

III ESCOLA REGIONAL DE INFORMÁTICA DE PERNAMBUCO GARANHUNS, PE 6-8 DE NOVEMBRO

Tema:

"Desafios da Computação: Academia-Indústria-Comunidade"

Anais

ISSN 2317-5346

Realização/Promoção:





Apoio:







© 2013 ERIPE

Anais da III Escola Regional de Informática de Pernambuco (2013)

Sociedade Brasileira de Computação (SBC)

Paulo Roberto Freire Cunha (UFPE/CIn)

Sociedade Brasileira de Computação - Secretaria Regional de Pernambuco

Sérgio Castelo Branco Soares (UFPE/CIn)

Organizador e Coordenador Geral da III Escola Regional de Informática de Pernambuco (2013)

Sérgio Francisco Tavares de Oliveira Mendonça (UFRPE/UAG)

Coordenação de Editoração/Diagramação

Sérgio Francisco Tavares de Oliveira Mendonça (UFRPE/UAG)

Revisão de Texto

Wanessa Pereira Santos (UFRPE/UAG)

Capa/Editoração/Diagramação

Henrique Franklin Brazil da Silva (UFRPE/UAG) João Ferreira da Silva Júnior (UFRPE/UAG)

Elaboração da Ficha Catalográfica

Gracineide Santos da Silva (UFRPE/UAG)

E74a Escola Regional de Informática de Pernambuco Garanhuns, PE:

Desafios da Computação: academia-indústria-comunidade (3.:

2013: Garanhuns, PE).

Anais/ III Escola Regional de Informática de Pernambuco Garanhuns, PE: Desafios da Computação: academia-indústria-comunidade, 6-8 nov. em Garanhuns, PE._Garanhuns, PE: UFRPE, 2013.

219p.

ISSN: 2317-5346

1. Computação. 2. Informática. 3. Ciência da Computação. I. Mendonça, Sérgio Francisco Tavares de Oliveira (org.). II. Título.

CDU:004

Universidade Federal Rural de Pernambuco Unidade Acadêmica de Garanhuns Av. Bom Pastor, s/n – Boa Vista 55292-270 – Garanhuns – PE Tel.:(87) 3764-5500



Comissão Organizadora

Organização e Coordenação Geral

Prof. Sérgio Francisco Tavares de Oliveira Mendonça

Coordenação Local

Prof. Gersonilo Oliveira da Silva

Coordenação do Comitê Científico

Prof. Rodrigo Gusmão de Carvalho Rocha

Supervisão

Prof. Kádna Maria Alves Camboim

Coordenação de Tecnologia

Prof. Jean Carlos Teixeira de Araújo

Coordenação de Editoração/Diagramação

João Ferreira da Silva Júnior

Monitores-Estudantes

Anderson da Silva Santos

Anderson Melo de Morais

Diego das Neves Nascimento

Élissa Wanessa Oliveira Vanderlei

Felipe Dias de Oliveira

Graciele Silva de Oliveira

Henrique Franklin Brazil da Silva

Isabelle Azevedo Ferreira

João Ferreira da Silva Júnior

Jorge Veloso dos Santos

José Diego Tenório

Juan Augusto Victor Oliveira Albuquerque

Lara Thayanne Gonçalves de Melo Bezerra

Lucas Felipe da Silva Paiva

Luciano Serafim de Souza

Luma Cardoso Ferro de Almeida

Marcus Vinícius Ferreira Teixeira

Maria Beatrice Gueiros Silva

Marrone Silvério Melo Dantas

Natanael Rêgo Soares

Thiago Henrique Braz de Oliveira

Vinícius Ferreira de Souza

Yasmin Lúcio Martins



Entidades Envolvidas

Realização

Universidade Federal Rural de Pernambuco (UFRPE) Unidade Acadêmica de Garanhuns (UAG) Bacharelado em Ciência da Computação

Promoção

Sociedade Brasileira de Computação (SBC)

Apoio

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

Universidade de Pernambuco (UPE) Campus Garanhuns



Comitê Científico

Adauto Trigueiro de Almeida Filho (UPE/Campus Garanhuns - Garanhuns/PE)

Assuero Fonseca Ximenes (UFRPE/UAG - Garanhuns/PE)

Bruno Santana da Silva (UFRN - Natal/RN)

Catarina de Souza Costa (UFAC - Rio Branco/AC)

Érica Teixeira Gomes de Sousa (UFRPE/Sede - Recife/PE)

Fernando Kenji Kamei (IFPE - Recife/PE)

Hilário Tomaz Alves de Oliveira (Capital Login - Recife/PE)

Gabriel de França Pereira e Silva (UFRPE/UAG - Garanhuns/PE)

Jean Carlos Teixeira de Araújo (UFRPE/UAG - Garanhuns/PE)

João Emanoel Ambrósio Gomes (FAPCE - Juazeiro do Norte/CE)

João Paulo Fechine Sette (UNIPÊ - João Pessoa/PB)

José Jair Cavalcante de Figueirêdo (UFAC - Rio Branco/AC)

Kádna Maria Alves Camboim (UFRPE/UAG - Garanhuns/PE)

Luis Filipe Alves Pereira (UFRPE/UAG - Garanhuns/PE)

Marcos Pinheiro Duarte (FAPCE - Juazeiro do Norte /CE)

Maria Aparecida Amorim Sibaldo de Carvalho (UFRPE/UAG - Garanhuns/PE)

Priscilla Kelly Machado Vieira (UFRPE/UAG - Garanhuns/PE)

Renê Nóbrega de Sousa Gadelha (UNINASSAU - Recife/PE)

Rodrigo Lins Rodrigues (UFRPE/Sede - Recife/PE)

Rodrigo Gusmão de Carvalho Rocha (UFRPE/UAG - Garanhuns/PE)

Ryan Ribeiro de Azevedo (UFRPE/UAG - Garanhuns/PE)

Sérgio Francisco Tavares de Oliveira Mendonça (UFRPE/UAG - Garanhuns/PE)

Tiago Buarque Assunção de Carvalho (UFRPE/UAG - Garanhuns/PE)



Sumário

Prefácio	8
Artigos Completos	9
LCCAReader: Software Proposal for Ubiquitous Applications Through RFID Sy	ystems
Um estudo sobre as dificuldades no processo de aprendizagem de programaç Curso de Análise e Desenvolvimento de Sistemas na FAFICA – Faculdade de File Ciências e Letras de Caruaru-PE	ção no osofia,
Colaboração organizacional através de <i>cloud computing</i> : um estudo de caso no Digital – PE	
Avaliação de Desempenho de Simuladores de Redes Ad Hoc	37
Um Sistema de Recomendação Baseado em Nuvem	46
Investigação de uma Arquitetura para Gerenciamento de Documentos	54
Avaliação de Ontologias de Domínio para o Desenvolvimento Distribuído de So	
Análise da Gerência de Configuração de uma Empresa de TI com Base no ITIL	
NVDA: Avaliação de Usabilidade da interação de usuários com deficiência visua a Web	
TVD.Calc: Um Aplicativo Dirigido à Educação Econômica em Ambientes Telev	
Malware em JAVA para Acessar Múltiplos Sistemas de Arquivos de Comput. Remotamente	
Resumos Expandidos	104
Uma análise comparativa entre protocolos de comunicação autonômicos basead gossip	
DinnerRace: Um simulador para técnicas de exclusão mútua através do proble. Jantar dos Filósofos	
Desenvolvimento de um ambiente colaborativo para ensino de geometria usa Facebook	
Simulação: Pseudoaleatoriedade, um estudo sobre o método do meio do quadrado	123
Inserção da mulher na Ciência da Computação no município de Garanhuns	128
Fomentando a Tecnologia da Informação e Comunicação no Agreste Meridional a de empresa júnior	
Olimpíadas de Matemática	136
Uma métrica para recomendação de federação em rede ponto-a-ponto	140

III ESCOLA REGIONAL DE INFORMÁTICA DE PERNAMBUCO 6-8 DE NOVEMBRO DE 2013, GARANHUNS, PE, BRASIL



Análise e Modelagem de Algoritmos para Eleição de Líder em Sistemas Distri	
Tutoriais	
Lógica de Programação: Iniciação Lúdica com App Inventor for Android	155
Startups Dirigidas à Inovação de Software: Da Universidade ao Mercado	162
SHARPE Software Package: Modelos Markovianos para Simulação de Sistemas .	170
Robótica Educacional: Uma Possibilidade para o Ensino e Aprendizagem	178
Desenvolvendo Aplicações com Java Card	185
Utilizando simulação para planejar a implantação de Redes Ad Hoc	194
Introdução a Modelagem 3D com Blender 3D	202
AppDay ERIPE 2013	210



Prefácio

Estes Anais contêm os artigos completos, resumos expandidos e tutoriais aceitos para apresentação na III Escola Regional de Informática de Pernambuco (ERIPE) 2013, realizado na Universidade Federal Rural de Pernambuco (UFRPE) — Unidade Acadêmica de Garanhuns (UAG), durante os dias 6 a 8 de novembro de 2013.

A ERIPE 2013 teve como objetivo geral possibilitar aos participantes um contato com tópicos que estão na vanguarda da pesquisa na área de Informática, Computação e Tecnologia da Informação, proporcionando um ambiente para a atualização técnico-científica, divulgação de atividades de pesquisa entre alunos, professores e profissionais destas áreas. Além disso, a ERIPE visa promover um maior intercâmbio entre as instituições de ensino médio, técnico e superior do Estado de Pernambuco que tenham cursos na área de Informática e afins.

O tema da 3ª edição foi "*Desafios da Computação*: *Academia-Indústria-Comunidade*". Deste modo a Academia dialogou com a Indústria a fim de compartilhar e, até mesmo, aprimorar conhecimentos na área da Computação e apresentar os seus resultados à Comunidade em geral.

O Comitê Científico da III ERIPE foi formado por um grupo multidisciplinar de especialistas relacionados à Computação, constituído por 23 pesquisadores, de diferentes áreas, instituições e empresas do país, aos quais foi atribuída a missão de avaliar os trabalhos submetidos para cada uma das áreas de interesse relacionadas ao evento.

Na III ERIPE foram submetidos cerca de 55 trabalhos, em forma de artigos completos, resumos expandidos e tutoriais, sobre pesquisas que tem sido desenvolvidas na academia e na indústria. Destes, 28 foram aceitos para publicação, sendo 11 artigos completos, 9 resumos expandidos e 8 tutoriais. A publicação dos anais é disponibilizada em formato eletrônico (*Internet*), com o número de ISSN 2317-5346.

Agradecemos a todos que, direta ou indiretamente, colaboraram positivamente para a realização do III ERIPE (alunos envolvidos na comissão organizadora, palestrantes, minicursistas, autores, comitê científico, patrocinadores, instituições, familiares e participantes). Ao todo 213 estudantes, professores, pesquisadores, representantes da indústria e comunidade participaram da nossa III Escola Regional de Informática de Pernambuco (2013). Fizemos nosso melhor para oferecer um interessante encontro, bem como para proporcionar aos participantes ótimos dias durante a estadia em Garanhuns, PE.

Garanhuns, 8 de novembro de 2013

Prof. Sérgio Tavares de Oliveira Mendonça Coordenação da III ERIPE



III ESCOLA REGIONAL DE INFORMÁTICA DE PERNAMBUCO GARANHUNS, PE 6-8 DE NOVEMBRO

Tema:

"Desafios da Computação: Academia-Indústria-Comunidade"

Artigos Completos

ISSN 2317-5346

Realização/Promoção:

UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

UNIDADE
ACADÊMICA
DE GARANHUNS



Apoio:







LCCAReader: Software Proposal for Ubiquitous Applications Through RFID Systems

Samuel Souto¹, Emanoel Costa¹, David Beserra¹, Ramon Nascimento¹, Alberto Araujo¹

¹Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil

Abstract. The use of RFID technology for identification has grown in many parts of the world. Its applications cover different economic areas ranging from student cards, inventory management until control robotic systems. But the difficulty in finding software commonly used to control RFID systems has also increasing. In this paper we present the development stages of software able to communicate with UHF RFID's systems offering functions able to remedy difficulties encountered in the literature and commercial applications. The results will be used in developing applications on identification and location of objects.

1. Introduction

Pervasive computing can be defined as the computing branch who is present in an environment, interacts with the ambience as if it was part of it, but it is not perceived when used [Santini, 2008]. Intertwined with this context, arise the Ubiquitous computing (UbiComp), defined as the integration of mobility with distributed systems mostly, intelligent and highly integrated with computers and their applications for the benefit of final users, becoming invisible for those them. [Weiser, 1991] [Satyanarayanan, 2001].

One of the technologies that are strongly related to these concepts are those based on radio frequency communication. RFID stands for Radio Frequency Identification. Basically it uses radio waves to perform the communication between a moving object, containing labels, commonly called *Tags*, and a reader [Santini, 2008]. RFID have been gaining ground on the world stage as identifiers that use radio waves to collect, modify and transmit information ensuring a rapid flow of data in a short period of time.

RFID readers are classified according to their frequency band of operation. Its application will have relationship with that characteristic. Thus, there are LF readers (Low Frequency readers) which have a frequency range of 30-300 kHz and are commonly used for identifying animals and close reading of items with high water content. HF (High Frequency readers), operating in a frequency range of 3-30 MHz and used for access control to buildings. UHF (Ultra High Frequency readers) operating in the range of 300MHz-3GHz and used to identify boxes and cases. Finally, Microwave



having a frequency range of 3GHz and used for vehicle identification [Glover &. Bhatt. 2007].

One of the most important components in an RFID system is its software. This component is responsible for managing the information trafficked by the duty cycle of RFID. Such software should work as a middleware. That is, provide a communication interface between the reader and user with maximum possible transparency. The interface of an interactive system comprises the entire portion of the software in which the user holds conceptual or physical contact during its use [Barbosa and Silva, 2010].

In theory, the user doesn"t need to possess a working knowledge about the RFID system components, but does need to know how to operate the software.

However it is not difficult to find testimonials from people talking about how hard is to work with software responsible of hardware management, such as done with RFID systems. This difficulty is caused by several factors. Either by the deployment, having as main objective, specific functions of the equipment used, either by excessive restrictions on the functionality of the system. Develop a software interface with technical languages hinders the software dissemination capable of hardware controlling systems between users with little or, even, none specific knowledge for this computing branch.

The main goal of this paper is to present the stages of development software called LCCAReader. Such system is capable of performing communication with radio frequency identification readers and able to remedy difficulties encountered in the literature and commercial applications where RFID plays an essential role in the management and data transferring.

2. Related Works

In the literature we can find some article involving the development of software for use in hardware systems, such as RFID, and applications that are similar to the proposed paper. Trindade & Sobrinho (2008) presented the stages of software deployment able to control tires. The authors also show its application. In the research it was proposed to use the system for assets control by the carriers. Trying to avoid malicious exchange of new tires for old and worn. The operation of this system is achieved by the *Tag* identification located on the tires of trucks and monitoring the status of them through a database.

Many studies have been conducted in order to seek innovative and creative applications for RFID. An example is the intelligent use of RFID as done by Luimula *et al.*, (2010). This research was conducted in three different stages and was used RFID to control robots through identification of ways that the robotic system should make, only using the computer and the RFID system to control the machine. The survey showed all stages of development an RFID system, from conception (literature review), deployment of a robotic control system, until the effective control of the robotic platform. The results showed that these systems can be used to control robots efficiently.

To achieve the goal of developing management software for hardware, such as the purpose of this paper, RFID equipment analysis involved in its deployment is required. In literature we can find some work involving this topic. Ramakrishnan (2003) verified



the performance of *Tags* identified by passive UHF readers having as protocol identification EPC (Electronic Product Code), in different environments. Including measurement of maximum reading distance, orientation sensitivity, read rate and performance variation. In the results, Ramakrishnan concluded that the system, used by him, recorded a reading beam on average 18 meters.

All studies previously reported have shown results that make possible the development of software whose purpose is to cater to a context or specific application. Even though the studies in the literature showing us good results about the integration between hardware and software, the difficulty relative to location of researches involving the development of ubiquitous computing systems, more specifically talking about RFID systems, enabling its use for other needs or welcome other realities is large and stumble into obstacles that slow your progress. Therefore, such questioning was used as motivating agent for improvement of found studies and development new research related to these topics.

3. LCCAReader: Design and Development.

Most of the computing area and, in particular, the sub-area of software engineering, is interested in building interactive systems more efficient, robust, error-free and easy maintenance [Barbosa & Silva, 2010].

The LCCAReader, name of the proposed software, is a system that has as main feature the communication between personal computers (PC) and UHF RFID devices, manufactured by Think Magic Company. This software has to perform this information exchange using the RS232 port of the computer. This is because the reader that has greater identification range, used in this study, identifying the world's most used radio frequency's communication protocol (EPC Global), is the Mercury 5e, manufactured by this company. The system also uses an API that provides communication with the serial port described above and consequently the reader desired.

Observed the key attributes required by other RFID systems, the software should have the following features: restricting access to the system, data and objects registration, user registration, RFID readers communication (produced by Think Magic) and reading/writing *Tags*. The description of each function is given below:

- User Registration: Registration of all users who have access to the system. During the registration, will be asked some information, such as password and login name
- **Restricting Access to the System:** Being an extension of the functionality "User Registration", the user will need to supply username and his password to proceed with system access.
- **Data and Object Registration:** The LCCAReader will have, as one of its proposal, the function to record data and objects which is related to a *Tag* number. As the user performs the identification of these labels, the information contained in the database will be displayed on the screen for him.
- **RFID Readers Communication:** The proposed system has the capability to perform communication with the reader desired, provided that it is compatible with the software developed.



- **Reading/Writing** *Tags*: Besides the function of communicating with the reader, the system can identify *Tags* and submit information, if any. Another important feature
- is the writing *Tags*. If the *Tag* identified is for the type Active (able to store data about an object within it), the system will provide an interface to perform data update contained within this *Tag*.
- **Graphic Interface:** Interface in which the user will requirehis actions to the system.
- **Database Communication:** The proposed software will have access to the database in which user's data, objects' data and *Tags*' data will be stored and available.

We used, as process of software development, the method Incremental Development. In this method the system is divided and developed in parts (called increments). Immediately after the specifying software requirements for each increment, the architecture of the system is designed, that is, their behavior is specified such as call model function, distribution of package, among others. Following the flow model for structuring software, we have the stage of development and validation. All increments are validated to be integrated with the rest of the system. If there is any error in the validation of the system altogether, the whole process is redirected to the stage of development. This cycle continues until all the software meets the initial system requirements.

In order to improve the understanding of the entire system, we developed some UML diagrams (Unified Modeling Language). UML is a set of notations that are intended to support the modeling of object-oriented systems, providing a partial representation of the system. We develop two diagrams that enabled a greater understanding of the LCCAReader's functioning. They are Use Case diagram and Sequence diagram.

The sequence diagram (Figure 1) represents the sequence of the system's global behavior detailing all the steps that a particular user needs to run to access or require any function. In the figure, we can see that it's necessary present information to the system in order to validate this access for a given user.

The Use Case diagram (Figure 2) shows the functions that can be controlled and managed by the user. We can quote as an example, the role of "Managing Tags". This functionality according to figure 2, comprise remove, insert, update, and retrieval data associated with the Tag numbers recognized by the system.



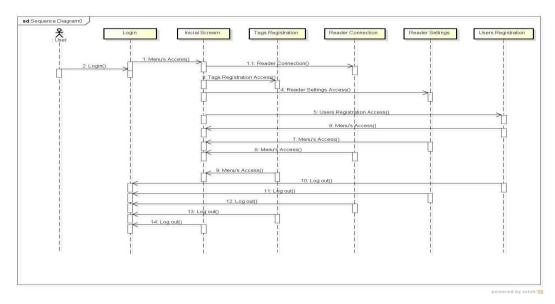


Figure 1. Sequence Diagram

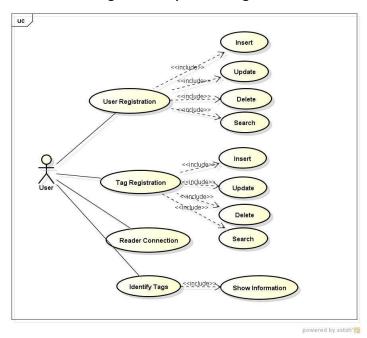


Figure 2. Use Case Diagram

The LCCAReader was developed using the programming platform NetBeans IDE 7.0.1 and Java Development Kit (JDK) 1.7_01. The communication with the reader Mercury 5e was based on a Think Magic's library equipment. The serial communication functions were implemented using the C programming language, We decided to do that because of the fact that this language is more flexible for information exchange with the RS-232 port than the Java language.

The development was divided into four increments representing all four main system functionality: user registration, Tag registration, RFID Readers communication and read/write Tags. All increments were developed in parallel, properly validated and



finally incorporated into the system source code. For a better understanding, we divided the software development in four primary steps described below.

- Serial Communication: This phase was focused on develop an algorithm capable of performing serial communication with external devices. The algorithm execute the following steps: first it's looked for a directory with all the DLLs (functions used by programs to perform tasks) necessary to communicate with the RS-232 port. This is required so the software can continue with its execution. Soon after this check, the code makes a call to system native methods (functions implemented using C language) and provides the information exchange through the serial port. If this process is not, by some reason, running, failures in connection with the port may be one of the reasons for this error, the algorithm returns a message stating the problems and possible causes of the error
- **RFID Reader Connection:** Main function of the LCCAReader software, the algorithm of the connection with the reader has the following operating cycle: a Java class named 'Reader' is called and instantiated. In order for this to occur, it's necessary the instantiation of one specific method named "create" (contained within this class). This function requires a parameter to execute its role; this variable is the URI, port address where the reader is located.
- Thereupon, there's a call for the method 'conect' (also contained within the class 'Reader'). This new incoming call is performed so that the software can achieve the communication with the reader. If the answer is positive the algorithm makes another call for method, this time for one named 'paramSet'. Such function required as parameter, the region to be identified. Think Magic's manufacturer readers claim a specific region ID to perform the identification. After these processes, the reader should be connected successfully. If there is any error during the process, the execution is stopped and an error message appears.
- Reading/Writing Tags: The reading/writing Tags algorithm, subdivision of the connection code with the reader showed above, was developed in this increment. After making sure that the reader is connected, this being the first step of the algorithm, the method 'read' (implemented in class 'Reader' discussed earlier), is called passing, as parameter, the reading time. If the reader is not connected an error message is sent to the user alerting him about event. For the accomplishment of writing, the process is the same. As the information that will be displayed on the screen include data contained within the Tags, if the user wants to change them, just edit the desired field and the algorithm makes sure that the changes will be saved.
- **Graphic Interface:** The graphical interfaces developed for the software is presented through the figures 3, 4, 5 and 6. The figure 3 shows the screen corresponding to the function of *Tag* registration. Once the user accesses this screen you can navigate through sub-functions related to registration functions, such as insertion, update and search for *Tags*.

The LCCAReader allows the user to edit some connection information related to the reader. This function is illustrated through figure 4. In this function the user can update the name of the reader as well as specifying what type of *Tags* the system can



identify. The main function of the software is shown in Figure 5: reader connection. At first, this frame will show the possibility of running a function: communicate with some reader. At the time that this communication is established, other functions will be presented to the user such as reading *Tags*, information verification in the database, among others.

Finally, the figure 6 shows the Unit Tests for registration functions present in the software. Through these tests it was concluded that the system meets all the functions satisfactorily, since, during the execution of the evaluation, the database was being monitored, as well as the execution of each command.





Figure 3. Tag registration screen



Figure 4. Setting

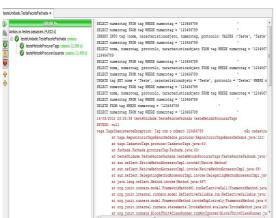


Figure 5. RFID Reader communication screen

Figure 6. Unit tests screen

4. Final Thoughts

In this work was presented all stages of development of software capable of performing the communication interface with an RFID system respecting usability requirements necessary to ensure communication between users and system. During the planning and requirements gathering, was considered others RFID software with specific applications and even software available by manufacturers of readers. This research allowed us to have clarity of running good software for this computing branch.



As development process, we adopt the incremental method. This decision was made based on the fact that the RFID hardware uses various artifice, in a way, independent of each other. We adopt the parallel development of software. In this type of development, the deployment of the increments is performed simultaneously. Therefore, the incremental method didn't delay the development process, on the contrary, has showed an improvement, since the validation of a test served to increase the system completely. The validation of a given increment, for example, the *Tags* registration has the same results when the system is fully tested. This feature is possible because the software includes modules for communication, in a way, independent of each other. What communicates all portions of the system is the access to the database. In this way we can advance the implementation schedule.

On the other hand, the system programming presented difficulties on its communication reader step. This occurred because the reader, chosen by us, has protocols for information exchanging through serial port of the computer and doesn't have knowing algorithms messaging. It was necessary to study the entire API used by this equipment, searching for knowledge of its operation. this problem identified took us the need to divide this increment in development units even smaller than we usually did with the aim of reducing the problem in minimum increments and find individual solutions for the errors identified. The LCCAReader presented a satisfactory response to events it has undergone. At the moment the system is in the testing phase and usability issues such as efficiency of algorithms should be remedied in future versions of the software.

The difficulties in the adoption of RFID systems exist, but they are not insurmountable, says Wu et al. (2005). Like most technologies increasing nowadays, is simply a matter of time until the promises of RFID becoming a real thing.

References

- Barbosa, S. D. J. And Silva, B. S. (2010). "Interação Humano-Computador". Rio de Janeiro: Editora campus.
- Glover, B., and Bhatt, H. (2007) "Fundamentos de RFID". Rio de Janeiro: Alta Books. Luimula, M., Sääskilahti, K., Partala, T., Pieskä, S., Alaspää, J. (2010) "Remote navegation of a mobile robot in an RFID-augmented environment". Pers. Ubiquit Comput.
- Ramakrishnan, K. N. M. (2003) "Benchmarks for Passive UHF RFID *Tags*". College of Engineering, Guindy Anna University, Chennai, Indiana.
- Santini, A. G. (2008) "RFID Radio Frequency Identification". Rio de Janeiro: Ciência Moderna.
- Satyanarayanan, M. (2001). "Pervasive Computing: Vision and Challenges", http://www.cs.cmu.edu/~aura/docdir/pcs01.pdf, July.
- Sonntag, N. L. Barbosa, D. N. F. Barbosa, J. L. V. Pinto, S. C. C. S. (2012). "Gerenciador de Objetos de Aprendizagem para um Ambiente de Educação
- Ubíqua". Biblioteca Digital Brasileira de Computação BDBComp. Avaliable in: http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=8654. March.



- ThinkMagic (2007) "Mercury4/5 User Guide: (TM-M4/5-X-INSTALL) Rev. 2.4.5", http://thinkmagic.com, July.
- Trindade, J. A., Sobrinho, E. (2008) "Sistema RFID: Controle de Pneus". Revista de Engenharia da Computação, v. 4, n. 1. 2008.
- Wang, S. W., Chen, W. H., Ong, C. S., Liu, L., Chuang, Y. W. (2006) "RFID application in hospitals: A case study on a demonstration RFID project in a Taiwan hospital" System Sciences, in HICSS '06. *Proceedings of the 39th Annual Hawaii International Conference. Volume 8.*
- Weiser, M. (1991) "The Computer for the 21st Century". Scientific American, pages 94-104.
- Wu, N. C. Nystrom, M. A. Lin, H. C. Y. "Challenges to global RFID adoption". (2005). Avaliable in: http://www.sciencedirect.com. March.



Um estudo sobre as dificuldades no processo de aprendizagem de programação no Curso de Análise e Desenvolvimento de Sistemas na FAFICA – Faculdade de Filosofia, Ciências e Letras de Caruaru-PE

Márcia Valéria Rocha de Souza¹, A. César C. França¹

¹Faculdade de Filosofia Ciências e Letras de Caruaru (FAFICA) R. Azevedo Coutinho – Petrópolis – Caruaru – PE - CEP 55030-240

{marciavr.souza,cesarfranca}@gmail.com

Resumo. Para formar bons profissionais, uma instituição de nível superior deve oferecer cursos de qualidade para os seus alunos incentivando-o de forma lúdica para o aprendizado. O presente artigo trata de um estudo realizado na Faculdade de Filosofia Ciências e Letra de Caruaru sobre as dificuldades dos alunos do curso de Análise e Desenvolvimento de Sistemas do primeiro ao quarto período no processo aprendizagem. Foram apontadas as principais dificuldades e a sugestão de algumas ferramentas de apoio ao ensino dos principais conceitos de programação.

Abstract. In order to develop high-level professionals, a higher education course should offer good quality courses for their students, encouraging them to learn. This article reports a study conducted at the Faculty of Philosophy, Sciences and Letter of Caruaru (FAFICA) on the difficulties of students of Analysis and Systems Development from the first to the fourth period in programming learning. The survey data pointed out the major difficulties faced by the students, and we conclude suggesting potential tools to support the teaching of the key programming concepts.

1. Introdução

Um bom profissional em programação deve possuir habilidades para codificar instruções executáveis por um computador. Portanto, durante os estágios iniciais de seu aprendizado, adquirir uma boa base dos conhecimentos básicos em linguagem de programação assim como o desenvolvimento em lógica facilita na construção de algoritmos executáveis.

O currículo de referência da Sociedade Brasileira de Computação (SBC) prevê que os cursos que tenham sistemas computacionais e suas aplicações como atividadefim, atividade-meio ou cursos de Licenciatura em Computação, deverão possuir em sua grade curricular a disciplina de linguagem de programação, sendo esta a que possui um acentuado grau de dificuldade por parte dos alunos [Mota et al. 2003].

O objetivo deste artigo é apontar as principais dificuldades dos alunos na aprendizagem de programação do curso de Análise e Desenvolvimento de Sistemas desta instituição. Espera-se que após a mensuração dos dados fiquem evidentes os pontos de dificuldade dos alunos podendo contribuir também para a tomada de decisões e ações de



professores e coordenadores para a melhoria do curso e consequentemente o aprendizado em sala de aula.

O artigo está organizado da seguinte forma: a Seção 2 apresenta a contextualização referente ao primeiro trabalho onde foi feito um estudo comparativo com ferramentas de auxílio ao aprendizado em programação; a Seção 3 trata-se da forma como foram coletados os dados para a pesquisa; na Seção 4 são apresentados os resultados da coleta dos dados e a discussão destes. Por fim, na Seção 5 estão as conclusões deste trabalho.

2. Contextualização

Em um trabalho realizado anteriormente foi apontado através da literatura, algumas dificuldades dos alunos iniciantes em entender a lógica de programação e desenvolver meios necessários para chegar à resolução de um problema. Elaborou-se uma pesquisa de algumas ferramentas disponíveis na internet, gratuitas, que podem auxiliar na compreensão. As ferramentas foram classificadas de acordo com os conceitos básicos como é apresentado na Tabela 1(vide Souza e França [2013]). Compararam-se as competências como sugere Gomes, Henriques e Mendes [2008] em um modelo construtivista de aprendizagem em três fases: na resolução de problemas diversos, reconhecimento da utilidade da programação e a capacidade de construção de algoritmos. Após esta etapa as ferramentas foram catalogadas em três níveis: básico, intermediário e avançado. A finalidade deste agrupamento é uma possível aplicação em sala de aula, para isso é necessário identificar quais principais dificuldades dos alunos do primeiro ao quarto período o qual se refere o presente artigo e, por fim, utilizar a ferramenta adequada aos problemas identificados.

Tabela 1. Conceitos trabalhados nas ferramentas avaliadas (vide Souza e França [2013])

	Compilação	Alocação de memória	Alocação vetorial	Tipos de dados	Operações Aritméticas	Operações Lógicas	Estr. de Controle de	Funções	Recursão	Depuração passo-a-passo
CodeMonster	Não	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	Não
FutCode	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	Nã o
Guido VanRobot	Não	Não	Não	Não	Nã o	SIM	SIM	SIM	Nã o	\mathbf{SIM}
KidsRuby	Nã o	SIM	Nã o	Nã o	SIM	SIM	SIM	Não	Nã o	Nã o
RobotProg	Não	SIM	Não	Não	SIM	SIM	SIM	SIM	SIM	SIM
TBC-AED	Não	SIM	SIM	SIM	SIM	SIM	SIM	SIM	Nã o	SIM
Visual Alg	Nã o	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM	SIM
Web Portugol	Não	SIM	SIM	Não	SIM	SIM	SIM	Não	Não	SIM

3. Método

Com o objetivo de analisar as dificuldades dos estudantes na aprendizagem em programação, no fim do primeiro semestre acadêmico de 2013, foi realizada uma pesquisa com os alunos do primeiro ao quarto período do curso de Análise e Desenvolvimento de Sistemas desta instituição. Aplicou-se um questionário presencial constituído de três partes: (i) dados demográficos, (ii) estudo de programação e (iii) auto avaliação. A primeira parte consiste em coletar informações referentes aos dados



populacionais. A segunda parte visa avaliar o interesse do aluno em relação aos estudos e o conhecimento das principais dificuldades no aprendizado em programação. A terceira parte do questionário propõe uma autoavaliação do aluno sobre alguns conceitos iniciais em linguagem de programação como compilação, alocação de variáveis e vetores, tipo abstrato de dados, operações de aritmética e lógica, estrutura de controle de luxo, funções e métodos, ponteiros e referências, recursão e depuração.

A coleta dos dados foi realizada entre os dias 06 e 10 do mês de maio. Foram aplicados, respectivamente, questionários para duas turmas do 1° e do 3° Período e uma turma do 2° e 4° Período. O número de matriculados nas disciplinas Programação I, Programação II, Programação Orientada a Objetos II e Programação Web, na devida ordem, nos 1°, 2°, 3° e 4° períodos foram de 198 (cento e noventa e oito) alunos, porém na aplicação do questionário, devido à ausência de uns, contabilizou-se 137 (cento e trinta e sete) respondidos totalizando, aproximadamente, 70% do total de alunos nestes períodos letivos.

4. Resultados e Discussão

O curso oferecido pela instituição pesquisada é de nível tecnólogo e oferece ao aluno o contato com linguagens de programação de alto nível largamente utilizadas no mercado profissional. Dado o foco da instituição em preparar rapidamente os alunos para o mercado de trabalho, as disciplinas de programação focam no aprendizado da linguagem Java® e no paradigma orientado a objetos.

Inicialmente observou-se uma redução gradual no número de alunos por período letivo. Gomes, Henriques e Mendes [2008] comentam que o desinteresse pode acontecer por dificuldades de interpretação ou devido os alunos se sentirem demasiados ansiosos para começar a codificar sem antes compreender os dados do problema, dificultando a fase seguinte, a construção do algoritmo. A dificuldade de empregar o raciocínio lógico, afirmam Barros, Delgado e Machion [2004], também gera um ambiente desmotivante, justificando esta redução de alunos através de uma possível evasão ou reprovação na disciplina.

Dificuldades pedagógicas e metodológicas

Jenkins [2002] cita as várias causas do insucesso generalizado em disciplinas de programação, como: o baixo nível de abstração, a falta de competências de resolução de problemas, a inadequação dos métodos pedagógicos aos estilos de aprendizagem dos alunos, entre outros; e afirma que as linguagens de programação possuem sintaxes adequadas para profissionais, mas não para aprendizes inexperientes.

Dentre os entrevistados, apenas 7 (sete) estudantes são atuantes no mercado profissional, e apenas quatro destes estão acima dos 26 anos. Entre as principais dificuldades apontadas deste grupo estão o método de ensino, tempo para dedicar aos estudos e apenas um deles registrou dificuldades no aprendizado com relação à insuficiência de material e compreensão de linguagens de baixo nível. Pouco mais de 80% dos alunos com até 18 anos estão no 1° período, e as principais dificuldades estão relacionadas com o aprendizado da lógica de programação seguido da metodologia de ensino dos professores. A lógica de programação e a metodologia aplicada pelos professores também estão entre os principais problemas citados pelos alunos da faixa



etária dos 19 aos 25 anos, seguido do tempo para a dedicação aos estudos e o conhecimento na língua inglesa. Souza e França [2013] comentam que as dificuldades podem estar associadas tanto à forma de ensino, como também, na familiaridade prévia com noções básicas de criação de algoritmos.

Familiaridade prévia e disponibilidade de tempo para dedicação aos estudos

Com o propósito de identificar a familiaridade dos alunos com a programação antes da graduação, foi questionada com qual idade e qual foi a primeira linguagem de contato. Aproximadamente 44% dos estudantes responderam que tiveram seu primeiro contato com programação dos 8 aos 17 anos e 81 alunos tiveram como primeira linguagem o Java, seguido da linguagem C. Em relação aos alunos que já trabalham na área e o tempo de dedicação por semana, 4 praticam de 1 a 3h, dois praticam de 7 a 9h e apenas um dedica 10h ou mais de práticas em programação. Dos 137 alunos que responderam a pesquisa, 85 deles praticam programação entre zero a três horas por semana e um grupo de 52 alunos praticam mais de 4 horas. De acordo com o ranking do site Tiobe [2013] a linguagem Java aparece em 2° lugar por dois anos consecutivos entre as mais utilizadas no mercado.

A compreensão de outras linguagens já serve como um diferencial no mercado de trabalho. Dentre elas, as mais citadas são o HTML/CSS, o C/C++ e o Javascript. Das linguagens que receberam menos de dez votos estão o Ruby, ActionScript, VBNet e o Cobol/Fortran, Objective C e o Prolog. As linguagens Perl e Lisp/Haskel não receberam nenhum voto. Entre as citadas pelos alunos que não continham na lista do questionário estão o Portugol, ShellScript, Visualg, ADVPL, Progress 4gl, Basic, Natural Clipper, Lua, Assembly e Grails.

Lógica de programação e metodologia de ensino

A literatura concorda que a metodologia de ensino é um desafio relevante para o ensino básico de lógica de programação. Segundo Gomes, Henriques e Mendes [2008], a um nível mais básico, o ensino das linguagens de programação tem como propósito permitir que os alunos desenvolvam as suas capacidades, adquirindo os conhecimentos básicos necessários para conceber programas capazes de resolver problemas reais simples. Sendo assim, somente após o aprendizado destes conceitos, o estudante pode travar contato com uma linguagem de programação concreta [Santos e Costa 2006]. Almeida, Castro e Castro [2006] afirmam que um dos principais componentes na aprendizagem de programação é a organização das habilidades para a resolução de problemas e como tais habilidades são construídas, envolve identificar as dificuldades encontradas pelos alunos. Porém, em uma sala de aula, o nível de conhecimento e assimilação dos conteúdos passados é diferente de aluno para aluno o que sugere um ensino individualizado.

No entanto, Falckembach e Araújo [2005], discutem que esse tipo de ensino, personalizado e individualizado, entra em choque com o pressuposto básico da educação tradicional que é a padronização. Segundo Falckembach e Araújo [2005] a forma de ensino dentro da sala de aula é a mesma para todos os alunos, pois, é extremamente difícil para um professor levar em consideração o perfil, as metas, as necessidades e o nível de conhecimento de cada aluno, de modo a proporcionar a cada um, um ensino adaptado.



Disponibilidade de material de apoio ao aprendizado

A biblioteca da instituição possui um acervo vasto para o estudo em linguagens de programação e é eminente a amplitude de materiais relativos ao tema encontrados na internet que se tornou um grande aliado ao aprendizado. Por esse motivo, dificuldades registradas pelos alunos relativas à falta de material para estudo não são necessariamente justificáveis. Por outro lado, estes dados podem revelar uma ineficácia do material disponível às necessidades de aprendizado dos alunos. As ferramentas apresentadas na Tabela 1, por exemplo, oferecem estratégias interativas que se propõem a ser mais eficientes do ponto de vista da velocidade e da consistência do aprendizado. No entanto, não encontramos dados disponíveis que possam reforçar esta proposta.

Dificuldades relacionadas à complexidade dos conceitos de programação

Na terceira etapa do questionário, auto avaliação, os estudantes responderam sobre o grau de conhecimento em noções básicas de programação. As análises dos dados foram feitas por período já que alguns dos conceitos ainda não foram passados para alunos do primeiro período como Vetores, Tipo Abstrato de Dados, Funções e Métodos, Ponteiros, Referências e Recursão. Estão apresentados, neste tópico, apenas os conceitos que foram registrados foram do domínio dos alunos.

A ideia de abstração de dados refere-se a uma modelagem de uma estrutura de dados de acordo com seu funcionamento. No segundo período, 68,42% dos estudantes conhecem o conceito, mas desconhece a forma de aplicação. Menos da metade dos alunos do terceiro período dominam o conceito. No quarto período houve um empate entre os que também sabem do conceito, mas não sabem da aplicação e dos que dominam o assunto.

Tabela 2. Porcentagem dos estudantes por período letivo que dominam os conceitos citados

Período	N° Alunos	Compilação	Alocação de variáveis	Alocação vetorial	Tipos dados abstratos	Operações Aritméticas	Operações Lógicas	Estr. Controle de	Funções e Métodos	Ponteiros e Referências	Recursão	Depuração
1 °	62	54,8%	45,1%	1,6%	6,4%	66,1%	30,6%	35,4%	9,6%	6,4%	1,6%	20,9%
2 °	19	84,2%	94,7%	78,9%	15,7%	84,2%	47,3%	100%	52,6%	21,5%	5,2%	10,5%
3 °	28	71,4%	82,1%	60,7%	46,4%	96,4%	67,8%	92,8%	64,2%	25%	10,7%	35,7%
4 °	28	82,1%	78,5%	46,4%	39,2%	100%	78,5%	85,7%	71,4%	32,1%	21,4%	39,2%

Ponteiros, Referências, Recursão e Depuração são os conceitos de mais dificuldades entre os alunos. A depuração, apesar de ter sido discutida em sala de aula no primeiro período, 38,70% dos alunos nunca ouviu falar. No segundo período, a maioria, 47,36%, sabe do que se trata, porém não sabem usar. No terceiro período, houve um empate entre os que conhecem, mas não sabem da funcionalidade e os que dominam o conceito. O mais alarmante é o conceito de recursão, a grande maioria em todos os períodos nunca ouviram falar. O conceito de ponteiros e referências, do segundo ao quarto período, mais de 50% respondeu que sabem do que se trata, mas não sabem usar.



Problemas no aprendizado de conceitos de programação x Ferramentas de apoio

Na última parte do questionário os alunos avaliam sua compreensão dos conceitos citados em cinco opções, os extremos correspondem ou a total abstração ou o domínio. O domínio seria a compreensão e os demais apontam um déficit a ser resolvido. Analisando as dificuldades dos alunos com a referenciada Tabela 1, podemos identificar quais das ferramentas, estudadas anteriormente, poderão ser aplicadas. O primeiro dos conceitos trabalhados segundo as competências é o de compilação, apesar de mais de 50% da turma do primeiro período dominar o conceito, uma parcela considerável de 28 alunos sente dificuldades entre a usabilidade e a utilidade na aplicação. Das ferramentas citadas na Tabela 1, apenas o FutCode trabalha este conceito, porém como no FutCode também é aplicado conceitos ainda não estudados pelas turmas do primeiro período, convém analisar uma forma desta ferramenta contribuir na aprendizagem.

Menos da metade dos alunos dominam o conceito de alocação de varáveis no primeiro período, 34 dos 62 alunos responderam que possuem algumas dificuldades neste conceito. Todas as ferramentas citadas na Tabela 1, com exceção do Guido VanRobot, trabalham o conceito de alocação de variáveis, porém o TBC-AED trabalha conceitos mais avançados e não possuem uma versão para iniciantes, o CodeMonster e o Kids Ruby utilizam as linguagens javascript e Ruby, respectivamente, o que seria desaconselhável para alunos do primeiro período que ainda não tiveram nenhum contato com linguagens de programação em sala de aula. As demais turmas apresentaram um nível satisfatório de compreensão do conceito de alocação de variáveis.

Nos terceiro e quarto período, onze e quinze alunos, respectivamente, dos 28 entrevistados em cada turma, responderam que sentem alguma dificuldade em alocação vetorial. Entre as ferramentas que podem ser aplicadas para este conceito segundo a Tabela 1, estão: CodeMonster, FutCode, TBC-AED, VisuAlg e Web Portugol. No conceito de tipo abstrato de dados, do segundo ao quarto período, mais da metade dos que responderam o questionário afirmaram que sentem algum bloqueio. As ferramentas que podem ser indicadas para servir como apoio são: CodeMonster, FutCode, TBCAED e VisuAlg.

Sobre o conceito de operações aritméticas, há uma distribuição ínfima do segundo ao quarto período, de alunos com bloqueio neste conceito. Já no primeiro período, 21 alunos dos 62 entrevistados possuem dificuldades. Em operações lógicas, mais da metade no primeiro e segundo períodos possuem dificuldades, no terceiro e quarto períodos somam 15 alunos. Já no conceito de controle de fluxo, o primeiro período tem mais dificuldades, 40 alunos dos 62 entrevistados. Nos terceiro e quarto períodos somam 06 alunos que sentem algum bloqueio. Todas as ferramentas citadas na Tabela 1 trabalham estes três conceitos, exceto o Guido VanRobot.

Os conceitos de funções, métodos, ponteiros, referências e recursão, são trabalhados inicialmente no segundo período, onde 9 dos 19 alunos apontaram dificuldades em funções e métodos. No terceiro e quarto períodos, respectivamente 10 e 8 sentem dificuldades. De acordo com a tabela 3 deste artigo, observa-se que os conceitos mais críticos são os de ponteiros, referências e recursão. O conceito de recursão chega a ser o mais grave onde mais de 50% de todas as turmas respondeu que nunca ouviu falar. Infelizmente a nossa tabela referenciada de um estudo anterior não avalia as ferramentas com o conceito de ponteiros e referências. As ferramentas



CodeMonster, FutCode, RobotProg e VisuAlg são as únicas que trabalham Funções, Métodos e Recursão.

A depuração é um processo de análise de código passo a passo e aplicável desde o primeiro período. Porém, mais de 50% em todas as turmas relataram algum tipo de dificuldade neste conceito. E das ferramentas que podem se aplicadas estão o Guido VanRobot, RobotProg, TBC-AED, VisuAlg e Web Portugol.

Segundo as competências trabalhadas sugeridas por Gomes, Henriques e Mendes [2008], aponta-se o FutCode como uma ferramenta educacional que possibilita o aluno reconhecer alguns dos principais conceitos em programação. Na Tabela 3, estão as análises das habilidades dos alunos, onde eles avaliaram suas competências em programação ficando notável a dificuldade dos alunos em resolver problemas reais. Sugerindo, então, o FutCode ou uma adaptação desta ferramenta para auxiliar na compreensão dos alunos.

3° 1° 2° **4**° Períodos (n=62)(n=19)(n=28)(n=28)Dificuldade na construção de algoritmos simples 14% 5,26% 7,14% 7,14% Constrói algoritmos mas não visualiza a utilidade 26% 26,31% 25% 32,14% Programa bem e visualiza a utilidade mas tem 50% 57,89% 50% 42,85% limitações para resolver problemas reais Programa bem e visualiza a utilidade e é capaz de 8% 10,52% 17,85% 17,85% resolver problemas reais

Tabela 3. Análise das Habilidades

No trabalho anterior, as ferramentas foram categorizadas em três níveis de complexidade. Iniciantes, onde são trabalhados os conceitos básicos como operações aritméticas, lógicas e estrutura de controle, focando a construção de algoritmos, podendo ser indicada para turmas iniciantes, como o primeiro período. Entre as ferramentas citadas para o nível iniciante estão o RobotProg e o Guido VanRobot. As ferramentas de nível intermediário dedicam-se aos conceitos de alocação de memória em variáveis e vetores, tipo abstrato de dados e funções. Com o propósito de trabalhar com linguagens de programação reais, passaria a transparecer para o aluno a utilidade da programação. Seriam as ferramentas de nível intermediário o CodeMonster e o KidsRuby. Para o nível avançado, a proposta é fazer com que os alunos visualizem a solução de problemas em domínios reais. O FutCode atende o requisito onde são trabalhados conceitos avançados, como compilação e recursão, este último com o pior índice registrado na instituição.

5. Conclusão

De acordo com os dados apresentados, constata-se que há um déficit no aprendizado dos alunos da Faculdade de Filosofia Ciências e Letras de Caruaru. A proposta desde artigo é contribuir com a instituição em oferecer para os seus alunos um curso satisfatório. A procura pela graduação em Análise e Desenvolvimento de Sistemas é grande, mas por se tratar do aprendizado de conceitos complexos, é significativo o número de alunos que desistem ao longo do curso, como pudemos observar nos números dos matriculados do primeiro ao quarto período. Outro agravante é como um aluno chega ao quarto período



sem saber dos conceitos básicos de programação como alocação de uma variável, por exemplo. Neste artigo, estão em evidência as principais dificuldades dos alunos e espera-se que com esses dados, coordenadores e professores possam tomar medidas para reduzir a problemática. A contribuição central deste estudo consiste na avaliação da adequação de meios alternativos de ensino de programação, evidenciando que estes podem trazer resultados positivos tanto para os alunos quanto para a instituição. Ressaltando a queixa da maioria dos entrevistados, é necessário que os docentes possam acompanhar os alunos no processo aprendizagem, identificar as dificuldades e dispor de estratégicas de ensino simples, como as ferramentas apresentadas, que de forma lúdica acabam contribuindo em sala de aula.

Embora este artigo relate um passo fundamental para a adoção de ferramentas de apoio ao aprendizado em cursos de programação, os dados sobre a influência de tais ferramentas sobre o real aprendizado dos alunos ainda são insipientes. Pesquisas futuras devem investigar a adoção prática de uma ou mais destas ferramentas em sala de aula, e avaliar se os benefícios propostos se confirmam ou não.

Referências

- Almeida, N. F. A., Castro, T., Castro, A. N. (2006). "Utilizando o Método Clínico Piagetiano para Acompanhar a Aprendizagem de Programação". http://www.lbd.dcc.ufmg.br/colecoes/sbie/2006/019.pdf, Junho 2013.
- Barros, L. N., Delgado, K. V., Machion, A. C. G. (2004). "An ITS for programming to explore practical reasoning". http://goo.gl/z4f50, Junho 2013.
- Deitel, H. M.; Deitel, P. J. (2010). "Java Como Programar". Ed. Pearson Prentice Hall, Junho 2013.
- Falckembach, G. A. M., Araújo, F. V. (2005). "Aprendizagem de algoritmos: dificuldades na resolução de problemas". http://www.fabricioviero.com.br/artigos/a4 siie.pdf, Junho 2013.
- Gomes, A., Henriques, J., Mendes, A. J. (2008). "Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores", http://goo.gl/DGLXh, Junho 2013.
- Jenkins, T. (2002). "On the difficulty of learning to program", http://www.ics.heacademy.ac.uk/Events/conf2002/tjenkins.pdf, Junho 2013.
- Mota, M. P., Brito, S. R., Moreira, M. P. and Favero, E. L. (2009) "Ambiente Integrado à Plataforma Moodle para Apoio ao Desenvolvimento das Habilidades Iniciais de Programação", http://goo.gl/dFHKS. Junho 2013.
- Santos, R. P.; Costa, H. A. X. (2006). "Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos Iniciantes em Computação e Informática." http://goo.gl/RcCjk, Junho 2013.
- SBC Sociedade Brasileira de Computação (2003). Currículo de Referência da Sociedade Brasileira de Computação para Cursos de Graduação em Computação e Informática". http://www.sbc.org.br. Junho 2013.
- Souza, M. V. R., França, A. C. C. (2013). "Ferramentas de Auxílio ao Aprendizado de

III ESCOLA REGIONAL DE INFORMÁTICA DE PERNAMBUCO 6-8 DE NOVEMBRO DE 2013, GARANHUNS, PE, BRASIL



Programação: Um Estudo Comparativo". http://www.erbase2013.itatechjr.com.br/index.php/programacao/weibase. Junho 2013.

Tiobe. (2013). "TIOBE Programming Community Index for july 2013". Disponível em http://www.tiobe.com/>. Acessado em Julho 2013.



Colaboração organizacional através de *cloud computing*: um estudo de caso no Porto Digital – PE

Lara T. G. M. Bezerra¹, Cristovão S. Brito¹, Patrícia T. Endo¹

¹Faculdade de Ciências e Tecnologia de Caruaru – Universidade de Pernambuco (UPE) Rodovia BR 104, Km 62 – Polo Caruaru – PE – Brasil

{upe.upeduca, patriciaendo}@gmail.com, cristovaos.brito@yahoo.com.br

Resumo. O uso de ferramentas que integram comunicações é evidente no mundo atual. Dentre essas ferramentas, cloud computing vem se destacando por possibilitar mobilidade no acesso e compartilhamento de informações. Diante disso, este artigo objetiva analisar a opinião dos gestores do Porto Digital-PE a respeito do uso de cloud computing nos processos colaborativos, bem como vantagens e desafios na utilização dessa tecnologia. Foi realizada uma pesquisa exploratória de conteúdo quantitativo. Diante dos resultados, este artigo dá ênfase maior ao entendimento dos gestores a respeito da utilização de cloud computing em ambientes colaborativos.

Abstract. The tools usage in order to integrate communication is evident in the actual world. Cloud computing has been highlighted due to the possibility of mobility access and information sharing. Because of that, this article analyzes the managers's opinions from Porto Digital – PE about the cloud computing usage in the colaborative processes, as well as the advantages and challenges in the use of these technologies. An exploratory quantitative research was done. Analyzing the results, this article emphasis the understanding of the utilization of cloud computing in collaborative environments.

1. Informações Gerais

De acordo com IBM (2008), o aumento na produtividade e a diminuição dos custos na realização de atividades estão associados à coordenação síncrona de informações dentro e fora das corporações através de sistemas de informação. No entanto, o uso de sistemas de integração demanda altos custos de implantação e, muitos deles, não são compatíveis na troca de dados.

Dessa forma, para reduzir custos de implantação de infraestrutura de TI, muitas alternativas vêm sendo elaboradas e continuamente postas em prática, como o uso de aplicativos sem custos de licença e o aluguel de hardwares. Entretanto, essas práticas ainda não solucionaram problemas como o fluxo descontínuo de informações e a má gestão de processos colaborativos.

Diante desse cenário, surgiu um novo modelo computacional denominado de cloud computing ou Computação em Nuvem. Para CAMEIRA et al. (2012), esse modelo poupará as instituições dos altos custos de implantação e manutenção de recursos computacionais, pois os serviços serão oferecidos sob demanda. Além disso, por ser baseado na Internet, esse modelo possibilita integrar fluxos de informações em tempo real a nível global.



Portanto, este artigo pretende analisar como os gestores do Porto Digital – PE estão utilizando *cloud computing* nas práticas colaborativas organizacional, bem como as vantagens e desafios inerentes a utilização dessa tecnologia.

Depois dessa introdução, na seção 2, é apresentada a importância da colaboração organizacional. Na seção 3, são descritos alguns conceitos básicos de *cloud computing*. Na seção 4, são apresentadas as principais vantagens e desafios na utilização de *cloud computing* nas práticas colaborativas das organizações. Na seção 5 e 6 são apresentados, respectivamente, a metodologia empregada para a construção desse trabalho e o motivo pelo qual o Porto Digital - PE foi escolhido como objeto de estudo. Na seção 8 é apresentada uma análise discursiva a respeito dos dados coletados e, por fim, na seção 9, são feitas as considerações finais e propostas de trabalhos futuros.

2. Organizações colaborativas

A importância da colaboração organizacional está relacionada à automatização de processos através de complexas redes de relacionamentos com parceiros de negócios para estimular a inovação e diferenciar-se no mercado cada vez mais competitivo. Com base nisso, CHRISTOPHER (2009) afirma que, o correto gerenciamento do fluxo de informação na cadeia de suprimentos (*supply chain* – SC) tem sido vista atualmente como ponto crítico das corporações.

Diante desse cenário, segundo IBM (2008), não é uma maneira produtiva as empresas se isolarem dentro de escritórios a procura de brilhantes ideias. Nos dias atuais, são a partir das práticas colaborativas que surgem as melhores ideias.

Ainda vale frisar que, com o processo de aquisição de ideias inovadoras a partir da colaboração, é possível obter vantagem competitiva pelo fato de se conseguir organizar o poder inovador do conhecimento compartilhado. Por isso, inúmeras tecnologias vêm surgindo para dar suporte a práticas colaborativas. Dentre essas tecnologias, *cloud computing* vem se destacando como solução para automatizar, por meio de baixos custos, práticas colaborativas.

3. Cloud computing

Para NIST (2012), *cloud computing* é um modelo computacional que permite de maneira onipresente e conveniente o acesso a dados e informações disponíveis na rede. NIST (2012) também afirma que, com esse modelo é possível compartilhar recursos computacionais de forma rápida, podendo ser provisionados e liberados com o mínimo de esforço de gerenciamento e interação com os provedores dos serviços prestados.

Para *CLOUD SECURITY ALLIANCE* (2012), os principais serviços oferecidos por essa tecnologia são:

- a) Software como Serviços (Software as a Service SaaS) através de portais web, são disponibilizadas aplicações para usuários em geral.
- b) Plataformas como Serviços (*Plataform as Service PaaS*) oferece infraestrutura de alto nível para desenvolvedores projetarem e testarem aplicações.



c) Infraestrutura como Serviços (*Infrastructure as a Service - IaaS*) – é oferecido infraestrutura tecnológica, como capacidades de armazenamento, processamento e comunicação de hardwares.

4. Vantagens e desafios na implantação de *cloud computing* nos processos colaborativos organizacional

Atualmente, *cloud computing* vem se destacando por prover integração e mobilidade nas comunicações a partir de diferentes dispositivos, inclusive a partir de dispositivos móveis. Em relação a custos, *cloud computing* oferece serviços que são fornecidos e gerenciados como *utility* que, para CAMEIRA *et al.* (2012), significa obter recursos computacionais sob demanda, assim como acontece com a eletricidade. Portanto, é sensato imaginar que, custos serão poupados na aquisição e manutenção de tecnologias, bem como com a diminuição do tempo de ociosidade de hardwares/softwares e gastos com energia.

Esses e outros fatores levaram empresas como Google, CISCO, IBM e Amazon a visar *cloud computing* como principal metodologia de TI do futuro. NEOGRID (2012), por exemplo, afirma que, *cloud computing* não se restringe a mais uma ferramenta computacional e sim, a mais atual e inovadora lógica de colaboração, que envolve não somente processos operacionais, mas, acima de tudo, envolve a convergência de pessoas e tecnologias para integrar sistematicamente todos os processos organizacionais e evitar distorções das informações compartilhadas.

Diante disso, o objetivo principal de *cloud computing* é permitir maior poder na velocidade das transações e acesso a serviços computacionais a um preço acessível.

No entanto, apesar das vantagens, existem desafios inerentes à adoção de *cloud computing*. Em primeiro lugar, é importante frisar que, a tecnologia evolui com o propósito de automatizar processos gerenciais, portanto, não adianta investimentos em tecnologia, se as organizações não possuírem metas e objetivos bem definidos. Diante disso, o mais importante desafio a ser considerado não está na tecnologia adotada, mas sim, no alinhamento da tecnologia aos processos gerenciais da organização.

O desafio que a tecnologia empregada oferece é o próximo passo a ser considerado. Especificamente, segundo SOUZA *et al.* (2010), alguns dos principais desafios encontrados na adoção de *cloud computing* são:

- a) Disponibilidade de serviços se *cloud computing* armazena dados através da Internet, é possível que em algumas situações os serviços de rede não estejam disponíveis. Portanto, é necessário implantar modelos de nuvem diferentes, pois, se uma das nuvens parar de funcionar, a outra nuvem será acionada.
- **b)** Escalabilidade e desempenho *cloud computing* deve oferecer desempenho e flexibilidade para se adaptarem a quantidades enormes de requisições, principalmente se o modelo de implantação for público.
- c) Avaliação de serviços em nuvem cloud computing deve oferecer ferramentas que gerem de forma eficiente cargas de trabalho, monitoramento do desempenho do sistema e simuladores para acompanhar processos inerentes à implantação de nuvens.



d) Segurança dos serviços de dados – políticas de segurança devem ser continuamente postas em prática, a fim de prover confidencialidade, autenticidade e integridade dos dados que trafegam em *cloud*.

Diante dos desafios, a Intel formou uma aliança denominada de Aliança para Data Centers Abertos (*Open Data Center Alliance*) com mais 70 empresas globais de TI para definir padrões específicos para *cloud computing*, bem como definir como será o seu futuro, criar planos para a próxima geração de data centers e, sobretudo, facilitar o seu entendimento por parte do mercado (INTEL, 2010).

5. Metodologia

O objetivo central dessa pesquisa é analisar a opinião dos gestores do Porto Digital - PE sobre a utilização de cloud computing nas práticas de colaboração organizacional. Para a realização dessa pesquisa foram utilizados os seguintes procedimentos: revisão bibliográfica, pois se baseou em material disponível já trabalhado por outros pesquisadores (SEVERINO, 2007) e estudo de caso, pois foram utilizadas técnicas de interpretação da realidade do objeto escolhido e, posteriormente, obtidos resultados de forma quantificável. A coleta de dados se deu através do questionário, que para CERVO et al (2007), é o método que possibilita medir com mais precisão o que se deseja analisar. O questionário foi elaborado a partir de uma ferramenta denominada de Google Docs e o link do questionário foi enviado por e-mail às empresas situadas no Porto Digital – PE. Este questionário, por sua vez, ficou disponível entre os dias 18/04/2013 a 30/05/2013.

Em relação à população e a amostra, foi utilizada a perspectiva mais atual de GOUVEIA (2006) que diz que "a estrutura da amostra tem de ser consistente com a estrutura da população", ou seja, deve privilegiar o grau de variabilidade dos elementos que a compõe. Deste modo, quanto mais homogênea for a composição da população, menor será o tamanho da amostra e vice-versa. O universo desta pesquisa foi composto por 181 empresas, tanto de TI quanto de Economia Criativa, que estão incubadas no Porto Digital – PE. Para calcular o tamanho da amostra, foi utilizado o conceito de ARSHAM'S (http://home.ubalt.edu/ntsbarsh/business-stat/opre504.html) e, com o resultado, foi possível identificar que seriam necessários 14 questionários respondidos. O número obtido foram 28 questionários respondidos, número suficiente para analisar os resultados.

6. Estudo de caso

O Porto Digital (http://www.portodigital.org) foi escolhido para esse estudo por representar um ambiente de inovação que integra de forma colaborativa instituições, universidades e governos para fomentar mudanças econômicas e sociais e gerar riqueza, emprego e renda.

Situado na cidade do Recife/PE, o principal objetivo do Porto Digital é favorecer ambientes de pesquisa de base tecnológica para a geração de conhecimento na região.



7. Análise dos resultados

Para DAFT (2002), o relacionamento da organização com todos os parceiros de negócio, passou a ser considerado ponto critico para a sobrevivência das atuais corporações.

Com isso, questionou-se aos gestores do Porto Digital – PE, como a colaboração organizacional produz resultados em suas equipes. Diante das respostas, 48% dos gestores afirmaram que "Produz resultados positivos", enquanto que 52% afirmaram que "Produz resultados excelentes". As opções "Não produz resultados" e "Produz resultados negativos" não obtiveram respostas, confirmando assim, a importância da colaboração para a geração de valor. Conforme mostrado no Gráfico 1.

Como a colaboração produz resultados?

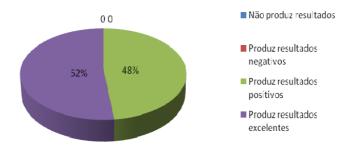


Gráfico 1. Colaboração Organizacional

Diante da intensa necessidade de colaboração, um novo paradigma computacional denominado de *cloud computing* vem sendo considerado a mais inovadora lógica de colaboração em rede. Dessa forma, investigou-se a utilização dessa tecnologia nas práticas colaborativas das empresas situadas no Porto Digital-PE. Conforme mostrado no Gráfico 2, os resultados demonstraram que, 90% das empresas utilizam *cloud computing* em processos colaborativos, enquanto que 10% não utilizam.



Gráfico 2. Utilização de *cloud computing* nos processos colaborativos na organização

Através desse resultado, percebeu-se que, apesar de ser amplamente utilizada, alguns gestores não percebem a utilização dessa tecnologia. Para MARTINS (2010), cloud computing não é algo novo, algo com um alto nível de inovação ou quebra de paradigma, mas sim, uma ideia que surgiu na década de 60 e que, desde então, vem



sendo utilizada e continuamente aperfeiçoada nos seus pontos críticos. O principal exemplo de *cloud* são os e-mails. Ainda segundo MARTINS (2010), a evolução que está ocorrendo é que algumas organizações o estenderam para prover servidores virtuais, onde usuários poderão requerer acesso a recursos computacionais sob demanda. Com base nisso, o questionário utilizado para a realização dessa pesquisa foi respondido através de e-mails, fato que, por si só, afirma a utilização de *cloud computing*. O uso de dispositivos móveis é outro ponto que afirma a utilização de *cloud computing*.

A questão 2 torna-se ainda mais especial, devido a esses resultados terem sido obtidos em um dos maiores centros tecnológicos do país. Portanto, fica evidente a necessidade de estabelecer métodos que venham a fornecer o entendimento de *cloud computing* por parte do mercado.

Diversas empresas influentes, como Amazon e Google atribuem diversas vantagens no uso de *cloud computing*.

Com isso, investigou-se a opinião dos gestores em relação a principal vantagem de *cloud computing* nas práticas colaborativas. Os resultados demonstraram que, 39% enxergam como principal vantagem a "TI como *utility*", 29% os "Custos baixos na aquisição de recursos computacionais", 13% os "Menores gastos com energia e equipamentos eletrônicos" e 18% dos gestores responderam "Outros".

Principal vantagem na utilização do cloud computing nos processos colaborativos na organização

18% 29% Custos baixos na aquisição de recursos computacionais Menores gastos com energia e equipamentos eletrônicos Outros

Gráfico 3. Principal vantagem na utilização de *cloud computing* nos processos colaborativos da organização

Com isso, foi questionado a opinião dos gestores que responderam "Outros", quais são as outras vantagens atribuídas a *cloud computing*. Dentre as mais citadas, destacou-se a mobilidade no acesso a informações independente do local geográfico.

Em seguida, analisou-se a opinião dos gestores a respeito do principal desafio oferecido por *cloud computing*. Diante dos resultados, 53% dos gestores enxergam "Segurança de Informação", 13% a "Interoperabilidade nos Dados", 21% a "Rapidez no Acesso as Informações", 3% a "Escalabilidade e Desempenho" e, por fim, 11% responderam "Outros". Conforme mostrado no Gráfico 4.



Principal desvantagem oferecida pela cloud computing nos processos colaborativos na organização

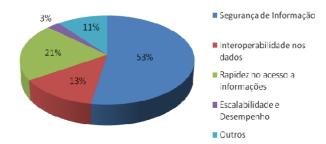


Gráfico 4. Principal desvantagem oferecida pela *cloud computing* nos processos colaborativos da organização

Dessa forma, questionou-se a opinião dos gestores que responderam "Outros", quais são os outros desafios na utilização de *cloud*. Dentre os mais citados, destacaramse: problemas com o acesso a internet e adaptação da ferramenta ao processo.

Cloud computing possui uma abordagem baseada em utility, que possibilita a obtenção de recursos computacionais sob demanda. Com isso, é possível simplificar a gestão de infraestrutura de TI, bem como diminuir os custos com manutenção, pois estes aspectos ficarão sob responsabilidade dos provedores dos serviços. Dessa forma, questionou-se a opinião dos gestores a respeito do que essa tecnologia proporciona em relação aos custos de implantação e gestão de recursos tecnológicos. Diante dos resultados, 4% dos gestores enxergam que cloud computing oferece "Baixa redução de custos, 65% "Moderada redução de custos" e 31% "Alta redução de custos". Conforme apresentado no Gráfico 5.

Custos em relação a implantação e gestão de recursos tecnológicos

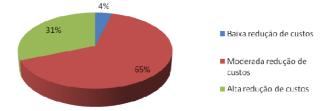


Gráfico 5. Implantação e gestão de recursos tecnológicos

Apesar dos desafios inerentes ao uso dessa tecnologia, sua utilização é crescente nos negócios. Dessa forma, objetivou-se analisar a opinião dos gestores a respeito dos investimentos que *cloud computing* merece. Diante das respostas, 39% afirmaram que merece "Altos investimentos", 54% afirmaram "Moderados investimentos" e apenas 7% afirmaram que "Baixos investimentos" devem ser direcionados a *cloud computing*. Conforme apresentado no Gráfico 6.



Investimentos em relação à cloud computing

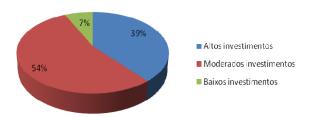


Gráfico 6. Investimentos em relação à cloud computing

Por fim, questionou-se a opinião dos gestores a respeito de qual será o futuro de *cloud computing* nas práticas de colaboração organizacional. Foram obtidas inúmeras respostas, entretanto, a mais frequentes foi: Será o padrão, o novo degrau da inovação em colaboração.

8. Análise e discussão

Com os resultados obtidos, foi possível perceber que, apesar dos desafios e da falta de entendimento de algumas pessoas, o crescimento dessa tecnologia nas práticas colaborativas é evidente. Além disso, *cloud computing*, para muitas das organizações estudadas, não se restringe a mais uma ferramenta computacional, mas sim, ao mais novo degrau da inovação em colaboração entre corporações.

9. Conclusões e Trabalhos Futuros

Este estudo apresentou a opinião dos gestores do Porto Digital – PE a respeito das principais questões de *cloud computing* nas práticas de colaboração organizacional e relacionou os resultados obtidos a produção teórica atualmente disponível. De acordo com os resultados obtidos, foi mostrado que, apesar dos desafios e da falta de entendimento de algumas pessoas, *cloud computing* vem ganhando espaço nas práticas colaborativas por possibilitar acesso a recursos computacionais sob demanda e o acesso a informações de maneira onipresente.

As empresas também opinaram sobre qual será o futuro dessa tecnologia nas práticas de colaboração. De acordo com as respostas, algumas empresas já concluíram que, *cloud computing* é o mais novo degrau de inovação em colaboração organizacional.

Para estudos futuros, é necessário que existam contribuições contínuas para o aperfeiçoamento dos pontos críticos de *cloud computing*, bem como o desenvolvimento de métodos que levem ao conhecimento geral das pessoas a respeito da utilização dessa tecnologia.

Referências

A. Gouveia de Oliveira (2006). Bioestatística, Epidemiologia e Investigação. LIDEL - Edições Técnicas, Lda. ISBN 978-972-757-558-9.

Alliance, C. S. Guia de segurança para áreas críticas em Computação em Nuvem V2. 1. (2010). Disponível em:



- Badge, L.; Grance, T.; Corner. R. P.; Voas, J. Cloud Computing Synopsis and
- Recommendations. Recommendations of the National Institute of Standards and Technology. (2012).
- Cameira, R. F.; Belloni, G. P. V.; Rosa, T. F. O poder das nuvens: integração e inteligência (2012)
- Cervo, A. L.; Bervian, P. A.; Silva, R.; Metodologia Cientifica. 6. Ed. São Paulo: Pearson Prentice Hall, (2007).
- Christopher, M. Logística e gerenciamento da cadeia de suprimentos, 2ª edição (2009).
- Daft, R. L. Organizações: teorias e projetos. São Paulo: Pioneira Thomson Learning, (2002).
- Dr.Arsham's. Statistics Site. Statistical Thinking for Managerial Decisions. Disponível em: http://home.ubalt.edu/ntsbarsh/business-stat/opre504.htm
- Intel. Intel e grandes empresas anunciam uma nova missão para a computação em nuvem: Data Centers abertos (2010). Disponível em: http://newsroom.intel.com/community/pt_br/blog/2010/10/28/intel-e-grandesempresas-anunciam-uma-nova-miss%C3%A3o-para-a-computa%C3%A7%C3%A3oem-nuvem-data-centers-abertos
- Martins, A.; Fundamentos de Computação em Nuvem para Governos.
- Neogrid. O poder de uma malha global (2012). Disponível em: http://www.neogrid.com/pt/novidades/217>
- Severino, A. J.; Metodologia do trabalho científico. 23º Edição Editora Cortez., 2007
- Souza, F. R. C.; Moreira, L. O.; Machado, J. C. Computação em nuvem: Conceitos, Tecnologias, Aplicações e Desafios (2010).



Avaliação de Desempenho de Simuladores de Redes Ad Hoc

Rafael R. Souza¹, Ricardo B. Rodrigues¹, Jamilson R. Dantas¹, Kádna M. A. Camboim²

¹Centro de Informática, Universidade Federal de Pernambuco (UFPE) Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária – 50.740-560 – Recife – PE – Brasil

² Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil

{rafaelmarlin,ricardobatistarodrigues}@gmail.com,jrd@cin.ufpe.br, kadna@uag.ufrpe.br

Resumo. Tradicionalmente, pesquisadores utilizam simulações para testar redes ad hoc. No entanto, a simulação de grandes redes ainda é uma tarefa muito tediosa que consome uma grande quantidade do poder computacional da ferramenta. A ferramenta de simulação a ser comparada é o JiST/SWANTS, que é relativamente nova e tem a promessa de fornecer importantes vantagens de desempenho em comparação a ferramenta mais utilizada, o NS-2. Este trabalho tem como objetivo principal, apresentar uma comparação entre os simuladores de redes ad hoc, utilizando o protocolo e o modelo de mobilidade implementados nos mesmo. Usando parâmetros idênticos de entrada, foram comparadas e analisadas as reais diferenças entre os simuladores.

Abstract. Traditionally, researchers use simulations to test ad hoc networks. However, the simulation of large networks is still a very tedious task that consumes a large amount of computational power of the tool. The tool to be purchased is the JiST/SWANTS, which is relatively new and has the promise of providing significant performance advantages in comparison to what already exists, as well-known NS-2. This work has as main objective, show a comparison between the simulation of ad hoc networks, using the protocol and the mobility model implemented in the same. Using identical parameters for entry, which is when the results are comparable and analyze their real differences. Ad hoc networks have attracted great interest in the academic, military and private companies. In this evaluation are used NS-2 and JIST/SWANTS.

1. Introdução

Atualmente, o mundo vem se deparando com grandes e rápidas transformações no campo da TIC (Tecnologia da Informação e Comunicação), passando a exigir uma maior desenvoltura dos sistemas e dos meios para operá-los. Há crescente demanda por dispositivos portáteis, como *notebook* e telefones celulares. Isso fez com que nos últimos anos ocorresse uma vasta proliferação de tecnologias de redes wireless WLAN



(Wireless Local Area Network) tais como Bluetooth, WIMAX (Worldwide Interoperability for Microwave Access), 3G (Third Generation), 4G (Fourth Generation). Todos esses equipamentos têm oferecido uma solução atraente aos usuários que pretendem obter uma instalação rápida, simples e sem os problemas associados ao cabeamento.

De acordo com [Perkins 2008] as redes *ad hoc*, são redes sem fio que não precisam de infraestrutura para se comunicar, permitindo que dispositivos portáteis possam se comunicar e interagir entre si de forma espontânea, prática, portátil e flexível. Referem-se basicamente a uma rede de computadores em que não são necessários cabos, tendo como exemplos *notebook*, PDA (*Personal Digital Assistant*) e qualquer outro item do gênero, que utiliza a comunicação de redes sem fio, por meio de rádio frequência ou infravermelho. Fatores primordiais como topologia dinâmica, flexibilidade e mobilidade estão alimentando o crescimento explosivo do mercado de computação móvel.

Um dos métodos de avaliar os *softwares*, em redes *ad hoc*, é por simulação, mas existem outras formas, tais como método analítico ou algébrico, implicado num profundo conhecimento matemático. Dependendo do modelo matemático a resolução torna-se extenuante, o que obriga a simplificações do modelo, o que pode resultar em imperfeições na representação do sistema.

2. Fundamentação Teórica

2.1. Redes Móveis Ad Hoc

As redes *ad hoc* sem fio, referenciado pelo IEEE (*Institute of Electrical and Electronics Engineers*) como MANET são redes móveis sem fio (ou roteadores), que formam, dinamicamente, uma rede temporária sem a utilização de qualquer infraestrutura de rede existente ou de administração centralizada e que podem comunicarse, entre si, sem a necessidade de uma estação de suporte ou um ponto de acesso centralizado [Perkins 2008]. A Figura 1 mostra uma adaptação de mobilidade de redes *ad hoc*.

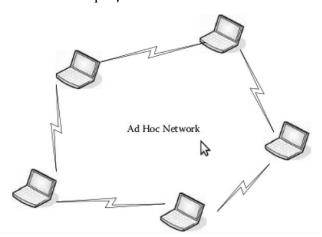


Figura 1. Mobilidade de Redes ad hoc. Adaptado de [Perkins 2008]



2.2. Protocolo de Roteamento

As características das redes *ad hoc* e os protocolos de roteamento atuam de duas formas para atender aos tipos de estrutura que são [Raju and Garcia-Luna-Aceves 2000]:

Pró-Ativo: nesse caso, o algoritmo está armazenando na sua tabela, em que todas as rotas estão inerentes ao nó de modo que, quando uma das rotas for requisitada, ela pode ser utilizada imediatamente. Uma das vantagens é o rápido acesso a uma rota conhecida, mas com isso ocorre um consumo maior de energia, por estar guardando as informações.

Reativo: nessa perspectiva de ação, o algoritmo não armazena continuamente as informações na tabela, mas espera que alguma rota seja requisitada, e então passar para um procedimento de busca. Com isso, o redirecionamento da rota fica mais lento, porém, ganha-se com a redução no consumo de energia.

O protocolo AODV [Perkins and Royer 1999], detém características como topologia dinâmica, que sofre mudanças constantes. E um protocolo reativo de formá a construir suas rotas só quando necessário. Assim sendo, acontece um processo que utiliza medidas como inundação para descobrir possíveis rotas de sua necessidade, mostrando, dessa forma, uma grande característica do protocolo que é tentar aumentar a largura de banda disponível, procurando minimizar o fluxo de mensagens emitidas para atualizar suas rotas.

Esta descoberta permite que cada *host* móvel na rede atue como um roteador especializado e as rotas são obtidas quando necessárias, tornando, assim, uma rede autopartida. Cada nó na rede mantém uma tabela de roteamento com as entradas de informações de roteamento para os nós vizinho, onde só é guardado o próximo salto até o destino. A sua consulta por rota é feita pelo acesso à tabela de roteamento, característica herdada do protocolo DSDV (*Destination-Sequenced Distance* - Vector) [Perkins and Bhagwat 1994].

2.3. Modelo de Mobilidade

O modelo de mobilidade pode ser definido como um modelo matemático para tentar representar o comportamento da movimentação dos nós móveis dentro de uma rede *ad hoc*. Ele determina como os componentes vão se movimentar ou variar ao longo do tempo. Um papel muito importante do modelo de mobilidade é a determinação do desempenho no protocolo da MANET, que busca representar o comportamento de nós móveis, para determinar o impacto da mobilidade em sistemas de comunicação.

3. Simuladores de Redes Ad Hoc

São simplesmente ferramentas de simulação orientada a eventos que se tem revelado muito úteis no estudo de natureza dinâmica para redes de ad hoc, bem como funções da MANET e seus protocolos (por exemplo, algoritmos de roteamento, TCP, UDP), fornecendo aos usuários uma maneira de especificar os protocolos de rede, e simular os comportamentos correspondentes. Nas subseções seguintes serão mostrados detalhamentos sobre os simuladores.



3.1. JiST/SWANTS

O JiST é uma plataforma *open source*, desenvolvido em Java, para simulação de eventos discretos, sendo executado sobre uma máquina virtual Java. Desenvolvido por Rimon Barr na Universidade de Cornell. Algumas características do seu projeto trouxeram os seguintes benefícios: (i) utilização da linguagem de programação java para a plataforma de simulação; (ii) grande número de bibliotecas de simulação; (iii) não sendo necessário desenvolver um novo *kernel* do sistemas para simulação [Barr 2004].

O SWANS é construído sobre a plataforma JIST de forma independente, permitindo, assim, a construção de simulação da rede sem fio (BARR et al. 2006). JiST/SWANS é um *software* de simulação desenvolvido em Java, o que torna possível instalá-lo e executá-lo em diversas plataformas como Linux, Windows e Solaris, Mac, e etc, sendo este um acréscimo de grande vantagem.

O SWANS é organizado em componentes independentes de *software* que podem ser compostos de forma a completar as configurações de uma rede sem fio ou rede de sensores, mostrando-se uma arquitetura bastante modular. Este simulador apresenta capacidades semelhantes às do NS-2, além de dar suporte à simulação de redes maiores, ou seja, com uma maior quantidade de estações. A comunidade acadêmica vem investindo muito nesse simulador [Barr *et al.*].

3.2. NS-2

NS (versão 2), mais conhecido como NS2, é simplesmente uma ferramenta de simulação de *open source*. Desenvolvido a partir do projeto VINT (*Virtual InterNetwork Testbed*) [Bajaj et al. 1998]. Sua criação em 1989 tem vindo a ganhar enorme interesse da indústria, academia e governo. Sem dúvida, NS2 tornou-se um dos mais utilizados simuladores de rede de fonte aberta. É um *software* de simulação bastante difundido na academia, utilizado em muitas pesquisas e é muito estável [Issariyakul and Hossain 2008]).

Este simulador é orientado para eventos e tem suporte para pilha de protocolos TCP/IP, que pode ser usado para redes LAN (*Local Area Network*), MAN (*Metropolitan Area Network*) e WAN (*Wide Area Network*). Outro fato atrativo da ferramenta NS-2 é que seu código-fonte é aberto e se encontra devidamente documentado.

4. Estudo de Caso

Qualquer sistema, independentemente da área em que estiver sendo realizado, deve ter seu desempenho avaliado. Para isso, vários métodos são encontrados e eles diferem na forma com que os resultados são gerados e avaliados. Este trabalho realizou as análises em três cenários, sendo executadas dez simulações em cada cenário para cada simulador, realizando sessenta simulações no total.

4.1. Comparação

Antes de comparar as simulações, queremos mostrar alguns aspectos diferentes entre o JiST/SWANTS e NS-2. Ambos fornecem pacotes de simulações usando uma abordagem baseada em eventos discretos. Esta semelhança nos permite concentrar-nos em apenas aspectos chaves desses dois simuladores.



As análises dos simuladores serão obtidas através de duas formas. Para JiST/SWANS será feito através da interface da aplicação, a qual mostra já os resultados, e para o NS-2 será feito através do *script genstats.awk*, desenvolvido para capturar automaticamente os processos de memória, tempo de execução e compara os resultados gerados pelo NS2. E, seguidamente, foram gerados os gráficos.

Nesta seção, serão apresentadas todas as simulações realizadas neste trabalho com seus respectivos resultados. As simulações foram conduzidas com a finalidade de avaliar o comportamento entre simuladores abordados anteriormente.

4.2. Qualidade da Comparação

Foram aplicados cenários como descrito anteriormente, tanto para JIST/SWANTS quanto para NS-2 e utilizado o protocolo de roteamento AODV e o modelo de mobilidade RWP, sendo os mais conhecidos e implementados por padrão em ambas as ferramentas para executar simulações de redes ad hoc. O requisito mais importante para uma comparação válida dos dois simuladores é uma definição concisa da configuração da simulação. Por exemplo, é vital assegurar que os dados e configurações foram gerados da mesma maneira.

4.3. Ambiente de Simulação

As simulações foram feitas por meio do NS-2 [Fall and Varadhan 2009]. A versão utilizada foi a 2.34. O outro simulador foi o JiST/SWANTS na versão utilizada foi a 1.0.6. Ambos foram simulados em um ambiente composto do sistema GNU/Linux e dos softwares JiST/SWANTS que utilizou a JVM versão 1.6.0 e o Eclipse Ganymede e o NS-2 foi executado direto no Ubuntu. As simulações realizadas durante o desenvolvimento deste trabalho foram feitas em uma máquina com processador Intel Core i5, 2,5Ghz, cada qual com frequência de clock, 4GB de memória RAM e 2GB de swap, disco rígido 500gb.

4.4. Configuração dos cenários

Esta seção detalha todos os parâmetros do ambiente de simulação das redes ad hoc. A princípio, para se realizar uma simulação, tem-se como necessidade a definição das seguintes etapas:

- 1. Planejando a Simulação;
- 2. Definindo os nós:
- 3. Definindo a movimentação;
- 4. Definindo o Tráfego;
- 5. Visualizando a Simulação;
- 6. Analisando o trace;
- 7. Gerar os gráficos.

Neste trabalho foi executada uma bateria de 90 simulações. É interessante ressaltar os padrões de cenários globais e as variantes. A bateria de simulação ocorrerá segundo as Tabelas 1 e 2:

Tabela 1. Canário globais.

The departmentation of the description of the descr	N. de Simulações	N. de Execuções	Otde de Nós	Área (m²)
--	------------------	-----------------	-------------	-----------



1	10	50	1500 × 500
2	10	500	1500 × 500
3	10	5000	1500 × 500

Tabela 2. Cenários Variantes.

Parâmetros	Dados	
Tamanho dos pacotes	64bytes	
Taxa de transmissão (pacotes UDP por segundo)	4 p/s	
Protocolo enlace: CSMA/CA (802.11) alcance de transmissão	275m	
Velocidade dos Nós	[0,20]m/s	
Tempo de pausa	0,0	
Tráfego	30 msn por segundo	
Tempo de simulação	900s	
Protocolo de roteamento	AODV	
Modelo de Mobilidade	Randon Waippon	

4.5. Resultados

Nesta sessão serão demonstrados os resultados gerados pelas simulações de interesse principal deste trabalho, os quais visam reproduzir o comportamento dos simuladores de redes *ad hoc* na medida em que estas redes escalonam com o aumento dos nós, ou seja, na medida em que aumentamos em grande escala a quantidade de nós presentes e proporcionalmente a quantidade de nós comunicantes.

Como a proposta deste trabalho é avaliar os simuladores utilizando um cenário específico, as variações dos parâmetros precisam obedecer às restrições impostas por este cenário. As métricas utilizadas para comparar a escalabilidade e o poder computacional de cada ferramenta foram: processamento; memória e tempo de execução. As variações dos parâmetros têm a finalidade de propiciar a avaliação do desempenho dos simuladores quando impostos às condições críticas ou buscar os melhores parâmetros para esse tipo de rede.

4.5.1. Simulação 50, 500, 5000 nós área 1500 x 500m²

Na Figura 2 ilustra-se a medida de tempo em relação à execução da simulação entre os simuladores. Através deste estudo, foi possível mostrar que o tempo de execução do JiST/SWANS foi menor em relação ao NS2 que se manteve crescente ao tempo de execução da simulação com o aumento dos nós na rede.





Figura 2. Resultado do Primeiro Cenário.

A Figura 3 ilustra a comparação entre os simuladores em relação ao consumo de processamento na execução da simulação. Vista a constância do JiST/SWANS e o alto crescimento do NS-2 com o aumento dos nós, nota-se uma melhor adoção à utilização do JiST/SWANS por manter uma constância no processamento ao aumento do numero de nós na rede; dado o fato de que é em Java.

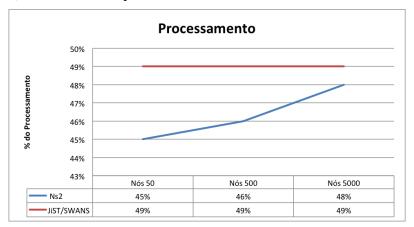


Figura 3. Resultado do Primeiro Cenário.

Outro importante recurso para medir a escalabilidade das ferramentas NS2 e JiST/SWANS é a memória. A Figura 4 na página seguinte, ilustra o consumo de memória ao longo das variações de nós nas simulações. O NS2 consome muitos recursos de memória do que o JiST/SWANS. Visto que, ao simular 5000 nós na rede, o NS2 precisou de 2 Gigabyte de memória enquanto o JIST/SWANS necessitou apenas de 58 Megabyte de memória mostrando, assim, uma melhor adequação do JiST/SWANS a uma rede com grande cenários. Os requisitos de memória tendem a ser um fator limitante para a capacidade de grandes cenários.



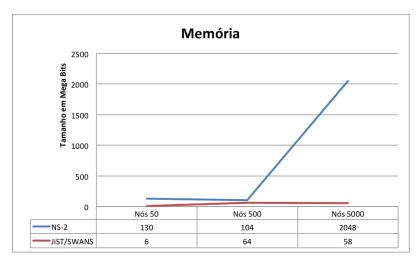


Figura 4. Resultado do Primeiro Cenário.

Esta seção apresentou os resultados de uma bateria de simulação entre os simuladores JiST/SWANS e NS2, utilizando 3 diferentes cenários. Cada simulador executou dez vezes em cada cenário. Sendo, um total de 60 simulações, com isso encontrou-se sua média ponderada e expressa através de gráficos e tabela e as vantagens e desvantagens de cada simulador, mostrando assim seu poder computacional, percentualmente ao aumento da quantidade de nós na rede ad hoc e sua usabilidade.

5. Conclusão

Este trabalho abrangeu um estudo de simuladores de rede *ad hoc* para avaliar seu poder computacional e escalabilidade de ambos e uma pesquisa sobre os protocolos de roteamento e modelo de mobilidade para ser utilizado na simulação. Ao final, para sedimentar esses conhecimentos, foram escolhidos os simuladores JiST/SWANS e NS2, utilizando o modelo de mobilidade *randon waippon* e o protocolo aodv. Ambos os simuladores escolhidos através deste trabalho, para ser estudado e usado nos testes para avaliação dos simuladores. Pode-se concluir que esta pesquisa alcançou as metas projetadas e poderá servir em trabalhos futuros na área, sejam estes trabalhos com o simulador JiST ou trabalhos com NS2

Referências

Bajaj, S., Breslau, L., Estrin, D., Fall, K., Floyd, S., Haldar, P., Handley, M., Helmy,

A., Heidemann, J., Huang, P., Kumar, S., McCanne, S., Rejaie, R., Yu, H., Xu, Y., and et al. (1998). Virtual internetwork testbed: Status and research agenda. Technical report.

Barr, R. (2004). Jist - java in simulation time users guide.

Barr, R., Hass, Z. J., and van Renesse, R. JiST/SWANS: Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator. http://jist.ece.cornell.edu/.

Fall, K. and Varadhan, K. (2009). The ns Manual (formerly ns Notes and Documentation).

Issariyakul, T. and Hossain, E. (2008). *Introduction to Network Simulator NS2*. Springer Publishing Company, Incorporated, 1 edition.



- Perkins, C. and Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on*, pages 90–100.
- Perkins, C. E. (2008). Ad Hoc Networking. Addison-Wesley Professional, 1 edition.
- Perkins, C. E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of the conference on Communications architectures, protocols and applications*, SIGCOMM '94, pages 234–244, New York, NY, USA. ACM.
- Raju, J. and Garcia-Luna-Aceves, J. (2000). A comparison of on-demand and table driven routing for ad-hoc wireless networks. In *Communications*, 2000. ICC 2000. 2000 IEEE International Conference on, volume 3, pages 1702–1706 vol.3.



Um Sistema de Recomendação Baseado em Nuvem

Ricardo Batista Rodrigues¹, Frederico A. Durão³, Rodrigo E. Assad², Vinicius C. Garcia¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE) Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária – 50.740-560 – Recife – PE-Brasil

²Universidade Federal Rural de Pernambuco (UFRPE) Rua Dom Manoel de Medeiros, Campus Dois Irmãos – 52.171-900 – Recife – PE – Brasil

³Instituto de Matemática — Universidade Federal da Bahia (UFBA) Av. Adhemar de Barros, Campus de Ondina — 40170-110 — Salvador — BA — Brasil

{rbr,vcg}@cin.ufpe.br, freddurao@dcc.ufba.br, rodrigo.assad@gmail.com

Resumo. Os sistemas de recomendação têm como objetivo amenizar a sobrecarga de informação auxiliando usuário na busca pela informação desejada. Este artigo apresenta um mecanismo de recomendação de arquivos baseado em nuvem, em um ambiente de armazenamento de dados na nuvem. Com o Sistema de Recomendação, os usuários recebem recomendações de arquivos que são similares as suas preferencias, baseada nos arquivos no qual o usuário salva em sua conta no ambiente. Ao mesmo tempo, que os arquivos recomendados pelo sistema atendem os fatores da nuvem, recomenda ao usuário, arquivos com maior disponibilidade no ambiente.

Abstract. The recommendation systems aim to minimize information overload by helping users in searching desired information. This paper presents a recommendation engine for cloud-based files in an environment of data cloud storage. With the Recommendation System, users receive recommendations of files that are similar to their preferences based on files saved by the user in his account on the environment. At the same time that the recommended system files meet the factors of the cloud assures the user file recommendations with greater availability in the environment.

1. Introdução

Com o advento da computação em nuvens, surgiram os sistemas de armazenamento em nuvem, que possibilitam aos seus usuários armazenar arquivos na nuvem. Com o crescimento da utilização destes sistemas, a massa de dados armazenados na nuvem se tornou humanisticamente impossível de ser processada, implicando na ocultação de informações relevantes aos usuários, que deixam de descobrir novos conteúdos por não disporem de meios eficientes que os auxilie na filtragem de dados em busca de conhecimento relevante e que atenda suas expectativas. Diante deste cenário, sistemas de recomendação se tornam uma alternativa, para auxiliar os usuários na tomada de



decisão por qual arquivo escolher e a filtrar informações relevantes em meio a uma imensidão de dados.

Sistemas de recomendação (SR) são *softwares* e técnicas que fornecem sugestões de itens para usuários [Pazzani and Billsus 1997] [Phelan et al. 2009]. Esses sistemas fazem parte de nossas vidas, diariamente nos deparamos com recomendações via *email* ou em páginas na *web*. Muitas lojas *online* e plataformas oferecem serviços de recomendação, por exemplo, Amazon (http://www.amazon.com) e BarnesAndNoble (http://www.amazon.com) e BarnesAndNoble (http://www.barnesandnoble.com) [Melville et al. 2002]. Existem duas abordagens predominantes na construção de SR, Filtragem Colaborativa (CF) e Filtragem Baseada em Conteúdo (CB). Sistemas CF recomendam itens que são similares às características do usuário, por exemplo, o seu perfil em uma rede social. Sistemas CB recomendam ao usuário itens semelhantesãqueles em que ele demonstrou interesse em experiências anteriores. Para tanto, o sistema analisa as descrições dos conteúdos dos itens avaliados pelo usuário para montar o seu perfil, o qual é utilizado para filtrar os demais itens da base [Blanco-Fernandez et al. 2008] [Ricci et al. 2011] [Phelan et al. 2009].

Os sistemas de recomendação têm por objetivo reduzir a sobrecarga de informação, realizando filtragem de itens baseado nos interesses do usuário. Das diversas' técnicas existentes para realizar tal tarefa, a abordagem utilizada neste artigo é a Filtragem Baseada em Conteudo, que se baseia em arquivos no qual o usuário demostrou interesse no passado [Shardanand and Maes 1995].

Este trabalho apresenta um mecanismo de recomendação em um ambiente de armazenamento de dados na nuvem. Na geração de recomendações e utilizada a técnica de Filtragem baseada em conteudo e características da nuvem. O objetivo do modelo de recomendação aqui propostoé recomendar ao usuário, arquivos que sejam similares as suas preferências e que atendam os fatores da nuvem, desta forma, um arquivo recomendado ao usuário, sempre estará disponível e acessível no ambiente de armazenamento em nuvem, além de proporcional redução no tempo gasto no *download* de um arquivo recomendado e na filtragem de conteúdo relevante em meio a imensidão de dados disponíveis na nuvem.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta o modelo proposto e os resultados. Na Seção 4 são apresentadas as conclusões e os trabalhos futuros deste trabalho.

2. Trabalhos Relacionados

Existem alguns trabalhos na literatura que discutem questões a respeito de sistemas de recomendações em nuvem. Nesta Seção, serão apresentados alguns deles, enfatizando a similaridade e as diferenças em relação ao modelo de recomendação proposto nesta pesquisa.

Lee [Lee et al. 2010], apresenta uma proposta de SR que utiliza dados armazenados na nuvem para proverem suas recomendações, distinguindo-se da proposta deste trabalho que além de recomendar arquivos armazenados na nuvem, tem como objetivo utiliza fatores da nuvem para garantir a disponibilidade dos arquivos recomendados aos usuários. Lai [Lai et al. 2011], apresentam o trabalho que mais se aproxima da proposta desta pesquisa. Onde propõem um SR de programas de TV em



nuvem, com objetivo principal de oferecer um sistema escalável, que tenha uma alta taxa de disponibilidade para o sistema.

O modelo proposto nesta pesquisa utiliza fatores da nuvem para gerar suas recomendações, para garantir a disponibilidade dos arquivos recomendados e a economia do tempo gasto para download de arquivos recomendados, e que as recomendações atendam as preferências dos usuários.

3. O Sistema de Recomendação

Esta seção descreve o desenvolvimento do SR resultado desta pesquisa, onde apresentamos o ambiente de desenvolvimento, o modelo de recomendação, os fatores utilizados na geração de recomendações, e os resulados desta pesquisa.

3.1. Ambiente de Desenvolvimento

O SR desenvolvido nesta pesquisa foi implementado em um ambiente real de armazenamento de dados na nuvem, o Ustore (http://usto.re), que consiste em uma solução para armazenamento e *backup* de arquivos em nuvem privada, que tem como proposta principal as funcionalidades de restore de arquivos (*download*, *upload* e compartilhamento), o ambiente tem como objetivo, prover segurança, alta disponibilidade, ganho de desempenho e redução no tempo de resposta, assim como a utilização de recursos computacionais ociosos.

O ambiente Ustore permite a qualquer usuário fazer *upload*, *download* e compartilhamento de arquivos, disponibilizando a opção de tornar os seus arquivos públicos ou por *default* privados. O SR desenvolvido utiliza em suas recomendações os arquivos marcados como públicos para gerar novas recomendações aos usuários. Na geração de novas recomendações, o SR calcula a similaridade entre o conteúdo dos arquivos públicos armazenados na nuvem do ambiente e os arquivos que o usuário demonstrou as suas preferências, e, em seguida, aplica os fatores da nuvem, propostos nesta pesquisa e apresentados nas seções a seguir.

3.2. Modelo de Recomendação

O modelo de recomendação proposto nesta pesquisa e baseado em fatores da nuvem é em conteudo. Seguindo este modelo, são recomendados aos usuários arquivos similares aos que eles demonstraram interesse no passado, representados pelos arquivos salvos pelo usuário em sua conta no ambiente de armazenamento de dados em nuvem, e que atendam os fatores da nuvem. Na geração de recomendações, propomos a utilização de cinco fatores:

- Disponibilidade
- Similaridade
- Taxa de Download
- Quantidade de Downloads
- Tamanho do Arquivo

A seguir, são delineados cada fator e o processo de recomendação.

Fator Disponibilidade: refere-se ao número de horas em que um peer está disponível na nuvem. Um arquivo só poderá ser recomendado ao usuário se o peer que



o armazena estiver disponível, tornando possível o *download* deste arquivo. A Equação a seguir apresenta o cálculo do fator disponibilidade.

$$Dp = h_i \cdot \frac{1}{n}$$

Na equação anterior, é calculado o fator disponibilidade, ondé *h* e a quantidade de tempo em que um peer *i* está disponível na rede, e *n* representa o número total de horas que um *peer* pode tornar-se disponível na rede (24 horas). O número de horas que um *peer i* está disponível na rede e normalizado para um valor entre 0 e 1. O exemplo a seguir mostra como o fator disponibilidade contribui para a geração de uma recomendação: considerando que dois arquivos A e B são semelhantes, o arquivo A está armazenado em um peer que se encontra disponível na rede de 14 a 16 horas, num total de duas horas de disponibilidade. O arquivo B está armazenado em outro *peer*, que está disponível na rede de 14 a 18 horas, totalizando 4 horas de disponibilidade. Desse modo, o arquivo B e o arquivo que deve ser recomendado ao usuário, pois se encontra disponível na rede por um tempo maior do que o arquivo A, possibilitando a realização do *download*. O objetivo é diminuir o risco de o utilizador não conseguir fazer a transferência.

Fator Similaridade: refere-se a similaridade entre o conteúdo do arquivo armazenado na nuvem e o arquivo pelo qual o usuário demonstrou preferência. Para extrair os arquivos foi usado o Apache Lucene (http://lucene.apache.org), um mecanismo de busca de alta performance, e o Apache Tika (http://tika.apache.org), um detector e extrator de conteúdo de metadados e texto estruturado, podendo ser utilizado para a extração de conteúdo de arquivos de diversos formatos, como HTML, XML, OLE2 e OOXML do Microsoft Office, Opendocument Format, PDF, ePUD, RTF, arquivos compactados e empacotados. Este fator é obtido a partir da técnica *Cosine Similarity*, que e a diferença angular entre dois vetores, através do cálculo do cosseno do ângulo entre eles, independente de seus tamanhos [Baeza-Yates and Ribeiro-Neto 1999]. O resultado do cálculo será sempre um valor entre 0 e 1, onde 0 significa 0% de similaridade e 1 significa 100% de similaridade. A Equação a seguir mostra o cálculo da similaridade.

$$St = cos(\theta) = \frac{A \cdot B}{\parallel A \parallel \cdot \parallel B \parallel}$$

Na equação acima, calcula-se a similaridade de dois vetores A e B, de onde se obtém o produto de A e B e se calcula a magnitude dos vetores A e B. Tais magnitudes são multiplicadas e divididas pelo produto escalar dos vetores A e B.

Fator Taxa de Download: refere-se a taxa de *download* disponível para a realização do *download* de um arquivo que foi recomendado. O objetivo é recomendar arquivos vindos de peers, que são dados com uma conexão melhor para o usuário do *peer*. Este fator e medido de 0 a 20 megabits por segundo (Mbps). A Equação a seguir mostra o cálculo do referido fator.

$$Td = ns \cdot (\frac{1}{n})$$



Na Equação acima é calculado o fator taxa de download Td, onde ns e a taxa de transferência de rede Mbps, sendo este valor normalizado para um valor entre 0 e 1, em que n representa o valor mais alto da taxa de download na rede MBps.

Fator Quantidade de Download: corresponde ao número de *downloads* de um determinado arquivo no ambiente. Isso indica a popularidade e a importância social de um arquivo na mesma rede, sinalizando que muitos usuários se interessaram por um mesmo arquivo. A Equação abaixo mostra o cálculo desse fator.

$$Qd = Q_D \cdot (\frac{1}{0})$$

A Equação a cima calcula o fator quantidade de Download. Para cada *download* realizado de um arquivo específico, o contador de *download* de arquivos e incrementado por 1, este valor e medido de 0 a n, onde *n* corresponde ao maior numero de *downloads* realizados de um único arquivo na rede. O valor de *n* é obtido por meio da observação do histórico de *downloads* de arquivos do ambiente. O cálculo do fator, o número de *downloads* de um arquivo Qd é normalizado para um valor de 0 a 1, multiplicando a quantidade Q_D de *downloads* pela normalização limiar obtida pela notação $\left(\frac{1}{n}\right)$ que divide 1 por *n*, que representa o maior número de download de um arquivo realizado no mesmo ambiente.

Fator Tamanho do Arquivo: refere-se ao tamanho do arquivo a ser recomendado. Esse fator penaliza o score de recomendação quando a taxa de *download* e baixa. Por exemplo, um arquivo A e semelhante ao arquivo B, o arquivo A tem tamanho igual a 9 gigabytes e sua taxa de *download* e de 600 Kbps, já o arquivo B possui 2 gigabytes de tamanho e sua taxa de *download* e de 1 Mbps. Assim, o arquivo B terá maior score de recomendação, por apresentar as melhores condições para a realização de seu *download* (menor tamanho e maior taxa de *download*). A equação a seguir mostra o cálculo desse fator.

$$S = \left(\frac{2^{30}}{S_b}\right) \cdot \frac{1}{n}$$

Na equação a cima, o fator tamanho e representado por S, que corresponde ao tamanho do arquivo a ser recomendado, o qual e obtido em bytes e convertido em gigabytes (GB). A conversão e calculada dividindo-se 1 GB 2^{30} pelo tamanho do arquivo em Bytes S_b , obtendo-se, desta forma, o tamanho do arquivo em gigabytes. O tamanho do arquivo e multiplicado por $\frac{1}{n}$, que será representado por um valor de 0 a 1, sendo o valor 1 dividido por n correspondente ao tamanho máximo de um arquivo que pode ser armazenado no ambiente Ustore. Este valor pode ser definido nas configurações pelo administrador, durante o desenvolvimento desta pesquisa, o valor de n no sistema foi configurado para 10 gigabytes.

Os fatores são ponderados por pesos ω , a depender da sua relevância na geracão de uma recomendação. Os pesos dos fatores são multáveis e configuráveis pelo administrador do sistema, e podem variar de acordo com as características do ambiente. Inicialmente, foram definidos considerando a relevância de cada fator no sistema de recomendações desenvolvido nesta pesquisa e as características do ambiente de armazenamento utilizado. Na Tabela 1 apresentamos os pesos de cada fator.



O fator Similaridade tem peso 5 e representa 50% do *score* de recomendação, para garantir que o conteúdo de um arquivo recomendado ao usuário seja similar ao conteudo do arquivo que o usuário demonstrou interesse. Um arquivo que possua similaridade 0 atribuída as preferências do usuário não deve ser recomendado.

Fator Peso

Similaridade 5

Disponibilidade 2

Taxa de Download 1

Quantidade de 1

Downloads 1

Tamanho do Arquivo 1

Tabela 1. Pesos dos Fatores

O fator Disponibilidade tem peso 2, o que representa 20% do *score* de recomendação, por designar o tempo em que um peer de dados está disponível na rede, tornando possível o *download* de um arquivo recomendado. Um arquivo somente poderá ser recomendado ao usuário se o mesmo estiver armazenado em um *peer* que esteja disponível.

O fator Taxa de Download tem peso 1, representando 10% de uma recomendação. Este fator simboliza a taxa de *download* disponível para realizar o *download* de um arquivo recomendado ao usuário. Um arquivo que possua uma baixa taxa de *download* e que tenha um tamanho maior que o dos demais arquivos similares a ele, não deverá ser recomendado ao usuário, pois o seu processo de recomendação demandará mais tempo e processamento.

Ao fator Quantidade de Downloads e atribuído o peso 1, o que corresponde a 10% do *score* de recomendação. Este fator tem o seu peso inferior aos demais por não ser um fator crítico. Desta forma, um arquivo que não seja popular na nuvem pode ser recomendado ao usuário, o mesmo ocorre com arquivos novos na rede.

Ao fator Tamanho do Arquivo e atribuído o peso 1. Este fator tem peso inferior aos demais fatores por não ser um fator crítico. Desta forma, um arquivo que tenha o tamanho máximo aceito pelo ambiente (10 Gigabytes) pode ser recomendado se a sua taxa de *download* for proporcional, garantindo bom desempenho no *download* do arquivo.

As recomendações são representadas pelo *score* de recomendação, que e o resultado do cálculo mostrado na Equação a seguir.

$$Score = (((St \cdot \omega_S) \cdot (Dp \cdot \omega_D)) \cdot ((Td \cdot \omega_T) + (Qd \cdot \omega_Q))) \cdot \frac{1}{n} - ((S \cdot \omega_Z) \cdot \frac{1}{n})$$

No cálculo apresentado na Equação anterior, o *score* de recomendação é igual ao resultado da soma dos fatores Taxa de Download Td e Quantidade de Downloads Qd, multiplicados por seus respectivos pesos ω_T , ω_Q . O resultado desta notação é multiplicado pelo produto dos fatores Similaridade St e Disponibilidade Dp,



multiplicados por seus pesos ω_S , ω_D , e normalizado por $\frac{1}{n}$, sendo n o maior valor possível desta Equação. Recomendações com score igual ou inferior a 0 são descartadas. Um arquivo somente é recomendado ao usuário se os valores de seus fatores Similaridade e Disponibilidadé forem maiores que 0. Desta forma, o conteúdo de um arquivo recomendado ao usuário sempre sera similares preferências do usuário, e sempre estará disponível para *download*. As recomendações são ordenadas de forma decrescente, pelo valor do *score* obtido na Equação anterior.

3.3. Resultados

O experimento realizado neste trabalho foi executado em um ambiente real de armazenamento de dados na nuvem. O experimento apresentado a seguir, apresenta resultados parciais desta pesquisa, gerado pela simulação de usuários utilizando o sistema. Os principais objetivos desses experimentos são: Analisar a relevância dos arquivos recomendado em relação a preferência elicitada pelo usuário.

O experimento realizado neste trabalho avaliou as recomendações realizadas pelo sistema. Neste experimento foi utilizada uma base de dados contendo 100 artigos científicos de domínio público, a partir desta base na nuvem, foram solicitadas recomendações para arquivos de distinto conteúdo. No total foram avaliadas 50 recomendações, que foram avaliadas como *Like* ou *Dislike*. No caso em que uma recomendação não atenda as preferências e expectativas do avaliador a mesma deve receber a avaliação *Dislike*, ou *Like* no caso da recomendação atender as preferências do avaliador. A Figura 1 apresenta o resultado das avaliações realizadas.



Figura 1. Resultado das Avaliações

A partir da análise dos valores apresentados na Figura 1, inferimos que 85% das recomendações receberam avaliações positivas, o que representa que, a maior parte das recomendações geradas atenderam as expectativas do avaliador. Desta forma, valida as recomendações geradas e o modelo de recomendação proposto.

Para atingir e validar os objetivos desta pesquisa serão realizados experimentos em um ambiente de armazenamento em nuvem disponível em um meio acadêmico, utilizando usuários reais, para elicitarem suas preferências e avaliarem as recomendações recebidas. Assim como a realização de outros experimentos que possam apresentar os ganhos em termo de tempo de *download* dos arquivos recomendados, para que possa se validar o modelo proposto nesta pesquisa.



4. Conclusão e Trabalhos Futuros

Este artigo investiga o impacto de fatores oriundos da nuvem, na geração de recomendações em um ambiente de armazenamento em nuvem. Foi apresentado o modelo proposto nesta pesquisa, bem como os fatores que formam o modelo proposto baseado em nuvem. O desenvolvimento do sistema e os experimentos iniciais foram implementados e executados em um ambiente real de armazenamento de dados na nuvem.

Como trabalhos futuros, julga-se importante refazer e melhorar os experimentos apresentados neste artigo, utilizando usuários reais do ambiente na nuvem, assim como realizar novos experimentos, a fim de comparar os resultados obtidos neste modelo com os demais modelos disponíveis na literatura. Particularmente, pretende-se propor novos fatores baseados em nuvem, que possam contribuir para a melhoria do modelo proposto.

Referências

- Baeza-Yates, R. A. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. AddisonWesley Longman Publishing Co., Inc., Boston, MA, USA.
- Blanco-Fernandez, Y., Arias, J. J. P., Gil-Solla, A., Cabrer, M. R., and Nores, M. L.' (2008). Providing entertainment by content-based filtering and semantic reasoning in intelligent recommender systems. *IEEE Trans. Consumer Electronics*, 54(2):727–735.
- Lai, C.-F., Chang, J.-H., Hu, C.-C., Huang, Y.-M., and Chao, H.-C. (2011). Cprs: A cloudbased program recommendation system for digital {TV} platforms. *Future Generation Computer Systems*, 27(6):823 835.
- Lee, S., Lee, D., and Lee, S. (2010). Personalized dtv program recommendation system under a cloud computing environment. *Consumer Electronics, IEEE Transactions on*, 56(2):1034–1042.
- Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Eighteenth national conference on Artificial intelligence*, pages 187–192, Menlo Park, CA, USA. American Association for Artificial Intelligence.
- Pazzani, M. J. and Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331.
- Phelan, O., McCarthy, K., and Smyth, B. (2009). Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 385–388, New York, NY, USA. ACM.
- Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.
- Shardanand, U. and Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth". pages 210–217. ACM Press.



Investigação de uma Arquitetura para Gerenciamento de Documentos

Sérgio Francisco Tavares de Oliveira Mendonça¹, Edson Alves de Carvalho Júnior¹, João Ferreira da Silva Júnior¹

¹Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n – Boa Vista – 55.292-270 – Garanhuns – PE, Brasil

²Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco (UFRPE)

Rua Dom Manoel de Medeiros, s/n, Dois Irmãos - 52.171-900 - Recife - PE - Brasil sftom@uag.ufrpe.br, edacjr@msn.com, joaoferreirape@gmail.com,

Resumo. Um dos grandes desafios no gerenciamento de documentos consiste na identificação adequada, através da especificação e extração da informação, através de diversas abordagens e características. Definir e implementar técnicas para aquisição, pré-processamento, extração de características e reconhecimento de imagens de documentos. Comparar as técnicas implementadas e, dessa forma verificar a adequação das mesmas no ambiente proposto. O resultado prático, desse trabalho, servirá de protótipo para sistemas de análise de imagens de documentos.

Abstract. A major challenge in document management is the proper identification, through the specification and information extraction, using a variety of approaches and features. Define and implement techniques for acquisition, preprocessing, feature extraction and recognition of document images. Compare the techniques implemented and thus verify the adequacy of the same in the proposed environment. The practical result of this work will serve as a prototype for systems analysis of document images.

1. Introdução

O atual sucesso comercial em toda empresa de sistemas de gestão de documentos é um indicativo da necessidade de adoção de técnicas e estratégias semânticas para a especificação e extração de informações em um conjunto de documentos. A fim de tratar documentos de forma mais eficaz, departamentos introduziram conceitos de redes locais (LANs) nos anos 1980, permitindo aos usuários compartilhar um espaço de arquivo comum. Produtos de *groupware*, como o Lotus Notes facilita o fluxo de informações nas empresas, mas a forma da informação é mais do que uma base de dados que apoia documentos. Outros produtos que trabalham com a gestão de documentos, como Excalibur e Autonomia começaram a meta-estruturar documentos, fornecendo a possibilidade de realizar pesquisa em todo o documento, através de espaços heterogêneos. Embora semanticamente adequado, sistemas de recuperação de



tais informações negligenciam a importância do documento para a organização, sabendo pouco sobre a razão de existência do documento.

Uma solução efetiva adotada por grande número de empresas tem sido a gestão documental, que deve considerar uma série de abordagens diferentes que podem ser usadas em conjunto para satisfazer as demandas identificadas no início desta proposta. Tais abordagens são de fluxo de trabalho e esforços de colaboração, o aumento do uso de hipertexto e linguagens de marcação, e a aplicação de tecnologias *Web* voltados à resolução dos problemas de gerenciamento de documentos, que descrevemos a seguir. A presente proposta apresenta um ambiente de gestão de documentos que foi dividido em dois módulos: Especificação e Extração (Trabalhos Futuros). No módulo de Especificação, o usuário informa a estrutura do documento. O módulo é responsável por capturar informações físicas, semânticas e relativas ao processamento do documento. Essas informações serão usadas pelo módulo de Extração para o processamento de um lote de documentos da mesma classe do documento especificado.

O módulo de Extração possui duas fases principais: Geração do Lote de Documento e Reconhecimento. A primeira fase realiza as tarefas de aquisição, indexação e armazenagem das imagens dos documentos. A fase de Reconhecimento faz uso das informações do módulo de Especificação para pré-processar, extrair características e reconhecer os campos de cada documento do lote.

Adicionalmente, o ambiente oferece a possibilidade de correção de eventuais erros causados pelo classificador, várias formas de visualizar o resultado do reconhecimento, além de permitir a reconstrução do documento através de uma referência cruzada com as informações do módulo de Especificação.

1.1. Justificativa

Memórias organizacionais são motivadas pelo desejo de preservar e compartilhar o conhecimento e experiências que residem em uma organização. A memória organizacional é definida como uma explícita, representação, persistente do conhecimento e da informação em uma organização.

Vale a pena notar que as exigências descritas acima se aplicam aos documentos e informações mais estruturadas da mesma forma. Embora a relação entre potência e custo de *hardware* de computação tem multiplicado por um fator de um milhão nos últimos 30 anos, o custo de organizar as informações dentro desses sistemas tem aumentado [Goranson92].

Há uma necessidade evidente de melhorar a gestão da informação digital de todos os tipos. No entanto, de acordo com o Relatório Gilbane de Informação e de Sistemas de Documentos Abertos cerca de 80% da informação eletrônica corporativa está na forma de documentos, ao contrário dos estruturados registros de banco de dados. Esta estatística destaca a necessidade de foco, pelo menos, tanto na gestão de documentos sobre a gestão de informações mais estruturadas dentro de uma empresa, o que justifica a realização deste projeto de pesquisa.

Sistemas como o Excalibur, Autonomia e Lotus Notes são habilitados para *Internet*, mas não *internet-driven* [Bentley97]. É evidente que uma convergência de



técnicas de gerenciamento de documentos e distribuição WWW seria vantajosa para facilitar uma visão de toda a empresa de documentos.

1.2. Objetivos

O principal objetivo desse trabalho é o desenvolver uma arquitetura para um ambiente computacional, que possa extrair informações dos mais diferentes tipos de documentos em papel, transformando-os em documentos eletrônicos estruturados.

Para se conseguir sucesso nessa empreitada, foi necessário a interação de diversos mecanismos de Processamento de Imagens e Sistema de Informação. Esses mecanismos realizarão tarefas que vão desde a captura da imagem e conversão da mesma para um formato digital, passando pelo reconhecimento de suas partes, até o armazenamento dos dados classificados.

O ambiente possuirá dois módulos principais: Especificação e Extração de Informações. O primeiro especifica um documento em papel, baseado em um modelo orientado a objetos. Partindo dessa especificação, o segundo módulo reconhecerá, de forma automática, o conteúdo de vários documentos do mesmo modelo, adquiridos através de algum dispositivo de entrada. Os conteúdos classificados serão armazenados em um banco de dados.

É possível desmembrar o objetivo principal desse trabalho, que é desenvolver uma metodologia para a especificação e extração de informações de imagens, em alguns objetivos secundários, entre eles, podem-se destacar os seguintes:

- Definir e implementar técnicas para aquisição, pré-processamento, extração de características e reconhecimento de imagens de documentos;
- Comparar as técnicas implementadas e, dessa forma verificar a adequação das mesmas no ambiente proposto;
- Confeccionar um banco de dados, com imagens de documentos, que será utilizado durante o estudo experimental;
- Desenvolver um sistema computacional, que valide a metodologia proposta.

O resultado prático, desse trabalho, servirá de protótipo para sistemas de análise de imagens de documentos.

2. A Gestão de Documentos

Uma solução efetiva adotada por grande número de empresas tem sido a gestão documental, pois considera uma série de abordagens diferentes que podem ser usadas em conjunto para satisfazer as demandas identificadas no início desta proposta. Tais abordagens são de fluxo de trabalho e esforços de colaboração, o aumento do uso de hipertexto e linguagens de marcação, e a aplicação de tecnologias *Web* voltados à resolução dos problemas de gerenciamento de documentos, que descrevemos a seguir.

2.1. Fluxo de Trabalho e Colaboração

Muitos sistemas de informação se concentram nos aspectos informativos do ambiente de trabalho em que eles são usados. Grande parte da literatura de gestão de documentos está preocupada com as maneiras em que o conteúdo informacional de documentos pode ser capturado e manipulado. No ambiente de trabalho muitos destes sistemas



implicitamente apoiam os colaboradores em suas funções operacionais e de tomada de decisão. Abordagens como Computer Supported Co-operative Work (CSCW) fazem o papel explícito de que sistemas de informação realizam para seus usuários. [Swaby98] reconhece três requisitos gerais de motivação que a CSCW procura abordar:

- Articulação do trabalho cooperativo de Coordenação entre pessoas e recursos para contribuir para o desempenho de uma tarefa comum;
- Compartilhamento de um espaço de informação para assegurar que os membros do grupo possam compartilhar dados, informações, conceitos e heurísticas de uma forma estruturada;
- Adaptação da tecnologia para a organização criando um contexto organizacional no qual as atividades do grupo possam ser situadas, e permita interações apropriadas nesse cenário.

Este artigo não replica a revisão do CSCW que pode ser encontrada em [Swaby98], mas não reconhece e apoia a necessidade de uma abordagem de fluxo de trabalho para sistemas de informação. Produtos de *workflow*, como Lotus Notes pode auxiliar o fluxo de informações em torno de uma organização distribuída, mantendo um modelo de processos de trabalho dos usuários finais.

'Colaboração' o termo tradicionalmente sugere um nível de trabalho cooperativo com um objetivo comum. No contexto desta proposta o objetivo comum é a construção e manutenção de uma coleção de recursos voltados à especificação e extração de informações presentes em um documento que satisfaz um conjunto de metas semanticamente estabelecido.

A primeira classe de ferramentas, aquelas de curtas distâncias conhecidas entre os trabalhadores são exemplificadas por espaços de trabalho compartilhados e sistemas de conferência [Greenwood95]. Um requisito fundamental para essa colaboração é a integridade garantida de um recurso compartilhado em todos os momentos. Produtos como o Lotus Notes fornece esses mecanismos há alguns anos, mas em um formato proprietário e dentro de uma rede restrita. WebDAV [Whitehead98] é um desenvolvimento atual que amplia a *World Wide Web* (WWW) para fornecer uma infraestrutura padrão para criação colaborativa assíncrona através da *Internet*.

2.2. Hipertextos e Linguagens de Marcação

O campo de hipertexto teve início, em 1945, com o trabalho de Vannevar Bush que criou uma máquina que associava dois documentos, similar à maneira que a mente opera por associação [Bush45]. Como uma definição, um hipertexto é um conjunto de nós de informação com a máquina apoiando ligações entre esses nós e uma interface de usuário comum [Cunningham93]. Hipertexto expressa as relações entre os documentos. O primeiro protótipo de hipertexto eletrônico, NLS Aumentar, foi demonstrado em 1968. Muitos dos princípios fundamentais caracterizado por sistemas hipermídia de hoje já estavam presentes no protótipo inicial. O criador, Doug Engelbert, mesmo assim, viu o potencial para o computador de suporte aumentar o QI coletivo de comunidades ou a inteligência coletiva das organizações [Balasubramanian94].

A WWW tem popularizado a noção de hipertexto, com sua habilidade para saltar de uma parte de um documento para outra parte de outro documento ou para outra parte do mesmo documento. A sobrecarga cognitiva de seguir o *link* é menor do que o de



realizar uma pesquisa através de um espaço de documento para encontrar os recursos adequados. [Golovchinsky97] sugere que as consultas de *link*-dirigido são mais eficazes do que as consultas específicas na recuperação de informações relevantes. A mecânica de hiperlinks é poderosa, mas a qualidade da ligação é dependente da pessoa que criou o *link*, ou seja pode ocorrer problemas adicionais com hiperlinks.

Em 1969 a IBM desenvolveu a linguagem de marcação generalizada (GML), destinada desde o início para separar forma e conteúdo na exibição de documentos [Berghel99]. Em 1986, o Padrão Generalizado de Marcação foi estabelecido como um padrão aceitável pela comunidade.

2.3. Tecnologias Web

A rede de computadores reconhecida pela primeira vez, a ARPANET, foi formada em dezembro de 1969. Ela consistia de quatro nós e tinha a intenção de descentralizar as informações militares e controle. Os usuários da ARPANET começaram a usar a rede para mais do que a intenção de computação de longa distância. Pesquisadores começaram a usar ARPANET para colaborar em projetos, ao comércio e para enviar mensagens de uma natureza mais social. Não houve dependência do tipo de computador usado para conectar a ARPANET, a única condição era o protocolo de rede usado para se comunicar. A ARPANET formou os primórdios da *Internet* como é hoje: um conjunto diversificado de computadores capazes de se comunicar com qualquer outra por meio de protocolos comuns e de uma rede de comunicação generalizada.

A computação cliente/servidor pode ser pensada como "processos de computador para a comunicação entre os computadores" em que um processo é o solicitante e os serviços de outro processo do pedido, normalmente através de uma rede, utilizando um protocolo comum a todos [Schwartz95]. ARPANET foi concebido desde o início para implementar uma arquitetura cliente-servidor, e a rede mundial moderna *Wide* é baseado no mesmo conceito.

A descentralização do poder computacional e informamacional permitem a possibilidade de ultrapassar a organização de gestão de documentos: retenção local de 'propriedade' de documentos e acesso global a outros documentos dentro ou fora da organização. O sistema de gestão de documentos fornece controle central sobre os dados com acesso distribuído a esses dados através de soluções baseadas em padrões, fornecendo redes e protocolos abertos através da qual os servidores podem transportar dados a seus usuários distribuídos.

O sucesso sem precedentes da WWW levou as empresas a introduzir tecnologias *Web* para sua infraestrutura organizacional, denominando "*intranets*" as redes resultantes. O protocolo de rede padrão e a linguagem de marcação da WWW deve fornece uma solução leve, voltada à resolução dos problemas de distribuição de muitas informações, com uma possibilidade de interrupção mínima em toda a organização.

Um navegador-cliente padrão é o requisito único para o *desktop*, com as complexidades de ser gerida de forma centralizada no servidor. No entanto, devido à falta de estrutura semântica dentro da linguagem de marcação atualmente utilizados em todo o WWW (HTML) e *intranets* têm tendência para Sistemas de Informação baseados na *Web* (WIS), em vez de baseado em Sistemas de Gestão de Documentos *Web*. Função que o servidor responsabiliza-se em codificar e separar as informações que fluem entre



o sistema e o usuário, tratando assim do valor agregado. É evidente que as tecnologias da *Web* atual não podem apoiar plenamente em toda a empresa de gestão de documentos. [Balasubramanian98].

2.4. A Natureza da Documentação

Tal como acontece com muitos termos amplamente usados não existe uma definição axiomática de "documento", mas a literatura aponta a pesquisa a alguns temas comuns. Ele também destaca que a atual e abrangente definição de "documento" nem sempre foram tão abertas. Definições mais tradicionais começam e terminam com um documento como "fornecer informações ou provas" um artefato de papel [Levy99].

Embora não seja construtivo para debater a questão em profundidade, um breve olhar sobre as atitudes diferentes aos documentos é útil.

O debate filosófico sobre a forma e a finalidade de documentos tem sido defendido ao longo dos anos, abrangendo muitas tecnologias. Em 1951 Suzanne Briet, um cientista da informação pioneiro, argumentou que o escopo de "documento" prolongado para além de texto serve para qualquer forma de material de provas. Embora útil como um marcador, a definição Briet é demasiadamente ampla para ser útil a este artigo (Briet foi tão longe para concluir que, sob certas circunstâncias, mesmo um antílope poderia ser um documento [Levy99]). Por outro lado, Sandy Ressler afirma que "todo o conceito de um documento está se tornando cada vez mais obsoleto" [Ressler95] como documentos e bases de dados convergem. Entre estes pontos de vista opostos são algumas outras definições importantes e relevantes.

2.5. Documentos como Objetos Black Box

Um documento pode ser descrito como uma unidade de "informação estruturada gravada para o consumo humano" citado em [Sprague95]. Esta definição implica uma distinção entre o método de armazenamento (informação registrada) e os meios pelos quais o conteúdo é entendido (consumo humano). Ele constrange o apoio da máquina potencial de pouco mais de um mecanismo de transporte para objetos *Black Box*. A noção de um documento como um objeto binário grande (BLOB): uma *Black Box* é armazenada e recuperada de acordo com alguma classificação externa.

Durante o desenvolvimento das redes de distribuição da indústria de computação houve uma preocupação em ajudar o usuário a criar, armazenar e recuperar documentos, mas não forneceu suporte explícito para a compreensão de conteúdo do documento fora do aplicativo de *software*. Ao considerar os documentos como objetos *Black Box* é confiar unicamente nos esquemas de classificação e métodos de gestão que a comunidade de usuários tem aplicado ao espaço do documento independentemente do conteúdo dos documentos armazenados.

Não são apenas as classificações subjetivas de documentos [Nurnberg97], impulsionado pelo contexto local e prática, mas quantidades crescentes de documentação exacerbam os problemas do uso de classificações aplicado apenas explicitamente como uma chave em uma loja de documento [Apte94]. A classificação única do documento só pode, em parte, descrever a finalidade ou função de um documento [McCarty97] e uma classificação mal aplicada pode processar um documento incoerente desde o início.



2.6. A Composição de Documentos

Ao considerar os documentos como objetos *Black-Box* ignora-se a intenção original e o propósito da criação do documento. Para compreender a relevância dos documentos para uma organização é necessário para entender o que um documento contém. Um documento é um objeto instantâneo de um conjunto de informações que podem incorporar muitos tipos de informação, existem em vários lugares através de uma rede, dependem de outros documentos de informação, mudança de objetivos, e têm uma estrutura complexa [Sprague95]. Esta definição realça a natureza de composição de documentos.

Heeman discute dois componentes separados: a estrutura do documento e conteúdo do documento [Heeman92], com destaque para uma distinção entre o conteúdo físico de um documento e sua estrutura lógica. O documento descrito por suas marcações mais o conteúdo do documento separado de sua apresentação, criando três camadas amplas, mas distintas: estrutura, conteúdo e apresentação (ver Fig. 1).

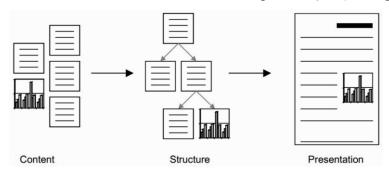


Figura 1: Estrutura básica de um documento (Heeman, 1992).

Na Fig. 1 descrevemos *Content*: os componentes (palavras, imagens, etc) que compõem um documento; *Structure*: a organização e os inter-relacionamentos entre os componentes; *Presentation*: como um documento parece e os processos que são aplicados a ele.

O conteúdo (tipo) foi reorganizado para formar uma estrutura (página) e depois apresentado em um meio adequado (papel ou meio digital). Métodos modernos facilitam a variância em todas as três camadas: a personalização da apresentação, validação da estrutura e da reutilização do conteúdo. Atualmente linguagens de marcação continuam a desenvolver as três camadas de separação e, sobretudo, aumentar a conscientização e aceitação das distinções entre as camadas. O reconhecimento da estrutura dentro de documentos é significativo em relação ao documento de discussão do banco de dados [Ressler95].

A maneira central no qual um tipo de informação difere da outra está em suas categorias semânticas: os objetos que são tidos como primitivos e as relações lógicas entre esses objetos. Convencionalmente o tipo de informação que precisa ser capturado determina a escolha do método e suporte de *software* subsequente [Fillion95]. Se ambos os dados tradicionais e os documentos podem ser descritos em termos de conteúdo e estrutura não deve haver distinção no método de seu tratamento. Dados e documentos estruturados são diferentes apenas na granularidade, não em sua natureza [Heeman92].



3. Metodologia

A Figura 2 mostra uma breve descrição do fluxo de informações no ambiente proposto. A partir dessa representação pictórica do sistema é possível identificar a ordem na qual as ações devem ser realizadas.

O primeiro passo é a criação de uma representação para o documento em questão — <u>Especificação</u> — Uma vez que o documento foi especificado será necessário reunir um conjunto de documentos, todos do mesmo tipo do documento especificado, e formar um lote — <u>Geração do Lote de Documentos</u>. Unindo as informações da especificação a cada documento do lote gerado, o sistema pode reconhecer os campos especificados — <u>Reconhecimento</u>.

Esses 3 (três) primeiros passos devem ser seguidos estritamente na ordem em que foram apresentados. Já que não é possível, nesse sistema, reconhecer um documento que não possua uma especificação. Entretanto, nada impede que a especificação de um documento possa ser mudada, ou que, sejam incluídos mais documentos em um determinado lote.

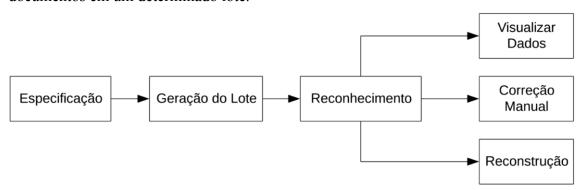


Figura 2: Fluxo de informações no ambiente.

Na Figura 2, nota-se, ainda, outros 3 (três) passos: Visualização dos Dados, Correção Manual e Reconstrução, que encontram-se desenhados com linhas pontilhadas. Após o Reconhecimento qualquer um desses passos pode ser acessado. No passo de <u>Visualização dos Dados</u>, como o próprio nome já diz, é possível verificar o resultado dos algoritmos de classificação. Já no passo de <u>Correção Manual</u>, o usuário pode corrigir eventuais erros gerados por uma má classificação. Além disso, é possível "recriar" o documento a partir dos dados classificados e da especificação do mesmo – <u>Reconstrução</u>.

4. Conclusões e Trabalhos Futuros

A arquitetura desenvolvida para um ambiente computacional conseguiu extrair informações dos mais diferentes tipos de documentos em papel, transformando-os em documentos eletrônicos estruturados, baseados em uma estrutura e metodologia prédefinidas, através uma especificação e extração de informações de imagens. Foram definidas e implementadas técnicas para aquisição, pré-processamento, extração de características e reconhecimento de imagens de documentos. Em seguida, identificamos comparamos as técnicas implementadas e, dessa forma foi verificada a adequação das mesmas no ambiente proposto.



Como Trabalhos Futuros, criamos alguns modelos relacionais de um banco de dados, para armazenamento das imagens de documentos, que serão utilizadas durante o estudo experimental. Resta ainda desenvolver o sistema computacional, que valide a metodologia proposta. Espera-se que resultado prático desse trabalho, sirva de protótipo para sistemas de análise de imagens de documentos.

Referências

- [Apte94] Chidanand Apte, Fred Damerau and Sholom Weiss. Automated Learning of Decision
- Rules for Text Categorisation. ACM Transactions on Information Systems, 12(3):233{251,July 1994.
- [Berghel99] Hal Berghel and Douglas Blank. The World Wide Web. Marvin Zelkowitz, editor, Advances in Computers Volume 49, pages 178 [218. Academic Press, October 1999. [Balasubramanian94] V Balasubramanian and Helen Ashman. Long Distance Perspectives on Hypermedia: Keynote Address at the 1994 European Conference on Hypermedia Technology (ECHT94) by Douglas C. Engelbart and Christina Engelbart, Bootstrap Institute, September 1994.
- [Balasubramanian98] V Balasubramanian and Alf Bashian. Document Management and Web Technologies: Alice Marries the Mad Hatter. Communications of the ACM, 41(7):107{115, July 1998.
- [Bentley97] R Bentley, W Appelt, U Busbah, E Hinrihs, D Kerr, K Sikkel, J Trevor and G Woet-zel. Basi Support for Cooperative Work on the World Wide Web. International Journal of Human Computer Studies: Speial Issue on Novel Appliations of the World Wide Web, 46(6):827{846, June 1997.
- [Bush45] Vannevar Bush. As We May Think. The Atlanti Monthly, pages 101{108, July 1945.
- [Cunningham93] Donald J Cunningham, Thomas M Duy and Randy A Knuth. The Textbook of the Future. C M Knight, A Dillon and J Rihardson, editors, Hypertext A Psyhologia Perspetive, pages 19{49. Ellis Horwood Series in Interative Information Systems, 1993.
- [Fillion95] Florene Fillion, Christopher Menzel, Thomas Blinn and Rihard Mayer. An Ontology-Based Environment for Enterprise Model Integration. Proceedings of the
- International JointConference on Arti ial Intel ligene (IJCAI), Montreal, Canada, August 1995.
- [Golovhinsky97] Gene Golovhinsky. What the Query Told the Link: The Integration of Hy-pertext and Information Retrieval. Proceedings of the 8th ACM International HypertextConferene (Hypertext97), Southampton, UK, pages 67{74, April 1997.
- [Goranson92] H T Goranson. Some Initial Results from Enterprise Integration Studies. Reports of the International Conference on Enterprise Integration Modeling Technology (ICEIMT) Special Interest Groups, Department of Defense, USA, August 1992.



- [Greenwood95] R M Greenwood, I Robertson, R A Snowdon and B C Warboys. Ative Models in Business. Proeedings of the 5th Conference on Business Information Tehnology (CBIT95), Manhester, UK, November 1995.
- [Heeman92] Frans C Heeman. Granularity in Strutured Douments. Eletronic Publishing, 5(3):143 {155, September 1992.
- [Levy99] David M Levy. The Universe is Expanding: Rections on the Soial (and Cosmi) Signiane of Documents in a Digital Age. Bul letin of the American Society for Information Science, 25(4):17{20, 1999.
- [MCarty97] Willard MCarty. The Shape of Things to Come is Continuous Change: Fundamental Problems in Eletronic Publishing. Presented at Electronic Publishing A Day Conference, University of London, UK, January 1997.
- [Nurnberg97] Peter J Nurnberg, John J Leggett and Erich R Schneider. As We Should Have Thought. Proceedings of the 8th ACM Conference on Hypertext (Hypertext97), Southampton, UK, pages 96{101, April 1997
- [Ressler95] Sandy Ressler. Network Based Documents. OII Spectrum Management
- Handbook. January 1995. Available on-line at http://www.itl.nist.gov/iaui/ovrt/people/sressler/netDocs/netDocs.fm.html (Last vericed 8th March 2000).
- [Shwartz95] Gary Shwartz. Everything You Ever Wanted to Know About Client/Servers. Information Technology Article, April 1995. Available on-line at http://magi.com/ mmelick/it95aprl.htm (Last veried 8th March 2000).
- [Sprague95] Ralph Sprague. Electronic Document Management: Challenges and Opportunities for Information Systems Managers. Management Information Systems Quarterly, 19(1):29{49, March 1995.
- [Swaby98] Mike Swaby. A Model-Based Approach to Construction of Integrated Internet CSCW Systems. PhD thesis, School of Computer Studies, University of Leeds, UK, 1998.
- [Whitehead98] E James Whitehead Jr and Meredith Wiggins. WEBDAV: IETF Standard for Collaborative Authoring on the Web. IEEE Internet Computing, 2(5), September/October 1998.



Avaliação de Ontologias de Domínio para o Desenvolvimento Distribuído de Software

Ana Raquel M. Alves¹, Israel Felipe L. A. Silva¹, Renan L. Fernandes¹, Rodrigo C. Rocha¹, Thais A. Burity¹, Ryan R. de Azevedo¹

¹Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n – Boa Vista – 55.292-270 – Garanhuns – PE, Brasil

{anaraqueldemorais,israelita.felipe,rlfnan326}@gmail.com, {rodrigo,thais.burity,ryan}@uag.ufrpe.br

Resumo. O Desenvolvimento Distribuído de Software se tornou uma opção para empresas de software explorarem os benefícios desta abordagem. No entanto, esse conceito traz consigo novos desafios para o cenário de software, como a inexistência de um modelo público formal e normalizado das informações do projeto, o que dificulta a comunicação e o entendimento dos membros e dos artefatos. Dessa forma, este trabalho propõe uma avaliação parcial das ontologias de domínio para DDS, apresentando seus pontos em comuns, entendimento e campo de atuação. Servindo para concepção de novas ontologias para esse contexto com inferência de informações concisas, precisas e não ambíguas, de forma semiautomática dentro do ambiente.

Abstract. Distributed Software Development became an option for software companies to explore the benefits of this approach. However, that concept brings new challenges for the software scenario; the absence of a formal public model and standardized project information is one of these, which make it difficult the communication and understanding of members and artefacts. Thus, this paper proposes a partial evaluation of domain ontologies for DDS, showing their common points, knowledge and acting field. Give the basement for designing new ontologies for this new context with concise information, accurate and non-ambiguous, so semiautomatic within the environment.

1. Introdução

Motivadas pelas vantagens, como disponibilidade de especialistas, redução de custos, incentivos de governos locais e baixas de rotatividade de pessoal, diversas empresas de software tem adotado equipes formadas por pessoas geograficamente distribuídas, configurando o Desenvolvimento Distribuído de Software (DDS).

Essa modalidade de trabalho traz novos desafios para o cenário de desenvolvimento de Software. De acordo com [Carmel 1999] e [Komi-Sirvo and Tihinen 2005] os seguintes fatores podem levar ao fracasso de um projeto distribuído: comunicação ineficiente, falta de gerência e diferenças culturais. Nesse contexto, é possível afirmar que a inexistência de um modelo público, formal e normalizado das informações do projeto dificulta a comunicação e o entendimento dos membros e dos



artefatos, o que pode ser agravado quando a cultura e demais características dos membros das equipes, como o idioma, são distintas entre si.

Para reduzir essa complexidade, o uso de ontologias pode ser útil. [Wongthongtham et al. 2006] afirma que a utilização de ontologias é um novo paradigma para Engenharia de Software e que pode ser utilizado principalmente para conceder semântica para ferramentas auxiliares, prover forte comunicação baseado no conhecimento, centralização e disponibilidade de informação e também, prover uma base de conhecimento para que agentes de software capazes de formar um ambiente automatizado possam apoiar o DDS.

Dessa maneira, uma Ontologia voltada para o DDS permite que todos os membros da equipe tenham a mesma visão e compreensão das informações compartilhadas, além de estarem todos igualmente atualizados em qualquer mudança ou alteração feita no projeto. Como é uma ferramenta que, visa resolver problemas mais específicos que surgem nesse tipo de desenvolvimento, torna-se mais eficaz que a maioria das ferramentas de bate-papo utilizadas nesse contexto. Porém, para analisar a eficácia de uma ontologia é preciso analisar para qual área de desenvolvimento ela foi criada, e quais problemas ela resolve.

O presente artigo tem como objetivo analisar ontologias de domínio para DDS, mostrando seus pontos em comuns, disponibilidade e campo de atuação, a fim de possibilitar a produção de ontologias mais refinadas no futuro, com inferência de informações concisas, precisas e não-ambíguas, de forma semi-automática dentro do ambiente de desenvolvimento.

O restante do artigo está organizado como descrito a seguir. Na seção 2 é abordado o conceito de ontologias e suas estruturas, sendo também apresentado o ambiente de desenvolvimento distribuído de software. Na seção 3 são mostradas ontologias de domínio para DDS. A seção 4 apresenta a análise das ontologias para DDS, bem como a comparação de como essas ontologias agem no processo de desenvolvimento de software. Por fim, a seção 5 apresenta as considerações finais, evidenciando as contribuições do trabalho e a perspectiva de trabalhos futuros.

2. Referencial Teórico

2.1. Ontologias

Diversas definições são dadas a fim de descrever o significado computacional para ontologias. A mais popular é: "uma especificação formal e explícita de uma conceitualização compartilhada" definição dada por [Gruber 1995], onde ser formal implica em declarativamente definida e compreensível para agentes inteligentes de software; explícita significa que os elementos e suas restrições estão claramente definidos; conceitualização trata de um modelo abstrato de uma área de conhecimento ou de um universo limitado de discurso, compartilhada, indica um conhecimento consensual, seja uma terminologia comum da área modelada.

Assim, em um nível de abstração mais alto, ontologias estabelecem uma terminologia comum e não ambígua para os diversos domínios de conhecimento. Segundo [Guizzard 2000], uma ontologia é composta de conceito, relação, função, axiomas e instâncias. Em resumo, conceito pode ser "qualquer coisa" a respeito de



"algo" que será explicado. Já o tipo de interação entre conceitos de um domínio e seus atributos é chamado de relação, cujo tipo denomina-se de função.

Os axiomas modelam sentenças que são sempre verdadeiras. Por fim, as instâncias representam elementos do domínio associados a conceitos específicos.

O uso de ontologias tem se popularizado através de diversas outras subáreas da Ciência da Computação, tais como: Engenharia de Software, Inteligência Artificial, Banco de Dados e Sistemas de Informação. Um dos principais responsáveis por esse fenômeno é o criador da Web Semântica, [Berners-Lee 2001].

São várias as motivações para o desenvolvimento de uma ontologia, [Noy e McGuinness 2001] e [Freitas 2003].

- Compartilhar entendimento comum da estrutura de informação entre pessoas ou entre agentes inteligentes de software;
- Reuso de conhecimento de um domínio. Caso exista uma ontologia que modele adequadamente certo conhecimento de um domínio, ela pode ser compartilhada e usada por engenheiros e desenvolvedores de ontologias, bem como, por equipes que desenvolvem aplicações semânticas e cognitivas;
- Tornar explicitas pressuposições de um domínio. As ontologias fornecem um vocabulário para representação de conhecimento e seu uso evita interpretações ambíguas;
- Possibilidade de tradução entre diversas linguagens e formalismos de representação do conhecimento. A tradução concretiza um ideal perseguido por gerações de pesquisadores de Inteligência Artificial. Ela facilita o reuso de conhecimento, e pode vir a permitir comunicação entre agentes em formalismos diferentes, uma vez que este serviço encontra-se disponível em um número cada vez maior de formalismos de representação do conhecimento. Outra forma de alcançar esse intento são os editores de ontologias em que se pode escolher em que linguagem de representação será escrito o código gerado.
- Mapeamento entre formalismos de representação de conhecimento que, inspirado no componente de conectividade para sistemas gerenciadores de bancos de dados ODBC (Open Database Connectivity), liga dois formalismos, criando uma interface interoperável de acesso comum para eles, permitindo a um agente acessar o conhecimento de outro agente.

2.2. Desenvolvimento Distribuído de Software

O desenvolvimento de software de forma co-localizada tem se tornado cada vez mais custoso e menos competitivo para as organizações. Visando a redução de custos, melhoria da qualidade, aumento de produtividade e competitividade global, várias empresas optam por distribuir seus processos de desenvolvimento em lugares diferentes [Audy e Prikladnicki 2008]. Sendo assim, surge o Desenvolvimento Distribuído de Software (DDS), onde os envolvidos em um determinado projeto estão dispersos geograficamente.

De acordo com [Prikladnicki 2003], as vantagens que o DDS oferece são:

• Possibilidade de desenvolvimento *follow-the-sun*, que permite o aumento de produtividade e a redução dos prazos de entrega dos produtos;



- Disponibilidade de recursos globais com baixos custos e a qualquer hora;
- Disponibilidade de recursos qualificados em áreas especializadas;
- Possibilidade de formação de equipes virtuais.

Entretanto, existe uma série de desafios inerentes a este ambiente de desenvolvimento. De acordo com [Lopes 2004], os principais desafios e questões envolvidas neste ambiente são: (1) diferenças culturais; (2) dispersão geográfica; (3) coordenação e controle; (4) comunicação; e (5) espírito de equipe.

3. Ontologias de Domínio para o DDS

Ontologias de Domínio tratam uma área específica de atuação, partindo de um ponto mais genérico [Semprebom, Camada and Mendonça 2007], e possuem grande conteúdo conceitual e relações de informação dentro do domínio. Uma vez que os usuários têm papéis bem definidos em um projeto de desenvolvimento de software, torna-se muito mais fácil colher informações sobre atividades e problemas propostos pelos próprios usuários, informações estas que enriquecem o conteúdo da ontologia. Utilizar-se desse ambiente para desenvolvimento distribuído de software traz grandes vantagens como, por exemplo, informações consistentes formuladas por especialistas, dentro de seu campo de atuação.

Nesse contexto, esse trabalho utiliza como base um mapeamento sistemático realizado por [Borges 2013], onde foram identificadas 4(quatro) ontologias de domínio para o contexto de DDS. Resumidas na Tabela 1 a seguir.

Ontologias Descrição Ontologia que descreve conceitos de componentes Componentes de de software, como suas descrições, operações, Software atributos, restrições e relacionamentos. Ontologia que descreve conceitos relacionados com **OFFLOSC** atores, artefatos, atividades e recursos de comunidades de desenvolvimento Open Source. Ontologia que descreve conceitos e relacionamentos de atores, regras, atividades, Comunidades Open processos, artefatos e ferramentas de comunidades Source de projetos Open Source. Ontologia que descreve conceitos e elementos que são representados e compartilhados em um ONTODISEN ambiente de DDS, como usuários, ferramentas, ambientes, atividades e processos.

Tabela 1. Ontologias de Domínio para o DDS

Tais ontologias foram desenvolvidas especificamente para equipes distribuídas, e portanto, possuem conceitos e características para esse contexto. É possível observar que muitos outros trabalhos utilizam ontologias para resolver ou mitigar desafios em ambientes distribuídos, mas que não são especificas para esse ambiente, estando voltadas para a Engenharia de Software em geral.



4. Avaliação de Ontologias para o contexto distribuído

Esta seção apresenta uma breve análise das ontologias de domínio para DDS resumidas na Tabela 1.

A primeira trata do OntoDiSENv1 [Chaves, Leal and Steinmacher 2011], que na verdade é uma evolução do OntoDiSEN [Chaves, Huzita, Leal and Steinmacher 2010] para ambiente DiSEN. O OntoDiSEN é um modelo baseado em contexto para obtenção de informação em um domínio, que trata e seleciona as informações colhidas de acordo com o domínio da aplicação; a definição de aplicação, isto é, o ambiente ao qual se refere; ao seu objetivo, que tem como foco estruturar o contexto de interação dos membros; e quanto aos usuários, que podem ir desde a própria ontologia até usuários do ambiente. Para tais finalidades a ontologia apresenta então um classe principal de gerenciamento de contextos, denominada *Context*.

A evolução para o OntoDiSENv1 deu-se na questão da obtenção e classificação de informações sobre os membros do ambiente, no qual apoiado no modelo DiSENCSE (DiSEN-Context Sensitive Environment) [Chaves, Huzita, Leal and Steinmacher 2010], foram introduzidas mais duas superclasses para inferência de informações, action mapping e amended context registration. Este modelo visa fazer com que todos os membros do domínio fiquem cientes da influência desta sobre ações de outros usuários e influências que tais ações têm o domínio, de forma semiautomática. Assim a ontologia ficou dividida em três superclasses, Context, classe principal do OntoDiSEN; Actions e AmendedContext.

Outra ontologia bem semelhante ao OntoDiSENv1 é OFLOSSC, que na questão de desenvolvimento distribuído e com base no FLOSS [Mirbel 2009] foca em comunidades de desenvolvimento de código aberto. As duas se assemelham bastante uma vez que apresentam captura de informações e obtenção de conhecimento. Apesar disso, OFLOSSC é constituído de ontologias presentes no Dhruv [Mirbel 2009], OSDO [Mirbel 2009] e SIOC[Mirbel 2009], e além disso apresenta um modelo de cooperação (O'CoP [Mirbel 2009]), com foco para as comunidades web, sua classificação de informações dar-se pela interação entre usuários, ou atividades, enquanto o OntoDiSENv1 trata de ações tomadas de acordo a problemas propostos, analisando a área de atuação o OFLOSSC captura as informações pela interação entre membros de acordo a seus papéis nos projetos.

A terceira ontologia analisada foi a TeamWeaver [Maalej and Happel 2008] que é voltada para desenvolvimento de software em geral. Possui ferramentas para organizar as informações e facilitar o acesso e entendimento ás mesmas. No quesito de correção de *bugs*, fornece ajuda ao mostrar *bugs* (erros) relacionados e as soluções que foram tomadas nesses casos, auxiliando assim, na resolução dos *bugs* que eventualmente aparecerem. Utiliza a experiência e ações de cada membro para analisar as áreas em que ele tem conhecimento e selecionar membros mais aptos a solucionarem problemas em cada situação em que for necessário, a experiência de um membro é formada pelos projetos que ele já trabalhou nessa ontologia, sua área de atuação, problemas que foram resolvidos por ele e a área de cada um deles, e outras ações realizadas no ambiente da ontologia. Em cada solicitação seleciona o membro mais habilitado para realizar as atividades requeridas. A ontologia já foi desenvolvida e está disponível.



Finalizando com a ontologia OSDO [Simmons and Dillon 2008], esta é voltada para desenvolvimento de software *Open Source*. Seu principal objetivo é organizar a grande quantidade de informações geradas num projeto *open source*, possui um banco de dados para todos os artefatos que forem gerados no projeto. As informações são guardadas de forma categórica ajudando no entendimento e no acesso. Também há classificação dos membros de uma equipe, organizando assim, o que cada um está habilitado a fazer ou não.

Na Tabela 2 são apresentadas algumas das características identificadas nas ontologias, como por exemplo, se ela é destinada ao desenvolvimento *open source*, a criação de perfil para membros através da experiência destes, se era uma ferramenta dedicada à correção de *bugs* e se as mesmas também se destinam a Engenharia de Software tradicional.

Características	OntoDisenv1	OFLLOSC	TeamWeaver	OSDO
Dedicada a Desenvolvimento Open Source	X	✓	x	✓
Cria perfil do membro através da experiência	✓	✓	✓	✓
Ferramenta dedicada a correção de bugs	X	✓	✓	X
Suporte ao ES tradicional	✓	√	√	✓

Tabela 2. Características das Ontologias

É importante ressaltar que a avaliação e análise das ontologias foi comprometida pelo fato de que apenas uma das ontologias, a TeamWeaver estava disponível, as outras não foram encontradas por buscas no google, por buscas em repositórios específicos de ontologias e por busca nos próprios links citados nos artigos das ontologias, sendo estes endereços inexistentes.

5. Considerações Finais

No contexto da globalização, a distribuição dos processos de desenvolvimento de software tornou-se um fato cada vez mais comum. O trabalho em ambientes de DDS é muito complexo e ainda não existem práticas maduras para esse contexto. Neste sentido, a utilização de ontologias pode trazer benefícios, como compreensão compartilhada das informações, facilidade de comunicação entre equipes distribuídas e eficácia no gerenciamento das informações.

Dessa forma, esse trabalho analisou as ontologias de domínio para DDS, comparando seus métodos de tratamento de informações obtidas em ambientes de produção de softwares, propiciando futuras evoluções das ontologias estudadas ou criação de novas mais especificamente para DDS, uma vez que as ontologias atuais não



dão completo suporte a esse tipo de desenvolvimento. Os métodos utilizados foram a pesquisa de disponibilidade da ontologia, as características das ontologias e as funcionalidades do ambiente ao qual as ontologias se aplicam.

A pesquisa mostrou que as ontologias abordadas nos artigos analisados não suportam por completo o Desenvolvimento Distribuído de Software, no aspecto de inferências de informações, interferência e influência nas ações dos membros dos ambientes nos quais as ontologias estão presentes, pois cada uma se mostra eficiente em uma determinada área do domínio, mas não como um todo. Por isso a evolução ou criação de ontologias para suporte complete a DDS é de fundamental importância para a comunidade de DDS bem como a Engenharia de Software no geral.

Como trabalhos futuros, realizar análise da ontologia DKDOnto [Azevêdo and Costa 2013] que está na fase final de desenvolvimento, e é um domínio específico de ontologia para projetos DDS. Utiliza uma ferramenta chamada DKDs, que auxilia na transmissão, geração e distribuição do conhecimento. É uma ferramenta que apoia a tomada de decisão em DDS. O sistema acessa sua base de conhecimentos, e, baseado nas experiências (projetos distribuídos, suas configurações, desafios enfrentados e soluções utilizadas) o sistema sugere possíveis soluções ao usuário para superar os desafios.

Por fim, também pretende-se tentar estabelecer o contato com os pesquisadores responsáveis pelas ontologias analisadas a fim de aprofundar a análise sendo possível detalhar melhor características e funcionalidades de ambiente e realizar uma análise através de alguma ferramenta própria para validação e avaliação de ontologias.

Referências

- Audy, J. and Prikladnicki, R. (2008) Desenvolvimento Distribuído de Software, Editora Elsevier.
- Azevêdo, Ryan; Costa, Catarina; Duarte, Marcos; Fechine, João Paulo; Freitas, Fred; Meira, Silvio;Rocha, Rodrigo G. C.. (2013). An Ontology-based System to Support Distributed Software Development. In The Eighth International Conference on
- Software Engineering Advances (ICSEA) 2013. October 27 November 1, 2013 Venice, Italy
- Borges, A., Rocha, R., Costa, C., Tomaz H., Soares S., Meira S. (2013) "Ontologies Supporting the Distributed Software Development: a Systematic Mapping Study". In Proceedings of the International Conference on Evaluation & Assessment in Software Engineering (EASE). Porto de Galinhas, PE, Brasil.
- Carmel, E. (1999) Global Software Teams: Collaboration Across Borders and Time Zones. Prentice-Hall, EUA.
- Chaves, Ana Paula; Huzita, Elisa H. M.; Leal, Gislaine; Steinmacher, Igor. "OntoDiSEN: uma ontologia para apoiar o desenvolvimento distribuído de software" (2010). Faculdade Integrado de Campo Mourão; Universidade Estadual de Maringá; Universidade Tecnológica Federal do Paraná.
- Chaves, Ana Paula; Steinmacher, Igor. Leal, Gislaine Camila Lapasini. "OntoDiSENv1: an Ontology to Support Global Software", (2011). Federal Technological University



- of Paraná Coordination of Technology Systems for Internet; State University of Maringá Departament of Production Engineering.
- Freitas, F. (2003). Ontologias e a web semântica. In: Renata Vieira; Fernando Osório. (Org.). Anais do XXIII Congresso da Sociedade Brasileira de Computação Campinas: SBC, 2003. v. 8, p. 1-52.
- G. Simmons & T.S Dillon. "Semantic Web Support For Open Source Software Development" (2008). Digital Ecosyst. & Bus. Intell. Inst., Curtin Univ. of Technol., WA, Australia.
- Gruber, T. R. (1995). "Toward Principles for the Design of Ontologies used for Knowledge Sharing". In formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers.
- Guizzardi. G. (2000). "Uma abordagem metodológica de desenvolvimento para e com reuso, baseada em ontologias formais de domínio." Dissertação de Mestrado. Universidade Federal do Espírito Santo.
- J. H. Tim Berners-Lee and O. Lassila,. (2001). "The semantic web," Scientific American Magazine.
- Komi-Sirvo, S; Tihinen M. (2005). Lessons Learned by Participants of Distributed
- Software Development. Journal Knowledge and Process Management, vol. 12 n° 2 p. 108–122.
- Lopes, L. T. (2004) "Um Modelo de Processo de Engenharia de Requisitos para Ambientes de Desenvolvimento Distribuído de Software", Dissertação de Mestrado em Ciências da Computação, Pontificia Universidade Católica do Rio Grande do Sul, Porto Alegre.
- Maalej, W. & Happel, Hans-Jörg "A Lightweight Approach for Knowledge Sharing in Distributed Software Teams" (2008). Technische Universität München, Munich, Germany. FZI Research Center for Information Technologies, Karlsruhe, Germany.
- Mirbel, Isabelle. "Oflosse, An Ontology For Supporting Open Source Development
- Communities" (2009). INRIA Sophia Antipolis, 2004 route des lucioles BP 93, FR06902 Sophia Antipolis, Cedex France Laboratoire I3S, Route des Lucioles, BP 121, FR-06903 Sophia Antipolis, Cedex France.
- N. F. Noy and D. L. Mcguiness. (2001.). "Ontology development 101: A guide to creating your first ontology.". Online, 2001. [Online]. Available:
- http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noymcguinness.html
- Prikladnicki, R. (2003) "MuNDDoS: Um Modelo de Referência para Desenvolvimento Distribuído de Software", Dissertação de Mestrado, Pontificia Universidade Católica do Rio Grande do Sul, Brasil, 2003.
- Semprebom, Tiago; Camada, Marcos Yuzuru; Mendonça, Igor. "ONTOLOGIAS E PROTÉGÉ" (2007). Universidade Federal de Santa Catarina (UFSC) Florianópolis, SC, Brazil.



Wongthongtham, P., Chang, E., Dillon, T. S., Sommerville, I. (2006). Ontology-based Multi-site Software Development Methodology and Tools. J. of Systems Architecture. 640–653.



Análise da Gerência de Configuração de uma Empresa de TI com Base no ITIL

Thiago Henrique de Almeida Espinhara¹

¹Universidade Federal Rural de Pernambuco (UFRPE) Av. Bom Pastor – Boa vista – Garanhuns-PE

thiagohe@gmail.com

Abstract. This article seeks to conduct an analysis of Configuration Management from a company of Information Technology (IT), using as a basis the best practices described in ITIL. The Configuration Management process is considered as the basis for other ITIL processes, being crucial for generating relevant information through a database. However, some companies working with IT, often do not give importance to this management, and sometimes precarious mechanisms to accomplish this task. The objective of this work is to understand how this process happens currently in the company and bring those aspects that could be improved, according to ITIL.

Resumo. Este artigo busca realizar uma análise da Gerência de Configuração de uma empresa de Tecnologia da Informação (TI), utilizando como base as boas práticas descritas no ITIL. A Gerência de Configuração é considerada como processo base para outros processos do ITIL, sendo de fundamental importância para gerar informações relevantes através de um Banco de Dados. Contudo, algumas empresas que trabalham com TI, muitas vezes não dão importância a este gerenciamento, tendo por vezes mecanismos precários para realizar esta tarefa. O objetivo deste trabalho é entender como este processo acontece atualmente na empresa e trazer os aspectos que poderiam ser melhorados, de acordo com ITIL.

1. Introdução

Por bastante tempo foi possível para algumas empresas dar continuidade ao seu negócio sem ou com muito pouco apoio de Tecnologias da Informação. Nos dias atuais, com o avanço tecnológico, dificilmente encontram-se empresas que não utilizem TI. Desta forma, a Tecnologia da Informação passa a ser um fator crucial para o sucesso dessas organizações, se tornando em alguns casos o seu diferencial. Seria inimaginável atualmente, ramos de negócio existirem sem o apoio de TI, como por exemplo, o sistema bancário. Pois o apoio da TI traz benefícios competitivos para as organizações, o que proporciona um diferencial.

Para muitas empresas, a TI hoje se tornou um parceiro estratégico, integrando-se a ela e fazendo parte do negócio. Atualmente a área de TI não é tratada de maneira isolada dentro de uma organização, sendo suas decisões sobre investimentos tratadas em reuniões de planejamento estratégico da empresa. Desta forma, a TI deixou de ser tratada exclusivamente por técnicos da área, passando a ser incorporada na estratégia da empresa para alcançar os seus objetivos.



Com o aumento do peso de importância dentro da organização, a TI passou a ter alguns desafios, tais como o alinhamento dos serviços de TI com as necessidades da organização, ambientes de TI cada vez mais complexos, dependência de TI para o negócio, redução de riscos, etc. Desta forma, a ITIL foi criada com o objetivo de se reunir as melhores práticas para os processos em Tecnologia da Informação.

A ITIL, abreviação do inglês para Information Technology Infrastructure Library ou Biblioteca de Infraestrutura para Tecnologia da Informação, é um conjunto de boas práticas de gerenciamento de serviços de TI (ITSM), que se concentra em alinhar os serviços de TI com as necessidades das empresas. Na sua forma atual (conhecido como ITIL edição de 2011), é descrito em uma série de cinco publicações principais, cada uma das quais cobre uma fase do ciclo de vida ITSM. A ITIL sustenta a ISO / IEC 20.000 (anteriormente BS15000), a Norma Internacional de Gestão de Serviços para a gestão de serviços de TI, embora as diferenças entre as duas estruturas existam (ITIL, 2010).

A ITIL busca promover a gestão com foco no cliente e na qualidade dos serviços de Tecnologia da Informação. A ITIL lida com estruturas de processos para a gestão de uma organização de TI apresentando um conjunto abrangente de processos e procedimentos gerenciais, organizados em disciplinas, com os quais uma organização pode fazer sua gestão tática e operacional em vista de alcançar o alinhamento estratégico com os negócios (HANNA, RUDD, MACFARLANE, WINDEBANK e RANCE, 2007). Este trabalho busca analisar de que maneira os processos são feitos atualmente na empresa, referentes a gerência de configuração do ITIL, identificando possíveis problemas, falhas e deficiências de gestão dos Itens de Configuração (IC's) de TI da empresa.

2. A ITIL

A ITIL foi desenvolvida inicialmente pela CCTA (Central Computing and Telecommunications Agency) atual OGC (Office of Government Commerce). O OGC é órgão do Governo britânico que tem como objetivo desenvolver metodologias e criar padrões dentro dos departamentos do governo britânico, buscando aperfeiçoar e melhorar os processos internos. A biblioteca da ITIL foi desenvolvida pela CCTA, e tinha como objetivo melhorar os processos dos departamentos de TI do governo britânico. Desde o seu surgimento em 1980, as empresas e outras entidades do governo perceberam que as práticas sugeridas poderiam ser aplicadas em seus processos de TI também. Em 1990 a ITIL acabou se tornando um padrão de fato em todo o mundo, e a partir dela houve várias adaptações de outros fornecedores, como a Microsoft, IBM e HP (ITIL, 2010).

A ITIL atualmente desperta grande interesse no mercado, pois há uma preocupação com o Gerenciamento de Serviços de TI nas empresas. Como citado anteriormente, a grande dependência da TI para os negócios faz com que os gestores de TI busquem a adoção das melhores práticas com o objetivo de trazer resultados positivos, como redução de custos e agilidade em seus processos.

A ITIL oferece um framework comum para todas as atividades do departamento de TI, como a parte da provisão dos serviços, baseada na infraestrutura de TI. Estas atividades são divididas em processos, que fornecem um framework eficaz para um



futuro Gerenciamento de Serviços de TI aprimorado. Cada um destes processos cobre uma ou mais tarefas do departamento de TI, tais como desenvolvimento de serviços, gerenciamento da infraestrutura, fornecimento de serviços e suporte a serviços (EXAMES, 2013).

3. Gerência de Configuração

Este trabalho visa estudar exclusivamente a parte do Gerenciamento de Configuração abordado pelo ITIL, buscando entender os procedimentos já utilizados em uma empresa de Tecnologia da Informação, que presta serviços de software e hardware, analisando se estes estariam de acordo com a proposta do ITIL. Em caso negativo, realizar o levantamento dos pontos em que poderiam haver melhorias.

O Processo de Gerenciamento de Configuração possibilita à equipe de TI obter um controle sobre os inúmeros componentes de infraestrutura de TI sob sua responsabilidade, os quais constituem os ativos da área. Estes ativos são chamados de Itens de Configuração (ICs). Segundo (BON, 2005), a Gerência de configuração pode ser considerada um processo pivô para todos os outros, sendo considerado o processo central, dando suporte aos demais e fornecendo informações sobre a infraestrutura de TI.

O Gerenciamento de Configuração tem por prioridade identificar e definir os componentes que fazem parte de um serviço de TI, registrar e informar o estado desses componentes e das solicitações de mudanças a eles associadas, verificando se os dados relacionados foram fornecidos e se estão corretos. Assim, a gerência de configuração proporciona uma base para o bom desempenho dos demais processos da ITIL (MAGALHÃES e PINHEIRO, 2007).

Para fornecer as informações a respeito da infraestrutura de TI, a Gerência de Configuração precisa manter um Banco de Dados, geralmente chamado de BDGC (Banco de Dados de Gerência de Configuração). A depender do tamanho da infraestrutura que está sendo trabalhada e do nível de detalhe que se pretende alcançar, inserir todos os dados dos IC's pode ser uma tarefa custosa, podendo se tornar um processo lento. A grande vantagem de investir na construção do BDGC é que depois de implantado, informações em abundância podem ser geradas, como relatórios úteis para o Gerenciamento de TI. A Figura 1 ilustra o BDGC como base para os outros processos da ITIL.





Figura 1. BDGC como base para outros processos da ITIL. (Fonte: IMPULSE WEAR BRASIL, 2013)

Segundo (HANNA, RUDD, MACFARLANE, WINDEBANK e RANCE, 2007), as atividades do processo de Gerenciamento de Configuração, são:

- 1. **Planejamento**. Definição dos objetivos, escopo, políticas, procedimentos e interação esperada com outros processos.
- 2. **Identificação**. Coletar todas as informações dos IC's. Para tal, cada IC deve ser etiquetado para propósitos de controle.
- 3. **Controle**. Procedimentos de controle devem existir para que as mudanças no BDCG sejam documentadas e realizadas apenas com autorização.
- 4. **Acompanhamento do Status**. Registra o estado atual e anteriores do IC, podendo assim existir um acompanhamento do IC.
- 5. **Verificação e Auditoria**. Através da auditoria é possível verificar se todos os IC's estão registrados corretamente. Auditorias precisam ser regulares.

4. Análises Realizadas na Empresa

4.1 Metodologia Adotada

Para realizar a pesquisa de campo deste artigo, foi utilizada a observação dos procedimentos realizados na empresa alvo de estudo, durante a realização de Estágio Curricular Obrigatório. Desta forma, o estagiário foi responsável por analisar os métodos utilizados, para posteriormente identificar em que nível se encontra a empresa em relação ao Gerenciamento de Configuração do ITIL. O principal método de colheita dos dados foi a observação pelo estagiário dos procedimentos adotados pela empresa. Sendo possível desta forma identificar os pontos fracos relacionados a gestão dos ativos de TI e posteriormente sugerindo as melhorias, de acordo com o ITIL.

4.2 Levantamento da Situação Atual

A empresa alvo deste estudo, de pequeno porte, está localizada na cidade do Recife, capital Pernambucana, mais especificamente no Porto Digital (aglomerado de empresas



da área de TI – havendo concorrência razoável). Esta empresa presta serviços de TI na área de automação comercial de bares e restaurantes, onde o produto principal da empresa é um sistema para o gerenciamento desses estabelecimentos. Possui cerca de 70 (setenta) clientes e o capital humano é formado por cinco funcionários, sendo dois graduados na área de computação, dois graduandos também na área de computação e um graduando na área de administração de empresas. O nome da empresa não é citado neste trabalho por questão ética. Entre os serviços prestados pela empresa, estão os aluguéis e suporte técnico a: Software, Impressoras fiscais, não fiscais, de cheques e código de barras, Leitores de código de barras, Balanças eletrônicas, Computadores, Displays, Estabilizadores, Leitores de cartão, Nobreaks, Terminais de consulta, Teclados para automação.

Dependendo da necessidade do estabelecimento, o contrato de serviços firmado entre a empresa e o cliente estabelece quais itens são necessários a ele, sendo a empresa responsável por implantar e dar suporte a todos os itens do sistema: os de hardware e software. O autor deste artigo esteve mais próximo do software comercializado pela empresa, prestando serviços de manutenção do código-fonte ao implementar melhorias e defeitos no mesmo.

De acordo com os critérios de avaliação de mudanças da empresa, são definidos dois tipos de mudança distintos no software: 1) Mudanças que buscam corrigir falhas e 2) Mudanças de adição ou melhoria de funcionalidades do sistema. Assim, uma solicitação de mudança no software por parte do cliente só é atendida de imediato, se ela se enquadra em mudanças que configuram falhas no sistema (1) e que afetem no mínimo 80% (oitenta por cento) dos clientes da empresa. Esta é uma regra interna da empresa e devidamente acordada no contrato de prestação de serviços.

Quando um cliente entra em contato com o suporte técnico da empresa solicitando uma modificação no software, o suporte técnico que o atende abre uma nova solicitação em uma planilha de solicitações de mudanças. Nesta planilha constam as informações referentes aos detalhes da mudança, incluindo uma nota de prioridade da solicitação, de 0 a 10, que é solicitada ao cliente para uso futuro na avaliação da mesma. Nota maior tem prioridade maior. É perguntado ao cliente o relato dos detalhes da mudança, que é inserida na planilha com o estado pendente.

Uma análise é realizada para determinar se as mudanças serão atendidas e a devida ordem de atendimento. Tal análise é feita pelo analista de sistemas da empresa. Quando uma solicitação foi julgada como importante para o cliente (nota entre 7 e 10), mas não se enquadra na política de mudanças da empresa, este analista então, estabelece um custo para aquela mudança. Este custo é calculado com base na complexidade da mudança, sendo proporcional a quantidade de requisitos que esta afeta no sistema (maior dependabilidade significa maior custo).

Uma atividade de rotina da função de desenvolvedor, com objetivo de realizar a manutenção do código-fonte, é consultar a planilha de solicitação de mudanças, que é compartilhada internamente na empresa através da rede de computadores, utilizando o Microsoft Excel para visualização e edição. Essa planilha traz outras informações adicionais, além das citadas anteriormente, necessárias para a realização de correções: ID, nome da funcionalidade, programador responsável (atribuído pelo gerente de projetos), descrição do problema e estado (pendente, em desenvolvimento, pronta).



Quando uma solicitação de mudança está atribuída a um desenvolvedor e tem o status pendente, significa que aquela mudança já foi julgada pelo analista nos critérios já citados anteriormente. O desenvolvedor sabe, então, que precisa propor uma solução.

Este então altera o estado na planilha para "em desenvolvimento" e a implementa de fato no código-fonte. Consumada e testada a mudança pelo desenvolvedor, o mesmo modifica o estado para "pronta". É função do analista de testes realizar todos os testes dos impactos dessas mudanças antes de disponibilizar uma nova versão do software para os clientes. Um esquema do processo de mudanças pode ser visto na Figura 3.

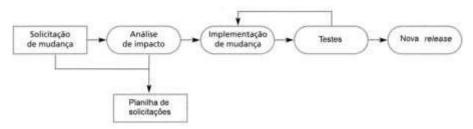


Figura 3. Representação do Modelo de Processo de Mudanças da Empresa.

4.3 Análises da Compatibilidade com o ITIL

Na empresa alvo do estudo, foram identificados alguns problemas que dizem respeito à gestão de TI em relação a forma com que alguns procedimentos eram realizados. Uma crítica que pode ser feita ao modelo apresentado na seção anterior, seria em relação à ferramenta de apoio para a execução do processo de mudança. A planilha, com ajuda do Microsoft Excel funciona relativamente bem e até cumpre os objetivos, com o número de pessoal que trabalha hoje. Mas, supondo que no futuro a empresa vá expandir os negócios e aumentar o número de funcionários, muito provavelmente este tipo de ferramenta se tornará ineficiente. Assim, a migração desse processo para uma nova ferramenta, que seguisse as boas práticas descritas no ITIL (citadas anteriormente), que conseguisse automatizar todo o processo seria ideal. De acordo com o referencial da Seção 4, este tipo de procedimento realizado atualmente causa deficiências no Controle, Acompanhamento do Status e Verificação.

Em relação a gerência de liberações, alguns problemas ocorrem com frequência, pois é difícil gerenciar de forma correta as versões dos softwares que os clientes possuem. Existe uma outra planilha que controla qual versão do sistema está atribuída a qual cliente. Porém, por vezes, erros humanos de atualização dos dados causam transtornos. De acordo com a Gerência de Configuração do ITIL, citado na Seção 4, é importante haver um controle central dos IC's para que as mudanças sejam consistentes e bem documentadas, além de realizadas somente com devida autorização. Desta forma, este tipo de procedimento não atende a estas boas práticas, onde, podem haver incompatibilidades no controle dos dados.

A Gerência de Configuração da ITIL diz que é importante possuir um Banco de Dados central onde se cadastrem todos os itens de configuração envolvidos com a TI. Assim, é possível sempre estar alimentando o banco de dados com todos os tipos de incidentes relacionados aos itens que a TI trabalha. Desta forma fica mais fácil de identificar problemas e resolvê-los definitivamente ou gerar indicadores que



possibilitem mensurar a qualidade do serviço que está sendo prestado. Porém, foi observado na empresa, no que diz respeito aos Itens de Configuração, que não há uma forma efetiva de identificar quantos equipamentos a empresa possui em cada cliente e muito menos um histórico de incidências do mesmo ou o seu estado atual. Desta forma, a implantação de um BDCG como citado na Seção 3, trariam informações muito importantes para a empresa, possibilitando ter um controle mais efetivo.

5. Considerações Finais

As boas práticas do ITIL citadas ao longo do trabalho e os ajustes aqui propostos para a empresa alvo do estudo, em relação a Gerência de Configuração, caso implantadas, trariam no mínimo maior qualidade das informações dos efetivos de TI, pois estas estariam bem estruturadas em um Banco de Dados onde informações em abundância poderiam gerar valor agregado a empresa. Desta forma, possibilitariam meios de mensurar o seu nível de gestão, além de trazer maior confiabilidade. Como se trata de uma pequena empresa, um planejamento seria necessário para uma migração gradual, definindo um escopo para saber como e de que forma tais modificações seriam implantadas, para que tais mudanças não fossem tão bruscas, fornecendo meios para o pessoal da empresa se adaptar ao novo paradigma de gestão, entendendo assim a sua importância.

Em relação a implantação de um Banco de Dados central para todos os Itens de Configuração, de acordo com a Seção 3.1, a parte que diz respeito ao Software poderia ser implantada primeiro, sendo assim apenas um pilar do Banco de Dados. Portanto, gradativamente outras áreas seriam contempladas: como a parte do Hardware (dos equipamentos que a empresa aluga junto com o sistema), de Documentação, de Redes, etc. Após a implantação de toda a Gerência de Configuração na empresa, através de alguma ferramenta (ferramentas livres estão disponíveis em abundância — trabalhos acadêmico), e ao se atingir um nível de maturidade (quando este novo paradigma fizer parte da cultura da empresa), ficaria como sugestão a implantação de outros tipos de gerências que a ITIL define, como os Gerenciamentos de Incidentes e Problemas, por exemplo, pois as informações de que elas necessitam dependem principalmente da Gerência de Configuração, que serve como base para o ITIL.

Referências

- BON, J. V. Foundations of IT Service Management, based on ITIL. Lunteren Holanda: Van Haren Publishing, 2005.
- BON, J. V.; CLIFFORD, D. Implementing ISO/IEC 20000 Certification: The Roadmap. ITSM Library. Van Haren Publishing. 2008. ISBN 90-8753-082-X.
- EXAMES. TI.exames. ITIL Information Technology Infrastructure Library. O que é ITIL? Disponível em http://www.tiexames.com.br/ITIL. Acessado em 7 jul. 2013.
- HANNA, A.; RUDD, C.; MACFARLANE, I.; WINDEBANK, J.; RANCE, S. An Introductory Overview of ITIL V3. 2007. ISBN 0-9551245-8-1.
- IMPUSE WEAR BRASIL. Consultoria em Processos ITIL Gerenciamento de



- Configuração. Disponível em http://www.impulsewear.com.br/consultoriaitil/gerconfiguracao.html>. Acessado em 07 jul. de 2013.
- IT PROCESS WIKI. IT service management. Disponível em <wiki.en.itprocessmaps.com>. Acessado em 8 jul. 2013.
- ITIL, Official Site. OGC withdrawal of ITIL version 2. Disponível em
- http://www.itil-
- fficialsite.com/News/ITILV2QualificationsWithdrawalstarts30June2010.aspx>. Acessado em 4 jun. 2013.
- MAGALHÃES, I.; PINHEIRO, W. Gerenciamento de Serviços de TI na Prática Uma abordagem com base no ITIL. 2007. ISBN: 978-85-7522-106-8.
- QUINTELLA, H.; CÔRTES, R. Estudo Comparativo da Compatibilidade entre IRM e ITIL na Gestão de Ativos de TI (Estudo de Caso). Universidade Federal Fluminense. 2005.

ISSN: 2317-5346



NVDA: Avaliação de Usabilidade da interação de usuários com deficiência visual com a *Web*

Roseane Martins¹

¹Universidade de Pernambuco (UPE) Campus Garanhuns, CEP 55.294-902 – Garanhuns – PE – Brasil

rose.ane.oliveira@hotmail.com

Abstract. This paper discusses the methods and results of Usability Evaluation in the context of Web of the NVDA screen reader - software speech synthesizer, which allows visually impaired to use the computer. As an instrument for data collection was used two methods: an interface embossed (made with EVA) with the aim of presenting a web page to users through touch and a roadmap with the technique of Usability Evaluation-Communicability Test. With this evaluation sought to understand the degree of usability of the NVDA in the access the Web by people with visual impairments.

Keywords: NVDA, Usability Evaluation, Visual Impairment.

Resumo. Neste artigo serão discutidos os métodos e resultados da Avaliação de Usabilidade no contexto Web do leitor de tela NVDA - software sintetizador de voz, o qual permite ao deficiente visual utilizar o computador. Como instrumento de coleta de dados utilizou-se dois métodos: uma interface em alto relevo (confeccionada com E.V.A) com o objetivo de apresentar uma página web aos usuários através do tato e um roteiro com a técnica de Avaliação de Usabilidade Teste de Comunicabilidade. Com essa avaliação procurou-se compreender qual o grau de Usabilidade do NVDA no acesso a Web por pessoas com deficiência visual.

Palavras-chave: NVDA, Avaliação de Usabilidade, Deficiência visual.

1. Introdução

Atualmente vivemos em uma sociedade que está em constantes transformações e a intensidade das mudanças tecnológicas se aceleram cada vez mais. Um fator determinante para esse avanço é o acesso e utilização das Tecnologias de Informação e Comunicação (TIC). Nesse panorama, pessoas com necessidades especiais encontram diversas barreiras que limitam ou impedem sua interação com o mundo digital.

De acordo com o censo demográfico do IBGE de 2010, cerca de um quarto da população brasileira tem pelo menos um tipo de deficiência, seja ela visual, auditiva, motora ou intelectual, totalizando cerca de 45 milhões de pessoas. O número equivale a 24% dos 190 milhões de habitantes do País. A deficiência visual foi a mais citada, detectou-se que 18,8% dos brasileiros têm dificuldade para enxergar ou possui cegueira. Diante de dados tão significativos, surge à necessidade de neutralizar as barreiras causadas pela deficiência visual no acesso a Web.



O acesso aos recursos tecnológicos influencia decisivamente nos processos de formação do conhecimento, portanto a Web deve ser flexível o suficiente para atender às necessidades e preferências do maior número possível de pessoas, além de ser compatível com as Tecnologias Assistivas usadas por pessoas com necessidades especiais (DIAS, 2003). Tecnologias Assistivas referem-se a qualquer ferramenta ou recurso destinado a proporcionar à pessoa portadora de deficiência maior independência, qualidade de vida e mobilidade (FERREIRA; NUNES, 2011).

Nesse contexto, este artigo pretende responder a seguinte questão: como as Tecnologias Assistivas auxiliam o acesso a Web de pessoas com deficiência visual. Para isso, foi avaliada a Usabilidade do leitor de tela NVDA¹ visto que este *software* propõe ampliar as possibilidades de comunicação e autonomia pessoal da pessoa com deficiência visual, minimizando ou compensando as restrições procedentes da ausência da visão.

Este artigo está estruturado da seguinte forma: A Seção 2 apresenta a revisão da literatura com conceitos sobre Usabilidade e Acessibilidade, visão geral sobre Tecnologias Assistivas para deficientes visuais. A Seção 3 contém a metodologia utilizada para a Avaliação de Usabilidade do NVDA. A Seção 4 apresenta os resultados da análise dos dados coletados. Por fim, a Seção 5 descreve as considerações finais e contribuições da pesquisa.

2. Revisão da Literatura

O diálogo entre o usuário e o programa se estabelece por meio da interface, um dos principais objetivos no desenho de interfaces é fazê-las amigáveis, ou seja, que não apresentem dificuldades ao usuário e assim estimulem a utilizá-las (FERREIRA; NUNES, 2011). Atualmente as pessoas com deficiência visual encontram barreiras de acessibilidade que dificultam ou mesmo tornam impossível o acesso a conteúdos na Web. A Acessibilidade na Web é a possibilidade de qualquer pessoa que apresente alguma deficiência ou necessidade especial a ter acesso aos seus recursos (FERREIRA; NUNES, 2011).

Nesse cenário, a Web tem um papel de ser facilitadora do acesso à informação, difusora de conhecimento e responsável também pela minimização de barreiras físicas e virtuais, incluindo, especialmente, pessoas portadoras de deficiência visual (SCATOLIM, 2009). Uma Web acessível e que permita a participação de pessoas portadoras de necessidades especiais na sociedade é fundamental para proporcionar oportunidades iguais para todos nas diversas áreas (W3C, 2005). Quando se trata de usuários com deficiência visual, torna-se fundamental identificar qual sua percepção com relação ao sistema e quais são as imposições e limitações a que estão sujeitos de acordo com seus modelos conceituais estabelecidos. O modelo conceitual possibilita aos usuários a rapidamente aprender a utilizar o sistema e o utilizar eficientemente (PREECE; ROGER; SHARP, 2002).

De maneira geral, interação de usuários com deficiência visual com o computador ocorre através dos leitores de tela. Os leitores de tela, em especial, o

¹ Non Visual Desktop Access, em português: Acesso Não-Visual ao Ambiente de Trabalho. Download disponível em: http://www.nvda-project.org/wiki/Download



NVDA permite aos usuários navegar pelo conteúdo da Web de duas maneiras: o usuário pode simplesmente deixar que o leitor de tela leia toda tela de cima para baixo, uma linha de cada vez, ou o usuário pode usar a tecla TAB para navegar entre os *links*. As escolhas aleatórias do usuário só funcionam se o sistema tiver sido previamente programado, determinando as possíveis ligações e tornando-as explícitas (PADOVANI; MOURA, 2008), portanto, para a avaliação da adequação, eficácia e qualidade dos sistemas é utilizada uma avaliação de Usabilidade. A avaliação de usabilidade caracteriza-se por utilizar diferentes técnicas voltadas em sua maioria para a avaliação da ergonomia dos sistemas interativos. Para esta pesquisa utilizou-se com instrumento de avaliação o método de Avaliação de Usabilidade Teste de Comunicabilidade.

Para Souza (1993), o Teste de Comunicabilidade de um software tem como objetivo avaliar a sua interface com relação à sua propriedade de comunicabilidade, ou seja, visa à avaliação de sua interface quanto à possibilidade que ela tem de conversar com o usuário. Conforme NETTO (2010), a aplicação do Teste de Comunicabilidade pode ser dividida em duas etapas: a coleta de dados e a análise destes dados.

Para a coleta de dados os passos a serem realizados são: a) Solicitar ao usuário a execução de uma tarefa pré-determinada no sistema, b) Gravar a interação do usuário com o sistema (anotações do aplicador do teste e gravação em vídeo podem ser feitas para enriquecer os dados durante o processo), c) Uma vez coletados os dados, passa-se para sua análise e d) Tabular as informações obtidas de acordo com as interjeições (conjunto de questionamentos que o usuário pode usar para se exprimir em uma situação onde acontece uma ruptura na sua comunicação com o sistema) e com os problemas de Usabilidade e Acessibilidade associados a elas, obtendo então um mapa dos pontos críticos da interação e um perfil da interação do sistema. A Tabela 1 apresenta as etiquetas (interjeições de comunicabilidade) que foram utilizadas para identificar rupturas de comunicação na execução das tarefas.

Tabela 1. Conjunto de interjeições de acordo com seu significado e situação Fonte: Adaptado de Netto, 2010, p. 200-201.

Interjeição de comunicabilidade	Significado	Ação
Cadê?	operação que deseja executar, mas não a encontra de imediato na interface.	inspecionando diversos
	O usuário não sabe o que fazer e procura descobrir qual é o seu próximo passo.	Inspeciona os menus de forma aleatória ou sequencial.
O que é isto?	Ocorre quando o usuário não sabe o que significa um elemento de interface.	O usuário procura um menu de ajuda ou equivalente.



Epa!	O usuário realizou uma ação indesejada e, percebendo imediatamente que isto ocorreu, desfaz a ação.	O usuário executa uma ação e, em seguida a desfaz.
Onde estou?	O usuário efetua operações que são apropriadas para outros contextos, mas não para o contexto.	3
Por que não funciona?	A ação executada não obtém o resultado esperado, no entanto o usuário não entende porque esse resultado não foi alcançado.	Repetição na ação.
O que houve?	O usuário não percebe ou não entende a resposta dada pelo sistema para a sua ação (ou o sistema não dá resposta alguma).	Repete a ação e busca uma alternativa para obter o resultado esperado.
Desisto!	O usuário não realizar a tarefa e desiste.	Interrupção da tarefa.
Não, obrigado.	O usuário entende a solução, mas prefere outras formas de interação.	A ação é seguida de uma ou mais formas alternativas para se alcançar um mesmo resultado.
Socorro!	O usuário não consegue realizar sua tarefa através da exploração da interface.	Recorre à explicação de outra pessoa.

3. Metodologia

A escolha dos métodos foi definida com a intenção de responder a questão desta pesquisa. As fases que compõem a pesquisa desde o planejamento até os resultados estão ilustradas na Figura 1.

Esta pesquisa foi desenvolvida durante um projeto acadêmico. Para a coleta de dados foi selecionada uma amostra de cinco pessoas com deficiência visual, número defendido por Nielsen (1990) como suficiente para encontrar 80% dos problemas usabilidade de uma interface. Como instrumento de coleta de dados utilizou-se dois métodos: uma interface em alto relevo (confeccionada com E.V.A²) e um roteiro com a técnica de Avaliação de Usabilidade Teste de Comunicabilidade para verificar a adequação das ferramentas de acesso a Web do NVDA.

² E.V.A (Etil Vinil Acetato) é um material emborrachado que tem sido utilizado de diversas maneiras nos acessórios domésticos, na decoração de festas infantis, em painéis nas escolas e em tudo mais que a imaginação permitir.





Figura 1. Etapas da Metodologia

Inicialmente, foi apresentado os principais cenários do NVDA através de uma interface em alto relevo (Figura 2) com o objetivo de apresentar o *layout* de uma página Web aos usuários através do tato. Nessa interface tátil, os usuários com deficiência visual puderam perceber a organização dos principais componentes da interface Web a ser utilizada tais como: botões, menus, barras de rolagem, disposição do conteúdo, caixa de busca, localização da logomarca e rodapé do site.



Figura 2. Usuário utilizando a interface em alto relevo confeccionada com E.V.A

Após essa etapa, foi realizada a coleta de dados através do Teste de Comunicabilidade. Para a coleta de dados os passos a serem realizados são: a) solicitar ao usuário a execução de uma tarefa pré-determinada no sistema, b) gravar a interação do usuário com o sistema (anotações do aplicador do teste e gravação em vídeo podem ser feitas para enriquecer os dados durante o processo), c) uma vez coletados os dados, passa-se para sua análise e d) tabular as informações obtidas de acordo com as interjeições (conjunto de questionamentos que o usuário pode usar para se exprimir em uma situação onde acontece uma ruptura na sua comunicação com o sistema) e com os problemas de Usabilidade associados a elas, obtendo então um mapa dos pontos críticos da interação e um perfil da interação do sistema.

Durante a execução das tarefas foi avaliada a facilidade ou a dificuldade que os usuários têm em encontrar uma determinada ação/comando (navegação), a facilidade ou dificuldade de entender os termos utilizados pelo programa e também a facilidade ou dificuldade sentida pelos usuários durante a realização de uma tarefa.



Coletados os dados durante o Teste de Comunicabilidade, passou-se para sua análise (categorização dos problemas de Usabilidade). As tarefas foram etiquetadas conforme o método de Avaliação da Comunicabilidade (NETTO, 2010). As etiquetas (ou interjeições de comunicabilidade) foram utilizadas para identificar problemas de interação na execução das tarefas.

4. Resultados

Nesta seção, descrevem-se os resultados obtidos com o Teste de Comunicabilidade fazendo-se uma análise dos procedimentos aplicados. O objetivo dessa análise é avaliar as principais características e o modo de funcionamento do NVDA com relação ao acesso à Web. A seguir, são apresentados os resultados referentes à utilização do NVDA pelos participantes da pesquisa.

4.1 Principais características do NVDA

A interação dos usuários com o leitor de tela NVDA ocorre através da comunicação entre o software e o sistema operacional *Windows*. Suas principais características incluem: Anúncio automático do texto onde o mouse estiver posicionado, habilidade para rodar a partir de um cabo USB ou qualquer mídia portátil sem a necessidade de instalação, suporte para *Adobe Reader* e *Prompt* de comandos do *Windows* e suporte para *Microsoft Outlook Express / Windows mail*.

4.2 Teste de Comunicabilidade

Esta fase teve como propósito avaliar o uso do NVDA no acesso ao serviço de email do Gmail onde foi pedido aos usuários com deficiência visual que realizassem quatro tarefas: a) acesso do usuário ao email (através da identificação do nome de usuário e senha), b) ler emails da caixa de entrada, c) enviar um email, d) enviar email com arquivo anexo. As tarefas foram etiquetadas conforme o método de Avaliação da Comunicabilidade (NETTO, 2010).

As etiquetas (ou interjeições de comunicabilidade) foram utilizadas para identificar problemas de interação na execução das tarefas. A Tabela 2 mostra os resultados obtidos com o Teste de Comunicabilidade utilizando o NVDA.

Tabela 2. Interjeições de comunicabilidade conforme frequência e problemas de interação com o NVDA

Interjeição de comunicabilidade	Número de ocorrências	Problema
Cadê?	3	Navegação
E agora?	5	Navegação
O que é isto?	3	Atribuição de significado



Epa!	3	Navegação/ Atribuição de significado
Onde Estou?	4	Navegação/ Atribuição de significado
Por que não funciona?	5	Atribuição de significado
O que houve?	3	Atribuição de significado
Não, obrigado.	3	Recusa de usar
Socorro!	4	Incompreensão de como usar

Como demonstra a Tabela 2, as interjeições "Cadê?" apontada por 3 usuários e "E agora?" apontada pelos 5 usuários apresentam falhas com relação à navegação devido a manipulação dos objetos da interface depender do uso constante da tecla TAB (navegação via teclado) para passagem de um conteúdo para outro, fazendo com que o usuário tenha que passar por muito botões e menus até encontrar a opção desejada. A interjeição "O que é isto?" expressada por 3 usuários, refere-se ao fato deles não conseguirem diferenciar o texto do email do conteúdo exposto por banners de publicidade presentes na caixa de entrada do email.

A dificuldade encontrada para a interjeição "O que é isto?" indagada por 3 usuários, está no fato do NVDA ler o texto do interior do objeto, sendo este constituído por nome, valor e descrição do link, botão ou menu, ou seja, as informações contidas na página não são exatamente obtidas pelo que aparece na tela, mas sim através do código por detrás dela e que a produziu. Essa falha de atribuição de significado ocorreu quando o usuário ao passar de um menu para outro o sintetizador de voz do NVDA anunciou a palavra "lista" que identifica que no menu selecionado há mais conteúdos a serem explorados, o mesmo fato ocorreu para os botões e links, gerando nos usuários confusão de entendimento do real conteúdo do objeto selecionado.

Com relação às interjeições "Epa!" apontada por 3 usuários e "O que houve?" apontada por 4 usuários foi observado o uso das teclas ENTER e ALT onde seu uso não era pertinente as tarefas solicitadas como uma tentativa de concluir ou sair de uma ação. Para a interjeição "Porque não funciona?" falha apontada pelos 5 usuários, foi observado que as páginas do Gmail, no momento desta pesquisa não apresentavam conteúdos acessíveis, o que impediu que o software interpretasse adequadamente o que era apresentado na tela.

A interjeição "Não, obrigado" verbalizada por 3 usuários, está relacionada a recusa dos usuários em utilizar uma determinada ação/comando devido as variações de uso de um mesmo recurso, por exemplo: para voltar uma página do site você pode pressionar simultaneamente as teclas ALT + seta para esquerda ou utilizar a tecla TAB até que o software leia a opção voltar e/ou avançar presente na barra de menu do navegador e teclar ENTER. Para a interjeição "Socorro!", 4 usuários relataram que não



compreendiam o que era falado pelo sintetizador de voz do NVDA, por esta falar muito rápido mesmo tendo sua velocidade reduzida dentre as opções existentes.

5. Considerações finais

Ao fim da avaliação, foi possível levantar dados importantes em relação à Usabilidade do leitor de tela NVDA. Constatou-se que suas ferramentas de acesso a Web falham nos quesitos de Usabilidade tais como: navegação, atribuição de significados e compreensão de como usar adequadamente o software, como foi demonstrado através da aplicação do Teste de Comunicabilidade. Espera-se com este artigo apresentar aos desenvolvedores quais elementos do NVDA necessitam de maiores cuidados e correções.

Este artigo tem a intenção ainda de promover estudos para o desenvolvimento e aperfeiçoamento do leitor de tela NVDA e de sites acessíveis para pessoas com deficiência visual proporcionando assim, novos estudos na área de Usabilidade, Acessibilidade na Web e Interação Humano-Computador (IHC).

Referências

- Dias, C. (2003), Usabilidade na Web: criando portais mais acessíveis. RJ: Alta Books.
- Ferreira, S. B. L. and Nunes, R. R. (2011), e-usabilidade. Rio de Janeiro RJ: LTC.
- IBGE. (2010), Censo demográfico 2010: Pessoas Deficientes, http://www.ibge.gov.br/home/estatistica/populacao/censo2010/resultados_preliminar es amostra/default resultados preliminares amostra.shtm.
- Netto, A. A. de O. (2010), IHC e a engenharia pedagógica. Florianópolis: Visual Books.
- NVDA. (2012) Manual NVDA: Leitor de Tela Guia do Usuário, http://acessibilidadelegal.com/33-manual-nvda.php.
- Padovani, S. and Moura, D. (2008), Navegação em Hipermídia: uma abordagem centrada no usuário. Rio de Janeiro RJ: Editora Ciência Moderna LTDA.
- Preecce, J. and Rogers, Y. and Sharp, H. (2005), Design de interação: além da interação homem-computador. BOOKMAN Companhia.
- Scatolim, R. L. (2009) A importância da acessibilidade como mediadora da informação na internet para os deficientes visuais. Intercom Sociedade Brasileira de Estudos Interdisciplinares da Comunicação, Curitiba, p.1-12.
- Souza, C.S. de. (1993). The Semiotic Engineering of User Interface Languages. International Journal of Man-Machine Studies 39. Cambridge: Academic Press, 1993.



TVD.Calc: Um Aplicativo Dirigido à Educação Econômica em Ambientes Televisivos

Arthur H. do Sacramento Barros¹, Rodrigo Cândido Borges¹

¹Departamento de Informática – Instituto Federal de Goiás (IFG) – Campus Inhumas Av. Universitária s/n - 75400-000 – Inhumas – GO – Brasil

{4rthurhenriqu3, rodrigocand}@gmail.com

Abstract. The Brazilian economy is among the largest in the world. Nevertheless, in order to control inner inflation, high interest rates are applied to the values of goods and services. The rising of the rates in general leverage the values of the final products, resulting in fees masked by much of the population. Given this scenario, this paper presents the development of a Java DTV interactive application, which encapsulates its functionality between financial calculations and concepts related to the variables used in these calculations. This application can be loaded on demand or even reside in television receivers, providing a basis for educational and informative about taxes, depreciation, and financing.

Resumo. A economia brasileira está entre as maiores do mundo. Apesar disto, visando o controle inflacionário interno, elevadíssimas taxas de juros são aplicadas aos valores de bens e serviços. A alta dos juros em geral alavanca os valores finais dos produtos, acarretando em taxas pagas de forma mascarada por grande parte da população. Mediante este cenário, esse artigo apresenta o desenvolvimento de um aplicativo interativo Java DTV, que encapsula entre suas funcionalidades, cálculos financeiros e conceitos relativos às variáveis utilizadas nestes cálculos. Tal aplicação pode ser carregada sobre demanda ou mesmo residir em receptores televisivos, oferecendo uma base educacional e informativa sobre juros, amortizações, câmbio e financiamentos.

1. Introdução

Segundo dados do Fundo Monetário Internacional (*International Monetary Fund* - IMF) [IMF 2013], o Brasil possui a sexta maior economia do mundo, representada pelo Produto Interno Bruto (PIB) de cada nação, ficando atrás apenas dos Estados Unidos, China, Japão, Alemanha, e França. Embora economicamente essa estatística seja significativa para o país, alguns setores ainda tendem a evoluir almejando notas de países desenvolvidos, a destacar pela inclusão social e aspectos como saúde, trabalho, lazer e educação.

Apesar do ritmo econômico brasileiro, o país possui uma das maiores cargas tributárias do mundo. Dentre os fatores para o alto índice estão as medidas taxativas para o controle inflacionário, onde sua elevação visa desacelerar a aquisição de bens, reduzindo o consumo extremista da população.

Mesmo após medidas taxativas, alguns cidadãos ocasionalmente não controlam seus gastos, podendo sofrer com a geração de seriadas dívidas. Considerando fatos, este

ISSN: 2317-5346



artigo apresenta a implementação de um aplicativo denominado TVD.Calc, que possibilita aos seus usuários consultar e realizar cálculos financeiros (juros simples e compostos, descontos simples e compostos, amortizações, financiamentos, taxa de câmbio e caderneta de poupança) de forma interativa. Para isso, serão empregadas tecnologias televisivas digitais: o meio de comunicação mais acessado pela massa populacional [IBGE 2011].

2. Impactos da Digitalização Televisiva

A TV Digital (TVD) consiste na transmissão digital dos sinais audiovisuais, possibilitando a otimização de imagens e sons, bem como outras melhorias: mobilidade e interatividade [Lemos 2010]. Dentre esses benefícios, o de maior impacto estrutural é a interatividade. O responsável por tornar possível a interação e execução de aplicativos na televisão é o *middleware* [Caporuscio, Raverdy and Issarny 2012], camada intermediária de software que tem a responsabilidade de gerenciar os aplicativos, abstraindo dos programadores e usuários da aplicação detalhes do hardware.

Os principais sistemas de transmissão digitais mundiais são: o padrão norteamericano conhecido como ATSC [ATSC 2013] utilizando *middleware DTV Application Software Environment* (DASE), o padrão europeu conhecido como DVB [DVB 2013] utilizando *middleware Multimedia Home Plataform* (MHP) e o padrão japonês conhecido como ISDB, utilizando *middleware Association of Radio Industries and Businesses* (ARIB). Os demais sistemas de transmissão digital existentes são evoluções e/ou adaptações desses padrões, como por exemplo, o padrão brasileiro (ISDB-TB).

O ISDB-TB é o padrão de radiodifusão de serviços multimídia desenvolvido no Japão, juntamente com as suas atualizações tecnológicas feitas pelo Brasil. Seus recursos são basicamente os mesmos do ISDB, porém com o *upgrade* realizado na codificação de vídeo, utilizando a codificação MPEG4, e áudio, utilizando a codificação AAC-HE (Figura 1).

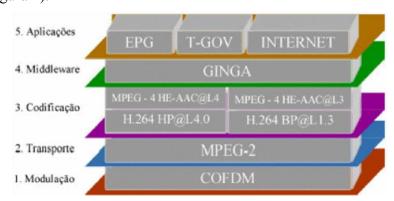


Figura 1. Pilha de Protocolos e Plataforma do ISDB-TB (Fonte: ginga.org.br)

O *middleware* do ISDB-TB é denominado Ginga [Ginga 2013]. As aplicações executadas sobre o Ginga podem ser escritas de duas formas: procedural e declarativa.

Na aplicação descrita neste trabalho, é utilizada a síntese procedural, especificamente a interface de programação Java TV [Oracle 2013], encapsulando cálculos oriundos da matemática financeira.

ISSN: 2317-5346



3. Trabalhos Relacionados

No Brasil, a maioria das emissoras de TV aberta já disponibiliza sua programação em sinal digital, sendo que algumas de suas transmissões também contêm aplicativos interativos [Cruz 2010]. As aplicações transmitidas pelas emissoras são desenvolvidas almejando fazer com que o telespectador não tenha a necessidade de sair do ambiente televisivo. Para tal caso, são desenhadas trazendo informações que as pessoas tendem a visualizar na internet, como por exemplo: resumo de novela, tabela do campeonato de futebol, basquete e etc.

Para exemplificação, a Rede Globo de televisão transmitiu um aplicativo na copa do mundo de 2012, em que os usuários poderiam ter acesso aos dados dos jogos, tabelas e resultados das partidas [Cruz 2010]. Outra aplicação, também transmitida pela emissora, permitia participação em enquetes e até a realização de compra de produtos.

Além das emissoras, outros órgãos governamentais também fazem a utilização de programas interativos na TV digital. Um desses órgãos é a Dataprev, que possui um software da Previdência Social que possibilita a realização de algumas ações feitas convencionalmente por telefone ou pessoalmente [Dataprev 2011].

Considerando o cenário educacional, podemos mencionar os seguintes aplicativos abertos para TV digital: Júri Virtual [Becker 2004], TV Escola [Silva et al. 2013] e a Market-TV [Sedrez 2008]. Assim como a proposta descrita neste trabalho, estas iniciativas também alcançam bases educacionais e informativas. Todavia, não implementam em seus contextos a divulgação e esclarecimentos sobre taxas adotadas sobre produtos, sintetizando fórmulas da engenharia econômica.

A próxima seção apresentará os principais fundamentos envolvidos no desenvolvimento do TVD.Calc, tema deste artigo. Inicialmente, discorreremos sobre a API Java DTV, empregada no desenvolvimento. Na sequência será exposto o Ginga-J, módulo procedural para execução do aplicativo e adotado pelo padrão brasileiro de TVD.

4. Java DTV e Ginga –J

Java é uma linguagem de programação orientada a objetos, desenvolvida pela Sun comprada pela Oracle Corporation, onde seus códigos são compilados para um código intermediário *(bytecode)*, o qual é executado e interpretado por uma máquina virtual nomeada por *Java Virtual Machine* (JVM).

Java DTV [Oracle 2013] é uma especificação para auxiliar no desenvolvimento de programas interativos em Java para TV digital que utilizam o *middleware* Ginga. Java DTV consiste na API Java DTV e na API Java TV acrescentadas à base comum dos componentes do *Java Runtime*, incluindo o *Connected Device Configuration*, o *Foundation Profile* e o *Personal Basis Profile* (ABNT NBR 15606-6).

O Ginga-J [Ginga 2013] foi desenvolvido para o processamento de aplicações imperativas realizadas em Java para TVD. Uma aplicação para TVD é um software executado no receptor televisivo. Aplicações Java, como é o caso deste trabalho, são denominadas Xlets, que são similares a outros programas Java, como os Applets. Uma Xlet deve ser implementado de modo a seguir um fluxo de execuções.



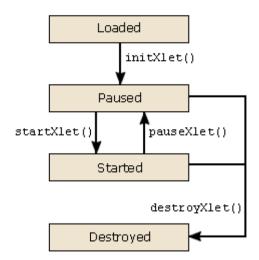


Figura 2. Diagrama com estados do ciclo de vida de um XIet (Fonte: ABNT NBR 15606)

Conforme mostrado na Figura 2, a interface Xlet dispõe de quatro métodos indispensáveis para seu funcionamento: "initXlet()", "startXlet()", "pauseXlet()" e "destroyXlet()".

O método *initXlet* é usado apenas quando o Gigna-J está carregando a aplicação. O estado "*Loaded*" tem como responsabilidade inicializar o aplicativo e colocá-lo em estado de pausa. A inicialização dos recursos utilizados é feita pelo *initXlet()*. A requisição deste construtor é dependente da implementação.

O método *startXlet* é chamado quando a aplicação é carregada sem qualquer tipo de erro. Neste caso, o Ginga-J a coloca em estado "*Started*", iniciando sua execução. O método *pauseXlet* é carregado quando o *middleware* coloca o programa em estado "*Paused*". Quando isso ocorre, os aplicativos diminuem a utilização de recursos, objetivando aumentar sua sobrevivência na memória. Por fim, o método *destroyXlet* é disposto quando o gerenciador da aplicação está na eminência de finalizar a execução do aplicativo. Uma instância pode entrar neste estado apenas uma vez.

5. Implementação

O aplicativo TVD.Calc tem por finalidade a realização de cálculos financeiros a partir de interações feitas pelo usuário e um aparelho digital de TVD, um *Set-top-box* ou emulador instalado em computadores convencionais. Podemos dividir a fase de execução da aplicação em dois focos: o momento em que o aplicativo está sendo carregado no aparelho residente, estando disponível para iniciar a interação; e o instante da interatividade, quando o usuário aciona e inicia a execução na TVD (Figura 3).

Inicialmente, uma chamada mostra durante a transmissão uma imagem dizendo que há um aplicativo sendo transmitido pela emissora e que este pode ser iniciado a partir do procedimento especificado, neste caso é necessário pressionar o botão ENTER do controle remoto ou teclado para iniciar interatividade. Após a aplicação ser iniciada pelo usuário, entraremos no segundo momento do TVD.Calc, conforme demonstrado pela Figura 3.





Figura 3. Interface Principal, carregada pós interação com o Usuário

Pós-inicialização, o usuário deve navegar pelos menus para poder visualizar as funcionalidades disponibilizadas. A Figura 4 apresenta a fórmula para cálculo de juros simples. Para tal ação, o usuário pressionou o botão requerendo detalhes sobre o tema.



Figura 4. Apresentação da Fórmula para Cálculo de Juros Simples

A base das fórmulas para os cálculos financeiros foi encapsulada no aplicativo cada qual com sua classe correspondente. O diagrama de classe exposto na Figura 5 denota tal questão.

ISSN: 2317-5346



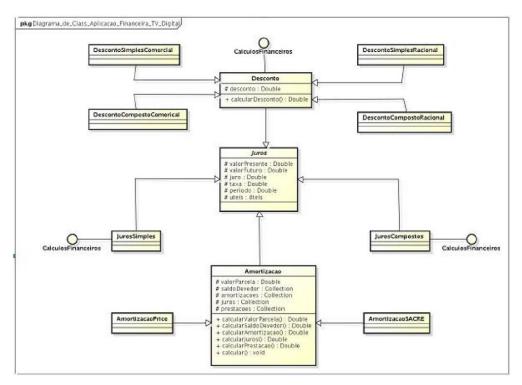


Figura 5. TVD. Calc - Diagrama de Classes

Também há espaço na aplicação para o usuário entrar com valores e obter resultados de forma direta. Essa função foi pensada, sobretudo objetivando englobar possíveis anúncios de companhias que mascaram taxas em seus produtos.



Figura 6. Front Taxa de Câmbio

Visando facilitar os anúncios de instituições financeiras e jornais, um método desenvolvido traz informações atualizadas como taxa de câmbio e taxa do Sistema Especial de Liquidação e de Custódia (SELIC). A Figura 6 mostra a apresentação do front vinculado à taxa de câmbio. As variações são importadas da bolsa de negociação brasileira em tempo real.



6. Avaliação

Para avaliação do software, foram instalados emuladores TVD em computadores convencionais, e aplicado na sequência questionários a 94 alunos das disciplinas Engenharia Econômica (Curso Superior em Computação) e Matemática III (Curso Técnico em Informática).

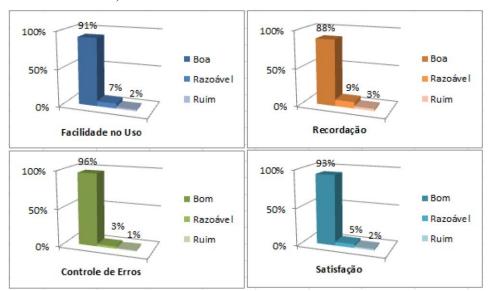


Figura 7. Avaliações

Nos questionários aplicados, foram pautadas questões que levaram cada usuário a interpretar o TVD.Calc de modo a avaliar quesitos referentes a sua usabilidade. As indagações foram formuladas abordando: a facilidade no uso, recordação das ações, controle de erros e satisfação.

Dados relatados pós-avaliação apresentaram como principal qualidade do aplicativo a possibilidade de interagir com o conteúdo durante a apresentação de uma vídeo-aula. Em hipermídias convencionais, não destinadas a ambientes televisivos, seria necessário o usuário pausar o vídeo e buscar alguma fonte que contemplasse os conteúdos propostos. Considerando a facilidade no uso, 91% avaliaram como boa, 7% como razoável e 2% como ruim. A Figura 7 também mostra dados relacionados aos outros critérios avaliativos.

7. Conclusão

Nota-se que a TVD é um revolucionário paradigma para a televisão, podendo proporcionar novas experiências em um ambiente televisivo para o telespectador ou usuário.

A aplicação TVD.Calc foi desenvolvida utilizando recursos para televisão, uma forma de se atingir a massa populacional. Desta forma, torna-se possível o provimento de informações contidas no aplicativo, transmitindo a diversas pessoas e localidades conhecimentos acerca de impostos, juros, amortizações, financiamentos e taxas de câmbio.

Julgando o emprego do middleware Ginga no Brasil e o alcance da digitalização televisiva mundial, como trabalhos futuros, propôe-se a integração de mais variáveis



financeiras à proposta. Também é sugerida a execução do aplicativo em ambientes de larga escala.

Referências

- ATSC (2013) "Advanced Television Systems Committee". Disponível em: http://www.atsc.org/cms/, acesso em outubro de 2013.
- Becker, Vargas, Filho e Montez (2004) "Juri Virtual I2TV Uma Aplicacao para TV Digital Interativa baseada em JavaTV e HyperProp". Disponível em: http://www.tvdi.inf.br/site/artigos/, acesso em outubro de 2013.
- Caporuscio, Mauro; Raverdy P.; Issarny, Valerie (2012) "ubiSOAP: A Service-Oriented
- Middleware for Ubiquitous Networking," IEEE Transactions on Services Computing, vol. 5, no. 1, pp. 86-98, Jan.-March.
- Cruz, Renato (2010) "Interatividade chega à TV Digital". Disponível em: http://orlandobarrozo.blog.br, acesso em outubro de 2013.
- Dataprev (2011) "TV Digital Social da Dataprev recebe título internacional de inovação". Disponível em: http://portal.dataprev.gov.br/tag/tv-digital/, acesso em outubro de 2013.
- DVB (2013) "Digital Video Broadcasting Project". Disponível em: http://www.dvb.org/, acesso em outubro de 2013.
- Fórum SBTVD (2013) "Fórum do Sistema Brasileiro de TV Digital Terrestre". Disponível em: http://forumsbtvd.org.br/, acesso em outubro de 2013.
- Ginga (2013) "TV Interativa se faz com Ginga". Disponível em: http://www.ginga.org.br/, acesso em outubro de 2013.
- IBGE (2011) "Instituto Brasileiro de Geografia e Estatística". Disponível em: www.ibge.gov.br, acesso em outubro de 2013.
- IMF (2013) "International Monetary Fund" www.imf.org, acesso em outubro de 2013.
- Lemos, Guido. Palestra Software Livre: TV Digital e Ginga-J Oportunidades para o Brasil. Produção de Guido Lemos. Local: Campus Party, 2010.
- Oracle Technology Network (2013) "Java TV". Disponível em: http://www.oracle.com/technetwork/java/javame/javatv/overview/getstarted/index.ht ml, acesso em outubro de 2013.
- Sedrez, Fernando (2008) "Desenvolvimento de um aplicativo para TV Digital Interativa utilizando a tecnologia Java TV". Disponível em: http://www.tvdi.inf.br/site/artigos/, acesso em outubro de 2013.
- Silva, Fernada; Moura, Thiago; Santos, Alice;. Ribeiro, Marília; Lemos, Guido e Brennand, Edma (2004) "TV Escola Interativa: uma Proposta Educativa para TV Digital". Disponível em:
- http://www.ufrgs.br/niee/eventos/RIBIE/2004/comunicacao/com670-679.pdf, acesso em setembro de 2013.



Malware em JAVA para Acessar Múltiplos Sistemas de Arquivos de Computadores Remotamente

Thiago Henrique de Almeida Espinhara¹

¹Universidade Federal Rural de Pernambuco (UFRPE) Av. Bom Pastor – Boa vista – Garanhuns-PE

thiagohe@gmail.com

Abstract: In this work we implemented a tool like malware in JAVA programming language to be able to obtain remote access to the infected computer and the application server control over the file system of the client application. We begin with a literature review on the state of malware today, types, their purposes and the strategies used for defense. Next is presented a methodology for the implementation of the malware that invades the file system, listing the features that will be available. Finally the results.

Resumo: Neste trabalho foi implementada uma ferramenta do tipo malware na linguagem de programação JAVA para que se conseguisse obter acesso remoto do computador infectado, tendo a aplicação servidor um controle sobre o sistema de arquivos da aplicação cliente. Inicialmente é feita uma revisão bibliográfica sobre o estado dos malwares na atualidade, os tipos, seus propósitos e as estratégias utilizadas para defesa. Em seguida é apresentada uma metodologia para a implementação do malware que invade o sistema de arquivos, listando as funcionalidades que estarão disponíveis. Por fim os resultados obtidos.

1. Introdução

Os vírus de computadores ou *malwares* estão presentes no nosso cotidiano, atuando como pragas e com um crescimento considerável a cada dia. A cada novo momento estão sendo utilizadas diversas estratégias distintas de ataque a computadores, de forma que os antivírus modernos não utilizam apenas vacinas para um determinado vírus, mas sim analisam o comportamento dos programas em execução em busca de anomalias.

A maioria das contaminações ocorre pela ação do usuário, executando arquivos infectados de diversas formas: recebido como anexo de um e-mail, propagado como um link em redes sociais, através de *pen-drives* ou dispositivos de entrada e saída, etc. Outra causa relevante diz respeito a falhas de segurança do sistema operacional. Daí a importância de mantê-lo sempre atualizado, para que as possíveis vulnerabilidades sejam corrigidas. Vírus mais elaborados tem hora programada para entrar em ação, ficando assim ocultos para o usuário.

Este trabalho objetiva implementar um *malware* na linguagem de programação JAVA, utilizando uma estratégia para acesso remoto não autorizado à arquivos da família de sistemas operacionais Windows. Dessa forma, esse vírus busca a quebra dos princípios da confiabilidade e integridade dos arquivos computador alvo.



2. Malwares

Malware, abreviação de software mal-intencionado, é um software programado para interromper o funcionamento do computador, coletar informações confidenciais ou obter acesso a sistemas de computador particular. Ele pode aparecer sob a forma de código, scripts ou de outro software [5]. *Malware* é um termo geral utilizado para designar uma variedade de formas de softwares hostil ou intruso [7].

Malware inclui vírus, ransomware, worms, trojan horses, rootkits, keyloggers, dialers, spyware, adware, BHOs malicioso, software de segurança desonestos e outros programas maliciosos, a maioria das ameaças de malware ativos são geralmente worms ou trojans em vez de vírus [7]. Existe uma distinção entre malware e softwares defeituosos, que são softwares legítimos, mas contém erros prejudiciais que não foram corrigidos antes de sua liberação. No entanto, malwares geralmente estão disfarçados como software genuíno, e podem vir de um site oficial da empresa, na forma de um programa útil ou atraente que tem o malware prejudicial embutido nele, juntamente com o software de rastreamento adicional que reúne as estatísticas convenientes.

Anti-vírus, anti-*malware* e firewalls são vastamente utilizados pelos usuários em todos os níveis: pequenas e grandes organização, nível de nação e global. Tais softwares podem ajudar a proteger os computadores contra ataques, ajudando a identificar e prevenir a propagação.

2.1 Propósitos

Muitos programas maliciosos, incluindo o primeiro *worm* de Internet, foram escritos como experiências ou brincadeiras. Hoje, os *malwares* são utilizados principalmente para roubar informações confidenciais de importância pessoal, financeira ou de negócios por hackers com intenções prejudiciais.

Malwares são muitas vezes usados de forma ampla contra o governo ou sites corporativos para coletar informações ou para interromper sua operação em geral. No entanto, o malware é muitas vezes usado contra indivíduos para obter informações pessoais, como senhas, números de cartões bancários ou de crédito, e assim por diante. Computadores pessoais utilizando rede de computador correm riscos consideráveis. Estes são frequentemente prevenidos por vários tipos de firewalls, software antivírus e hardware de rede.

Desde 2003, a maioria dos vírus e *worms* foram concebidos para assumir o controle de computadores dos usuários para a exploração do mercado negro [6]. São os chamados "computadores zumbis", que são utilizados para enviar SPAM, para hospedar dados de contrabando, como pornografía infantil [1] ou se engajar em ataques distribuídos de negação de serviço como uma forma de extorsão [8].

Outra categoria estritamente para fins lucrativos de *malware* surgiu, chamado de *spyware*. Estes programas são projetados para monitorar a navegação na web dos usuários, exibir propagandas não solicitadas ou redirecionar receitas de marketing da filial para o criador do *spyware*. Os programas de *spyware* não se espalham como vírus, em vez disso eles são geralmente instalados através da exploração de falhas de segurança. Eles também podem ser embalados em conjunto com o software instalado pelo usuário, tais como aplicações *peer-to-peer*.



2.2 Estratégias dos antimalwares

Como os ataques de *malware* se tornaram mais frequentes, a atenção com este tipo de praga começou a mudar. Atualmente existem diversos programas que foram desenvolvidos especificamente para o combate a *malwares*.

Um componente específico do anti-vírus ou do software anti-malware comumente referido como o scanner on-access ou em tempo real, conecta profundamente o núcleo do sistema operacional ou funções do kernel de uma forma similar a como um malware que tente operar, embora com a permissão do usuário informado para proteger o sistema. Toda vez que o sistema operacional acessa um arquivo, o scanner on-access verifica se o arquivo é um arquivo "legítimo" ou não. Se o arquivo é considerado um malware pelo scanner, a operação de acesso será interrompida, o arquivo será tratado pelo scanner em modo pré-definido (como o programa anti-vírus foi configurado durante a instalação) e o usuário será notificado. Isso pode retardar consideravelmente o sistema operacional dependendo de quão bem o scanner foi programado. O objetivo é interromper quaisquer operações que o malware possa tentar fazer no sistema antes que elas ocorram, incluindo as atividades que podem explorar bugs ou desencadear um comportamento inesperado no sistema operacional.

Os programas anti-malware podem combater os malwares de duas maneiras:

- 1. Eles podem fornecer proteção em tempo real contra a instalação de software de *malware* em um computador. Este tipo de proteção contra *malware* funciona da mesma maneira como a de proteção antivírus em que o software anti-*malware* verifica todos os dados de rede de entrada para o software de *malware* e bloqueia quaisquer ameaças que se depara.
- 2. Programas de software anti-*malware* podem ser usados exclusivamente para detecção e remoção de *malwares* que já foram instalados em um computador. Este tipo de software anti-*malware* verifica o conteúdo do Registro do Windows, arquivos do sistema operacional e programas instalados em um computador e fornecem uma lista de todas as ameaças encontradas, permitindo ao usuário escolher quais arquivos apagar ou manter, ou para comparar esta lista para uma lista de componentes de *malware* conhecidos, removendo arquivos que combinam.

3. Malware para Acessar o Sistema de Arquivos

Neste trabalho, foi implementada uma ferramenta para obter acesso remoto não autorizado ao Sistema de Arquivos de diversos Sistemas Operacionais, com o objetivo de comprovar vulnerabilidades de confidencialidade e integridade de arquivos ou diretórios. O requisito básico para o seu funcionamento é que o computador da vítima tenha uma JRE (*Java Runtime Environment*) instalada. O que não é um grande problema, pois as versões atuais dos principais Sistemas Operacionais trazem junto consigo o pacote do Java previamente instalado. O StuffedBiscuit possui duas aplicações: Cliente e Servidor. A aplicação Cliente é o módulo que deve ser executado no computador da vítima, possuindo o código-malicioso. Esta aplicação funciona



basicamente como um cliente-escravo, esperando os comandos enviados pelo servidor, executando-os e devolvendo as respostas. A aplicação Servidor é executada no computador em que se quer obter o controle sobre as máquinas clientes que se conectaram. A ferramenta tem a possibilidade de ter multiconexões, assim, diversas instâncias de conexões com as vítimas (clientes) são mantidas e podem ser acessadas de acordo com a vontade do invasor.

De alguma forma a aplicação cliente precisa ser executada no computador da vítima. Diferentes técnicas podem ser utilizadas para conseguir isto. Após a inicialização do cliente, automaticamente este se conecta ao servidor através de um número de IP (previamente conhecido por ele) e uma porta de comunicação (1234), utilizando para isto a classe java.net.Socket. Neste momento, a aplicação Servidor deve estar pronta para receber novas conexões. Caso novos clientes venham a se conectar após a primeira conexão, o Servidor tem a capacidade de aceitar e manter todas as novas conexões, dando a possibilidade do usuário escolher qual Cliente quer invadir. É importante destacar que esta aplicação é multiplataforma, ou seja, independente do Sistema Operacional sob o qual a aplicação Cliente está executando. Isto por se tratar de um programa Java, que executa sob a Máquina Virtual.

Consumada uma conexão entre as aplicações, o usuário do servidor pode então interagir com o cliente através de comandos, com formato similar aos do MS-DOS do Windows, previamente especificados, que estarão sendo executados remotamente no computador da vítima. Assim, a aplicação servidor pode acessar o sistema de arquivos do Cliente sem que nenhum mecanismo de defesa (firewall, antivírus) possa detectar.

3.1 Funcionalidades Implementadas

O usuário da aplicação Servidor pode interagir com o computador Cliente (vítima) utilizando os comandos descritos na Tabela 1. Estes comandos podem ser executados através de um Console um de cada vez. Quando executados quebram princípios de segurança de confidencialidade e integridade dos dados.

Tabela 1. Comandos Implementados pela Aplicação Servidor

Comando	Descrição
dir	Lista todos os diretórios e arquivos indexando-os com um número (para facilitar a usabilidade).
cd diretório	Acessa o diretório através do número a ele indexado.
cd	Retorna ao diretório anterior.
len arquivo	Mostra o tamanho em KB do arquivo especificado.



get arquivo	Realiza o download do arquivo especificado e salva-o no mesmo diretório da aplicação servidor.
del arquivo	Deleta o arquivo especificado.
del all	Deleta todos os arquivos, pastas e subpastas do diretório atual.
ping inicio fim	Realiza um ping remoto de acordo com a faixa de IPs passada como parâmetro.
list	Lista todos os computadores conectados.
con ip	Conecta ao IP especificado, alternando o controle.
exit	Termina a conexão e execução das aplicações Cliente e Servidor.

3.2 Arquitetura do sistema

A arquitetura do sistema baseia-se basicamente no modelo Cliente-Servidor (Master/Slave), onde existe a comunicação entre Cliente e Servidor através da Internet ou qualquer Rede de Computador que utilize o protocolo TCP/IP. Para o StuffedBiscuit em particular, a aplicação Cliente trata-se de um cliente-escravo, onde seu objetivo é realizar ações recebidas pelo Servidor e enviar-lhe a resposta adequada. Isto pode ser visualizado na Figura 1, onde o módulo Slave do Cliente recebe as informações do módulo de Rede. A comunicação em rede do sistema utiliza Socket, trazendo um elo bidirecional de comunicação. No Servidor, o usuário pode interagir com o sistema através de comandos em uma Interface que utiliza o Console (com aparência similar a um prompt de comando do Windows). Estes comandos são passados para o Controlador Master, que os processa e envia-os para o módulo de Rede do Servidor, responsável por transmitir o comando ao Cliente e aguardar a resposta. Quando a resposta é recebida, o resultado do comando é exibido na tela do console para o usuário.

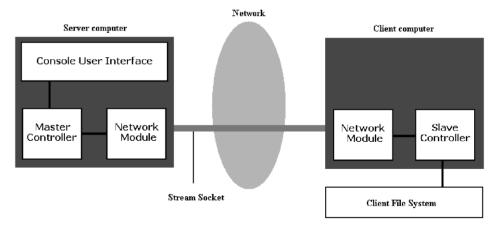


Figura 1. Arquitetura do Sistema.

ISSN: 2317-5346



3.3 Resultados

Foi observado que a aplicação cliente é de extremo risco para os usuários que possuem a máquina virtual do Java instalado, independente do Sistema Operacional que esteja em uso, demonstrando uma falha de segurança gravíssima em Java. Para constatar tal afirmação, foram realizados dois experimentos: o primeiro com máquinas com o Windows 7 e o segundo com Linux Ubuntu. Os resultados foram surpreendentes, onde foi possível obter acesso remoto aos arquivos dos usuários sem que nenhum antivírus, firewall ou restrições de acesso dos Sistemas Operacionais impedissem o funcionamento. Assim, mesmo o vírus executando sob usuários sem poder de administrador, o acesso e a modificação aos dados foi possível.

A funcionalidade mais perigosa implementada foi a denominada *del all*, que quando executada excluía todos os arquivos e pastas do diretório atual. A Figura 2 ilustra uma ação dessa função. As figuras 2.a e 2.c são as capturas de tela da aplicação servidor antes e depois da execução do comando, respectivamente. As figuras 2.b e 2.d são as capturas de tela do cliente (computador da vítima) antes e depois da ação executada pelo servidor.

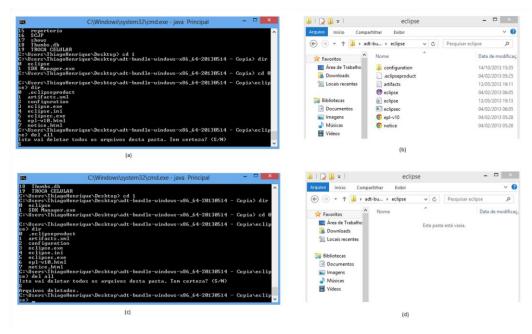


Figura 2. Ação da Funcionalidade del all.

4. Considerações Finais

O código produzido para a realização deste projeto foi desenvolvido para fins de pesquisa acadêmica. A intenção é de se estudar vulnerabilidades nos Sistemas Operacionais, através das redes de computadores, utilizando a linguagem de programação JAVA, e não para utilização em atividades criminosas, como previsto na Lei 12.737.

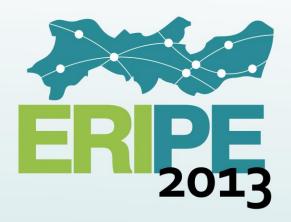
Observou-se que a aplicação do vírus não teve o seu funcionamento barrado por nenhum tipo de proteção, como antivírus ou *firewall* de ambos os Sistemas Operacionais testados: Windows e Linux. Esta não detecção se dá provavelmente por



dois motivos: (a) A aplicação não abre portas no computador da vítima (para entrada), fazendo uma conexão TCP através de um Socket (referenciando portas de saída). O bloqueio de pacotes de saída é mais complicado porque, embora muitos sistemas adotem convenções padrão para numeração de portas, eles não são obrigados a fazê-lo [13]. Para alguns serviços importantes, como FTP (*File Transfer Protocol*), esta atribuição é feita dinamicamente. (b) Todo programa JAVA executa sobre sua Máquina Virtual, o que pode dificultar na identificação do vírus. Desta forma, a portabilidade do vírus em relação ao Sistema Operacional ficou garantida, por utilizar JAVA. Ou seja, através deste é possível ter acesso simultâneo ao Sistema de Arquivos de quaisquer Sistema Operacional, executando os comandos *list* e *con* da Tabela 1 para realizar a migração.

Referências

- [1] Child Porn: Malware's Ultimate Evil". 2009.
- [2] Continuing Business with Malware Infected Customers". Gunter Ollmann. 2008.
- [3] F-Secure Reports Amount of Malware Grew by 100% during 2007" (Press release). F-Secure Corporation. 2007.
- [4] F-Secure Quarterly Security Wrap-up for the first quarter of 2008". F-Secure.
- [5] ICS-CERT. "An Undirected Attack Against Critical Infrastructure". Us-cert.gov. http://ics-cert.us-cert.gov/pdf/undirected_attack0905.pdf. Acessado em 24 de maio de 2013.
- [6] Malware Revolution: A Change in Target". 2007.
- [7] Microfost. TechNet library. "Defining Malware: FAQ". http://technet.microsoft.com/en-us/library/dd632948.aspx. Acessado em 24 de maio de 2013.
- [8] PC World Zombie PCs: Silent, Growing Threat.
- [9] Symantec Internet Security Threat Report: Trends for July–December 2007 (Executive Summary) (PDF) XIII. Symantec Corp. 2008.
- [10] New Research Shows Remote Users Expose Companies to Cybercrime". Webroot. 2013.
- [11] Symantec names Shaoxing, China as world's malware capital". Engadget. 15/04/2010.
- [12] Rooney, Ben (23/05/2011). "Malware Is Posing Increasing Danger". Wall Street Journal.
- [13] Tanenbaum, A. S. Redes de computadores. Tradução Vamdemberg D. de Souza. 13ª edição. Elsevier. Rio de Janeiro. p. 827. 1994.



III ESCOLA REGIONAL DE INFORMÁTICA DE PERNAMBUCO GARANHUNS, PE 6-8 DE NOVEMBRO

Tema:

"Desafios da Computação: Academia-Indústria-Comunidade"

Resumos Expandidos

ISSN 2317-5346





Apoio:







Uma análise comparativa entre protocolos de comunicação autonômicos baseados em *gossip*

Eduardo C. B. Julião¹, Émerson R. Silva¹, Patricia T. Endo¹

¹Faculdade de Ciências e Tecnologia de Caruaru - Universidade de Pernambuco (UPE) Rodovia BR 104, Km 62 - Polo Caruaru - Caruaru - PE – Brasil

{eduardocesarbj, emersonremigio, patriciaendo}@gmail.com

Resumo. Os protocolos autonômicos e distribuídos baseados em gossip são bastante utilizados em cenários onde há necessidade de uma rapidez na comunicação e de uma fácil escalabilidade. O presente resumo tem como principal objetivo apresentar o comportamento desses protocolos, possibilitando uma análise comparativa acerca dos mesmos. São apresentados alguns exemplos de utilização desses protocolos, discutindo suas correspondentes otimizações.

Abstract. Autonomic and distributed protocols based on gossip are widely used in scenarios where is a fast communication and an easy scalability are needed. This article has as main objective presenting these protocols behavior, enabling a comparative analysis between them. We present some usage examples of such protocols, discussing their corresponding optimizations.

1. Introdução

Desde o advento da computação, os sistemas de computadores estão em constante evolução. Dentre essas evoluções está o avanço do paradigma de sistemas centralizados para os sistemas distribuídos. Os sistemas centralizados funcionam, basicamente, de modo que um servidor concentra todas as informações que serão utilizadas ou requisitadas por seus clientes. Porém, esses sistemas possuem alguns pontos fracos bastante conhecidos, tais quais: único ponto de falha ou gargalo, poder computacional limitado e centralizado. Por outro lado, os sistemas distribuídos, definidos por Coulouris, Dollimore e Kindberg (2007, p. 15) como "aqueles nos quais os componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens", minimizam os pontos fracos dos sistemas centralizados citados anteriormente. Agora, pelo fato dos computadores estarem distribuídos fisicamente, mas conectados uns aos outros através de uma rede e poderem trocar informação entre si, o sistema como um todo pode viabilizar uma maior capacidade de processamento, maior robustez e uma maior escalabilidade.

Deste modo, a comunicação realizada entre os computadores de um sistema distribuído se dá por trocas de mensagens, que são padronizadas por protocolos de comunicação. Os protocolos abordados neste trabalho realizam a disseminação das mensagens baseadas em *gossip* (fofoca). Esses tipos de protocolos estão sendo cada vez mais utilizados nos sistemas distribuídos por apresentarem muitas vantagens, como a simplicidade, escalabilidade e a alta velocidade de difusão de informação. Porém,



alguns pontos fracos também podem ser identificados nos mesmos, como por exemplo, a transmissão de mensagens redundantes e a dificuldade de identificar uma condição de parada da propagação de uma informação após o seu início.

Este resumo tem como principal objetivo realizar um levantamento do estado da arte sobre os protocolos de comunicação baseados em *gossip* e apresentar uma análise comparativa entre os mesmos, destacando os pontos fortes e fracos de cada tipo de implementação. Para tanto, o resumo está estruturado da seguinte forma: a Seção 2 apresenta os protocolos baseados em *gossip* e uma classificação para os mesmos; a Seção 3 descreve e discute sobre alguns protocolos existentes; e por fim a Seção 4 apresenta as conclusões obtidas e os trabalhos futuros.

2. Protocolos baseados em gossip

Os protocolos baseados em *gossip* têm esse nome particular por realmente parecerem com uma fofoca, pois funcionam de modo que uma informação, no caso a fofoca, se propaga rapidamente em pouco tempo. Por isso, os nós que possuem uma informação fazem a disseminação da mesma entre um número aleatório de vizinhos visando atingir a maior quantidade de nós possível.

Demers et al. (1987) foram os pioneiros ao trabalharem com protocolo baseado em *gossip*, usando-o para manutenção de banco de dados replicados. Isto serviu como ponta pé inicial para que seu uso fosse amplamente estendido para diversas soluções em sistemas distribuídos, tais como comunicação, gerenciamento de recursos até a detecção de falhas.

2.1. Gossip genérico

O esquema genérico dos protocolos baseados em *gossip* é baseado em duas *threads*, sendo uma *thread* ativa e a outra passiva, como mostra a Figura 1. Cada nó possui as duas *threads* implementadas, sendo a ativa utilizada no momento em que um determinado nó deseja iniciar uma comunicação, ou seja, basicamente um nó que deseja transmitir uma mensagem escolhe alguns nós vizinhos randomicamente para então difundir essa mensagem; e a passiva executada sempre em um nó, pois é a responsável por receber e aceitar requisições provenientes dos vizinhos.

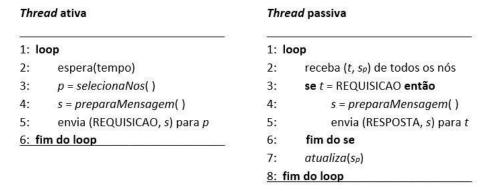


Figura 1. O esquema genérico do gossip (Traduzido de MONTRESOR, 2008)

A thread ativa seleciona primeiramente, a cada determinado espaço de tempo e através do método selecionaNos(), uma quantidade aleatória de nós vizinhos que é atribuída à variável p. Posteriormente, é com o método preparaMensagem() que torna-



se possível armazenar informações sobre estado local do nó na variável s, para que então essa seja enviada, através do método envia(REQUISIÇÃO,s), para os nós selecionados.

Já a thread passiva espera mensagens vindas dos seus nós vizinhos por tempo indefinido. Também faz uso do método preparaMensagem(), caso receba uma requisição de um vizinho, fazendo com que seja enviado ao remetente as informações sobre seu estado local. Por fim, o método $atualiza(s_p)$ é responsável por realizar a atualização do estado local do nó pelo que foi recebido por seu vizinho.

Montressor (2008) define o esquema mostrado na Figura 1 de forma genérica. Devido a isso, ele lista uma série de características que diferenciam os protocolos baseados em *gossip* dos que não são:

- A seleção dos nós vizinhos deve ser randômica;
- Apenas informações locais estão disponíveis em todos os nós;
- A comunicação é periódica;
- A transmissão e a capacidade de processamento são limitadas;
- Todos os nós executam o mesmo protocolo.

2.2. Variações do protocolo gossip genérico

Existem dois parâmetros fundamentais na configuração dos protocolos *gossip* segundo Leitão (2007), que podem influenciar no desempenho dos mesmos: *fanout* e *maximum rounds*. Ambos estão representados na Figura 2, onde *fanout* representa o número de nós selecionados pelo protocolo para cada mensagem recebida pela primeira vez; e o segundo parâmetro, o *maximum rounds*, descreve o número máximo de vezes que uma determinada mensagem será retransmitida pelos nós.

Dessa forma, o autor apresenta dois modos nos quais os protocolos baseados em *gossip* podem atuar, sendo o Modo Ilimitado, onde não é estabelecido o parâmetro *maximum rounds*, e o Modo Limitado, o qual determina que o valor do parâmetro *maximum rounds* seja maior que 0, limitando então a retransmissão de cada mensagem.

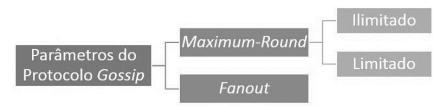


Figura 2. Parâmetros de configuração dos protocolos *gossip* (baseado em Leitão, 2007)

Através da Figura 3, torna-se possível a visualização do comportamento do protocolo baseado em *gossip*, onde no primeiro momento o círculo cinza escuro, que representa a fonte da mensagem que deve ser transmitida para todos os outros círculos do sistema, inicia a difusão da mensagem para um número aleatório de vizinhos. Após isso, cada um dos círculos que recebeu a mensagem repete o procedimento, atingindo uma outra quantidade de vizinhos aleatórios que darão continuidade ao processo. Com isso, é bem provável que, após alguns rounds, os círculos que ainda não tinham recebido



a mensagem recebam. A propagação da mensagem, então, torna-se extremamente rápida por difundila com crescimento exponencial entre os participantes de um sistema.

2.3. Classificação

Os algoritmos baseados em *gossip* podem ser divididos, segundo Renesse et al. (2008), em duas categorias, sendo elas: *Anti-Entropy* e *Rumor Mongering*. A primeira categoria propaga a informação desejada até o momento em que ela seja desatualizada por uma nova informação, já a segunda espalha a informação por um período de tempo definido previamente, como descrevem, sucintamente, os autores. A Figura 4 tem a finalidade de resumir o comportamento desses protocolos, bem como os relacionar com um dos parâmetros vistos na Seção anterior. Onde esse parâmetro, o *maximum-round*, é responsável por limitar ou não o número de vezes que uma mensagem será transmitida.

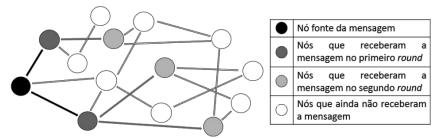


Figura 3. O comportamento do algoritmo baseado em gossip

2.3.1. Anti-Entropy

Revière e Voulgaris (2011, p. 257) detalham o comportamento desse algoritmo: "No AntiEntropy, cada nó, fofoca periodicamente, ou seja, periodicamente ele escolhe randomicamente um nó entre todos os outros". Nesse sentindo, pode-se concluir que essa categoria age semelhante ao algoritmo genérico, pois, ele somente preocupa-se na propagação da mensagem, diferente do que acontece no Rumor Mongering, que será visto no tópico à seguir. A categoria atua, basicamente, de forma que a cada espaço de tempo os nós que possuem uma mensagem escolhem de maneira aleatória algum vizinho para enviá-la.

2.3.2. Rumor Mongering

Esse protocolo é caracterizado por atribuir um determinado número à mensagem que se pretende transmitir, número que, por sua vez, pode ser chamado de tempo de vida da mensagem, é responsável por representar a permanência da mensagem na rede e é decrescido a cada vez em que a mensagem é transmitida.

Então, ele funciona de modo que um nó inicia a transmissão de uma mensagem escolhendo um número aleatório de vizinhos, que por sua vez, dão continuidade ao processo de propagação da mensagem. Essa propagação só é realizada uma única vez para cada mensagem recebida até que o tempo de vida da mensagem seja igual a zero, ou seja, quando um nó recebe uma mensagem, ele escolhe alguns de seus vizinhos randomicamente para enviar a mensagem, e, então, para o processo de propagação até que uma nova mensagem seja recebida.



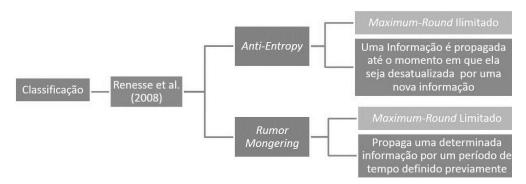


Figura 4. Classificação do Gossip

3. Implementações existentes

Atualmente a utilização dos protocolos baseados em *gossip* abrangem uma extensa área da computação, sendo utilizados tanto para detecção de falhas quanto para manutenção de banco de dados replicados. Nas sub-seções a seguir serão abordados alguns protocolos e suas respectivas utilidades.

3.1. NeEM

O Network Friendly Epidemic Multicast é um exemplo de protocolo baseado em *gossip* e fora apresentado por Pereira et al. (2003). O protocolo proposto tem como característica crucial ser *network-friendly*, o que faz com que a comunicação entre os nós dependa de um protocolo da camada de transporte orientado a conexão.

O principal objetivo do protocolo é garantir que os nós não congestionem ainda mais a rede durante um período de sobrecarga. Para que esse objetivo fosse atingido, os autores desenvolveram dois mecanismos que o diferenciam de um protocolo baseado em *gossip* padrão. Sendo o primeiro mecanismo o uso do protocolo TCP, propiciando assim, o uso seguro da largura de banda disponível e a diminuição de perdas de mensagens motivadas pelo congestionamento da rede. E o segundo uma técnica de gerenciamento de *buffer*, que tem a finalidade de descartar mensagens em excesso e manter as informações relevantes.

Portanto, foi realizado um experimento com o jogo *Microsoft Flight Simulator* 2002 através da execução de uma simulação com um número fixo de 500 nós. O que tornou possível demonstrar que os resultados do desempenho do protocolo foram satisfatórios dentre eles, tanto quanto a atomicidade, pois, aumentou o número de jogadores simultâneos de 5 para 20, quanto latência, devido ao aumento da mesma em decorrência da interrupção, quando possível, da transmissão de mensagens obsoletas.

3.2. HEAP

Frey et al. (2009) propõem em seu trabalho o protocolo HEAP (*Heterogeneity-Aware Gossip Protocol*), e o designam para distribuição de conteúdo colaborativo em ambientes heterogêneos. Os autores descrevem-no sobre duas perspectivas, sendo em primeiro lugar os resultados matemáticos confirmando a eficiência da disseminação de mensagens dos protocolos baseados em *gossip*, e, em segundo, a possibilidade da implementação de algum protocolo de agregação, que faz com que seja fornecido



continuamente a cada nó uma aproximação de sua capacidade de largura de banda relativa.

Dessa forma, são relatadas avaliações que permitem observar algumas melhorias desse protocolo quando comparado ao protocolo baseado em *gossip* tradicional, que tem funcionamento ideal em cenários homogêneos e em redes sem restrições, mas, se mostrou ineficiente em cenários heterogêneos. O HEAP, por sua vez, otimizou o uso da largura de banda, pois ele maximiza a utilização da banda de nós com grande capacidade, da mesma forma que minimiza os que possuem menor capacidade, melhorando assim, a utilização de largura de banda total e fazendo com que haja uma contribuição mais eficiente da largura de banda entre os nós.

Foram realizados experimentos com o número aproximado de 270 nós *PlanetLab* em uma aplicação de *streaming* de vídeo, comparando a adaptação proposta com o protocolo clássico. Então, ficou comprovado que o HEAP adapta a carga de cada nó de acordo com a sua largura de banda, melhora a qualidade de *streaming* de todos os nós, progride o *lag* de *stream* de 40% para 60% em relação ao *gossip* padrão, e, por fim, resiste bem a cenários de falha, diferente do que acontece no *gossip* genérico.

3.3. CREW

O CREW (Concurrent Random Expanding Walkers) foi desenvolvido por Deshpande et al. (2006) e tem como meta principal manter a escalabilidade e a resistência às falhas dos protocolos baseados em gossip continuando com a rápida transmissão de mensagens em redes heterogêneas. O tipo de disseminação utilizada pelo protocolo é a Disseminação Flash, a qual consiste em uma rápida propagação de informações para uma grande quantidade de nós em um período curto de tempo e tem como principais características: imprevisibilidade, escalabilidade e heterogeneidade da rede e de conteúdo.

A implementação do CREW se fez a partir do zero e foi feita através de uma plataforma *middleware* escalável, cujas conexões entre os nós usam o TCP, para que seja estimada a largura de banda disponível. Onde o Protocolo *Bounce*, que funciona como um serviço básico de *membership* passou a ser utilizado, e otimizações foram realizadas, o que tornou possível a diminuição do envio de mensagens redundantes, a redução da sobrecarga na rede e a sua adaptação em redes heterogêneas, fazendo com que a transmissão de dados seja continuamente rápida e eficiente.

O protocolo foi testado por meio da ferramenta *Modelnet*, que permite a moldagem do tráfego de rede em tempo real e a configuração de várias topologias de redes, e tomou como base alguns parâmetros, sendo uns deles: rapidez da disseminação de uma mensagem para nós espalhados em uma rede ampla, sobrecarga dos dados e adaptação em redes heterogêneas. Foi comparado com os sistemas de disseminação: *BitTorrent*, *Bullet*, *SplitStream* e *Asynchronous TCP Gossip*, o que permitiu enxergar que o CREW os demais sistemas tanto quanto ao desempenho, como também nos vários aspectos observados.

3.4. Análise e discussão

Os protocolos baseados em *gossip* estão sendo utilizados e modificados constantemente desde sua criação. Assim, a Tabela 1 apresenta um resumo dos protocolos abordados no



tópico anterior, onde os autores de todos os protocolos estudados propuseram otimizações quanto ao *gossip* genérico que se mostraram eficientes nos experimentos realizados.

Como pôde ser observado, os protocolos foram utilizados com objetivos e áreas de atuação diferentes. No entanto, todos mantiveram como foco principal a rápida propagação de informação, que é a finalidade primordial do protocolo de comunicação baseado em *gossip*. Além de realizarem otimizações que visam solucionar alguns pontos fracos do *gossip*, como a transmissão de mensagens redundantes e da sobrecarga na rede. Como mencionado na Seção 2, as mensagens redundantes e a sobrecarga na rede são consequência da forma de difusão das mensagens: os nós que precisam enviar uma mensagem, escolhem seus vizinhos randomicamente e, portanto, alguns vizinhos podem receber a mesma mensagem diversas vezes. Por outro lado, deve-se considerar que essa redundância auxilia na questão da confiabilidade do protocolo, pois, caso haja alguma falha de comunicação na rede, um nó poderá receber uma mensagem (considerada perdida) em um outro momento, caso seja escolhido por seu vizinho.

No NeEM uma otimização bastante significativa se deu pelo uso do TCP, que proporcionou uma confiável disseminação de mensagens mesmo durante congestionamentos na rede. E também o emprego de uma técnica de gerenciamento de buffer, que tem como principal função a eliminação de mensagens em excesso.

O HEAP considera que o *gossip* genérico não apresenta a mesma eficiência em redes heterogêneas como possui em redes homogêneas. Os autores otimizaram o protocolo, tendo como principal aprimoramento a maximização do uso da banda, onde há uma adaptação de carga de cada nó, a fim de evitar o congestionamento daqueles que possuem uma baixa capacidade de carga e de aumentar a contribuição dos nós que detém uma maior largura de banda.

No CREW os autores também se preocuparam com a perda de eficiência do gossip genérico em redes heterogêneas, otimizando o mesmo, com base do emprego de duas técnicas: redução de mensagens redundantes e diminuição da sobrecarga na concorrência entre os nós, que se tornaram possível através da utilização dos módulos Bandwidth Manager, que calcula e otimiza a largura de banda, e Random Walker, que garante a conexão entre os nós.

Tabela 1. Otimização do gossip



	NeEM	HEAP	CREW
Atuação	Congestionamento de redes	Distribuição de conteúdo colaborativo em ambientes heterogêneos	Disseminação Flash
Objetivo	Garantir que os nós não congestione ainda mais a rede durante um período de sobrecarga	Adaptar a contribuição de cada nó de acordo com sua largura de banda	Manter a escalabilidade, resistência a falhas e a rápida transmissão de mensagens em redes heterogêneas
Otimização	Uso do TCP e técnica de gerenciamento de buffer	Maximização do uso da banda	Utilização de conexões TCP, do Protocolo Bounce, entre outras
Experimento	Jogo Microsoft Flight Simulator 2002 com 500 nós	Aplicação de streaming de vídeo cerca de 270 nós	Ferramenta Modelnet
Resultado	Aumento do número de jogadores simultâneos e aumento da latência	Melhora na utilização da largura de banda	Redução do envio de mensagens redundantes, diminuição da sobrecarga na rede e adaptação em redes heterogêneas
Referência	Pereira et al. (2003)	Frey et al. (2009)	Deshpande et al. (2006)

4. Conclusões e trabalhos futuros

Este resumo apresentou um estudo sobre protocolos autonômicos e distribuídos baseados em *gossip*. Onde foi apresentado o *gossip* genérico com alguns parâmetros importantes e uma classificação bastante relevante. Por fim, foram exibidos exemplos otimizados de protocolos baseados em *gossip*, comprovando assim o uso desse tipo de protocolo em diversos âmbitos e a sua rapidez na propagação da informação.

Como contribuição, o presente resumo apresenta uma base para estudos mais aprofundados sobre os protocolos estudados e também para a implementação de otimizações aplicáveis aos mesmos. E como trabalhos futuros, pretende-se modelar e simular diferentes implementações de protocolos baseados em *gossip* com o objetivo de realizar uma avaliação quantitativa entre os mesmos.

Referências

COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas Distribuídos:** conceitos e projeto. 4ª ed. Bookman, 2007.

DEMERS, Alan; GREENE, Dan; HAUSER, Carl; IRISH, Wes; LARSON, John; SHENKER, Scott; STURGIS, Howard; SWINEHART, Dan; TERRY, Doug. **Epidemic algorithms for replicated database maintenance**. In: 6th ACM Symposium on Principles of distributed computing. Nova Iorque, 1987.

DESHPANDE, M.; XING, B.; LAZARDIS, I.; HORE, B.;

VENKATASUBRAMANIAM, N.; MEHROTRA, S. Crew: A Gossip-based Flash-

Dissemination System. In: 26th International Conference on Distributed Computing Systems. Washington, DC, EUA, 2006.



- FREY, Davide; GUERRAOUIL, Rachid; KERMARREC, Anne-Marie; KOLDEHOFE, Boris; MOGENSEN, Martin; MONOD, Maxime; QUÉMA, Vivien. **Heterogeneous Gossip**. In: International Conference on Middleware. Nova Iorque, 2009.
- HOLLERUNG, Tim Daniel; BLECKMANN, Peter. **Epidemic Algorithms**. Universitat Paderborn, 2004.
- KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet: uma abordagem top-down. 5ª ed. Pearson, 2010.
- LEITÃO, João Carlos Antunes. **Gossip-based broadcast protocols**. 2007. 92 f. Dissertação (Mestrado em Engenharia Informática) Universidade de Lisboa. Lisboa, 2007.
- MONTRESOR, Alberto. **Intelligent Gossip**. In: Intelligent Distributed Computing, Systems and Applications. Springer Berlin Heidelberg, 2008.
- PEREIRA, J.; RODRIGUES, L.; MONTEIRO, M. J.; OLIVEIRA, R.; KERMARREC, A. M. **NeEM**: Network-friendly epidemic multicast. In: 22th Symposium on Reliable Distributed Systems. Florência, Itália, 2003.
- RIVIÈRE, Etienne; VOULGARIS, Spyros. Gossip-based networking for internetscale distributed systems. In: E-Technologies: Transformation in a Connected World. p. 253-284. Springer Berlin Heidelberg, 2011.
- TANENBAUM, Andrew S.; VAN STEEN, Maarten. **Sistemas Distribuídos:** princípios e paradigmas. 2ª ed. Pearson, 2008.



DinnerRace: Um simulador para técnicas de exclusão mútua através do problema do Jantar dos Filósofos

Amanda Barbosa¹, Witássio Oliveira¹, Kádna Camboim¹

¹Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n – Boa Vista – 55.292-270 – Garanhuns – PE, Brasil

amandaoliveirabr@gmail.com, witassio@gmail.com, kadna@uag.ufrpe.br

Resumo. Este artigo discute sobre o uso e aplicação do DinnerRace, um software que simula a aplicação de técnicas de exclusão mútua sobre o Problema Jantar dos Filósofos, como auxiliar no ensino e aprendizagem de conceitos na disciplina de Sistemas Operacionais em cursos de computação.

Abstract. This paper discuss the development and application of the DinnerRace, a simulator that applies some methods for mutual exclusion on the Dining Philosophers problem, and how to use it as a tool for visual aid in teaching the mutual exclusion concepts on a Operacional Systems course.

1. Introdução

O estudo de Sistemas Operacionais (SO) é fundamental em cursos de Computação e Informática. No estudo da disciplina são introduzidos conceitos como gerência de memória, sistema de arquivos, gerência de processos, etc, que são importantes à compreensão do funcionamento e integração entre *software* e *hardware*. Os conceitos abordados, têm, em sua maioria, um alto nível de abstração e complexidade e nem sempre os alunos conseguem obter um entendimento completo e um bom desempenho na disciplina. Segundo Machado e Maia (2004), o problema está tanto no modelo de ensino como na falta de ferramentas capazes de traduzir para a realidade os conceitos teóricos apresentados. Geralmente, as aulas sobre Sistemas Operacionais são teóricas, com poucos exemplos práticos, tornando o aprendizado mais difícil.

Um dos principais conceitos abordados na disciplina é o de exclusão mútua, que está relacionado à concorrência de processos. Para a área de projeto de processadores, esse é um conceito chave em sistemas operacionais, mas, a abstração tem impedido muitos estudantes de compreenderem e aplicarem os conceitos aprendidos. Muitas abordagens foram e vem sendo estudadas a fim de tornar o estudo mais dinâmico, como a construção de laboratórios voltados para a disciplina, o uso prático do kernel estruturado de Linux a fim de trazer conceitos básicos para o estudo da disciplina. Mas a dificuldade de construir e manter esses laboratórios, bem como de familiarizar os estudantes ao uso de Linux, torna essas abordagens, muitas vezes, inviáveis.

Este trabalho aborda o uso de um *software* que simula várias soluções de exclusão mútua, sobre o problema Jantar dos Filósofos, que foi desenvolvido como ferramenta auxiliar para o ensino dos conceitos relacionados à disciplina de Sistemas Operacionais.



2. Jantar dos Filósofos

O problema do Jantar dos Filósofos é frequentemente usado em programação concorrente para demonstrar conceitos de sincronização. Inicialmente, o problema foi proposto por E. W. Dijkstra, um renomado cientista da computação alemão, em 1965. A ideia do jantar dos filósofos pode ser representada como na Figura 1, onde há cinco filósofos prontos para jantar em uma mesa que contém cinco pratos e cinco garfos. Cada filósofo pode comer, pensar ou dormir. Porém, para um filósofo comer são necessários dois garfos (direita e esquerda). Sempre que um filósofo terminar de comer, ele deve colocar os garfos de volta na mesa. O compartilhamento do recurso (garfo) limita a quantidade de filósofos que podem comer ao mesmo tempo, e gerenciar esse recurso é o real problema.



Figura 1: Ilustração do Problema Jantar dos Filósofos

3. Exclusão mútua

No estudo dos Sistemas Operacionais, no que diz respeito à comunicação entre processos, há a comunicação por mensagens e a comunicação através de recursos compartilhados. Essa última pode ser interpretada de maneira similar ao Problema do Jantar dos filósofos. A solução para este tipo de comunicação é a exclusão mútua.

Frequentemente, processos (programas em execução) compartilham dados da memória - variáveis ou arquivos. Essa área compartilhada é chamada de região crítica. É recomendado evitar o acesso simultâneo aos dados, pois, isso pode gerar inconsistência e até erros de sistema.

Exclusão mútua é o que garante que, quando um processo está fazendo uso de sua região crítica, nenhum outro processo poderá fazer uso dessa região. Para garantir a exclusão mútua é preciso respeitar as seguintes premissas (TANENBAUM, 2003): 1) Nenhum dos dois processos pode estar simultaneamente dentro da sua região crítica; 2) Nenhuma suposição pode ser feita sobre as velocidades ou sobre o número de CPUs; 3) Nenhum processo que executa fora de sua região crítica pode bloquear outro processo; 4) Nenhum processo deve ter de esperar eternamente para entrar em sua região crítica.

A fim de garantir a exclusão mútua no problema do Jantar dos Filósofos, o simulador desenvolvido para a realização deste trabalho aborda os quatro algoritmos a seguir: desativação das interrupções, alternância estrita, variável de bloqueio e *sleep and wait*.



3.1. Desativação das interrupções

Cada processo desativa as interrupções imediatamente depois de entrar em sua região crítica. As interrupções são reativadas imediatamente após o processo deixar a região crítica. Com essa solução, a CPU fica impedida de alternar entre os processos para realizar a multiprogramação. Em geral, a CPU só alterna de um processo para outro como resultado de interrupções, como por exemplo, interrupção de relógio. Com a incapacidade de alternar entre os processos, o processo que entrou na região crítica poderá examinar e atualizar a memória compartilhada com a certeza de que nenhum outro processo irá interferir. Essa técnica, apesar de simples traz alguns problemas, pois, não é aconselhável permitir que processos de usuário desativem instruções e não permitam a execução de interrupções. Isso vai de encontro a segurança dos SOs, dado que uma ação como esta poderá levar a uma situação extrema, em que as instruções desativadas nunca virem a ser executadas (SILBERSCHATZ, 2001).

3.2. Variável de Bloqueio

Essa técnica define uma variável de bloqueio compartilhada, que poderá assumir os valores zero (0) e um (1). Inicialmente, essa variável recebe o valor 0. Se um processo entra na sua região crítica, essa variável compartilhada receberá o valor 1.

3.3. Alternância estrita

A solução baseada em alternância estrita, assim como a variável de bloqueio, também utiliza uma variável compartilhada, denominada *turn*, que pode assumir os valores zero (0) e um (1). Essa variável monitora de qual processo é a vez de entrar na região crítica. Para *turn* igual a 0, o primeiro processo terá a vez de entrar na região crítica. O segundo processo testa a variável *turn* continuamente, até que ela seja 1. Quando *turn* igual a 1, o processo entrará na região crítica. Essa solução acarreta numa situação que desperdiça tempo de CPU, a qual é conhecida como espera ativa já que a variável é testada continuamente, até que um determinado valor apareça.

3.4. Sleep and Wake up

A primitiva *sleep* causa o bloqueio do processo que fez a chamada, suspendendo-o até que o outro processo o acorde. A primitiva *wakeup* é responsável por desbloquear o processo. Esta solução evita o esperdício de tempo da CPU.

4. Software DinnerRace

O *software DinnerRace* implementa as quatro propostas (acima citadas) para exclusão mútua. A tela inicial do *software DinnerRace* é apresentada na Figura 2. Como pode ser observado, apenas a escolha do algoritmo deve ser feita. Após isso, inicia-se a simulação que dura por um tempo aleatório. O *software* termina de simular cada um dos algoritmos quando todos os filósofos passam por todos os estados.

Após a escolha do algoritmo, o *software* simula o jantar dos filósofos, o que implica em um determinado tipo de execução para cada algoritmo. É exibido na tela o status atual de cada filósofo de acordo com o algoritmo escolhido, sendo os status: Comendo, Faminto e Pensando. A Figura 3 apresenta o *DinnerRace* executando o algoritmo de



alternância estrita. Vale ressaltar, que o exemplo dado é apenas uma parte da simulação, pois, se for observada a barra de rolagem da figura 3, parte da execução não está visível.

O simulador foi desenvolvido utilizando a linguagem Java, cuja escolha se deu pelo fato de ela ser portável – já que geralmente os computadores dos estudantes possuem variadas configurações de hardware e software – bem como, melhor domínio por parte dos desenvolvedores. Com seu uso, os estudantes podem constatar o efeito dos diferentes algoritmos sobre o problema, enfatizando que o *DinnerRace* gerencia os recursos compartilhados, garantindo que não haverá conflitos entre os processos na busca pelo recurso. Caso um processo esteja utilizando o recurso, o processo que chegou por último deve esperar até que o recurso esteja disponível para o seu uso. Assim, pode-se dizer que houve exclusão mútua.



Figura 2. Tela inicial do DinnerRace



Figura 3. Resultado da execução da simulação com Alternância Estrita

Como propostas para trabalhos futuros, pretende-se implementar os algoritmos de semáforos, monitores e instrução TSL para exclusão mútua do Jantar dos Filósofos. Além disso, pretende-se aprimorar a interface gráfica que permitirá a visualização dos filósofos ao levantar os garfos e pô-los novamente na mesa, enfatizando quais filósofos sentem fome, pensam ou comem. E ainda, serão realizados estudo de casos com alunos da disciplina de SO para corrigir eventuais problemas.

Referências

MACHADO, F.B., MAIA, L.P. Um Framework Construtivista no Aprendizado de Sistemas Operacionais - Uma Proposta Pedagógica com o uso do Simulador SOsim. XII Workshop de Educação em Computação (WEI), XXIV Congresso da Sociedade Brasileira de Computação (SBC), Salvador, BA, ago. 2004.



LAADAN, O., NIEH, L., VIENNOT, N., Teaching Operating Systems Using Virtual

Appliances and Distributed Version Control. 41st ACM Technical Symposium on Computer Science Education (SIGCSE 2010), March 2010

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE Greg. Conceitos de SistemaOperacional. Sexta Edição. Elsevier, 2001.

TANENBAUM, Andrew. S. Sistemas Operacionais Modernos. LTC, Segunda Edição, São Paulo: Prentice-Hall, 2003. (Cap. 2).



Desenvolvimento de um ambiente colaborativo para ensino de geometria usando o Facebook

Wanessa Ferreira de Lima¹, Paulo S. G. de Mattos Neto¹, Sônia Fortes¹

¹Universidade de Pernambuco (UPE) Campus Garanhuns – PE – Brasil

{wanessa.upe, fortes.sonia}@gmail.com, paulo.salgado@upe.br

Resumo. Este artigo apresenta um ambiente colaborativo construído a partir do Facebook para o ensino da Geometria, visando facilitar a aprendizagem dessa disciplina. Os resultados parciais mostram que o uso do ambiente colaborativo a partir de recursos como vídeos, arquivos em texto, software e a comunicação entre estudantes e professor ajuda no aprendizado dos alunos e na interação entre professor, aluno e conhecimento.

Abstract. This paper presents a collaborative environment developed from the Facebook for teaching of the Geometry, aiming to facilitate the learning of this discipline. The partial results shows that the use of the collaborative environment from the resources, such as videos, texts, software and the communication between students and teacher helps in the learning of the students and in the interaction between teacher, students and knowledge.

1. Introdução

Diante de índices elevados referentes ao mal desempenho de alunos ante a assimilação dos conceitos matemáticos, novas estratégias vêm sendo estudadas para melhorar o ensino-aprendizagem da Matemática [GOMES 2012]. Dentre os conceitos matemáticos, alguns apresentam um grau de dificuldade mais elevado, como é o caso da Geometria [CARVALHO e ANDADE 2012]. Neste sentido, Pereira (2001) salienta que o ensino da Geometria ainda encontra-se em defasagem, sendo considerada pelos alunos e professores uma disciplina complexa.

Atualmente, as Redes Sociais têm alcançado destaque na forma de interação dos estudantes [MUÑOZ e TOWNER 2011]. A Boo-Box e a Hello Research (http://oglobo.globo.com/megazine/pesquisa-79-dos-jovens-que-usam-internet-no-brasil-saomembros-do-facebook-4473320#ixzz2CspnJAYC) realizaram uma pesquisa em 2011 com 3.427 jovens e constatou que 79% dos estudantes usam Facebook, sendo a Rede Social líder em acessos. De acordo com Muñoz e Towner (2011) 58% dos estudantes conectados ao Facebook fazem perguntas a outros estudantes sobre tarefas e projetos e 45% trocam informações sobre as provas.

Neste contexto, o Facebook pode ser um meio de promover um local para a colaboração no contexto escolar. A partir do conceito de aprendizagem colaborativa [TORRES, 2004], esse trabalho propõe um ambiente colaborativo utilizando o Facebook como ferramenta para facilitar o ensino e aprendizagem da Geometria. O trabalho está organizado como se segue. A Seção 2 apresenta o ambiente colaborativo



desenvolvido. Na Seção 3, o *locus* da pesquisa é apresentado, bem como os resultados parciais da pesquisa. Na Seção 4, as considerações finais são discutidas.

2. Ambiente Colaborativo Proposto

O ambiente colaborativo utiliza como base o Facebook e foi desenvolvido seguindo a realidade da disciplina de Matemática. O espaço conta com grupos para postagens de vídeos, listas e textos para problematização e arquivos diversos referentes ao conteúdo ministrado. Ainda existe um mural para postagens de softwares e outros conteúdos de interesse.

O espaço oferece ferramentas de suporte para discussão entre professor-aluno e aluno-aluno, envio de mensagens e ambiente para postagens de fotos referentes a algum evento realizado pela turma. O ambiente pode ser integrado a outras Redes Sociais e ainda permite o acesso ao conteúdo através de dispositivos móveis.

A Figura 1 mostra o ambiente colaborativo desenvolvido e as ferramentas que são exploradas para a melhoria do ensino-aprendizagem da Geometria, como a ferramenta de bate-papo, a Web Livrografia e a Videoteca. A ferramenta bate-papo possibilita que os alunos tirem dúvidas de forma rápida com o professor, ou com os outros colegas de classe. Nesse sentido, arquivos sobre o conteúdo e softwares para auxílio da aprendizagem podem ser compartilhados e comentados pelos usuários do bate-papo.



Figura 1. (a) Tela inicial do Facebook da disciplina de matemática. (b) Videoteca conjunto de vídeos relacionados com a disciplina

3. Resultados

A experiência está sendo realizada na Escola de Aplicação Professora Ivonita Alves Guerra, que faz parte da rede pública estadual de ensino, situada no município de Garanhuns. O trabalhado está sendo realizado com uma turma do 2º ano do Ensino Médio, com o apoio do professor da disciplina de Matemática. Para a validação do ambiente proposto as seguintes etapas foram executadas: sensibilização do professor e dos estudantes para o uso do ambiente, acompanhamento de comentários e "curtidas" e a verificação da interação dos alunos diante dos conteúdos disponibilizados.



A Figura 2 mostra a interação dos alunos com o ambiente colaborativo. Diante dos textos e listas os alunos apresentam suas dificuldades na realização das tarefas. Em relação às atividades recreativas e arquivos relacionados ao conteúdo estudado, os alunos se apresentam estimulados e comentam como o conteúdo ajudou a resolucionar suas dúvidas. Em relação aos vídeos, os alunos, relatam através de comentários como os recursos audiovisuais ajudam a assimilar de forma atrativa os conteúdos apresentados.



Figura 2. (a) Web Livrografia conjunto de arquivos sobre a disciplina. (b) Lista textos conjuntos de listas e textos para fixação do conteúdo.

O objetivo é fazer com que os alunos possam finalizar esta experiência, estimulados a aprender os conceitos de Geometria a partir do ambiente colaborativo. Os resultados parciais mostram que o ambiente instiga o estudo autônomo, conduzindo o aprender, por meio da busca, da investigação e da colaboração, melhorando o desempenho dos estudantes no decorrer da disciplina, facilitando a assimilação e compreensão dos conteúdos apresentados.



4. Considerações Finais

A dificuldade de aprendizagem referente aos conceitos geométricos é uma realidade vivenciada pelos alunos [PEREIRA 2001]. Desse modo, novas formas de abordar este conteúdo são de extrema relevância para o ensino-aprendizagem da Geometria. Neste contexto, o Facebook é uma ferramenta que pode ser usada para subsidiar o ensino de Geometria, já que faz parte do cotidiano do aluno e possibilita a criação de um ambiente de colaboração.

No estágio atual do trabalho, verifica-se que os alunos vêm apresentando uma interação regular a partir do ambiente colaborativo criado. A interação entre os colegas se dá através de *curtidas*, comentários às postagens realizadas e compartilhamento de arquivos. Apesar de ainda não ocorrer de forma assídua, a interação entre os estudantes e o professor tem aumentado com o passar do tempo e espera-se que esse ambiente contribua como agente facilitador para debates entre professor e alunos.

Como uma abordagem inicial, esse trabalho indica que as Redes Sociais podem ser uma alternativa para subsidiar o ensino de Matemática, envolvendo os alunos em uma experiência instigadora e enriquecedora, que vise à aprendizagem colaborativa, a interatividade e a estimulação à pesquisa e investigação diante da edificação do saber.

Referências

- CARVALHO, Maria José de Oliveira Rodrigues e ANDRADE, António Manuel Valente. (2012) Aprendizagem da geometria em b-learning no ensino básico. Educação, Formação & Tecnologias, v.5, n.1, pag. 62-71.
- GOMES, A. S. (2012) Situações Didáticas e Rede Social Educacional no Ensino de
- Matemática. III SIPEMAT- Simpósio Internacional de Pesquisa em Educação Matemática.
- MUÑOZ, C. L. e TOWNER, T. (2011) Voltar para o "muro": Como usar o Facebook em sala de aula de faculdade. First Monday. v. 16, n. 12, 2011.
- PEREIRA, M. R. O. (2001) A Geometria escolar: uma análise dos estudos sobre o seu abandono. Pontificia Universidade Católica, São Paulo. Dissertação de Mestrado.
- TORRES, Patrícia Lupion (2004) "Grupos de consenso: uma proposta de aprendizagem colaborativa para o processo de ensino-aprendizagem". Revista Diálogo

Educacional, Curitiba, v. 4, n.13, p.129-145, set./dez.



Simulação: Pseudoaleatoriedade, um estudo sobre o método do meio do quadrado

João Ferreira da Silva Júnior¹, Sérgio Francisco Tavares de Oliveira Mendonça¹, Edson Alves de Carvalho Júnior²

¹Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil

²Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco (UFRPE)

joaoferreirape@gmail.com, sftom@uag.ufrpe.br, edacjr@msn.com

Resumo. Área da matemática estreitamente relacionada com os métodos computacionais, a geração de números pseudoaleatórios é bastante discutida e tem aplicação em várias situações como por exemplo na análise de algoritmos e na criptografia. Neste resumo é discutido o método dos meios quadrados para a geração de números pseudoaleatórios.

Abstract. Field mathematics closely related to the computational methods of generating pseudorandom numbers is widely debated and has application in various situations for example in the analysis of algorithms and encryption. This is discussed in the summary of the half-square method to generate pseudorandom number.

1. Introdução

Durante o estudo do funcionamento de sistemas, encontramos algumas questões relevantes que ditam regras de inferência. Como por exemplo, pode ser extremamente dispendioso do ponto de vista financeiro replicar algum sistema para estudo prático, ou este pode ser impossível de ser copiado ou mesmo antiético. Em situações como estas torna-se necessário buscar alternativas que possibilitem o estudo e que sejam validadas pelo método científico. A partir de um modelo bem definido, que represente adequadamente o sistema, simulamos o funcionamento deste, de modo que nos permita observar seu comportamento por meio de algumas características.

No modelo de sistemas, estas características são chamadas de variáveis, e possuem comportamento aleatório com intuito de simular comportamentos bem próximos aos encontrados na natureza. Tais variáveis recebem o nome de números pseudoaleatórios. São utilizados quando é necessário que não existam relações entre eventos independentes, e é determinado como uma série numérica na qual não é possível prever o próximo número a partir de membros anteriores.



2. Números pseudoaleatórios

Computacionalmente existem várias formas de se gerar números aleatórios. Entretanto, iremos nos direcionar ao estudo do método do meio do quadrado.

Proposto por John Von Neumann o método do meio do quadrado utiliza como entrada um número composto de quatro dígitos, denominado semente, este número é então elevado ao quadrado e deste são extraídos os quatro algarismos do centro que irão formar a semente da próxima iteração, daí novamente eleva-se a semente ao quadrado e assim por diante, conforme ilustrado na "Figura 1. Método do meio do quadrado".

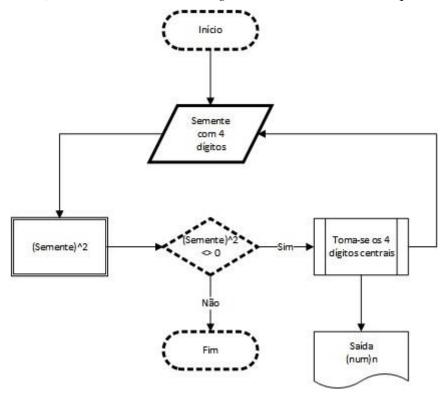


Figura 1. Método do meio do quadrado

Este é um algoritmo determinístico, pois sempre que a semente for repetida a sequência de números gerada será a mesma.

2.1. Implementação do código

O código abaixo foi implementado pelo autor para este estudo a partir da definição do algoritmo de Von Neumann e escrito na linguagem de programação Python:

```
#!/usr/local/bin/python
# -*- encoding: utf-8 -*-
'''

Geração de números aleatórios pelo método do meio do quadrado
''''

seed = int(raw_input(u"Semente inicial com quatro dígitos xxxx: "))

def meiodoquadrado(seed):
    if len(str(seed)) < 4:
        print u"Semente inválida!\r\nMenos de 4 dígitos..."
        return 0
    else:
        seed = seed ** 2
        return str(seed)[((len(str(seed))/2) - 2):((len(str(seed))/2) - 2) + 4]

i = 1</pre>
```



```
while True:
    seed = meiodoquadrado(int(seed))
    if seed == 0:
        break
    else:
        print u"#%i: %s" % (i, seed)
        i += 1
```

3. Problemática

Este algoritmo traz diversas desvantagens: a primeira é que a sequência de números gerada tende a se repetir após poucas iterações, a segunda se dá no caso quando o meio do quadrado coincidir com uma sequência de zeros, nessa situação teríamos a parada do algoritmo.

4. Solução proposta e otimização

Como proposta de solução, a cada iteração do algoritmo, deveremos somar à semente o número sequencial desta iteração elevado ao quadrado. Assim, teremos um deslocamento exponencial do alcance numérico gerado para a semente.

4.1. Investigação e testes de otimização do algoritmo

O código a seguir atende à proposta de solução. Nele omitimos a saída que informa a validade da semente quanto ao seu tamanho, ao invés de informar em tela a saída de erro, a implementação retorna para o laço onde gera nova semente e segue com o algoritmo.

```
#!/usr/local/bin/python
# -*- encoding: utf-8 -*
from math import sqrt
'''
Geração de números aleatórios pelo método do meio do quadrado
'''
seed = int(raw_input(u"Semente inicial com quatro dígitos xxxx: "))
def meiodoquadrado(seed):
    if len(str(seed)) < 4:
        return 0
    else:
        seed = seed ** 2
        return str(seed)[((len(str(seed))/2) - 2):((len(str(seed))/2) - 2) + 4]
i = 1
while True:
    seed = meiodoquadrado((int(seed) + i) ** 2)
    print u"#%i: %s" % (i, seed)
    i += 1</pre>
```

4.2. Gráfico de dispersão

Observamos que a dispersão da amostra tende a ficar uniforme conforme aumentamos o número de iterações. Nas ilustrações "Figura 2. Dispersão da amostra 1k iterações" e "Figura 3. Dispersão da amostra 10k iterações" exibimos tal efeito.

Isto se dá pelo fato de que o universo de saída da amostra é delimitado, no caso analisado, dadas as características do algoritmo dos meios dos quadrados, são gerados números entre 1.000 e 10.000, e isto faz com que após 9.000 interações haja ao menos uma repetição.



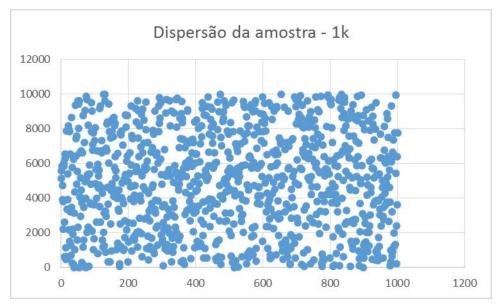


Figura 2. Dispersão da amostra 1k iterações

Quando estendemos a amostra para 10 mil iterações o efeito se torna mais evidente.

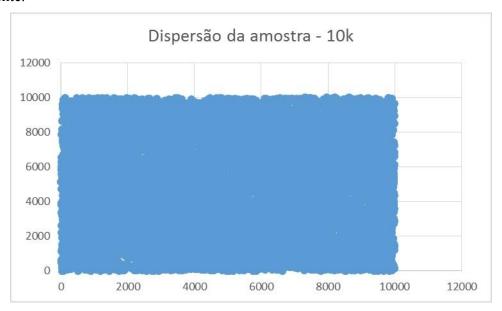


Figura 3. Dispersão da amostra 10k iterações

5. Conclusão

O trabalho está em fase de estudos. Até aqui pudemos observar que é possível, a partir de poucas modificações no algoritmo original, gerar uma grande margem de números pseudoaleatórios pelo método de Von Neumann.

Referências

Moreira, Laís Aparecida, and Rausley A. A. de Souza. "Métodos Computacionais para Geração de Sinais Aleatórios Aplicados a Sistema de Transmissão Digital." (2012).



Chwif, Leonardo, and Afonso Celso Medina. "Modelagem e simulação de eventos discretos." Afonso C. Medina, (2006).

da Rosa, Fernando Henrique Ferraz Pereira, and Vagner Aparecido Pedro Junior, and Eduardo Colli. "Gerando números aleatórios." (2002).



Inserção da mulher na Ciência da Computação no município de Garanhuns

João Ferreira da Silva Júnior¹, Sérgio Francisco Tavares de Oliveira Mendonça¹

¹ Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil joaoferreirape@gmail.com, sftom@uag.ufrpe.br

Resumo. O número de mulheres na Computação tem se mostrado cada vez menor. Na Universidade, em especial na Universidade Federal Rural de Pernambuco, em sua Unidade Acadêmica de Garanhuns, este número tem se apresentado de forma muito tímida. Em uma turma ingressante de um total de 40 estudantes, menos de 15% são mulheres. Claudia Maria Bauzer Medeiros, presidente da Sociedade Brasileira de Computação (SBC), informa que em 2005, no Brasil, dos estudantes de pós-graduação em Ciência da Computação, somente 25% eram mulheres, e entre docentes, de 25 a 30%. Em Ciência da Computação, há quatro ou cinco mulheres em turmas de 50 alunos. Espera-se como resultado, além do aumento do número de mulheres no curso de Ciência da Computação, uma maior participação de cada estudante do ensino médio, como embaixadora do curso de Ciência da Computação da UFRPE/UAG, para que estas estudantes atuem em suas escolas e comunidades como agentes de divulgação e assim, favorecer a inclusão da mulher na Ciência.

Abstract. The number of women in computing has become ever smaller. At the University, particularly in the Federal Rural University of Pernambuco, in his Academic Unit of Garanhuns, this number has appeared very timid. An entrant class a total of 40 students, less than 15 % are women. Bauzer Claudia Maria Medeiros, president of the Brazilian Computer Society (SBC), reports that in 2005, in Brazil, students graduate in Computer Science, only 25 % were women, and between teachers, 25 to 30 %. In Computer Science, there are four or five women in classes of 50 students. The expected result, in addition to increasing the number of women in the course of Computer Science, greater participation of every high school student, as an ambassador of the course Computer Science UFRPE / UAG, for these students to act on their schools and communities as agents of disclosure and thus, promote the inclusion of women in science.

1. Introdução

A presente proposta tem como premissa básica, a inserção da mulher na ciência, mais especificamente no curso de Bacharelado em Ciência da Computação, na Universidade Federal Rural de Pernambuco, no município de Garanhuns, uma vez que o número de mulheres ingressantes neste curso é inferior a 15% de um total de 40 estudantes. Para



isso, realizar-se-á ações de apresentação e ensino básico (minicursos) de conceitos iniciais das diversas áreas da computação, de hardware e de desenvolvimento lógico para a construção de programas, além de demonstrar iniciativas computacionais e empreendedoras na área do negócio computacional, para estudantes-mulheres do ensino médio, nas escolas públicas e privadas no município de Garanhuns.

Espera-se como resultado, além do aumento do número de mulheres no curso de Ciência da Computação, uma maior participação de cada estudante do ensino médio, como embaixadora do curso de Ciência da Computação da UFRPE/UAG, para que estas estudantes atuem em suas escolas e comunidades como agentes de divulgação e assim, favorecer a inclusão da mulher na Ciência.

2. Justificativa

O número de mulheres na Computação tem se mostrado cada vez menor. Na Universidade, em especial na Universidade Federal Rural de Pernambuco, em sua Unidade Acadêmica de Garanhuns, este número tem se apresentado de forma muito tímida. De uma turma ingressante de um total de 40 estudantes, menos de 15% é de mulheres. Claudia Maria Bauzer Medeiros, presidente da Sociedade Brasileira de Computação (SBC), informa que em 2005, no Brasil, dos estudantes de pós-graduação em Ciência da Computação, somente 25% eram mulheres, e entre docentes, de 25 a 30%. Em Ciência da Computação, há quatro ou cinco mulheres em turmas de 50 alunos.

Esta realidade poderia ser mudada se as meninas desde cedo fossem apresentadas ao mundo da informática, como aponta Sulamita Garcia: Algumas pessoas acreditam que mulheres têm uma predisposição genética para não gostarem de computador.

Isso é reforçado pelas mães, secretárias, namoradas que repetem com orgulho, que não entendem nada e odeiam computadores. Garcia (2004) reforça ainda, que existem muitos homens que não gostam de computadores, o que invalidaria essa ideia. Sulamita apresenta um projeto criado pelo governo da Tailândia que ensina informática nas escolas federais. Nessas aulas, os alunos aprendem, além de editoração de textos, lógica de programação, noções de hardware e programação. O resultado desse projeto é que a participação feminina na área de informática no país é maior que a masculina. Isso mostra que se as meninas parecem não gostar de computadores é por que, muitas vezes, elas não foram incentivadas. Segundo Schiebinger, a falta de modelos femininos a serem seguidos também ajuda a afastar as mulheres da área.

2. Objetivos do projeto

Apresentar a informática, não apenas apresentar ferramentas de editoração de textos, planilhas, mas principalmente, apresentar as diversas áreas da computação, lógica de programação, noções de hardware e de programação, para estudantes-mulheres que cursam os segundos e terceiros anos do ensino médio, com a intenção de inserção da mulher no desenvolvimento da ciência e tecnologia, no município de Garanhuns. Para isso, traçamos como objetivos específicos:

• Elaborar minicursos que possibilitem o esclarecimento da presente proposta, que relacione a participação da mulher na ciência e tecnologia.



- Apresentar as diversas áreas, a fim de desenvolver o interesse das estudantes participantes deste projeto, aprofundando assim, os estudos sobre o tema de escolha, além da apresentação das alunas ao ensino superior, e ao desenvolvimento das pesquisas realizadas dentro da Universidade.
- Estabelecer os princípios computacionais e lógicos, no desenvolvimento de hardware e de programas, através de linguagem de programação lúdica, até alcançar a complexidade satisfatória para o desenvolvimento de um produto.
- Aprender e desenvolver estratégias de empreendedorismo, a partir de conceitos dessa disciplina, relacionadas aos diversos casos de sucesso nos negócios para Internet.

3. Metodologia e avaliação

Partindo do Aprendizado Baseado em Problemas do cotidiano, durante os minicursos, serão apresentados problemas a fim de desenvolver nas estudantes uma estrutura cognitiva lógica para a resolução desses problemas.

Os desafios serão voltados ao aprofundamento, por parte da participante, em nível estruturado de aprendizado e de pesquisa, acerca do tema proposto, para que se possa desenvolver o interesse pela área.

Espera-se como resultado principal o aumento do número de mulheres no curso de Ciência da Computação, bem como a possibilidade destas atuarem em suas escolas e comunidades como agentes de divulgação do curso de Ciência da Computação da UFRPE/UAG, e, assim favorecer a inclusão da mulher na Ciência.

4. Resultados

O projeto conseguiu atingir seus objetivos junto à comunidade obtendo um impacto social satisfatório:

- Divulgação do projeto e proposta junto às escolas de ensino médio, da comunidade Escolar de Garanhuns
- Seleção de estudantes do ensino médio
- Minicurso de Introdução à Lógica Computacional
- Minicurso de Introdução à Ciência da Computação

5. Conclusões

Através deste trabalho foi possível observar que há um certo desconhecimento da área de Ciência da Computação por parte da comunidade do Ensino Médio, não apenas das alunas, objetivo deste trabalho, mas também dos alunos.

Há um direcionamento natural para as áreas de Medicina, Direito e Engenharia Civil. Grande parte pelo histórico familiar. Tivemos relatos de alunas que mesmo tendo estreita relação e afinidade com a computação preferiram se direcionar para o curso de Medicina por acharem que a área de computação é masculinizada ou por sofrerem pressão dos pais.



Por fim, o Projeto oportuniza à academia a possibilidade de se apresentar às futuras alunas do curso de Ciência da Computação esclarecendo assim dúvidas e fomentando o ingresso de mulheres na área.

Agradecimentos

Agradecemos à Universidade Federal Rural de Pernambuco por financiar este projeto, através de Bolsa de Extensão, Edital BEXT-2013, Síntese da Proposta SIGProj n. 135941.585.124256.09112012.

Referências

- [1] F. J. V. Passos, P. C. Braathen, M. Guerreiro, M. A. Arruda, J. C. Bohnenberger, "Programa de tutoria: uma experiência. XXIX Congresso Brasileiro de Ensino de Engenharia, CBE-013, 2001".
- [2] F. J. V. Passos, P. C. Braathen, M. Guerreiro, M. A. Arruda, J. C. Bohnenberger, "Programa de tutoria: uma esperança. XXIX Congresso Brasileiro de Ensino de Engenharia, CBE-014, 2001".
- [3] Miranda, M. A.; Barbosa, L. S. O. Ensino de Lógica e Linguagem de Programação, de autoria dos professores do Centro de Estudo Superiores de Itacoatiara da Universidade do estado do Amazonas (CESI/UEA).



Fomentando a Tecnologia da Informação e Comunicação no Agreste Meridional através de empresa júnior

Isabelle Azevedo Ferreira¹, Aline Daniele Domingos de Souza¹, Bruno Costa e Silva Nogueira¹

¹ Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil

{belleaferreira,alynedanielebcc,nogueirabruno}@gmail.com

Resumo. Este artigo apresenta as atividades realizadas pela empresa júnior TIME Jr., através de projeto de extensão, cujo principal objetivo é fomentar o desenvolvimento de tecnologias da informação e comunicação (TICs) no Agreste Meridional. A TIME Jr. é a empresa júnior do curso de Bacharelado em Ciência da Computação da Unidade Acadêmica de Garanhuns/UFRPE. Ela é uma associação civil sem fins econômicos e totalmente gerenciada por alunos e sob a orientação de professores. Sua missão é "oferecer serviços e produtos em TIC, visando o desenvolvimento econômico e tecnológico do agreste meridional pernambucano, bem como, desenvolver as habilidades profissionais de seus membros".

Abstract. This paper presents the activities of the junior enterprise TIME Jr., through extension program, whose main objective is to foster the development of information and communication technologies (ICTs) in the Agreste Meridional. TIME Jr. is the junior enterprise of the course of Bachelor of Computer Science at Academic Unit of Garanhuns / UFRPE. It is a civil non-commercial association and fully managed by students and under the guidance of teachers. Its mission is "to provide ICT services and products, targeting the economic and technological development of the Pernambuco Agreste Meridional, as well as develop the professional skills of its members."

1. Introdução

As empresas juniores estão presentes em todas as grandes universidades brasileiras, elas contribuem com uma importante parcela no desenvolvimento empresarial e econômico do país. Atualmente existe uma demanda crescente por conhecimento, e também, por profissionais capacitados em tecnologia da informação (TI) para desenvolver soluções nas mais diversas áreas, tais como: automação industrial, telecomunicações, eletrônica automotiva, etc. Para competir num mercado cada dia mais globalizado, este tipo de conhecimento é essencial para as empresas locais. Sabendo que atualmente o desenvolvimento da área de TI no agreste meridional de Pernambuco está em fase inicial, uma empresa júnior forte e atuante nesta área é, portanto, de particular importância.



2. Justificativa

Faz parte do objetivo da TIME Jr. contribuir com a sociedade no que diz respeito ao desenvolvimento econômico e tecnológico. Isso é feito capacitando seus membros nas áreas de gestão e tecnologia, assim oferecendo à sociedade profissionais diferenciados e com maior maturidade e vivência de mercado do que um aluno sem experiência profissional; prestando serviços que proporcionem ao micro, pequeno e médio empresário um trabalho de qualidade a preços acessíveis; realizando eventos e promovendo ações que incentivem o uso de tecnologias no dia a dia das pessoas e fomentem o empreendedorismo tecnológico na região.

3. Objetivos

O objetivo geral é fomentar o desenvolvimento de tecnologias da informação e comunicação no Agreste Meridional por meio de atividades vinculadas à empresa júnior TIME Jr. Para isso, foram definidos os seguintes objetivos específicos:

• Realização da 2ª edição do Prêmio TIME Jr. de Inovação.

O prêmio TIME Jr. de inovação é uma competição de caráter empreendedor e tecnológico. Nesta competição são apresentados projetos inovadores (ideias e produtos de TI) ao público da região e ao júri, que escolhe o projeto de maior potencial de sucesso. O projeto escolhido recebe o título de campeão do prêmio TIME Jr. de Inovação.

• Realização da 1ª Maratona de cursos TIME Jr.

A Maratona de cursos TIME Jr. é um evento periódico que oferece a custos acessíveis vários cursos de tecnologia para a comunidade, alunos, profissionais de TI e curiosos. Ela tem como objetivo auxiliar na formação de profissionais competentes em tecnologias de demanda no mercado, além de suprir necessidades de conhecimento dos alunos do curso de BCC/UAG.

• Manutenção e evolução dos serviços prestados aos clientes da TIME Jr.

Para o ano de 2013, a empresa júnior planejou pelo menos triplicar o seu número de cliente (anteriormente em 1 cliente), fechando até 3 contratos de prestação de serviços. Ao oferecer produtos e serviços de TI a TIME Jr. está agregando valor a sociedade, incentivando o uso de tecnologias no mundo empresarial.

• Realização do 3º processo seletivo para admissão membros para a TIME Jr.

O processo seletivo TIME Jr. surge da necessidade de captar alunos de BCC/UAG para trabalhar na empresa e assim dar continuidade ao trabalho que por ela foi iniciado. Uma vez membro da TIME Jr. o aluno desenvolverá as características necessárias para entrar no mercado de trabalho. Isso se dá através da aplicação prática do conhecimento teórico adquirido no decorrer de sua graduação e do contato com o dia a dia de uma empresa do setor de TI. Dessa forma, o aluno chega ao mercado com uma maior maturidade e vivência empresarial.



4. Metodologia e Avaliação

As atividades realizadas são acompanhadas e avaliadas periodicamente por meio de reuniões semanais envolvendo todos os colaboradores do projeto. A metodologia adotada para a realização de cada atividade é descrita a seguir:

1. Maratona de cursos TIME Jr.

Definição dos cursos; Levantamento do orçamento; Desenvolvimento do site do evento; Inscrições dos alunos; Realização dos cursos; Pesquisa de satisfação.

2. Manutenção e evolução dos serviços prestados aos seus clientes.

Manutenção dos serviços prestados aos clientes atuais; Prospecção de novas oportunidades de negócio.

3. Processo seletivo para admissão membros para a TIME Jr.

Seleção dos novos membros; Treinamento dos membros admitidos.

5. Resultados

O projeto foi desenvolvido visando melhorar e fomentar o mercado de tecnologia da informação e comunicação na cidade de Garanhuns e região, aproximando da melhor forma as tecnologias mais usadas da atualidade. Diante disso, a TIME Jr. realizou várias atividades, como as que seguem:

- Maratona de Cursos TIME Jr., com custos a um valor muito baixo, principalmente se comparado a outros cursos tecnológicos oferecidos no mercado (Figura 1).
- Eventos de divulgação do curso de Bacharelado em Ciência da Computação, das empresas juniores da cidade, sobre o empreendedorismo e sobre TICs. Com a Semana Global de Empreendedorismo foram apresentadas à população a TIME Jr. e as outras empresas juniores das instituições UFRPE, UPE e AESGA. Outras atividades foram palestras em escolas públicas e órgãos públicos, dando ciência a vários segmentos da importância da tecnologia e como usá-la para melhorar processos e gerenciamentos, para aproximação da comunidade à TIC.
- Realização do 3° e 4° processos seletivos, nos quais foram admitidos 13 membros à empresa, todos alunos do curso de bacharelado em Ciência da Computação.
- Captação de 4 novos clientes, entre pequenos empresários da região e a Câmara Municipal de Garanhuns (Figura 2), para a qual foi desenvolvido um portal de divulgação de notícias e meio de comunicação com a população da cidade.
- Parceria com outros cursos da Unidade, como Zootecnia e Medicina Veterinária para a realização do V EMVAPE - Encontro de Medicina Veterinária do Agreste Pernambucano.





Figura 3. Banner Maratona de Cursos



Figura 4. Portal da Câmara Municipal de Garanhuns

Referências

Dal Piva, A.R., Pilatti, L.A. Ferraza, DC Silva, E. Empresa Júnior: Um Laboratório de Aprendizagem como Diferencial para a Formação Acadêmica. Simpósio de Engenharia de Produção, 2006;

Júnior, B. Conceito Nacional de empresa júnior. Acessado em 12/10/2012: http://www.fundasul.br/download/ConceitoNacionaldeEmpresaJunior.pdf;

Oliveira, E.M. Empreendedorismo social e empresa júnior no Brasil: O emergir de novas estratégias para formação profissional. Franca-SP: Ribeirão Gráfica e Editora, 2003.



Olimpíadas de Matemática

Ricardo Normando B. N. Neto¹, Rodrigo Lira de Oliveira¹

¹ Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil

r1n2b3@hotmail.com, rodrigolira1@outlook.com

Resumo. A Proposta resume-se em preparar alunos do ensino fundamental e médio das Escolas públicas de Garanhuns, interessados em participar de um grupo de estudos em matemática na Universidade Federal Rural de Pernambuco - Unidade Acadêmica de Garanhuns, sob a Coordenação do Prof. Ricardo Normando e com a participação de outros professores da área de matemática da mesma Unidade com o intuito de aprofundar os conteúdos de matemática básica através da metodologia de Resolução de Problemas propostos nos materiais distribuídos aos professores e Escolas da rede Pública de todo país pelo IMPA-Instituto de Matemática Pura e Aplicada para treinamento dos alunos participantes da OBMEP-Olimpíadas Brasileiras de Matemática das Escolas Públicas.

Abstract. The proposal boils down to prepare elementary students and middle public schools Garanhuns interested in joining a group of studies in mathematics at the Federal Rural University of Pernambuco - Academic Unit of Garanhuns, under the coordination of Prof. Ricardo Normando and with the participation of other teachers in the area of mathematics in the same unit with the aim of deepening the content of basic math through the methodology Troubleshooting proposed in materials distributed to teachers and public schools across the country by IMPA-Institute of Pure and Applied Mathematics for the training of students participating OBMEP-Brazilian Olympics of Mathematics Public Schools.

1. Introdução

A Universidade Federal Rural de Pernambuco é parceira do Instituto de Matemática Pura e Aplicada (IMPA) na implementação do projeto OBMEP (www.obmep.org.br). A OBMEP, representada pelo Professor Ricardo Normando B. N. Neto (UAG/UFRPE), na qualidade de Coordenador Regional no Estado de Pernambuco além da Professor de Matemática Jorge Antonio Hinojosa Vera (Sede/UFRPE) na qualidade de Coordenador Regional de Iniciação Científica e de vários professores que atuam diretamente na preparação dos alunos.

O objetivo deste projeto de extensão é dar um suporte aos alunos da rede pública de Garanhuns e municípios circunvizinhos que já se destacaram por sua nota na OBMEP 2012, mas não obtiveram medalhas, para que no ano seguinte, eles possam ter uma melhor colocação na olimpíada e desta forma passem a participar do projeto OBMEP



recebendo inclusive incentivos financeiros (bolsa) do CNPq além de aulas preparatórias de matemática que lhes auxiliem em sua futura formação acadêmica.

2. Justificativa

Para justificar nosso projeto, cabe antes explicar o que é a Olimpíada Brasileira de Matemática das Escolas Públicas (OBMEP), pois ela é a base para a existência, necessidade e implementação desse nosso pequeno projeto local:

A Olimpíada Brasileira de Matemática das Escolas Públicas - OBMEP é uma competição nacional de Matemática entre as escolas públicas, seguida de programa de Iniciação Científica Jr. para alunos premiados.

A OBMEP é uma promoção do Ministério da Educação (MEC) e do Ministério da Ciência e Tecnologia (MCT) e é realizada pelo Instituto Nacional de Matemática Pura e Aplicada (IMPA) e pela Sociedade Brasileira de Matemática (SBM).

Já em sua 8a edição neste ano de 2012, vem contribuindo de forma bastante significativa no aprimoramento do estudo de matemática das escolas públicas, haja vista a participação maciça das Escolas Públicas e de seus respectivos alunos.

A OBMEP é dirigida aos alunos de 5ª à 8ª série (6º ao 9º ano) do Ensino Fundamental e aos alunos do Ensino Médio das escolas públicas municipais, estaduais e federais, que concorrem a prêmios de acordo com a sua classificação nas provas. Professores, escolas e Secretarias de Educação dos alunos participantes também concorrem a prêmios.

A OBMEP premia alunos, professores, escolas e secretarias de educação. Essa premiação baseia-se exclusivamente no resultado das provas da Segunda Fase. As notas da Primeira Fase não são utilizadas na classificação final.

A premiação de alunos totaliza 500 (quinhentas) medalhas de ouro, 900 (novecentas) medalhas de prata, 1800 (mil e oitocentas) medalhas de bronze. Além disso, serão concedidos certificados de Menção Honrosa a até 30.000 alunos.

A justificativa do nosso projeto segue de maneira natural, visando o aprimoramento do aprendizado em matemática dos estudantes das escolas públicas de Garanhuns, com vistas à obtenção de bons resultados nessa Olimpíada Nacional. Desta forma, estamos promovendo a cidadania e inclusão social, contribuindo para a transformação social da comunidade no entorno de nossa Universidade e fomentando a realização de ações de extensão na área do ensino, além disso, estamos propiciando ao nosso aluno agraciado com uma bolsa de extensão, o contato com a atividade pedagógica em sala de aula.

Vale ressaltar que várias Universidades do país estão desenvolvendo ações como esta, de forma independente, para dar oportunidade aos alunos de seu entorno, de ingressarem num projeto de tamanha abrangência e cunho social, de forma que eles possam prosseguir em sua vida acadêmica com autonomia nos estudos e processos de aprendizagem.

3. Objetivos

Geral: Estimular e promover o estudo da Matemática.



Específicos:

- 1. Contribuir para a melhoria da qualidade da Educação Básica nas Escolas Públicas de Garanhuns.
- 2. Contribuir para a integração das escolas públicas de Garanhuns com a universidade.
- 3. Promover a inclusão social por meio da difusão do conhecimento.
- 4. Estimular a participação de nossos professores e alunos de graduação em projetos institucionais de extensão.

4. Metodologia e avaliação

O projeto "Olimpíadas de Matemática" se desenvolverá da seguinte forma:

• A seleção dos alunos será com base nos resultados das provas aplicadas da OBMEP 2012 que estarão disponíveis no site da OBMEP (www.obmep.org.br).

Selecionaremos uma turma de 30 alunos do ensino fundamental e outra de 30 alunos do ensino médio com base em sua classificação.

Uma vez selecionados os alunos, iremos às suas Escolas de origem fazer um convite formal à direção, dirigido aos alunos para participarem do projeto.

As aulas serão dadas aos sábados pela manhã para a turma de ensino médio e à tarde para a turma de ensino fundamental pelos monitores selecionados (extensionistas do projeto) e pelo professor Ricardo Normando, coordenador do projeto.

O local onde se dará as aulas será na própria sede da Unidade Acadêmica de Garanhuns.

Os materiais didáticos provenientes do IMPA serão distribuídos gratuitamente aos alunos e monitores envolvidos.

O resultado da aprendizagem será obtido pela classificação nacional após as provas da OBMEP 2012 de 1a e 2a fases.

5. Resultados

Como já foi dito na metodologia e avaliação, esperamos que após a aplicação das provas da Olimpíada Brasileira de Matemática das Escolas Públicas no ano de 2013 nossos alunos (público alvo) tenham um bom desempenho e isto significa obtenção de prêmios, sejam eles medalhas de ouro, prata ou bronze ou ainda Certificados de menção honrosa distribuídos pelo Instituto de Matemática Pura e Aplicada (IMPA) e Sociedade Brasileira de Matemática (SBM).

6. Conclusões

Por meio deste projeto pudemos verificar que o nível dos alunos do ensino fundamental e médio das Escolas selecionadas para participar do projeto, em geral, está muito baixo, desta forma oferecer uma oportunidade como esta favorece em muito para que os alunos possam ter uma melhor preparação para enfrentarem um desafio como é essa olimpíada para eles.



As provas da 2ª fase ocorreram no dia 14 de setembro de 2013, e o resultado sairá em dezembro deste mesmo ano. Após as provas obtivemos comentários dos alunos que participaram do projeto que foi muito proveitoso para eles porque puderam ter uma idéia das questões que iriam se deparar no dia da prova.

Além disso, o projeto também oportuniza o estreitamento dos alunos da graduação com a vivência pedagógica em sala de aula, experiência ímpar de suma importância em sua vida acadêmica.

Agradecimentos

Agradecemos à Universidade Federal Rural de Pernambuco por financiar este projeto, através de Bolsa de Extensão, Edital BEXT-2013, Síntese da Proposta SIGProj n. 47206.585.57986.14102012

Referências

JURKIEWICZ, Samuel. Divisibilidade e Números Inteiros - Introdução à Aritmética Modular. IMPA, Rio de Janeiro, 2006.

CARVALHO, Paulo Cezar Pinto. Métodos de Contagem e Probabilidade. IMPA, Rio de Janeiro, 2006.

WAGNER, Eduardo. Teorema de Pitágoras e áreas. IMPA, Rio de Janeiro, 2006.

HEFEZ, Abramo. Indução Matemática. IMPA, Rio de Janeiro, 2006.

JURKIEWICZ, Samuel. IMPA, Rio de Janeiro, 2006

BANCO DE QUESTÕES 2010, 2011, 2012. IMPA, Rio de Janeiro.



Uma métrica para recomendação de federação em rede ponto-a-ponto

Wilton Oliveira Ferreira^{1,2,3}, Ricardo Batista Rodrigues², Rodrigo E. Assad³, Rafael R. de Souza², Legilmo M. F. de Oliveira¹, Josino R. Neto^{2,3}, Julio C. Damasceno²

¹C.E.S.A.R – Centro de Estudos Avançados de Recife Rua Bione, 220 – Cais do Apolo – Bairro do Recife, Recife – PE, Brasil

²Centro de Informática – Universidade Federal de Pernambuco (UFPE) Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, Recife – PE, Brasil

³Ustore

Rua do Apolo, 161, Cais do Apolo, Bairro do Recife, Recife – PE, Brazil {wof,lmfo}@cesar.org.br, {wof,rrs,vcg,jrn,jcd}@cin.ufpe.br

Abstract. This paper proposes a metric for recommendation engine for creating federations of peers in P2P, systems using as criteria the similarity on the processing capacity of the peers.

Resumo. Este artigo propõe uma métrica para mecanismo de recomendação para a criação de federações de pares em sistemas P2P, usando como critério a semelhança na capacidade de processamento dos pares.

1. Introdução

Redes ponto-a-ponto também conhecidas como P2P (*peer-to-peer*) são redes onde seus participantes são igualmente privilegiados. As ferramentas que utiliza P2P tem suas tarefas ou serviços divididos entre os participantes da rede. Os assuntos sobre P2P sempre foram discutidos nos meios científicos desde o surgimento da *Internet*. Mas só no início da década de 1990 foi que o P2P tornou-se um estilo arquitetônico [Clements et al. 2010]. Algumas ferramentas como o Napster [Parker et al. 2004] [Jiaqing et al. 2006] [Parker et al. 2004] são ferramentais que usam P2P para troca de arquivos entre os participantes e consequentemente popularizou ainda mais este estilo arquitetônico.

Inicialmente as ferramentas desenvolvidas utilizando P2P apresentavam bons desempenho, baixo custo de implementação e velocidade na implantação atendendo as necessidades da comunidade. Porém, com o sucesso destas ferramentas, muitas pessoas começaram a instalar em seus computadores e isto ampliou de forma descontrolada a rede P2P por trás de tais ferramentas.

Com o crescimento descontrolado a qualidade dos serviços começaram a diminuir, a velocidade para encontrar serviços nas redes que antes era um atrativo começou a ser um fator preocupante. Surgiu então necessidade de organizar os *peers* de forma a permitir escalabilidade, velocidade e garantia mínima de disponibilidade. Estas necessidades levaram ao desenvolvimento de pesquisas e experimentos, que observou a necessidade de mecanismos para organizar os *peers* automaticamente [Oliveira et al. 2005] [Ranjan et al. 2008].



A disposição dos *peers* de forma organizada possibilista o direcionamento de requisições de serviços de forma eficiente e consequentemente melhoria no desempenho, e também permite garantia mínima de disponibilidade. Alguns estudos propõem soluções que permitem organizar recursos computacionais em forma de federações [Duarte et al. 2010] [Mancini et al. 2009]. Este artigo apresenta um mecanismo para recomendação de *peer* na criação de federações. Além desta seção introdutória.

2. GroupUsto.re

O GroupUsto.re é uma ferramenta para recomendação na criação de federações de *peers* em sistemas P2P. Esta abordagem realiza agrupamento de *peers* que apresenta o máximo de similaridade entre si. A similaridade entre os *peers*, auxilia para que não haja gargalo nem na troca de mensagens e nem nos processamentos de dados. Este fator auxilia também para que os *peers* de uma determinada federação tenham a mesma capacidade de atender as demandas impostas por outra federação.

Desta forma, todos os *peer* serão submetidos à mesma carga de trabalho, e naturalmente terão tempo de resposta semelhante evitando espera. Além dos fatores relacionados ao desempenho, também são acrescentadas as capacidades de gestão otimizada de um nu' mero menor de *peer*, e possibilidade de direcionar fluxos de armazenamento de dados para federações especificas.

As recomendações são feitas a partir de análise no perfil e avaliação da capacidade de cada *peer*. A métrica utilizada para determinar o perfil e a capacidade individual do *peer* utiliza valores obtidos através de agentes. Cada agente presente na arquitetura ilustrada na Figura 1, atua na coleta de diferentes informações. Todas as informações coletadas pelos agentes são processadas e para cada grupo de informação é atribuído uma determinada função. Como por exemplo: todos os *peers* que estiver em um mesmo segui- mento de rede farão parte de uma mesma federação, exceto quando houver limites de *peers* por federação. E para estes casos, após formadas todas federações possíveis, serão indicadas federações com o resto dos *peers* que sobrarem de um mesmo seguimento de rede juntamente com outros que estiverem mais próximos possível. Esta proximidade será determinada pelo agente *HopAgent* apresentado na Seção 2.1.1.

2.1. Controlador de Agentes

As federações para recomendação são criadas por meio da união de *peers* em um único grupo. Os critérios para esta união é a similaridade entre os *peers*, que são mensuradas através de informações coletadas a partir de agentes. O módulo controlador de agentes tem como objetivo gerenciar os agentes descrito na Seção 2.1.1.

As funcionalidades executadas por este módulo são: enviar parâmetros de execução para os agentes; verificar se todos os agentes estão em execução; receber os dados enviados por cada um dos agentes; invocar o serviço de padronização de dados para padronizar as informações coletadas pelos agentes e entregar os dados já padronizados para o gerenciador de federação.



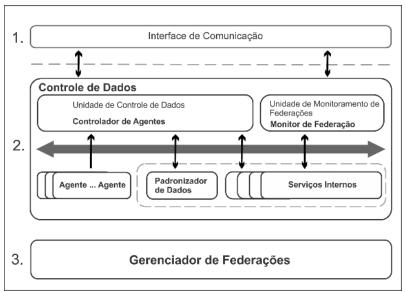


Figure 1. Arquitetura GroupUsto.re

2.1.1. Agentes

No contexto deste trabalho, os agentes são rotinas de software autônomas que tem finalidades especificas. Como já mencionado, os agentes são responsáveis por coletas as informações em cada *peer*, e transmitir tais informações para o controlador de agentes. A seguir são apresentados os agentes que compõe esta ferramenta:

- *LatenceAgent* tem como objetivo verificar a velocidade que o *peer* transferi dados pela rede;
- *MemoryAgent* tem como objetivo verificar a capacidade de memória RAM (*Random Access Memory*) disponibilizada para uso da aplicação P2P;
- ProcessorAgent tem como objetivo extrair as informações referentes ao número de núcleo de processador disponibilizado para o uso da aplicação P2P;
- **StorageAgent** tem por objetivo coletar informações referente a capacidade de armazenamento de dados do *peer*, este agente também observa a quantidade de dados já armazenado;
- *HopAgent* tem o objetivo de medir a quantidade de saltos por roteadores que uma mensagem deve percorrer para sair de um *peer* e alcançar os outros *peers*;
- AvailabilityAgent tem o objetivo de mensurar quanto tempo o peer tem disponível para o sistema;
- ReliabilityAgent tem como objetivo medir o quanto o peer é confiável com relação a todas as características e serviços por ele disponibilizados;

Todo o processo de coleta das informações por parte dos agentes pode ser parametrizado, isto permite que ao processar tais informações possam ser aplicadas ponderações. As ponderações dos valores coletados pelos agentes são realizadas através da configuração individual. Este procedimento não é padrão da ferramenta, porém, permite que valores como os coletados pelos agentes *AvailabilityAgent* e *ReliabilityAgent*, tenham maior expressividade no momento das recomendações.



3. Avaliação dos Resultados

Os resultados obtidos nestes experimentos são ilustrados na Figura 2. Como pode ser observado, houve redução da dissimilaridade em todos os casos do experimento. Em alguns casos esta redução de dissimilaridade foi maior, já em outros casos a ela foi pouco notável.

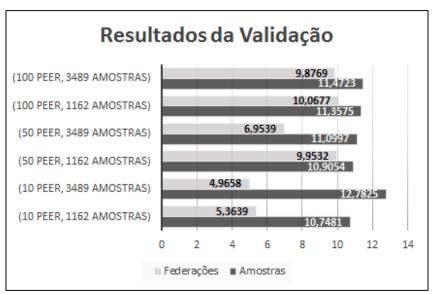


Figure 2. Resultados e Validações

4. Conclusão

Como foi observado na Figura 2, os experimentos apresentaram-se satisfatórios. Com isso conclui-se que o GroupUsto.re sugere organização de *peers* em federações de forma eficiente. E ainda observando o gráfico é possível notar que quanto maior o número de *peers* conectados, mais eficiente torna-se o GroupUsto.re.

Desta forma conclui-se que esta ferramenta atende satisfatoriamente demandas de escalabilidades do sistema.

Referências

Clements, P., Bachmann, F., Bass, L., Garlan, D., Ivers, J., Little, R., Merson, P., Nord, R., and Stafford, J. (2010). Documenting Software Architectures: Views and Beyond (2nd Edition). Addison-Wesley Professional, 2 edition.

Duarte, M., Assad, R., Ferraz, F., Ferreira, L., and de Lemos Meira, S. (2010). An availability algorithm for backup systems using secure p2p platform. In Software Engineer- ing Advances (ICSEA), 2010 Fifth International Conference on, pages 477 – 481.

Jiaqing, L., Shijie, Z., Chunjiang, W., Yiyi, D., and Xiaoqian, Y. (2006). Adaptive flooding routing algorithm in unstructured p2p. In Communications, Circuits and Systems Proceedings, 2006 International Conference on, volume 3, pages 1557–1561.

Mancini, E., Rak, M., and Villano, U. (2009). Perfcloud: Grid services for performance-oriented development of cloud computing applications. In Enabling Technologies:



Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on, pages 201 –206.

Oliveira, M., Garcia, I., and Nunes, A. (2005). Resource discovery em uma arquitetura p2p aplicado à distribuição e compartilhamento de componentes de software. In I Workshop de Redes Peer-to-Peer - WP2P 2005, pages 73–84.

Parker, D., Collins, S., and Cleary, D. (2004). Building near real-time p-2-p applications with jxta. In Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on, pages 338 – 345.

Ranjan, R., Harwood, A., and Buyya, R. (2008). A case for cooperative and incentive-based federation of distributed clusters. Future Generation Computer Systems,24(4):280 – 295.



Análise e Modelagem de Algoritmos para Eleição de Líder em Sistemas Distribuídos

Émerson R. Silva¹, Eduardo C. Julião¹, Patricia T. Endo¹

¹GRupo de Estudos Avançados em Tecnologia da Informação e Comunicação (GREAT)

Universidade de Pernambuco (UPE) – Campus Caruaru – Caixa Postal 55014-908 – Caruaru – PE – Brazil

{emersonremigio,eduardocesarbj,patriciaendo}@gmail.com

Abstract. Currently, the distributed systems area is expanding and concepts arise according to the need. In some distributed systems, a component must behave differently and this component is selected by an election process. This work aims to compare existing distributed solutions to elect leaders by modeling algorithms in different scenarios and making a comparison between the features and behaviors of them, showing their limitations and resources used in an election.

Resumo. Atualmente a área de sistemas distribuídos está em expansão e novos conceitos vão surgindo de acordo com a necessidade. Em alguns sistemas distribuídos, um componente precisa se comportar de maneira diferenciada e única. Este componente é escolhido através de um processo de eleição. Este trabalho tem como objetivo principal comparar soluções distribuídas existentes para escolha de líderes através da modelagem dos algoritmos em diferentes cenários e realizando uma comparação entre as funcionalidades e comportamentos deles, mostrando como resultado final as suas limitações e recursos utilizados na execução de uma eleição.

1. Introdução

A evolução computacional que vem acontecendo nas últimas décadas, permitiu que vários computadores pudessem comunicar uns com os outros em alta velocidade e trocar grande quantidade de dados através de redes. Existiam sistemas centralizados que para ter uma grande capacidade de processamento eram necessários *mainframes* de alto custo, mas mesmo estes ficavam sujeitos a falhas, por estarem implementados em um único local, qualquer problema neste servidor central faria com que o serviço ficasse indisponível ao usuário. Isto viabilizou o surgimento de sistemas distribuídos, que são componentes com um comportamento autônomo que se comunicam através de mensagens. Colouris, Dollimore e Kindberg (2007, p. 1) definem que: "*um sistema distribuído é aquele no qual componentes localizados em computadores interligados em rede se comunicam e coordenam suas ações apenas passando mensagens*".

Uma questão muito importante a ser tratada em sistemas distribuídos é a concorrência dos recursos utilizados pelo sistema para que todos os computadores possam utilizar determinado recurso sem que exista um conflito ou inconsistência. Segundo Tanenbaum e Steen (2007), isso é realizado pelos algoritmos distribuídos de exclusão mútua, que organizam para que os processos acessem o recurso de acordo com



a requisição e não todos ao mesmo tempo. Em sistemas distribuídos existe um único algoritmo implementado em todos os nós, porém em determinados momentos existe a necessidade de que um nó se comporte de maneira diferente (ou seja, passe a atuar como um **líder**), e para escolher o líder é necessário realizar um **processo de eleição**.

Este artigo tem como principal objetivo comparar algoritmos de eleição de líder em sistemas distribuídos realizando uma análise das variáveis com relação ao tempo e aos recursos (quantidade de mensagens) utilizados em cada algoritmo. Após análise do funcionamento de cada algoritmo nas modelagens, espera-se identificar qual tem o melhor desempenho para determinado cenário ou topologia, visando também identificar as limitações, sejam elas de implementação ou de topologia, que cada um possui.

Para tanto, este artigo está organizado da seguinte forma: a Seção 2 descreve as especificações de cada algoritmo que será analisado e mostra o funcionamento; na Seção 3 são apresentados os resultados das análises comparativas entre os algoritmos, e a Seção 4 conclui o trabalho e apresenta modificações e trabalhos futuros que podem ser implementados neste artigo.

2. Algoritmos de Eleição para Sistemas Distribuídos

Os algoritmos distribuídos muitas vezes precisam que um nó se comporte de maneira diferente, desempenhando assim uma função especifica e única para todo sistema (neste artigo, este nó será chamado de **líder**). Para escolher o nó existe um processo chamado de eleição que é executado por um **algoritmo de eleição** por todos os nós do sistema, dentre todos apenas um torna-se o líder. Alencar (1998 p. 4) explica que: "O principal requisito para algoritmos de eleição de líder é de que a escolha do elemento eleito seja única, ainda que vários nós estejam tentando eleger-se.".

O líder pode ser escolhido de acordo com vários fatores, dentre eles pode-se citar como exemplo: um endereço IP, endereço físico do nó, quantidade de processamento ou qualquer identificação única, para que cada nó seja distinto. Ao abordar o assunto de algoritmos de eleição são tomados como base de estudos dois algoritmos tradicionais, pela simplicidade de processos para eleição: o algoritmo do valentão (Garcia-Molina, 1982) e o algoritmo em anel (Tanenbaum e Steen, 2007); além de um algoritmo mais recente, proposto por (Santoro, 2007), denominado Yo-Yo.

2.1. Algoritmo do valentão

O algoritmo do valentão recebe esse nome porque o nó mais robusto vence a eleição e torna-se líder. Antes que a eleição tenha inicio há duas premissas: cada nó tem uma identificação única e conhece os vizinhos maiores que ele próprio. Então, um nó ativo qualquer vai perceber se não há líder ativo para poder iniciar uma eleição.

Segundo Garcia-Molina (1982) o nó que inicia a eleição envia uma mensagem "ELEIÇÃO" para todos os nós maiores que ele, ativos ou não (Figura 1.a), e aguarda uma resposta positiva (Figura 1.b). Caso ele receba ao menos uma resposta, seu processo de eleição é finalizado e um outro nó vai continuar com a eleição. Porém, caso nenhum nó responda, ele torna-se o líder.

Um nó pode receber a mensagem de eleição a qualquer momento, quando recebe ele envia um "OK" de volta como resposta e toma o poder de fazer a eleição novamente.



Assim em algum momento todos os nós maiores do que o nó que iniciou a eleição vão ter tomado o poder e iniciado também a eleição (Figura 2.a), após receberem as respostas "OK" (Figura 2.b) o nó maior uma hora vai realizar o mesmo processo e então vencer a eleição e mandar a mensagem "LÍDER" a todos os nós do sistema (Figura 2.c).

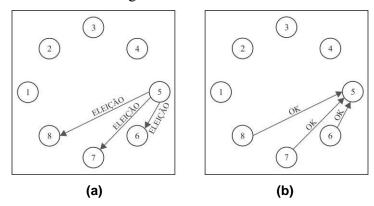


Figura 1. Iniciando a eleição com o Algoritmo do Valentão

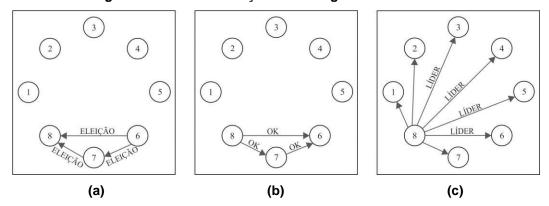


Figura 2. Nó sendo eleito líder do sistema

2.2. Algoritmo em anel

O algoritmo em anel, segundo Tanenbaum e Steen (2007), consiste na utilização de um anel em que os nós estão ordenados por ordem física ou lógica, e ao terminar a eleição elege o nó mais robusto como líder.

A eleição é iniciada quando qualquer nó percebe que não há líder ativo, porém mais de um nó pode perceber isso simultaneamente e iniciar eleições em paralelo; de toda forma, o resultado é o mesmo. O nó que percebe a ausência do líder cria uma mensagem "ELEIÇÃO" para ser repassada no anel (como mostra a Figura 3.a), que contém inicialmente a identificação do próprio nó, e a envia para o próximo nó do anel. Este acrescenta na mensagem a sua identificação e a envia novamente. Este processo ocorre até que todos os nós do anel tenham realizado esta tarefa. A mensagem de eleição para de circular quando o nó inicial recebê-la novamente (Figura 3.b) e, comparando o primeiro item da lista, perceber que a primeira identificação é a dele.

Ao fim da volta no anel, o nó que iniciou a eleição tem uma lista com a identificação de todos os nós ativos, então ele escolhe o nó com a maior identificação e realiza o processo de passar a mensagem pelo anel novamente, mas agora é uma mensagem "LÍDER" com a identificação do nó que se tornou o líder. Esta mensagem



para de circular no anel quando chegar ao nó que a criou, e então o mesmo finaliza o processo de eleição.

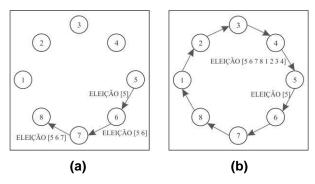


Figura 3. Iniciando eleição em anel e repassando a mensagem de eleição

2.3. Algoritmo Yo-Yo

O Yo-Yo (Santoro, 2007) é um algoritmo de busca mínima, onde o nó com menor identificação (ID) é escolhido como líder. Ele consiste em duas partes: préprocessamento e sequência de iterações.

2.3.1. Setup

A fase de pré-processamento é chamada de *setup*, onde todos os componentes tem um ID único e conhecem os de seus vizinhos, onde são criados *links* diretos apontando do menor ID para o maior, construindo assim um grafo direcionado.

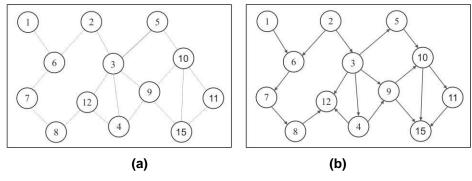


Figura 4. Organização do cenário no setup

Na Figura 4.a é mostrado o sistema inicial e na Figura 4.b o *setup* onde, após criado o grafo, cada nó já sabe o papel que irá desempenhar na eleição. Neste cenário os nós 1 e 2 serão *sources*, os nós 15 e 12 serão *sinks* e os demais nós serão *internals*. Segundo Santoro (2007) o grafo resultante é acíclico e neste grafo existem três tipos de nós: *source*, *sink* e *internal*. O *source* é um nó que é possui ID menor que todos os seus vizinhos e onde os *links* só saem dele (ele é um mínimo local); *sink* é um nó que é o maior que todos os vizinhos, onde os links só chegam nele e nenhum sai dele (ele é um máximo local); e *internal* é um nó que não é um *source* e nem um *sink*.

2.3.2. Iteração

A essência do algoritmo é uma sequência de iterações onde cada iteração consiste em um estágio da eleição onde os aptos a líder são os *sources*.



2.3.2.1 YO-

Segundo Santoro (2007), esta fase é iniciada pelos *sources* para que através dos vizinhos ele possa mandar a sua mensagem (seu próprio ID) até os *sinks*. Inicialmente, um *source* manda o valor para os *links* de saída, os nós *internals* aguardam até receber o valor de todos os *links* de entrada e em seguida pega o valor mínimo e passa adiante para os *links* de saída (Figura 5.a). Os nós *sinks*, por sua vez, aguardam receber o valor de todos os vizinhos com *links* de saída, escolhe o menor ID e inicia a segunda fase, - YO.

2.3.2.2. -YO

Segundo Santoro (2007), esta fase é iniciada pelos *sinks* para eliminar alguns candidatos a líder e transformar alguns sources em *sinks* ou *internals*. Um *sink* vai mandar para todos os vizinhos com *links* de entrada pra ele um "sim", cuja mensagem possui um ID menor, e um "não" para aqueles que tiverem mensagem com ID maior. Um nó interno espera receber a mensagem de todos os *links* de saída e se receber um "sim" de todos, ele manda "sim" para os vizinhos que tem *links* de saída para ele com menor ID e um "não" para os demais. Por fim, os *sources* esperam até receber mensagens de todos os seus *links* de saída. Se todos os votos forem "sim", ele continua ativo para a próxima iteração, se pelo menos um voto for "não", ele não mais é um candidato (Figura 5.b).

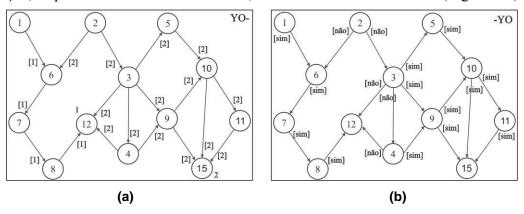


Figura 5. Sequência de iterações YO- e -YO

A Figura 5 mostra como vão ocorrer as fases YO- e -YO. Na fase YO- a mensagem com a identificação dos *sources*, nós 1 e 2, é passada vizinho a vizinho e só pode ser repassada após todos vizinhos que direcionam um *link* para o nó mandarem suas respectivas mensagens, então é feita uma comparação e segue a menor mensagem. As mensagens vão seguir por todos os nós até os *sinks* e lá é iniciada a fase -YO que é responsável por reduzir o número de *sources*. É feita uma comparação entre a mensagem que o *sink* possui e a mensagem dos vizinhos, para os que tiverem a mesma mensagem recebem um "sim" e os que não tiverem recebem um "não", como o nó 2 recebeu "não" como resposta então ele não está mais apto, acaba a eleição com o nó 1 sendo eleito líder.

3. Resultados

Utilizou-se a ferramenta NetLogo para a modelagem e simulação dos três algoritmos de eleição apresentados anteriormente: valentão, anel e Yo-Yo. O objetivo principal das



simulações é analisar como os mesmos se comportam em tempo de execução, com relação à quantidade de tempo para finalizar uma eleição e quantidade de mensagens trocadas entre os nós. Para tanto, foram utilizadas topologias de rede com 15, 30 e 60 nós com 30, 60 e 12 repetições, respectivamente.

3.1 Quantidade de tempo para execução de uma eleição

Os gráficos das Figuras 6.a, 6.b e 6.c, mostram a média da quantidade de tempo (em ticks, que representam a unidade de tempo da ferramenta NetLogo) de cada algoritmo para finalizar seu processo de eleição de líder, nos cenários com 15, 30 e 60 nós, respectivamente.

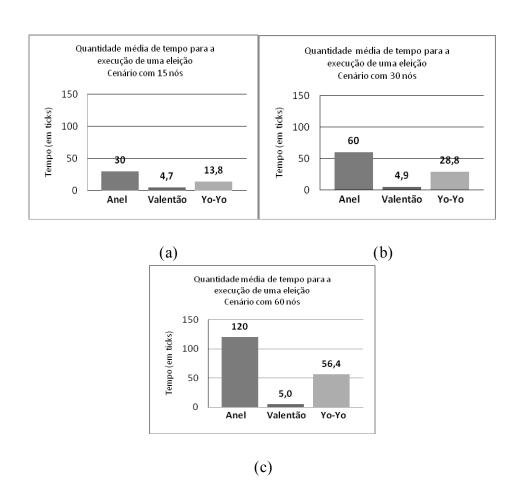


Figura 6. Gráficos de quantidade média de tempo para execução de uma eleição

Para concluir a eleição no cenário com 15 nós, em média, o algoritmo em anel demorou 30 *ticks*, o valentão demorou 4,7 *ticks* e o yo-yo demorou 13,8 *ticks*. Já no cenário com 30 nós, o algoritmo em anel demorou 60 *ticks*, o valentão demorou 4,9 *ticks* e o yo-yo demorou 28,8 *ticks*. E no cenário com 60 nós, o algoritmo em anel demorou 120 *ticks*, o valentão demorou 5 *ticks* e o yo-yo demorou 56,4 *ticks*.



3.2 Quantidade de mensagens trocadas

Já os gráficos das Figuras 7.a, 7.b e 7.c mostram a média da quantidade de mensagens trocadas entre os nós de cada algoritmo para finalizar seu processo de eleição de líder, nos cenários com 15, 30 e 60 nós, respectivamente.

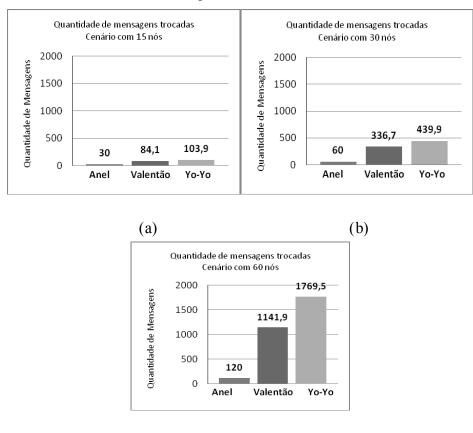


Figura 7. Gráficos de quantidade média de mensagens trocadas para eleição

(c)

Para concluir a eleição no cenário com 15 nós, em média, o algoritmo em anel trocou 30 mensagens, o valentão trocou 84,1 mensagens e o yo-yo trocou 103,9 mensagens. Já no cenário com 30 nós, o algoritmo em anel trocou 60 mensagens, o valentão trocou 336,7 mensagens e o yo-yo trocou 439,9 mensagens. E, por fim, no cenário com 15 nós, o algoritmo em anel trocou 120 mensagens, o valentão trocou 1141,9 mensagens e o yo-yo trocou 1769,5 mensagens.

3.3 Análise e Discussão

A partir dos resultados das simulações, observou-se que o algoritmo em anel é o que mais consome tempo para finalizar uma eleição, porém a quantidade de mensagens utilizada pelo mesmo é a menor. Percebeu-se também que, ao aumentar a quantidade de nós na topologia, o tempo de uma eleição do algoritmo em anel aumenta consideravelmente em relação aos outros dois algoritmos.

O algoritmo do valentão demora menos para executar uma eleição e o tempo utilizado se mantém constante independente da quantidade de nós, mas o número de mensagens enviadas para o pouco tempo que ele passa sendo executado mostra que ele



consome muito do sistema deixando a rede sobrecarregada durante a eleição. Com o tempo constante, quanto maior a quantidade de nós, mais a rede fica sobrecarregada.

O algoritmo Yo-Yo, em todas simulações, apresentou o maior número de mensagens de acordo com a quantidade de nós, de maneira que, ao dobrar a quantidade de nós, as mensagens aproximadamente triplicam. Ao analisar o funcionamento é visto que ele é um algoritmo muito síncrono, e sincronia é o principal problema para ser resolvido em sistemas distribuídos.

A Tabela 1 mostra uma comparação entre os três algoritmos com relação a tempo de execução e quantidade de mensagens.

Algoritmo	Topologia	Tempo de execução (em ticks)	Quantidade de mensagens
Valentão	Totalmente conectada	Menor dentre os três	Possui quantidade média dentre todos
Anel	Em anel	Maior dentre os três	Menor dentre os três
Yo-Yo	Em árvore	Possui tempo médio	Maior dentre os três

Tabela 1. Comparação entre os algoritmos

O algoritmo do valentão é restrito a uma topologia na qual todos os nós possuem conexões com todos; o algoritmo em anel utiliza a topologia em anel e limita-se a esta, pois para que funcione necessita de que os nós estejam ordenados em anéis físicos ou lógicos; já o algoritmo Yo-Yo utiliza topologia em árvore e limita-se a topologias acíclicas, porque ele precisa de um grafo para a execução do algoritmo de eleição. Assim, o algoritmo do valentão apresenta o melhor tempo de execução e o algoritmo em anel apresenta o melhor resultado com relação a quantidade de mensagens enviadas.

4. Conclusão e Trabalhos Futuros

Este artigo descreveu o funcionamento de três algoritmos para eleição de líder em sistemas distribuídos e mostrou as especificações e limitações de cada um. O objetivo principal deste artigo foi servir como fonte de estudo inicial de algoritmos de eleição de líder em sistemas distribuídos, mostrando que pode ser escolhido de diversas formas.

Os resultados apresentados mostraram que os algoritmos foram desenvolvidos para cada tipo de topologia e que dificilmente funcionariam em outra, então isso representa uma grande limitação dos algoritmos.

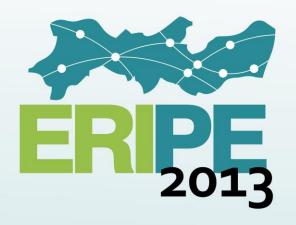
Como trabalhos futuros, pretende-se aprimorar o estudo e as modelagens dos algoritmos, comparando-os com outros algoritmos distribuídos. Também pretende-se implementar e estudar a questão de tolerância a falhas dos algoritmos e avaliar o desempenho dos mesmos.



Referências

- Alencar, J. F. "Algoritmos para eleição de líder em sistemas distribuídos", Universidade Estadual de Campinas Instituto de Computação. Campinas, 1998.
- Colouris, G.; Dollimore, J.; Kindberg, T. "Sistemas Distribuídos Conceitos e Projeto", 4. ed. Porto Alegre: Bookman Editora, 2007.
- Garcia-Molina, H. "Elections in a Distributed Computing System.", IEEE Transactions on Computers. v. C-31. n. 1. p. 48-59, 1982.
- Santoro, N. (2007) "Design and Analysis of Distributed Algorithms", Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
- Tanenbaum, A. S. e Steen, M. V. (2007) "Sistemas distribuídos: princípios e paradigmas", 2. ed. São Paulo: Pearson Prentice Hall, 2007.
- Wilensky, U. (1999) "NetLogo", http://ccl.northwestern.edu/netlogo, Último Acesso em: Setembro de 2013.

ISSN: 2317-5346



III ESCOLA REGIONAL DE INFORMÁTICA DE PERNAMBUCO GARANHUNS, PE 6-8 DE NOVEMBRO

Tema:

"Desafios da Computação: Academia-Indústria-Comunidade"

Tutoriais

ISSN 2317-5346



DE GARANHUNS











Lógica de Programação: Iniciação Lúdica com App Inventor for Android

Tancicleide C. S. Gomes¹, Jeane C. B. de Melo¹

¹Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco (UFRPE), Recife – PE – Brasil.

tancigomes@bsi.ufrpe.net, jeane.ufrpe@gmail.com

Resumo. Os desafios encontrados no processo de ensino-aprendizagem de lógica de programação resultam na busca por métodos, técnicas e ferramentas que tornem este processo mais intuitivo e divertido. Neste contexto, o presente trabalho apresenta uma proposta metodológica cuja abordagem envolve as possibilidades trazidas pela linguagem de programação visual App Inventor e as características intrínsecas dos jogos.

Abstract. The challenges in the process of teaching and learning of programming logic results in the search for methods, techniques and tools that make this process more intuitive and fun. In this context, this paper presents a methodology whose approach involves possibilities brought by the visual programming language App Inventor and the intrinsic characteristics of the games.

1. Introdução

O processo de ensino-aprendizagem dos conteúdos de lógica de programação apresenta algumas dificuldades a serem tratadas: a falta de motivação dos estudantes por considerarem o assunto de difícil compreensão, exigindo um nível de abstração na solução dos problemas que os mesmos ainda não desenvolveram, aliados às metodologias comumente utilizadas, as quais não consideram as transformações na aquisição do conhecimento dos novos aprendizes, dentre outros problemas relatados na literatura [Gomes e Melo 2013].

Diante deste contexto, novas práticas e métodos de ensino-aprendizagem têm sido propostos buscando superar tais obstáculos, facilitar a aprendizagem e atrair novos talentos para Computação. O presente trabalho corrobora com este cenário, apresentando uma proposta metodológica que se apropria das possibilidades didáticas de uma linguagem de programação visual, a saber, o App Inventor, para através de uma abordagem com ênfase prática apresentar os conteúdos de lógica de programação para iniciantes. Adicionalmente, a abordagem é apoiada em dois elementos considerados relevantes para a geração atual de aprendizes, os ditos nativos digitais: os dispositivos móveis e os jogos digitais. Os nativos digitais possuem significativa familiaridade com as tecnologias da informação e comunicação (TICs), uma vez que cresceram cercados por múltiplos recursos tecnológicos, exigindo assim uma formação que atenda às suas necessidades específicas, não se adequando à simples apresentação e memorização de conteúdos comumente empregadas [Prensky 2001].



Os dispositivos móveis, cada vez mais presentes no cotidiano da sociedade, 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]. Além disto, os jogos digitais podem ser definidos como ambientes atraentes e interativos, os quais capturam a atenção do jogador ao oferecer desafios que exigem níveis crescentes de destreza e habilidades [Savi e Ulbricht 2008]. O presente trabalho se utiliza da ludicidade inerente de tais recursos como um fator propulsor da motivação dos estudantes em aprender lógica e programação.

Este artigo traz uma proposta metodológica para o ensino de lógica de programação através da criação de jogos de pouca complexidade para dispositivos móveis Android. A linguagem visual de programação App Inventor é apresentada na Seção 2. A Seção 3 apresenta a fundamentação teórica e a estruturação metodológica da presente proposta. A experiência desenvolvida e as considerações finais são discorridas na Seção 4.

2. App Inventor: Novos Caminhos?

O atual cenário revela uma preocupação recorrente em relação ao ensino de lógica de programação. Diante disto, surge a linguagem de programação visual App Inventor (AI), projetada com a missão de provocar mudanças positivas nas experiências introdutórias de programação. A partir de uma premissa que enfatiza "[...]as pessoas e suas interações com os outros e com o mundo ao seu redor, em detrimento de cenários que incluem estudantes desmotivados, sentados em laboratórios escrevendo programas que ninguém jamais usará" [Abelson 2009], o App Inventor 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.

O desenvolvimento de uma aplicação nesta ferramenta é realizado através de duas janelas: *App Inventor Designer* e *Blocks Editor*. A janela *App Inventor Designer* é executada a partir do navegador e permite criar visualmente a interface do usuário, ao clicar e arrastar os componentes da *Palette*, tais como botões, caixas de texto, figuras, animações, sons, dentre outros, para o *Viewer* (Figura 01).

Na janela *Blocks Editor* (Figura 02) é manipulado o comportamento dos componentes selecionados na janela *App Inventor Designer*. Neste ambiente, o usuário encontra blocos conectáveis, em uma interface do tipo arrastar e soltar, que operam *strings* e listas, realizam ações de controle (*e.g. if, else, foreach*, etc.) e operações matemáticas, dentre outras funcionalidades. A fim de executar as aplicações criadas, o usuário pode utilizar qualquer dispositivo *Android* (*smartphone* ou *tablet*) que esteja



conectado ao computador, ou mesmo, através do emulador que acompanha o *Blocks Editor*.

Uma ampla variedade de materiais de estudo e referências podem ser encontradas no site oficial [MIT 2012], abrangendo desde a instalação, a criação do primeiro aplicativo, bem como uma série de tutoriais e os respectivos *códigos*, criados por professores e estudantes de várias instituições de diversos países, assim como muitos relatos de experiência de uso do *App Inventor* nos mais variados contextos.



Figura 5. App Inventor Designer

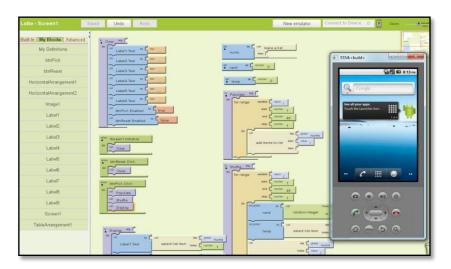


Figura 6. App Inventor: Janela Blocks Editor. Exemplo de Aplicativo e Emulador em Execução

3. Aprendendo Lógica de Programação com App Inventor e Jogos

O cenário de desafios que envolve o processo de ensino-aprendizagem de lógica de programação tem sido alvo de diversos esforços em pesquisa, algumas destas iniciativas envolvem o uso de jogos digitais para o ensino de conteúdos específicos de lógica de programação e algoritmos.

ISSN: 2317-5346



Visando desenvolver o raciocínio lógico-matemático no ensino fundamental, temse o *Terra das Corres* [Leitão *et al.* 2012]. Ainda voltados ao mesmo público, exploram o conceito de números binários, o *SokoBin* [Alencar, Scaico e Silva 2012], projetado para ser utilizado como um complemento as aulas de Matemática e Informática, e o *Computino* [França, Silva e Amaral 2013]. Com vistas a atender o ensino superior com o tema de algoritmos de ordenação, tem-se como exemplo, o *Sortia* [Battistella, von Wangenheim e von Wangenheim 2012].

Os jogos digitais aparecem como um recurso didático cujas características podem trazer diversas contribuições ao processo de ensino-aprendizagem, como por exemplo [Savi e Ulbricht 2008]: *Efeito Motivador* — Os jogos demonstram ter alta capacidade de entreter e divertir pessoas ao mesmo tempo que proporcionam o aprendizado, desta forma a inserção dos componentes de prazer e diversão no processo educacional é relevante, pois com o aprendiz mais relaxado há maior disposição e recepção para o aprendizado. *Facilitador do Aprendizado* - Ao jogar, o aprendiz exerce a responsabilidade de tomar decisões e está exposto a níveis crescentes de desafios, buscando exercitar a resolução de problemas, o raciocínio dedutivo e a memorização.

Convém mencionar que os jogos desenvolvidos neste contexto, embora estejam envolvidos no processo de aprendizagem, não visam apresentar conteúdos educacionais. Ou seja, a aprendizagem se consolida não por meio do uso do jogo, mas no seu processo de criação, de modo que o processo de aprendizagem se dá em criar os jogos e não em utilizá-los.

Outra questão relevante relacionada à motivação diz respeito ao modo como os conteúdos são abordados através dos jogos, pois a criação de um aplicativo onde um personagem interage com o meio, ganha ou perde pontos é mais atraente do que criar aplicações envolvendo conceitos matemáticos que eles já têm dificuldade desde o ensino médio. Envolver o simples uso destes conteúdos pode tornar a atividade de aprender lógica de programação uma experiência frustrante, visto que o estudante associa o aprendizado novo, de lógica de programação, ao, na maioria das vezes, deficiente, aprendizado de matemática, ciência que também fundamenta o aprendizado de programação.

Desta forma, a presente proposta metodológica apropria-se da integração das possibilidades oferecidas pelo paradigma de programação pelo usuário final e a criação de jogos digitais, para enriquecer e tornar mais divertida a apresentação dos fundamentos de lógica de programação. Além disto, a abordagem com ênfase prática corrobora para a maturação do conhecimento apreendido através do reforço das atividades, consolidando informações recém incorporadas. Os conteúdos apresentados se distribuem em quatro curtos módulos: 1. Introdução ao Desenvolvimento de Aplicativos para Dispositivos Móveis, 2. Linguagem de Programação Visual App Inventor, 3. Testes e por fim, 4. Publicando seus Aplicativos na Web.

A presente proposta é fruto de experiências de ensino de programação realizadas no âmbito do projeto Aplicativos Educacionais para Dispositivos Móveis, que atendeu estudantes do Ensino Médio de escolas da rede pública estadual de Pernambuco [Gomes e Melo 2012] e do projeto EduMobile, que ao longo de 2013 tem atendido estudantes do ensino médio e superior [Gomes e Melo 2013].



O módulo *Introdução ao Desenvolvimento de Aplicativos para Dispositivos Móveis* tem por objetivo apresentar uma breve introdução aos desafios, oportunidades e possibilidades na produção de aplicativos para dispositivos móveis a fim de motivar o estudo de lógica de programação e, por conseguinte, o aprimoramento através do aprendizado de linguagens de programação comerciais.

O módulo 02 - A Linguagem de Programação Visual App Inventor, apresenta a ferramenta App Inventor, sua interface e seus principais componentes. Neste módulo, os conteúdos de lógica de programação são apresentados à medida que os componentes são explorados, assim, mesmo em um nível inicial, os aprendizes são levados a criar jogos que abordem os conteúdos propostos, cuja complexidade aumenta ao passo que os assuntos avançam (Tabela 01).

Tabela 1. Aplicativos Criados versus Conteúdos Associados

Aplicativos	Conteúdos Associados
App 01. MiniPaint	Constantes / Variáveis
App 02. Toupeira	Constantes / Variáveis
	Funções / Procedimentos
App 03. Quiz	Constantes/ Variáveis
	Operações Matemáticas e Lógicas
	Operações com Strings
	Funções / Procedimentos
	Estruturas de Decisão/ Repetição
	Listas
App 04. SpaceInvader	Constantes / Variáveis
	Operações Matemáticas e Lógicas
	Funções / Procedimentos
	Estruturas de Decisão/ Repetição
	Funcionalidades Mobile Específicas

Durante o módulo *Testes*, os participantes são convidados a testar as aplicações desenvolvidas tanto em um *smartphone* quanto no emulador acoplado ao *Blocks Editor*.

A fase de Testes é muito importante, principalmente por dois aspectos: 1. Permitir ao aprendiz testar a lógica desenvolvida na aplicação e obter o *feedback* imediato a fim de avaliar a corretude da solução proposta, 2. Identificar falhas e/ou necessidades de melhorias referentes à interface da aplicação que possam gerar desconforto no uso ou induzir o usuário a cometer erros.

O quarto módulo, *Publicando seus Aplicativos na Web*, visa familiarizar os estudantes com o processo de instalação de aplicativos em *smartphones*, o uso de *QRCodes* e o compartilhamento dos aplicativos criados com outros estudantes, e por conseguinte, compartilhar o conhecimento adquirido ao longo do curso.



4. Resultados Obtidos e Considerações Finais

Ao fim do curso, os aprendizes tiveram a oportunidade de criar pelo menos quatro jogos que abrangiam os principais tópicos de lógica de programação. Uma vez que os estudantes não necessitavam aprender uma sintaxe em específico e a programação dos componentes era facilitada pelo uso da linguagem visual, os estudantes demonstraram interesse pela possibilidade de modificar alguns aspectos da aplicação à medida que se apropriavam dos conceitos apresentados e se familiarizavam com a ferramenta, realizando tais atividades de maneira criativa.

Assim, ao longo do curso, os aprendizes inseriam outras funcionalidades, regras e elementos nos jogos, modificavam componentes e no fim, o resultado era um jogo único, no sentido de muitas de suas características terem sido bastante modificadas.

No decorrer da criação de aplicativos com o *App Inventor*, surge um novo elemento em particular: a interface da aplicação. Este é um aspecto que comumente não é relevante em práticas de ensino de lógica de programação, ora devido à linguagem de programação utilizada, ora em relação às limitações de conhecimento dos aprendizes. Embora seja um aspecto adicional, os aprendizes demonstraram estar motivados com a possibilidade de aprimorar o layout da interface de suas aplicações, personalizando livremente botões, fontes, organização dos elementos na tela, dentre outras.

No decorrer da realização das atividades propostas, pôde-se constatar que os estudantes aprenderam os conteúdos explanados, bem como seus discursos enunciaram motivação, engajamento e interesse pelos conceitos apresentados.

Referências

- Abelson, H. (2009) "App Inventor for Android". http://googleresearch.blogspot.com/2009/07/app-inventor-for-android.html, Março.
- Alencar, Y. M., Scaico, P. D., Da Silva, J. C. (2012). Jogando com Números Binários: uma Possibilidade para Estimular o Raciocínio Lógico e o uso da Matemática. Conferencias LACLO, v. 3, n. 1.
- Battistella, P. E., von Wangenheim, A., von Wangenheim, C. G. (2012). SORTIA-Um
- Jogo para Ensino de Algoritmo de Ordenação: Estudo de caso na Disciplina de Estrutura de Dados. In: Anais do XXIII Simpósio Brasileiro de Informática na Educação.
- França, R. S., Silva, W. C., Amaral, H. J. C. (2013). Computino: um jogo destinado à aprendizagem de Números Binários para estudantes da educação básica. In: Anais do XXI Workshop sobre Educação em Computação XXXIII Congresso da Sociedade Brasileira de Computação. Maceió, AL Brasil.
- Gomes, T., Melo, J. (2012). Lógica de Programação com App Inventor: Um relato de experiência no nível médio. In: XII Jornada de Ensino, Pesquisa e Extensão da Universidade Federal Rural de Pernambuco. Recife, PE.
- (2013). O Pensamento Computacional no Ensino Médio: Uma Abordagem Blended-Learning. In: Anais do XXI Workshop sobre Educação em Computação XXXIII Congresso da Sociedade Brasileira de Computação. Maceió, AL Brasil.



- Leitão, A. H. B, Gonçalves, G. G. A., Ribeiro, W. M., de Oliveira, B. C., do Nascimento Salgueiro, V. S., e Madeiro, F.(2012). Terra das Cores: Uma Proposta de Jogo Educacional Infantil para o Exercício do Raciocínio Lógico-Matemático. In Anais do XXIII Simpósio Brasileiro de Informática na Educação.
- MIT Massachusetts Institute of Technology. (2012) "MIT App Inventor", http://appinventor.mit.edu/, Novembro.
- Morelli, R. (2012). "CPSC: Computing with Mobile Phones", http://www.cs.trincoll.edu/~ram/cpsc110/, Novembro.
- Prensky, M.(2001). "Digital Natives, Digital Immigrants". MCB University Press, http://www.marcprensky.com/writing/Prensky%20%20Digital%20Natives,%20Digit al%20Immigrants%20_%20Part1.pdf
- Savi, R.; Ulbricht, V. R. (2008). Jogos Digitais Educacionais: Benefícios e Desafios. Revista Novas Tecnologias na Educação, v. 6, n. 2, p. 10.
- Tyler, J. (2011). App Inventor for Android: Build Your Own Apps No Experience is Required. Wiley, 1^a edição.
- Valentim, Hugo (2009). Para uma Compreensão do Mobile Learning. Reflexão sobre a utilidade das tecnologias móveis na aprendizagem informal e para a construção de ambientes pessoais de aprendizagem. Tese de mestrado em Gestão de Sistemas de eLearning, Universidade Nova de Lisboa, Lisboa
- Wolber, D. (2012). "AppInventor.org", http://www.appinventor.org/course-in-a-box, Fevereiro.
- Wolber, D., Abelson, H., Spertus, E. and Looney, L. (2011). App Inventor: Create Your Own Apps. O'Reilly Media, 1^a edição.



Startups Dirigidas à Inovação de Software: Da Universidade ao Mercado

Ricardo Batista Rodrigues¹, Ricardo T. A. de Oliveira², Rafael Roque de Souza²

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE) Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, 50.740-560, Recife, PE, Brasil

²DEInfo - Universidade Federal Rural de Pernambuco (UFRPE) Rua Dom Medeiros, s/n, Dois Irmãos, CEP-52.171-900, Recife, PE, Brasil

rbr@cin.ufpe.br,{ricardotavares.gm,rafaelmarlin}@gmail.com

Resumo. Este trabalho tem como objetivo apresentar o conceito de Startup e os principais métodos, processos, técnicas e ferramentas utilizadas na atualidade por empreendedores de sucesso. A ideia é detalhar como iniciar um startup, desde a elaboração de um modelo de negocio ao desenvolvimento de um produto mínimo viável (MVP), incubação e aceleração, apresentar técnicas e ferramentas que podem maximizar as chances da criação de negócios bem sucedidos e o papel do investidor anjo. Por fim, mostrar porque errar rápido e errar barato tornou-se o caminho trilhado por vários startups de sucesso e como adotar tal técnica em seu startup.

Abstract. This paper aims to introduce the concept of Startup and the main methods, processes, techniques and tools used today by successful entrepreneurs. The idea is to detail how to start a startup, from the preparation of a business model to develop a minimum viable product (MVP), incubation and acceleration, present techniques and tools that can maximize the chances of creating successful businesses and the role of angel investor. Finally, to show why err cheap and fast became the path trodden by several successful startups and how to adopt this technique in your startup.

1. Introdução

O empreendedorismo junto a todo o ecossistema de startups vem ganhando força dentro das universidades, que se consolidam como instituições de fomento à inovação em novos negócios. Podemos considerar ainda que o contexto atual é bastante favorável ao crescimento do empreendedorismo e da inovação, enquanto alternativa a oportunidades provindas do aquecimento em diversos setores da economia, incluindo, mas não se limitando, à área de Tecnologia da Informação e Comunicação (TIC).

É em busca de novos conhecimentos e soluções inovadoras que os países e universidades vêm investindo em centros de pesquisas e tecnologia que deem sustentação para que esse tipo de empreendimento cresça e possa desenvolver soluções que resolvam problemas reais do dia a dia.



Segundo Eric Ries (2012), startup é uma instituição humana projetada para criar novos produtos e serviços sob condições de extrema incerteza. Os startups também possuem um norte verdadeiro, um destino em mente: criar um negocio próspero e capaz de mudar o mundo. Para alcançar essa visão, os startups empregam uma estratégia, que inclui um modelo de negócios, um plano de produto, um ponto de vista a cerca dos parceiros e dos concorrentes, e as ideias a respeito de quem são os clientes. O produto é o resultado final dessa estratégia, como mostra a Figura 1.

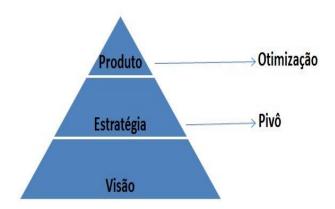


Figura 7. Processo de Startups [Ries, 2012]

Um startup não precisa ser um negocio novo no sentido de um novo CNPJ, novo no sentido de se criar uma nova empresa, um novo negocio pode ser algo que rola numa empresa existente, como um novo produto ou serviço. Mas pode ser novo, mesmo no sentido de estar começando agora, do zero, com você desenhando a primeira versão do que vai ser sua proposta de valor para o mercado. Já a noção de "inovadores de crescimento empreendedor" descreve negócios [ou produtos, ou serviços] que mudam comportamentos [inovação é a mudança do comportamento de agentes, nos mercados, como fornecedores e consumidores de produtos e {ou} serviços] e que crescem num ritmo bem maior do que o mercado em que estão. O papel do empreendedor, no processo de crescimento do negócio, é fazer o seu negócio crescer muito mais rápido do que o mercado [Meira, 2013].

Este artigo está organizado da seguinte forma: seção 2 apresenta o processo de inicialização de um startup. A seção 3 apresenta as principais técnicas utilizadas no desenvolver de um startup. A seção 4 apresenta as conclusões.

2. Lean Startup

Desenvolver um startup é um exercício de desenvolver uma instituição, portanto, envolve necessariamente administração. Muitas vezes, isso surge como uma grande surpresa para os aspirantes a empreendedores, pois suas associações com essas duas palavras são diametralmente opostas [Ries, 2010].

No processo Lean Startup, conforme a Figura 2, a partir das idéias você constrói um produto mínimo viável (código), mede os resultados, coleta dados e aprende alguma lição. E continua a executar este laço de aprendizagem, o mais rápido possível, fazendo ajustes até atingir o casamento do produto com o mercado ou mudar algum item do modelo de negócios fazendo o pivô e começando tudo de novo. O objetivo é conseguir



um modelo de negócio de valor, ou seja, que deixe o cliente feliz e gere lucro [Grando, 2012].

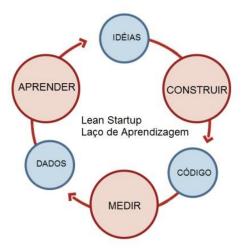


Figura 8. Processo The Lean Startup [Ries, 2012].

A idéia de Lean Startup é fazer tudo da forma mais simples possível, usando o mínimo de velocidade para economizar dinheiro e diminuir riscos. Esta empresa enxuta começa com um produto mínimo viável e através de um processo iterativo de aprendizagem e validação qualitativa busca o ajuste do produto ao mercado para só então crescer em escala e estrutura.

3. Técnicas

Nesta seção apresentaremos algumas das técnicas mais utilizadas por idealizadores de startups.

3.1. O Business Model Canvas

O Business Model Canvas é uma ferramenta de gerenciamento estratégico, que permite desenvolver e esboçar modelos de negócio novos ou existentes. É um mapa visual préformatado contendo nove blocos do modelo de negócios, como mostra a Figura 3.

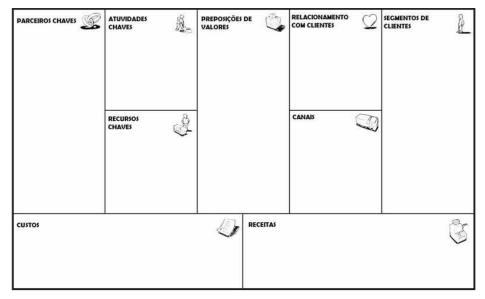




Figura 9. Modelo Business Model Canvas.

O objetivo do modelo Canvas é simplificar o árduo trabalho de criar um plano de negócio, que na maioria das vezes se torna uma tarefa cansativa e complexa. O modelo Canvas é uma ferramenta ideal para novos empreendedores que estão iniciando seus startups. A seguir detalhamos cada item do modelo Canvas apresentado na Figura 3

- Parceiros Chaves: neste item devem ser descritos as instituições ou pessoas que são parceiros da Startup, contribuindo de forma direta ou indiretamente com o modelo de negócio da Startup, por exemplo, a empresa que irá fornecer os servidores para hospedagem do site da Startup.
- Atividades Chaves: este item, são as principais atividades que devem ser desenvolvidas pelo time da Startup, para que seja produzido o produto objetivado no projeto, por exemplo, desenvolver o site da Startup.
- **Preposições de Valores:** são os valores que serão agregados com o projeto, o diferencial do produto com os demais similares a ele, por exemplo, uma nova tecnologia como todos os serviços rodando em nuvem.
- **Relacionamento com Clientes:** são os meios em que a Startup se relacionara com os clientes, por exemplo, atendimento via telefone.
- Seguimentos de Clientes: neste item devem ser apresentados quais são os clientes alvo do produto, por exemplo, crianças e idosos ou pequenos empresários.
- **Recursos Chaves:** são todos os recursos que o time da Startup já possui para o desenvolvimento do projeto, por exemplo, mão de obra qualificada.
- Canais: são os meios por aonde o produto chegará ao cliente alvo, por exemplo, redes sociais ou o próprio site da Startup.
- Custos: neste item são descritas todas as despesas do projeto, como por exemplo, contratação de pessoal.
- Receita: neste item são descritas as fontes de receitas do projeto, de onde e como serão extraídas as receitas para a manutenção e o lucro da Startup, por exemplo, venda de assinaturas.

3.2. MVP

Na metodologia de startups, a primeira versão do produto deve ser um produto mínimo viável ou MVP, ou seja, o produto mínimo concebível que pode encontrar um conjunto de clientes que estão animados o suficiente para usarem e pagarem pela visão de longo prazo do produto ideal [Ries, 2012].

O primeiro objetivo em criar um produto de sucesso deve ser encontrar os usuários visionários ou evangelistas que querem e precisam do produto. Esses usuários serão capazes de ter a visão final do produto, por isso vão ignorar as falhas temporárias e acabarão ajudando a aprimorar o produto da Statup [Ries, 2012].

A idéia por trás do produto mínimo viável é que uma Startup pode eliminar o desperdício limitado a primeira versão de um produto para as funcionalidades absolutamente essenciais que validam a visão de longo prazo e as hipóteses fundamentais da Startup.



3.3. Modelo de Negócios

Desta forma, a ideia é fornecer um produto ou serviço gratuitamente, para criar demanda de mercado para outro produto ou serviço. Logo, esse modelo associa uma versão gratuita e paga do produto ou serviço, onde a versão paga proporciona ao consumidor uma série de vantagens em relação à versão gratuita. Nesse modelo de negócio ainda que a fração de usuário que usam a versão paga seja minúscula, quando comparado com a versão gratuita, essa pequena fração é capaz de manter o produto ou serviço no mercado e ainda gerar lucros.

3.4. Oceano Azul

A estratégia do oceano azul visa fornecer uma forma para avaliar o mercado; levantar os atributos que regem a concorrência; bem como definir critérios de consumo, satisfação e atendimento aos clientes, com o objetivo de propor uma solução que se diferencie das demais.

A inovação resulta da busca por soluções diferenciadas e elegantes que visem resolver um problema real ou atender uma demanda latente, que gerem valor para os clientes e/ou que alcem a organização a uma posição privilegiada no mercado, onde elegância é encontrar a solução certa para um problema com simplicidade, criatividade, inteligência, sutileza, economia e qualidade [Kim, 2005].

A estratégia do oceano azul trata de dois cenários distintos. O primeiro é conhecido como "oceano vermelho", este possui vários consumidores sendo levados a consumir produtos e serviços de vários fornecedores que oferecem pouco ou nenhum diferencial entre seus produtos e serviços. Neste cenário podemos imaginar uma "sangrenta" competição por pequenos espaços de mercado entres os concorrentes em busca da sobrevivência [Kim, 2005].

O segundo cenário é o "oceano azul" cuja disputa por espaço não mais existe, pois neste caso os fornecedores oferecem algo diferenciado o bastante para tornar a concorrência irrelevante. O oceano azul é um novo espaço de mercado que ainda não havia sido descoberto, explorado pela concorrência [Kim, 2005].

A sobrevivência de uma organização está relacionada a como ela reinventa o setor e se diferencia dos concorrentes. A diferenciação hoje ganha lugar como estratégia mais poderosa no mundo dos negócios e principalmente da inovação [Neumeier, 2010].

O principal mote da estratégia do oceano azul é evitar as batalhas dentre os concorrentes e oferecer aos consumidores algo realmente novo e exclusivo, que até então não tinha sido explorado, produzindo assim a chamada inovação de valor, que alinha inovação com utilidade imediata e preço competitivo. A execução desta abordagem é auxiliada pelas ferramentas conhecidas como matriz de avaliação de valor e modelo das quatro ações.

3.4.1. Matriz de Avaliação de Valor

A matriz de avaliação de valor é usada para mapear os atributos que orientam a competição atual no segmento de mercado, bem como o nível de oferecimento desses atributos. A análise dessas informações possibilita a reorientação do foco estratégico e a



identificação de novos atributos de valor que vão além das fronteiras que atuam no momento.

A Figura 4 apresenta um modelo de referência para a matriz de avaliação de valor. O eixo horizontal define os atributos usados como fator de competição, estes fatores são usados na comparação entre os concorrentes encontrados no mercado e a abordagem proposta. O eixo vertical da matriz indica o nível de qualificação do atributo observado, em uma escala de 0 (baixo) a n (mais alto).

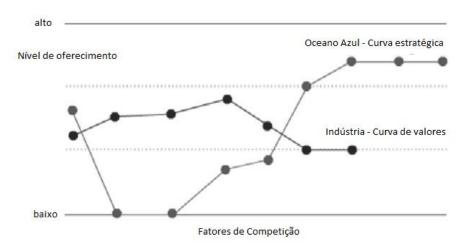


Figura 10. Matriz de Avaliação de Valor [Kim e Mauborgne, 2011].

3.4.2. Modelo das Quatro Ações

O Modelo das Quatro Ações visa definir uma nova curva de valor para a solução proposta, através das respostas as seguintes questões:

- A. Quais atributos considerados dispensáveis pelos consumidores devem ser eliminados?
- B. Quais atributos devem ser reduzidos abaixo dos padrões de consumo do mercado?
- C. Quais atributos devem ser elevados acima dos padrões do mercado?
- D. Quais atributos nunca oferecidos pelo setor devem ser ofertados?

Ao seguir tais etapas, podemos criar um produto ou serviço inovador que se distancie das ofertas dos concorrentes, passando a velejar em um oceano azul.

As técnicas tradicionais que apenas perguntam às pessoas o que elas querem, indicam melhorias incrementais, mas nunca levam às ideias revolucionárias capazes de mudar paradigmas e/ou reorientar um mercado [Ries, 2012].

Segundo Kim (2005) o resultado da aplicação do Modelo das Quatro Ações e a matriz de avaliação de valor do setor é a descoberta de novos aspectos na leitura de velhas verdades não questionadas, ou seja, aplicar esse Modelo das Quatro Ações à lógica vigente e examinar as alternativas e os não clientes, com isso é possível ter a possibilidade de chegar a um oceano azul.



3.5. Modelo de Negócio Grátis

Uma pesquisa recente em âmbito mundial apontada no trabalho de Mckinsey e Company (2010), revelou que 84% dos executivos entrevistados declararam que a inovação é extremamente ou muito importante para a estratégia de crescimento de suas organizações. Para garantir essa sobrevivência, as empresas recorrem a estratégias de inovação, como a que será apresentada nessa subseção.

Cada organização possui um modelo de negócio e para sobreviver deve criar e entregar um valor ao consumidor final, além disso, deve gerar renda suficiente para cobrir seus custos. A diferença é uma questão de foco: alguns empreendimentos visam o lucro ou maximizar ganhos, enquanto outras organizações têm missões não financeiras, focadas na ecologia, em causas sociais e concessionárias de serviço público que simplesmente oferecem um serviço essencial para uma população.

A economia do grátis vem ganhando força pelas tecnologias digitais, como diz a lei de Moore que o preço da unidade de armazenamento computacional vem caindo pela metade a cada dois anos, além do preço da banda larga que está caindo rapidamente [Botura, 2012].

O barateamento dos dispositivos de armazenamento, processamento e banda larga, vem permitindo a redução dos preços de fabricação de um produto ou serviço, sendo possível distribuir produtos e serviços com valor quase zero ou muitas vezes gratuitos como é caso do Google [Botura, 2012].

O real valor de um produto não está em seu preço, mas sim o bem que pode proporcionar para um determinado indivíduo. Este valor é algo extremamente complexo e intrigante. Portanto, informações e serviços que são abundantes podem e devem ser disponibilizados gratuitamente, mas é possível lucrar com informações e serviços exclusivos, isto é, com a escassez [Anderson, 2009].

Desta forma, a ideia é fornecer um produto ou serviço gratuitamente, para poder criar uma demanda de mercado para outro produto ou serviço, condicionado ao primeiro. Logo, esse modelo de negócio associa uma versão gratuita e paga do produto ou serviço, conhecida também como versão freemium e premium. Esta versão proporciona ao consumidor uma série de vantagens em relação a versão gratuita. Nesse modelo de negócio ainda que a fração de usuário premium seja minúscula quando comparado com os usuários gratuitos, essa pequena fração é capaz de manter o produto ou serviço no mercado e ainda gerar lucros (Aanderson, 2009).

Um dos princípios importantes apontado por Botura (2012) diz que o modelo grátis pode não ser interessante em determinadas situações, ou seja, às vezes pode não ser valioso dar algo de graça, quando antes era vendido, isso pode indicar que algo de graça pode não ter valor ou um valor menor que o produto pago.

Alguns dos negócios mais conhecidos do mundo, que oferecem seus produtos gratuitamente na internet vão desde o Skype, Google, Facebook ao Dropbox e atraem milhares de usuários por todo mundo através de serviços gratuitos de altíssima qualidade.



3.6. Modelo da Cauda Longa

O conceito de Cauda Longa está diretamente relacionado à economia da abundância, na qual o mercado apresenta diversos concorrentes que buscam oferecer produtos de massa (hits) aos consumidores. Já o mercado de oportunidades pouco exploradas ou inexistentes, este é mais conhecido na literatura como nicho. A cauda longa do mercado apresenta inúmeras variedades de nichos que muitas vezes ainda não foram exploradas pelo mercado atual. Além disso, existe abundância de consumidores na cauda longa, estes, por sua vez, nem mesmo sabem o que realmente precisam até que o produto ou serviço esteja disponível no mercado, exemplo disso é o produto *Tablet* e o serviço do *Netflix*.

4. Conclusão

Neste trabalho apresentamos o conceito de startups, detalhando tecnologias e técnicas que são utilizadas por novos e experientes empreendedores que se aventuram no mundo do empreendedorismo digital.

Referências

Anderson, C. Free-gratis: o futuro dos preços. Rio de Janeiro: Elsevier, 2009.

Botura, Ana C. O futuro dos preços. 2012. Disponível em:

http://www.fatecie.com/revista cientifica/2012/06.pdf>. Acesso em: 07/07/2013.

Grando, Nei. 2012 "Lean Startup – Para Iniciar uma Empresa Enxuta e Ágil", Último acesso em Out. de 2013, Disponível em:

http://neigrando.wordpress.com/2012/05/23/lean-startup-para-iniciar-uma-empresaenxuta-e-agil/>.

Kim, C.; Mauborger, R. (2005) The Blue Ocean Strategy. Harvard Business School. Mckinsey e Company. Mckinsey global survey results: innovation and commercialization. Disponível em:

http://www.mckinsey.com/insights/innovation/innovation_and_commercialization_2010 mckinsey global survey results. 2010>.

Meira, Silvio. (2013) "Novos Negócios Inovadores de Empreendimento Inovador no Brasil", Editora Casa das Palavras, Recife, Pernambuco.

Ries, Eric. (2012) "A Startup Enxuta", São Paulo, SP, Editora Lua de Papel.

ISSN: 2317-5346



SHARPE Software Package: Modelos Markovianos para Simulação de Sistemas

Kádna Camboim¹, Anderson Nascimento², Rafael Roque², Jean Araujo¹

¹ Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil

²Centro de Informática − Universidade Federal de Pernambuco (UFPE) Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, 50.740-560, Recife − PE − Brasil

Resumo. A necessidade por sistemas complexos e seguros é fator de relevância para as empresas e sociedade, que cada vez mais dependem da tecnologia em larga escala para promover o crescimento da capacidade computacional e garantir avanços relacionados ao desempenho e confiabilidade das novas aplicações. Dessa forma, é importante garantir que determinadas medidas de dependabilidade sejam delimitadas para evitar, por exemplo, a ocorrência de eventos de falhas. Assim, podemos usar modelos Markovianos para representar o comportamento de sistemas e facilitar a avaliação de dependabilidade.

Abstract. The need for complex and reliable systems is relevance factor for business and society, which increasingly rely on technology on a large scale to promote the growth of computing power and ensure progress related to the performance and reliability of new applications factor. Thus, it is important to ensure that certain measures of dependability are delimited to avoid, for example, the occurrence of fault events. So we can use Markov models to represent the behavior of systems and facilitate the evaluation of dependability.

1. Introdução

Os sistemas de informação aparecem em diversas áreas que fazem parte do nosso dia-adia, podemos citar educação, saúde, transporte, entretenimento, entre outros. Com os avanços constantes na tecnologia da informação surge a necessidade de suporte em determinados ambientes, além das preocupações relacionadas aos fatores como desempenho, disponibilidade, confiabilidade, etc.

A avaliação de dependabilidade denota a capacidade que um sistema tem de oferecer um serviço de forma confiável. As medidas de dependabilidade são confiabilidade, disponibilidade, manutenabilidade, performabilidade, segurança, testabilidade, confidencialidade e integridade [Laprie et al 1992]. A representação de sistemas por meio de técnicas de modelagem, permite a obtenção de informações úteis



sobre a estrutura e o comportamento dinâmico do sistema, fornecendo assim, suporte para análise de dependabilidade. As técnicas baseadas em modelagem podem ser classificadas como técnicas analíticas e técnicas baseadas em simulação [Lilja 2000].

Algumas ferramentas acadêmicas e comerciais permitem a modelagem de sistemas e a descrição de seu comportamento, seja através de Diagramas de Bloco de Confiabilidade (*Reliability Block Diagram* - RBD), Grafo de Alcançabilidade (*Reliability Graph* - RG), Cadeias de Markov (*Markov Chain* - MC), Redes de Petri Estocásticas (*Stochastic Petri Net* - SPN) ou outras. Dentre as várias ferramentas existentes para modelagem de sistemas, destacam-se: SHARPE Software Package [Sanher at al. 1996], Block-Sim [Relia Soft 2010], TimeNet [Zimmerman 2012] e Asto Tool [Silva at al. 2010]. Os modelos criados para realização deste trabalho compreendem as Cadeias de Markov para simular a ocorrência de determinados eventos utilizando o SHARPE Software Package [Sanher at al. 1996].

O presente artigo está organizado como segue: a seção 2 apresenta os conceitos sobre Cadeias de Markov, a seção 3 descreve o SHARPE Software Package, a seção 4 apresenta o estudo de caso utilizando modelos markovianos. Finalmente, a seção 5 apresenta as conclusões deste trabalho.

2. Cadeias de Markov

Um processo estocástico X(t), $t \in T$ é um conjunto de variáveis aleatórias definidas sobre o mesmo espaço de probabilidades, indexadas pelo parâmetro de tempo $(t \in T)$ e assumindo valores no espaço de estados $(si \in S)$.

A cadeia de Markov constitui um tipo particular de processo estocástico com estados discretos e com o parâmetro de tempo podendo assumir valores contínuos ou discretos. As Cadeias de Markov de tempo contínuo são chamadas CTMC (continuous-time Markov chains) e as cadeias de Markov de tempo discreto são chamadas DTMC (discrete-time Markov chains). A propriedade markoviana (ausência de memória) destaca que eventos futuros não estão condicionados a eventos passados, ou seja, são dependentes apenas do estado presente [Jain 1991].

A Cadeia de Markov possibilita a descrição do funcionamento de um sistema utilizando um conjunto de estados e transições entre esses estados. As transições entre os estados são modeladas por um processo estocástico de tempo contínuo ou discreto definidos por distribuições exponenciais ou geométricas.

Uma Cadeia de Markov do tipo Irredutível significa que todos os estados são comunicantes, ou seja, qualquer estado pode ser alcançado a partir de outro estado, assim, todos pertencem a uma mesma classe.

3. SHARPE Software Package

SHARPE é uma ferramenta de modelagem que serve para descrever comportamentos de um sistema em relação ao tempo, apresentando estruturalmente sua representação, a sigla é proveniente de *Simbolic Hierarchial Automated Reliabilit and Performance Evalutor*. Possui sua própria sintaxe para operar por linha de comando, através de dois modos de operação: 1) *Batch mode*: que pode ler entradas de um ou mais arquivos; 2) *Interactive mode*: que pode ler entradas de um terminal. A linguagem usada no modo



interativo é um subconjunto da linguagem *batch mode*, onde algumas palavras chave não são usadas porque o contexto de uma sessão interativa as faz desnecessárias.

A ferramenta oferece também especificação para sua própria linguagem de programação e ainda pode ser usada por meio de sua interface gráfica. Para resolução do problema do estudo de caso será usada sua interface gráfica. A estrutura de um sistema pode ser especificada, por exemplo, na forma de uma árvore, gráfico de alcançabilidade, Cadeia de Markov, etc. A seguir são listados os oito tipos diferentes de modelos que podem ser criados em casos específicos: Fault tree; Reliability block diagram; Reliability graph; Markov chain; Product-form queuing network; Multi-chain Product-form queuing network; Series-Parallel graph; Generalized stochastic Petri net (GSPN). Ao iniciar um projeto, um destes modelos deve ser escolhido.

A barra de Menu do SHARPE é composta pelas opções mostradas na Tabela 1, onde são mostradas suas funcionalidades. Para realizar a análise dos modelos são oferecidas diferentes opções para cada tipo de modelo. A Figura 1 apresenta as opções para cadeias de Markov. Por exemplo, você pode realizar a análise transiente, análise do estado estacionário, análise do tempo médio para falha do sistema e do tempo médio para restauração do sistema, entre outras.

Tabela 1 - Opções de Menu do SHARPE e suas funcionalidades

Menu		Funcionalidades			
•	File	Contém funções para manipulação dos projetos. criar, abrir, salvar, imprimir, etc.			
•	Model Editor	É possível salvar a tela de modelagem em formato jpg, editar as configurações dos modelos pertencentes ao projeto atual, definir constantes, variáveis e funções, bem como opções para adicionar ao projeto um dos tipos de modelagem suportadas pelo SHARPE, por exemplo <i>GSPN</i> , <i>Reliability Graph</i> .			
•	Analysis Editor	Análise do modelo selecionado.			
•	Plot	Plotar o resultado de um modelo graficamente.			
•	Browse Examples	Contém exemplos de modelos e suas respectivas análises. Também uma opção para dar entrada em arquivos para processamento.			
•	Help	Para obter informações sobre como utilizar a ferramenta SHARPE.			



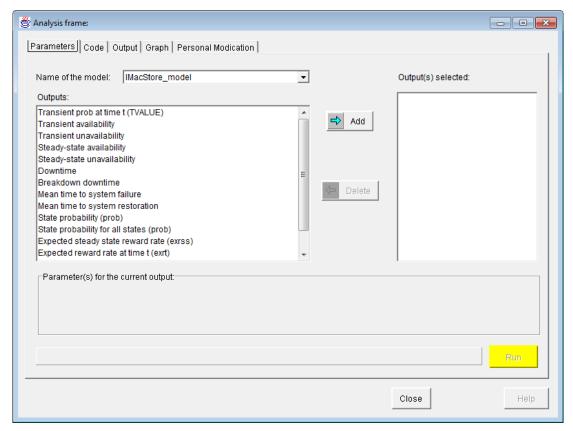


Figura 1. Opções de análises para modelos markovianos

A seguir será apresentado o estudo de caso e a utilização do SHARPE para modelagem com cadeias de Markov.

4. Estudo de Caso

Para realização do estudo de caso o seguinte problema foi considerado: Uma loja armazena computadores que podem ser comprados semanalmente. D1, D2,..., representa a demanda para os computadores (o número de unidades que deveriam ser vendidos se o estoque não é esgotado) durante a semana 1, semana 2, ..., respectivamente. Dado que X0 representa o número de computadores inicialmente, X1 o número de computadores no final da semana 2 e assim por diante. Assume-se que X0 = 3. No sábado à noite a loja faz o pedido de computadores para o fornecedor, o qual realizará a entrega apenas na próxima segunda-feira. A loja utiliza a seguinte política de compra: se não há computadores no estoque, a loja compra 3 computadores. Entretanto, se há algum computador no estoque, nenhum computador é comprado. Vendas são perdidas quando a demanda excede o estoque. Assim, {Xt} para t = 0, 1, 2,... é um Processo Estocástico. Os Estados possíveis do processo são os inteiros 0, 1, 2, 3, representando o número de computadores no final da semana t. As variáveis randômicas Xt são dependentes e podem ser avaliadas pela expressão:

$$X(t+1) = \begin{cases} \max\{3 - D(t+1), \ 0\} & if \ X(t) = 0 \\ \max\{X(t) - D(t+1), \ 0\} & if \ X(t) > 0 \end{cases}$$



A partir da expressão dada foi definida a matriz de transição P, seguindo a distribuição de Poisson. Então, a ferramenta SHARPE foi utilizada para modelar a Cadeia de Markov referente ao problema apresentado; calcular a matriz de probabilidade de n-passo; calcular a matriz de probabilidade de estado estacionário e analisar propriedades da Cadeia de Markov.

Inicialmente foram criados os estados utilizando o botão *Node*. No caso do problema proposto 'Estoque da loja de computadores', o modelo apresentará quatro estados que representam o número de computadores no final da semana t, iniciando com a marcação 0 até 3. Observe a Figura 2.



Figura 2. Estados do problema Estoque da loja de computadores

Após a criação dos estados foi acionado o botão *Rate Matrix* para adicionar as probabilidades de se chegar em um determinado estado a partir de um estado inicial. Por exemplo, pode-se perguntar: Qual é a probabilidade de chegar no estado 2, dado que o estado atual é 3? Resposta: 0.368. Assim, os arcos de saída de um estado com sua respectiva probabilidade de se chegar ao seu destino foram criados. Como são quatro estados, a matriz é uma 4x4. Os estados apresentados na vertical esquerda são considerados como Estado Inicial, e os estados na horizontal superior são os estados de destino, conforme ilustrado na Figura 3.

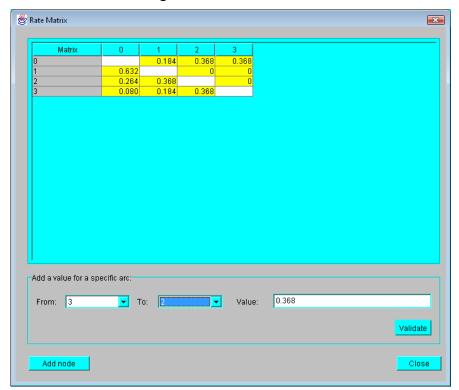


Figura 3. Matriz de Probabilidades de Estados

A matriz deve ser preenchida utilizando as opções *From*, *To* e *Value*. *From* - indica qual é o estado inicial. *To* - indica qual é o estado destino. *Value* - deve ser



informado o valor da probabilidade. *Validate* - valida o valor especificado em *Value* dado o estado inicial e o estado destino e adicionando o valor na matriz. Quando este botão é acionado, o arco que interliga os estados é criado no modelo.

A Figura 3, apresentada anteriormente, contém a matriz devidamente preenchida. Os espaços em branco representam a saída de um estado para ele mesmo, como no caso de 0x0, 1x1, 2x2 e 3x3. Nestes casos a ferramenta SHARPE não aceita inserção de valores. Os espaços preenchidos com valor zero (0) indicam que não há possibilidade de sair do estado de origem para o estado de destino. Representando o exemplo do estoque da loja de computadores, a matriz apresentada indica que 1) Se a loja tem 1 computador no estoque (estado inicial) é impossível vender 2 computadores (estado final). 2) Se a loja tem 1 computador no estoque (estado inicial) é impossível vender 3 computadores (estado final). 3) Se a loja tem 2 computadores no estoque (estado inicial) é impossível vender 3 computadores (estado final). As situações citadas representam uma demanda maior que o número de computadores no estoque da loja, ou seja, a loja está deixando de vender.

Ao final do preenchimento da matriz e sua validação, temos uma Cadeia de Markov representando o problema proposto. A Figura 4 apresenta o modelo obtido com as probabilidades de transição de um estado para outro.

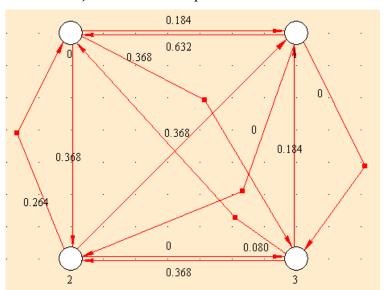


Figura 4. Cadeia de Morkov com diagrama de transição de estados

O cálculo da matriz de probabilidade em N passos é importante, pois esta análise se faz necessária para conhecermos as probabilidades da matriz após N unidades de tempo, isto é, N passos que representam as semanas que se passaram. Após a modelagem será possível determinar a matriz de probabilidade do estoque e como ela ficará após 3 semanas. Pois na modelagem, foi criada a matriz de transição de probabilidades de estado para um passo no tempo.

O SHARPE software em sua opção de análise para Cadeias de Markov, tem uma função chamada TVALUE que pode ser aplicada às cadeias de Markov irredutívies (que é o caso), pois fornece resultados transitórios para um único valor de t. Para construir a matriz de probabilidade em 3 unidades de tempo (3 semanas), será necessário definir o Vetor de Probabilidade de Estado em Função do Tempo, para cada um dos estados. Isso



porque, antes de realizar o cálculo, o vetor tem que estar definido previamente, para cada estado que estiver sendo calculado.

Colocando 1 no valor do vetor, significa que estamos definindo exatamente aquele estado como estado inicial e que queremos ir para o Estado de destino. Na tabela 2 são apresentados os valores definidos no Vetor de Probabilidade de Estado em Função do Tempo. Os resultados obtidos após as análises podem ser vistos na Tabela 3, que mostra a Matriz de Probabilidade no tempo 3.

 Estado
 Vetor

 0
 1 0 0 0

 1
 0 1 0 0

 2
 0 0 1 0

Tabela 2 - Vetor de Probabilidade de Estado (π)

Tabela 3 - Matriz de Probabilidade para 3 semanas

0001

3

Estado Inicial Estado Destino	0	1	2	3
0	0,285	0,257	0,273	0,184
1	0,317	0,323	0,207	0,153
2	0,285	0,291	0,306	0,118
3	0,235	0,257	0,273	0,234

5. Conclusões

Este artigo apresentou um modelo Markoviano gerado com a ferramenta SHARPE Software para resolução do problema de estoque de uma loja de computadores que, não deve acumular computadores no estoque e nem deve perder venda por falta de estoque. Foram definidas a matriz de probabilidade dos estados, bem como, o vertor de probabilidade e a matiz de probabilidades para 3 semanas.

References

Jain, R. (1991) "The Art of Computer Systems Performance Analysis", John Wiley & Sons, New York.

Laprie, J.C.C., Avizienis, A. and Kopetz, H. (1992) "Dependability: Basic Concepts and Terminology." Springer-Verlag New York, Inc. Secaucus, NJ, USA.

Lilja, D. J. (2000) "Measuring Computer Performance: A Practitioner's Guide," Cambridge University Press © 2000.



- Sahner, R., Trivedi, K. and Puliafito, A. (1996) "Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package." Kluwer Academic Publishers, Norwell, MA, USA.
- Silva, B., Maciel, P., Tavares, E., Araujo, C., Callou, G., Sousa, E., Rosa, N., Marwah, M., Sharma, R., Shah, A., Christian, T. and Pires, J.P. (2010) "Astro: A tool for dependability evaluation of data center infrastructures." In Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on, pages 783–790, oct. 2010.

Zimmermann, A. (2012) "Modeling and evaluation of stochastic petri nets with timenet 4.1." Ilmenau, Germany. http://valuetools.org/2012/show/program-final.

ISSN: 2317-5346



Robótica Educacional: Uma Possibilidade para o Ensino e Aprendizagem

Luma Cardoso Ferro de Almeida¹, Jhonatan Sérgio Diniz Marques da Silva¹, Haroldo José Costa do Amaral¹

¹Universidade de Pernambuco (UPE), Campus Garanhuns CEP 55.294-902 – Garanhuns – PE – Brasil

{lumaalmeidal8, jhonatan.sergio haroldo.amaral}@gmail.com

Resumo. A Robótica Educacional constitui um recurso didático-pedagógico, onde abordagens de ensino inovadoras poderão promover a aprendizagem de conteúdos diversos, de maneira interdisciplinar. Ao contrário do que se pensa a sua aplicação é simples, não sendo necessários muitos conhecimentos técnicos, além de ter um baixo custo atrelado. Este trabalho, através de uma oficina, pretende desmistificar a robótica e suas tecnologias, como recurso educativo, compreendendo conceitos teóricos e práticos, acessíveis inclusive para aqueles que nunca tiveram acesso.

Abstract. The Educational Robotics is a pedagogical didactic resource, where innovative teaching approaches may promote learning of diverse content, in an interdisciplinary way. Contrary to what you think your application is simple, not much technical knowledge is required, in addition to having a low cost. This paper through a workshop aims to demystify robotics and its technologies, as an educational resource, comprising theoretical and practical concepts, accessible even to those who never had access.

1. Introdução

É comum, quando ouvimos falar em robótica, imaginarmos imagens semelhantes a dos filmes de ficção cientifica. Porém, é na indústria, onde sua presença é mais forte, fazendo com que os robôs industriais pareçam apenas versões antigas. Entretanto, atualmente, a robótica está cada vez mais presente em outros contextos, a saber: na realização de cirurgias; em tarefas caseiras; em atividades de entretenimento; em atividades perigosas, como trabalhos com materiais químicos, explorações espaciais, entre outras.

Na educação, sua atuação acontece de forma a tornar o aprendizado mais significativo, promovendo, através de seu uso pedagógico, diferentes tipos de conhecimentos e competências. Nesse sentido, a robótica aplicada à educação – denominada de Robótica Educacional, Robótica Educativa ou Robótica Pedagógica [d'ABREU et. al., 2012] – tem o poder de formar cidadãos com competências e habilidades necessárias para conviver e prosperar em um mundo cada vez mais contemporâneo e global, contribuindo assim, com o desenvolvimento social e econômico do nosso País.



Diferentemente do que se imagina em relação à complexidade da construção de artefatos robóticos, na Robótica Educacional, através de kits de montagem, trabalhos com sucata e softwares especializados, os alunos terão a oportunidade de adquirir meios de solucionar problemas diversos das mais variadas áreas do conhecimento, aumentando, dessa forma, o seu poder cognitivo. Além disso, o professor poderá demonstrar, na prática, diversos conceitos teóricos, incluindo os de difícil compreensão, motivando todos os envolvidos nas atividades e favorecendo a processos de ensino e de aprendizagem.

O restante deste artigo está organizado como segue: a Seção 2 um breve relato histórico da Robótica e dos seus benefícios; a Seção 3 discute a Robótica Educacional, como recurso didático-pedagógico, em processos de ensino e de aprendizagem; na Seção 4, evidenciamos a robótica livre, como meio de viabilizar a Robótica Educacional a baixo custo, e apresentamos uma proposta de oficina para aplicação da Robótica Educacional; por fim, a Seção 5 traz algumas considerações acerca deste trabalho.

2. Robótica: Breve Histórico

A palavra robô, do Checo *robota*, significa trabalho forçado [FORESTI, 2006]. O Minidicionário Contemporâneo da Língua Portuguesa Caldas Aulete [AULETE, 2009] define robô como "máquina que, mediante instruções nela introduzidas, é capaz de executar ações e movimentos semelhantes aos humanos e, em certos casos, de identificar estímulos e responder a eles" e define robótica como a "ciência que estuda a construção e o emprego de robôs".

É remota a intenção dos homens de projetar mecanismos que tentem reproduzir características naturais. A primeira marca é dos Gregos (222 a.C) que já projetavam peças, podendo ser consideradas os primeiros robôs, mas estes tinham um caráter meramente lúdico ou estético [AZEVEDO, 2010]. Nota-se a presença também dos Árabes, que desenvolveram inúmeros mecanismos baseados nos estudos Gregos, os quais já se preocupavam com a utilidade das criações [PIRES, 2002].

No século XVI, Leonardo DaVinci dedicou-se ao estudo da robótica, tendo acesso aos projetos Gregos através de documentos escritos pelos Árabes [AZEVEDO, 2010]. Alguns pesquisadores acreditam que ele projetou e talvez até criou um robô espetacular de aspecto humano, mas não se tem nenhuma prova da existência desse projeto, pois, do seu livro Codex Atlanticus, faltam páginas precisamente na altura em que parecia preparar-se para projetar um robô [PIRES, 2002]. Apesar de seu feito, DaVinci ainda não possuía tecnologia suficiente para o seu robô, pois precisava de componentes construídos com grande precisão e forte fonte de energia. Outra contribuição forte para a robótica veio de Nicola Tesla, que criou um barco submersível, controlado a distância usando impulsos hertzianos codificados [PIRES, 2002].

Apesar de sua história ter início há 222 anos a. C, o termo robô foi usado pela primeira vez em 1921, por Karel Capek, em sua peça de teatro "Rossum's Universal Robots" [ROCHA, 2006 apud SALANT 1990]. Já Issac Asimov foi a primeira pessoa a utilizar o termo Robótica em 1941, no seu filme de ficção científica [FORESTI, 2006].

Podem-se encontrar os primeiros trabalhos em robótica de manipulação após a segunda guerra mundial. As máquinas eram, na sua maioria, para manipular materiais perigosos e elétricos, como o Master-Slave (1940-50), que manipulava materiais



radioativos e o Planetbot (1957), primeiro manipulador elétrico com juntas de revolução [PIRES, 2002].

O Tortoise foi construído em 1950 por W. Grey Walter e é considerado um dos primeiros robôs móveis [FORESTI, 2006]. Já o Unimate, desenvolvido por George Devol e Joseph Engelberger, é o primeiro robô industrial (1959-62) [PIRES, 2002; FORESTI, 2006]. Só em 1974 é que surge o primeiro robô controlado por computador (retroação sensorial), o T3, que foi comercializado a partir de 1978 [PIRES, 2002].

Após a popularização dos sistemas computacionais, na década de 70, é que começam a surgir sistemas de processamento central. E os robôs passam a ser criados não apenas para manipulação nas fábricas, mas também para conviver com os humanos [FORESTI, 2006]. Com a inteligência artificial, foi desenvolvido em 2006 o Cog, um robô que interage com seres humanos e aprende como uma criança [FORESTI, 2006].

Desde então, a robótica está cada vez mais presente no dia a dia das pessoas: ao entrar no elevador e pressionar o botão, passamos a utilizar a robótica; ao andarmos na escada rolante, estamos usufruindo das tecnologias robotizadas; ao utilizarmos o caixa eletrônico, estamos usando um equipamento da robótica.

3. Robótica Educacional

A Robótica Educacional pode ser definida como "um conjunto de recursos que visa o aprendizado científico e tecnológico integrado às demais áreas do conhecimento, utilizando-se de atividades como design, construção e programação de robôs" [LOPES, 2008, p. 41].

O grande precursor da robótica na escola foi o cientista Seymourt Papert, pois logo após sua saída do Centro de Epistemologia Genética de Genebra, integrou-se ao Laboratório de Inteligência Artificial do Instituto de Tecnologia de Massachusetts (MIT), em 1964, e lá dá inicio aos seus trabalhos acerca da robótica na educação. Ele via no computador um recurso que atraia as crianças e com isso facilitava o processo de aprendizagem [GOMES *et. al.*, 2010].

Nas atividades de robótica, surge a possibilidade de controlar artefatos reais e não apenas os virtuais, num ambiente onde o estudante pode manusear, criar, programar e, através desta prática lúdica, desenvolver o raciocínio lógico e algorítmico tão importante nas diversas áreas do conhecimento [DO ROCIO ZILLI, 2004].

Um dos trabalhos mais célebres de Papert foi a criação da tartaruga controlada em LOGO, nome dado a uma linguagem de programação de fácil assimilação, a qual serve para a comunicação do homem com o computador [GREGOLIN, 2009]. A linguagem LOGO abriu um espaço de criação com capacidade de simular formas, imagens e comandos bem acessíveis a qualquer idade, abrangendo desde as ciências até as artes. Esta forma simples de programação atraiu a atenção de muitos [CASTILHO, 2006].

No Brasil, várias linguagens de programação estão sendo fabricadas, baseandose na linguagem LOGO, e sendo incorporadas à robótica, como forma de auxiliar no controle dos robôs [SILVA et. al, 2009]. Como exemplo, podemos citar a versão traduzida do original pelo Núcleo de Informática Educacional (NIED), a SuperLogo [GREGOLIN, 2009].

ISSN: 2317-5346



A robótica na escola possibilita a autonomia do aluno, permite que ele saia da carteira para atuar em atividades práticas e virar um observador/inventor. Ele passa a aprender através de seus próprios erros e acertos, sendo autônomo na construção do seu conhecimento, investigando, explorando, planejando e dando forma a suas ideias. Como destaca Silva (2009), "com o uso da robótica, muitos conteúdos podem ser analisados pelos jovens e professores de melhor forma, uma vez que essa tecnologia pode ser usada de diversas maneiras e em diferentes níveis de aprendizagem" [SILVA, 2009, p. 117].

Observa-se assim a importância desta atividade interdisciplinar na escola, pois envolve conhecimentos diversos, como: componentes eletrônicos, cinemática, mecânica básica, informática, física, inteligência artificial, ambientes de programação, entre outros [GOMES et. al., 2010].

4. A Robótica Livre

Diversos são os materiais utilizados para a construção e programação de robôs, porém muitos desses materiais (kits comerciais com softwares e hardwares proprietários) custam caro, inviabilizando muitas vezes a sua compra e aquisição pelas escolas [CELINSKI et. al., 2012]. Dessa forma, procura-se por materiais de baixo curso e observa-se a possibilidade de utilizar o chamado lixo eletrônico (nome dado a produtos eletroeletrônicos que são descartados) – sua reutilização também é chamada de metareciclagem e deste "lixo" é possível aproveitar a parte mecânica e eletrônica.

Para a programação de robôs de baixo custo, utiliza-se de hardware e software livres, além de que qualquer pessoa poder fazer as modificações necessárias e adaptar à sua realidade [PINTO, 2012]. Um exemplo de microcontrolador, entre os existentes, é a placa de Arduino, criada na Itália em 2005.

4.1. A Tecnologia Arduino

Arduino é um projeto que tem o "objetivo de oferecer uma plataforma de prototipagem eletrônica (placa eletrônica programável) de baixo custo e de fácil manuseio por qualquer pessoa interessada em criar projetos com objetos e ambientes interativos" [ARDUINO apud CELINSKI et. al., 2012, p. 3]

A placa de Arduino possui IDE (*Integrated Development Environment*) própria, que consiste em uma linguagem de programação padrão do microcontrolador, desenvolvida com referência as linguagens C e C++, e com uma interface construída em Java [MCROBERTS, 2011]. A IDE da placa Arduino é simples de utilizar e entender, traz muitos exemplos de códigos prontos e que podem ser modificados, permitindo desenvolver o software e enviá-lo para a placa. A IDE é disponibilizada para download em http://arduino.cc/en/main/software, onde é fornecido também o seu código fonte. Dessa forma, são muitos os projetos realizados com a plataforma Arduino, por se tratar de open source, diversas também são as versões da placa que são desenvolvidas.

Ainda, existem diversos ambientes de colaboração para pesquisadores em robótica, que discutem e promovem a aplicação da Robótica Educacional, de forma interdisciplinar. Em Pernambuco, podemos destacar a Plataforma Robô Livre, disponível em http://robolivre.org, que, com o seu slogan "É fácil fazer", evidencia a robótica como uma área de experimentação e pesquisa aberta a qualquer pessoa,



independente de conhecimento prévio sobre o tema, formação, idade ou condição sociocultural.

4.2. Uma Proposta de Oficina com a Robótica Livre

No intuito de promover a Robótica Livre, como recurso didático-pedagógico, a contribuição deste trabalho consiste em uma oficina, fazendo uso de uma metodologia de caráter inovador, pois é fundamentada em abordagens não tradicionais. Com base em abordagens de ensino e aprendizagem, que já vêm sendo testadas por multiplicadores da Plataforma Robô Livre, ela está apoiada nos seguintes pilares:

- Abordagem Horizontal: são apresentadas as possibilidades, mudando apenas o aprofundamento de acordo com o artefato que os participantes optarem por desenvolver;
- Planificação da Relação Facilitador-Aluno: o palestrante estará presente não na figura de um "professor", que apenas dita como as atividades deverão ser realizadas, mas sim como facilitador, que procura desmistificar e questionar junto com o participante, bem como apontar os caminhos a serem seguidos para a finalização do artefato robótico.

Durante o desenvolvimento, dois momentos distintos ocorrerão: (i) primeiramente, uma parte teórica, com questionamentos para a reflexão dos participantes, objetivando especificar o que é a robótica, destacando as considerações históricas mais importantes (surgimento, áreas do conhecimento envolvidas, entre outras), e discutindo a robótica educacional e suas possibilidades; (ii) no segundo momento, a prática, consistirá na construção e programação de robôs, onde serão usados softwares e hardwares livres, bem como materiais de lixo eletrônico. Neste momento, os participantes se reunirão em grupos para o desenvolvimento de um artefato escolhido. Durante a prática, o grupo se deparará com questões como: Como será nosso robô? Quais atividades ele irá desempenhar? Será possível construir diante do tempo e dos materiais que temos? Quais serão as peças necessárias para o seu desenvolvimento? Entre outras diversas questões que deverão ser analisadas pelo grupo.

Dessa forma, a aprendizagem acontecerá por descobertas que o próprio participante fará, partindo da integração e colaboração com o grupo, pois, assim como afirma Valente [VALENTE, 1999], para que exista uma aprendizagem onde a informação é processada pelos esquemas mentais, é necessário que os participantes sejam desafiados e, consequentemente, elaborem estratégias para a resolução de problemas.

5. Considerações Finais

A Robótica Educacional desenvolve competências e habilidades cada vez mais necessárias, principalmente por desmistificar a tecnologia em um mundo cada vez mais global e tecnológico.

Assim, é possível perceber possibilidades para novas abordagens de ensino, onde os estudantes se sentirão mais envolvidos e desafiados a resolver problemas e, como consequência, buscarão soluções, construindo o seu próprio conhecimento. Observa-se ainda outras possibilidades para escolas, que não possuem estrutura para



adquirir kits proprietários, poderem utilizar a robótica livre a fim de promover o conhecimento sobre diversas áreas, diante da sua característica interdisciplinar.

Referências

AULETE, Caldas. Minidicionário contemporâneo da língua portuguesa/Caldas Aulete.

2 ed. Rio de Janeiro: Lexikon, 2009.

CASTILHO, Maria Inês. Robótica na educação: com que objetivos?. 2002. Disponível em

http://www.pgie.ufrgs.br/alunos espie/espie/mariac/public html/robot edu.html>.

Acesso em: 31 agos. 2010.

- CELINSKI, T. M.; CERUTTI, D. M. L.; CELINSKI, V. G.; CERUTTI, I. A.; IELO, F. G. P. F. (2012) Robótica Educativa: uma proposta para o reuso do lixo eletrônico em uma atividade de extensão universitária. In IV Congresso Internacional de Educação, Pesquisa e Gestão. Disponível em: http://www.resol.com.br/textos/01340544057.pdf>. Acesso em: 14 set. 2013.
- DO ROCIO ZILLI, S.(2004). A robótica educacional no ensino fundamental: Perspectivas e prática. Disponível em: < https://repositorio.ufsc.br/bitstream/handle/123456789/86930/224814.pdf?sequence= 1>.

Acesso em: 05 agos. 2013.

FORESTI, H. B. (2006) Desenvolvimento de um robô bípede autônomo. Disponível em: http://www.robolivre.com/download/dissertacao_banca2B.pdf. Acesso em: 15 jul.

2010.

- GOMES, C. G.; SILVA, F. O. da; BOTELHO, J. C.; SOUZA, A. R. de. A robótica como facilitadora do processo ensino-aprendizagem de matemática no ensino fundamental. 2010. Disponível em: < http://books.scielo.org/id/bpkng/pdf/pirola9788579830815-11.pdf. Acesso em: 10 jul. 2013.
- LOPES, D. de Q. A exploração de modelos e os níveis de abstração nas construções criativas com robótica educacional. Porto Alegre (2008). Disponível em: http://www.lume.ufrgs.br/handle/10183/16173. Acesso em: 04 ago. de 2013.
- MCROBERTS, M. Arduino Básico. Tradução de Rafael Zanolli. São Paulo: Novatec Editora, 2011.
- PINTO, M. C.; ELIA, M. F.; SAMPAIO, F. F. (2012) Formação de professores em robótica educacional com Hardware Livre Arduino no contexto Um Computador por Aluno. In XVII Workshop de Informática na Educação. Disponível em: < http://www.br-ie.org/pub/index.php/wie/article/view/2100>. Acesso em: 15 set. 2013.



- PIRES, J. N. (2002) Robótica: Das Máquinas Gregas à Moderna Robótica Industrial. Disponível em: http://robotics.dem.uc.pt/norberto/nova/pdfs/gregosxxi.pdf>. Acesso em: 01 abr. 2013.
- SILVA, A. F. (2009) RoboEduc: Uma metodologia de Aprendizado com a Robótica Educacional. Disponível em: < http://bdtd.bczm.ufrn.br/tde_arquivos/19/TDE-2009-0609T062813Z-2013/Publico/AlziraFS.pdf Acesso em: 14 out. 2013.
- D'ABREU, J. V. V.; RAMOS, J. J. G.; MIRISOLA, L. G. B.; BERNARDI, N. (2012)
- Robótica Educativa/Pedagógica na Era Digital. In II Congresso Internacional TIC e Educação. Disponível em: < http://ticeduca.ie.ul.pt/atas/pdf/158.pdf>. Acesso em: 15 out. 2013.
- AZEVEDO, S.; AGLAÉ, A.; PITTA, R. (2010) Minicurso: Introdução a Robótica Educacional. In 62^a Reunião Anual da SBPC. Disponível em: < http://www.sbpcnet.org.br/livro/62ra/minicursos/MC%20Samuel%20Azevedo.pdf >.

Acesso em: 24 out. 2013.

ROCHA, R. (2006) Utilização da Robótica Pedagógica no Ensino-Aprendizagem de Programação de Computadores. Disponível em: < http://www.files.scire.net.br/atrio/cefet-mg-ppget_upl/THESIS/130/rogerio_rocha.pdf>.

Acesso em: 10 ago. 2013.

- GREGOLIN, V. R. (2009) Linguagem LOGO: Explorando Conceitos Matemáticos. In Revista Tecnologias na Educação. Disponível em: < http://tecnologiasnaeducacao.pro.br/revista/a1n1/art8.pdf. Acesso em 27 out. 2013.
- SILVA, A. A. R. S. da; COELHO, M. G. P.; BARROS, R. P.; GONÇALVES, L. M. G. A
- (2009) Robótica Pedagógica no Contexto da Educação Infantil: Auxiliando o Alfabetismo. In Revista Tecnologias na Educação. Disponível em: < http://tecnologiasnaeducacao.pro.br/revista/a1n1/art13.pdf>. Acesso em: 27 out. 2013.



Desenvolvendo Aplicações com Java Card

Wilton Oliveira Ferreira^{1,2}, Ricardo Batista Rodrigues², Julio Cezar Damasceno³, Josino Rodrigues Neto²

¹Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R) Rua Bione, 220 - Cais do Apolo - Bairro do Recife 50.030-390 - Recife, PE - Brazil

²Centro de Informática, Universidade Federal de Pernambuco (UFPE) Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária - 50.740-560 - Recife, PE - Brasil

³Departamento de Estatística e Informática - Universidade Federal Rural de Pernambuco (UFRPE)

Rua Dom Manoel de Medeiros, s/n, Dois Irmãos 52.171-900 - Recife, PE - Brasil

{wof,rbr,jcd,jrn4}@cin.ufpe.br

Resumo. Este trabalho tem como objetivo apresentar o conceito de Java Card e os principais métodos, técnicas e ferramentas utilizadas na atualidade por desenvolvedores. A ideia é apresentar a tecnologia e detalhar o processo de como iniciar uma aplicação, apresentar técnicas e ferramentas que podem maximizar as chances da criação de aplicações bem sucedidas e as principais vantagens da tecnologia. Por fim, mostrar o passo a passo do desenvolvimento de uma pequena aplicação, para que seja colocado em prática as técnicas apresentadas.

Abstract. This paper aims to introduce the concept of Java Card and the main methods, techniques and tools used by developers today. The idea is to present the technology and detail the process of how to start an application, present techniques and tools that can maximize the chances of creating successful applications and the main advantages of the technology. Finally, to show the step by step development of a small application, to be put into practice the techniques presented.

1. Introdução

O foco do mercado nos últimos anos tem sido a emissão de Smart Cards para diversas aplicações. Porém, essa tendência vem se alterando para um maior esforço de melhor utilização dos cartões já existentes no mercado, visto que é possível fazer atualizações no Smart Card sem que seja necessário o retorno ao fabricante ou a emissão de mais cartões. Dessa forma, os setores público, financeiro e de telecomunicações, que são os mais presentes nesse mercado, serão os que mais investirão em novas formas de exploração do Smart Card [Queiroz 2013].

Em novembro de 1996, um grupo de engenheiros da Schlumberger introduziram uma API Java Card para desenvolvimento em seus Smart Cards, propondo também um rascunho inicial para a especificação Java Card. Meses depois Bull, Gemplus e



Schlumberger se unem para fundar o Java Card, formam um consórcio industrial criado para identificar e resolver questões relacionadas a tecnologia Java Card. Finalmente a Sun Microsystems dá suporte oficialmente a especificação Java Card 1.0 que não passava de uma API e não uma Plataforma extensivel. Em novembro de 1997, a Sun Microsystems anuncia a especificação Java Card 2.0 com diferenças significantes como uma maneira de se escrever applets utilizando a Orientação a Objetos e especifica mais o Ambiente em Tempo de Execução [Massao 2013].

Em Março de 1999 é anunciada a especificação Java Card 2.1, pode ser considerada uma Plataforma completa pois é especificado o Java Card API, JCRE (Java Card Runtime Environment) e o JCVM (Java Card Virtual Machine). Em maio de 2000 é anunciada a especificação Java Card 2.1.1. Em setembro de 2002 é anunciada a especificação Java Card 2.2 o qual introduz a característica RMI. Em outubro de 2003 é anunciada a especificação Java Card 2.2.1. Em março de 2006 é anunciada a especificação Java Card 2.2.2, essa é a ultima versão 2 e introduziu interfaces para desenvolvimento de Biometria. Em maio de 2009 é anunciada a especificação Java Card 3.0.1 não mais pela finada Sun Microsystem mais pela então Oracle [Massao 2013].

Smart Cards representam uma das menores plataformas computacionais do mundo, isso quer dizer, são limitados em recursos de processamento e memória. No início, desenvolver para Smart Cards era um processo lento e difícil, pois embora os cartões possuíssem padrões no tamanho e no protocolo de comunicação, cada um funcionava de maneira diferente de fabricante para fabricante [Massao 2013].

As ferramentas de desenvolvimento eram fornecidas pelos fabricantes dos Chips utilizando como base a linguagem Assembler, os SDKs (Software Developement Kit) eram dedicados a apenas um tipo ou família de chips, ou seja, o desenvolvimento de uma aplicação era específico para cada fabricante e determinado Chip [Oracle 2013].

2. Smart Card

O Smart Card é um cartão de plástico com um chip acoplado que armazena e processa dados. Essa tecnologia é usada em diversas áreas, dentre os quais podem-se citar aplicações na área de saúde, bancária, de entretenimento e de transportes. Tais aplicações se utilizam da segurança que os Smart Cards providenciam, sendo assim um fator chave para sua utilização. Para tal, utiliza-se a tecnologia Java Card, que permite que pequenas aplicações, chamadas applets, sejam empregadas em Smart Cards, utilizando a plataforma Java [Massao 2013].

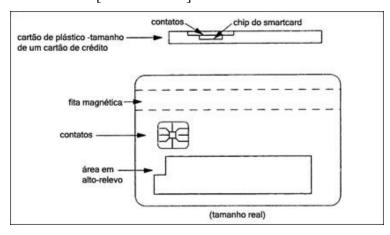




Figura 1. Esquema simples de um Smart Card [Queiroz 2013]

Um benefício para sua ampla utilização é seu poder computacional embutido, além de segurança, portabilidade e facilidade de uso que esta tecnologia oferece. Hoje, nos Estados Unidos, Smart Cards são utilizados em grande parte do tempo por seus consumidores para variadas atividades, como visitar bibliotecas, assistir filmes no cinema ou fazer compras, provando-se altamente presente na vida cotidiana atual [Massao 2013].

2.1. Conceitos Fundamentais de Java Card

Java Card é uma Interface de Programação de Aplicativos API desenvolvida e distribuída pela Oracle que permiti que aplicativos Java, chamados applets, seja executado nos chips de Smart Cards, acessando seus sistemas de arquivos e microprocessador [Oracle 2013].

A Tecnologia Java Card oferece uma maneira prática de vencer esses obstáculos, ocultando as implementações de Hardware do cartão. Ou seja, desde que o cartão implemente a especificações Java Card, não importa o tipo de processador, ou se é arquitetura Cisc ou Risc, ou a capacidade de memória. Essencialmente a Tecnologia Java Card define a segurança, portabilidade e multiaplicativos que incorporam as várias vantagens da Linguagem Java, como por exemplo, Write once run anywhere (escreva uma vez e execute em qualquer lugar) [Massao 2013].

A segurança é uma preocupação primordial, uma das características, por exemplo, é o nível de acesso a todos os métodos e variáveis que são estritamente controlados, e não existe modo de forjar ponteiros para ativar programas maliciosos na memória. Java Cards permite que sejam instalados vários Applets e cada Applet dentro do cartão é isolado por um firewall, isso evita que uma aplicação hostil comprometa o sistema [Queiroz 2013]. A seguir apresentamos as principais características da API Java Card:

- Ambiente com baixo poder computacional: executar applets Java em chips de baixo poder computacional como o de Smart Cards é um desafio. Sendo assim, a tecnologia Java Card implementa a máquina virtual Java de forma dividida entre o sistema do cartão e o sistema operacional do terminal. No cartão, são implementados o modelo de execução, comunicação, segurança e o gerenciamento da memória. Processos como carregamento de classes, verificação de bytecode, e etc. são executados no sistema externo ao cartão.
- **Programação de alto nível**: outra característica implementada pela API é a separação entre o sistema e as aplicações. Assim como a linguagem Java para desktops, a API permite o gerenciamento dos dados através de requerimentos bem definidos e de alto nível.
- **Portabilidade**: como aplicativos em Java para desktops, os applets Java Card são independentes de plataforma. Dessa forma, applets iguais podem ser usados em Smart Cards com capacidade de memória e processamento diferente, além de estruturas e arquiteturas diversas. Isso não se limita a cartões, mas também a ambientes em que há pequeno poder de processamento.
- Multifuncionalidade: cartões com a tecnologia Java Card podem conter mais de um applet. A API Java Card gerencia o compartilhamento de dados entre



applets diferentes, protegendo dados internos aos applets e sincronizando o acesso a dados compartilhados.

3. Arquitetura Java Card

A Tecnologia Java Card oferece um ambiente seguro para as aplicações que rodam em smart card e outros dispositivos com memória e capacidade de processamento limitada.

Os smart card permitem a implantação de vários aplicativos em um único dispositivo, geralmente os aplicativos são implementados em tempo de fabricação. Porém, novos aplicativos podem ser adicionados a ele mesmo depois de estar sendo utilizado pelo usuário final. Java Card também inclui um conjunto de ferramentas únicas para o desenvolvimento de novos produtos. Por causa das limitações de hardware a plataforma Java Card é dívida em duas partes, sendo uma, presente no smart card e a outra no computador convencional, utilizado para o desenvolvimento dos aplicativos Java Card [Rankl and Effing 2002]. A plataforma completa de desenvolvimento do Java Card, consiste das partes a seguir devidamente documentada em especificações.

- A especificação JCVM (Java Card Virtual Machine), define um subconjunto da linguagem de programação Java e definição também a JCVM adequada para o desenvolvimento de aplicação smart card [Oracle 2013].
- A especificação JCRE (Java Carde Runtime Environment), descreve precisamente como deve comportar-se as aplicações Java Card em tempo de execução. Nesta documentação também incluem gerenciamento de memória, gerenciamento dos Applets e também de outros recursos em tempo de execução smart card [Oracle 2013].

3.1. API Java Card (Subconjunto da Linguagem Java)

A tecnologia Java Card suporta apenas um subconjunto das funcionalidades da linguagem Java. Do mesmo modo, a JCVM (Java Card Virtual Machine) suporta apenas os recursos que são necessários para o subconjunto da linguagem Java Card. Quaisquer características da linguagem Java não suportados pelas JCVMs e utilizadas em um Applet são detectados pelo conversor no momento da conversão.

A plataforma Java Card apresenta um subconjunto seleto das funcionalidades da linguagem Java convencional. Esta característica se dá em razão das limitações dos dispositivos que utiliza esta plataforma. As funcionalidades selecionadas para compor a API Java Card têm como principal característica, o tamanho reduzido de ocupação de memória. A Tabela 1 destaca as principais funcionalidades suportadas, e as notáveis funcionalidades não suportadas.

Tabela 1: Funcionalidades suportadas e não suportadas do Java Card

Funcionalidades Suportadas	Funcionalidades não Suportadas
•Pequenos tipos de dados primitivos: boolean, byte, short	•Grandes tipos de dados primitivos: long, double, float
•Array unidimensional	•Caracteres e Strings
•Pacotes java, classes, interfaces e	•Carregamento de Classes dinâmicas



tratamento de exceções

- •Funcionalidades da orientação a objeto: herança, métodos virtuais, sobrecarga e criação dinâmica do objeto, escopo de acesso, e regras de vinculo
- •Tipo inteiro de 32 bits é suportado apenas em algumas versões da API.
- •Gerenciamento de Segurança
- •Coletor de lixo e finalização de objetos
- Threads
- •Serialização de objeto
- •Serialização de objeto
- •Clonagem de objeto.

3.2. JCVM (Java Card Virtual Machine)

A JCVM (Java Card Virtual Machine) apresenta os mesmos conceitos da JVM (Java Virtual Machine), com apenas algumas diferenças. A primeira diferença entre estas duas plataformas é caracterizada pelo fato da JCVM ser dividida em duas partes separadas, como mostra a Figura 2 [Chen 2000]. A parte on-card da JCVM incluem o interpretador de bytecode Java Card. A parte off-card incluem um conjunto de ferramentas para analisar o código java, converter e encapsular as classes em um pacote especifico denominado CAP (Converter Applet). O pacote CAP pode ser implantado no smart card ou qualquer outro dispositivo que suporte Java Card, e as classes presentes neste pacote são interpretadas pelo interpretador on-card.

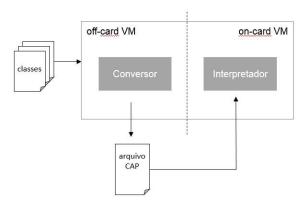


Figura 2. Java Card Virtual Machine

3.2.1. Arquivo CAP e Arquivo Export

A tecnologia Java Card introduz dois novos formatos de arquivos binários, que permitem o desenvolvimento distribuído, independente da plataforma de execução. O arquivo CAP (Converter Applet) contém a representação binária executável das classes em um pacote Java. O arquivo CAP é um arquivo JAR (Java Arquive) que contém um conjunto de componentes também encapsulado de forma individual em arquivos JAR [Chen 2000].



Cada componente descreve um aspecto do conteúdo do arquivo CAP, tais como informações das classes, bytecodes executáveis, links de informações, verificação de informações e assim por diante. O formato de arquivo CAP é otimizado para pequenos dispositivos, usando estruturas de dados compactos. Além do arquivo CAP também é gerado o arquivo Export, que auxilia a criação do próprio CAP.

Os arquivos Exports é utilizado na fase de pré-compilação do projeto Java Card. Esta pré-compilação utiliza recursos que não fazem parte do smart card, este conjunto de recurso é denominado artefatos off card. Os artefatos off card são auxiliares, e sua utilização só ocorre na fase de formação do CAP. O Export é composto pelas várias bibliotecas necessárias para a compilação inicial, e fase de montagem do arquivo CAP. Este arquivo poderá ser descartado imediatamente após realizada a conversão (compilação e empacotamento) do projeto Java Card em pacote CAP.

Ao contrário da máquina virtual Java, que processa uma classe de cada vez, a unidade de conversão processa o pacote por completo. Os arquivos das classes são produzidos por um compilador Java a partir do código fonte. Então, o conversor préprocessa todos os arquivos de classe que compõem um pacote Java e converte o pacote em um arquivo CAP. Na Figura 3 é mostrado a estrutura completa do processo de conversão.

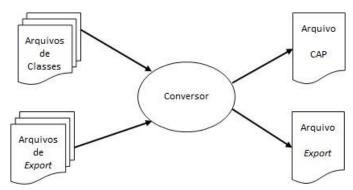


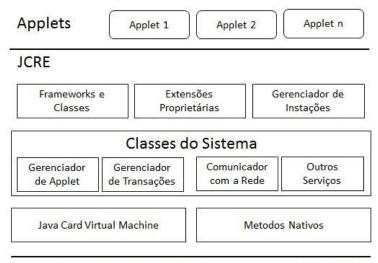
Figura 3. Conversor de pacotes

3.3. Java Card Runtime Environment (JCRE)

O JCRE (Java Card Runtime Environment) consiste em um componente Java Card que executa dentro do smart card. O JCRE é responsável pela gestão dos recursos do smart card, tais como comunicações com a rede, a execução dos Applet e dos sistemas de segurança do/para o Applet. Este componente é essencial para a solução smart card. A Figura 4 apresenta a posição onde se localiza o JCRE dentro da arquitetura da solução smart card. Como pode ser observado, o JCRE oferece os vários serviços de Frameworks, implementação de especificações proprietária e ferramentas para gerir a implantação dos Applets no smart card.

ISSN: 2317-5346





Hardware e sistemas nativo do smart card

Figura 4. Arquitetura Smart Card

Como pode ser observado na Figura 4, a camada superior à camada de hardware é composta pela JCVM e por métodos nativos a qual atua como um mediador entre o hardware e as camadas de software superiores a ela. Logo, imediatamente superior, temos as Classes do Sistema que mais atua em prol do apoio à infraestrutura como um todo, dentre os vários serviços desta camada, se destacam: gerenciador de applet, gerenciador de transações e gerenciador das comunicações de redes. Acima da camada de Classes do Sistema está localizada a JCRE (Java Card Runtime Environment) que por sua vez conta com alguns pacotes especifico que são utilizados para auxiliar a execução dos Applets que está na camada imediatamente superior.

4. Desenvolvendo uma Aplicação com Java Card

O desenvolvimento de uma aplicação Java Card precede os mesmos cuidados inerentes a programação na linguagem Java convencional. A seguir serão apresentados algumas das necessidades e exigências desta tecnologia.

4.1. Configuração do Ambiente de Desenvolvimento

O ambiente de desenvolvimento destinado à tecnologia Java Card pode ser montado utilizando ferramentas como eclipse e plug-ins específicos para tal [Eclipse 2013][Youssef 2013]. Um plug-in para o desenvolvimento Java Card bastante eficiente e o Eclipse JCDE (Eclipse Java Card Development Environment) [Youssef 2013]. Observando apenas a necessidade de além do JDK (Java Development Kit) e JRE (Java Runtime Environment) também é necessário o JCDK (Java Card Development Kit).

Durante o processo de configuração do ambiente, deve ser observado a compatibilidade do smart card com a versão do JCDK, a ISO/IEC (International Organization for Standardization / International Electrotechnical Commission) cuida da regulamentação dos smart cards em todos seus aspectos. Que vai desde a estrutura física do dispositivo até os padrões de comunicação incluindo o suporte ao Java Card. Já o responsável por manter a compatibilidade do Java Card com os padrões ISSO/IEC fica a par da Oracle [Oracle 2013] atual mantenedora da tecnologia.



4.2. Emulação de Smart Card

O desenvolvimento dos aplicativos Java Card é facilitado pela funcionalidade padrões da ferramenta Eclipse [Eclipse 2013]. Desta forma, desenvolver os Applets para Java Card será algo trivial do ponto de vista de codificação. Do ponto de vista da semântica da linguagem, deve ser observado as limitações quanto aos tipos de dados suportados. Há frequentes casos que, por exemplo, a compilação é permitida porem a execução dos testes geram uma exceção. Detectar uma exceção em um Applet Java Card é extremamente complicado, isto, porque, quando o mesmo está sendo executado diretamente na mídia smart card, não é possível analisar o log muito menos visualizar a execução. Para estes casos só é possível contar com as várias mensagens contendo códigos de erros que podem ser facilmente interpretados com base na ISO 7816 [ISO/IEC 2013].

Durante o processo de desenvolvimento, os códigos gerados podem ser testados utilizando ferramentas que emulam o ambiente smart card. Desta forma, os resultados obtidos a partir da emulação do smart card aproxima, em muito, as características do hardware real. Além de manter a facilidade na ausência da necessidade da implantação do Applet em uma mídia smart card, a ferramenta de emulação também permite facilmente a depuração do código, o que reduz a perda de tempo com frequentes testes durante o desenvolvimento.

5. Conclusão

O estudo apresentado teve como desafio apresentar as características básicas da tecnologia Java Card. Bem como instruir aos interessados, como realizar pesquisas envolvendo esta tecnologia. Foram apresentadas as principais características da arquitetura e também algumas aplicações que admitem o uso de tal tecnologia. Resumidamente, pode ser dito que os padrões de segurança oferecido pela plataforma smart card/Java Card é extremamente robusta no que se refere a capacidade de proteção. Todas as funcionalidades de segurança do smart card podem ser estendidas através do uso de applets. Alguns casos de sucesso, bem conhecidos, são os cartões de créditos e tokens utilizado em operações financeiras. Além disso, esta tecnologia permite que um smart card possa ser atualizado após ser liberado pelo fabricante. Quando modificamos uma senha de um cartão bancário por exemplo, o fazemos alterando informações interna do smart card

A tecnologia Java Card ainda é recente, logo ainda existem muitos desafios de pesquisa a serem explorados. Ainda existe uma certa dificuldade enfrentada para os pesquisadores/desenvolvedores que estão se envolvendo com tal tecnologia no que se refere a conteúdo de apoio. Algumas referências como [Chen 2000], [Finkenzeller 2003], [Rankl and Effing 2002], [Haghiri and Tarantino 2002], [Ferrari et al. 1998], podem ajudar a quem de fato pretende se empenhar nos estudos do Java Card para smart cards.

Referências

Chen, Z. (2000). Java Card Technology for Smart CCard – Architecture and Programmer's Guide. II. Addison-Wesley, first edition edition.

Eclipse (2013). Eclipse.



Ferrari, J., Mackinnon, R., Poh, S., and Yatawara, L. (1998). Smart Cards: A Case Study. IBM, first edition.

Finkenzeller, K. (2003). RFID HANDBOOK - Fundamentos and Applications in Contactless Smart Cards and Identification. WILEY, second edition.

Haghiri, Y. and Tarantino, T. (2002). Smart Card Manufactoring - A Pratical Guide. WILEY.

ISO/IEC (2013). Iso/iec 7816 - cards and personal identification.

Massao, R. (2013). Planet smart cards.

Oracle (2013). Oracle – java card technology. Technical report, Oracle.

Queiroz, A. R. (2013). Smar card e java card.

Rankl, W. and Effing, W. (2002). Smart Card Handbook. Wiley, third edition edition.

Youssef, A. (2013). Eclipsejcde user guide.

ISSN: 2317-5346



Utilizando simulação para planejar a implantação de Redes Ad Hoc

Rafael Souza¹, Ricardo Rodrigues¹, Ahyalla Pereira¹, Kádna Camboim¹, Jean Texeira¹

¹Centro de Informática, Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brasil
Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária – 50.740-560 – Recife – PE-Brasil

{rrs4,rbr,carlp,kmac,jcta}@cin.ufpe.br

Resumo. Este trabalho descreve a importância que os dispositivos móveis ganharam na atualidade no campo da Tecnologia da Informação. Com isso, surgiram novos tipos de tecnologia sem fio, para romper a necessidade de conexão por meio de cabos, permitindo que o usuário ganhe mais flexibilidade, entre outras vantagens. Utilizando o conceito de simulação para reproduzir um ambiente de redes Ad hoc e avaliar o desempenho de diferentes tipos de ambiente. Permitindo, assim, um melhor planejamento do ambiente de rede ad hoc.

Abstract. This paper describes the importance that mobile devices have gained today in the field of Information Technology. With this came new types of wireless technology, the need to break the connection through cables, allowing the user to gain more flexibility, among other benefits. Using the concept of simulation to reproduce an environment of Ad hoc networks and evaluate the performance of different types of environment. Thus allowing better planning of the ad hoc network environment.

1. Introdução

Nos últimos anos ocorreu uma vasta proliferação de tecnologias de redes wireless WLAN (Wireless Local Area Network) tais como Bluetooth, WIMAX (Worldwide Interoperability for Microwave Access), 3G (Third Generation), 4G (Fourth Generation). Todos esses equipamentos têm oferecido uma solução atraente aos usuários que pretendem obter uma instalação rápida, simples e sem os problemas associados ao cabeamento.

As redes ad hoc, são redes sem fio que não precisam de infraestrutura para se comunicar, permitindo que dispositivos portáteis possam se comunicar e interagir entre si de forma espontânea, prática, portátil e flexível (Perkins, 2008). Referem-se basicamente a uma rede de computadores em que não são necessários cabos, tendo como exemplos notebook, PDA (*Personal Digital Assistant*) e qualquer outro item do gênero, que utilizam a comunicação de redes sem fio, por meio de rádio frequência ou infravermelho. Fatores primordiais como topologia dinâmica, flexibilidade e mobilidade estão alimentando o crescimento do mercado de computação móvel.



Diante das inúmeras mudanças e adequações relacionadas às tecnologias de dispositivos portáteis, preocupar-se com as necessidades de soluções tecnológicas alternativas, vem se destacando como problema a ser solucionado buscando obter vantagens na utilização dessas alternativas. Estas soluções visam atender as regiões que apresentam ainda uma infraestrutura de comunicação precária. Impulsionando, assim, o surgimento de um novo paradigma computacional em que o sistema de redes de computadores é autônomo. Este tipo de sistema é capaz de operar independentemente de infraestrutura e é auto-organizável. Por sua vez, os dispositivos móveis passam a se referenciado como nós, onde pode se conectar espontaneamente com os demais dispositivos na rede, atuando como plataformas móveis portáteis, conhecidos por redes ad hoc (LAGO, 2008).

Um dos métodos de avaliar software, em redes ad hoc, é por simulação, apesar de existir outras formas, como método analítico ou algébrico, implicado em um denso conhecimento matemático. Dependendo do modelo matemático a resolução torna-se extenuante, o que obriga a simplificações do modelo, podendo resultar em imperfeições na representação do sistema.

Por outro lado, o método por simulação permite a confecção de modelos complexos e resolução destes com menor desenvolvimento matemático. Para isso, emprega-se o poder computacional para as iterações numéricas requeridas, sugerindo, a depender dos resultados desejados, grande consumo computacional.

1.2. Motivação

As MANET (*Mobile Ad hoc NETwork*) têm despertado grande interesse nas universidades, nas empresas privadas e nas comunidades militares, onde são motivados tanto por aspectos da inviabilidade da instalação de redes com fio em certos lugares, como pelo barateamento dos equipamentos sem fio. Outros fatores relacionam-se com as facilidades de mobilidade, flexibilidade e a capacidade de serem auto configuráveis que a MANET oferece (PERKINS, 2008).

Um dos motivos que levaram o pesquisador a optar pelos softwares de simulação NS-2 (Network Simulator) e JiST/SWANTS (*Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator*), é por representarem um diferencial a respeito das simulações que foram executadas, pois a avaliação de simuladores tem como objetivo descobrir qual é o melhor, de acordo com os experimentos propostos. Os simuladores utilizados neste trabalho foram NS-2 (FALL, VARADHAN, 2002). De acordo com Kurkowski et al. (2005) o NS-2 é o mais utilizado em pesquisa sobre redes *ad hoc* sem fio, em que 43% dos pesquisadores utilizaram o NS2 como ferramenta de simulação. A Figura 1, ilustra a pesquisa sobre a utilização de simuladores, mostrando que o NS-2 é o mais utilizado na investigação MANET.



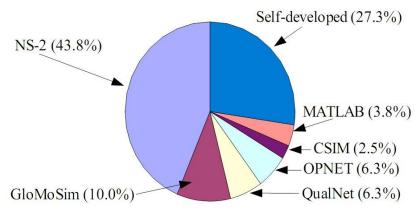


Figura 1: Pesquisa sobre a utilização de simuladores. Fonte: (KURKOWSKI et al. 2005).

Por outro lado a escolha do JiST/SWANS, teve como base de escolha os requisitos: i) software recente; ii) poucos trabalhos científicos relacionados; iii) desenvolvido em Java; e, iv) *open source*. De acordo com Barr (2006), há um grande desempenho em escalabilidade e poder computacional. Juntamente como os requisitos: i) uso da linguagem Java, amplamente conhecida e utilizada; ii) bom desempenho e um baixo consumo de memória nas simulações; iii) e a existência de toda a estrutura necessária para realizar a simulação de uma rede *ad hoc*.

Para a execução de uma simulação é necessário a escolha de um modelo de mobilidade que determinará seu padrão de movimentação e um protocolo de comunicação, sobre o qual eles vão se comunicar, o que o estudo em questão, teve como escolhido o AODV (*Ad hoc On-demand Distance Vector*) (JAYAKUMAR, 2008). O modelo de mobilidade escolhido para este trabalho foi o RWP (*Random WayPoint*) (YOON et al. 2003), que tem como base de escolha a grande utilização na comunidade científica até o momento (MORAES et al. 2009), sendo, sem dúvidas, o mais utilizado modelo de mobilidade na comunidade científica. Este modelo tornou-se, de fato, padrão de pesquisa em computação móvel. Por exemplo, nove em cada dez artigos publicados sobre redes *ad hoc*, no ACM MobiHoc 2002, usaram o Modelo *Random WayPoint*.

2. Redes Móveis Ad hoc

As redes *ad hoc* sem fio, referenciado pelo IEEE (*Institute of Electrical and Electronics Engineers*) como MANET são redes móveis sem fio (ou roteadores), que formam, dinamicamente, uma rede temporária sem a utilização de qualquer infraestrutura de rede existente ou de administração centralizada e que podem comunicar-se, entre si, sem a necessidade de uma estação de suporte ou um ponto de acesso centralizado (PERKINS, 2008) como ilustra a Figura 2.

ISSN: 2317-5346



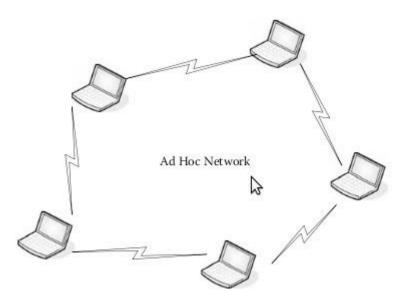


Figura 2: Mobilidade de Redes ad hoc. Fonte: (PERKINS, 2008).

De acordo com Chlamtac (2003), a rede *ad hoc* teve início na década de 70, quando uma entidade de pesquisa dos Estados Unidos começou a estudar como poderia ser a comunicação via rádio em um ambiente tático militar, ambiente em que há grande necessidade de mobilidade dos dispositivos que se comunicam através da rede e de facilidade na conexão de novos dispositivos.

Em 1972, o DoD (*Department Of Defense*) dos EUA, lançou um novo programa sobre PRNET (Packet Radio NETworks), com a intenção de criar tecnologias para o campo de batalha que não precisem de uma infraestrutura fixa já implantada, o programa fomenta a forma de comunicação dinâmica, mesmo quando alguns dos nós falham ou são destruídos. A primeira tecnologia de acesso ao meio utilizada, foi o protocolo alhora, desenvolvido na Universidade do Hawai, por Norman Abrhamson (ZAKON, 2005). Assim, a primeira rede *ad hoc* já estava usando a tecnologia LAN (Local Area Network) sem fio, como o MAC (Media Access Control) subjacente e camadas PHY (*PHysical Layer*).

Posteriormente, no início de 1980, o DARPA (Defense Advanced Research Projects Agency) lançou o programa SURAN (SURvivable Adaptive Network) para expandir a tecnologia desenvolvida no projeto PRNE, nas áreas de escalabilidade na rede, segurança, capacidade de processamento e gerenciamento de energia. O principal objetivo desta iniciativa foi desenvolver algoritmos de redes de apoio às redes que podem escalar para milhares de nós e suportam ataques de biossegurança e, com isso, ter suporte a grandes redes e desenvolver protocolos de rede adaptativos, nos quais pudessem ajustar às rápidas mudanças de condições num ambiente táctico (BASAGNI et al. 2008).

No início da década de 1990, o DARPA, para satisfazer os sistemas de informações da defesa, de forma que fossem robustos e rapidamente expansíveis, lançou o GloMo (*Global Mobile Information Systems*), que teve início em 1994. Enquanto as comunicações tácticas militares permaneciam sendo a principal aplicação das redes *ad hoc*, havia um número crescente de aplicações não militares, por exemplo: vídeo conferência e suporte a salvamentos. Também na década de 1990 a computação móvel

ISSN: 2317-5346



tornou-se acessível para população, nas formas de laptops, notebooks e PDA. Ao mesmo tempo, hardware e software, especialmente software open-source, tornaram-se amplamente disponíveis para a interligação de computadores, sendo trivial à ligação à rede global emergente, a Internet (BASAGNI et al. 2008).

3. Simulação

O termo simulação é bem antigo, surgiu por volta da década de 1940, tendo como pioneiro John Von Neumann, que tinha como ideia a execução de múltiplas repetições de um determinado modelo (TANENBAUM, 2003).

A simulação é uma técnica que busca imitar uma sequência de processos, a qual pode testar o comportamento e reações de determinados sistemas em diversas situações de modo seguro e prático, oferecendo um baixo custo pela sua utilização, bem como, evita riscos que poderiam ocorrer em testes reais.

A simulação é considerada uma das mais poderosas ferramentas de análise de desempenho. Normalmente, é utilizada na realização desse tipo de experimento, complementando as ferramentas de análise. Não se pode imaginar a realização de um grande projeto, por exemplo, de engenharia, sem uma prévia simulação de seu comportamento, avaliando as respostas do sistema para os mais diversos valores de entrada.

Um modelo tem como finalidade retratar todas as circunstâncias de um determinado cenário, que tenta chegar o mais próximo possível da realidade. Atualmente, existe um número limitado de modelos, cuja simulação entende como um algoritmo na estrutura de um grafo, permitindo executar, modelar e avaliar o desempenho de sistemas (TANENBAUM, 2003).

Uma das vantagens da simulação é o custo e o benefício proporcionado por essa técnica. Geralmente, uma simulação tem um custo relativamente pequeno se comparando a uma implementação, além de auxiliar na obtenção de resultados num período de tempo menor relacionado ao tempo que seria necessário ser empregado na realização do experimento em um cenário real.

4. Simuladores de Redes Ad hoc

São ferramentas de simulação orientada a eventos, tem se revelado muito útil no estudo de natureza dinâmica para redes de *ad hoc*, bem como funções da MANET e seus protocolos (algoritmos de roteamento, TCP, UDP, etc), fornecendo aos usuários uma maneira de especificar os protocolos de rede, e simular os comportamentos correspondentes.

Nas subseções seguintes serão mostrados detalhamentos sobre os simuladores utilizados na pesquisa.

4.1. **JiST**

O JiST é uma plataforma *open source*, desenvolvido em Java, para simulação de eventos discretos, sendo executado sobre uma máquina virtual Java. Desenvolvido por Rimon Barr, na Universidade de Cornell. Algumas características do seu projeto trouxeram os seguintes benefícios: i) Utilizar uma linguagem já conhecida para a



plataforma de simulação; ii) Não é necessário criar uma biblioteca de simulação; iii) Não desenvolver um novo kernel de sistemas para simulação.

A idéia principal, definida por Barr, é transformar uma máquina virtual Java em um escalonador de eventos, modificando a forma como as chamadas de métodos entre as entidades de simulação são conduzidas. Algumas das características de criação de sistemas com JiST, são: i) Eficiência (Sistemas de simulação altamente otimizados); ii) Transparência (Simulações são automaticamente transformadas para rodar com semânticas de tempo); iii) Padronização (Capacidade de escrever simulações com um sistema convencional de linguagem de programação).

Com base nas características e facilidades do simulador, foi escolhido o JiST/SWANTS. Hoje este projeto não é mais mantido pelo seu criador Rimon Barr. No entanto, um bom número de melhorias e aprimoramentos foi recentemente lançado pela Universität Ulm (BARR, 2006).

4.2. SWANS

Construído sobre a plataforma JiST, o SWANS é concebido como independente e distinto, mas são módulos de software que podem ser combinados para formar uma rede sem fio (BARR et al. 2006).

JiST/SWANS é um software de simulação desenvolvido em Java, o que torna possível instalá-lo e executá-lo em diversas plataformas como Linux, Windows, Solaris, Mac, mas não se limitando a estas.

O SWANS é organizado em componentes independentes de software que podem ser compostos completando as configurações de uma rede sem fio ou rede de sensores, mostrando-se uma arquitetura bastante modular. Este simulador apresenta capacidades semelhantes às do NS-2, além de dar suporte à simulação de redes maiores, ou seja, com uma maior quantidade de estações. A comunidade acadêmica vem investindo muito nesse simulador (BARR. 2006b).

4.3. NS-2

NS (versão 2), mais conhecido como NS2, é simplesmente uma ferramenta de simulação *open source*. Desenvolvido a partir do projeto VINT (Virtual InterNetwork Testbed) (KUMAR et al. 1997). Sua criação em 1989 tem vindo a ganhar enorme interesse da indústria, academia e governo. Sem dúvida, o NS-2 tornou-se um dos simuladores mais utilizados de rede de fonte aberta. É um software de simulação bastante difundido na academia, utilizado em muitas pesquisas e é muito estável (ISSARIYAKUL, HOSSAIN, 2009).

Este simulador é orientado a eventos e tem suporte para pilha de protocolos TCP/IP, que pode ser usado para redes LAN (Local Area Network), MAN (Metropolitan Area Network) e WAN (Wide Area Network). Outro fato atrativo da ferramenta NS-2 é que seu código-fonte é aberto e se encontra devidamente documentado.

Hoje em dia, o desenvolvimento do NS-2 é mantido pela DARPA e por colaboração de projetos como SAMAN (Simulation Augmented by Measurement and Analysis for Networks) (HEIDEMANN et al. 2001), CONSER (Collaborative Simulation for Education and Research) (CHEN, 2001), NSF (National Science Foundation, EUA). O



qual já recebe apoio de várias empresas como Lawrence Berkeley National Laboratory, do Xerox PARC

5. Conclusão

Este trabalho abordou simuladores de rede *ad hoc*, o poder computacional, escalabilidade de ambos e uma pesquisa sobre os protocolos de roteamento e modelo de mobilidade para serem utilizados na simulação. Ao final, para sedimentar esses conhecimentos, foram escolhidos os simuladores JiST/SWANS e NS2 utilizando o modelo de mobilidade *randon waippon* e o protocolo aodv. Ambos os simuladores foram escolhidos para este trabalho, a fim de apresentar suas características e vantagens no planejamento de redes ad hoc, por meio de simulação. Portando, pode-se concluir que simuladores são de grande relevância para o planejamento das infraestruturas de rede ad hoc.

Referências

BARR, Rimon. JiST- Java in Simulation Time User Guide, 2006 Cornell University. .Disponível em: <jist.ece.cornell.edu/docs.html >. Acesso em: 13 maio de 2009.

EUA

- BASAGNI, Stefano; CONTI, Marco; GIORDANO, Silvia; STOJMENOVIC, Ivan. MOBILE AD HOC NETWORKING. New York: editora IEEE Press Wiley, 2008.
- ______, SWANS- Scalable Wireless Ad hoc Network Simulator User Guide. EUA: Cornell University, 2006. Disponível em: <www.jist.com>. Acesso em: 13 mai de 2009.
- ______, HASS, Zygmunt, J; RENESSE, Robbert, Van, JiST: An efficient approach to simulation using virtual machines. Nova York, 2004. Disponível em:http://www.cs.uga.edu/~maria/pads/papers/jist-040827-jist-vmsim-spe. pdf >, Acessado em 2 fev 2009.
- CHEN, Xuan. CONSER (Collaborative Simulation for Education and Research), 2001. Disponível < http://www.isi.edu/conser/index.html>. Acessado em 15 mai de 2009.
- CHLAMTAC, Imrich; CONTI, Marco; LIU, Jennifer J-N. Mobile ad hoc networking: imperatives and challenges. Editora Elsevier. Texas. 2003. Disponível em:http://perso.ens-lyon.fr/isabelle.guerin-lassous/Enseignement/survey-adhoc.pdf>. Acesso em 05 jan de 2010. Pag. 1
- FALL, Kevin; VARADHAN, Kannan. The ns Manual, A collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2000. Disponível em: http://www.isi.edu/nsnam/ns/ns-documentation.html>. Acesso em 26 jun 2009.
- HEIDEMANN, John; Estrin, Deborah; Ramesh, Govindan; Goel, Ashish. SAMAN (Simulation Augmented by Measurement and Analysis for Networks). Disponível em: http://www.isi.edu/saman/index.html>. Acesso em: 10 de maio 2009.
- ISSARIYAKUL, Teerawat; HOSSAIN, Ekram. Introduction to Network Simulator NS2, editora Springer, 2009



- JAYAKUMAR, Geetha; GOPINATH, G. Performance comparison of two on-demand routing protocols for ad-hoc networks based on random waypoint mobility model. Índia 2008. Disponível em: <www.scipub.org/fulltext /ajas/ ajas56659-664.pdf>. Acesso em 10 set de 2008.
- KUMAR, Satish; HELMY, Ahmed. VINT (Virtual InterNetwork Testbed), 1997. Disponível em < http://www.isi.edu/nsnam/vint/index.html>. Acessado em 05 de mai de 2009.
- MORAES, Renato, M; ARAÚJO, Fagner. P; PONTES, Alisson. S, L; Uma proposta para estabilizar o modelo de mobilidade Random Waypoint em redes Ad hoc sem fio. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2009, Recife-PE. Disponível em: <>. Acessado em 15 de set de 2008.
- PERANCONI, Daniela, Saccol; MUHAMMAD, Hisham, H; BARCELLOS, Marinho
- P. Modelo de arquitetura para simulação de redes moveis sem o ad hoc no Simmcast.
- Rio Grande do Sul, 2003. P. 1 2. Disponível em: < wait4.org /~marinho/simmcast/pubs/errc200-daniela-hisham-marinho.pdf >. Acesso em: 06 ago de 2009.
- PERKINS, Charles, E. Ad hoc Networking, Boston, USA, Pearson, 2008 . p 1-28.
- KURKOWSKI, Stuart; CAMP, Tracy; COLAGROSSO, Michael. MANET Simulation Studies: The Incredibles; MCS Department, Colorado School of Mines, Golden, Colorado, USA. 19 ago 2005. Disponível em www.cl.cam.ac.uk/~sos22/p50kurkowski.pdf>. Acesso em: 16 fev 2009. Pag 1-2
- YOON, Jungkeun; LIU, Mingyan; NOBLE, Brian. Random Waypoint Considered Harmful; Proceedings of IEEE Conference on Computer Communications (INFOCOM), San Francisco, USA, 2003. Disponível em: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.1241. Acesso em: 26 de Nov de 2009.
- ZAKON, Robert H'obbes. Hobbes' Internet Timeline. 28 ago de 2005. Disponível em: http://www.zakon.org/robert/internet/timeline/>. Acesso em: 02 fev 2009.



Introdução a Modelagem 3D com Blender 3D

Thiago H. B. de Oliveira¹, Gabriel de França Pereira e Silva¹

¹Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista – 55.292-270 – Garanhuns – PE – Brasil {thiagophoc, gfps.ufrpe}@gmail.com

Resumo. Esse artigo tem por objetivo apresentar a ferramenta Blender 3D aplicada a modelagem de objetos em 3D. O Blender 3D é uma ferramenta open-source que possibilita ao usuário a manipulação de imagens, criação de objetos tridimensionais, edição de filmes e desenvolvimento de jogos pelo seu próprio motor gráfico o Blender Game Engine. Desta forma a ferramenta é utilizada para diversos usos como: animações, desenvolvimento de jogos e propagandas.

Abstract. This article aims to present the Blender 3D modeling tool applied to 3D objects. Blender 3D is an open-source tool that allows the user to manipulate images, creating three-dimensional objects, film editing and game development by its own graphics engine Blender Game Engine. This tool used for various purposes such: animation, game development and advertising.

1.Introdução

Na pesquisa acadêmica, muitas vezes desenvolvemos aplicações voltadas para a área de Computação Gráfica sem acompanhar como essas ferramentas serão utilizadas posteriormente pelos usuários finais. A proposta deste mini-curso é diminuir a distância entre a teoria e a utilização, demonstrando as técnicas aplicadas à computação gráfica e sua relação de dependência/autonomia em relação às ferramentas disponibilizadas pelos softwares. Para isso, será adotada a ferramenta Blender 3D versão 2.5. Portanto, este mini-curso irá abordadar técnicas de Modelagem 3D para iniciantes e revisar conceitos importantes inerente a área da Computação Gráfica.

A ferramenta Blender foi concebida em dezembro de 1993 e tornou-se um produto oficialmente em agosto de 1994, como uma aplicação integrada que permite a criação de uma diversificada gama de conteúdos 2D e 3D [3][5]. A ferramenta oferece uma série de recursos que permitem elaborar complexas modelagens e avançadas animações. Essas características tornam o Blender um dos aplicativos gráficos *Open Source* 3D mais populares do mundo.

Atualmente o Blender é mantido pela Blender Foundation, na Dinamarca, e já foi usado profissionalmente na confecção do curta metragem, Elephants Dream e dos Filmes Homem Aranha 2 e Plumíferos [2], enquanto segue sendo utilizado por agências de publicidade e comunicação ao redor do mundo. Aparentemente o Blender é um programa dificil manipulação, contudo é preciso levar em consideração que todos os softwares da área são, por natureza, complexos. Dessa forma, é correto afirmar que o



Blender não é mais difícil que os equivalentes mais utilizados do mercado, como o 3D Studio Max [3] ou o Maya [4].

2. Conceitos necessários antes de modelar

Antes de começar a trabalhar com qualquer pacote 3D, é crucial entender os conceitos matemáticos envolvidos na Computação Gráfica [6]. Esta seção irá apresentar os conceitos básicos que estão diretamente relacionados a produção de modelos tridimensionais (3D).

2.1 Coordenadas

Se você trabalhou com qualquer aplicação de construção 3D antes, você irá se familiarizar com o conceito de eixo Z. O eixo Z, em adição aos existentes, X para horizontal e Y para vertical, representa profundidade. Nas aplicações 3D, você verá informações sobre objetos estabelecidos em formato X, Y, Z – estes são conhecidos como método de Coordenadas Cartesianas, ver Figura 1.

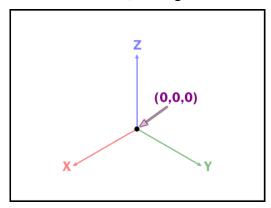


Figura 1. Modelo de Coordenadas do Blender 3D.

Agora, a nossa origem passa a ter três valores, o X, o Y e o Z. Mais do que isso, ela passa a ter três cores também:

X = vermelho

Y = verde

Z = azul

No Blender é importante apenas identificar os eixos e as cores. O mais interessante é que essas cores são padronizadas nos programas de modelagem mais populares do mercado.

2.2 Espaços

Em qualquer pacote 3D, o mundo que você irá trabalhar é tecnicamente infinito, e pode ser dificil controlar a localização dos objetos dentro dele. Em todo mundo 3D, há um ponto de origem, frequentemente referenciado como 0, como é representado pela posição (0, 0, 0). Todas as posições de objetos do mundo em 3D são relativas ao zero do mundo. Entretanto, para tornar as coisas mais simples, adota-se um conjunto de espaços específicos.



- Object Space ou Modelling Space (Espaço local) é o sistema de coordenadas relativas a um objeto (ou grupo de objetos).
- World Space (Espaço Global) engloba todo o universo e permite exprimir as coordenadas de forma absoluta. É neste espaço que os modelos são compostos para criar o mundo virtual.
- Camera Space (Espaço da Câmara) Este sistema de coordenadas esta associado ao observador, ou câmara. A sua origem é a posição da câmara e seu sistema de eixos é determinado pela orientação da câmara.
- Screen Space (Espaço de Tela) Espaço 2D onde é visualizado o mundo virtual.

2.3 Vetores

Você também poderá ver vetores 3D descritos em coordenadas cartesianas. Assim como seus homólogos 2D, os vetores 3D são simplesmente linhas desenhadas no mundo 3D que tem uma direção e um comprimento. Vetores podem ser movidos no espaço do mundo, mas permanecem inalterados. Vetores são úteis no contexto de uma *game engine*, já que eles permitem calcular distâncias, ângulos relativos entre objetos, e a direção dos objetos.

2.4 Câmeras

Câmeras são essenciais no mundo 3D, como elas agem como o *viewport* (janela de exibição) da tela. Tendo uma forma de pirâmide de visão (Frustum), as câmeras podem ser colocadas em qualquer ponto do mundo, animadas, ou anexadas aos personagens ou objetos, como parte do cenário do jogo. Com um campo de visão ajustável, as câmeras 3D são a janela de exibição do mundo 3D. As ferramentas de modelagem, informam quais efeitos – como luz, borrões de movimento, entre outros – que são aplicados a câmera, para ajudar com a simulação do jogo de como o olho do usuário tem a visão do mundo.

2.4 Polígonos, arestas, vértices e malhas

Na construção de formas 3D, todos os objetos geralmente são compostos de formas 2D interconectadas conhecidas como polígonos. Polígonos triangulares (também referidos como faces) por sua vez são compostos por três arestas. Os locais em que esses vértices se encontram são conhecidos como pontos ou vértices, ver Figura 2. Por conhecer esses locais, *game engines* são capazes de fazer cálculos sobre os pontos de impacto, conhecidos como colisões, quando usamos detecção complexa de colisão com *Mesh Colliders* (Colisores de Malha), tais como em jogos de tiro para detectar a localização exata em que uma bala terá atingido outro objeto.



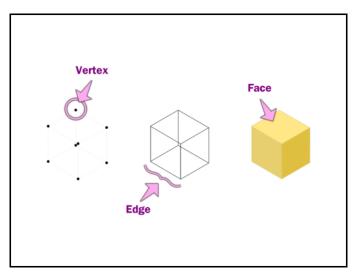


Figura 2. Elementos de modelagem do Blander.

Ao combinar vários polígonos ligados, aplicativos de modelagem 3D nos permitem construir formas complexas, conhecidas como malhas. Além de construir formas 3D, os dados armazenados nas malhas podem ter outras formas de uso, por exemplo, pode ser usado como dados de navegação de superfície tornando os objetos no jogo seguindo os vértices.

Em um projeto de modelagem, é crucial para o desenvolvedor compreender a importância da contagem de polígonos. A contagem de polígonos é o número total de polígonos, muitas vezes em referência a um modelo, mas também em referência a um nível de jogo inteiro ou animação. Quanto maior o número de polígonos, maior será o trabalho o custo computacional para renderizar os objetos na tela. Devido ao aumento do poder computacional e a redução dos custos do hardware, na última década, houve um aumento no nível de detalhamento e realismo dos jogos e animações. Como resultado da evolução da tecnologia, desenvolvedores de jogos agora são capazes de modelar personagens 3D e mundos de jogos que contenham uma contagem de polígonos muito maior e isso tende a continuar inevitavelmente.

3. Iniciando técnicas de modelagem

No Blender, dentre outras ferramentas, existem formas primitivas que servem como base para formar desde o modelo mais simples ao mais complicado. Estas formas estão representadas na Figura 3.

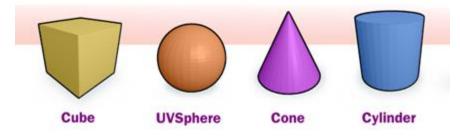


Figura 3. Formas primitivas 3D do Blander.



Através de transformações que deformam o objeto ou a forma primitiva, é possível conseguir grandes resultados como o mostrado da Figura 04, através dos comandos de *Extrude* (E) e *Scale* (S) e *Translate* (G) conseguiu-se um ótimo resultado.

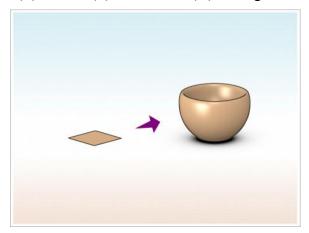


Figura 4. Criação de pote por meio de aplicação de deformações no Blender.

Porém antes de tudo é necessário saber manipular os objetos na cena e conhecer um pouco da interface do software, para selecionar um objeto clica-se com o botão direito do mouse sobre ele como mostra a Figura 5.



Figura 5. Exemplo de manipulação de Objetos no Blender.

Para editar um objeto em cena basta apenas apertar a tecla TAB, ou simplesmente ir ao menu e selecionar o modo de edição como mostra a Figura 6.

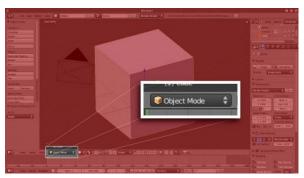


Figura 6. Exemplo de edição no Blender.



No modo de edição é possível selecionar vértices, arestas e faces (*vertex*, *edge* e *face*), para selecionar mais de um vértice basta pressionar o botão SHIFT + clique com botão esquerdo do mouse, isso aplica-se também as arestas e faces. A Figura 7 apresenta uma sequência de comandos básicos de edição no Blender.

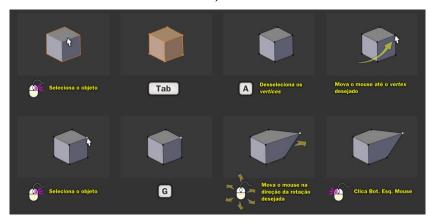
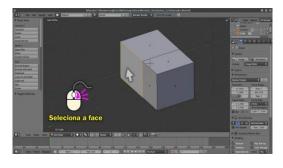
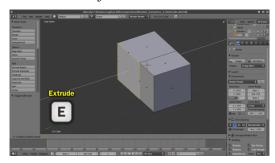
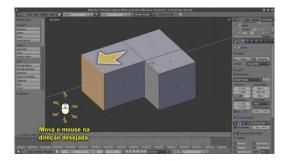


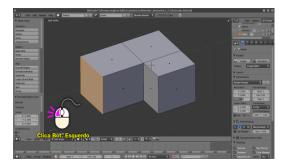
Figura 7. Exemplo de comandos de edição no Blender.

Para todo artista modelador 3D iniciante, é interessante que se saiba como os comandos são realizados e quais seus efeitos, é muito comum os iniciantes seguirem tutoriais que ensinam a modelar espadas, carros, taças e uma diversidade de outros objetos, isso não é errado, mas, conhecer melhor cada comando possibilita o exercício de seu poder de abstração e da técnica de como reproduzir um objeto do mundo real para o virtual no caso o 3D. A Figura 8 exemplifica o uso do comando *Extrude*, a sequência de imagem não tem por objetivo modelar algo representativo do mundo real, mas sim mostrar o uso do comando e seus efeitos em um objeto.

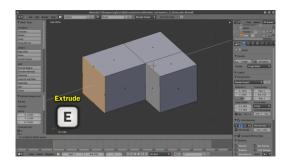


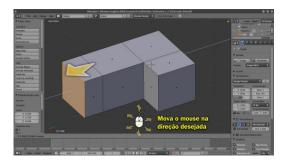


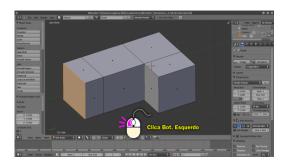


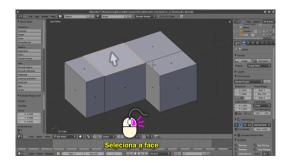


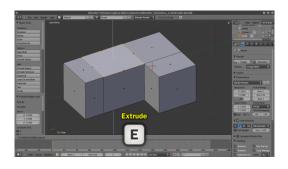


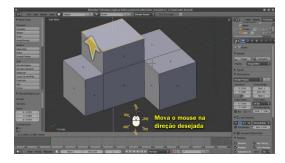


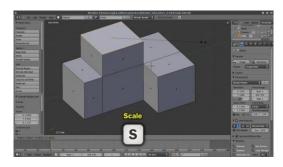












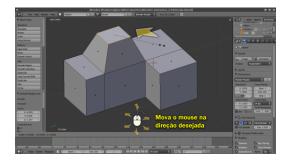


Figura 8 - Exemplo do uso do comando Extrude.

Se prosseguirmos com seqüência de extrusões e escalas para redimensionamentos de faces poderemos obter talvez uma câmera digital através dessas transformações.

4. Conclusão

No mercado existem diversas ferramentas que, assim como o Blender 3D, possibilitam o desenvolvimento de objetos 3D, porém, a ferramenta aqui apresentada destaca-se pelo



seu alto grau de qualidade em desenvolvimento gráfico e por possibilitar executar além de modelagem, projetos de animações, tratamento de imagens, edições de vídeo e jogos. O usuário que aprende a trabalhar com o Blender 3D não terá dificuldade em trabalhar com mais nenhuma outra ferramenta que possibilite fazer as mesmas tarefas, o exercício no uso da ferramenta é essencial para quem deseja tornar-se um entusiasta da modelagem 3D de objetos, vale lembrar também que existe uma diversa gama de material disponível na internet sobre o Blender que contemplam desde o iniciante ao mais avançado.

Referências

- [1] Blender. Disponível em http://www.blender.org, acessado em 09/02/2014.
- [2] Blender. Disponível em http://orange.blender.org/download, acessado em 09/02/2014.
- [3] AutoDesk. Disponível em http://www.autodesk.com.br/products/autodesk-3ds-max/overview, acessado em 09/02/2014.
- [4] AutoDesk. Visão Geral (OverView). Disponível em http://www.autodesk.com.br/products/autodesk-maya/overview, acessado em 09/02/2014.
- [5] Blender. Disponível em http://wiki.blender.org/, acessado em 12/02/2014.
- [6] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes. Computer Graphics: Principles and Practice in C, 2nd Edition, Ed Hardcover, 2001.



AppDay ERIPE 2013

João Ferreira da Silva Júnior¹, Severino José da Costa Neto¹, Sérgio Francisco Tavares de Oliveira Mendonca¹

¹ Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n, Boa Vista - 55.292-270 - Garanhuns - PE - Brasil {joaoferreirape, sjcostaneto}@gmail.com, sftom@uag.ufrpe.br

Resumo. O objetivo deste trabalho é apresentar o AppDay, uma iniciativa da Microsoft do Brasil para promover o desenvolvimento de aplicações e as novas tecnologias embarcadas no Microsoft Windows 8. Formatado como um minicurso de um dia, o AppDay será ministrado por João Ferreira e Severino Neto, alunos do curso de Bacharelado em Ciência da Computação e membros do programa Microsoft Student Partners, durante a III Escola de Informática de Pernambuco – ERIPE 2013, que ocorrerá entre os dias 06, 07 e 08 de novembro de 2013.

Abstract. The objective of this study is to present AppDay, Microsoft's initiative in Brazil to promote the development of applications and new technologies embedded in Microsoft Windows 8. Formatted as a one-day short course, the AppDay will be taught by Severino Neto and João Ferreira, students of Bachelor of Computer Science and members of the Microsoft Student Partners program, during the III School of Informatics of Pernambuco - ERIPE 2013, that will occurs between days 06, 07 e 8 November 2013.

1. Programas Microsoft de Incentivo à Inovação

A Microsoft investe em inovação, auxiliando estudantes de todo o mundo a desenvolverem uma carreira de formação sólida e voltada para o mercado. Para tanto, oferece diversos programa de inovação, por hora citemos aqueles que estão mais relacionados ou serão tratados durante este minicurso [1].

O programa MSP – Microsoft Student Partner é voltado para estudantes da área de tecnologia e visa estimular competências que promovam o desenvolvimento de habilidades técnicas, acadêmicas e sociais. Criado em 2002, o programa se desenvolve em torno de um objetivo comum, professores, estudantes e profissionais. Os membros mais antigos no programa têm um papel essencial na tutoria do programa. Este trabalho em conjunto com a comunidade acadêmica visa promover o uso efetivo das tecnologias de informação e comunicação por meio do acesso às ferramentas tecnológicas e de capacitação para o domínio e sua aplicação no dia-a-dia.

O *DreamSpark* é um programa da Microsoft que dá suporte a educação técnica fornecendo acesso a software para fins de aprendizado, ensino e pesquisa. Está presente em mais de 80 países, tendo auxiliado 35 milhões de estudantes em todo o mundo.



- O *DreamSpark* é simples: ele se destina a dar a alunos ferramentas de designer e de desenvolvedor de nível profissional da Microsoft sem custo, para que os alunos possam realizar seus sonhos e criar a próxima grande novidade tecnológica, ou apenas um bom início de carreira.
- O *DreamSpark* ajuda a educadores a ensinarem as tecnologias mais recentes e as experimentarem em pesquisas. A Microsoft sabe que, para tornar o aprendizado mais motivante, relevante e interessante para os alunos de hoje em dia, é preciso haver um conjunto variado de recursos. O *DreamSpark* dá aos educadores os recursos para garantir que suas salas de aula sempre tenham as tecnologias mais recentes para desafíar, motivar e manter os alunos interessados de novas maneiras.
- O *DreamSpark* também é uma assinatura para instituições acadêmicas: ele fornece uma maneira econômica de colocar as ferramentas, as plataformas e os servidores de desenvolvedor da Microsoft em laboratórios, salas de aula, e nos computadores dos seus alunos e corpo docente com fins de aprendizado e pesquisa. Ele reduz os custos de laboratório e o orçamento de ensino

A MVA - Microsoft Virtual Academy oferece treinamento online realizado por especialistas, com centenas de cursos, em 11 idiomas diferentes. A missão aqui é ajudar desenvolvedores, profissionais de TI e estudantes a aprender sobre a mais moder tecnologia, desenvolver suas habilidades e progredir em sua carreira.

2. O Mercado de Apps

Estudo realizado pela DISTIMO entre maio de 2011 e naio de 2012, mostra que a produção de aplicativos para dispositivos móveis cresceu 83% no Brasil. Somos considerados o sexto maior mercado do setor, e o maior da América Latina [2]. O mercado nacional de soluções móveis corporativas dobra de tamanho a cada ano [3].

Ademais, o brasileiro gasta em média, 84 minutos por dia usando o smartphone, o que representa quase 15% a mais do que os usuários de outros países, que ficam em média 74 minutos com o aparelho nas mãos [4].

3. O AppDay

Somente em sua primeira rodada no Brasil, o AppDay atingiu 119 cidades em 5 regiões, contando com uma infraestrutura on e off-line. O conteúdo técnico trouxe informações para o desenvolvimento da primeira app, entre os temas estavam: Criando uma conta na Windows Store, como criar uma app através de esboços ou templates do Codeplex, além de dicas e truques para publicar a primeira App.

A maior gratificação do evento é a inclusão digital promovida através de influenciadores, que alcançaram lugares e pessoas nunca alcançados antes, gerando atualização profissional e a oportunidade de participar de um treinamento de alto nível om os especialistas técnicos da Microsoft.

3.1. O AppDay em Números

- 4 salas online com a capacidade máxima de 1,250 participantes atingida.
- 1 sessão extra para endereçar FAQs (Frequently Asked Questions)



- 54 influenciadores engajados, MSPs, MVPs e MTACs.
- 16 horas de ensaio
- 8 horas de reunião para preparar os influenciadores
- Todo o time de Marketing e de Técnicos respondendo perguntas durante 8 horas.
- + 4,000 participantes online
- + 3,000 participantes off-line
- +13,000 registros de evento
- +1,000 seguidores no Twitter
- + 900 novos membros no grupo técnico de discussão no Facebook.
- + 10.000 visualizações de "Ideias para sua App"
- + 6.000 acessos de conteúdo
- 119 cidades participantes no Brasil.

4. Preparação do ambiente

Para possuir uma experiência completa do desenvolvimento desses aplicativos, teremos alguns requisitos de sistema. Para a criação de aplicativos para plataforma *Windows 8*, teremos que possuir o sistema operacional *Microsoft Windows 8* seguido do *SDK – Software Development Kit.* Ambos estão disponíveis para download, gratuitamente, através do link http://goo.gl/qPPLM9.

4.1 Instalação do Windows 8

Temos duas opções para a instalação do sistema operacional, a primeira é uma instalação de upgrade, para quem possui os sistemas operacionais *Microsoft Windows 7* ou *Windows Vista*, onde a única intervenção do usuário será iniciar o processo de instalação, após isso, a instalação será feita sozinha. É importante frisar a necessidade de realização de backup dos dados [5].

A segunda opção é fazermos uma instalação limpa, onde teremos que dar o boot do computador pelo drive de CD/DVD, onde o sistema de instalação do *Windows 8* inicializará conforme a "Figura 11. Windows 8 setup".



Figura 11. Windows 8 setup



Na tela de setup teremos algumas opções importantes para a instalação e configuração do sistema, a primeira é escolher a linguagem do sistema operacional, a segunda é a escolha da região e a terceira opção é o formato do teclado. Após essas configurações vamos prosseguir clicando no botão próximo. Trata-se de uma etapa bastante intuitiva e fácil de compreender. Ao passar dessa tela, o *Setup* fará o pedido da chave do produto, necessária para a ativação, o código apresenta 25 dígitos alfanuméricos. Mas vamos prosseguir sem digitar a chave do produto clicando no botão próximo onde veremos uma mensagem informando sobre o uso do sistema operacional por 90 dias.

Após confirmar que concorda com os termos de uso do sistema operacional, vamos escolher que tipo de instalação iremos fazer, vamos ter duas opções, a de upgrade como já foi explicado, e a segunda opção "customizada" a qual iremos escolher. Assim vai ser apresentada a tela partições, onde vai estar listada todas as partições presentes do HD, a "Figura 12. Tela de partições" demonstra essa fase da instalação.

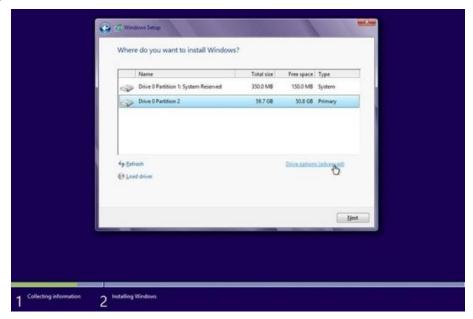


Figura 12. Tela de partições

Selecione a partição do seu HD e apague clicando em "Formatar". Lembre-se que os seus dados serão apagados. Após a formatação é só clicar no botão de próximo para iniciar a instalação do sistema operacional. O processo de instalação pode durar entre 10 a 30 minutos. Tudo vai depender do hardware que você possui. A "Figura 13. Tela de instalação" demostra como a tela do seu computador vai estar durante essa fase final.



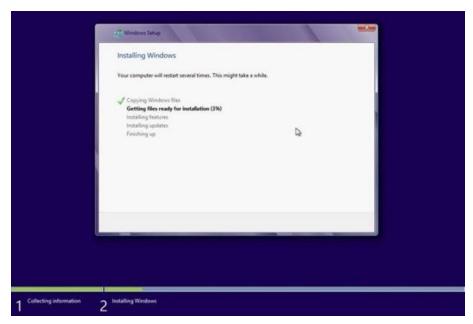


Figura 13. Tela de instalação

Quando sair dessa tela a instalação estará completa, faltando somente efetuar as configurações finais referentes ao seu perfil de usuário, escolhendo a cor de fundo e dando um nome ao seu computador. Se desejar pode ser inserida uma conta de e-mail que será utilizada para efetivar o *login* do usuário. A "Figura 14. Windows 8 Interface Metro" demonstra a tela inicial do *Windows* 8.



Figura 14. Windows 8 Interface Metro

4.2 Instalação do SDK

Só o sistema operacional não é o suficiente para a criação de aplicativos, necessitamos de ferramentas para desenvolvimento, como uma *IDE – Integrated Development Environment*.

Usaremos para este minicurso a IDE oferecida pela *Microsoft* chamada *Visual Studio Express 2012*. Para fazer o download desta ferramenta acesse o link http://goo.gl/2Z2cnG. Após *download*, a instalação é fácil e intuitiva. Primeiramente



temos que escolher o local de instalação, feito isso os próximos passos são informativos onde será necessário apenas clicar no botão *next* e aguardar a conclusão do processo. Ao finalizar a instalação do *Visual Studio Express*, já poderá ser executado quando será solicitado o registro. Frise-se que o registro da versão *Express* é gratuito. Agora você possui tudo que é necessário para o desenvolvimento de aplicativos para a plataforma *Windows* 8.

5. Windows 8 App Day - Template Catálogo Eletrônico

Este projeto visa facilitar o desenvolvimento da sua *App* para o *Windows 8*, não tente publicar uma *App*, sem antes pensar em seu propósito e muito menos deixe de alterar o *templates*.

Se a *App* que deseja utilizar não enquadrar neste *template*, veja os outros disponíveis na página https://www.meetmyapp.in/meetmyapp/windows_8_build.aspx ou mesmo no *Codeplex* https://www.codeplex.com/site/users/view/win8templates.

Neste projeto é apresentado o template *Catálogo Eletrônico*, com o qual é possível criar um catálogo, apresentando produtos ou serviços de qualquer ramo.

Características do *template*:

- O *template* inclui algumas funcionalidades como o carregamento de dados do *XML*, contrato de busca, contrato de *share source* e o contrato de *settings*.
- O template não consome dados de outras fontes que não seja a própria app, ou seja, não consome serviços da internet.
- Os dados utilizados neste template são extraídos de arquivos *XML* presente no projeto da *App*, assim como as imagens que são utilizadas.



Figura 15 - Template CodePlex

6. Windows Store

Com o *Windows* 8, a Microsoft lançou a *Windows Store*, loja de aplicativos da empresa para o sistema. Vamos mostrar como acessar a loja e instalar aplicativos.



Após reservar o nome do aplicativo, configurar a conta de pagamento, criar os pacotes do aplicativo e testar o aplicativo com o *Kit de Certificação de Aplicativos Windows*, você estará preparado para iniciar o processo de envio.

6.1. Reservando um nome de aplicativo

Você pode reservar um nome de aplicativo antes de enviar o aplicativo (ou antes mesmo de criá-lo). Após reservar o nome, você tem um ano para enviar o aplicativo à *Windows Store*. Se o envio não for feito dentro desse período, você perderá o nome reservado, e ele será disponibilizado para outro desenvolvedor que queira usá-lo.

Observação: Certifique-se de ter os direitos de uso do nome reservado. Caso alguém tenha registrado o nome, ele poderá relatar uma infração, e você não poderá continuar usando o nome.

Para reservar um novo nome de aplicativo:

- 1. Vá para o Painel da *Windows Store*. (Você pode fazer isso clicando no link "Painel" próximo ao topo de qualquer página no *Centro de Desenvolvimento de Aplicativos* da *Windows Store*.)
- 2. Clique em Enviar um aplicativo.
- 3. Clique em Nome do aplicativo.
- 4. Na caixa de texto Nome do aplicativo, digite o nome que deseja usar. Observação: Informe o nome exatamente como deseja que apareça. Revise ortografia, maiúsculas e minúsculas, etc.
- 5. Clique em Reservar nome do aplicativo.

 Observação: Se o nome digitado já estiver reservado ou sendo usado por outro desenvolvedor, você verá uma mensagem de erro e não poderá reservar o nome.
- 6. Use o nome do aplicativo reservado como *DisplayName* no manifesto do aplicativo.
 - Observação: Esse valor deve ter no máximo 256 caracteres.

Observação: Se você não puder reservar um nome registrado por você ou sobre o qual você possua outros direitos jurídicos (ou se achar outro aplicativo na *Windows Store* com o mesmo nome), fale com a Microsoft.

6.2. Configurando sua conta de pagamento

Para receber dinheiro pela venda de aplicativos na *Windows Store*, você deve configurar sua conta de pagamento e preencher os formulários fiscais necessários. O dinheiro dos aplicativos listados na *Windows Phone Store* é tratado separadamente, você pode configurar uma conta de pagamento e formulários tributários para eles em seu Painel do *Windows Phone*.

Se você pretende listar apenas os aplicativos gratuitos (e não planeja oferecer compras no aplicativo por meio da *Windows Store*), não precisa configurar uma conta de pagamento nem preencher formulários tributários. Se você mudar de ideia mais tarde e decidir que quer vender aplicativos (ou compras no aplicativo), poderá configurar sua conta de pagamento e preencher os formulários tributários nessa ocasião.



Observação: Em certos mercados, apenas aplicativos gratuitos podem ser enviados. Sua conta precisa estar registrada em um desses mercados para que você tenha a opção de configurar uma conta de pagamento.

O processo completo pode ser acessado através do link: http://goo.gl/kVXF55

6.3. Requisitos do pacote do aplicativo

Siga estas diretrizes para preparar pacotes do aplicativo para enviar para a *Windows Store*.

Observação: Examine a seção *Antes de enviar e a Lista de verificação de envio de aplicativos* para garantir que tudo esteja certo antes de enviar pacotes.

Importante: Caso seu aplicativo colete informações pessoais ou use software fornecido por terceiros, você terá que fornecer uma política de privacidade ou termos de licença adicionais junto com o aplicativo. Por padrão, muitos projetos do *Microsoft Visual Studio* declaram o uso da funcionalidade *internetClient*. O uso dessa funcionalidade requer o fornecimento de uma política de privacidade que possa ser acessada pela página de detalhe do aplicativo e pelo painel Configurações. Você pode remover a funcionalidade *internetClient* de seu projeto caso não precise dela.

Antes de compilar o pacote do seu aplicativo para a *Windows* Store, teste seu aplicativo com o *Kit de Certificação de Aplicativos Windows* para conferir se ele está de acordo com os requisitos técnicos de aplicativos da *Windows Store*. Nós recomendamos também, testar seu aplicativo em diferentes tipos de hardware. Até o aplicativo ser certificado e disponibilizado na *Windows Store*, ele só poderá ser instalado e executado em computadores com licenças de desenvolvedor.

O processo completo pode ser acessado através do link: http://goo.gl/EdlpQB

6.4. Processo de envio de aplicativos

Para enviar um aplicativo, você precisa entrar no seu Painel da *Windows Store*. (Acesseo clicando no link "Painel" próximo ao topo de qualquer página no Centro de Desenvolvimento de Aplicativos da *Windows Store*.)

- 1. No Painel da *Windows Store*, clique em Enviar um aplicativo. Em seguida, a página de resumo de versão será exibida, listando as etapas que preparam seu aplicativo para a certificação da *Windows Store*.
 - Observação: Se você já tiver reservado um nome de aplicativo, verá o bloco desse aplicativo no Painel. Se for esse o caso, clique no bloco do aplicativo e não em Enviar um aplicativo.
- 2. Clique em Nome do Aplicativo para digitar e reservar o nome do seu aplicativo, se ainda não o tiver feito.
- 3. Clique em Detalhes de venda para inserir informações como preço, categoria/subcategoria e mercados do seu aplicativo.
- 4. Clique em Serviços para configurar recursos como, por exemplo, notificações por *push* e compras no aplicativo.



- 5. Clique em Classificação etária e certificados de classificações para selecionar a classificação etária para seu aplicativo e carregar certificados de classificações.
- 6. Clique em Criptografía para declarar se o aplicativo usará algum tipo de criptografía.
- 7. Clique em Pacotes para carregar os pacotes do seu aplicativo.
- 8. Clique em Descrição para inserir a descrição do seu aplicativo.
- 9. Clique em Notas para os testadores para fornecer aos testadores do processo de certificação informações que os ajudem testar seu aplicativo.
- 10. Clique em Enviar para certificação para enviar o aplicativo.

Você receberá um e-mail quando o processo de certificação for concluído, também é possível consultar o status de certificação do aplicativo a qualquer momento.

6.4.1. Processo de certificação de aplicativos

Durante o processo de certificação, seu aplicativo passa por diversas etapas:

- Carregamento: o aplicativo inicia o processo de certificação quando você o carrega na loja. Durante o processo de carregamento, verificamos os pacotes do aplicativo em relação à conformidade técnica com os requisitos de pacotes do aplicativo. Se o aplicativo for aprovado nesses testes, você verá uma mensagem de carregamento bem-sucedido na página *Pacotes*. Se ele falhar, você verá uma mensagem de erro. Veja *Solucionando erros de carregamento de pacote* para obter ajuda sobre a correção de problemas de carregamento.
- Pré-processamento: Depois que você carrega os pacotes do aplicativo com êxito e envia o aplicativo para certificação, os pacotes são colocados em fila para testes automatizados.
- Testes de segurança: O primeiro teste verifica se há vírus e *malware* nos pacotes do aplicativo. Se o aplicativo falhar no teste, você precisará conferir o sistema de desenvolvimento executando o software antivírus mais recente e, em seguida, recompilar o pacote do aplicativo em um sistema limpo.
- Testes de conformidade técnica: A conformidade técnica é testada pelo Kit de Certificação de Aplicativos Windows (você deve garantir o teste do aplicativo com o Kit de Certificação de Aplicativos Windows antes de enviá-lo à Loja).
- Conformidade do conteúdo: Nossos testadores de certificação instalam e examinam o aplicativo para testar a conformidade do conteúdo. O tempo necessário varia dependendo da complexidade do aplicativo, da quantidade de conteúdo visual e de quantos aplicativos foram enviados recentemente. Não se esqueça de fornecer todas as informações importantes aos testadores na página, Notas para os testadores.
- Relatório de certificação: Após a conclusão do processo de certificação, você receberá um relatório de certificação que o informa se o aplicativo passou ou não na certificação. Caso seu aplicativo não passe na certificação, o relatório indicará em qual teste ele falhou ou qual requisito de certificação não foi



atendido. Veja *Solucionando erros de certificação* para obter ajuda na solução de problemas, e pode enviar o aplicativo para certificação novamente. Caso a falha tenha ocorrido por falha ou falta de resposta do aplicativo durante os testes, você receberá dados do relatório de falhas para identificar e resolver o problema.

- Lançamento: Quando o aplicativo passa na certificação, ele vai direto para o
 processo de assinatura e publicação, a menos que você tenha especificado que
 ele não deve ser lançado antes de uma data específica.
- Assinatura e publicação: Os pacotes do aplicativo são assinados digitalmente para ficarem protegidos contra falsificação após serem lançados. Após o início dessa fase, você não pode mais cancelar o envio.

Depois de passar pelas etapas acima com êxito, seu aplicativo estará disponível na *Windows Store* para download e você poderá criar links para a página de detalhes do aplicativo. Lembre-se que leva um pouco mais de tempo para que os detalhes do aplicativo comecem a aparecer nos resultados de pesquisa.

Referências

- [1] Microsoft Press, "Microsoft Commits \$50 Million in Higher Education Resources, Training and Certifications to Drive Economic Recovery," [Online]. Available: http://goo.gl/Y4NdqZ. [Acesso em 15 10 2013].
- [2] H. KOEKKOEK, "Emerging App Markets: Russia, Brazil, Mexico and Turkey," 28 06 2012. [Online]. Available: http://goo.gl/C4Eq1O. [Acesso em 23 10 2013].
- [3] S. TAUHATA, "Os segredos de quem ganha milhões com aplicativos," *Pequenas Empresas & Grandes Negócios*, nº 260, 09 2010.
- [4] Acontecendo Aqui, "Mercado de criação de aplicativos pode movimentar US\$ 25 bilhões no Brasil em 2013," 25 09 2013. [Online]. Available: http://goo.gl/pR896r. [Acesso em 25 10 2013].
- [5] J. F. SILVA JR, "SISTEMA RÁPIDO DE FULL BACKUP: Uma alternativa para empresas de suporte técnico," UNOPAR, Garanhuns, 2013.
- [6] A. CHARLANDE e B. LEROUX, "Mobile Application Development: Web vs. Native," *Communications of the ACM*, vol. 54, pp. 49-53, 05 05 2011.
- [7] V. Souza, "Windows8 App Day Template Catálogo Eletrônico," Microsoft, 12 04 2013. [Online]. Available: http://win8appdaytemplate1.codeplex.com/. [Acesso em 2013 10 08].
- [8] Microsoft, "Aplicativos da Windows Store," Microsoft, 2013. [Online]. Available: http://msdn.microsoft.com/pt-br/windows/apps/. [Acesso em 28 09 2013].