, T2=0, T3=0;

7.  @imprimir(@textostr, &GP,&GN,&VF,&VC,&E);
8.  P = 5*GP - GN + 3*VF + 2*VC + E;
9.  T1=P;
                ...

10.   if((T1>T2) @e(T1>T3)){
11.             @imprimir( @textostr );
12.             if(T2>T3){
13.
@imprimir( @textostr );
14.                         }
15.                         else{
16.
@imprimir( @textostr );
17.                                 }
18.                                 }
                ...

19.       @imprimir( @textostr );
20.  return 0;
21. }
```

Conforme havíamos destacado, é comum o plagiador de códigos de programação fazer alterações na declaração das variáveis. Isso pode ser observado claramente na Figura 3, comparando as linhas 5-6 de *al85* com as linhas 6-8 de *al80*. Além dessas alterações, o autor da solução *al85* fez outras pequenas modificações, mas o seu programa não funcionou, conforme a ausência dos *tokens @compila* e *@executa* em seu código. Isso evidencia que

o autor de *a85*, além de não ter compreendido a construção de *al80*, não compreendeu as próprias modificações que fez, o que praticamente confirma sua fraude.

Nesse estudo de caso, o *PCodigo* identificou a similaridade de 100% entre as soluções *al80* e *al85*, mas o professor não. Prova disso é que o professor atribuiu nota 3.7 a *al80* e 2.7 a *al85* porque este último não funcionou.

As soluções *al80* e *al85* da Figura 3 são apenas dois exemplos de 100 amostras de exercícios de programação corrigidas manualmente por um professor. Seria, portanto, difícil para esse professor identificar todas as suspeitas de plágio e analisá-las. Por isso, o *PCodigo* mostrou-se como uma ferramenta adequada para auxiliá-lo na correção de exercícios oferecendo-lhe recursos de representação, visualização e análise de indícios de plágios.

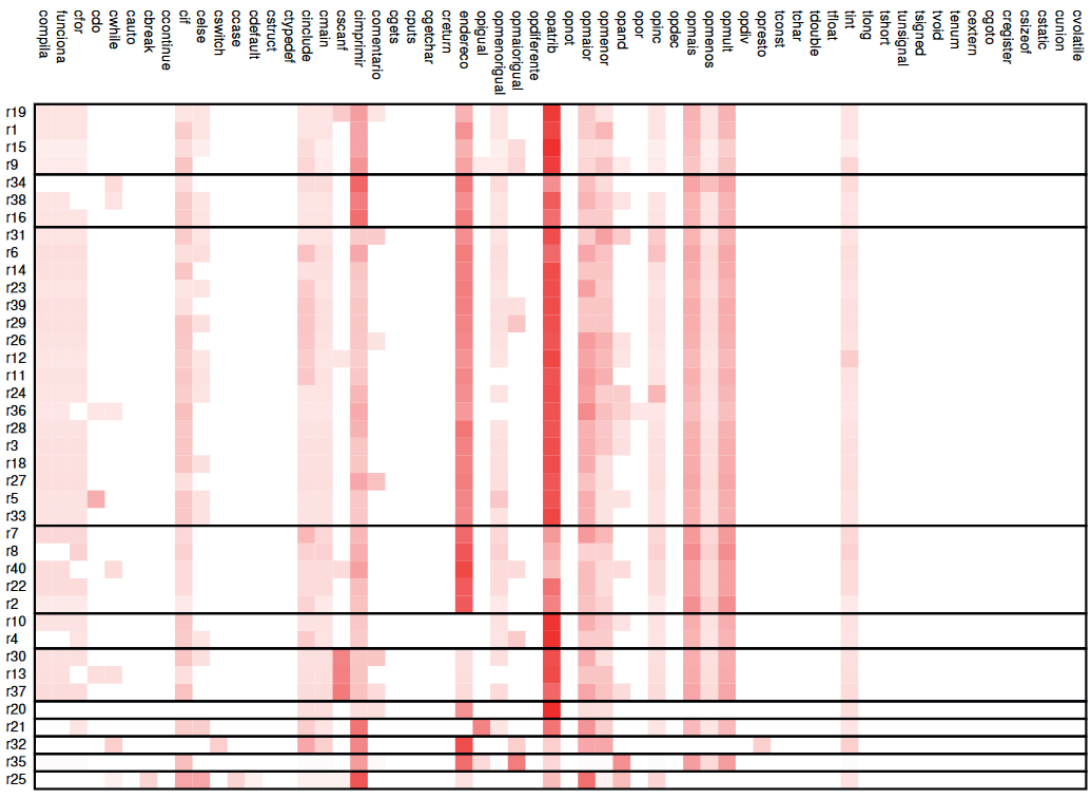
Reconhecimento de soluções divergentes e classes de soluções

O gráfico de *Clustering* da Figura 4 é emitido pelo *PCodigo* para auxiliar o professor na análise das soluções desenvolvidas por seus alunos. Nesse gráfico de *Clustering*, é possível visualizar simultaneamente cerca de 40 exercícios, verificar que instruções, estruturas e operadores da Linguagem C os alunos utilizaram e identificar as soluções que funcionam.

No gráfico da Figura 4, cada *cluster* é separado por uma linha horizontal preta. As linhas do gráfico são os vetores das submissões e as colunas, a frequência de ocorrência de *tokens* da Linguagem C. As tonalidades de cores em cada relação *Perfil x Componente de habilidade* indicam quão importante é uma *Componente de habilidade* em um *Perfil*. Dessa forma, quanto mais forte for a tonalidade, mais importante será a componente e, quanto mais próxima for da cor branca (de importância nula), menos importante ela será.

Observe no gráfico da Figura 4 que as componentes com tonalidades mais fortes em cada perfil de um mesmo *cluster* são as mesmas. Os algoritmos de *Clustering* apontam essas características como descritivas de um *cluster*.

Figura 4. Gráfico de *Clustering*



No gráfico da Figura 4, os *clusters* da parte inferior que contêm apenas uma solução, isto é, os *clusters* de *r25*, *r32*, *r35*, *r20* e *r21*, respectivamente de baixo para cima, isolam as soluções que divergem da solução de gabarito. Uma evidência disso é que todas elas receberam notas muito baixas em uma escala de 0.0 a 5.0. A Tabela 1 apresenta as características dessas soluções e explica por que elas foram consideradas divergentes.

Tabela 1. Análise de divergência de soluções

Análise de soluções		
Solução	**Nota**	**Caracterização**
r25	0.25	Excesso de instruções
r32	0.0	Programa muito curto sem coerência lógica
r35	0.0	Excesso de instruções
r20	0.0	Programa contendo apenas instruções iniciais
r21	0.25	Confusão nas expressões lógicas

A solução *r32* do gráfico de *clustering* da Figura 4, por exemplo, comparado à solução do gabarito na Figura 5, é uma solução divergente. Ao comparar as duas soluções, observa-se claramente que *r32* só apresenta alguma semelhança com o gabarito nas linhas iniciais, isto é, nas linhas 1 a 5. A partir daí, o autor de *r32* perdeu as rédeas no desenvolvimento de sua solução de programação.

Figura 5. Solução do gabarito e solução divergente

Gabarito

```
1. @compila   @funciona
2. #include<stdio.h>
3. #include<math.h>
4. main()

5. {
6.       int pnts, result = 0, tchamp = 0,
7.           pchamp = 0, tvice = 0, pvice = 0, i;
8.       for(i = 1; i@menorigual 3; i@inc)
9.       {
                  ...

10.      if(result @maiorigual pchamp)
11.      {
12.               pvice = pchamp;
13.               tvice = tchamp;
14.               pchamp = result;
15.               tchamp = i;
16.      }
17.      else if(result @maiorigual pvice)
18.      {
19.               pvice = result;
20.               tvice = i;
21.      }
22.      }                ...
23. }
```

Solução divergente r32

```
1. #include <stdio.h>
2. #include <math.h>

3. main ()
4. {
5. int gp,gn,vf,vc,e,camp,vice,times=0;
          ...

6. while(times@maiorigual10)
7. {
8. switch
9. }
```

O reconhecimento automático de classes de soluções e de soluções divergentes de um exercício é importante para descobrir as possíveis soluções para um problema e avaliar os casos atípicos de soluções. Além disso, as soluções mais representativas de cada *cluster* podem ser utilizadas como gabaritos de soluções para avaliação manual e até para geração de modelos de avaliação semi-automática de exercícios.

CONSIDERAÇÕES FINAIS

Este trabalho apresentou o *PCodigo* como um instrumento adequado de apoio à prática assistida de programação por oferecer recursos de execução em massa e de análise de exercícios submetidos por alunos via *Moodle*.

As principais vantagens do *PCodigo* em relação a outros sistemas de submissão de exercícios de programação são a flexibilização do formato de submissão com arquivo *makefile*, a execução distribuída de programas através de máquinas virtuais e os recursos de análise de plágios e de classificação de soluções.

Como trabalhos futuros a partir deste, sugerimos a integração dos recursos do *PCodigo* a outros ambientes virtuais e a criação de outros mecanismos de representação de códigos, como a representação de perfis por medidas que quantifiquem esforço e qualidade de programação.

Em resumo, a contribuição do *PCodigo* para a aprendizagem de programação é oferecer ao professor mais agilidade e melhor acompanhamento individual da aprendizagem e, ao aluno, *feedbacks* mais imediatos.

REFERÊNCIAS

1. CAMPOS, C. P.; FERREIRA, C. E. BOCA: um sistema de apoio a competições de programação (BOCA: A support system for programming contests). In: **Anais do Congresso da SBC-Workshop de Educação em Computação**. 2004.

2. OLIVEIRA, Márcia Gonçalves; CIARELLI, Patrick Marques; OLIVEIRA, Elias. Recommendation of programming activities by multi-label classification for a formative assessment of students. **Expert Systems with Applications**, v. 40, n. 16, p. 6641-6651, 2013.

3. DE SOUZA, Draylson Micael; MALDONADO, Jose Carlos; BARBOSA, Ellen Francine. *ProgTest*: An environment for the submission and evaluation of programming assignments based on testing activities. In: **2011 24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)**. IEEE, 2011. p. 1-10.

4. FRANÇA, Allyson Bonetti et al. Um sistema orientado a serviços para suporte a atividades de laboratório em disciplinas de técnicas de programação com integração ao ambiente Moodle. **RENOTE-Revista Novas Tecnologias na Educação**, v. 9, n. 1, 2011.

5. IHANTOLA, Petri et al. Review of recent systems for automatic assessment of programming assignments. In: **Proceedings of the 10th Koli Calling International Conference on Computing Education Research**. 2010. p. 86-93.

6. MOREIRA, Mireille Pinheiro; FAVERO, Eloi Luiz. Um ambiente para ensino de programação com feedback automático de exercícios. In: **Workshop sobre Educação em Computação (WEI 2009)**. 2009.

7. MOUSSIADES, Lefteris; VAKALI, Athena. *PDetect*: A clustering approach for detecting plagiarism in source code datasets. **The Computer Journal**, v. 48, n. 6, p. 651-661, 2005.

8. OLIVEIRA, Márcia G. et al. Análise de componentes latentes da aprendizagem de programação para mapeamento e classificação de perfis. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**. 2014. p. 134.

9. OLIVEIRA, Marcia; OLIVEIRA, Elias. Metodologia de Diagnóstico e Regulação de Componentes de Habilidades da Aprendizagem de Programação. In: **Anais do XXII Workshop sobre Educação em Computação**. SBC, 2014. p. 120-129.

12

“Softwares em metatranscriptoma: o exponencial impacto do big data na saúde humana

- Diego Lisboa **Rios**
  UFMG
- Pedro Gontijo **Carneiro**
  UFMG
- Silvia de Siqueira **Costa**
  UNIFENAS
- Cássio Siqueira Souza **Cassiano**
  UFSJ
- Thiago Araújo **Andrade**
  UFMG
- Welington Ribeiro Aquino **Marques**
  UFOP
- Paula Margarita Salazar **Torres**
  UFSJ
- João Batista **Matos Júnior**
  IFAC
- Lucas Roberto da **Silva**
  UFSJ
- Fabyola Antunes Gonçalves **Souza**
  UFOP

Artigo original publicado em: 2020.
Revista Eletrônica Acervo Saúde - ISSN 21782091.

Oferecimento de obra científica e/ou literária com autorização do(s) autor(es) conforme Art. 5, inc. I da Lei de Direitos Autorais - Lei 9610/98

doi 10.37885/201101943

RESUMO

Objetivo: Reconhecendo a importância dos estudos de microbioma para a saúde humana e além, nos esforçamos para fornecer uma revisão abrangente das tecnologias, algoritmos e *softwares* na análise metatranscriptômica existentes, especificamente os métodos de análise de dados. **Resultados:** Foram discutidas e comparadas as seguintes análises: a tecnologia de sequenciamento de alto rendimento; softwares de pré-processamento dos *reads* e análise de alta qualidade de dados; ferramentas e funções dos montadores de *novo*; atribuições das ferramentas de classificação de taxonomia*;* ferramentas de anotação funcional; desafios na análise de expressão diferencial de genes; e por último detalhes de fluxos de trabalho, recursos e as ferramentas específicas de bioinformática. **Considerações finais:** Atualmente, os estudos exigem um grande número de ferramentas para ajudar em todos os aspectos da análise de dados. Avanços em tecnologias de sequenciamento continuarão impulsionando o desenvolvimento da análise do metatranscriptoma. Um *workflow* padrão deve ser aceito para comparar novas ferramentas, o que ajudaria o progresso de trabalho mais precisos e apropriados. Apesar de alguns dos problemas com a metatranscriptômica como método, fica claro que a próxima geração de ferramentas será uma grande promessa para impulsionar e facilitar nossa compreensão da fração biologicamente ativa de microbioma das vias relevantes envolvidas.

Palavras-chave: Metatranscriptoma, Microbioma, Software, *Big Data*, RNAseq.

INTRODUÇÃO

Para além do corpo humano, comunidades de microrganismos são encontradas em diversos outros ambientes, como oceano, solo, plantas e outros animais (Li et al., 2017). Cada uma dessas comunidades microbianas tem suas próprias complexidades, diversidades e ecossistemas. Os membros das comunidades interagem entre si e cooperam com seus ambientes ou hospedeiros. Recentemente, com o desenvolvimento de tecnologias de sequenciamento que permitem aos pesquisadores estudar genomas da microbiota (também conhecido como microbioma), se descobriu que a microbiota humana está intimamente relacionada a várias doenças. Uma das áreas mais críticas é a pesquisa do microbioma humano associada a doenças humanas bem conhecidas, como obesidade, doença inflamatória intestinal (DII) e gêmeos magros ou obesos (Chen et al., 2018). Além disso, evidências crescentes indicam que a microbiota dentro do corpo humano, especialmente o microbioma intestinal, desempenha um papel fundamental na fisiologia humana (Liu et al., 2016).

Atualmente, vários métodos são aplicados para inferir diferentes níveis de informação sobre o microbioma. A análise *Whole genome sequencing* (WGS) usa informações de todos os genes para identificar taxonomicamente a comunidade microbiana em nível de espécie ou linhagem. Já a análise *shotgun* do transcriptoma completo da amostra, chamado de metatranscriptoma, permite a observação de padrões na expressão gênica e funcional de toda a comunidades microbiana. No entanto, apesar das diversas vantagens em se utilizar esse método, algumas desvantagens são observadas, sendo esta muito carente de *softwares* específicos para análise dos dados (Bushmanova et al., 2019).

Na busca por padrão e banco de dados mais curado, vários estudos de microbioma com base populacional foram criados. O *Human Microbiome Project*, foca no estudo de comunidades microbianas que habitam o corpo humano de indivíduos saudáveis, com ênfase nas áreas nasal, oral, cutânea, gastrointestinal e urogenital. Já o *Interactive Human Microbiome Project* se concentra na compreensão das interações humano-microbioma, por meio de estudos longitudinais que reúnem múltiplos conjuntos de dados ômicos do microbioma humano (Nurk et al., 2017). Além disso, outro projeto chamado Metagenômica do Trato Intestinal Humano (MetaHIT) se baseia na compreensão da relação entre a microbiota intestinal humana e a os benefícios à saúde (Ugarte et al., 2018). Não obstante, o MetaHIT também estuda obesidade e DII. Por fim, o *Earth Microbiome Project* (EMP) se concentra na caracterização da diversidade, distribuição e estrutura dos ecossistemas microbianos em todo o planeta e já reuniu mais de 30.000 amostras de diversos ecossistemas, incluindo humanos, animais, plantas terrestres, marinhas, ambientes reconstruídos e entre outros (Bushmanova et al., 2019). EMP é um dos projetos pioneiros de microbioma a definir alguns protocolos padrão para outros estudos de microbioma.

Com as limitações crescentes na compreensão dos mecanismos de um microbioma individual, e em escala global, além das dificuldades associadas à cultura de espécies microbianas individuais, a metatranscriptômica têm sido usada com mais frequência em estudos recentes (Tarazona et al., 2015). Reconhecendo a importância dos estudos de microbioma para a saúde humana e além, nos esforçamos para fornecer uma revisão abrangente das tecnologias, algoritmos e *softwares* na análise metatranscriptômica existentes, especificamente os métodos de análise de dados. Esperamos que as informações fornecidas ajudem mais pesquisadores a identificar as ferramentas apropriadas para seus estudos de microbioma em seus respectivos projetos.

REVISÃO DE LITERATURA

Análise bioinformática de dados de sequenciamento metatranscriptômico

Em virtude da complexidade do microbioma, o sequenciamento de alto rendimento na forma de *reads* relativamente curtas geralmente são geradas a partir da tecnologia de sequenciamento Illumina. Tem sido mais frequentemente aplicado para estudos de metatranscriptoma, particularmente quando são necessárias várias amostras e cobertura profunda, como nos casos de estudos de expressão gênica diferencial. Uma vez que, *a priori*, a maioria das informações sobre as amostras são desconhecidas, assim como sua composição microbiana, abundância relativa de membros da comunidade, tamanhos do genoma e expressão relativa dentro e entre os genomas, não é trivial encontrar parâmetros experimentais corretos, como profundidade de sequenciamento para metatranscriptômica. Embora o sequenciamento de reads mais longas possa produzir mRNAs completos ou quase completos que podem ajudar a discriminar entre diferentes isoformas (Celaj et al., 2014), e fornecer trechos mais longos de sequência para pesquisas de similaridade e às várias tecnologias de leitura longa atualmente apenas desempenham um papel de apoio que não estão sendo ativamente usadas sozinhas para estudos de metatranscriptoma. Nessa revisão, nos concentramos nas ferramentas e fluxos de trabalho disponíveis para processamento e análise de dados de metatranscriptoma, que se concentram em dados de reads curtas.

PRÉ-PROCESSAMENTO

Semelhante a outros conjuntos de dados NGS, uma das primeiras etapas no processamento de dados de RNASeq é executar o Controle de Qualidade (QC) e remover ou aparar leituras falsas ou errôneas. São várias as ferramentas para fazer o QC das reads de RNASeq (Tabela 1).

TABELA 1. SOFTWARES DE PRÉ-PROCESSAMENTO E DE ANÁLISE DA QUALIDADE DOS DADOS

Software	Função	Autor
FastQC	Entre as mais utilizadas atualmente, usadas para *reads* curtas derivadas de sequenciadores Illumina.	(Andrews, 2010)
FaQCs	Entre as mais utilizadas atualmente, usadas para *reads* curtas derivadas de sequenciadores Illumina.	(Lo e Chain, 2014)
Fastp	Entre as mais utilizadas atualmente, usadas para *reads* curtas derivadas de sequenciadores Illumina.	(Chen et al., 2018)
Trimmomatic	A mais utilizada atualmente, usadas para *reads* curtas derivadas de sequenciadores Illumina.	(Bolger et al., 2014)
SortMeRNA	Remoção de rRNAs.	(Kopylova et al., 2012) *
Barrnap	Remoção de rRNAs.	(Seemann, 2014) *
BMTagger	Pesquisam *kmers* humanos específicos nas *reads.*	(Rotmistrovsky e Agarwala, 2011)
-	*Reads* podem ser removidas usando métodos tradicionais de mapeamento que marcam e removem *reads* que mapeadas no genoma humano.	(Li et al., 2017)

* Uma das etapas mais importantes que devem ser levadas em consideração é a remoção ou depleção física dos transcritos de RNA ribossômico altamente abundante (rRNA) das amostras, pois eles geralmente representam mais de 90% de todos os dados gerados no sequenciamento, geralmente removidos usando técnicas moleculares antes do sequenciamento, mas sua abundância nas amostras resulta em uma certa quantidade de contaminação por nas *reads*.

A MONTAGEM DE NOVO

As leituras pré-processadas e de alta qualidade agora podem ser montadas em transcrições putativas, *contigs,* usando montadores *de novo*. Dado que a maioria dos microbiomas não é adequado para o alinhamento caracterizado pelos genomas de referência, os montadores *de novo* fornecem uma solução de para a referência, representando segmentos de genoma mais longos e expressos que podem fornecer um conjunto de genes de referência. Isso fornece aos pesquisadores a capacidade de encontrar homólogos de maneira mais direta, estabelecer origem taxonômica e servir como referência para mapeamento e análise de expressão (Tabela 2).

TABELA 2. MONTADORES *DE NOVO* ATUAIS

Software	Função	Autor
MEGAHIT		(Li et al., 2015)*
IDBA-UD	Montadores metagenômicos projetados para lidar com metagenomas complexos que podem compartilhar alguma semelhança de sequência em regiões altamente conservadas, mas podem variar em termos de abundância relativa dentro do microbioma e também podem abrigar variação populacional (nível de deformação).	(Peng et al., 2012)*
metaSPAdes		(Nurk et al., 2017)*

Software	Função	Autor
Trans-ABySS	Montadores que tentaram resolver os problemas na análise de metatranscriptomas, mas foram originalmente projetados para montar transcriptoma de um único organismo.	(Robertson et al., 2010)**
Trinity		(Grabherr et al., 2011)**
BinPacker		(Liu et al., 2016)**
Oásis		(Schulz et al., 2012)**
SOAPdenovo- Trans		(Xie et al., 2014)**
IDBA-Tran		(Peng et al., 2012)**
rnaSPAdes		(Bushmanova et al., 2019)**
IDBA-MT	Montadores de novo projetados especificamente para metatranscriptomas que levam em conta as características únicas dos transcritos e a natureza complexa das comunidades microbianas.	(Leung et al., 2013)***
IDBA-MTP		(Leung et al., 2014)****
Transcript Assembly Graph (TAG		(Ye e Tang, 2016)*****

* No entanto, a eficácia desses montadores na reconstrução de transcritos que possuem peculiaridades próprias, como íntrons e éxons, diferentes isoformas e RNAs não codificadores mais curtos (ncRNA), raramente são testados; portanto, é recomendável que se use montadores específicos para dados metagenômica em conjuntos de dados de metatranscriptoma.

** Tentaram explicar os problemas no sequenciamento de transcriptomas, mas foram originalmente projetados para montar transcrições de um único organismo. Apesar de seu design para conjuntos de dados transcriptômicos e não metatranscriptômicos, as comparações entre alguns montadores mostraram que, em geral, os montadores testados Oases, Trinity e Metavelvet melhoraram o número de genes anotados nos contigs resultantes, com o montador Trinity superando os demais (Celaj et al., 2014).

*** Baseado no IDBA-UD, usa-se múltiplos valores em um gráfico *de Bruijn*, enquanto considera-se os recursos associados aos mRNAs, como profundidade de sequência desigual e padrões comuns de repetição em diferentes mRNAs, diminuindo assim a taxa de erros de montagem (Leung et al., 2013).

**** Derivado do IDBA-MT para poder montar mRNAs de baixa expressão. Ele usa as informações de sequências de proteínas conhecidas para orientar a montagem, começando com *k-mers* menores para construir sequências de mRNA que são então incluídas com base na sua similaridade (Leung et al., 2014).

*****. Montador que, comparativamente, também usa um gráfico *de Bruijn*, mas para montar o metagenoma correspondente, que é usado como referência para posteriormente mapear as leituras do metatranscriptoma e reconstruir sequências de mRNA. É ineficaz para uso em microbiomas que, além de bactéria, também englobam alguns fungos. Além disso, existe a suposição implícita entre o metagenoma da comunidade para que todos os genes expressos possam ser mapeados (Ye e Tang, 2016).

O estado atual da montagem *de novo* para conjuntos de dados metatranscriptômicos ainda está em estágios iniciais. Apenas algumas ferramentas foram desenvolvidas especificamente para a metatranscriptômica, mas sua eficácia em diversos conjuntos de dados não

foi testada e seu hardware ou requisitos de memória em uma variedade de complexidades da comunidade e volume de dados também não foram rigorosamente estabelecidos.

TAXONOMIA DOS TRANSCRITOS.

Semelhante ao perfil taxonômico, que é frequentemente realizado com dados metagenômicos de *shotgun*, pode-se usar o mesmo conjunto de ferramentas para realizar atribuições taxonômicas baseadas em *reads* ou *contigs*, a fim de entender quais os organismos estão expressando ativamente em determinada amostra. Um método separado e distinto é focar apenas nos rRNAs para avaliar membros ativos de uma comunidade, embora, como mencionado acima, eles sejam frequentemente removidos (tanto nos protocolos de laboratório quanto no pré-processamento dos dados brutos).

Ferramentas de classificação de taxonomia baseadas em *reads*, como Kraken (Wood e Salzberg, 2014), GOTTCHA (Freitas et al., 2015), MetaPhlan2 (Truong et al., 2015) etc. podem ser usadas para metatranscriptomas (Neves et al., 2017). Como essas ferramentas funcionam em sequências curtas e se baseiam em combinações de nucleotídeos, seu uso efetivo é limitado a microrganismos, mas cujos os membros possuam vizinhos próximos às sequência existentes nos bancos de dados que foram montadas em *contigs* mais longos e possivelmente transcritos completos podem ser usados por várias ferramentas, como Centrífuge (Kim et al., 2016) e Kraken 2 (Wood e Salzberg, 2014), para identificar potencialmente uma maior proporção dos membros da comunidade sequenciados.

As atribuições taxonômicas que utilizam *reads* ou regiões de codificação previstas têm um grande número de limitações, incluindo os algoritmos necessários para processar grandes volumes de dados ou acomodar sequências curtas e a escassez de citações nos bancos de dados de referência. Para compor essas questões, está o fato de que a maioria das ferramentas de bioinformática utilizam-se de apenas um subconjunto de genomas disponíveis ou se concentra em certos organismos. Por exemplo, muitas ferramentas não possuem eucariotos como parte de seus bancos de dados. Houveram alguns esforços recentes, como o desenvolvimento de novas ferramentas e melhorias nas ferramentas existentes para incluir genomas eucarióticos em seus bancos de dados, como MetaPhlan2 e Kaiju (Truong et al., 2015), porém sua eficácia na classificação de eucariotos é desconhecida. Além disso, muitas vezes é difícil discernir acerca de baixa abundância de acertos falsos positivos, o que é um problema inato nos estudos de microbioma. A falta de conhecimento sobre a diversidade microbiana geral ou em qualquer sistema biológico também pode limitar a utilização das ferramentas de classificação da taxonomia.

ANOTAÇÃO FUNCIONAL

Um dos principais objetivos da metatranscriptômica é avaliar a atividade funcional de uma comunidade microbiana. Como os transcritos expressos representam um *proxy* para o fenótipo real, caracterizar a função dos transcritos é uma tarefa fundamental para a metatranscriptômica. A anotação funcional pode ser realizada usando *reads* ou *contigs* montados.

Os perfis funcionais baseados em *reads*, como MetaCLADE (Ugarte et al., 2018), HMM-GRASPx (Zhong et al., 2016) e UProC (Meinicke, 2015) utilizam bancos de dados específicos e exigem frames de leitura abertos previstos como entrada de outras ferramentas, como FragGeneScan (Rho et al., 2010). O MetaCLADE é uma das ferramentas mais recentes e usa um banco de dados que consiste em 2 milhões de modelos probabilísticos derivados de 15.000 domínios da Pfam, portanto centenas de modelos representando um único domínio, para abranger a diversidade de cada domínio na árvore da vida. Uma pesquisa nesse banco de dados resulta em um grande número de ocorrências por leitura que são filtradas com base em redundância, probabilidade e pontuação de *bits* (Ugarte et al., 2018).

De maneira alternativa, a anotação de genes pode ser realizada a partir de *contigs* montados. A anotação de transcritos montados é semelhante às anotações de genomas e metagenomas. A descoberta de genes usando programas como Prodigal (Hyatt et al., 2010) e FragGeneScan (Rho et al., 2010) é seguida por atribuição funcional com base em pesquisas de similaridade usando ferramentas como DIAMOND para pesquisar sua funcionalidade em bancos de dados, como KEGG, NCBI RefSeq, UniProt, entre outros (Buchfink et al., 2015). Outras ferramentas, pipelines e plataformas abrangem uma variedade de utilitários de bioinformática (incluindo busca e anotação de genes), como o Prokka (Seemann, 2014), EDGE Bioinformatics (Li et al., 2017) e MG-RAST (Wilke et al., 2016) que combinam várias pesquisas de similaridade em vários bancos de dados ou podem até unir a montagem, identificação de genes e anotação por meio de pesquisas de similaridade. Depois que as anotações são executadas, as funções enzimáticas também podem ser mapeadas para as vias metabólicas conhecidas, usando ferramentas como MinPath (Ye e Doak, 2009) ou iPath (Yamada et al., 2011).

ANÁLISES DE EXPRESSÃO DIFERENCIAL

Além da descrição simples de quem são os membros ativos e quais genes estão sendo expressos em um único momento ou amostra, estão os estudos de expressão diferencial de genes, nos quais a metatranscriptômica pode ser usada para comparar diferentes condições, parâmetros ambientais e seus efeitos na comunidade. Existem muitas ferramentas originalmente desenvolvidas para uso de genomas únicos que possam ser aproveitadas para

estudos de expressão gênica diferencial metatranscriptômica. Essas ferramentas requerem como dados a abundância de entrada por gene (ou transcrito) e por amostra (representando a expressão sob uma condição específica ou um momento específico). A abundância pode ser obtida de várias maneiras, mas geralmente envolvem algumas formas de alinhamentos/mapeamentos das *reads* para um genoma de referência, um conjunto de referência ou um conjunto de genes de referência.

O programas EdgeR (Robinson et al., 2010), DeSeq2 (Love et al., 2014) e LIMMA (Ritchie et al., 2015) são pacotes do R que são frequentemente usados, juntamente com as informações de abundância, para identificar genes que são estatisticamente diferencialmente expressos entre várias amostras. Da mesma forma, ferramentas como Análise do Conjunto de Genes (GAGE) podem ser usadas para identificar caminhos enriquecidos em uma condição sobre outra (Li et al., 2017). Como as amostras de metatranscriptômica replicadas não são triviais comparadas aos estudos transcriptômicos com organismos isolados, métodos não paramétricos como a implementação no NOISeq (Tarazona et al., 2015) também devem ser considerados.

Existem peculiaridades nas análises metatranscriptômicas que tornam as análises de expressão diferencial bastante desafiadoras, principalmente o resultado do sequenciamento de uma grande diversidade de transcritos (de uma ampla variedade de organismos). Problemas como genes compartilhados entre organismos intimamente relacionados e variação na composição taxonômica dos transcritos podem resultar em avaliação incorreta dos perfis de expressão gênica. Recentemente, foi proposto um método de normalização que pode minimizar a influência da diversidade taxonômica na amostra, normalizando os dados de contagem com base na composição taxonômica em diferentes amostras, porém esse método também é influenciado pela representação em bancos de dados taxonômicos (Klingenberg e Meinicke, 2017).

PIPELINES - FLUXOS DE TRABALHO DISPONÍVEIS PARA ANÁLISE METATRANSCRIPTÔMICA

A análise de um conjunto de dados de metatranscriptoma está repleta de etapas bioinformáticas com muitas opções de ferramentas para qualquer etapa. Quais etapas e ferramentas devem ser selecionadas são frequentemente escolhidas pelos objetivos do experimento, cujos os detalhes podem crescer em complexidade com base nas especificidades do estudo. No entanto, existem fluxos de trabalho em bioinformática, chamados de *pipelines* ou *workflow*, que visam otimizar parte dessa complexidade conectando várias ferramentas individuais que transformam *reads* brutas de sequenciamento e as processam, fornecendo arquivos de dados que representam os resultados das saídas, caracterizando identidades

taxonômicas, genes funcionais e transcritos diferencialmente expressos. Os detalhes de oito fluxos de trabalho disponíveis, seus recursos e as ferramentas específicas de bioinformática que eles usam podem ser encontrados como um resumo na Tabela 3.

TABELA 3. *PIPELINES* DE METATRANSCRIPTÔMICA E SEUS RECURSOS.

		Baseado em Reads					Baseado em montagem		
		MetaTrans	COMAN	FMAP	SAMSA2	HUMAnN2	SqueezeMeta	IMP	MOSCA
Processamento	QC	√	√	√	√	×	√	√	√
	Remoção das reads do hospedeiro	×	×	√	×	×	×	√	×
	Remove rRNA	√	√	×	√	×	√	√	√
Montagem de novo		×	×	×	×	×	√	√	√
Binning		×	×	×	×	×	√	√	×
Perfil Taxônomico	Reads	√	√	×	√	√	×	×	×
	Contigs	×	×	×	×	×	√	√	√
Anotação Funcional	Reads	√	√	√	√	√	×	×	×
	Contigs	×	×	×	×	×	√	√	√
Analise Pathway		√	√	√	×	√	√	√	×
Precisa de Metagenoma		×	×	×	×	×	×	√	×
Reporta resumos		×	×	×	×	×	×	√	×
Interface Web		×	√	×	×	×	×	×	×
Comparação de múltiplas amostras		√	√	√	√	√	√	×	√
Expressão Diferencial		√	√	√	√	×	×	×	√
Docker		×	×	×	×	√	×	√	√
Conda		×	×	×	×	√	×	√	×
Suporta reads longas		×	×	×	×	×	√	×	×
Repositório de código público.		√	×	√	√	√	√	√	√

Em comparação às análises baseadas em leitura, os fluxos de trabalho baseados em montagem abrigam uma etapa analítica extra. As análises são reunidas primeiramente em *contigs* maiores, ajudando a reduzir o tamanho dos dados que precisam ser processados para análises posteriores e aumentando a quantidade contínua do comprimento das transcrições expressas.

Como um dos principais objetivos das análises de metatranscriptoma é obter uma quantificação relativa da expressão gênica, todos os fluxos de trabalho baseados em leitura e em montagem fornecem alguma forma de cobertura por gene ou métrica de abundância (por exemplo, contagem bruta por gene ou número de leituras por kb por milhão de leituras sequenciadas). Esses valores de abundância podem usar de ferramentas adicionais para comparar a expressão relativa de genes entre condições de crescimento ou durante experimentos ao longo do tempo, cujo o objetivo é frequentemente ajudar a entender quais genes e caminhos podem ser importantes para um fenótipo específico em estudo. Para esses tipos de estudos, muitas vezes são necessárias experiências com replicação para obter resultados estatisticamente significativos, portanto, as comparações relativas de abundância de genes costumam ser uma comparação entre muitas amostras diferentes que incluem várias repetições biológicas.

A disponibilidade de vários fluxos de trabalho permite que os usuários escolham o que é mais apropriado para analisar seu metatranscriptoma. Embora os usuários devam selecionar fluxos de trabalho idealmente com base na capacidade/funcionalidade e qualidade dos algoritmos/ferramentas utilizados, considerações adicionais podem incluir os requisitos de recursos computacionais que variam entre os fluxos de trabalho e a frequência de manutenção ou desenvolvimento ativo do código-fonte, que podem sofrer modificações frequentes à medida que novos avanços, ferramentas ou métodos continuam sendo desenvolvidos. A Tabela 3 é uma compilação desses fluxos de trabalho disponíveis e pode ser usada como um guia em potencial para escolher um fluxo de trabalho com base em fatores importantes para abordar as perguntas de qualquer pesquisador. Por exemplo, se a análise de expressão diferencial é o objetivo de um estudo, a lista de fluxos de trabalho para escolher é limitada a cinco.

CONSIDERAÇÕES FINAIS

Atualmente, os estudos metatranscriptômicos são realizados principalmente com tecnologia de *reads* curtas (ou seja, Illumina), exigindo um grande número de ferramentas analíticas para ajudar em todos os aspectos da análise de dados. Nesta revisão, destacamos alguns dos principais métodos de análise de dados de metatranscriptômica, o que impacta diretamente no conhecimento humano da influência do microbioma na saúde humana. Algumas das ferramentas específicas de bioinformática usadas para realizar essas análises

e alguns fluxos de trabalhos metatranscriptômicos mais complexos combinam várias dessas ferramentas para abordar questões biológicas com esforço mínimo de usuário menos experientes. Conquanto, para tornar as coisas mais complexas, novos avanços em tecnologias de sequenciamento continuarão impulsionando o desenvolvimento de novas ferramentas e fluxos de trabalho. Um *pipeline* ou *workflow* padrão, uma estrutura, deve ser aceita para comparar novas ferramentas, o que ajudaria no progresso de trabalho e possivelmente se uniria a fluxos de trabalho mais precisos e apropriados. Apesar de alguns dos problemas com a metatranscriptômica como método, o desenvolvimento contínuo de novas ferramentas e algoritmos para análise de dados metatranscriptômicos, juntamente com nossa crescente compreensão dos desafios apresentados por esses conjuntos de dados, fica claro que a próxima geração de ferramentas de metatranscriptômica são uma grande promessa para impulsionar e facilitar nossa compreensão da fração biologicamente ativa de microbiomas das vias relevantes envolvidas.

REFERÊNCIAS

1. Andrews, S. (2010). FastQC: a quality control tool for high throughput sequence data.
2. Bolger, A. M., Lohse, M., Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30, 2114–2120. doi: 10.1093/bioinformatics/btu170
3. Buchfink, B., Xie, C., Huson, D. (2015). Fast and sensitive protein alignment using DIAMOND. Nat. Methods 12, 59–60. doi: 10.1038/nmeth.3176
4. Bushmanova, E., Antipov, D., Lapidus, A., Prjibelski, A. D. (2019). rnaSPAdes: a de novo transcriptome assembler and its application to RNA-Seq data. GigaScience 8. doi: 10.1093/gigascience/gi2100
5. Celaj, A., Markle, J., Danska, J., Parkinson, J. (2014). Comparison of assembly algorithms for improving rate of metatranscriptomic functional annotation. Microbiome 2, 39. doi: 10.1186/2049-2618-2-39
6. Chen, S., Zhou, Y., Chen, Y., Gu, J. (2018). fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics 34, i884–i890. doi: 10.1093/bioinformatics/bty560
7. Freitas, T.A.K., Li, P.-E., Scholz, M. B., Chain, P. S. G. (2015). Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. Nucleic Acids Res. 43, e69. doi: 10.1093/nar/gkv180
8. Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., et al. (2011). Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat. Biotechnol. 29, 644–652. doi: 10.1038/nbt.1883
9. Kim, D., Song, L., Breitwieser, F. P., Salzberg, S. L. (2016). Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res. 26, 1721–1729. doi: 10.1101/gr.210641.116

10. Klingenberg, H., Meinicke, P. (2017). How to normalize metatranscriptomic count data for differential expression analysis. PeerJ 5, e3859. doi: 10.7717/peerj.3859

11. Kopylova, E., Noé, L., Touzet, H. (2012). SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. Bioinformatics 28, 3211–3217. doi: 10.1093/bioinformatics/bts611

12. Leung, H. C. M., Yiu, S. M., Chin, F. Y. L. (2014). IDBA-MTP: a hybrid metatranscriptomic assembler based on protein information. Res. Comput. Mol. Biol. 22(5). doi: 10.1007/978-3-319-05269-4_12

13. Leung, H. C. M., Yiu, S.-M., Parkinson, J., Chin, F. Y. L. (2013). IDBA-MT: de novo assembler for metatranscriptomic data generated from next-generation sequencing technology. J. Comput. Biol. 20, 540–550. doi: 10.1089/cmb.2013.0042

14. Li, D., Liu, C. M., Luo, R., Sadakane, K., Lam, T. W. (2015). MEGAHIT: an ultra-fast single--node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics 31, 1674–1676. doi: 10.1093/bioinformatics/btv033

15. Li, P.-E., Lo, C.-C., Anderson, J. J., Davenport, K. W., Bishop-Lilly, K. A., Xu, Y., et al. (2017). Enabling the democratization of the genomics revolution with a fully integrated web-based bioinformatics platform. Nucleic Acids Res. 45, 67–80. doi: 10.1093/nar/gkw1027

16. Liu, J., Li, G., Chang, Z., Yu, T., Liu, B., Mcmullen, R., et al. (2016). BinPacker: packing-based de novo transcriptome assembly from RNA-seq data. PLoS Comput. Biol. 12, e1004772. doi: 10.1371/journal.pcbi.1004772

17. Lo, C.-C., Chain, P. S. G. (2014). Rapid evaluation and quality control of next generation sequencing data with FaQCs. BMC Bioinf. 15, 366. doi: 10.1186/s12859-014-0366-2

18. Love, M. I., Huber, W., Anders, S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 15, 550. doi: 10.1186/s13059-014-0550-8

19. Meinicke, P. (2015). UProC: tools for ultra-fast protein domain classification. Bioinformatics 31, 1382–1388. doi: 10.1093/bioinformatics/btu843

20. Neves, A. L. A., Li, F., Ghoshal, B., Mcallister, T., Guan, L. L. (2017). Enhancing the resolution of rumen microbial classification from metatranscriptomic data using Kraken and Mothur. Front. Microbiol. 8, 2445. doi: 10.3389/fmicb.2017.02445

21. Nurk, S., Meleshko, D., Korobeynikov, A., Pevzner, P. A. (2017). metaSPAdes: a new versatile metagenomic assembler. Genome Res. 27, 824–834. doi: 10.1101/gr.213959.116

22. Peng, Y., Leung, H. C. M., Yiu, S. M., Chin, F. Y. L. (2012). IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics 28, 1420–1428. doi: 10.1093/bioinformatics/bts174

23. Peng, Y., Leung, H. C. M., Yiu, S. M., Chin, F. Y. L. (2012). IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics 28, 1420–1428. doi: 10.1093/bioinformatics/bts174

24. Rho, M., Tang, H., Ye, Y. (2010). FragGeneScan: predicting genes in short and error-prone reads. Nucleic Acids Res. 38, e191. doi: 10.1093/nar/gkq747

25. Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., et al. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res. 43, e47. doi: 10.1093/nar/gkv007

26. Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S. D., et al. (2010). De novo assembly and analysis of RNA-seq data. Nat. Methods 7, 909–912. doi: 10.1038/nmeth.1517

27. Robinson, M. D., Mccarthy, D. J., Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26, 139–140. doi: 10.1093/bioinformatics/btp616

28. Rotmistrovsky, K., Agarwala, R. (2011). BMTagger: best match tagger for removing human reads from metagenomics datasets.

29. Schulz, M. H., Zerbino, D. R., Vingron, M., Birney, E. (2012). Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. Bioinformatics 28, 1086–1092. doi: 10.1093/bioinformatics/bts094

30. Seemann, T. (2014). Prokka: rapid prokaryotic genome annotation. Bioinformatics 30, 2068–2069. doi: 10.1093/bioinformatics/btu153

31. Tarazona, S., Furio-Tari, P., Turra, D., Pietro, A. D., Nueda, M. J., Ferrer, A., et al. (2015). Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package. Nucleic Acids Res. 43, e140. doi: 10.1093/nar/gkv711

32. Truong, D. T., Franzosa, E. A., Tickle, T. L., Scholz, M., Weingart, G., Pasolli, E., et al. (2015). MetaPhlAn2 for enhanced metagenomic taxonomic profiling. Nat. Methods 12, 902–903. doi: 10.1038/nmeth.3589

33. Ugarte, A., Vicedomini, R., Bernardes, J., Carbone, A. (2018). A multi-source domain annotation pipeline for quantitative metagenomic and metatranscriptomic functional profiling. Microbiome 6, 149. doi: 10.1186/s40168-018-0532-2

34. Wilke, A., Bischof, J., Gerlach, W., Glass, E., Harrison, T., Keegan, K. P., et al. (2016). The MG-RAST metagenomics database and portal in 2015. Nucleic Acids Res. 44, D590–D594. doi: 10.1093/nar/gkv1322

35. Wood, D. E., Salzberg, S. L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol. 15, R46. doi: 10.1186/gb-2014-15-3-r46

36. Xie, Y., Wu, G., Tang, J., Luo, R., Patterson, J., Liu, S., et al. (2014). SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads. Bioinformatics 30, 1660–1666. doi: 10.1093/bioinformatics/btu077

37. Yamada, T., Letunic, I., Okuda, S., Kanehisa, M., Bork, P. (2011). iPath2.0: interactive pathway explorer. Nucleic Acids Res. 39, W412–W415. doi: 10.1093/nar/gkr313

38. Ye, Y., Doak, T. G. (2009). A parsimony approach to biological pathway reconstruction/inference for genomes and metagenomes. PLoS Comput. Biol. 5, e1000465. doi: 10.1371/journal.pcbi.1000465

39. Ye, Y., Tang, H. (2016). Utilizing de Bruijn graph of metagenome assembly for metatranscriptome analysis. Bioinformatics 32, 1001–1008. doi: 10.1093/bioinformatics/btv510

40. Zhong, C., Edlund, A., Yang, Y., Mclean, J. S., Yooseph, S. (2016). Metagenome and metatranscriptome analyses using protein family profiles. PLoS Comput. Biol. 12, e1004991. doi: 10.1371/journal.pcbi.1004991

13

“Tecnologia vestível para detecção de GLP e monóxido de carbono do ambiente

- Mateus Emanuel Mamani **Araujo**
  IFES
- Magnu Windell Araújo **Santos**
  IFES
- Leticia Teixeira **Nascimento**
  IFES
- Pedro Dalvi **Boina**
  IFES
- Jefferson Oliveira **Andrade**
  IFES
- Karin Satie **Komati**
  IFES
- Richard Junior Manuel Godinez **Tello**
  IFES

doi 10.37885/201102092

RESUMO

Objetivo: este trabalho apresenta o desenvolvimento da construção de um protótipo de um sistema vestível para detecção de GLP e monóxido de carbono do ambiente, bem como monitorar a frequência cardíaca da pessoa que está usando a roupa inteligente. **Métodos:** para a construção do protótipo, os equipamentos utilizados foram: sensores para detecção dos gases, um smartwatch comercial que mede a frequência cardíaca por meio de fotopletismografia, um anel de LEDs com 24 LEDs de neopixel que muda de cor quando detecta a presença de um tipo de gás e, duas placas Arduinos Nano e um Raspberry Pi 3B+ foram usados como dispositivos de processamento de dados. Para a comunicação de dados dos vários dispositivos, foram usadas as tecnologias: Bluetooth, Wi-Fi e RF 2.4GHz. Para o aviso da presença dos gases para o usuário da roupa, o smartwatch apresenta mensagens em sua tela e ao mesmo tempo vibra, além da mudança na cor das luzes do anel de LEDs. As informações também podem ser monitoradas por um sistema *web* (*Web Monitor)* ou por meio de um aplicativo móvel (*App Monitor*). **Resultados:** testes iniciais de detecção de Gás Liquefeito de Petróleo (GLP, que é uma mistura de propano e butano), popularmente conhecido como gás de cozinha, foram realizados em um Centro de Educação e Instrução de Bombeiros. **Considerações finais:** Os resultados preliminares do protótipo foram promissores e motivam a continuidade de melhorias em novas versões de protótipo.

Palavras-chave: Tecnologia Vestível. Internet das Coisas. Detecção de Gases.

INTRODUÇÃO

O Fundacentro de Segurança e Medicina do Trabalho é um órgão do governo federal do Brasil, vinculado ao Ministério da Economia, que tem por objetivo elaborar estudos e pesquisas sobre as questões de segurança, higiene, meio ambiente e medicina do trabalho. Em evento de 2019, especialistas da Fundacentro e de outras instituições discutiram os riscos e segurança do gás liquefeito de petróleo (GLP), do monóxido de carbono (CO), dentre outros (SANTOS, 2019). De acordo com eles, o GLP pode trazer riscos à saúde dos trabalhadores e, também, da população e do meio ambiente. Salientam que a prevenção surgiu a partir de estudos e políticas de segurança e saúde no trabalho, bem como atenção às normas de segurança. Já a exposição ao CO pode desencadear vários problemas graves para a saúde da população e dos animais, provocando os seguintes sintomas: fraqueza, tontura, náuseas, dor no peito e cabeça, chegando a ser letal, levando a morte por asfixia (concentração de 0,01%).

O GLP, popularmente conhecido como “gás de cozinha”, é um combustível formado pela mistura de hidrocarbonetos (metade de propano e outra metade de butano) extraídos do petróleo, sendo usado tanto para uso doméstico quanto industrial. O CO é geralmente liberado na atmosfera após um processo de combustão incompleto, aquecedores de água a gás e motores de combustão são fontes comuns de CO encontradas em centros urbanos. Ambos os gases, GLP e CO, são inodoros e incolores, o que dificulta a detecção desses produtos sem o uso de sensores específicos (AHRIMAN et al., 2015).

Após os vários acidentes devidos ao vazamento de GLP (OLIVEIRA, 2018), normas e regulamentações foram introduzidas no Brasil para diminuir os riscos. O trabalho de Moreira (2015) detalha as medidas de segurança no armazenamento do GLP em suas várias etapas. Mas muitas das medidas de segurança são específicas do ambiente em que estão sendo manipulados os gases, e nem sempre os funcionários estão perto dos leitores dos sensores, que são fixos. Considera-se muito importante que exista equipamento de segurança individual (EPI) para proteger a integridade física do trabalhador durante a atividade de trabalho. Além disso, é importante ressaltar que o Brasil representa um dos maiores mercados de GLP do mundo com uma comercialização anual da ordem de 7,4 milhões de toneladas (SINDIGÁS, 2019), e que cada residência é um potencial local de vazamento. Muitos profissionais, como bombeiros e socorristas do SAMU entram em locais que podem estar com vazamento de gases e com iminência de explosão e incluir sensoriamento de gases em suas roupas seria muito útil para avisá-los de perigos.

O uso de tecnologia é fundamental para reduzir o risco causado por esses produtos. Atualmente, os conceitos de tecnologias vestíveis e Internet das coisas (IoT) trazem muitas possibilidades para aplicações na área de segurança. A aplicação de sensores de

gás, microcontroladores e transmissores embutidos nas roupas combinadas com sistemas computacionais é uma realidade viável para a detecção de perigos presentes na atmosfera (ALAVI et al., 2018).

Trabalho similar a esta proposta, pode ser visto no artigo de Hamadi et. al. (2019), os autores desenvolveram uma jaqueta inteligente. O sistema foi dividido em 5 blocos, incluindo: 1) uma jaqueta de combate a incêndio; 2) vários sensores (temperatura, distância, gás, frequência cardíaca); 3) um sistema de comunicação; 4) barras de LED; e 5) uma tela sensível ao toque. O sensor de gás pode medir o nível de monóxido de carbono (CO), gases combustíveis e fumaça na sala. O colete de segurança desenvolvido alerta os bombeiros via tela e barras de LED.

Neste trabalho, desenvolvemos um sistema vestível composto por uma Rede de Sensores Sem Fio (RSSF) embutida em uma roupa que pode ser capaz de detectar a presença de gases no ambiente, além do batimento cardíaco do usuário através de um sensor embutido em um *smartwatch* comercial. Os dados, depois de criptografados, são transmitidos por uma rede IP sem fio para um observador externo onde podem ser acessados por meio de um aplicativo com interface gráfica instalada em um dispositivo móvel (*smartphone*).

No protótipo deste trabalho, os equipamentos utilizados foram: sensores para detecção dos gases, um *smartwatch* comercial que mede a frequência cardíaca por meio de fotopletismografia, um anel de LEDs com 24 LEDs de neopixel que muda de cor quando detecta a presença de um tipo de gás e, duas placas Arduinos Nano e um Raspberry Pi 3B+ foram usados como dispositivos de processamento de dados. Para a comunicação de dados dos vários dispositivos, foram usadas as tecnologias: Bluetooth, Wi-Fi e RF 2.4GHz. Para o aviso da presença dos gases para o usuário da roupa, o smartwatch apresenta mensagens em sua tela e ao mesmo tempo vibra, além do brilho das luzes do anel de LEDs. As informações também podem ser monitoradas por um sistema *web* (*Web Monitor)* ou por meio de um aplicativo móvel (*App Monitor*). A roupa foi um colete de proteção de motociclistas, que é bem resistente a colisões e que pode ser usada sobre quaisquer outras roupas do usuário. Neste colete estarão acoplados os diversos dispositivos/elementos elencados.

A organização do texto segue com o objetivo, os equipamentos do protótipo são detalhados na Seção Métodos, seguido dos resultados funcionais do protótipo, discussão e considerações finais.

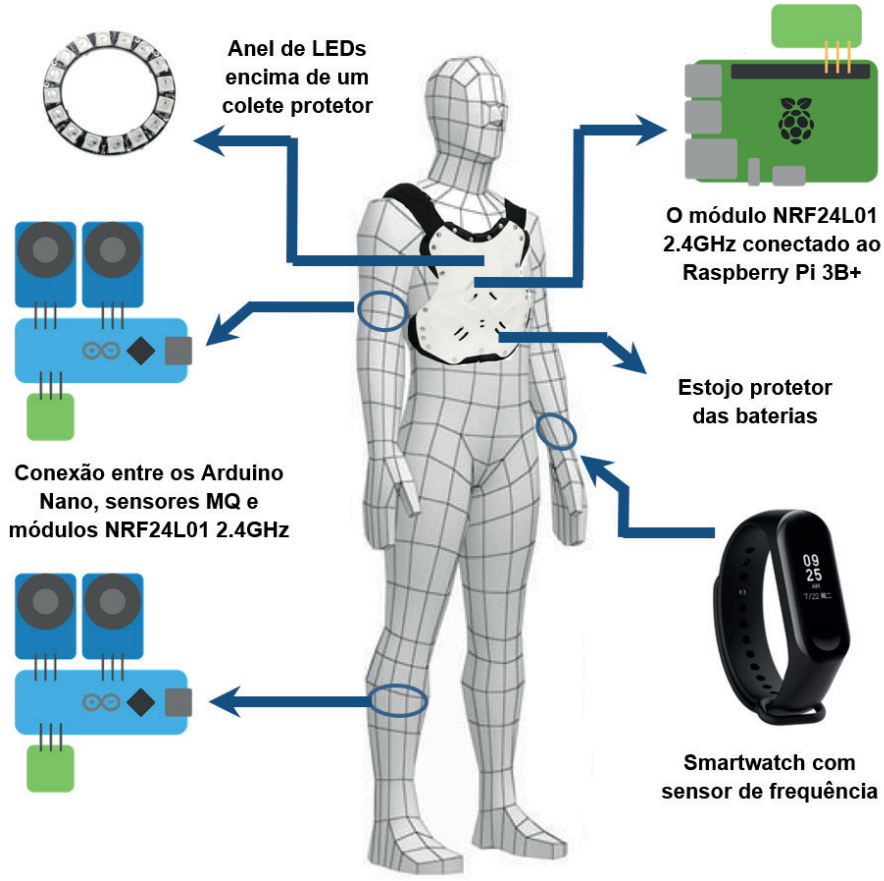
OBJETIVO

Apresentar o desenvolvimento da construção de um protótipo de um sistema vestível para detecção de GLP e monóxido de carbono do ambiente, bem como monitorar a frequência cardíaca da pessoa que está usando a roupa inteligente.

MÉTODOS

O sistema de hardware proposto inicia com um módulo de sensoriamento composto por duas placas Arduino Nano, cada uma conectada a um módulo de radiofrequência NRF24L01 e dois sensores de gás MQ. São dois módulos, pois há gases que são mais pesados que o ar e tendem a se concentrar perto do solo, logo estão localizados no tornozelo do usuário. Para as fases que são leves, há um outro módulo no braço do usuário, conforme a Figura 1 que apresenta a disposição dos equipamentos no corpo. Para a identificação e monitoramento dos gases, foram utilizados os sensores MQ-4[1] e MQ-7[2], que detectam os gases-alvo, GLP e CO, respectivamente. Fornecem sinais elétricos proporcionais à concentração de gases no ambiente, que por sua vez são processados por módulos Arduino Nano para serem enviados por transceptores de 2,4GHz - NRF24L01[3] - para o Raspberry Pi 3B+. O alcance do NRF24L01 pode chegar a 10 metros em ambientes internos e 50 metros em campo aberto.

Figura 1. Disposição dos equipamentos no corpo.



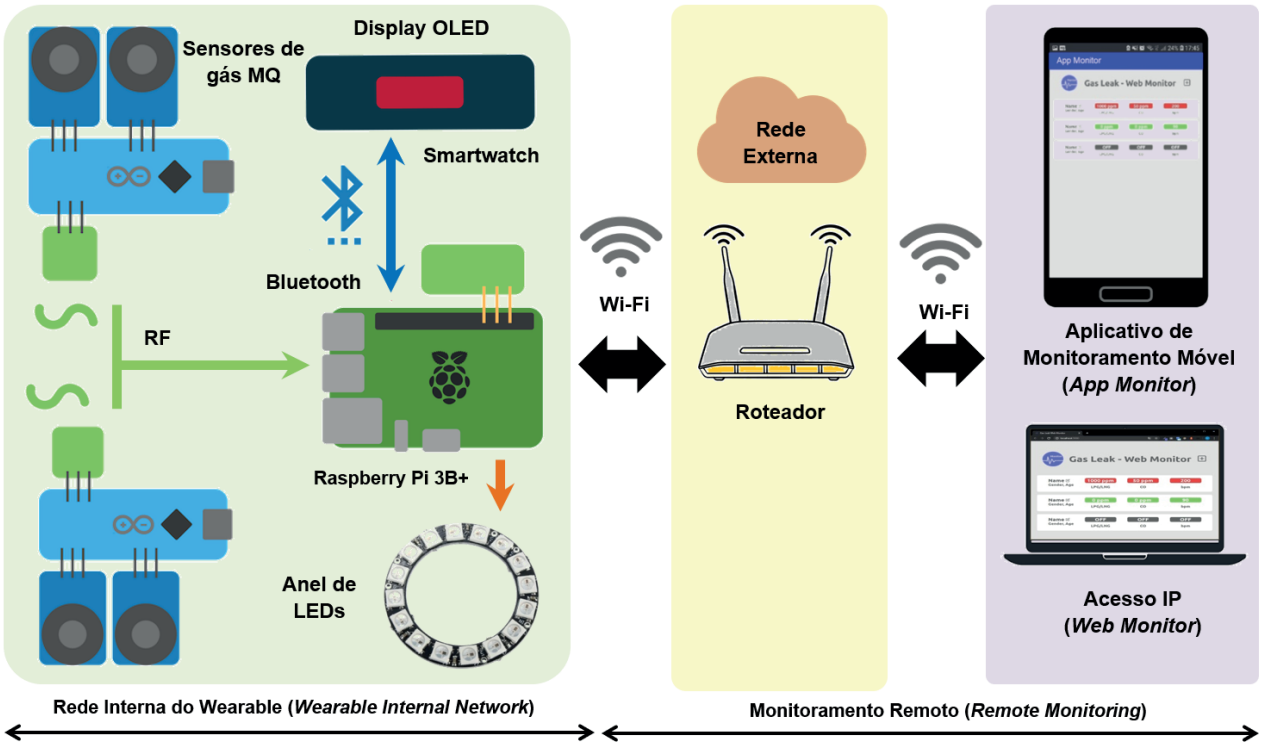
O Raspberry Pi 3B+, localizado na parte central do colete de proteção, captura os dados transmitidos pelos Arduinos Nano e a frequência cardíaca medida pelo *smartwatch* via

1 https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-4%20Ver1.3%20-%20Manual.pdf
2 https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf
3 https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf

Bluetooth. Em caso de detecção de gases perigosos, o anel de LEDs piscarão e o *smartwatch* vibrará e mostrará mensagens em sua tela.

A Figura 2 mostra um diagrama de blocos do sistema. A parte relacionada com a Rede Interna do Wearable (*Wearable Internal Network*) já foi descrita, pois é parte referente à roupa da Figura 1. Numa segunda parte chamada de Monitoramento Remoto (*Remote Monitoring*) do sistema, o Raspberry Pi 3B+, por meio de uma rede externa wi-fi, transmite os dados para um sistema central (*Web Monitor*) e que poderão ser visualizados por um aplicativo instalado em um dispositivo móvel (*App Monitor*). O Raspberry Pi 3B+ foi escolhido como unidade de controle e transmissão de dados por permitir a integração das tecnologias de rede escolhidas: RF, WI-Fi e Bluetooth.

Figura 2. Diagrama de blocos do protótipo.

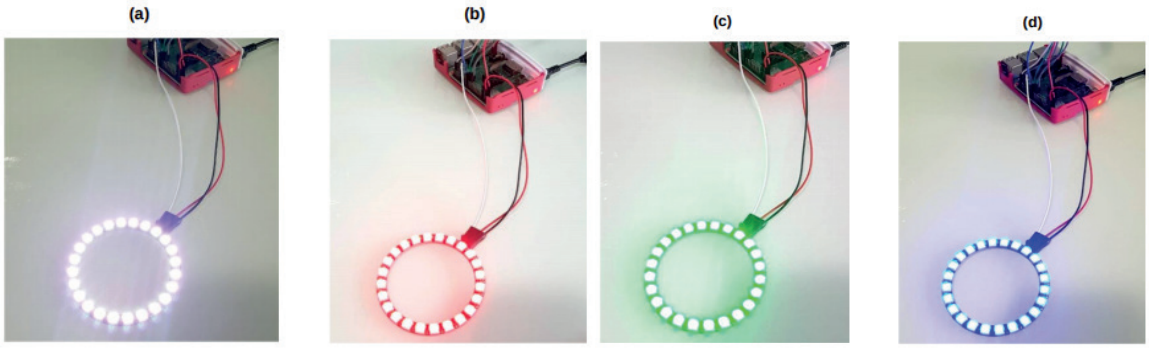


Os dados coletados dos sensores são medidos em valores de porcentagem e concentração de CO em partes por milhão (ppm). Se o valor do sensor de GLP exceder o limite de 5% ou o valor do sensor de CO exceder o limite de 50 ppm[4], o Raspberry Pi emite alertas mudando as cores do anel de LED e gerando as vibrações do smartwatch seguidas de mensagens na tela. Um ambiente seguro faz com que o anel de LEDs fique com a cor branca constante (Figura 3a). Se o valor limite for excedido detectados do sensor MQ-4, o anel de LEDs muda seu estado para LEDs azuis piscando (Figura 3d), se o valor limite for excedido em CO do sensor MQ-7, o anel de LEDs mudará seu estado para LEDs verdes piscando

4 https://www.osha.gov/laws-regs/regulations/standardnumber/ 1910 / 1910.1000

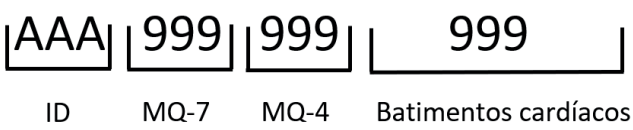
(Figura 3d), finalmente, se MQ-4 e MQ-7 excederem o limite, o anel de LEDs mudou seu estado para LEDs vermelhos piscando (Figura 3b).

Figura 3. Anel de LEDs nas diversas cores: (a) branca, (b) vermelha, (c) verde e (d) azul.



O Raspberry Pi 3B+ converte todas as informações recebidas no momento em um vetor de caracteres codificado (Figura 4). Isso representa a concatenação da identificação do Raspberry Pi 3B+ (IDs do Raspberry Pi), os valores MQ-4 e MQ-7 e a frequência cardíaca em bpm. Este vetor de caracteres é enviado para os aplicativos de monitoramento.

Figura 4. Vetor de caracteres enviado ao sistema de monitoramento.



Com relação ao monitoramento de aplicativos, foi desenvolvido um Web App. O App foi estruturado com uma API RESTful[5] no backend, montada em Flask[6], um *framework* Python e um *frontend* desenvolvido em React[7]. Estando na web, pode ser facilmente acessado por qualquer dispositivo com acesso à internet. Para mantê-lo simples e leve, apenas as classes de usuário e sensor vestíveis foram armazenadas. Para facilitar a visualização, cada sensor no *frontend* tem sua leitura mostrada em um bloco colorido (Figura 2), com as cores: verde para níveis seguros, vermelho para níveis fora da faixa de segurança e cinza para *offline*.

RESULTADOS

Os testes com sensores de gás foram realizados no Centro de Educação e Instrução de Bombeiros do Corpo de Bombeiros Militar do Espírito Santo, Brasil. Na Figura 5, são apresentados os instrumentos de testes, o GLP (do bujão de gás) é injetado numa caixa hermeticamente fechada, para que o gás não vaze da mesma. Um protótipo desenvolvido apenas com a parte de sensoriamento, uma placa Arduino Nano e o sensor MQ-4, ficou

5 https://restfulapi.net/
6 https://flask.palletsprojects.com/en/1.1.x/
7 https://reactjs.org/

dentro da caixa. Também dentro da caixa foi colocado um leitor comercial de GLP do próprio corpo de bombeiros para ter uma aferição de referência. O GLP foi injetado aos poucos, e depois os valores medidos pelo protótipo (armazenados no Arduino) foram comparados com os valores medidos pelo leitor comercial. Os resultados foram satisfatórios, dentro da margem de erro do sensor.

Figura 5. Experimento do teste de GLP.



Também foram realizados experimentos com o *smartwatch* comercial, quanto à frequência cardíaca medida. Um dos pontos positivos desse dispositivo é oferecer ergonomia e portabilidade. Os resultados estão de acordo com as especificações do fabricante do *smartwatch.*

DISCUSSÃO

Neste trabalho, um sistema vestível inteligente baseado em sensores de gás e de frequência cardíaca foi desenvolvido. Segundo os testes que incluem GLP foram realizados em um Centro de Educação e Instrução de Bombeiros, o uso do sensor MQ-4 é viável. No entanto, ainda é necessário executar experimentos com relação ao CO, mas o ambiente de testes ainda está em elaboração.

Embora neste trabalho, o sensor MQ4 esteja sendo usado para detectar GLP e MQ-7 para detectar monóxido de carbono, mas o sensor MQ-4 também tem a capacidade de identificar metano. Sendo assim, este protótipo pode ser facilmente ampliado para sensoriamento de metano, aproveitando outra funcionalidade do sensor. É possível avaliar a inclusão de outros sensores, tais como o MQ-2[8] para fumaça, o MQ-8[9] para hidrogênio e o MQ-135[10] para gases tóxicos como amônia, dióxido de carbono, benzeno, óxido nítrico, e também fumaça ou álcool.

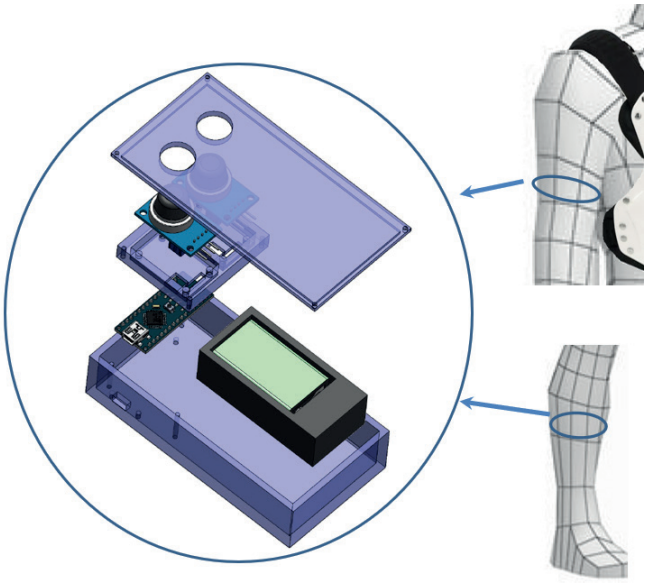
8 https://datasheetspdf.com/pdf/622943/Hanwei/MQ-2/1
9 https://cdn.sparkfun.com/datasheets/Sensors/Biometric/MQ-8%20Ver1.3%20-%20Manual.pdf
10 https://datasheetspdf.com/pdf/605076/Hanwei/MQ-135/1

CONSIDERAÇÕES FINAIS

Este estudo demonstra a viabilidade de desenvolver um sistema vestível utilizando diferentes tecnologias, vários sensores e tipos de transmissões, com o objetivo de proteger a vida de pessoas dado o ambiente em que se encontram. Os testes se concentraram na funcionalidade dos sensores.

Uma melhoria em andamento é o projeto de uma caixa de proteção para o módulo de sensoriamento que fica no braço e na perna, tal como mostrado na Figura 6.

Figura 6. Caixas de proteção do módulo de sensoriamento que ficarão no braço e na perna.



Ainda há muitos trabalhos futuros para serem realizados, como os testes com monóxido de carbono, testes de duração da bateria, e testes em ambiente hostil com altas temperaturas. Além disso, a roupa só foi usada em pessoas dentro de laboratório, não foram feitos testes com os usuários correndo ou fazendo esforços físicos para avaliar o conforto da roupa.

AGRADECIMENTOS

Os autores agradecem à FAPES (Fundação de Amparo à Pesquisa e Inovação do Espírito Santo) pelo apoio financeiro para a construção dos protótipos e pelas bolsas aos alunos de iniciação científica (Projeto número: 222/2018 do Edital FAPES/CNPq/Decit-SCTIEMS/SESA No 03/2018 – PPSUS).

REFERÊNCIAS

1. AHRIMAN, N. et al. *Wireless sensor network application for carbon monoxide monitoring*. In: IEEE. 2015 9th International Conference on Telecommunication Systems Services andApplications (TSSA). [S.l.], 2015. p. 1–4.

2. ALAVI, Amir H. et al. Internet of things-enabled smart cities: State-of-the-art and future trends. *Measurement*, v. 129, 2018.

3. HAMADI, AL AMAL et al. *Multi-functional safety jacket for firefighters*. In: IEEE. 2019 Advances in Science and Engineering Technology International Conferences (ASET). [S.l.], 2019. p. 1–5.

4. MOREIRA, A. M. *Segurança na utilização de gás liquefeito de petróleo*. 2015. 54 f. Monografia (Especialização) — Curso de Especialização em Engenharia De Campo.Universidade Federal do Espírito Santo, Vitória, ES, 2015.

5. OLIVEIRA, Alexandre Gustavo Souza de. *A importância da identificação dos riscos deacidentes em uma engarrafadora de gás liquefeito do petróleo (gás LP)*. 2018. Disponível em: <https://monografias.brasilescola.uol.com.br/engenharia/a-importancia-identificacao-dos-riscos-acidentes-uma-engarrafadora.htm>. Acesso em: 08 mar. 2020.

6. SANTOS, Débora Maria. Especialistas comentam sobre os riscos e segurança dos gases industriais. 2019. Disponível em: <https://www.gov.br/fundacentro/pt-br/assuntos/noticias/noticias/2019/10/especialistas-comentam-sobre-o-seminario-sobre-glp-gnv-monoxido-de-carbono-e-gases-naturais>. Acesso em: 11 out. 2020.

7. SINDIGÁS. *Panorama do setor de GLP em movimento*. 2019. Disponível em: <http://www.sindigas.org.br/novosite/wp-content/uploads/2019/02/Panorama-do-GLP_janeiro_2019.pdf>. Acesso em: 11 out. 2020.

14

“Uma ferramenta baseada em semântica para o apoio na navegação em ambientes inteligentes inclusivos

- José de Jesus Pérez-**Alcázar**
  EACH - USP
- Fábio **Nakano**
  EACH - USP
- Joedson Santos de **Santana**
  EACH - USP
- Henrique Nakaema **Simões**
  EACH - USP
- Felipe Morine **Magami**
  EACH - USP
- André Luis Meneses **Silva**
  UFS - Itabaiana

doi 10.37885/201102089

RESUMO

Há centenas de milhões de deficientes visuais no mundo inteiro. Indivíduos que enfrentam diariamente dificuldades relacionadas a sua mobilidade. O avanço das tecnologias móveis traz mais oportunidades para a construção de aplicativos para apoio ao deficiente visual. Propõe-se o desenvolvimento de um protótipo de um sistema de guia de um shopping center fictício para um deficiente visual utilizando uma abordagem web-semântica, com o intuito de verificar a viabilidade de um projeto similar ser implantado em um sistema com agentes e ambientes reais e de mostrar a metodologia por trás do desenvolvimento desse protótipo.

Palavras-chave: Web Semântica das Coisas, Navegação Indoor, OWL, RDF, Ambientes Inteligentes Inclusivos, Deficiente Visual,

INTRODUÇÃO

Estima-se que há 285 milhões de pessoas com algum tipo de deficiência visual [Pascolini e Mariotti 2011]. Deficiência esta que dificulta locomoção e mobilidade desses indivíduos. Em locais públicos de maior movimento, existem instalações como pisos táteis, informativos sonoros e em braille que visam melhorar a acessibilidade e mobilidade a essas pessoas. Por outro lado, a escassez de tais instalações visto o tamanho de uma cidade, bem como as limitações desses serviços, significa que a facilidade de locomoção e acesso providas por esses serviços é limitada localmente e sua utilidade depende de outras habilidades adquiridas pelo usuário. Por exemplo, disponibilizar um mapa em braile de um shopping quando a portadora de deficiência visual desconhece esse sistema de escrita.

Com o avanço da Internet e das tecnologias móveis em tempos recentes, há uma oportunidade de se mitigar os problemas destacados anteriormente. As informações sobre acomodação, lojas, comida, eventos e atrações estão muito mais acessíveis [Park et al. 2012]. As aplicações móveis facilitam o acesso a serviços remotos disponíveis a qualquer hora, em qualquer lugar, e podem se aproveitar da ubiquidade que a tecnologia wireless proporciona [Chiu et al. 2009].

O paradigma de Internet of Things (IoT, Internet das coisas) é definido pela tendência recente de se tentar “conectar os objetos do mundo físico com a rede de informações”, fazendo com que a distância entre o mundo físico e o virtual seja diminuída, e os objetos virem participantes ativos no processo de negócio [Weber and Weber 2010], permitindo pervasividade e ubiquidade, impactando a vida tanto dentro como fora de casa [Botta et al. 2016]. Um exemplo que reflete esse impacto é o conceito de Smart Houses (casas inteligentes), em que os objetos do seu lar, como TVs, geladeiras, fornos e até mesmo armários ou alarmes de incêndio estão conectados, com o objetivo de se obter conforto, automação e controle [Stefanov et al. 2004].

O objetivo deste trabalho foi *realizar um protótipo de um sistema para auxílio a portadores de deficiência visual utilizando conceitos de web semântica* sobre um local indoor simulado, um shopping center, a fim de verificar a viabilidade da aplicação antes de aplicá-la a um ambiente real, interagindo com agentes reais. Para isso, desenvolvemos um sistema que contém um aplicativo Android, que servirá como interface para o usuário cego; um servidor que, dado o objetivo do usuário, descobre serviços que possam atendê-lo; um módulo de execução desses serviços; e uma simulação de uma cadeia de sensores que estaria espalhada pelo nosso shopping. A descrição dos conceitos usados na base foi feita através da linguagem OWL (Web Ontology Language) (OWL, 2012) e os dados foram descritos em RDF (Resource Data Framework)(RDF, 2014), padrões definidos pela World Wide Web Consortium (W3C, 2020) Para gerenciar esses dados utilizamos o sistema livre

de código aberto Apache Jena (Apache Software Foundation, 2020), que nos possibilita construir aplicações baseadas em Web semântica e dados ligados (HEATH, T., 2011). Este tipo de solução se encaixa na área Web Semântica das coisas (“Semantic Web of Things”) (JARA, A. 2014) da qual falaremos mais na frente. Por fim, utilizando o sistema sobre o cenário simulado, verifica-se a viabilidade do projeto. Espera-se que o conhecimento gerado, com a realização deste protótipo auto-contido sobre o cenário simulado, possa ser aplicado e estendido a um cenário real.

Fundamentação teórica

Apresentamos nesta seção conceitos, ferramentas e tecnologias estudados e empregados no desenvolvimento deste projeto.

Web semântica

Atualmente, a web utiliza principalmente HTML (Hypertext Markup Language) para marcação de texto. Embora o HTML permite o controle sobre como a página web é apresentada, pouco pode ser estabelecido no que tange a semântica do conteúdo da página. Além disso, apesar de populares, recuperação de informação pela web é feita particularmente pela utilização de motores de busca como o Google. Esses motores funcionam apenas buscando correspondência de palavras-chaves, e portanto a relevância dos documentos recuperados pode ser baixa (Cai et al. 2012).

A web semântica é proposta como uma extensão da web atual, de forma que a informação disponível seja acompanhada por um significado, tal que essa semântica incluída permite que a informação possa ser processada e interpretada computacionalmente e portanto permitindo que máquinas tenham um papel mais ativo no consumo de dados e não apenas na transmissão de informação para consumo humano (Fensel et al. 2012). A web semântica sugere o uso de ontologias para a marcação das páginas, para a atribuição de significado à informação e assim a resolução dos problemas da web atual relatados acima (Cai et al. 2012).

Ontologia

Gruber define ontologia como “uma especificação explícita de uma conceitualização” (Gruber, 1993). Em 1998, Studer et al. (1998) complementou essa definição: “uma especificação formal e explícita de uma conceitualização compartilhada” .

Conceitualização é uma abstração do mundo que queremos representar. Ela deve ser uma especificação explícita e formal para não permitir ambiguidades e que possa ser

compreensível a uma máquina. Além disso, deve ser compartilhada para que seja aceita por grande parte da comunidade. Ela deve servir como um vocabulário comum que permita ser reusado. Uma discussão desta definição pode ser encontradas em (Guarino et al. 2009).

Uma ontologia deve conter classes e relacionamentos. A taxonomia da ontologia revela a estrutura das classes e seus relacionamentos. Esses relacionamentos podem ser divididos em geral em duas categorias: uma hierárquica, que expressa as relações entre superclasses e subclasses (e.g., a classe Médico poderia ser uma subclasse de Pessoa, é uma superclasse de Oftalmologista); e relações de propriedade, que expressam atributos de uma classe, sendo esse atributo uma associação a outra classe ou uma descrição de alguma característica importante para a classe [Yu 2011]. A seguir, descreveremos brevemente algumas linguagens padronizadas pela W3C para a construção de Ontologias e Bases de Conhecimento, hoje em dia chamados Grafos de Conhecimento ("Knowledge Graphs") (Fensel, D. et al., 2020): RDF e OWL.

RDF - Resource Data Framework

Um dos objetivos da Web Semântica é ver a Web como um grande repositório de dados, onde a Web passa de uma Web de Documentos para uma Web de Dados (Hogan, A., 2020). Essa Web vai permitir: criar enlaces não somente entre documentos, senão entre coisas arbitrárias (ex: pessoas, eventos, localizações, etc); explicitar a estrutura dos dados nas páginas web; as coisas descritas nas páginas terão uma identificação única (URI - Uniform Resource Identifier); e os enlaces entre coisas são explicitadas e tipadas. Para a criação da Web de Dados é usado um modelo de dados para os objetos e relações, onde os objetos (coisas) são descritos como nós e as arestas descrevem os enlaces entre os objetos. O formalismo usado para representar esses grafos é chamado de RDF (RDF, 2014). Esse tipo de estrutura é o suporte do conceito de dados ligados (Heath, T. et. al., 2011). Entretanto, para estruturar esses grafos e definir um esquema que prescreve uma estrutura de alto nível e/ou semântica que o grafo segue ou deveria seguir se faz necessário de associar uma Ontologia a esse grafo. Por isso, é importante a definição de linguagens ontológicas associadas a esses grafos com RDFS(RDF Schema)(Brickley e Guha, 2014) e OWL.

RDFS e OWL

RDFS e OWL são linguagens leves para representação do conhecimento. Elas permitem codificar conceitos e suas relações, conjuntamente designados 'dados', conforme convenções em lógica de predicados. Esta característica permite a inferência (dedução) de informação adicional à informação explicitamente definida. RDFS é uma linguagem lógica de expressividade muito baixa que permite algumas inferências muito simples, tais como

herança de propriedades sobre uma hierarquia de tipos, inferência de tipos de restrições de domínio e alcance das propriedades ou enlaces entre objetos. De forma similar, OWL é uma lógica um pouco mais rica, ainda que relativamente leve, que permite inferências adicionais tais como igualdade e desigualdade, restrições de cardinalidade, de existência de objetos, etc. (Antoniou, G. et al., 2012). A ideia por trás dessas descrições é permitir que os agentes de software (programas) possam raciocinar sobre o conteúdo existente na Web.

Ferramentas de Gerenciamento de dados RDF e raciocinadores OWL

Para o armazenamento e recuperação de dados RDF, existem várias ferramentas de gerenciamento, entre elas podemos enumerar; AllegroGraph (FRANZ INC., 2020), MarkLogic (MARKLOGIC, 2020), Virtuoso (OPENLINK, 2019), Apache Jena (Apache Software Foundation, 2020) (Carroll, et al., 2004), entre outros. Na aplicação apresentada aqui foi usada a ferramenta Apache Jena. Esta foi originalmente desenvolvida por pesquisadores dos laboratórios da HP em Bristol (UK) em 2000, desde 2010 é um software da fundação Apache. Jena é um arcabouço Java de código aberto para a criação de aplicativos em web semântica. Compreende APIs RDF e OWL, que permitem leitura e gravação de dados RDF em diferentes formatos (XML, N3, Turtle etc.), interface para serviços de raciocínio e um motor de consultas SPARQL (DuCharme, B., 2013). SPARQL é uma linguagem tipo SQL que permite consulta e manipulação de dados RDF. Sua escolha se deu por ser de domínio público e uma das mais populares e mais maduras, de acordo com o ranking DB_Engine (DB-Engines, 2020).

A Conexão de Web Semântica e Internet das coisas - “Semantic Web of Things”

O potencial dos sistemas em Internet das Coisas (IoT) (ATZORI, I. et al., 2010) tem origem na interconexão entre dispositivos físicos e no uso integrado dos dados e sistemas de informação. Quanto mais interconectados e integrados, maior a criação de serviços e aplicações de amplo alcance, devido a quantidade de dispositivos já existentes, e de alta especificidade, devido a quantidade de informação coletada. Isto pode ser usado para melhorar a vida das pessoas. Um dos principais obstáculos para realizar esse potencial é a atual falta de interoperabilidade dos dispositivos. Comparado ao que existe hoje, os sistemas IoT precisam ser mais abertos, interoperáveis e colaborativos.

Os sistemas IoT são conjuntos de dispositivos interconectados dentro da Internet. Existem várias ilhas de dispositivos, operando sobre diferentes protocolos e infra-estruturas, que têm em comum o uso do protocolo IPv6. A integração entre as ilhas foi estabelecida usando soluções como: 6LoWPAN (HUI; THUBERT, 2011), GloWBAL IPv6 (JARA; et al, 2012), etc. Uma vez alcançada a conectividade, precisou-se de um protocolo acima de

IPv6 para conectar dispositivos e serviços IoT. Protocolos da IETF como o Protocolo de Aplicações Restritas (COAP) (SHELBY; et al, 2013), que é mapeado para HTTP, e também para implementações leves de HTTP, usam dispositivos como recursos, mesmo que estes tenham poder computacional muito restrito. Esses protocolos oferecem infraestrutura para a descoberta e descrição de recursos, diretórios de recursos (para alocação e descoberta) e segurança. Desta forma foi construída a Web of Things (WoT). A WoT permite que as coisas interajam entre si, em sistemas de complexidade crescente. Desse modo, é possível a composição de serviços e soluções mais complexas. Isso é facilitado pela definição de APIs sobre protocolos como HTTP ou CoAP. Entretanto, o progresso na integração é dificultado pelo fato que "vários dispositivos não falam a mesma linguagem e não podem trocar dados através de diferentes 'gateways' e concentradores "inteligentes" (SHIAO, M., 2015). Isto é, os dados gerados pelos dispositivos sobre seu meio ambiente podem ter uma estrutura definida em um formato conhecido (ex: JSON/CSV/XML), mas os modelos de dados e esquemas adotados são diferentes e não são sempre compatíveis. Isso nos leva a um desafio de interoperabilidade, o qual tem sido reconhecido pelos órgãos de aliança da indústria tais como a Aliança pela Inovação em Internet das Coisas (AIOTI) e na "European IoT research roadmap" (OVIDIU, V. et al., 2011) como chave para alcançar "convergência a longo prazo" (SHIAO, M., 2015). Pesquisas têm levado à conclusão unânime que as tecnologias semânticas podem ajudar na integração e interoperabilidade ((JARA, A., et. al., 2014; WEI, W, et al., 2015), facilitando a anotação semântica de dispositivos e dados IoT (DE, S. et. al.,2017). Os agentes de software podem entender essas anotações e facilitar a integração e comunicação entre dispositivos de plataformas diferentes e que usam formatos de dados diferentes. O uso de tecnologias semânticas em IoT tem sido chamado na literatura como Web Semânticas das coisas (SWOT) (JARA, A., et. al., 2014). Especificamente, o uso de técnicas de modelagem baseadas em web semântica (modelos RDF e Ontologias OWL) podem facilitar meios escaláveis e homogêneos de acessar informação da WoT. Uma IoT baseada em semântica pode facilitar a criação de uma geração de soluções sensíveis ao contexto, serviços ubíquos e sistemas cyber-físicos inteligentes. O escopo da nossa proposta está incorporado dentro da área de SWOT.

Ontologias podem capturar as várias facetas de IoT em uma estrutura comum, formalizada além dos relacionamentos entre as coisas para formar o domínio de conhecimento da WoT (DE, S. et. al, 2017). Entretanto, dada a escala massiva das coisas envolvidas na IoT, a "criação de uma só ontologia para definir todos os recursos e gerenciar todos os aspectos relacionados a eles parece ser impossível" (MANATE, B., et al., 2014). Para isso se faz necessário o uso de ontologias que representem não somente coisas específicas a IoT como: sensores, dispositivos, atuadores etc., senão também o contexto no qual elas

operam. Um número de ontologias e modelos de contexto têm sido propostos pela comunidade de pesquisa para tratar os requisitos de modelagem existentes pelas aplicações de WoT. De, S., et al (2017) classificam as Ontologias em: Ontologias de domínio cruzado e ontologias de domínio específico. As ontologias de domínio cruzado consistem em conceitos de WoT que são compartilhados através de domínios e silos de aplicação verticais. Este tipo de ontologia é classificado em quatro subclasses: 1) de dispositivos, incluindo objetos inteligentes., sensores e atuadores entre outros; 2) de serviços, que cobrem o fornecimento de funcionalidades do dispositivo WoT como serviços; 3) de dados, que representam a anotação semântica de dados instantâneos e de streaming (incluindo unidades de medição); 4) de localização, que capturam tanto localizações de interiores e exteriores, tais como características geográficas. As ontologias de domínio específico estão relacionadas a aplicações de domínio específico como: casas inteligentes, prédios inteligentes, ambientes assistivos etc.

Trabalhos relacionados

Publicações que utilizam semântica para deficientes visuais são incomuns na literatura. Plessers em (Plessers, P., et al, 2004) discutem sobre as dificuldades que deficientes visuais enfrentam navegando a web. É proposta uma nova abordagem para desenvolvimento de páginas web combinando uma ferramenta que analisa páginas web, identifica os componentes que auxiliam na navegação do site e, utilizando uma ontologia, os transforma em anotações que serão acessadas por leitores de telas para deficientes visuais, e um método de design de páginas web para criação sistemática de sites. Um site desenvolvido usando esse método, portanto, seria mais facilmente analisado pela ferramenta, permitindo que mesmo páginas dinâmicas possam ser analisadas, e que grande parte das anotações geradas sejam automatizadas.

Lee em (Lee, K., et al. 2013) propõem um método de análise topológica de um shopping center utilizando ontologias. Empregando um método de análise de comparação de caracteres, em que a localização do usuário é encontrada, capturando, com a câmera de um aparelho móvel, dados descritivos, como placas ou nome de estabelecimentos, e uma ontologia que descreve as atividades do shopping. O estudo desenvolveu uma aplicação Android para buscas semânticas em torno das atividades desejadas pelo usuário e os locais que as oferecem. Adicionalmente, uma visualização 3D do shopping, bem como um cálculo de rota no local.

Grande parte das tecnologias voltadas para deficientes visuais focam especificamente em aparelhos para auxiliar o deficiente a identificar obstáculos, conhecidos como ETAs (Electronic Travel Aids). Dakopoulus e Bourbakis (Dakopoulos, D., et. al., 2010) estudam diferentes propostas de aparelhos, analisando seus funcionamentos no que tange ao hardware

usado pelo usuário e como cada aparelho identifica o obstáculo, e como é realizado o feedback para o usuário. Embora existam algumas semelhanças entre os diferentes projetos, como uma preferência em identificar o obstáculo usando ecolocalização, suas abordagens divergem em geral, e o estudo conclui que nenhum é totalmente satisfatório. EOAs (Electronic Orientation Aids) diferem de ETAs por se especializarem na orientação do usuário ao destino desejado; isto é, se preocupam com a rota entre a localização do usuário e seu destino. Farcy (Farcy, R. et. al., 2006) propõe, além de um projeto de ETA, um projeto de EOA, que utiliza GPS e um sistema de voz para assistir na rota do usuário. Nem ETAs nem EOAs, contudo, se preocupam com as preferências de seus usuários, nem o contexto do lugar em que se encontram.

NavCog (Ahmetovic, D. et. al., 2016) é um aplicativo iPhone proposto pela Carnegie Mellon University para navegação indoor. Também pode ser usado em lugares grandes e complexos como uma universidade, um aeroporto ou um hospital. Este aplicativo se comunica com o usuário por meio de sons simples e dicas de voz como instruções de navegação e localiza o usuário com auxílio de *beacons* Bluetooth Low Energy (BLE) e do uso do algoritmo K-Nearest Neighbor (KNN), de forma mais precisa que um GPS ou WiFi. Após localizar o usuário, traça uma rota baseada nas informações de mapa previamente colocadas. Desta forma, por meio de uma tecnologia de baixo custo, associado a um servidor web para envio de mapas e rotas, o aplicativo consegue criar rotas para deficientes visuais dentro de espaços fechados.

Na versão NavCog3 (Sato, D. et. al., 2017), houve uma melhoria na organização e uso dos *beacons* e na interface com o usuário. Mantendo na tecnologia BLE, acrescentando uma rede de compartilhamento de informação entre os *beacons* para melhorar a precisão do algoritmo de localização. Isto foi obtido com uso de *fingerprints*, obtidos por um indivíduo que se aproxima de cada *beacon* e registra sua posição no mapa, criando uma espécie de conjunto de treinamento que permite calibrar o algoritmo de localização. Desta forma, melhora-se a precisão do algoritmo de localização, aprimorando o roteamento. A interface recebeu um servidor de conversação, permitindo comandos mais complexos na troca de informações entre o usuário e o sistema de navegação.

Magami, F. et. al (Magami, F. et. al, 2017) propõem uma abordagem baseada em semântica, que organiza, relaciona e atribui significado a informações do espaço urbano, com o intuito de prover conteúdo adaptado às pessoas com deficiência. A ferramenta proposta, construída sobre WSML (Fensel, 2002), é validada em um sistema de navegação em shopping center para apoio a usuários com deficiência visual. O trabalho aqui proposto é uma evolução do Magami, F. et. al. Neste, WSML é substituído pela combinação das linguagens OWL e RDF, reúso de ontologias ou vocabulários para representação espacial (IndoorGML),

de sensores e dispositivos (SOSA/SSN), etc. A execução e armazenamento são feitos com o arcabouço Jena. O presente trabalho também é influenciado por NavCog3, especificamente no uso dos *beacons* e do algoritmo de localização.

OBJETIVO

Realizar um protótipo de um sistema para auxílio a portadores de deficiência visual utilizando conceitos de web semântica sobre um local indoor simulado, a fim de verificar a viabilidade da aplicação.

MÉTODOS

O presente sistema foi desenvolvido a partir das experiências anteriores dos autores. Estes iniciaram o projeto do sistema a partir das bases de conhecimento disponíveis, por composição, avançando para a implementação e consequente instanciação. A implementação utiliza o arcabouço Jena [Carroll et. al. 2004].

Esta seção inicia com uma descrição do sistema em dimensões de representação, onde são apresentados grandes blocos conceituais, funcionais e físicos. Estes são sucessivamente detalhados até o nível dos elementos citados na apresentação do capítulo.

Arquitetura do sistema

O sistema é dividido em três dimensões. Em cada dimensão, são apresentadas as projeções dos elementos do sistema projetados na respectiva dimensão. A Figura 1 apresenta dimensões e elementos em forma de diagrama.

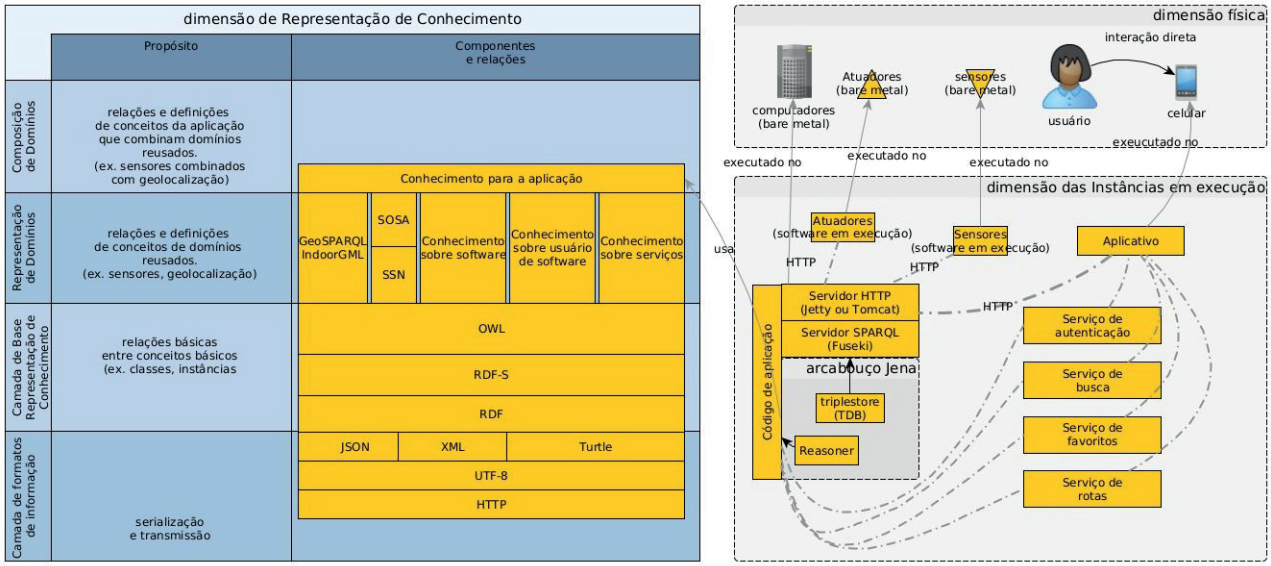
A 'dimensão da Representação de Conhecimento' mostra, em uma estrutura de camadas, a maneira como o conhecimento é representado no desenvolvimento. A camada de cima contém a representação dos conceitos específicos da aplicação: resolver o problema de achar uma rota (*wayfinding*), com o apoio de sensores e atuadores e personalizando a informação para o usuário. Neste domínio, conceitos específicos são criados sobre outros conceitos criados em outras aplicações e domínios mostrados na camada abaixo. Esta apresenta os domínios em que são definidos conceitos em: Geolocalização e localização indoor: GeoSPARQL [Perry e Herring, 2012], IndoorGML [Kang e Li, 2017], sensores e atuadores: SSN [Compton et. al., 2012], SOSA [Janowicz, 2019] e personalização para o usuário. Estas definições de conceitos são expressas usando os conceitos de linguagem definidos em OWL, RDFS e RDF. Estes são definidos sobre os padrões de associação de cadeias de caracteres XML, Turtle e JSON, que se baseiam na codificação de caracteres utilizada, no caso, UTF-8,

que é serializado e transmitido usando Hypertext Transport Protocol (HTTP). Por simplicidade, considera-se que as camadas abaixo: TCP, UDP, etc. estão contidas na camada HTTP.

Na 'dimensão das instâncias em execução', há vários serviços web que resolvem algum aspecto do sistema ou solicitação do usuário. Em todos os serviços, o conhecimento e as regras compõe um modelo, em parte codificado na aplicação, em parte armazenado em um banco de dados. Em aplicações de web-semântica, bancos de dados são estruturados como *triplestores*. Na aplicação específica, usou-se TDB. Consultas sobre os dados são feitos usando SPARQL sobre a infraestrutura oferecida por Jena. O servidor SPARQL utilizado é o Fuseki . A comunicação entre o servidor SPARQL e os clientes é feita através do protocolo HTTP, através de um servidor HTTP. Fuseki tem incorporado o servidor Jetty, que permite que o primeiro funcione como um servidor HTTP com interface de programação através de navegadores Web. O navegador web também é a interface para uso do serviço pelo usuário. Tanto TDB quanto Fuseki fazem parte do arcabouço Jena.

Na 'dimensão física' está o usuário, que interage com o celular para usar um aplicativo que acessa os serviços, como o de roteamento. Nesta dimensão, também há sensores e atuadores: dispositivos com pouca capacidade de processamento que executam programas, por isso são representados na dimensão de execução de programas, conectam-se e enviam dados ao servidor. Esta conexão pode ser mediada por outros serviços web que não são apresentados no diagrama. A informação gerada pelos sensores ou encaminhada pelos atuadores pode ser usada para enriquecer a experiência do usuário e/ou a acurácia do serviço.

Figura 1. Da esquerda para a direita e de cima para baixo, os quadros: 'dimensão da Representação de Conhecimento' ,'dimensão das instâncias em execução' e 'dimensão física'.



Fonte: os autores.

Definição do escopo do sistema

Durante a etapa inicial do projeto, para definição das funcionalidades que serão oferecidas pelo sistema, desenvolvemos casos de uso do usuário e selecionamos aqueles que identificamos como essenciais ao projeto. De forma resumida, são elas:

- Buscar Localidade: o usuário, através do navegador de seu celular, fará uma consulta, por alguma localidade. Ele envia como entrada ao serviço de busca características da localidade que deseja. Ele pode, portanto, buscar uma localidade como uma loja que vende certo produto, ou por um cinema, por exemplo. O servidor da aplicação recebe essa consulta e a executa sobre a base de conhecimento, retornando ao usuário informações das localidades que atendem sua busca;
- Buscar Rota: tendo procurado por uma localidade, ele poderá requisitar um serviço de roteamento, que deverá levá-lo até o seu destino desejado. O aplicativo web executado no navegador constrói e envia a informação, que inclui dados dos sensores embarcados no celular e respostas do usuário que permitem determinar a localidade atual e a localidade desejada. O servidor calcula a rota entre o local onde o usuário está e o destino. Sensores e atuadores espalhados pelo local auxiliarão o sistema para verificar se o usuário está seguindo o trajeto traçado com sucesso. O cálculo da rota pode ser feito de várias formas: desde aplicando um algoritmo de caminho mínimo, codificado na aplicação, até um Reasoner criado com os recursos do arcabouço Jena e que considera a informação contida na base de conhecimento para construção da rota;
- Favoritar Localidade: ”favoritar” e nesse caso explicitar que tal localidade é de sua preferência, e que ele deseja ser alertado quando passar por ela. Decidimos pelo uso do termo ”favoritar” por sua funcionalidade ser parecida com a de browsers e sites de música ou de vídeos, por exemplo, em vez de usar o mais genérico ”curtir”;
- Recomendar Localidade: assim que um sensor identifica o usuário próximo, ele manda uma resposta ao servidor. Se essa localidade representada pelo sensor for uma localidade previamente favoritada, ele alertará o usuário, recomendando essa localidade.

Descrição do shopping em ontologias

A referência a conceitos definidos em ontologias são notados como ontologia:conceito. Por exemplo IndoorCore:CellSpace é a referência ao conceito CellSpace definido na ontologia IndoorCore, gml:AbstractFeature é a referência ao conceito AbstractFeature definido na ontologia gml. Esta notação permite referenciar o conceito com suficiente precisão. Os conceitos

utilizados são acompanhados de uma breve explicação ao longo do texto. O detalhamento pode ser obtido na documentação da ontologia citada na referência.

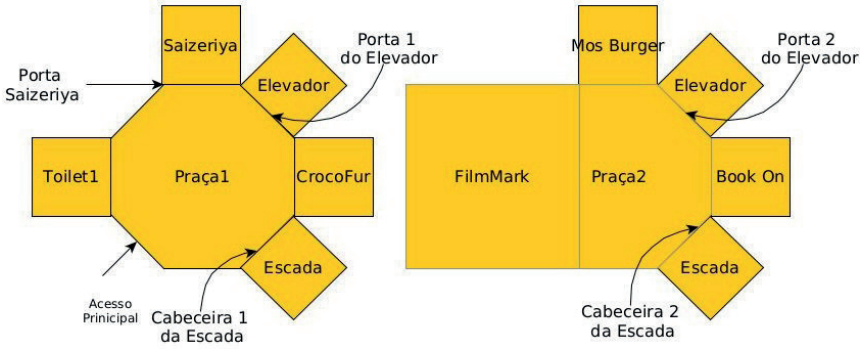
Os elementos de modelagem simplificada de shopping são definidos sobre IndoorGML. Esta ontologia define dois modelos: o primal e o dual.

No modelo primal, ambientes são coleções de IndoorCore:CellSpace. Este é subclasse de gml:AbstractFeature, que contém os predicados para conexão com ontologias para cidades, como CityGML e para descrição do espaço, como gml:description, que é usado como chave de busca nas solicitações do usuário. gml:AbstractFeature está relacionado a gml:Geometry, que contém os predicados associados à geometria do espaço. Corredores, escadas, elevadores são IndoorNavi:TransitionSpace, entradas, sejam de lojas ou outras dependências, são IndoorNavi:ConnectionBoundary, o espaço interno das lojas e outras dependências são rdfs:subClassOf (sub-classes) de IndoorNavi:CellSpace.

No modelo dual, IndoorNavi:CellSpace é mapeado em IndoorNavi:RouteNode e Indoor:ConnectionBoundary é mapeado em IndoorNavi:RouteSegment. Instâncias deste conectam dois IndoorNavi:RouteNode através do predicado IndoorNavi:connects. Isto permite, a partir do modelo dual, deduzir o grafo que representa a conexão entre os espaços do shopping. Sobre este grafo são aplicáveis algoritmos de roteamento, por exemplo, de algoritmos de caminho mínimo, resultando nas rotas que são informadas pelo serviço de busca de rota. Segue um exemplo para apresentar a aplicação dos conceitos.

A instância de teste do shopping tem dois pisos. O índice 1 é usado para elementos no piso 1, térreo, onde localiza-se o acesso principal ao shopping. Há um elevador, uma escada e banheiros. As lojas do térreo são Saizeriya e CrocoFur e as lojas do piso 2, superior, são FilmMark, BookOn e MosBurger (Figura 2).

Figura 2. Disposição física dos elementos na instância de teste do shopping. Este tem



Fonte: os autores.

A base de conhecimento contém conhecimento oriundo das ontologias concatenado às representações das particulares instâncias de teste. A classe mais abstrata, tanto no modelo primal quanto no modelo dual é gml:AbstractFeature. No modelo, o shopping é um

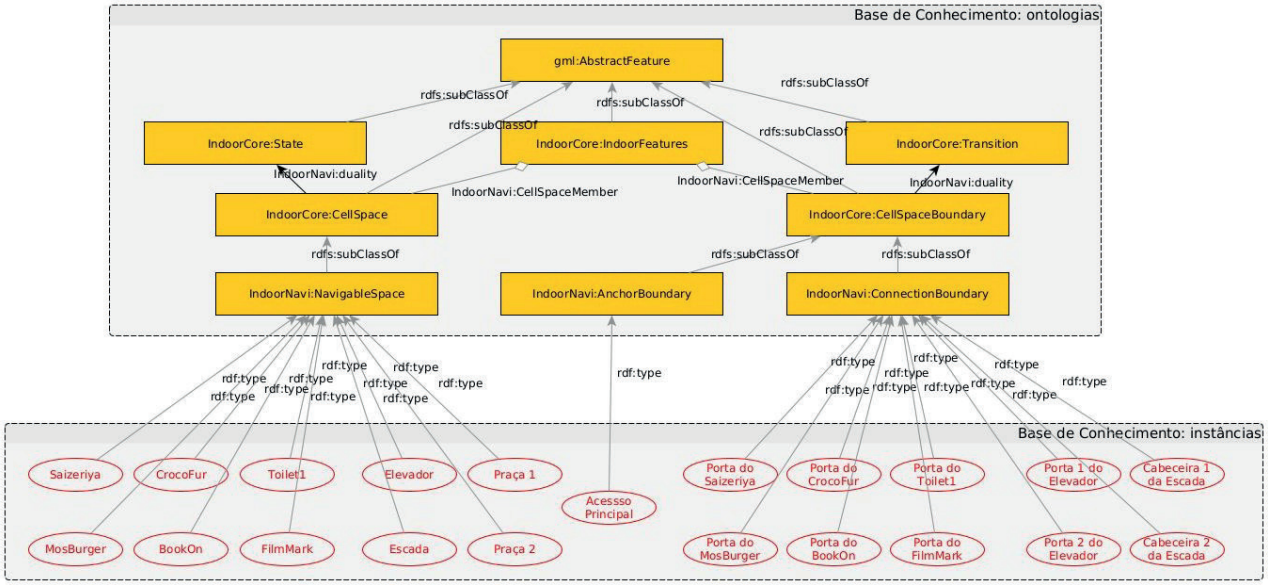
agrupamento de instâncias de lojas e outras dependências. Segundo a definição dada na ontologia, o shopping é instância de IndoorCore:IndoorFeatures. O Acesso Principal é uma instância de IndoorNavi:AnchorBoundary, que representa a conexão (ponto de ancoragem) entre o modelo para ambientes internos com modelos para ambientes externos, como CityGML. Lojas e outros espaços, cujo propósito não seja circulação, são instâncias de IndoorNavi:CellSpace. Portas e cabeceiras de escadas são fronteiras de conexão, instâncias de IndoorNavi:ConnectionBoundary. A informação sobre a conexão dos espaços reside no modelo dual. A conexão conceitual entre as classes do modelo primal e do modelo dual é feita por IndoorCore:State e IndoorCore:Transition (Figura 3).

IndoorNavi:CellSpace é associado, por dualidade, com IndoorNavi:RouteNode, tal como IndoorNavi:TransitionBoundary é associado com IndoorNavi:RouteSegment (Figura 4).

No modelo dual é definida a propriedade IndoorNavi:connects, cujas instâncias associam uma instância de IndoorNavi:RouteSegment a duas instâncias de IndoorNavi:RouteNode, e representa a conexão entre dois nós, intermediada por um segmento (Figura 5). IndoorNavi:RouteSegment herda características da geometria de gml:Curve, o que permite que distâncias sejam representadas. No escopo da navegação em ambientes internos, âncoras para ambientes externos tornam-se desnecessárias.

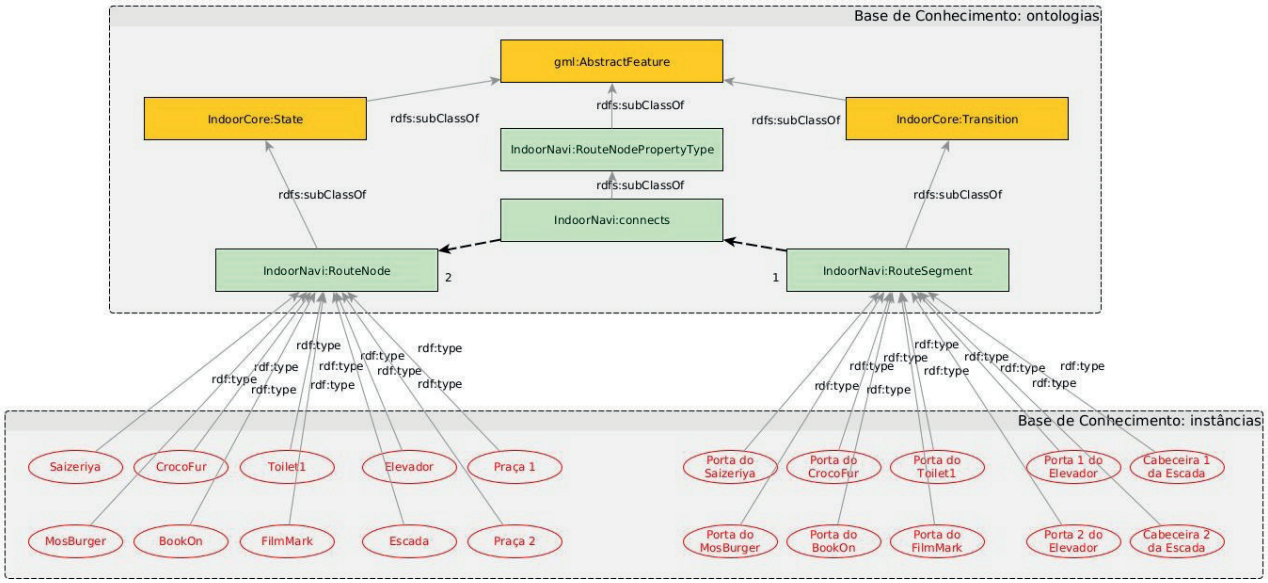
Em uma aplicação de busca de rota sobre o modelo neste nível de abstração do serviço, o usuário consulta o serviço por uma rota ao Saizeriya, informando que ele encontra-se no FilmMark . O sistema, a partir das instâncias armazenadas na base de conhecimento (Figura 6), executa um algoritmo de roteamento e sugere a rota: 1) passe pela porta do filmMark para a praça 2; 2) na praça 2 passe pela porta 2 do elevador para o elevador; 3) no elevador, passe pela porta 1 do elevador para a praça 1; 4) na praça 1 passe pela porta do Saizeriya para o Saizeriya (Figura 7).

Figura 3. No quadro superior, representação das classes definidas na ontologia e suas relações. No quadro inferior, instâncias que representam o shopping do exemplo.



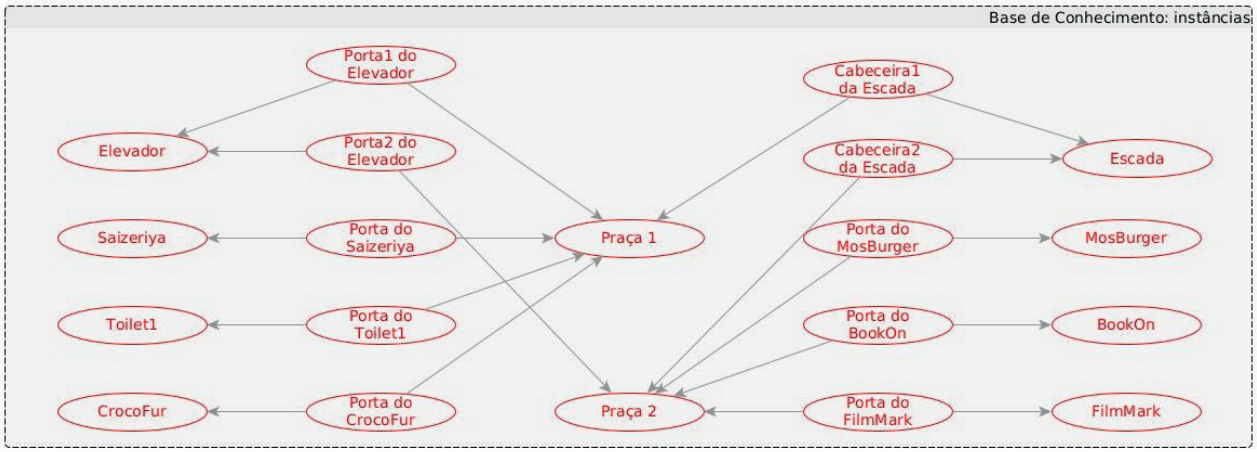
Fonte: os autores.

Figura 4. Transformação do modelo primal para o modelo dual.



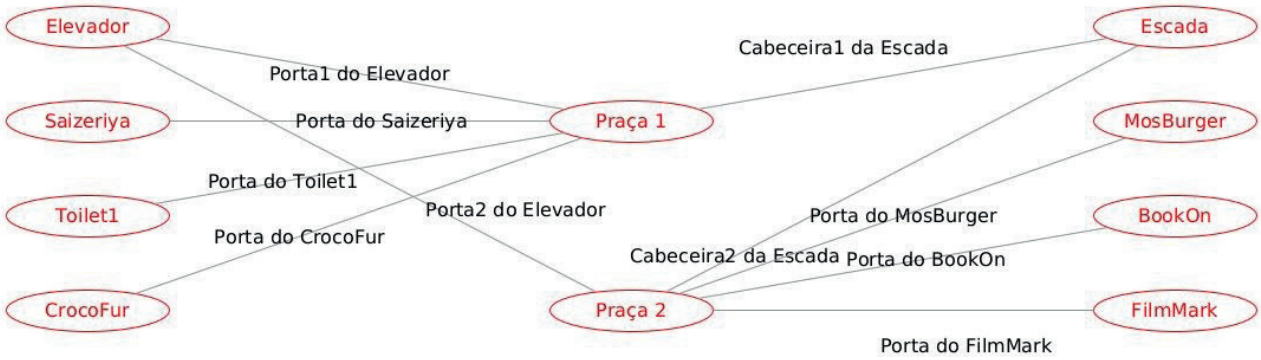
Fonte: os autores.

Figura 5. Apresentando somente as instâncias da base de conhecimento no modelo dual, reposicionadas em relação à **Figura 4.** Todas as arestas representam IndoorNavi:connects.



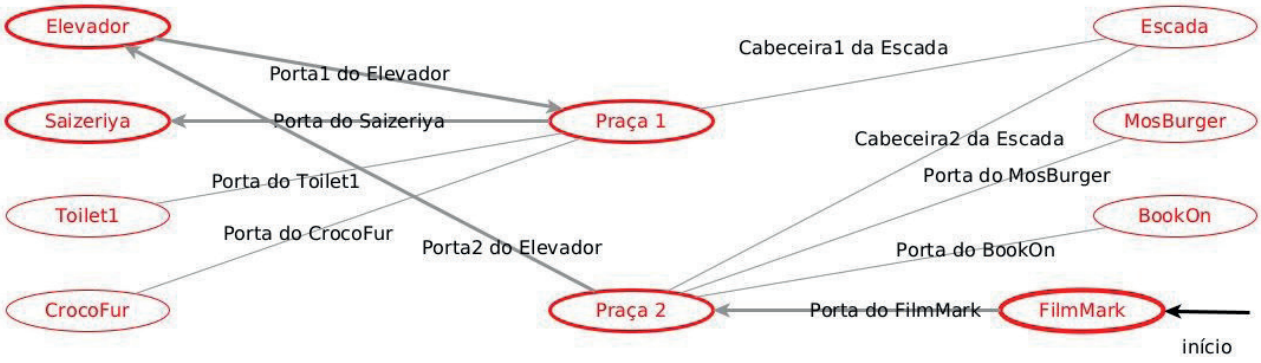
Fonte: os autores.

Figura 6. Ilustração substituindo instâncias de IndoorNavi:RouteSegment por arestas rotuladas.



Fonte: os autores.

Figura 7. Rota.



Fonte: Os autores.

Aplicativo Android

Para que o usuário consiga utilizar os serviços disponíveis nos servidores, desenvolvemos um aplicativo Android, em que o usuário dá como entrada ao sistema um comando por voz e o aplicativo o interpreta, cria uma consulta em SPARQL para o servidor de buscas

de serviços REST. Ao encontrar e chamar a função REST correspondente, envia resposta ao aplicativo, que informa ao usuário utilizando um sintetizador de voz.

Para utilizar o aplicativo, o usuário deve clicar em um botão, que inicia o reconhecimento de voz pelo aplicativo. Uma vez que esse comando, relacionado às funcionalidades descritas anteriormente, seja compreendido, o sistema criará tanto os parâmetros fundamentais da pesquisa no formato JSON-LD, como localidades, nome de produtos e favoritos, assim como criará também uma consulta descrita em SPARQL e enviará para o servidor de descoberta de serviços REST. Caso o servidor descubra algum serviço que contemple os parâmetros enviados, ele enviará a URL que localiza o serviço. O aplicativo Android irá, então, re-enviar os parâmetros para esse endereço.

O aplicativo também é responsável por comunicar ao usuário as informações recebidas dos servidores, seja informações do local onde o usuário está, bem como recomendar esta localidade caso ela seja marcada como favorita, informações sobre localidades relevantes a uma busca, ou, no caso de uma rota, instruções que o usuário deve seguir para chegar no seu destino desejado. Todas essas informações são notificadas pelo aplicativo ao usuário utilizando o seu próprio sintetizador de voz.

Servidor de descoberta de serviços REST

O servidor funciona da seguinte forma: recebendo uma chamada REST, com a consulta SPARQL, o servidor irá procurar dentro de seu diretório de serviços, aquele que atenda a essa consulta. Como esses serviços já são pré determinados, o servidor irá retornar a URL de localização desses serviços ao cliente.

Módulo de execução

O módulo de execução disponibiliza os serviços que podem ser descobertos. Cada serviço no módulo de execução é isolado dos outros, uma vez que, em um cenário real, seria comum que os serviços encontrados pelo descobridor de serviços estivessem em servidores totalmente diferentes. O módulo atual abrange as três funcionalidades básicas do sistema acrescidas de dois outros serviços de apoio. Descrevemos brevemente cada um deles a seguir.

Serviço de autenticação

O serviço de autenticação é necessário para guardar informações sobre o usuário, no sistema atual, o endereço de IP do aparelho móvel e a porta de escuta do aplicativo. Estas informações são utilizadas para notificar o aplicativo sobre sua posição atual.

Serviço de busca

O serviço de busca opera da seguinte forma: primeiramente, o serviço carrega as ontologias relativas ao sistema (i.e., o shopping modelado em OWL). Em seguida, ele executa a consulta SPARQL correspondente para achar a resposta dentro da Base de Conhecimento do Shopping. Finalmente, a resposta dessa busca é tratada e retornada ao aplicativo. As informações contidas no pacote de JSON de resposta para cada localidade encontrada são: o nome da localidade; a descrição; o nó; e se o local foi favoritado.

Serviço de favoritos

Este serviço é semelhante ao serviço de busca, com algumas mudanças: o serviço atualiza na Base de Conhecimento a localidade que deve ser curtida ou não. Ela responde para o aplicativo as informações atualizadas da localidade alterada.

Serviço de rota

Este serviço recebe, do pacote JSON vindo do aplicativo Android, uma instância com informação sobre dois nós, um inicial e outro final. Similar aos outros dois serviços anteriores, ele deve achar na Base de Conhecimento os nós recebidos. Achados, essa informação é usada por um programa em Java para o cálculo da rota. Quando o cálculo e finalizado, o serviço envia para o aplicativo um conjunto de pares de String, cada par contendo a descrição da rota que o usuário deficiente deve seguir, e o próximo nó por onde ele deverá passar caso siga as instruções corretamente, sendo que o último par contém o nó final.

Sensores de localidade

Os sensores de localidade utilizados no teste são *beacons*. Quando o *beacon* é detectado pelo aplicativo, este envia ao servidor o identificador do beacon (isto é, o identificador do nó pelo qual o usuário acabou de passar), e, utilizando as informações de IP e porta recebidas do aplicativo pelo serviço de autenticação, o serviço repassa essa informação do nó onde o usuário está localizado no momento, ajudando o aplicativo a se localizar. A opção por beacons é dominantemente econômica, pois se espera que seja grande a quantidade de sensores distribuídos no ambiente.

RESULTADOS

O sistema concebido utiliza as ontologias SSN/SOSA e IndoorGML, operacionalizado com o arcabouço Jena, detalha o uso das bases de conhecimento, apresentando claramente

o que são as instâncias, e o processo de obtenção do grafo de roteamento, que permite a aplicação de algoritmos de roteamento de complexidade variada.

Como foram utilizadas ontologias de diversos domínios e foram seguidos os padrões descritos pela W3C e OGC (Open Geospatial Consortium, 2020), é possível descrever diversos ambientes e desenvolver aplicativos compatíveis em domínio cruzado, como cidades inteligentes, sistemas assistivos. O funcionamento da arquitetura proposta engloba tanto como descrever pequenas áreas de uma casa, mas também ambientes grandes como escolas ou shoppings.

Notou-se que o sistema concebido pode ser generalizado com o uso de ontologias para serviços, que definem conceitos equivalentes aos definidos em WSML. Esta solução específica foi resolvida utilizando o conceito de procedimento definido em SOSA/SSN, o que enseja desdobramentos futuros.

DISCUSSÃO

A ferramenta simulada apresentou resultados similares aos obtidos em Magami, F. et. al (Magami, F. et al, 2017).

A abordagem proposta por Magami *et al*, por conta da adoção de WSML, apresentou dificuldades no seu desenvolvimento que impactaram na qualidade da solução final. Um dos problemas detectados foi no âmbito do motor de inferência IRIS, motor oficial para inferência na linguagem WSML. Apesar de WSML dar suporte ao desenvolvimento de regras recursivas, IRIS não consegue lidar com esse tipo de regra. Dessa forma, em Magami et al, regras tiveram de ser remodeladas ou até mesmo excluídas por conta dessa limitação. Tal limitação não existe mais na solução atual.

Adicionalmente, as ferramentas oficiais para manipulação de especificações WSML não apresentam suporte. As últimas versões apresentadas pelo repositório oficial do WSMT (The Web Service Modelling Toolkit) e do WSMX (Web Services Execution Environment) são de 2009. Por fim, embora a proposta do framework WSMX seja a de execução de serviços, seu desempenho é baixo e de difícil portabilidade, impossibilitando seu emprego para aplicações de IoT.

Para reconhecimento e síntese de voz, o sistema original adotou o Android Speech API (https://developer.android.com/reference/android/speech/package-summary) que evoluiu de forma significativa no decorrer dos anos. Na versão anterior ocorreram problemas com relação ao reconhecimento de voz, onde por limitações da época, foi adotado somente o inglês americano. Sendo assim, os problemas relatados em Magami et al foram solucionados na nova versão, inclusive o suporte ao português.

O uso de ferramentas com suporte da comunidade e de instituições normativas proporciona maior certeza de manutenção do desenvolvimento das ferramentas. Especificamente, a substituição de WSML, por OWL e ontologias sobre essa linguagem, que se estabeleceram como padrão, aumenta o poder de representação, flexibilidade e aplicabilidade do sistema proposto.

CONCLUSÃO / CONSIDERAÇÕES FINAIS

Esse trabalho teve como objetivo apresentar uma abordagem semântica para o desenvolvimento de sistemas voltados a pessoas com deficiência visual.

A combinação dos resultados obtidos nas versões deste trabalho permitem dizer que o objetivo foi atingido. A evolução obtida entre versões aumentou o poder de representação, flexibilidade e aplicabilidade do sistema proposto, como apresentado acima.

Há várias possibilidades de continuidade deste trabalho:

- sua generalização em um arcabouço para construção de aplicações que guiam clientes a serviços mais diversos, como transporte, saúde, serviços públicos, em diferentes espaços topográficos e conectados a modelos em escalas maiores, como cidades inteligentes;
- o ponto anterior implica na melhoria da descrição de serviços através da criação de uma ontologia própria para esta finalidade;
- com o uso dessa ontologia de serviços é possível explorar outras funcionalidades como a descoberta e composição automática de serviços (MENESES SILVA, et al., 2019);
- tratar aspectos de segurança de informação. Tanto no uso dos dados do usuário quanto na certificação de serviços e na comunicação entre eles;
- teste de novas tecnologias e dispositivos, como a funcionalidade de localização e orientação especificada em Bluetooth 5.1 (BLUETOOTH SPECIAL INTEREST GROUP, 2019).
- ampliar a quantidade de dispositivos interconectados passíveis de uso, através da adoção de uma camada que compatibilliza diversos protocolos com o uso da ontologia Thing Description descrita pela W3C (https://www.w3.org/2019/wot/td) e ferramentas como a proposta em Mozilla Web of Things (https://iot.mozilla.org/).

REFERÊNCIAS

1. AHMETOVIC, D. NavCog: turn-by-turn smartphone navigation assistant for people with visual impairments or blindness. *In*: W4A '16, 2016, Montreal Canada. **Proceedings of the 13th Web for All Conference** [...]. [*S. l.*]: ACM Press, 2016. p. 1–2.

2. ANTONIOU, Gregoris *et al*. **A Semantic Web Primer**. Third. ed. Cambridge, USA: MIT Press, 2012. 270 p.

3. APACHE SOFTWARE FOUNDATION. Apache Jena: A free and open source Java framework for building Semantic Web and Linked Data applications. disponível em : <https://jena.apache.org/>, acessado em: 26 out. 2020.

4. ATZORI, Luigi. The Internet of Things: A survey. **Computer Networks**, [*S. l.*], v. 54, n. 15, p. 2787-2805, 28 out. 2010.

5. BLUETOOTH SPECIAL INTEREST GROUP Enhancing Bluetooth Location Services with Direction Finding. Bluetooth SIG, 2019. Disponível em: https://www.bluetooth.com/bluetooth-resources/enhancing-bluetooth-location-services-with-direction-findin g/

6. BOTTA, A. et al. Integration of cloud computing and internet of things: a survey. **Future Generation Computer Systems**, Amsterdam, Holanda, 56:684–700, March, 2016. DOI: https://doi.org/10.1016/j.future.2015.09.021

7. BRICKLEY, D.; GUHA, R.V. Resource Description Framework (RDF) Schema Specification 1.1. W3C, 2014. Disponível em: https://www.w3.org/TR/2014/REC-rdf-schema-20140225/. Acesso: 09 de Nov., 2020.

8. CARROLL, J. et al. Jena: implementing the semantic web recommendations. In **Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters**, 2004. Association for Computing Machinery, New York, NY, USA, 74–83. DOI:https://doi.org/10.1145/1013367.1013381

9. CAI, Y.; AU YEUNG, Ching-man; LEUNG, Ho-fung. **Fuzzy Computational Ontologies in Contexts: Formal Models of Knowledge Representation with Membership Degree and Typicality of Objects, and Their Applications**. 2012, Berlim:Springer. ISBN 978-3642254550

10. CHIU, D. K. et al. Towards ubiquitous tourist service coordination and process integration: A collaborative travel agent system architecture with semantic web services. **Information Systems Frontiers**, Suíça, 11(3):241–256, July, 2009. DOI: https://doi.org/10.1007/s10796-008-9087-2

11. COMPTON, M. et al. The SSN ontology of the W3C semantic sensor network incubator group. **Journal of Web Semantics** v. 17, p. 25-32. 2012, DOI: h ttps://doi.org/10.1016/j.websem.2012.05.003.

12. DAKOPOULOS, D. *et al*. Wearable Obstacle Avoidance Electronic Travel Aids for Blind: A Survey. **IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS**, [*S. l.*], v. 40, n. 1, p. 25-35, 1 jan. 2010.

13. DB-ENGINES. **DB-Engines Ranking**. [*S. l.*], 2020. Disponível em: https://db-engines.com/en/ranking. Acesso em: 18 nov. 2020.

14. DE, Suparna *et al*. Ontologies and Context Modelling for the Web of things. *In*: SHENG, Quan Z. *et al*, (ed.). **Managing the Web of Things**: Linking the Real World to the Web. Cambridge, MA: Morgan Kaufmann Pubs., 2017. cap. 1, p. 3-36. ISBN 978-0-12-809764-9.

15. DUCHARME, Bob. **Learning SPARQL**: Querying and Updating with SPARQL 1.1. 2nd.. ed. Sebastopol, CA: O'Reilly Media, Inc, 2013. 366 p. ISBN 978-1-449-37143-2.

16. FARCY R., Leroux R., et al. Electronic Travel Aids and Electronic Orientation Aidsfor Blind People: Technical, Rehabilitation and Everyday Points of View, **Proceedings of the Conference and Workshop on Assistive Technologies for People With Vision and Hearing Impairments**, 2006.

17. FENSEL, Dieter. **Knowledge Graphs**: Methodology, Tools and Selected Use Cases. Switzerland: Springer Nature, 2020. 147 p. ISBN 978-3-030-37438-9.

18. FENSEL, D. et al. **Semantic web services**. 2012, Berlim:Springer. ISBN 978-3-642-19193-0.

19. FENSEL, D.; BUSSLER, C. The web service modeling framework WSMF. **Electronic Commerce Research and Applications**, 1(2):113–137. 2002.

20. FRANZ INC. **AlegroGraph**. [*S. l.*], 2020. Disponível em: https://allegrograph.com/. Acesso em: 18 nov. 2020.

21. GRUBER, T. R. A Translation Approach to Portable Ontology Specifications. **Knowledge Acquisition**, 5(2):199-220, 1993.

22. GUARINO, N.; OBERLE, D.; STAAB, S.. What is an ontology? In: STAAB, Steffen; STUDER, Rudi (Eds.) **Handbook on ontologies**, 2009, Springer, Berlin, p. 1–17. DOI: h ttps://doi.org/10.1007/978-3-540-92673-3

23. HEATH, Tom; BIZER, Christian. **Linked Data Evolving the Web into a Global Data Space**. [*S. l.*]: Morgan & Claypool, 2011. 122 p.

24. HOGAN, Aidan. **Web of Data**. Switzerland: Springer Nature, 2020. 680 p. ISBN 978-3-030-51579-9.

25. HUI, J. and THUBERT, P. (2011) **Rfc 6282: Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks**, IETF, Fremont, CA, USA.

26. JANOWICZ, K. et al. SOSA: A lightweight ontology for sensors, observations, samples, and actuators.**Journal of Web Semantics**, v. 56, p. 1-10. 2019. DOI: https://doi.org/10.1016/j.websem.2018.06.003.

27. JARA, Antonio *et al*. Semantic Web of things: am analysis of the application semanticsfor the IoT moving towards the IoT convergence. **International Journal of Web and Grid Services**, [*s. l.*], v. 10, n. 2/3, p. 244-72, 2014.

28. JARA, Antonio *et al*. Glowbal IP: an adaptive and transparent IPv6 integration in the Internet of Things. **Mobile Information Systems**, [*S. l.*], ano 2012, v. 8, n. 3, p. 177–197, 2012.

29. KANG, H.-K.; LI, K.-J. A Standard Indoor Spatial Data Model: OGC IndoorGML and Implementation Approaches. **ISPRS Int. J. Geo-Inf.** 2017, 6, 116.

30. LEE, Ho-Kyoung *et al*. Comparison Shopping Systems based on Semantic Web - A Case Study of purchasing Cameras. *In*: INTERNATIONAL CONFERENCE ON GRID AND COOPERATIVE COMPUTING, 2003, Shanghai, China. **Proceedings of the Second International Workshop, GCC 2003** [...]. [*S. l.*]: Springer Verlag, 2003.

31. MAGAMI, F. M. *et al*. Abordagem semântica para o desenvolvimento de sistemas de suporte a ambientes inteligentes inclusivos. *In*: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO UBÍQUA E PERVASIVA, 2017, São Paulo, Brasil. **Anais do IX Simpósio Brasileiro de Computação Ubíqua e Pervasiva** [...]. [*S. l.*]: SBC, 2017.

32. MANATE, B. *et al*. Towards a smarter internet of things: semantic visions. *In*: 2014 EIGHTH INTERNATIONAL CONFERENCE ON COMPLEX, INTELLIGENT AND SOFTWARE INTENSIVE SYSTEMS (CISIS), 2014, Birmingham, UK. **Proceedings of the 8th CISIS** [...]. [*S. l.*: *s. n.*], 2014. p. 582-7.

33. MARKLOGIC CORPORATION. **The data platform that simplifies data integration**. [*S. l.*], 2020. Disponível em: https://www.marklogic.com/. Acesso em: 20 nov. 2020.

34. MENESES SILVA, André Luis *et al*. Interoperability in Semantic Web of Things: design issues and solutions. **International Journal of Communication Systems**, [*S. l.*], ano 2019, v. 32, n. 6, p. 1-27, 31 jan. 2019.

35. OPENLINK SOFTWARE. **Data-driven agility without compromise**. [*S. l.*], 2019. Disponível em: https://virtuoso.openlinksw.com/. Acesso em: 20 nov. 2020.

36. OVIDIU, V. et al. **Internet of things strategic research roadmap**. In: Internet of things – global technological and societal trends. Series in communications. River Publishers; 2011. p. 9–52.

37. OPEN GEOSPATIAL CONSORTIUM About OGC **Open Geospatial Consortium**, 2020. Disponível em: https://www.ogc.org/about

38. OWL OWL 2 Web Ontology Language Document Overview (Second Edition). 11 dez. 2012. disponível em <https://www.w3.org/TR/owl-overview/>, acesso em 26 out 2020.

39. PARK, H.; YOON, A.; and KWON, H. Task model and task ontology for intelligent tourist information service. **International Journal of u-and e-Service, Science and Technology**, 5(2):43-58, June, 2012.

40. PASCOLINI, D.; MARIOTTI, S. P. Global estimates of visual impairment: 2010. **British Journal of Ophthalmology**, May, 2012 ;96(5):614-8. doi: 10.1136/bjophthalmol-2011-300539.

41. PLESSERS, P. et al. Accessibility: A Web Engineering Approach. *In*: WWW: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 2005, Chiba, Japan. **Proceedings of the 14th WWW2005** [...]. [*S. l.*]: ACM Press, 2005. p. 353-62.

42. RDF RDF 1.1 Concepts and Abstract Syntax. 25 fev. 2014. disponível em: <https://www.w3.org/TR/rdf11-concepts/>, acessado em 26 out. 2020.

43. SATO, D. *et al*. NavCog3: An Evaluation of a Smartphone-Based Blind Indoor Navigation Assistant with Semantic Features in a Large-Scale Environment. *In*: ASSETS'17, 2017, Baltimore, MD, USA. **Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility** [...]. [*S. l.*]: ACM Press, 2017. p. 270–279.

44. SHELBY, Z. et al. **Constrained application protocol (CoAP)**. [*S. l.*], 2013. Disponível em: http://tools.ietf.org/html/draft-ietf-core-coap-14. Acesso em: 19 nov. 2020.

45. STEFANOV, D. H.; BIEN, Z.; and BANG, W.. The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives. **IEEE transactions on neural systems and rehabilitation engineering**, 12(2):228– 250, June, 2004, DOI: 10.1109/TNSRE.2004.828423.

46. STUDER, R. *et al*. Knowledge engineering: Principles and methods. **Data & Knowledge Engineering**, [*s. l.*], v. 25, n. 1/2, p. 161-198, 1998.

47. PERRY, M.; HERRING, J. OGC GeoSPARQL: A Geographic Query Language for RDF Data. 2012 Open Geospatial Consortium. Disponível em: <http://www.opengeospatial.org/standards/geosparql>

48. SHIAO, M. **Internet of things standardisation and architectures**. workshop report. Technical report. Alliance for Internet of Things Innovation (AIOTI); 2015.

49. W3C. World Wide Web Consortium Standards. disponível em <https://www.w3.org/standards/> acesso em 26 out 2020.

50. WEI, Wang *et al*. An experimental study on geospatial indexing for sensor service discovery. **Expert Syst Appl.**, [*S. l.*], v. 42, n. 7, p. 3528-38, 2015.

51. WEBER, R. H. and WEBER, R.. **Internet of Things: Legal Perspectives**. Berlim:Springer, 2010. ISBN 978-3-642-11710-7

52. YU, L. **A developer's guide to the semantic Web**. Berlim:Springer, 2011. DOI https://doi.org/10.1007/978-3-642-15970-1

15

WIKIPOLIS: Cibercidades entre o atual e o virtual

Reydi **Kamimura**
Unesp

doi 10.37885/201102151

RESUMO

Transições espaciais. Com base na teoria de espaços antropológicos de Pierre Lévy, o presente artigo tem por objetivo contextualizar o surgimento do ciberespaço e sua potencialidade de se constituir um novo espaço antropológico, o "espaço do saber", com o atual cenário de instabilidade social em que as cidades brasileiras estão inseridas, elucidando a realidade mista entre o virtual e o atual na qual vive a sociedade contemporânea e apontando, destarte, para a necessidade da elaboração de um projeto de cidade que melhor se adeque para a habitação humana nessa sua nova realidade. Para tanto, tal trabalho busca primeiramente analisar o processo de conformação urbana e social das cidades atuais, assim como, a tendência conceitual pela qual se desenvolveu o ciberespaço, consolidando o virtual no cotidiano da sociedade, para que, na sequência, se estude a forma como se deu esse tipo de interação entre o mundo atual e o mundo virtual revelados por recentes manifestações populares, que insurgiram em cidades do Brasil e do mundo globalizado, os *Occupy*'s. A partir de tal análise, o estudo busca diagnosticar as potencialidades e limitações apresentadas nessa interação, revelando que o projeto do "espaço do saber" para a criação de uma "inteligência coletiva" ainda se apresenta inconclusa na presente estruturação informativa e comunicativa do ciberespaço. O trabalho, então, conclui-se com uma proposta de construção de uma interface gráfica no ciberespaço, o Wikipolis, capaz de criar cibercidades, uma nova camada eletrônica de cidade para a habitação humana no mundo virtual, com potencialidade interativa e imersiva para a constituição de um novo espaço antropológico que possa colocar corretamente os inúmeros problemas enfrentados pela sociedade em meio à nova realidade humana.

Palavras-chave: Cidades, Ciberespaço, Occupy's, Realidade Mista, Cibercidades.

INTRODUÇÃO: O APONTAR DE UM NOVO ESPAÇO ANTROPOLÓGICO

Este artigo procura abordar de forma crítica o atual cenário urbano social de evidentes instabilidades e transformações que deixam a população brasileira – e outras em todo o mundo globalizado - à deriva dos obscuros interesses e decisões dos detentores do poder do sistema político econômico em vigência. O presente contexto histórico se apresenta único, visto a consolidação de espaços virtuais de sociabilização e informação no cotidiano da sociedade contemporânea contidos nas ferramentas e interfaces gráficas do ciberespaço, e que, como apresentado em recentes manifestações populares que atingiram escalas globais, como foram os *Occupy*'s, têm a potencialidade de interagir e se relacionar diretamente na vida urbana das cidades atuais. E é nessa interação entre a realidade atual das cidades e a realidade virtual do ciberespaço que o presente artigo busca analisar, para que, a partir da definição dessa realidade mista hodiernamente vivida, justifique sobre a verdadeira necessidade de formulação de uma nova plataforma para habitação humana mais adequada à sua nova condição.

Para isso, o artigo se embasa nas teorias do antropólogo francês Pierre Lévy sobre a capacidade do ciberespaço se tornar um novo espaço antropológico "sistema de proximidade próprio do mundo humano, e portanto dependente de técnicas, de significações, da linguagem, da cultura, das convenções, das representações, e das emoções humanas" (LÉVY, 1994.) –, o "Espaço do Saber", que organizasse e filtrasse a informação para a orientação e navegação de seus usuários em vias de reunir suas forças mentais para a reconstrução dos laços sociais através do aprendizado e da troca de saberes recíproca de indivíduos valorizados em uma forma de transmutação efetiva das diferenças em riquezas coletivas para a formação de uma "inteligência coletiva". O "Espaço do Saber" teria a vocação de comandar os espaços anteriores e não de substituí-los ou suprimi-los, pelo contrário, sua instauração permitiria pôr corretamente e talvez resolver inúmeros problemas crucias que já não encontram formulação adequada nos conceitos e ferramentas que os exprimem nos espaços anteriores. Para Lévy, cada espaço antropológico deve agir apenas sobre si mesmo, interagindo com os outros espaços em uma relação de "causalidade sem contato", enquanto que a supressão de um sobre o outro caracterizaria uma situação de caos. Os espaços antropológicos precedentes são: a Terra, e a ligação do homem com o cosmo e com o próprio homem; o Território, e a criação de entidades territoriais definidas por fronteiras como a agricultura, as cidades e o Estado; e o Mercantil, constituído pelos fluxos de mercadorias, de energias, de informações, etc.

CIDADES LTDAS: O GRITO DE UMA SOCIEDADE

Cidades são artefatos construídos pelo homem para a organização de sua vida em comum, estruturando, assim, a vida cotidiana de determinada sociedade em um determinado espaço e em um determinado tempo. Seus modos de organização espacial ecoam e amplificam um conjunto de valores que implicam em uma visão de mundo que dada sociedade tem sobre seu respectivo contexto histórico. E, assim como as cidades-estados gregas estimulavam o debate político sobre a interação humana com seu território e com seus semelhantes através das ágoras e os burgos medievais exprimiam sua devoção à ordem sagrada envoltos às catedrais góticas e aos castelos reais, as cidades globalizadas organizam-se espacialmente conforme as regras e os interesses mercadológicos do capital financeiro. O processo de expansão capitalista do mercado transformou as cidades atuais não apenas em um objeto mercadoria para acumulação capitalista, mas também em um sujeito econômico integrante do mercado global.

Com o assentamento da geopolítica mundial em um novo regime internacional sob hegemonia norte-americana e a criação de instituições financeiras definidoras dos paradigmas para gestão monetária em escalas globais como o Fundo Monetário Internacional (FMI), o Banco Mundial e o Banco Interamericano de Desenvolvimento (BID) após o fim da Segunda Guerra Mundial, firmaram-se as práticas capitalistas de produção e gestão urbana através de exigências impostas em contrapartidas a empréstimos e financiamentos que as cidades, principalmente dos países periféricos, necessitavam para suprir suas demandas, limitando-as gradualmente a políticas públicas de ajustamentos fiscais, perante seu endividamento permanente, que favorecem o capital financeiro e contrariam os carecimentos de seu povo[2].

Essas medidas, culminados na formulação do Consenso de Washington, consolidaram o receituário neoliberal para a gestão urbana através do estabelecimento de parcerias público-privados para o planejamento urbano estratégico[3] em pontos que visam, sobretudo, ambições mercadológicas. A busca por investimentos privados para a construção urbana inseriu as cidades em um sistema de competição interurbana, passando a ser um ator econômico que deve vender uma boa imagem de prosperidade urbana e social, através do *marketing* urbano e da invisibilidade dos pobres em áreas de exclusão4, para que consigam atrair os melhores investimentos do mercado global em concorrência com outras cidades. Para isso, os gestores urbanos são ordenados pelos interesses do mercado a se focar em promover uma urbanização de alto padrão concentrada em pontos estratégicos para a ampliação da base de acumulação do capital, ao mesmo tempo que as práticas neoliberais privatizam empresas e instituições públicas e enfraquecem ainda mais as políticas públicas em questões socioambientais para o resto da cidade.

A crise de representatividade popular nas políticas e tomadas de decisões públicas fez com que eclodissem em diversas cidades do mundo financeiramente globalizado manifestações sociais de repúdio ao sistema político-econômico vigente, conhecidas como *Occupy*'s. As manifestações de protesto, que começaram em 2011 em países do norte africano na denominada Primavera Árabe e se espalharam em países europeus e americanos como os casos da Geração à Rasca de Portugal, os Indignados da Espanha, o *Occupy Wall Street* dos Estados Unidos e o *Yo Soy 132!* do México, se relacionam por serem respostas críticas radicais ao capitalismo como forma de produção e reprodução da vida social através da ocupação coletiva de espaços públicos como tática de luta e de exposição das misérias e contradições do sistema atual.

No Brasil, o processo de expansão capitalista produziu cidades segregadas espacial e socialmente em periferias distantes dos centros urbanos e privados do acesso aos serviços e direitos oferecidos pelas cidades, em contraposição à construção de largas avenidas e viadutos e de luxuosos empreendimentos de convivências reificadas em áreas pontuais. Insatisfeitos com as políticas públicas em prática, uma parte da população brasileira mobilizou, no inverno brasileiro de 2013, manifestações que ocuparam as ruas e vias públicas das cidades, em uma forma de determinação direta da população sobre os fluxos e usos desses espaços públicos e de exposição do sistema de transporte caótico das metrópoles brasileiras que privilegia o transporte particular individualizado ao transporte público coletivo, nisso que foi chamado de Jornadas de Junho de 2013.

No final do mesmo ano, outra movimentação popular, desta vez, hegemonicamente formada por jovens das periferias urbanas, mobilizou ocupações coletivas que evidenciaram as contradições da lógica neoliberal dos *shopping centers*, os Rolezinhos. O abandono político e a dominação espacial pelos carros nos espaços públicos urbanos fizeram das ruas lugares a serem evitados pelos jovens, sendo os *shopping centers*, e sua promessa de um passeio livre e solto, consolidados como o local que a sociedade estabeleceu como substituto das ruas. No entanto, a reação repressiva e discriminatória a essas reuniões juvenis, que levavam consigo a música e a cultura periférica do funk, evidenciou as regras claras e rígidas ao acesso daqueles que não estão lá para desfrutar dos privilégios dos cultos ao espetáculo do consumo que esses espaços representam: mesmo que circulem sem recursos para consumir, os pobres devem negar sua classe e apenas desejar consumir. Os Rolezinhos, porém, representam a afirmação dos jovens de sua classe num espaço que existe para negá-la; eles são um pedido dos jovens que querem a rua de volta.

Essas manifestações apresentaram em comum a luta da sociedade globalizada pela conquista do direito à cidade de cada cidadão e a reinvindicação por sistemas políticos participativos através de ocupações coletivas de espaços públicos em organizações horizontais

para exposição ao público das mazelas do capitalismo neoliberal sobre as cidades e as sociedades atuais. Ademais, tais movimentações foram demonstrações da interação entre os meios físicos das cidades atuais, os espaços públicos, e os meios digitais do ciberespaço, as redes virtuais. As mobilizações e repercussões atingidas por elas foram exponencialmente maximizadas graças ao uso das ferramentas comunicativas e informativas da Internet por seus manifestantes que definiram suas insurgências e seus encaminhamentos, evidenciando, assim, a constituição hodierna de uma realidade mista entre o atual e o virtual.

A REALIDADE MISTA E O PLANEJAMENTO DE CIBERCIDADES

Uma peculiaridade apresentada nessas recentes manifestações foi a presença das redes sociais dispostas na Internet e o papel fundamental que tiveram para suas mobilizações e organizações, como também, para seus encaminhamentos finais. A interatividade e imersividade das interfaces gráficas das redes sociais permitem criar ambientes virtuais de comunicação e informação onde todo usuário pode representar a si próprio através de seus perfis pessoais, emitindo e recebendo informações autonomamente. Isso potencializou o desenvolvimento de mídias virtuais com base na comunicação comunitária e na imprensa alternativa que utilizam as infovias da Internet à procura de uma comunicação mais próxima ao seu público. A adesão dessas ferramentais digitais por organizações sociais e por indivíduos indignados com a conjuntura atual permitiu com que estes se reunissem dentro de ambientes virtuais, estruturando suas ações e disseminando seus protestos, para a organização das manifestações que tomaram os espaços públicos de cidades em escalas globais.

Contudo, a consolidação das tecnologias digitais do ciberespaço nas sociedades globais despertou o interesse também de grandes corporações midiáticas pelas redes sociais e suas potencialidades difusoras como um novo mercado a ser explorado. A disputa entre as mídias alternativas e as grandes mídias tradicionais nas redes virtuais foi sintomático durante tais movimentações sociais. Como alertara o filósofo esloveno Slavoj Žižek[5], é preciso cuidado com os “falsos amigos” que fingem apoio aos manifestantes, mas que, no fundo, estão trabalhando duro para a diluição e desmoralização do movimento. Nos casos brasileiros, essa ação foi crucial para o decorrer das manifestações. Nas Jornadas de Junho, a forte adesão popular aos protestos fez com que a cobertura da grande mídia brasileira sobre os atos passassem da defesa à repressão policial para o apoio aos manifestantes. No entanto, conjuntamente ao apoio da mídia corporativa, o encaminhamento dos atos convocados pelos movimentos sociais organizadores tiveram claramente sua pauta diluída de questões específicas acerca do transporte público urbano e com objetivos mais claros para uma indignação generalizada antipartidária, dispersando seus objetivos originais e fugindo do controle dos organizadores iniciais. Já no caso dos Rolezinhos, sua insurgência mais espontânea

através de eventos criados por jovens nas redes sociais fizeram com que a grande mídia classificasse os atos como meras "brincadeiras" juvenis, em uma clara demonstração de tentativa de desmoralização do apelo político contido na atitude desses jovens defendido por movimentos e ativistas sociais, que foram refletidos na prevalência de comentários de cunho racistas e discriminatórios, como revelados em pesquisa feita sobre as reações nas redes sociais sobre os Rolezinhos[6].

Como se viu, apesar das redes sociais possibilitarem a criação de mídias alternativas para uso pessoal e social, elas também ampliam o campo de ação das práticas de mercado de grandes corporações e dos Estados neoliberais sobre esses usuários. Essas recentes manifestações, como afirma historiador Lincoln Secco, "são revoltas disseminadas pelas redes sociais nas quais as pessoas agem como singularidades, mas o conjunto é construído pela reação calculada dos donos do poder e da informação" (SECCO, 2013). Isso porque as redes sociais são produtos do desenvolvimento de um processo evolutivo do ciberespaço que tendenciou rumo à personalização da informação que favorece a dominação das ideologias conservadoras presentes nas mídias e representatividades políticas tradicionais sobre a maior parte das informações em circulação. Desde o desenvolvimento dos computadores pessoais (PCs), as tecnologias e ferramentas do ciberespaço vem em um processo de adaptação para as necessidades e preferências individuais dos usuários, disseminando-se pela sociedade através do avanço técnico em vias da portabilidade sem fio e da conectividade global, que permitiram uma comunicação personalizada com a inclusão do maior número de pessoas possível na qual a pessoa é o centro da informação. Tendência essa norteada pela busca do maior número de consumidores e, consecutivamente, da maior rentabilidade dessas inovações tecnológicas. Deste modo, os usuários do ciberespaço tornam-se consumidores de informação, que, com a criação das páginas pessoais das redes sociais, passam a oferecer seus perfis detalhados e especializados de consumidor para atender às lógicas capitalistas que regem esses ambientes virtuais. Ou seja, em nome da interatividade e imersão dos usuários, as interfaces gráficas das redes sociais, na verdade, abrem um novo espaço passivo de novas formas de controle e exploração dos "donos do poder e da informação" sobre a produção de conteúdo informativo que, por sua vez, encontram, em suas conexões e mecanismos, meios eficientes para atingir uma gama maior de consumidores através de maneiras diversas que personalizam e adequam a informação para cada tipo de perfil de consumidores específicos.

A partir de tal análise, é possível afirmar que as interfaces de redes sociais abrem as portas das potencialidades do ciberespaço para a apropriação midiática corporativa, logo, para o atendimento dos interesses do mercado; elas fazem da informação do ciberespaço uma mercadoria. Para libertar de seu confinamento perante as lógicas mercantis, é proposto

aqui uma virada da tendência personalista do desenvolvimento do ciberespaço para a coletivização de suas informações e mecanismos por meio da construção de cibercidades.

O planejamento de interfaces gráficas do ciberespaço historicamente se utilizou de analogias de referências físicas para atingir uma melhor associação e adaptação do usuário ao ambiente virtual criado, como, por exemplo, nos casos do *desktop* e das janelas virtuais que utilizam-se da analogia das escrivaninhas como ambiente de trabalho e das janelas como portais para novos ambientes. A ideia de criação de cibercidades faz uso de analogias das potencialidades das cidades reais de constituírem espaços públicos autênticos para a consolidação do ciberespaço como um reino verdadeiramente público onde usuários individualmente valorizados possam circular livremente em um espaço-informação de conteúdo público, fomentando a formação de coletivos inteligentes que tratem adequadamente as questões reais que assolam o mundo atual de maneira mais independente às propensões mercadológicas. Assim como os *Occupy*'s utilizaram da ocupação coletiva de espaços públicos urbanos para tornar públicos seus gritos e reivindicações, a ocupação de espaços públicos virtuais, as cibercidades, possibilitará a suas informações e atividades também se tornarem públicas, libertando o ciberespaço dos confinamentos que impedem o firmamento do "Espaço do Saber".

WIKIPOLIS.COM

É exposto agora o projeto de interface gráfica do Wikipolis, um ensaio para a criação de cibercidades dentro da Internet como um meio que utilize dos recursos virtuais do ciberespaço para dar visibilidade às cidades reais em sua integridade e expor suas informações públicas, promovendo um espaço público virtual de debates e ativismos democráticos para, assim, romper com a ideologia hegemônica dos interesses globalizadores do capital financeiro e, talvez, equacionar corretamente as incógnitas vividas atualmente em busca do positivo equilíbrio entre os espaços antropológicos em meio a esse turbulento período de transições espaciais que passam as sociedades e as cidades contemporâneas.

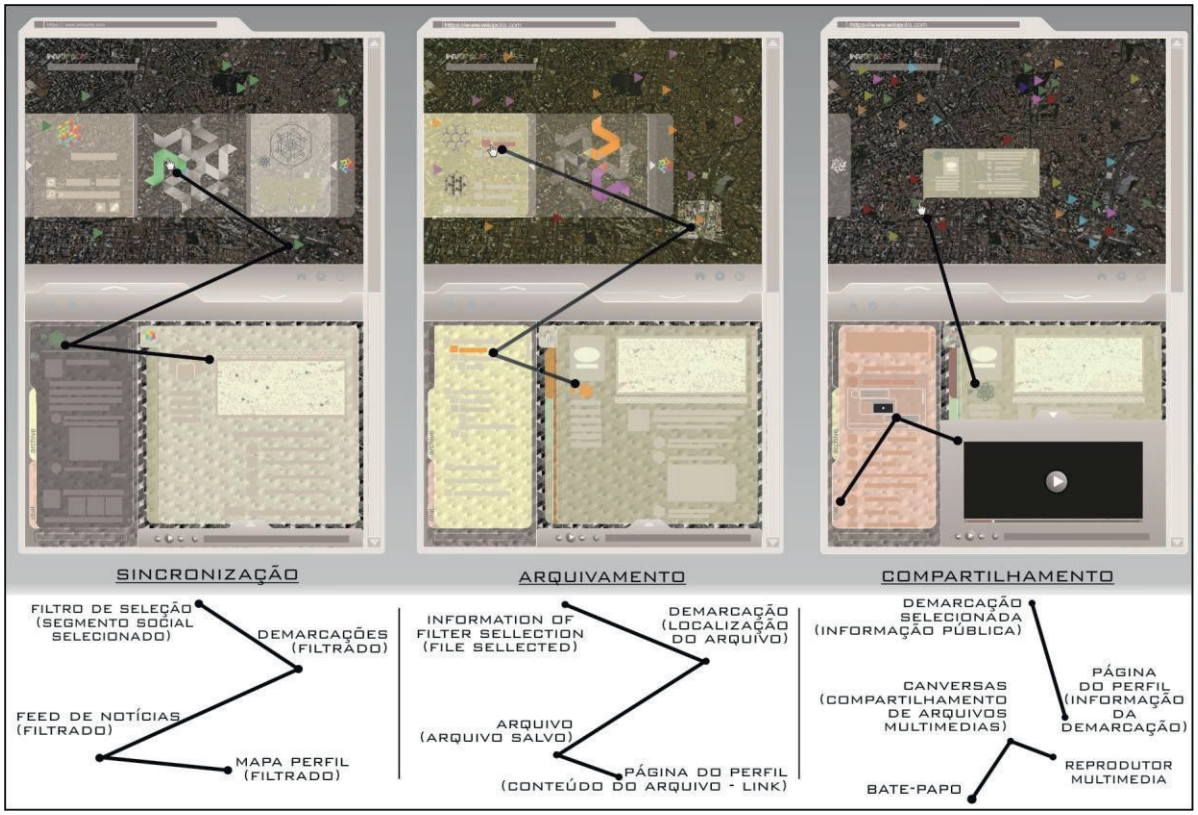
Figura 1. Interface gráfica e avatares.



A interface do wikipolis.com se caracterizaria por:

I. Wikipolis: (*wiki*-), termo vindo do idioma havaiano que significa “extremamente rápido” e é utilizado para identificar tipos de *softwares* colaborativos + (-*polis*), termo de origem grega que exprime a ideia de cidade;
II. Cibercidades: associação das informações de interesse público às localidades geográficas a que se referem através de demarcações em mapas virtuais das cidades atuais que se conectam ao conteúdo informativo contido no ciberespaço;
III. Oferecer uma visão geral das cidades atuais para conscientização dos usuários sobre a realidade em seus diversos pontos de vistas, incentivando à construção de coletivos inteligentes por meio das potencialidades do ciberespaço;
IV. Sincronização: agregar as potencialidades de encontro, circulação e trocas das cidades reais com as potencialidades comunicativas e informativas digitais do ciberespaço;
V. Alto grau de imersão dos usuários através de avatares representativos dos diversos segmentos sociais que atuam na construção das cidades;

Figura 2. Modos de ativação



VI. Ícone: figura para seleção e filtragem das informações das cibercidades por avatar;
VII. Outras ferramentas de filtragem para pesquisas textuais e também para pesquisas temporais, que selecionam as informações por meio de palavras-chaves e pelo período de tempo de sua última atualização;
VIII. Memória: ferramentas de arquivamento de informações para consultas posteriores e criação de mapas perfis para exposição das atividades realizadas acerca das cibercidades de cada avatar;
IX. Compartilhamento: espaço público para troca de informações públicas + ferramentas do ciberespaço para conversas pessoais;
X. Arquitetura líquida: interface gráfica moldada conforme utilização do usuário através de abas retráteis e modos de ativação;

CONCLUSÃO

O planejamento da interface gráfica do Wikipolis criaria uma nova camada eletrônica de informações e conexões virtuais sobre as cidades atuais. As cibercidades do Wikipolis dariam para os seus usuários uma visão geral sobre a espacialidade das cidades e as diversas atividades humanas que elas abrigam, libertando-as do controle e manipulação dos corporativismos empresariais e estatais acerca de suas informações de interesse público. E, através da utilização de seus avatares, os cidadãos poderão ter participação e

atuação em discussões e decisões a respeito das cidades em um processo de coletivização dos saberes e competências individuais que se culminaria na formação de uma "inteligência coletiva" abrigada no ciberespaço, efetivando-o no "Espaço do saber". Logo, a criação de cibercidades significaria, sobretudo, em uma virada na consciência de cada usuário pelo reconhecimento das diferenças individuais e da multiplicidade humana como riquezas coletivas através de um espaço para reunião dessas forças mentais em trocas de saberes recíprocos para a constituição de intelectuais coletivos que transformariam as perspectivas da sociedade atual sobre suas cidades, que, por conseguinte, passariam a refletir espacialmente essa nova cidadania.

NOTAS

[2] ver o processo de ajustamento fiscal permanente das cidades em ARANTES, 2006.

[3] ver sobre o discurso estratégico dos planejamentos urbanos em VAINER, 2000.

[4] ver sobre a conformação urbana resultantes das políticas globalizadoras em MARICATO, 2007.

[5] em texto ampliado pelo autor de seu discurso "The violent Silence of a New Begining", feito em Zuccotti Park (NY), em 10 de out. de 2011. (ŽIŽEK, 2012)

[6] pesquisa feita sobre comentários em blogs e redes sociais da Internet sobre os rolezinhos, em MACHADO; SCOLCO, 2014.

REFERÊNCIAS

1. ARANTES, P. F. O ajuste urbano: As políticas do Banco Mundial e do BID para as cidades. In: Revista Pósfau, nº 20, São Paulo, 2006.
2. LÉVY, P. A inteligência coletiva: por uma antropologia do ciberespaço. São Paulo: Edições Loyola, 2001.
3. MACHADO, R. P; SCALCO, L. M. Rolezinhos: Marcas, consumo e segregação no Brasil. 2014. In: Revista de Estudos Culturais nº 1 - USP, São Paulo, 2014.
4. MARICATO, E. Globalização e política urbana na periferia do capitalismo. In: RIBEIRO, L. C. Q; JUNIOR, O. A. S. As metrópoles e a questão social brasileira, Revan, 2007.
5. SECCO, L. As Jornadas de Junho. In: MARICATO, E. et al. Cidades rebeldes: Passe Livre e as manifestações que tomaram as ruas do Brasil. Boitempo: Carta Maior, 1ª ed., p. 71-79, São Paulo, 2013.
6. VAINER, C. B. Pátria, empresa e mercadoria: Notas sobre a estratégia discursiva do Planejamento Estratégico Urbano. In: ARANTES, O. B. F.; VAINER, C; MARICATO, E. A Cidade do Pensamento Único. Petrópolis, 2000.

7. ŽIŽEK, S. O violento silêncio de um novo começo. In: HARVEY, D. et al; [tradução: PESCHANSKI, J. A. et al.] Boi Tempo: Carta Maior, p. 15-27, São Paulo, 2012.

SOBRE O ORGANIZADOR

Prof. Me. 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, possui Pós-Graduação em Tecnologia em Gestão da Informação, Graduação em Ciência da Computação e Graduação em Sistemas de Informação. Professor de Informática no 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), certificado pelo IFG no CNPq. ORCID: https://orcid.org/0000-0002-1543-1108 .

ÍNDICE REMISSIVO

A

Algoritmo Polinomial: 11

Análise de Desempenho: 45

Análise de Programas: 170

Aplicativos: 34, 40, 63, 68

Autismo: 24, 25, 26, 27, 33, 37, 39, 40, 41, 42

Avaliação de Software: 7, 23, 43

Avaliação do Comportamento Dinâmico do Sistema de Memória: 45

C

Cadeia de Caracteres: 11

Cibercidades: 9, 225, 233

Cidades: 228, 235

Cor: 74

D

Deficiente Visual: 202

E

Educação Especial: 40

Engenharia de Software: 33, 88

Ensino: 41, 71, 120, 137, 148

Estações: 74, 84

Execução em Massa: 164, 169

F

Ferramenta de Benchmark: 45

Formação: 43

G

Gerenciamento de Projetos: 88

I

Infraestrutura: 150, 151

Internet das Coisas: 193, 203, 206

L

Livros: 7, 100, 101, 113, 116

M

Metatranscriptoma: 177

Metodologia Ágil: 88

N

Narrativa: 74

Navegação Indoor: 202

O

Occupy's: 226, 227, 229, 232

Otimização Combinatória: 11

P

Pcodigo: 170

Programação: 71, 121, 128, 153, 164, 167, 175, 237

R

Recurso: 41, 150

S

Scrum: 88, 90, 91, 92

Semiótica: 7, 73, 74, 75, 82, 84, 86

Sistema: 7, 8, 40, 65, 100, 101, 106, 109, 116, 119, 121, 123, 163

Sistemas: 46, 60, 61, 63, 134, 141, 143, 144, 158, 237

Software: 22, 24, 30, 31, 40, 41, 42, 43, 61, 88,

89, 99, 117, 161, 175, 180, 204, 206, 237

T

Tecnologia Vestível: 8, 191

W

Web Semântica das Coisas: 202



follow us

www.editoracientifica.org contato@editoracientifica.org



ISBN 978-658719655-8

editora científica