

**ENCOINFO 2009**

**XI Encontro de Estudantes de Informática do Tocantins**

12 e 13 de Novembro de 2009

Palmas – Tocantins

**ANAIS**

**Realização**

Curso de Bacharelado em Sistemas de Informação  
Curso Superior Tecnológico em Redes de Computadores

*Nota: Os conceitos e a redação contidos nos artigos e cursos são de exclusiva responsabilidade de seus autores, pois os mesmos foram transcritos na íntegra para esta publicação.*

**ENTIDADE MANTENEDORA**  
**COMUNIDADE EVANGÉLICA LUTERANA SÃO PAULO**

**Presidente**

Augusto Ernesto Timm Neto

**Vice-Presidente**

Joseida Elizabete Timm

**UNIVERSIDADE LUTERANA DO BRASIL**

**Reitor**

Marcos Fernando Ziemer

**Vice-Reitor**

Valter Kuchenbecker

**Pró-Reitor de Administração**

Ricardo Müller

**Pró-Reitor de Graduação**

Ricardo Prates Macedo

**Pró-Reitor de Pesquisa e Pós-Graduação**

Erwin Francisco Tochtrop Júnior

**Pró-Reitor de Extensão e Assuntos Comunitários**

Ricardo Willy Rieth

**Pró-Reitor Adjunto de Graduação**

Pedro Antonio Gonzalez Hernandez

**Capelão Geral**

Gerhard Grasel

**CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

**Diretora Geral**

Kelen Beatris Mânica

**Coordenadora do Núcleo de Atendimento Educacional**

Parcilene Fernandes de Brito

**Assessora do Núcleo de Atendimento Educacional**

Diêmy Sousa Freitas

**Coordenadora de Pesquisa, Pós-Graduação e Extensão**

Conceição Aparecida Previero

**Coordenador da Comissão Própria de Avaliação**

Fabiano Fagundes

**Secretaria Geral**

Lílian Silva de Moura

**Assessor Administrativo**

José Francisco Pereira de Castro

**Assessora da Direção Geral**

Sinara Goiás de Paiva

**Assessora de Comunicação Social**

Raquel dos Santos Sousa

**Capelão**

Ari Schulz



**Bacharelado em Sistemas de Informação**  
**Curso Superior Tecnológico em Redes de Computadores**

**Coordenadora**

Parcilene Fernandes de Brito

**Coordenador de Estágios**

Fabiano Fagundes

**Coordenador de Trabalhos de Conclusão de Curso**

Fabiano Fagundes

**Tutora do Núcleo Especial de Treinamento**

Cristina D'Ornellas Filipakis

**Coordenadora do Projeto Informática & Sociedade**

Cristina D'Ornellas Filipakis

**Coordenadora do Portal Acadêmico de Redes de Computadores**

Madianita Bogo Marioti

**Coordenador do POP NET – Rede Popular**

Ricardo Marx Costa Soares de Jesus

**Coordenador do Programa Pesquisador Jr.**

Fabiano Fagundes

**Coordenadores da Fábrica de Software**

Jackson Gomes de Souza

Fernando Luiz de Oliveira

Edeilson Milhomem da Silva

## **ENCOINFO 2009**

### **Comissão Organizadora**

Cristina D'Ornellas Filipakis  
Edeilson Milhomem da Silva  
Fabiano Fagundes  
Fernando Luiz de Oliveira  
Jackson Gomes de Souza  
Madianita Bogo Marioti  
Marciane Gomes de Souza  
Michael Schuenck dos Santos  
Parcilene Fernandes de Brito  
Ricardo Marx Costa Soares de Jesus

### **Diagramação**

Fabiano Fagundes

### **Arte e Capa**

Jackson Gomes de Souza

### **Encadernação**

Provisão Estação Gráfica Digital

### **Site do Evento**

Fábrica de Software

### **Sistema de Eventos**

Fábrica de Software  
Warley Ferreira Gois

### **Fábrica de Software**

Felipe Eduardo Bechert Schmitz  
Leandro Soares de Sousa  
Valdirene da Cruz Neves Júnior  
Willian Almeida Rodrigues

## **Comissão de Avaliação**

Angela Cristina Corrêa – CPGA/UFSC

Antonio Braz de Oliveira e Silva - IBGE/NITEG/UFMG

Antônio da Luz Junior - NPDI/DCC/UFMG

Bruno S. C. M. Vilar - IC/UNICAMP

Camila de Araujo – UERN

Daniel Costa de Paiva - USP

Demóstenes Santos de Sena - CEFET/RN

Eliane Pozzebon - UNIVASF

Flavius da Luz Gorgônio - CERES/DCEA/UFRN

Luciana de Oliveira Rech - INE/UFSC

Márcio Ribeiro - CIN/UFPE

Ricardo Furtado Rodrigues - UNICAMP

Sérgio Santos - DIMAP/DEE/UFRN

Thatiana Cunha Navarro de Souza - DIMAP/UFRN





## APRESENTAÇÃO

Quando os algoritmos começam a se aglomerar e a se distanciar da casa das unidades, começamos a ter noção de que o tempo corre e as coisas realmente acontecem, independente de sermos protagonistas ou simplesmente público passivo. Quando o tempo passa percebemos que o que fazemos pode não ser somente um conjunto de ações; pode ser história.

O Encoinfo, agora completando seu primeiro ano na casa das dezenas, ainda novo, se compararmos com a idade de muitos de nós que temos este material em mãos, está no rol daquelas ações que podemos dizer que fazem história.

Por aqui já passaram diferentes tecnologias, desde as que eram promessas incipientes – e hoje se mostram presentes no mundo, como o XML, para ficarmos em somente um exemplo – até aquelas que julgávamos meio esquecidas e, tais como Phoênix, aprontavam-se para ressurgir das cinzas.

Por aqui já passaram alunos, que viraram professores, pesquisadores e orientadores.

Por aqui passaram formatos de arquivos, meios de arquivamento, formas de diagramação, tipos de indexação. Até diferentes tipologias de fontes fazem parte dos que aqui deixaram sua impressão.

Mesmo tão novo, o Encoinfo partiu de uma época em que o celular era um avanço tecnológico recém surgido para um contexto em que este pequeno aparelho já possui mais memória do que os computadores existentes em suas primeiras edições. Do armazenamento em “discos pesados” para o armazenamento nas... nuvens!

Os que hoje estão com seus nomes registrados, alunos, professores, parceiros e apoiadores, com certeza serão parte da história do Encoinfo. Mas, mais importante do que isso, é saber que o Encoinfo, de alguma forma, é história na vida de muitos que por aqui passaram.

A todos, nosso agradecimento.

Comissão Organizadora

"A história do mundo é, essencialmente, história de idéias."

H. G. Wells



## SUMÁRIO

<b>Palestras .....</b>	<b>15</b>
<b>Minicursos</b>	
Conceitos Básicos de Algoritmos Genéticos: Teoria e Prática .....	19
<i>Thatiane de Oliveira Rosa, Hellen Souza Luz</i>	
Desenvolvimento de aplicações web com JSP .....	19
<i>Leandro Soares de Sousa, Paulo Henrique de Sousa</i>	
Desenvolvimento de Aplicações Web com PHP .....	20
<i>Felipe Eduardo Bechert Schmitz, Valdirene da Cruz Neves Júnior</i>	
Django, Desenvolvimento Ágil para a Web.....	20
<i>Vinicius Prado da Fonseca, Felipe Mendes Braga</i>	
Adobe Flash 8.....	21
<i>Gedilson Pessoa da Silva</i>	
Introdução ao XML.....	21
<i>Charles Alex Rockenbach, Lucas Moreno de Araujo, Moisés Lima dos Anjos, Roneylson de Alencar Neiva</i>	
Persistência de Dados em Java com JPA e Toplink .....	22
<i>Vinicius Teixeira Dallacqua</i>	
Qualidade de Software.....	22
<i>André Mesquita Rincon</i>	
VPN – Virtual Private Network.....	23
<i>Andrew Silva de Lima, Madianita Bogo, Ricardo Marx Costa Soares de Jesus</i>	
<b>Resumos</b>	
Comparativo entre protocolos de gerenciamento de mobilidade .....	27
<i>Helder Cleber Almeida Pereira</i>	
Comportamento de personagens em jogos de computador .....	27
<i>Deise Miranda Borges, Rafael Gonçalves Barreira, Jackson Gomes de Souza</i>	
CompWeb: Um Repositório de Componentes de Software Acessível por Web.....	28
Services REST	
<i>Leandro Oliveira Ferreira, Michael Schuenck</i>	
Controle de uma Estufa Agrícola Utilizando Controlador Lógico Programável .....	28
<i>Nathália Amaral Carvalho, Héverton de Melo Bezerra, Antonio Jeronimo Netto, Marcelo Lisboa Rocha</i>	

Desenvolvimento da Modelagem de uma Ferramenta para Gerenciar aplicações SaaS <i>Valéria Pereira Mota, Fernando Luiz de Oliveira, Isabel Pires da Silva, Roberto Mayer</i>	29
Estimativa de Tamanho de Software Utilizando APF e a Abordagem NESMA <i>Werley Teixeira Reinaldo, Cristina D'Ornellas Filipakis</i>	29
Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático <i>Márcia Savoie, Lucyano Martins, Mayton Rocha, Cirlene dos Santos</i>	30
Implementando maturidade e agilidade em uma fábrica de software através de Scrum e MPS.BR nível G <i>Fernando Szimanski, Jones Albuquerque, Felipe Furtado</i>	30
Interfaces Dinâmicas e Adaptativas para Celulares <i>Flavio Henrique Cardoso de Freitas, Silvano Maneck Malfatti</i>	31
Proposta de um Algoritmo Genético para o Problema de Árvore de Steiner Euclidiana Tridimensional <i>Marcelo Lisboa Rocha, Wellington Gomes Ribeiro</i>	31
Proposta de um Jogo Educativo para Simular o Corpo Humano baseado em Agentes Emocionais Hedonistas <i>Deise Miranda Borges, Douglas Neves de Jesus, Luane Gomes Cunha, Naara Soares Negre, Fernando Luiz de Oliveira</i>	32
Uma Abordagem no Uso de Rede de Energia Elétrica para Transferência de Dados <i>Timote Lima da Silva</i>	32
União Rede Social e Marketing Viral para ampliação da concorrência em processo seletivo (vestibular) de uma Instituição de Ensino Superior <i>Heitor de Sousa Miranda, Parcilene Fernandes de Brito, Fabiano Fagundes</i>	33
Uso do Netkit no Ensino de Roteamento Estático <i>Nyl Marcos Soares Barbosa, Moisés Lima dos Anjos, Madianita Bogo</i>	33
Utilização de Agentes Inteligentes no desenvolvimento de um Jogo RPG para dispositivos móveis <i>Heitor de Sousa Miranda, Fernando Luiz de Oliveira</i>	34
Utilização do Card Sorting na Implementação de um Sistema de Balanced Scorecard para Instituições de Ensino Superior <i>Rafael Silvestre de Souza Vilela, Katia Moraes de Carvalho, Fabiano Fagundes</i>	34
<b>Programação</b>	<b>35</b>
<b>Anotações</b>	<b>39</b>

## **PALESTRAS**



## **Empreendedorismo e Inovação: o Caso do C.E.S.A.R**

Prof. Dr. Sergio Cavalcante

*C.E.S.A.R./CIN-UFPE*

Tem PhD em Engenharia Eletrônica pela University of Newcastle Upon Tyne - Inglaterra (1997), mestrado em Ciências da Computação pela Universidade Federal de Pernambuco - UFPE (1990) e graduação em Engenharia Elétrica também pela UFPE (1986). Atualmente é CEO do C.E.S.A.R - Centro de Estudos e Sistemas Avançados do Recife, professor adjunto 1 do Centro de Informática da Universidade Federal de Pernambuco - Cin/UFPE, presidente do Comitê de Small Business e membro do conselho da AMCHAM - Câmara Americana de Comércio - Seção Pernambuco. Anteriormente foi Diretor-Presidente do C.E.S.A.R (2002-2005), Diretor de Divulgação e Marketing da Sociedade Brasileira de Computação - SBC (2001-2005) e Chefe do Depto de Ciência da Computação do Cin/UFPE (2002-2005). Como pesquisador atua em Tecnologias da Informação e Comunicação, com ênfase em Sistemas Embarcados e de Tempo Real.

## **Arquitetura de Informação e Usabilidade**

Prof. M.Sc. Guilherme Reis

*WebMotors/ECA-USP*

Consultor de Arquitetura de Informação e Usabilidade com vasta experiência área de web. Coordenou projetos de intranet, internet e CRM na Gessy Lever e Sensormatic. Coordenou a área de Arquitetura de Informação no Banco Real. Na Try Consultoria e Pesquisa foi gerente da equipe de testes de usabilidade responsável por coordenar pesquisas de usabilidade e projetos de Arquitetura de Informação para clientes como Shopping UOL, Gerdau, Citibank, Wal-Mart, Submarino, Shoptime, Redecard, entre outros. Ministra cursos sobre Arquitetura de Informação de curta duração pela Jump Education e disciplinas de Arquitetura de Informação e Usabilidade na nos cursos de pós-graduação da Unicid e Impacta. É membro dos comitês organizador e científico do EBAI - Encontro Brasileiro de Arquitetura de Informação. É graduado em Engenharia Elétrica - Ênfase Computação pela Escola Politécnica da Universidade de São Paulo (1996), pós-graduação em Marketing pela Escola Superior de Propaganda e Marketing (2000) e mestre em Ciência da Informação pela Escola de Comunicações e Artes da Universidade de São Paulo (2007). Atua principalmente nos seguintes temas: arquitetura de informação, usabilidade, user experience, design de interação, web design, internet, design centrado no usuário e interação humano-computador.

## **ITIL na Governança de TI**

Ricardo Nascimento

*CRP Consultores*

Bacharel em Ciência da Computação e Mestrando em Ciência da Computação. 10 anos de experiência. Consultoria em grandes Clientes: USP, Bradesco Cartões, Governo de Angola. Professor na disciplina de governança de TI da pós-graduação em Gestão da Tecnologia da Informação na Uniminas, Uberlândia-MG. Certificações: Aurus Software, Oracle, ITIL Foundation.





## **MINICURSOS**



# Conceitos Básicos de Algoritmos Genéticos: Teoria e Prática

Thatiane de Oliveira Rosa<sup>1</sup>, Hellen Souza Luz<sup>2</sup>

<sup>1</sup> Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas  
(CEULP/ULBRA)

Caixa Postal 160 – 77.054-970 – Palmas – TO – Brasil

<sup>2</sup> Curso de Ciências da Computação – Universidade Federal do Tocantins (UFT)

Caixa Postal 114 – 77.001-090 – Palmas – TO – Brasil

{thatiane, hellen.luz}@gmail.com

**Resumo.** *Algoritmo Genético (AG) consiste em uma técnica de Inteligência Artificial (IA) que se fundamenta em teorias e conceitos da genética e da evolução de populações de seres vivos. A implementação de tal técnica visa possibilitar que soluções ótimas, sejam encontradas para resolver problemas, do mundo real, detectados em diversas áreas do conhecimento. Assim, este trabalho apresenta os principais fundamentos teóricos relacionados a Algoritmos Genéticos, tais como: sua definição e etapas de execução, como também um exemplo prático, onde um problema do mundo real é solucionado utilizando como base a técnica de AG.*

## 1. Introdução

Criada recentemente, a disciplina de Inteligência Artificial (IA) possui diversas frentes de estudo, ou seja, é formada por vários ramos, os quais, segundo Russell e Norvig (2004, p 4), estão relacionados aos processos de pensamento e raciocínio, à forma de comportamento, à reprodução fiel do desempenho humano e à imitação da racionalidade. Deste modo, esta disciplina oferece soluções interessantes para problemas de diversos campos, sendo de grande relevância para qualquer área que trate de atividades que dizem respeito ao intelecto. Dentre as várias áreas que se baseiam em IA, podem-se destacar o processamento de linguagem natural, detecção de diagnósticos, planejamento autônomo e escalonamento, planejamento logístico, jogos, robótica e visão computacional.

Para que seja possível implementar tais aplicações, a IA oferece uma gama variada de técnicas, das quais as mais utilizadas são Lógica Fuzzy, Redes Neurais Artificiais, Sistemas Baseados em Casos, Sistemas Especialistas e Algoritmos Genéticos.

De acordo com o contexto deste trabalho, será estudado de forma mais detalhada, a técnica de Algoritmo Genético (AG), que consiste em uma área de estudo da computação evolutiva, que utiliza como base os conceitos de comportamento adaptativo (GOLDBERG e HOLLAND, 1988 apud BITTENCOURT, 2008) formalizados por Darwin, em sua Teoria da Evolução Natural. Vale ressaltar, que a computação evolutiva é um ramo da computação natural que aplica conceitos da natureza para soluções de problemas computacionais complexos, propondo uma forma alternativa para resolução desses problemas, sem o conhecimento prévio de como chegar a uma solução.

Desta forma, pelo fato deste trabalho estudar a adoção da técnica de AG na resolução de problemas do mundo real, o próximo tópico apresenta definição e funcionamento associados ao Algoritmo Genético.

## **2. Algoritmo Genético**

Algoritmo Genético é uma técnica de IA que, assim como definido por Goldberg (1989 apud TICONA, 2003, p. 46), foi criada com o intuito de imitar determinados processos observados na evolução natural das espécies. Deste modo, tal técnica fundamenta-se nas explicações oferecidas por Charles Darwin a respeito da seleção e evolução dos indivíduos na natureza, como também, em outras teorias de genética formuladas, posteriormente, por estudiosos tais como Gregor Mendel. Assim, o objetivo da técnica de AG consiste em solucionar problemas do mundo real de forma otimizada, sendo aplicável a diversas áreas.

Para compreender o funcionamento dos AGs faz-se necessário realizar uma analogia à explicação sobre a evolução das espécies. Assim, o AG trabalha da seguinte forma:

- Inicialmente é gerada uma população formada por um conjunto aleatório de indivíduos, que podem ser vistos como possíveis soluções do problema.
- Durante o processo evolutivo, esta população é avaliada, sendo que para cada indivíduo é atribuída uma nota, ou índice, que reflete sua habilidade de adaptação a determinado ambiente.
- Uma porcentagem dos indivíduos mais adaptados é mantida, enquanto os outros são descartados.
- Os membros mantidos pela seleção podem sofrer modificações em suas características fundamentais por meio de cruzamentos (*crossover*), mutações ou recombinação genética gerando descendentes para a próxima geração.
- Este processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada. Embora possam parecer simplistas do ponto de vista biológico, estes algoritmos são suficientemente complexos para fornecer mecanismos de busca adaptativos poderosos e robustos.

Após apresentar uma definição e explicar, de forma geral, o funcionamento dos Algoritmos Genéticos, é necessário discorrer a respeito dos vários processos executados dentro do mesmo. Sendo assim, o próximo tópico apresenta um estudo a respeito de tais processos, denominados operadores genéticos.

### **2.1. Operadores Genéticos**

Os operadores genéticos têm por objetivo realizar transformações em uma população, fazendo com que, a cada nova geração, indivíduos cada vez mais capazes sejam criados, contribuindo assim para que as populações evoluam a cada nova geração. Com isto, os operadores genéticos são classificados em: inicialização, função de aptidão, seleção, cruzamento, mutação, atualização e finalização. Sendo que destes, destacam-se os de seleção, cruzamento e mutação, responsáveis por conduzirem a busca no sentido da detecção da melhor solução.

#### **2.1.1 Inicialização**

Um Algoritmo Genético tem início a partir da execução do operador denominado inicialização. Tal operador consiste na criação de uma população inicial, na qual os demais serão aplicados. Em geral, pode-se dar início a uma população de forma aleatória, onde os cromossomos são gerados randomicamente, de forma aleatória e com nicho, em que os cromossomos são gerados e classificados em espécies ou ainda; determinística, em que os cromossomos são gerados de acordo com uma função heurística. Normalmente este operador gera uma população de forma aleatória, objetivando aumentar sua diversidade genética garantindo, desta forma, um maior alcance do espaço de busca. Caso a inicialização da população não ocorra de forma randômica, esta poderá convergir prematuramente, isso significa que em um curto espaço de tempo a população possuirá indivíduos muito semelhantes, ou seja, com pouca diversidade genética, o que dificultará na escolha da melhor solução possível para o problema estudado.

Após a criação da população inicial, é necessário avaliar todos os cromossomos gerados. Desta forma, a próxima seção define a função a ser executada para realizar tal avaliação.

### **2.1.2 Função de Aptidão**

No mundo real, a aptidão é exercida pelo meio em que o indivíduo vive. Quando acontece uma modificação no meio e determinado indivíduo não se encontra apto para sobreviver ali, ele tem menos probabilidade de se reproduzir e assim, passar os seus genes para as próximas gerações, tendendo à extinção.

Nos problemas de busca e otimização deve-se também determinar o quão bom é uma solução (indivíduo), para que se possa definir se esta contribuirá para a resolução do problema, esse trabalho é realizado pelo operador de função de aptidão. Assim, este operador confere uma nota para cada cromossomo de acordo com o problema. Sendo tal nota, posteriormente, utilizada no operador genético de seleção.

O cálculo da função de aptidão é o único elo entre o algoritmo genético e o problema proposto, é a única parte não genérica do AG e deve ser capaz de identificar todas as restrições e objetivos, ou seja, a função de aptidão deve ser específica para cada problema.

Existe uma distinção entre os termos função de avaliação e função de aptidão, onde: a função de avaliação é a responsável por fornecer uma medida de desempenho, no que diz respeito a um conjunto particular de parâmetros (LIMA, 2008, p 12), ou seja, determina a probabilidade do indivíduo transmitir os seus genes para a próxima geração.

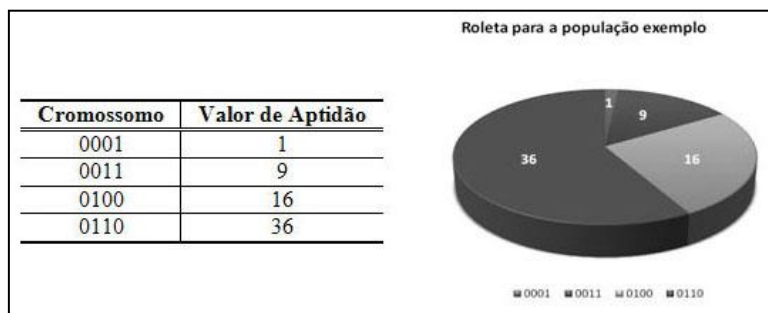
Já a função de aptidão é definida de acordo com outros membros da atual população em um algoritmo genético (LIMA, 2008, p 13). Assim, a aptidão é definida como  $f_i/f$  onde:  $f_i$  é a avaliação associada ao indivíduo  $i$  e,  $f$  é a avaliação média de todos os indivíduos na população. A aptidão pode também ser associada à classificação de um indivíduo ou outras medidas como seleção por torneio Goldberg (1990 apud LIMA, 2008, p. 13). A função de aptidão pode ser igual à função avaliação, ou resultado do escalonamento da função avaliação, ou baseada no *ranking* do indivíduo da população, ou ainda por meio de normalização linear Goldberg (1990 apud LIMA, 2008, p. 13).

### **2.1.3 Seleção**

O operador de seleção é executado logo após o cálculo da aptidão dos cromossomos gerados. Sua implementação é baseada no processo de seleção natural, onde os indivíduos mais capazes possuem maior probabilidade de gerar mais descendentes, enquanto que os menos capazes poderão ainda gerar descendentes, porém em uma escala menor. Logo, é necessário que os cromossomos que possuam um valor de aptidão maior sejam beneficiados, sem que os cromossomos com aptidão menor sejam desconsiderados. Desta forma, a seleção não deve ser baseada somente na escolha do indivíduo mais capaz, pois existe a probabilidade de um indivíduo menos capaz possuir propriedades genéticas favoráveis à geração de um cromossomo que possua a melhor solução para o problema analisado, considerando que tais propriedades favoráveis não estejam presentes nos demais cromossomos da população.

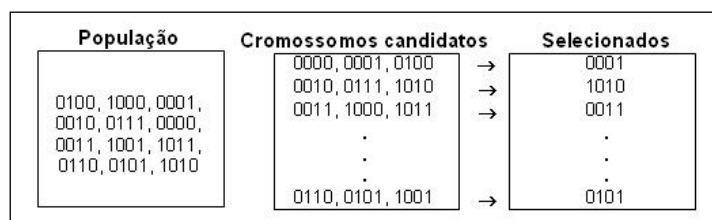
Existem várias estratégias de seleção, este trabalho apresenta três:

- **Roleta:** Nesta técnica, a seleção dos cromossomos ocorre de forma proporcional ao seu valor de aptidão, sendo que os cromossomos de uma população são representados em uma roleta, onde ocupam um espaço proporcional ao seu valor de aptidão. Deste modo, os cromossomos que possuem um alto valor de aptidão ocuparão uma maior fração da roleta, enquanto que os cromossomos com valor de aptidão inferior ocuparão menores frações (Figura 1). Para que seja possível obter o número de pares necessários para a execução dos processos de cruzamento e mutação a roleta é girada quantas vezes forem necessárias.



**Figura 1.** Esquema de funcionamento do método roleta (Modificado de LIDEN, 2006, p. 61-63).

- **Torneio:** A execução do método torneio se dá a partir da escolha de  $n$  cromossomos da população atual, de forma aleatória. Dentre tais cromossomos escolhidos, o com maior valor de aptidão é selecionado para compor uma população intermediária. Em seguida, os demais cromossomos são recolocados na população e realiza-se o mesmo processo até que a população intermediária esteja completa. A figura 2 ilustra o funcionamento de tal método.



**Figura 2.** Esquema de funcionamento do método de torneio.

- **Dizimação:** A dizimação é uma técnica de seleção que realiza a ordenação dos cromossomos de acordo com o seu valor de aptidão e, a partir disto, remove um número fixo de cromossomos que possuem baixo valor de aptidão. Dentre os cromossomos que sobreviveram a este processo de dizimação, escolhem-se os pais de forma aleatória (Figura 3)

Indivíduos	Aptidão	Selecionados
0100	26	0100
1000	15	1000
0010	12	0010
0111	10	0111
<del>0000</del>	<del>2</del>	

**Figura 3. Esquema de funcionamento do método dizimação.**

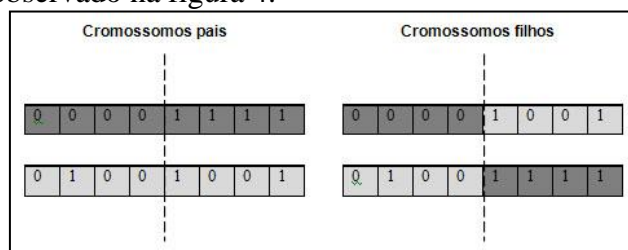
Considera-se que esta técnica possui uma grande desvantagem, pois permite que propriedades genéticas excepcionais sejam perdidas ao excluir cromossomos com baixo valor de aptidão, além de proporcionar a convergência prematura da população.

#### 2.1.4 Cruzamento

O processo de cruzamento é utilizado após a realização da seleção. Nesta fase ocorre a troca de segmentos entre “pares” de cromossomos selecionados para originar os novos indivíduos que virão a formar a população da geração seguinte. Assim, a idéia principal do cruzamento é propagar as características positivas dos indivíduos mais aptos da população por meio da troca de segmentos de informações entre os mesmos, originando novos indivíduos.

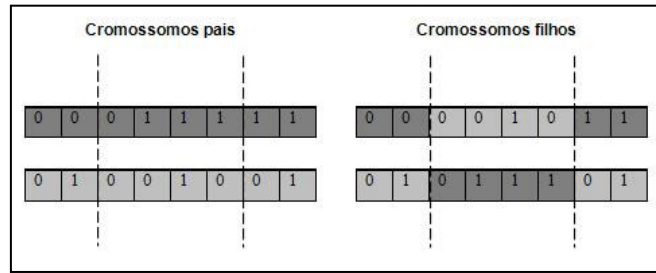
As formas mais comuns de troca de segmentos nos Algoritmos Genéticos são as de ponto único, duplo e cruzamento de pontos aleatórios.

- **Ponto único:** No método de ponto único, é escolhido um ponto de corte aleatório e a partir desse ponto o material genético dos pais é trocado dando origem a dois novos cromossomos, formados pela combinação das características genéticas dos pais, como pode ser observado na figura 4.



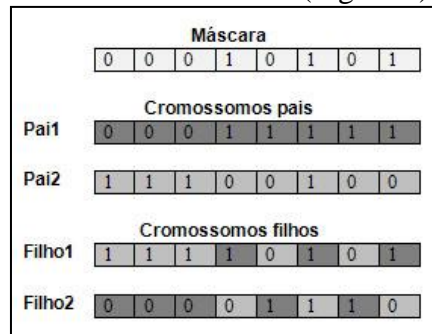
**Figura 4. Esquema de cruzamento de ponto único.**

- **Ponto duplo:** No método de ponto duplo são escolhidos dois pontos de cruzamento e a partir desses os materiais genéticos dos pais são trocados de forma intercalada, como pode ser observado na figura 5.



**Figura 5. Esquema de cruzamento de ponto duplo.**

- Pontos aleatórios: A técnica de cruzamento de pontos aleatórios utiliza como base máscaras, que consistem de cromossomos binários, onde no local em que os alelos possuem o valor 1 utiliza-se, inicialmente, o material genético de um dos pais, e onde possui o valor 0 pega-se o material do outro pai. A máscara pode ser invertida e inicia-se novamente o processo de cruzamento (Figura 6).

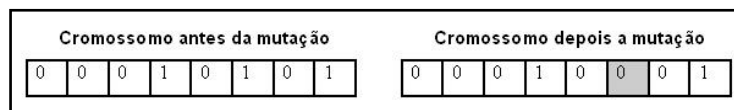


**Figura 6. Esquema de cruzamento de pontos aleatórios.**

### 2.1.5 Mutação

O operador de mutação é executado logo após o processo de cruzamento e tem por objetivo realizar modificações em determinadas propriedades genéticas de uma população, de forma aleatória. Tal operação mostra-se importante uma vez que possibilita à população atual obter propriedades genéticas que não existiam ou eram encontradas em baixa porcentagem. Deste modo, a execução do operador de mutação mostra-se indispensável, visto que este permite a introdução e manutenção da diversidade genética da população. Logo, com a execução do operador de mutação é possível garantir que o espaço de busca possivelmente não será igual a zero. Vale ressaltar, que a taxa de mutação, assim como na natureza, deve acometer uma pequena parcela da população, pois caso a porcentagem seja elevada, os indivíduos gerados pouco se assemelharão aos seus pais; caso contrário, a diversidade da população estará comprometida. Existem várias técnicas de mutação, dentre elas:

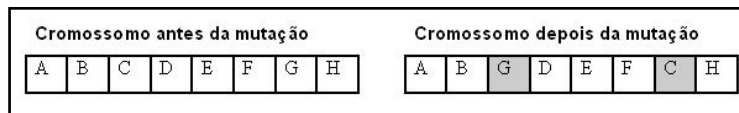
- Mutação aleatória: dentro de um alfabeto válido, um valor é sorteado para substituir o que sofrerá a mutação;



**Figura 7. Exemplo de mutação aleatória.**

- Mutação por troca:  $n$  pares de genes são sorteados; logo após o sorteio os pares trocam valores entre si;





**Figura 8. Exemplo de mutação por troca.**

Após o operador de mutação ser executado, o processo de geração de uma nova população foi então concluído, deste modo, a seção seguinte apresenta como se dá o processo de evolução de uma população.

### 2.1.6 Atualização

Na atualização a população antiga é substituída por uma nova população, formada pelo cruzamento dos indivíduos selecionados da população anterior. As formas mais conhecidas de atualização são  $(x+y)$  e  $(x,y)$ , também chamadas de estratégia soma e estratégia vírgula. Na estratégia soma, indivíduos da população anterior convivem com a população formada por seus filhos. Essa estratégia também pode ser chamada de elitismo e geralmente uma percentagem muito pequena é selecionada para a próxima geração, pois corre-se o risco de uma convergência prematura do AG. Na estratégia vírgula a população anterior não convive com a próxima população, perdendo-se então soluções boas encontradas.

### 2.1.7 Finalização

O operador de finalização é o responsável por determinar se a execução do Algoritmo Genético (evolução de população) será concluída ou não. Tal ação é realizada a partir da execução de testes baseados em uma condição de parada pré-estabelecida. Tal condição de parada pode variar desde a quantidade gerações desenvolvidas até o grau de proximidade dos valores de aptidão de cada cromossomo, de determinada população. (LUCAS, 2000, p. 23).

## 3. Descobrimo a Frase Secreta Utilizando Algoritmo Genético

Para que seja possível compreender melhor o funcionamento e aplicação dos operadores genéticos, explicados anteriormente, este tópico apresenta a resolução de um problema do mundo real, utilizando como base um modelo simples de AG. O problema a ser resolvido consiste em fazer com que o AG descubra uma frase digitada pelo usuário, utilizando como parâmetro apenas a quantidade de caracteres que a compõe.

A partir disto, definiu-se que os indivíduos (possíveis soluções) a serem criados deveriam ser compostos pela mesma quantidade de caracteres da frase informada pelo usuário, ou seja, caso a frase seja composta por 7 (sete) caracteres, então os indivíduos gerados possuirão 7 (sete) genes. Em seguida, para que fosse possível determinar o quão aptos estão os indivíduos gerados, para solucionar o problema, definiu-se que cada um dos cromossomos deveria ser submetido à seguinte função:

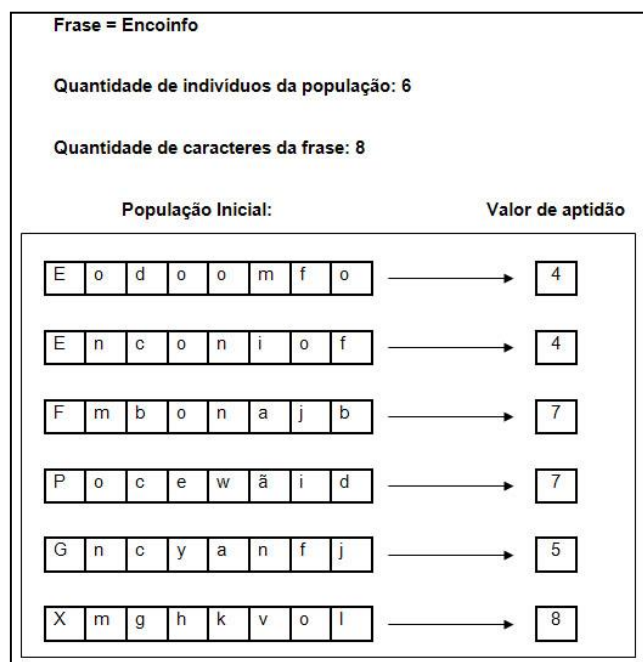
$$f = \sum_{i=1}^N 1 - \text{sinal}(\text{codASCfrase}[i] - \text{codASCcromo}[i]) \rightarrow \text{onde: } \text{sinal}(x) = \begin{cases} 1 & \text{se } x = 0 \\ 0 & \text{se } x \neq 0 \end{cases}$$

**Fórmula 1. Formula da Função de aptidão (Modificado de LARCERDA, 2008, p. 36).**

Para executar tal função é necessário conhecer o código ASCII de cada caractere da frase informada pelo usuário, como também dos caracteres que compõem as frases (indivíduos) geradas pelo AG. Por fim, definiu-se que o AG deveria finalizar a sua execução quando o valor de aptidão encontrado fosse igual a 0 (zero), tal decisão se justifica visto que executada a função de aptidão, caso a frase informada pelo usuário e uma frase gerada pelo AG sejam iguais o valor encontrado será 0 (zero).

Para a implementação de tal AG optou-se pela escolha das seguintes técnicas: para realizar a seleção dos cromossomos pais utilizou-se o método torneio; para efetuar o cruzamento entre os cromossomos selecionados fez-se uso da técnica de pontos aleatórios e, para executar o operador de mutação nos cromossomos gerados, aplicou-se a técnica de mutação aleatória. Além disso, a título de demonstração, vale lembrar que a população a ser criada será composta por 6 (seis) indivíduos e terá a probabilidade de 5% de mutação.

Para que seja possível compreender melhor como são processadas as operações dentro do AG, será realizada uma simulação apresentado passo a passo sua execução. Sendo assim, suponha que o usuário digite a frase Encoinfo. Como pode ser verificado na figura 9, a primeira operação, realizada, pelo AG, é a criação dos cromossomos (possíveis soluções) da população inicial, de forma aleatória. Por conseguinte, cada um dos cromossomos criados é submetido ao processo de avaliação de aptidão, utilizando a função definida anteriormente (Figura 9).



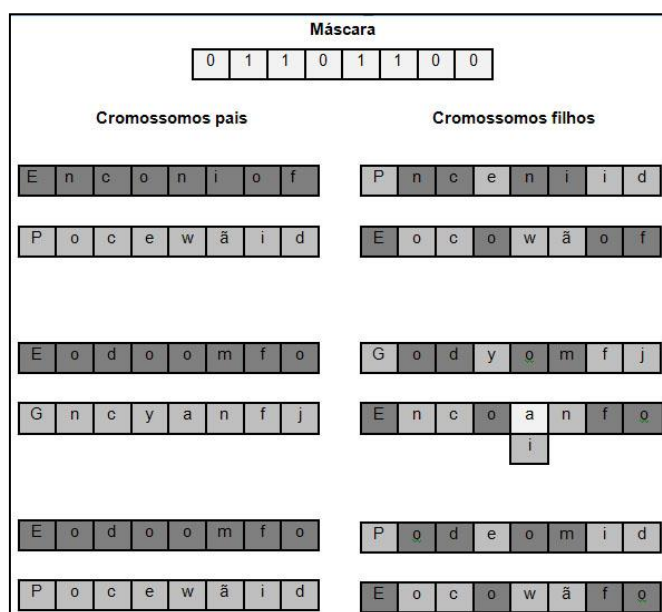
**Figura 9. Resultado dos operadores de inicialização e função de aptidão.**

Logo em seguida, como ilustrado na figura 10, é realizada a seleção, por torneio, dos indivíduos que participarão do processo de cruzamento.

População	Cromossomos candidatos		Selecionados
Eodoomfo	Enconiof, Gncyanfj, Gncyanfj	→	Enconiof
Enconiof	Fmbonajb, Pocewãid, Xmghkvol	→	Pocewãid
Fmbonajb	Eodoomfo, Gncyanfj, Enconiof	→	Eodoomfo
Pocewãid	Gncyanfj, Fmbonajb, Pocewãid	→	Gncyanfj
Gncyanfj	Eodoomfo, Xmghkvol, Gncyanfj	→	Eodoomfo
Xmghkvol	Xmghkvol, Pocewãid, Fmbonajb	→	Pocewãid

**Figura 10. Resultado do operador de seleção por torneio.**

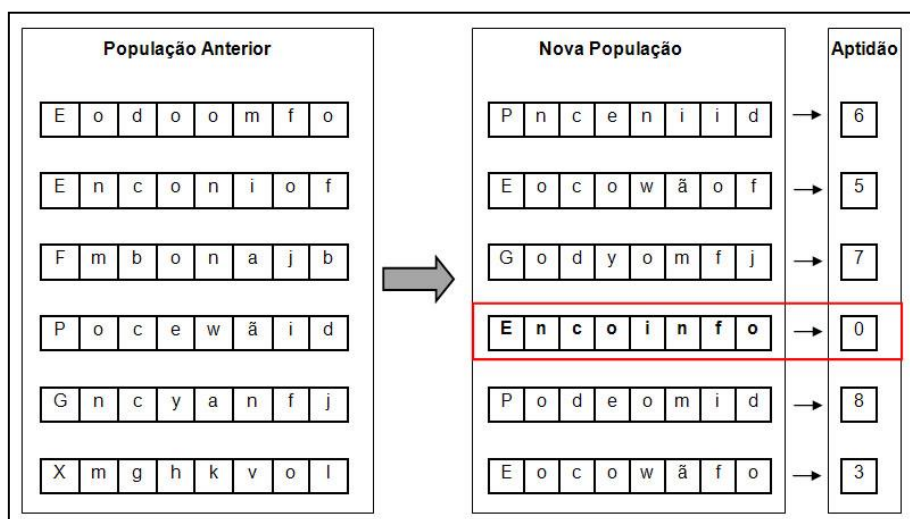
Após selecionar os cromossomos que darão origem à próxima geração, é realizado o cruzamento entre as características dos pares de cromossomos, utilizando para isto a técnica de pontos aleatórios. Assim, como pode ser conferido na figura 11, após o processo de cruzamento são gerados 6 (seis) novos indivíduos.



**Figura 11. Resultado dos operadores de cruzamento e mutação.**

Com a finalização do processo de cruzamento, é executada a operação de mutação, como também pode ser verificado na figura 11, na nova população gerada, ocorreu mutação com no quinto gene do segundo filho gerado pelo segundo par de cromossomos.

Após finalizar a execução do operador de mutação, é gerada uma nova população a qual é apresentada na figura 12, onde é demonstrada a substituição da população inicial pela a que acabará de ser gerada. Como a atualização da população, o AG verifica se a condição de parada especificada, foi alcançada, como pode ser verificado na figura 12 o quarto cromossomo que compõe a nova população possui valor de aptidão igual a 0 (zero), o que faz com que a execução do AG seja finalizada.



**Figura 12. Resultado dos operadores de atualização e finalização.**

De forma geral, ao analisar toda a implementação e resolução do problema proposto, percebe-se que, apesar de ser um modelo simples de AG, este proporciona um entendimento geral no que diz respeito aos operadores genéticos.

#### 4. Conclusões

O principal objetivo deste artigo era apresentar uma visão geral da técnica de Inteligência Artificial, denominada Algoritmo Genético, a fim de expor seus principais conceitos, estrutura e forma de funcionamento, como também demonstrar de forma prática a sua utilização para a resolução de um problema.

A partir do estudo apresentado, é possível verificar que a técnica de AG objetiva simular as operações genéticas da mesma forma como estas se dão na natureza, por meio da criação de uma população de indivíduos, onde cada um destes é reconhecido com um cromossomo, que passa por um processo que imita a seleção, reprodução e mutação, dentre outros, originando uma nova população. Assim, esta técnica mantém uma população de indivíduos, que representam possíveis soluções para um determinado problema do mundo real, armazenando informações a respeito de um determinado espaço de busca e utilizando o conhecimento obtido para realizar novas descobertas, permitindo, desta forma, que problemas sejam solucionados de forma otimizada. Com os testes executados com esta técnica de IA foi possível perceber que mesmo em uma implementação simples resultados satisfatórios são obtidos e que, caso o AG seja corretamente modelado para solucionar um problema específico, é possível obter ótimos resultados.

#### Referências Bibliográficas

(COSTA FILHO e POPPI, 1999) COSTA FILHO, Paulo Augusto da; POPPI, Ronei Jesus. Algoritmos Genéticos em Química. **Química Nova**. Campinas, v. 22, n. 3, jun. 1999. Disponível em: <<http://www.scielo.br/pdf/qn/v22n3/1094.pdf>>. Acesso em: 6 mar. 2007.

(GOLDBERG, 1989 apud TICONA, 2003) Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., Reading, MA. *apud* TICONA, Waldo Gonzalo Cancino. **Aplicação de Algoritmos**

**Genéticos Multi-Objetivo para Alinhamento de Sequências Biológicas.** 2003. 112 p. Dissertação (Mestrado em Ciências Matemáticas e de Computação) – Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo – São Carlos. Disponível em: <[http://www.lania.mx/~ccoello/EMOO/tesis\\_waldo\\_msc\\_final.pdf.gz](http://www.lania.mx/~ccoello/EMOO/tesis_waldo_msc_final.pdf.gz)>. Acesso em: 9 mai. 2007.

(GOLDBERG, 1990 apud LIMA, 2008) GOLDBERG, D. E. **Information Models, Views, and Controllers.** Dr. Dobb's Journal, July 1990. *apud* LIMA, Ednaldo Oliveira. **Algoritmo Genético Aplicado à Otimização de Funções.** 2008. 72 p. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Lavras – Departamento de Ciência da Computação. Disponível em: <[http://www.comp.ufla.br/monografias/ano2008/Algoritmo\\_genetico\\_hibrido\\_aplicado\\_a\\_otimizacao\\_de\\_funcoes.pdf](http://www.comp.ufla.br/monografias/ano2008/Algoritmo_genetico_hibrido_aplicado_a_otimizacao_de_funcoes.pdf)>. Acesso em: 1 out. 2008.

(GOLDBERG e HOLLAND, 1988 apud BITTENCOURT, 2008) D.E. Goldberg and J.H. Holland. Genetic algorithms and machine learning: Introduction to the special issue on genetic algorithms. *Machine Learning*, 3, 1988. *apud* BITTENCOURT, Guilherme. **Inteligência Computacional.** 2008. Disponível em: <<http://www.das.ufsc.br/gia/softcomp/node11.html>>. Acesso em: 1 out. 2008. (Apostila).

(LACERDA, 2008) LACERDA, Estéfane G. M. **Algoritmos Genéticos: Aspectos Práticos.** 2008. 36 p.. Disponível em: <[http://www.dca.ufrn.br/~estefane/metaheuristicas/ag\\_pratica.pdf](http://www.dca.ufrn.br/~estefane/metaheuristicas/ag_pratica.pdf)>. Acesso em: 01 out. 2008. (Apostila).

(LIMA, 2008) LIMA, Ednaldo Oliveira. **Algoritmo Genético Aplicado à Otimização de Funções.** 2008. 72 p. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Lavras – Departamento de Ciência da Computação. Disponível em: <[http://www.comp.ufla.br/monografias/ano2008/Algoritmo\\_genetico\\_hibrido\\_aplicado\\_a\\_otimizacao\\_de\\_funcoes.pdf](http://www.comp.ufla.br/monografias/ano2008/Algoritmo_genetico_hibrido_aplicado_a_otimizacao_de_funcoes.pdf)>. Acesso em: 1 out. 2008.

(LINDEN, 2006) LINDEN, Ricardo. **Algoritmos Genéticos: Uma importante ferramenta da inteligência computacional.** Rio de Janeiro: Brasport, 2006.

(LUCAS, 2000) LUCAS, Diogo Correa. **Algoritmos Genéticos: Um estudo dos seus conceitos fundamentais e aplicação no problema da grade horária.** 2000. 66 p. Monografia (Graduação em Informática) – Universidade Federal de Pelotas – Instituto de Física e Matemática. Disponível em: <<http://www.ufpel.tche.br/prg/sisbi/bibct/acervo/info/2000/Mono-Diogo.pdf>>. Acesso em: 4 abr. 2007.

(ROSA, 2007) ROSA, Thatiane de Oliveira. **Aplicação de Algoritmos Genéticos para Comparação de Sequências Biológicas.** 2007. 130 p. Monografia (Graduação em Sistemas de Informação) – Centro Universitário Luterano de Palmas, Palmas - TO.

(RUSSELL e NORVIG, 2004) RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial.** 2. ed. Rio de Janeiro: Editora Campus, 2004. 1020 p.

# Desenvolvimento de aplicações web com JSP

Leandro Soares de Sousa, Paulo Henrique de Sousa

Sistemas de Informação – Centro Universitário Luterano de Palmas (CEULP/ULBRA)

Cx. Postal 160 – 77054-970 – Palmas – TO – Brasil

{leandroleo2, paulohenrique.sous4}@gmail.com

*Resumo: O desenvolvimento de aplicações web tornou-se dinâmico e personalizado com a evolução da internet. Diversas tecnologias influenciaram nesta revolução. Dentre estas se destaca o JSP (Java Server Pages), pois todas as vantagens como portabilidade, independência de plataforma e programação orientada a objetos foram herdadas da linguagem Java. Isso tornou o JSP uma linguagem bastante difundida na construção de Web Sites.*

## 1. Introdução

Desde a criação da web a forma de disposição das informações foi sendo constantemente modificadas. No início, as informações era apresentadas de maneira estática, ou seja, sem modificações em tempo de execução. Atualmente os dados são manipulados tornando possível a personalização de apresentação da informação. No presente trabalho serão apresentados conceitos de construção de aplicativos web com a utilização de JSP. No mercado existem diversas ferramentas capazes de fornecer o ambiente para o desenvolvimento de aplicações com JSP. Para o mini-curso (que é o objetivo deste trabalho) será utilizada a *IDE* Netbeans, pois está dispõe de um ambiente completo no que tange o desenvolvimento de aplicações web com JSP. Este trabalho está organizado da seguinte forma: na seção 2 será apresentada a estrutura cliente/servidor, na seção 3 demonstra o processo de interpretação de paginas JSP, na seção seguinte (4) elucida a conexão a uma base de dados, por fim, na seção 5 é demonstrada a conclusão.

## 2. Arquitetura de aplicações web

A arquitetura em que o JSP foi concebido (cliente/servidor) permite que as páginas desenvolvidas troquem informações dentro de uma rede. Essa arquitetura é usualmente utilizada em contextos onde o servidor aguarda requisições, processa as solicitações enviadas e devolve o resultado deste processamento. A aplicação cliente é responsável por fazer chamadas a eventos do servidor além de receber e apresentar o resultado da solicitação (Duchessi & Chehgalur-Smith, 1998). A figura a seguir exemplifica o modelo de comunicação.



**Figura 1. Modelo de arquitetura cliente/servidor.**

Na figura anterior é possível observar que o funcionamento se dá pela solicitação ao servidor pelo navegador cliente via HTTP. O servidor por sua vez processa a solicitação e devolve o resultado pelo mesmo protocolo ao cliente.

### **3. Interpretação e compilação de páginas JSP**

JSP é uma tecnologia voltada para o desenvolvimento de web sites por meio da utilização da linguagem Java (JCP, 2003). Como consequência as aplicações desenvolvidas em JSP herdam todas as vantagens de utilização da referida linguagem: orientação a objetos, multiplataforma e gerenciamento de memória. Os documentos desenvolvidos em JSP apresentam o resultado do processamento do servidor mediante alguma solicitação. Quando uma página JSP é solicitada pela primeira vez, um *Servlet* correspondente a página é acionado, após sua execução/compilação uma página HTML é gerada e enviada para o navegador do cliente responsável pela requisição. A partir da segunda requisição o processo de compilação não será mais necessário trazendo um ganho no tempo de resposta ao cliente. A figura a seguir demonstra um exemplo simples de uma página JSP.

```

7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
9 "http://www.w3.org/TR/html4/loose.dtd">
10 <%@ page import="java.util.Calendar" %> %>
11 <html>
12 <head>
13 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14 <title>JSP Page</title>
15 </head>
16 <body>
17 <h2>ENCOINFO</h2>
18 <%
19     String[] diasSemana = {"Domingo", "Segunda-Feira", "Terça-Feira",
20     "Quarta-Feira", "Quinta-Feira", "Sexta-Feira", "Sábado"};
21
22     String[] meses = {"Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho",
23     "Julho", "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro"};
24
25     Calendar calendar = Calendar.getInstance();
26
27     String str = "Bom dia! " + diasSemana[calendar.get(Calendar.DAY_OF_WEEK)-1]
28     + ", " + calendar.get(Calendar.DAY_OF_MONTH) + " de "
29     + meses[calendar.get(Calendar.MONTH)] + " de "
30     + calendar.get(Calendar.YEAR) + " - "
31     + calendar.get(Calendar.HOUR_OF_DAY) + ":" + calendar.get(Calendar.MINUTE);
32     out.println(str);
33 %>
34 </body>
35 </html>

```

**Figura 2. Exemplo de uma página JSP.**

A figura acima apresenta uma estrutura básica de formação de um documento JSP. Nota-se a disposição de conteúdo estático, mas principalmente de conteúdo que é passível de processamento no servidor. Este conteúdo processável pelo servidor é delimitado pelos símbolos (*tags*) de “<%” e “%>”. Ao ser solicitado o conteúdo processado no servidor é transformado em HTML ocultando o código originário.

Em algumas situações deseja-se proteger determinada parte da informação, pois nem todos os usuários podem efetuar alterações ou até mesmo visualizar determinado conteúdo. Diante de tal situação, a tecnologia JSP dispõe de objetos implícitos responsáveis por auxiliar no controle de acesso a determinados arquivos. O principal objeto para tal controle é chamado de “*session*”. Dentre as suas características destacam-se (Nickk & Mark, 2003):

- Armazenar informações de data e hora do ultimo acesso para uma sessão particular;
- Permitir à adição de outros objetos a sessão para utilização posteriormente.

Com a utilização de tais características o controle das informações de acesso pode ser facilmente aplicado a todo o sistema, garantindo a integridade de informações particulares a determinado contexto.



#### 4. Acessando uma fonte de dados

Quando se necessita acessar qualquer base de dados é necessária a utilização de APIs JDBC (*Java Database Connectivity*). As classes JDBC's são compostas de métodos para interação direta com as bases de dados como: inserção, alteração, exclusão e consulta de dados. Diversas são as APIs encontradas no mercado. Para cada base de dados existente no mercado existe um drive JDBC correspondente.

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/minicurso", "root", "senha");
} catch (SQLException ex) {
    ex.printStackTrace();
}
```

**Figura 3. Modelo de conexão com base de dados**

A figura 3 apresenta os passos para estabelecimento de uma conexão com o banco de dados *MySQL*. Na primeira linha é recuperado o drive para a base de dados que será acessada, logo após, é estabelecida a conexão com a base por meio da instanciação da classe “*Connection*” através do método “*getConnection*” da classe “*DriverManager*”. Na instanciação da classe *Connection* é informado o tipo da conexão, caminho da base de dados, usuário e senha para o acesso a fonte de dados. A partir de então operações como abertura de conexão, inserção e deleção de dados tornam-se possíveis.

#### 5. Conclusão

Com a evolução da web e a competitividade cada vez maior no mercado de desenvolvimento de sistemas torna-se fundamental a utilização de maneiras dinâmicas e praticas de disponibilizar a informação. O presente trabalho teve como objetivo explicar brevemente sobre a tecnologia JSP e alguns de seus recursos, apresentado-a como uma opção relevante para aplicações web personalizadas. Isso se deve em parte pelas características herdadas da linguagem Java e pelo simplificado processo de interpretação e compilação de seus arquivos. Em meio às várias opções de tecnologias voltadas a construção de web sites o JSP torna-se bastante para o desenvolvimento de soluções multiplataformas e com as qualidades advindas de uma linguagem em plena expansão como é o caso do Java.

## Referências Bibliográficas

Duchessi, P. & Chengalur-Smith, I. (1998). Client/Server benefits, problems, Best practices. *Communications of the ACM*, 41(5). 87-94.

JCP (2003). JCP – JAVASERVER PAGES SPECIFICATION – JAVA COMMUNITY PROCESS, JSR – 000152 *JavaServer Pages™ 2.0 Specifications*. Disponível em: <<http://jcp.org/aboutJava/communityprocess/first/jsr152/index3.html>> Acessado em: Outubro de 2006;

JCP (2003b). SERVLET SPECIFICATION – JAVA COMMUNITY PROCESS, JSR – 000154 *Java™ Servlet 2.4 Specifications*. Disponível em: <<http://jcp.org/aboutJazva/communityprocess/final/jsr154/index.html>> Acessado em: Outubro de 2006;

T., Nick e S., Mark. (2003). *JavaServer Pages: o guia de desenvolvimento* / Tood, Nick e Szolkowski, Rio de Janeiro: Elsevier, 2003. ISBN 85-352-1324-4

# Desenvolvimento de Aplicações Web com PHP

Felipe Eduardo Bechert Schmitz, Valdirene da Cruz Neves Júnior

Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas  
(CEULP/ULBRA)

Caixa Postal 160 – 77.054.970 – Palmas – TO – Brasil

felipeebs@gmail.com, valdirene@inf.ceulp.edu.br

**Resumo:** *O desenvolvimento de aplicações para a web está cada vez mais fácil com a evolução da internet. Há poucos anos se falava em internet 2.0 como uma coisa de outro mundo, mas as ferramentas de desenvolvimento evoluíram tanto que hoje é possível se fazer um website interativo sem grandes dificuldades. Ao longo deste mini-curso desenvolveremos uma aplicação web simples utilizando a linguagem de programação para web PHP.*

## 1. Introdução

O PHP é a linguagem de programação voltada para a web mais antiga que existe. Sua estrutura simples e sofisticada com suporte a orientação a objetos, torna essa linguagem uma das principais ferramentas em soluções web.

Para desenvolver um sistema em PHP, precisa-se apenas de um editor de texto simples (existem diversas IDEs para trabalhar com PHP) e um servidor web que consiga interpretá-lo. Caso a aplicação desejada utilize banco de dados, será preciso uma ferramenta de sistema de gerenciamento de banco de dados.

Para este mini-curso utilizaremos o pacote XAMPP Lite, que vem com um servidor web Apache e o SGBD MySQL. Apresentaremos também um pouco da história do PHP, assim como o comportamento do PHP no servidor.

## 2. O PHP

Criado por Rasmus Lerdorf, o PHP consistia apenas em um conjunto de macros para enviar declarações SQL para banco de dados, processar formulários e controlar fluxo. Inicialmente o PHP foi desenvolvido sobre a linguagem Perl e se chamava PHP/FI (Personal Home Page / Form Interpreter).

Essas ferramentas foram escritas em C e nomeadas como PHP/FI 2.0, que teve seu código aberto, assim outras pessoas poderiam colaborar e corrigir bugs. Essa versão

não era tão diferente da sintaxe que conhecemos hoje, havia suporte para variáveis, matrizes associativas e funções.

Hoje os principais arquitetos do PHP são Zeev Suraski e Andi Gutmans, que reescreveram o PHP/FI 2.0 do zero e junto com Rasmus lançaram a versão 3, essa chamada simplesmente de PHP (PHP Hypertext Processor). Zeev e Andi criaram o mecanismo Zend (o nome deriva de **Zeev** e **Andi**) e com alguns ajustes foi lançada a versão 4 do PHP, essa já tinha suporte a orientação a objetos, mas não como seu foco principal.

“O PHP 5 representa, explicitamente, o endosso aos objetos e a programação orientada ao objetos” (Matt Zandstra). O PHP continua sendo uma linguagem que suporta o desenvolvimento orientado a objetos, porém, seu suporte a objetos está agora bem desenvolvido.

### **3. Servidor Web**

Um servidor é um sistema de computação capaz de fornecer serviços a outros computadores conectados a ele. Esse tipo de serviço é chamado de client-server (cliente-servidor), no qual um cliente pode fazer uma requisição ao servidor e o servidor retorna uma resposta ao cliente.

No PHP o servidor mais comum é o Apache Server, conhecido por ser um bem-sucedido servidor web, além de ser de utilização livre e ter seu código aberto, ou seja, sem custo de utilização. A compatibilidade do Apache com protocolo HTTP (HyperText Transfer Protocol / Protocolo de Transferência de Hipertexto) e a disponibilização em versões para vários sistemas operacionais, fizeram com que ele se tornasse atualmente um dos sistemas de servidores mais utilizados.

### **4. Sistema de Gerenciamento de Banco de Dados**

Os Sistemas de Gerenciamento de Bancos de Dados (SGBD) são responsáveis por manipular informações contidas em um banco de dados. O PHP tem compatibilidade com diversos SGBDs, como por exemplo, Oracle, MySQL, SQLite, MSSQL, PostgreSQL, entre outros.

Apesar de encontrarmos facilmente sistemas desenvolvidos em PHP utilizando alguns destes SGBDs, o mais comum entre eles é o MySQL. Esse sucesso deve-se a fácil integração com o PHP, incluindo quase que obrigatoriamente nos pacotes de hospedagem de sites.

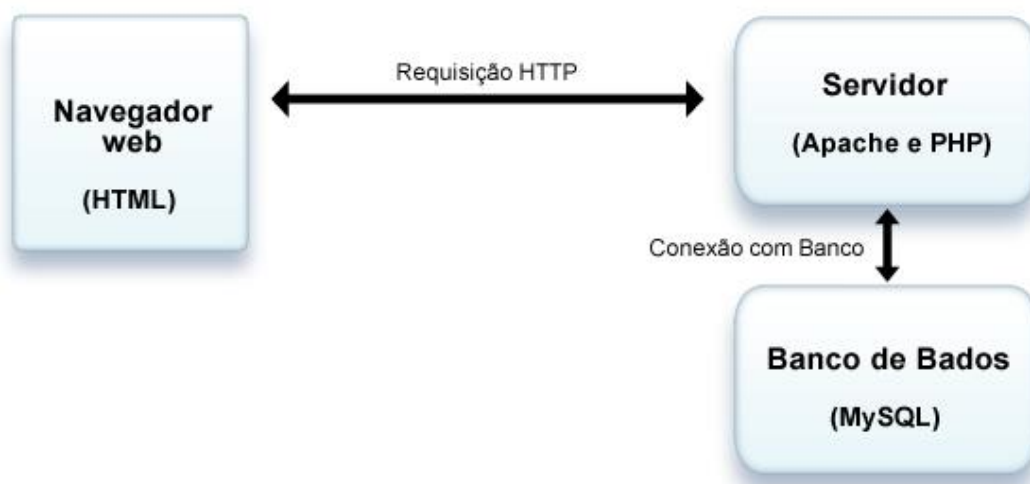
## 5. XAMPP Lite

O XAMPP é dos mais conhecidos pacotes AMP (Apache, MySQL e PHP) da atualidade. Seu nome é o acrônimo onde X (roda em vários sistemas operacionais), Apache, MySQL, PHP e Perl, já o Lite é a versão portátil do sistema.

Seu princípio era uma ferramenta para desenvolvimento local e teste, mas hoje é uma das ferramentas preferidas para servidor web por ser simples de instalar, segura e rodar em várias plataformas. O programa também é liberado sob licença livre.

## 6. Interpretação de páginas PHP

Podemos definir o PHP como uma linguagem de criação de scripts embutida em HTML no servidor. O resultado de um programa escrito em PHP é obtido através de um navegador web. Veja o exemplo na figura 1.



**Figura 1: Exemplo de serviço cliente-servidor**

O interpretador PHP ignora todo texto que esteja fora do limitador (normalmente as tags `<?php ... ?>`) e processa somente o código entre elas. O código pode ter várias tags PHP que são interpretadas gerando HTML que são enviadas ao navegador e apresentadas ao usuário.

As páginas com códigos PHP devem ter a extensão ".php", isso é obrigatório para que o servidor possa identificar qual o tipo de script contem a página, mas dentro de um arquivo PHP podemos usar tags HTML comum. A figura 2 representa um exemplo de utilização de código PHP com HTML.

```

1  <?php require "Classes/Exemplo.php"; ?>
2  <html>
3  <head>
4      <title>ENCOINFO XI - Mini-curso PHP</title>
5      <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
6      <meta name="description" content="" />
7      <meta name="keywords" content="" />
8      <meta name="author" content="Valdirene Neves Júnior" />
9      <meta name="language" content="pt-br" />
10 </head>
11 <body>
12 <h1><?php echo $Exemplo->getTitulo() ?></h1>
13 <p>
14     <?php if($Exemplo->getHora() >= 6 && $Exemplo->getHora() < 12): ?>
15         Bom dia <?php echo $Exemplo->getNome() ?>!<br />
16     <?php elseif($Exemplo->getHora() >= 12 && $Exemplo->getHora() < 18): ?>
17         Boa tarde <?php echo $Exemplo->getNome() ?>!<br />
18     <?php else: ?>
19         Boa noite <?php echo $Exemplo->getNome() ?>!<br />
20     <?php endif ?>
21     Seja bem-vindo ao mini-curso de PHP na 11ª edição do ENCOINFO.
22 </p>
23 </body>
24 </html>

```

Figura 2: Exemplo de código PHP dentro do HTML

## 7. Conclusões

Com a constante evolução e difusão da web, torna-se fundamental a utilização de linguagens dinâmicas para o desenvolvimento de aplicações para ela. O objetivo deste trabalho foi apresentar brevemente as funções principais da plataforma PHP com o desenvolvimento de uma aplicação útil de forma didática. Em meio à várias tecnologias para desenvolvimento de web sites, o PHP se destaca pela aplicabilidade, robustez de recursos e facilidade de aprendizado.

## 8. Referências Bibliográficas

ZANDSTRA, Matt. **Entendendo e Dominando o PHP**. São Paulo: Digerati Books, 2006

RAMOS, Ricardo; SILVA, Joel; ÁLVARO, Alexandre; AFONSO, Ricardo. **PHP para Profissionais**. São Paulo: Digerati Books, 2007.

# Django, Desenvolvimento Ágil para a Web

Vinicius Prado da Fonseca, Felipe Mendes Braga

Curso de Ciência da Computação – Universidade Federal do Tocantins (UFT)  
Caixa Postal 114 – 77001-090 – Palmas – TO – Brasil

{vinicius.prado,felipe.mendes}@uft.edu.br

***Resumo.** Django é um framework para desenvolvimento de aplicações web escrito em Python, que herda dessa linguagem de altíssimo nível principalmente a agilidade no desenvolvimento de soluções e a facilidade na manutenção do código.*

## 1. Introdução

Com o crescimento da *web* e o surgimento de novos modelos de aplicações, muitas delas de grande porte, os desenvolvedores detectaram a necessidade da utilização de *frameworks* que facilitassem o desenvolvimento da estrutura básica de projetos de software e proporcionassem uma melhor padronização dos mesmos. Nesse sentido o MVC – *Model-View-Controller* – se mostrou uma alternativa viável dentre os padrões já existentes, pelo fato de separar a lógica de negócio da apresentação [1]. O *framework Django* consegue reunir características decisivas para seu sucesso nessa nova realidade do desenvolvimento web, principalmente pela utilização do MVC como padrão arquitetural e pela fácil divisão de responsabilidades do projeto, foco central nas equipes de desenvolvimento, cada vez mais bem definidas e desacopladas.

## 2. Objetivo

Apresentar o *framework Django* focando as características principais do padrão de projetos MVC, ressaltando as vantagens e desvantagens de sua utilização, bem como sua aplicação no desenvolvimento Web.

## 3. Framework Django

O *Django* é um *framework* voltado para o desenvolvimento *web*, escrito em *python*, utilizando o padrão de projetos MVC [2]. Para a camada de modelo, o *Django* utiliza o ORM (Mapeamento Objeto-Relacional) que, basicamente, transforma objetos em estados num banco de dados. A camada de controle é comumente implementada utilizando-se da estrutura *views.py*, que é responsável pelas regras de negócio. Na camada de visão é utilizada a própria estrutura *views.py*, à respeito da renderização dos *templates*, *urls.py* para a configuração das *urls* e os *templates* propriamente ditos. Juntas essas peças implementam de forma simples o MVC [3].

## 4. Desenvolvimento Ágil e MVC

O MVC necessita de um tempo maior de análise e modelagem do sistema e não é aconselhável a sua implementação em pequenas aplicações, contudo o desenvolvimento de modelo, visão e controle pode ocorrer paralelamente, pois são independentes, agilizando o processo de desenvolvimento. Utilizar a mesma camada de modelo para vários visualizadores torna fácil manter, testar e atualizar sistemas múltiplos [4].

## 5. Desenvolvimento Ágil e Django

O Django, como arcabouço para o desenvolvimento de aplicações web, mostra-se vantajoso em relação ao desacoplamento de aplicações do projeto, à pequena quantidade de código que será escrita, ao rápido desenvolvimento de aplicações, à filosofia DRY (“*Don't Repeat Yourself*”) [3], que por utilizar uma linguagem totalmente orientada a objetos, potencializa o reuso do código e facilita sua manutenção.

## 6. Referências

- [1]Freeman, Eric e Freeman, Elisabeth. “Use a cabeça! Padrões de Projeto (Design Patterns)”. Alta Books, 2005.
- [2]Moore, D.; Budd R. e Wright W. “Professional Python® Frameworks Web 2.0 Programming with Django® and TurboGears™”. Wiley Publishing, Inc, 2007.
- [3]Holovaty, A. e Kaplan-Moss, J. “The Django Book”. Disponível em: <http://www.djangobook.com/>. Acessado em setembro de 2009.
- [4]Macoratti, J. C. “Padrões de Projeto: O modelo MVC - Model View Controller”. Disponível em: [http://www.macoratti.net/vbn\\_mvc.htm](http://www.macoratti.net/vbn_mvc.htm). Acessado em setembro de 2009.



# Adobe Flash 8

Gedilson Pessoa da Silva<sup>1</sup>

<sup>1</sup>Sistema de Informação – Centro Universitário Luterano de Palmas (CEULP-ULBRA)  
Caixa Postal 160 – 77.054-970 – Palmas – TO – Brasil

{ gedilsonpessoa }@gmail.com

***Resumo.** A tecnologia Flash (Shockwave), empregada pela a Adobe, vem adicionando uma parcela significativa aos Layouts dos sites. Devido ao poder de processamento multimídia, o Flash é utilizada por desenvolvedores na criação de banners, jogos e animações diversas que podem ou não interagir com os visitantes. Uma das suas características é o tamanho considerável do arquivo gerado, facilitando assim, a transferência pela a Internet.*

## 1. Introdução

Com a constante evolução da tecnologia e a necessidade cada vez maior de interação entre clientes, empresas de diversos ramos de atividades estão investindo cada vez mais em novas formas de apresentar seus produtos. Para isso o conceito funcional de interface rica RIA - *Rich Internet Applications* vem chamando a atenção do público adquirindo um lugar significativo nos sites destas empresas.

A tecnologia Flash (Shockwave), empregada pela a Adobe, vem adicionando uma parcela significativa aos Layouts dessas empresas. Devido ao poder de processamento multimídia, o Flash é utilizada por desenvolvedores na criação de banners, jogos e animações diversas que podem ou não interagir com os visitantes. Uma das suas características é o tamanho considerável do arquivo gerado, facilitando assim, a transferência pela a Internet. O plugin do Flash deve ser previamente instalado nas estações, para que os arquivos gerados possam ser reconhecidos e executados nos navegadores.

Shockwave é uma ferramenta baseada em imagens vetoriais permitindo que sons, imagens e animações sejam apresentadas nos navegadores. As imagens não são geradas por meio das combinações de pontos e sim com uso de fórmulas matemáticas. Desta forma os arquivos que contem um determinado desenho de tamanho (100x100), não sofrerão modificações no seu tamanho, caso o desenho seja redimensionado para (200x200). Além disso, os arquivos terão tamanhos relativamente pequenos, permitindo assim facilidades para seu tráfego na Internet.

“Como filmes, documentos do Flash dividem comprimentos de tempo em frames. Na timeline, você trabalha com estes frames para organizar e controlar o conteúdo do seu documento. Você insere frames na timeline quando você quiser que seus objetos nos frames apareçam no conteúdo finalizado” (ADOBEFLASH, 2008).

No decorrer do artigo será criado um jogo simples no Adobe Flash 8. Para conhecimento da aplicação serão utilizados: símbolos, biblioteca, linha do tempo e ActionScript.

## 2. ActionScript 2.0

“ActionScript é a linguagem usada para adicionar interatividade à aplicações Flash, onde poderão ser desenvolvidos simples ou ricas animações para a Internet. Não é preciso usar ActionScript para usar Flash, mas se pretende prestar serviços básicos ou complexos, usando interatividade com objetos diferentes daqueles incorporados a ferramenta (tais como botões e filmes), ou de outro modo, transformar um arquivo SWF em uma experiência mais robusta, provavelmente terá que usar ActionScript” (ADOBEFLASH, 2008).

## 3. Conhecendo o Flash 8

A figura 1.1 representa a janela do Flash com a barra de menu na parte superior, do lado esquerdo temos a barra de **ferramentas** (Tools) e a linha do tempo (Timeline) abaixo da barra de menu. Janelas flutuantes, como: biblioteca (Library), Cor (Color) entre outras poderão ser inserido em cena a qualquer momento através de seus respectivos atalhos ou pelo o acesso através da barra de menu.

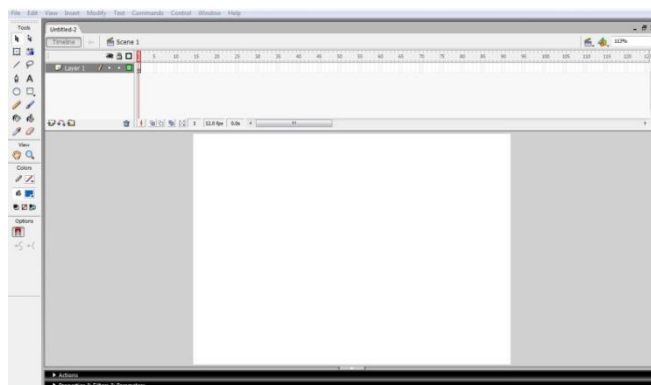


Figura 1.1

## 4. Projeto

Como proposta, temos a criação do jogo de quebra cabeça, para utilização dos recursos do Flash. A figura 1.2 ilustra o jogo a ser criado.

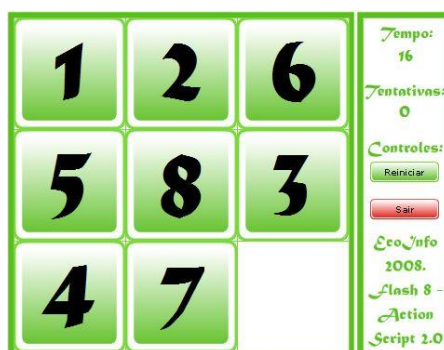


Figura 1.2

## 5. Iniciando o projeto

Crie um novo projeto, clicando em File – New, e selecione em seguida - Flash File (ActionScript 2.0), confirmando a operação. Redimensione o tamanho do Filme para 485x380, pressionando a tecla CTRL+J, informando os respectivos valores e clique em OK. Utilizando as ferramentas do Flash existente no Tools, desenhe a peça representada pela a figura 1.3, duplicando quantas vezes necessárias.



Figura 1.3

A figura 1.3 é um MovieClip formado por um Botão com seus referentes estados e um Text Tool com os respectivos valores: 1, 2, 3, 4, 5, 6, 7, 8. Todos os <Instance Name> dos MovieClips que seguem a estrutura da figura 1.3 deverão ser nomeados para mov1, mov2 ... mov8. Os botões pertencentes a estes MovieClips deverão ter seus <Instance Name> modificados para btn. Após a criação de todas as peças, deverão ser posicionadas em cena, como mostra a figura 1.2.

Crie uma nova camada, nomeando-a para Action e no primeiro Frame adicione o código a seguir.

```
stop();
```

```

tempo = 0;
inicio = 0;

objetivo = [[1, 2, 3],
            [4, 5, 6],
            [7, 8, 0]];
area = [ [1, -1, -1],
         [-1, -1, -1],
         [-1, -1, 0]];

deslocamento = 120;

this.mov1.btn.onPress = function()
{
    acaoBotao(mov1);
}
this.mov2.btn.onPress = function()
{
    acaoBotao(mov2);
}
this.mov3.btn.onPress = function()
{
    acaoBotao(mov3);
}
this.mov4.btn.onPress = function()
{
    acaoBotao(mov4);
}
this.mov5.btn.onPress = function()
{
    acaoBotao(mov5);
}
this.mov6.btn.onPress = function()
{
    acaoBotao(mov6);
}
this.mov7.btn.onPress = function()
{
    acaoBotao(mov7);
}
this.mov8.btn.onPress = function()
{
    acaoBotao(mov8);
}

posicionarPerca();

```

Adicione mais uma camada e nomeia para Functions, inserindo no primeiro Frame o código.

```

//Atualiza os valores da matriz de acordo com a posição dos
botões em cena.
function atualizaArea(l, c)
{
    area[l][c] = 0;
}

//Ação Click dos botões para verificar o destino quando forem
acionados.
function acaoBotao(mcBtn)
{
    l = -1;
    c = -1;
    if (mcBtn.linha >= 0 and (mcBtn.linha <= 2))
    {
        if ((mcBtn.linha + 1) <= 2 && _root.area[mcBtn.linha
+ 1][mcBtn.coluna] == 0)
        {
            mcBtn._y += _root.deslocamento;
            l = mcBtn.linha;
            c = mcBtn.coluna;
            mcBtn.linha += 1;
            _root.area[mcBtn.linha][mcBtn.coluna] =
parseInt(mcBtn.valor.text);
        }
        if ((mcBtn.linha - 1) >= 0 && _root.area[mcBtn.linha
- 1][mcBtn.coluna] == 0)
        {
            mcBtn._y -= _root.deslocamento;
            l = mcBtn.linha;
            c = mcBtn.coluna;
            mcBtn.linha -= 1;
            _root.area[mcBtn.linha][mcBtn.coluna] =
parseInt(mcBtn.valor.text);
        }
        if (l != -1 && c != -1)
            _root.atualizaArea(l, c);
    }
    if (mcBtn.coluna >= 0 and (mcBtn.coluna <= 2))
    {
        if ((mcBtn.coluna + 1) <= 2 &&
_root.area[mcBtn.linha][mcBtn.coluna + 1] == 0)
        {
            mcBtn._x += _root.deslocamento;
            l = mcBtn.linha;
            c = mcBtn.coluna;
            mcBtn.coluna += 1;
            _root.area[mcBtn.linha][mcBtn.coluna] =
parseInt(mcBtn.valor.text);
        }
        if ((mcBtn.coluna - 1) >= 0 &&
_root.area[mcBtn.linha][mcBtn.coluna - 1] == 0)
        {

```

```

        mcBtn._x -= _root.deslocamento;
        l = mcBtn.linha;
        c = mcBtn.coluna;
        mcBtn.coluna -= 1;
        _root.area[mcBtn.linha][mcBtn.coluna]
parseInt(mcBtn.valor.text);
    }
    if (l != -1 && c != -1)
        _root.atualizaArea(l, c);
}
_root.verificaStatusJogo();
if (l != -1 && c != -1)
    somaClique();
}

//Posiciona as perca de forma aleatória na cena.
posicao = [];
function posicionarPerca()
{
    btnCollection = [mov1, mov2, mov3, mov4, mov5, mov6, mov7,
mov8];
    array = [1, 2, 3, 4, 5, 6, 7, 8];
    if (posicao.length == 0)
    {
        for (i = 0; i < 8; i++)
            posicao.push([btnCollection[i]._x,
btnCollection[i]._y]);
    }

    pos = 0;
    for(l = 0; l < 3; l++)
    {
        max = l != 2 ? 3 : 2
        for(c = 0; c < max; c++)
        {
            indice = array.splice(random(array.length), 1);
            _root["mov" + indice].linha = l;
            _root["mov" + indice].coluna = c;
            _root["mov" + indice]._x = posicao[pos][0];
            _root["mov" + indice]._y = posicao[pos][1];
            _root.area[l][c] = indice;
            pos++;
        }
    }
    _root.area[2][2] = 0;
    this.txtClique.text = 0;
    inicio = getTimer()/1000;
}

//Verifica a cada clique se a matriz area esta igual à matriz
objetivo.
function verificaStatusJogo()
{

```

```

verifica = true;
for(i = 0; i < 3 && verifica; i++)
{
    for(j = 0; j < 3 && verifica; j++)
    {
        if (objetivo[i][j] != area[i][j])
        {
            verifica = false;
        }
    }
}
if(verifica)
{
    gotoAndStop(2);
}
}
function somaClique()
{
    this.txtClique.text = parseInt(this.txtClique.text) + 1;
}

```

Dentro de cada peça (botão), adicione uma nova camada, com o nome Action e no primeiro frame declare as variáveis.

```

linha;
coluna;

```

Adicione dois campos Text Tool na cena, nomeando-os para txtTempo e txtClique, onde o primeiro será o cronometro e o segundo o contador de jogadas efetuadas. Converta txtTempo para um MovieClip e mude o <Instance Name> para mc\_cronometro. Posicione os Text Tool como na figura 1.2 e insira no mc\_cronometro o seguinte código:

```

onClipEvent (enterFrame)
{
    _root.tempo = Math.round(getTimer()/1000 - _root.inicio);
    this.txtTempo.text = parseInt(_root.tempo);
}

```

Coloque um botão em cena para reiniciar o jogo quando necessário, e insira o código.

```

on (press) {
    posicionarPerca();
}

```

No botão sair informe o código.

```
on (press) {  
    fscommand("quit", true)  
}
```

Para testar se tudo esta funcionando como esperado, pressione CTRL + ENTER. Revise os passos caso exista erros.

## 6. Conclusão

A tecnologia Flash cresce a cada dia e sites usam-na, para a criação de inúmeras interações e efeitos visuais cada vez mais ricos. Esta tecnologia não está limitada somente às criações de jogos, é possível desenvolver sites completos, cheios de interatividade e ricos em detalhes. No artigo tivemos apenas uma simples, porém funcional, visão da ferramenta e podemos ver que os recursos são fáceis de serem entendidos e usados.

## 7. Referências Bibliográficas

(ADOBEFLASH, 2008) <Adobe Flash CS 4 Professional>. Documentação do produto. Disponível online em: [http://help.adobe.com/en\\_US/Flash/10.0\\_Welcome](http://help.adobe.com/en_US/Flash/10.0_Welcome). Último acesso em: 22/10/2008.



# Introdução ao XML

**Charles Alex Rockenbach, Lucas Moreno de Araujo, Moisés Lima dos Anjos,  
Roneylson de Alencar Neiva**

Sistemas de Informação - Centro Universitário Luterano de Palmas (CEULP/ULBRA)

{charlesrock007,lucasmaraujo,mozart08,roneylson}@gmail.com

**Resumo.** *Este artigo mostra uma visão geral da linguagem XML, apresentando um breve histórico, as principais características de formação, o desenvolvimento de uma DTD e algumas tecnologias utilizadas para apresentação, como o CSS e o XSLT. Fornece também uma breve explicação e referência à extensão docx e às linguagens baseadas em XML, como MathML e SVG.*

**Palavras chave:** XML, DTD, XSLT

## 1. Introdução

O XML (*eXtensible Markup Language*, ou Linguagem de Marcação Estendida) é uma linguagem que surgiu para solucionar problemas de outras linguagens já utilizadas. Nos anos 70 a IBM criou uma linguagem chamada GML (*Generalized Markup Language* ou Linguagem de Marcação de Geografica), que simplifica a descrição de um documento em termos de formato, estrutura organizacional, peças de conteúdo e sua relação, entre outras propriedades. Essa logo chamou a atenção da ISO (*International Organization for Standardization* ou Organização Internacional de Padronização) que é encarregada de padronizar diversas normas técnicas como por exemplo, tamanhos e unidades, códigos de moedas, padronização para linguagem de internet, entre outros. Por volta de 1986, a ISO começou a trabalhar na normalização da linguagem GML, criando a SGML que nada mais era que GML, porém agora se tornando um padrão (*Standard* em inglês). Por volta de 1989 Tim Berners-Lee criou uma linguagem baseada em SGML para a Internet, o HTML (*HyperText Markup Language*, ou Linguagem de Marcação de Hipertexto), tendo em suas primeiras versões regras de sintaxe flexíveis, facilitando a utilização por pessoas que não tinham grande conhecimento sobre publicação na Web. Mesmo hoje HTML sendo uma linguagem bem mais rígida quanto à sintaxe, os navegadores interpretam documentos em HTML por mais que eles não tenham códigos válidos.

Em 1998, a W3C - <http://www.w3.org/> - iniciou o desenvolvimento da linguagem XML, que tem a finalidade de eliminar problemas do HTML. Como se trata apenas de representar, e não de apresentar dados, o XML tem um conteúdo que não se mistura com estilos, como cores, tamanhos de fonte, estilo da fonte, entre outros, pois outras ferramentas se encarregam dessa tarefa, como o CSS (*Cascading Style Sheets* ou Folhas de Estilo em Cascata) e o XSLT (*eXtensible Stylesheet Language Transformation* ou Linguagem de Folhas de Estilo Extensível para Transformação), que serão descritas ao longo do texto.

Neste documento será apresentado como criar e formatar um documentos XML, seguindo as regras de sintaxe, como gerar uma validação dentro dos padrões de DTD (*Document Type Definition*, ou Definição do Tipo do Documento) e como apresentá-los utilizando a ferramenta XSLT.

## 2. Composição de um documento XML

Um documento XML é composto basicamente pela declaração XML, que indica que o documento é um documento XML, seguida pela versão do documento e o seu elemento raiz, que será explicado mais adiante.

A elaboração de um documento XML deve seguir um conjunto de regras, com a finalidade de promover o que se chama de um documento “bem formado”.

Um documento bem formado é um documento que está de acordo com algumas regras de formatação. Além disso, é necessário ainda que o documento seja válido. Para isso ele precisa estar de acordo com a sua estrutura.

### 2.1 Construção do documento

1. A primeira linha deve conter a declaração XML seguida da versão do documento.
2. O documento XML deve conter um único elemento raiz.

### 2.2 Construção dos elementos

- A construção de um elemento deve seguir uma mesma marcação (case-sensitive), ou seja, um elemento com o nome ENCOINFO é diferente de um com o nome Encoinfo.
- Os elementos do documento XML devem estar corretamente aninhados, isso significa que, quando um elemento for aberto, ele só poderá ser fechado após todos os elementos dentro dele terem sido fechados.
- O nome do elemento pode conter letras, dígitos, underscores, hífen, dois pontos, ou pontos, não podendo começar com um número ou caractere de pontuação.
- Todo elemento deve possuir uma *tag* de abertura e uma de fechamento.

A Figura 1, abaixo, apresenta um exemplo simples de um documento XML.

```
<?xml version =“1.0”>
<peessoa>
  <nome>
    José Carvalho da Silva
```

Figura 1. Documento XML

### 2.3 Conteúdo do documento

O conteúdo do documento é caracterizado tanto pelas tags (elemento) que o contém quanto pelo texto que o compõe. Existem caracteres especiais ( & , ' , < , > , ” ), utilizados na construção do documento, que se precisarem ser inseridos no conteúdo, devem ser substituídos pela sua referência de entidade, como exemplificado na Figura 2. É possível ainda identificar que os espaços em branco são preservados.

<pre> ... &lt;teste&gt;   JOSÉ é &amp;lt; que MARIA   JOSÉ é &amp;gt; que MARIA   JOSÉ &amp;amp; MARIA   JOSÉ filho de &amp;apos; MARIA&amp;apos;   Maria mãe de &amp;quot;JOSE&amp;quot;   José                E Maria &lt;/teste&gt; ... </pre>	<pre> ... &lt;teste&gt;   JOSÉ é &lt; MARIA   JOSÉ é &gt; MARIA   JOSÉ &amp; MARIA   JOSÉ filho de 'MARIA'   Maria mãe de "JOSE"   José                E Maria &lt;/teste&gt; ... </pre>
---	--

Figura 2. Conteúdo do documento XML

## 2.4 Estrutura em árvore

Um documento XML apresenta uma estrutura em árvore. Isso significa que elementos podem ter sub-elementos. Um elemento que está dentro de outro é considerado “filho”, e este filho tem um elemento dentro de si, este último é considerado “neto” daquele primeiro. Ou seja, se A contém B que contém C, A é pai de B e avô de C. Se além de B, A contivesse D, B seria irmão de D. Um filho pode ser um elemento ou uma String.

A Figura 3 apresenta exemplo em que “pessoa” é pai de “nome”, “sexo”, “identidade”, “endereço”, “dataNascimento” e “telefone”.

```

<?xml version="1.0">
<pessoa>
  <nome>José</nome>
  <sexo>Masculino</sexo>
  <identidade>9090909090</identidade>
  <endereco>Palmas Tocantins</endereco>
  <dataNascimento>15/08/1990</dataNascimento>
  <telefone>3217-9669</telefone>
</pessoa>

```

Figura 3. Estrutura em árvore

## 3. DTD (*Document Type Definition*)

A DTD tem por objetivo definir um modelo ou estrutura de documentos XML utilizando regras estabelecidas pelo desenvolvedor. Com a DTD pode-se estabelecer quais serão os nós que compõem o documento XML e quais são os tipos de dados de cada nó. Por exemplo, podem-se definir quais são os nós que necessariamente devem ser preenchidos dentro do seu arquivo XML.

A DTD pode ser utilizada para estabelecer um padrão do documento XML e através disso pode-se fazer a transferência de dados entre as aplicações. Também é possível evitar a repetição de dados dentro do XML utilizando artifícios da própria DTD. A figura 4 representa um exemplo simples da estrutura DTD:

```

<!DOCTYPE encoinfo[
  <!ELEMENT encoinfo (edicao, local)>
  <!ELEMENT edicao (#PCDATA)>
  <!ELEMENT local (#PCDATA)>
]>

```

Figura 4. Exemplo de DTD

### 3.1 Construindo uma DTD

A declaração de uma DTD pode ser feita *inline*, ou seja, dentro do próprio arquivo XML, ou pode ser feita em um documento externo, sendo que para isso será necessário criar um arquivo com extensão .dtd e associá-lo ao seu XML. A figura 5 descreve uma estrutura simples de DTD *inline*, enquanto a figura 6 mostra o documento XML com a associação de uma DTD externa e a figura 7 mostra a DTD associada ao documento da figura 3.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pessoa[
  <!ELEMENT pessoa (nome, sexo, identidade, endereco,
dataNascimento, telefone)>
  <!ELEMENT nome (#PCDATA)>
  <!ELEMENT sexo (#PCDATA)>
  <!ELEMENT identidade (#PCDATA)>
  <!ELEMENT endereco (#PCDATA)>
  <!ELEMENT dataNascimento (#PCDATA)>
  <!ELEMENT telefone (#PCDATA)>
]>

<pessoa>
  <nome>João Fulano</nome>
  <sexo>Masculino</sexo>
  <identidade>9090909090</identidade>
  <endereco>Palmas Tocantins</endereco>
  <dataNascimento>14 de março de 1979</dataNascimento>
  <telefone>12345678</telefone>
</pessoa>
```

Figura 5. Exemplo de documento XML com DTD *inline*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE pessoa SYSTEM "dtdExterna.dtd">

<pessoa>
  <nome>João Fulano</nome>
  <sexo>Masculino</sexo>
  <identidade>9090909090</identidade>
  <endereco>Palmas Tocantins</endereco>
  <dataNascimento>14 de março de 1979</dataNascimento>
  <telefone>12345678</telefone>
</pessoa>
```

Figura 6. Exemplo de Documento XML com indicação de DTD externa

```
<!ELEMENT pessoa (nome, sexo, identidade, endereco,
dataNascimento, telefone)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT sexo (#PCDATA)>
<!ELEMENT identidade (#PCDATA)>
<!ELEMENT endereco (#PCDATA)>
<!ELEMENT dataNascimento (#PCDATA)>
<!ELEMENT telefone (#PCDATA)>
```

Figura 7. Exemplo de DTD externo

### 3.2 Elementos

Um documento XML é composto por blocos de construção, que por sua vez são chamados de elementos, atributos ou entidades. Os elementos são os blocos principais de um XML. Um elemento pode assumir diversos tipos, entre eles existem PCDATA, CDATA, EMPTY e ANY. A tabela 1 apresenta as características de cada um deles e sua representação na DTD:

Tabela 1. Tipos de elementos

Notação	Característica do bloco	Definição na DTD
PCDATA	Blocos que terão cada caractere analisado pelo interpretador XML. Para isso são necessários alguns cuidados com os símbolos “<”, “>” e “&” substituindo-os pelas suas respectivas entidades.	<!ELEMENT nome (#PCDATA)>
CDATA	Blocos que não serão analisados pelo interpretador XML.	<!ELEMENT nome (#CDATA)>
EMPTY	Sem conteúdo.	<!ELEMENT nome EMPTY>
ANY	Contém qualquer tipo de conteúdo.	<!ELEMENT nome ANY>

É possível que um elemento apresente “filhos”. Nesse caso, é preciso especificar na DTD todos os “filhos” presentes no documento XML no local onde deveria ser especificado o conteúdo daquele elemento. A sintaxe da indicação de elementos-filhos na DTD pode ser visualizada na figura 8:

```
<!ELEMENT nome (filho1)>
<!ELEMENT nome (filho1, filhos2, ...)>
```

Figura 8. Sintaxe de elementos-filhos na DTD

Um exemplo de elemento-pai e elementos-filhos é mostrado na figura 9. Observa-se que o elemento “email” possui como filhos os elementos: “de”, “para”, “assunto” e “corpo”.

```
<!ELEMENT email (de, para, assunto, corpo)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT assunto (#PCDATA)>
<!ELEMENT corpo (#PCDATA)>
```

Figura 9. Exemplo de elementos-filhos na DTD

Outra característica do documento XML que deve ser expressa na DTD é a natureza de ocorrência dos elementos existentes. Para isso, utiliza-se alguns operadores especiais, que estão explicitados na tabela 2:

Tabela 2. Operadores especiais para definição da natureza de ocorrência

Ocorrência	Definição na DTD
uma vez	<!ELEMENT nome_do_elemento (tipo)>
uma ou mais vezes	<!ELEMENT nome_do_elemento (filho+)>
zero ou mais vezes	<!ELEMENT nome_do_elemento (filho*)>
opcional	<!ELEMENT nome_do_elemento (filho?)>
alternativa	<!ELEMENT nome_do_elemento (filho1   filho2)>

### 3.3 Atributos

Os atributos oferecem informações adicionais aos elementos. Pode-se observar a sintaxe da definição de um atributo na DTD na figura 10:

```
<!ATTLIST nome_do_elemento nome_do_atributo tipo_do_atributo valor_padrao>
```

Figura 10. Sintaxe de um atributo na DTD

No exemplo exibido na figura 11, é possível identificar a existência de um atributo “nome” em um elemento chamado “cantor”:

```
<!ELEMENT cantor (#PCDATA)>  
<!ATTLIST cantor nome CDATA #REQUIRED>
```

Figura 11. Exemplo de um atributo na DTD

A tabela abaixo mostra quais são os atributos que podem ser usados e suas respectivas descrições:

Tabela 3. Tipo e descrição dos atributos

Tipo	Descrição
CDATA	O valor é do tipo Character Data
(valor1, valor2...)	O valor deve ser um da lista
ID	O valor é um identificador único
IDREF	O valor é o ID de outro elemento
IDREFS	O valor é uma lista de outros IDs
NMTOKEN	O valor é um nome XML válido
NMTOKENS	O valor é uma lista de nomes XML
ENTITY	O valor é uma entidade
ENTITIES	O valor é uma lista de entidades
NOTATION	O valor é o nome de uma notação
xml:	Valor XML pré-definido

Através da DTD, é possível determinar se um atributo é obrigatório ou opcional, e ainda especificar um valor padrão. Alguns tipos de “valor padrão” são exibidos na tabela 4:

Tabela 4. Valores dos atributos

Valor	Descrição
“valor”	O valor padrão do atributo
#REQUIRED	O atributo é obrigatório
#IMPLIED	O atributo não é obrigatório
#FIXED “valor”	O valor do atributo é fixo

### 3.4 Entidades

Entidades são variáveis que guardam valores pré-determinados para serem usados a qualquer hora dentro do XML. A sintaxe da definição de uma entidade na DTD pode ser visualizada na figura 12:

```
<!ENTITY nome "valor">
```

Figura 12. Sintaxe de uma entidade na DTD

Para utilizar uma entidade dentro do XML, deve-se colocar primeiro o “&” seguido do nome da entidade e posteriormente o sinal “;”. A figura 13 traz um exemplo de uma entidade sendo utilizada em um documento XML.

```
<!ENTITY encoinfo "Encoinfo - Encontro de Estudantes de
Informática do Estado do Tocantins">
Exemplo do XML:
<evento>
    &encoinfo;
</evento>
```

Figura 13. Exemplo de entidade

### 3.5 Validação

Após a conclusão do documento XML e da DTD, é preciso verificar a sua validade. Essa validação é feita por diversas ferramentas, como por exemplo:

- Validome: Ferramenta on-line onde é possível realizar a verificação copiando e colando o código. Disponível em [www.validome.org](http://www.validome.org)
- XML SPY: Ferramenta desktop, porém paga. Disponível em [www.altova.com/xmlspy.html](http://www.altova.com/xmlspy.html)

## 4. Apresentação do XML

Até aqui, foi possível entender como o XML permite a modelagem de dados. Mas após realizar essa modelagem, é preciso encontrar uma forma de apresentar o que foi feito. Como o XML é uma linguagem de marcação, pode-se fazer isso utilizando CSS – *Cascading Style Sheet* (folhas de estilo que definem cores, fontes, tamanhos, espaçamentos, margens, etc).

Para inserir uma folha de estilos em um documento XML, basta incluir a seguinte linha:

```
<?xml version="1.0">
<?xml-stylesheet type="text/css" href="nomedoarquivo.css"?>
```

Figura 14. Associação de um documento XML a um documento CSS

A figura 15 apresenta o conteúdo do arquivo denominado exemplo.xml, contendo um elemento raiz denominado <peessoa> e alguns elementos-filhos, como <nome>, <endereço>, <dataNascimento> e <telefone>. A esse arquivo, está associada uma folha de estilos, apresentada na figura 16. Nessa folha de estilos, encontram-se diversos atributos: font-family (fonte do texto), font-size (tamanho do texto), font-weight (espessura do texto), color (cor) e display:block (determina uma quebra de linha). Quando o arquivo exemplo.xml é executado em um browser, os elementos do documento XML aparecem com os efeitos descritos no documento CSS. A execução do exemplo.xml no browser pode ser visualizada na figura 17.

```

<?xml version = "1.0" encoding = "ISO-8859-1"?>
<?xml-stylesheet type = "text/css" href = "exemplo.css"?>

<peessoa>
  <nome>Maria Carvalho de Souza</nome>
  <endereco>Avenida das Américas, 645</endereco>
  <dataNascimento>03/12/1967</dataNascimento>
  <telefone>063 3402-1919</telefone>
  <telefone>063 9923-1010</telefone>
</peessoa>

```

Figura 15. Exemplo.xml

```

peessoa{font-family:"Courier New";}
nome{font-weight:bolder; font-size:20pt; color:#6A5ACD; display:block;}
endereco, dataNascimento{font-size:16pt; color:#8470FF; display:block;}
telefone{font-size:14pt; color:#6495ED; display:block;}

```

Figura 16. Exemplo.css

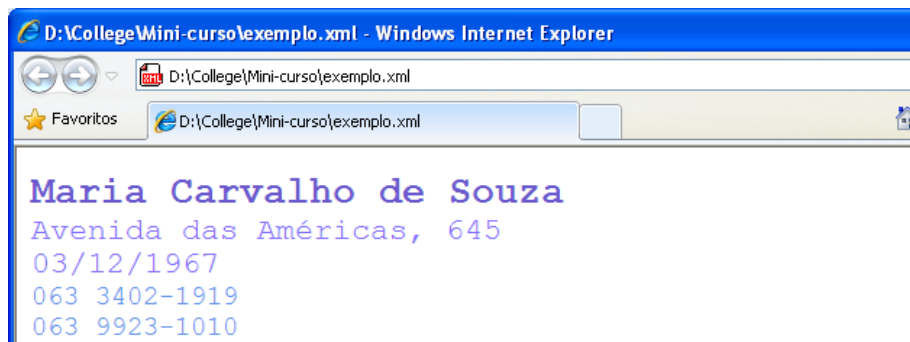


Figura 17. Apresentação do exemplo.xml no browser

O CSS é bastante eficiente para realizar a apresentação de alguns documentos. Apesar dos diversos recursos que oferece, ele apresenta algumas limitações para a formatação do XML. Buscando uma ferramenta mais poderosa, a W3C reuniu algumas recomendações em uma tecnologia denominada XSL – *eXtensible Stylesheet Language*. No mini-curso, serão abordadas duas partes do XSL:

- XSLT (*eXtensible Stylesheet Language Transformation*) – utilizada para transformar um documento XML em outros formatos, como DOC ou HTML, por exemplo.
- XPath – permite navegar pelo documento XML.

O XSLT funciona como uma linguagem de programação, sendo possível executar uma mesma instrução diversas vezes, como em um laço de repetição e fazer verificações, como em estruturas condicionais. Além disso, é possível definir algumas formatações como cores e tamanhos, assim como no CSS.

Para relacionar um documento XML com um documento XSL, deve-se inserir a linha:

```

<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "nomedoarquivo.xsl"?>

```

Figura 18. Associação de um documento XML a um documento XSL



A Tabela 5 apresentada abaixo mostra alguns elementos que podem ser utilizados em um documento XSLT:

Tabela 5. Elementos do XSLT

ELEMENTO	DEFINIÇÃO
xsl:value-of	Extrai o conteúdo de um elemento XML
xsl:for-each	Delimita um laço de repetição em um conjunto de elementos XML.
xsl:attribute	Permite incluir elementos CSS para formatar cores, tamanhos, margens, etc.
xsl:if	Testa uma expressão booleana.
xsl:sort	Ordena elementos.
xsl:choose	Funciona como um conjunto if-else. Nesse caso, utiliza-se when-otherwise.

Os elementos XSLT selecionam as partes do XML nas quais será realizada alguma ação através de expressões XPath, que possibilita uma navegação pelo documento XML. Os caminhos definidos pelo XPath são parecidos aos utilizados em um sistema de arquivos, sendo que a “/” indica o elemento raiz. Pode-se observar algumas expressões do XPath na Tabela 6:

Tabela 6. Expressões do XPath

EXPRESSÕES	DEFINIÇÃO
Nome_do_nó	Seleciona o nó e seus filhos
/	Seleciona a partir da raiz do documento XML
//	Seleciona um nó a partir do atual independente de onde esteja
.	Seleciona o nó atual
..	Seleciona o nó pai do nó atual
@	Seleciona atributos

O XPath ainda permite a inclusão de predicados no momento de selecionar um nó. Os predicados são importantes para selecionar um nó específico na árvore. A Tabela 7 apresenta alguns predicados:

Tabela 7. Predicados

PREDICADO	DEFINIÇÃO
nome_do_nó[posição]	Seleciona o nó de acordo à posição que ele ocupa na árvore.
nome_do_nó[last()]	Seleciona o último nó com aquele nome.
nome_do_nó[position() < X]	Seleciona o nó caso a condição de posição seja satisfeita.
nome_do_nó[@atributo]	Seleciona o nó se ele possuir o atributo indicado. É possível ainda testar o valor dos atributos no momento de realizar a seleção.

A figura 20 traz um código desenvolvido em XSLT para realizar a apresentação do documento XML exposto na figura 19 (só diferencia-se do exemplo.xml da figura 15 por estar associado a um outro arquivo, no caso, exemplo2.xsl) . O documento XML está sendo “transformado” em um documento HTML constituído por uma tabela que apresenta as informações de <essoa>.

```

<?xml version = "1.0" encoding = "ISO-8859-1"?>
<?xml-stylesheet type = "text/xsl" href = "exemplo2.xsl"?>
<peessoa>
  <nome>Maria Carvalho de Souza</nome>
  <endereco>Avenida das Américas, 645</endereco>
  <dataNascimento>03/12/1967</dataNascimento>
  <telefone>063 3402-1919</telefone>
  <telefone>063 9923-1010</telefone>
</peessoa>

```

Figura 19. Exemplo2.xml

Para estilizar a tabela, utiliza-se atributos semelhantes aos utilizados em CSS, sendo que eles são incluídos no elemento <xsl:attribute>. A extração dos elementos presentes no XML para a página HTML são realizadas com o elemento <xsl:value-of>, e a repetição de uma ação é feita utilizando um <xsl:for-each>.

```

<?xml version = "1.0" encoding = "ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match = "/">
  <html>
    <head><title>Exemplo de XSLT</title></head>
    <body>
      <table border = "1" width = "100%">
        <xsl:attribute name = "style">
          text-align:center;
          font-family: "Courier New";
          border-collapse:collapse;
        </xsl:attribute>
        <tr>
          <td colspan = "4">
            <xsl:attribute name = "style">
              font-weight:bolder; font-size:20pt; background:#CFD9E9;
            </xsl:attribute>
            CADASTRO
          </td>
        </tr>
        <tr>
          <xsl:attribute name = "style">
            font-weight:bolder; font-size:16pt;
          </xsl:attribute>
          <td>Nome</td>
            <td>Endereço</td> <td>Data de nascimento</td>
            <td>Telefone(s)</td>
        </tr>
        <tr>
          <td><xsl:value-of select= "peessoa/nome" /></td>
          <td><xsl:value-of select= "peessoa/endereco" /></td>
          <td><xsl:value-of select= "peessoa/dataNascimento" /></td>
          <td>
            <xsl:for-each select = "peessoa/telefone">
              <xsl:value-of select= "." /><br />
            </xsl:for-each>
          </td>
        </tr>
      </table>
    </body> </html>
  </xsl:template>
</xsl:stylesheet>

```

Figura 20. Exemplo2.xsl

A execução de exemplo2.xml no browser pode ser visualizada na figura 21.

CADASTRO			
Nome	Endereço	Data de nascimento	Telefone (s)
Maria Carvalho de Souza	Avenida das Américas, 645	03/12/1967	063 3402-1919 063 9923-1010

Figura 21. Apresentação do exemplo2.xml no browser

Alguns sites da W3C podem ajudar: <http://www.w3.org/Style/XSL/> - especificações, tutorias, artigos, referências e artigos sobre XSL; <http://www.w3.org/TR/xpath> - recomendações sobre XPath; <http://www.w3.org/TR/xslt> - recomendações sobre XSLT.

## 5. Utilizações do XML

### 5.1 MathML

MathML - *Mathematical Markup Language* ou Linguagem de Marcação Matemática - é uma tecnologia desenvolvida pela W3C que é utilizada para representar fórmulas matemáticas na *web* através da linguagem XML. Em 1999 surgiu a versão 1.01 do MathML e posteriormente, no ano de 2001, surgiu a versão 2.0.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <msqrt>
        <mrow>
          <mfrac>
            <mrow>
              <mn>162</mn>
            </mrow>
            <mrow>
              <mn>2</mn>
            </mrow>
          </mfrac>
        </mrow>
      </msqrt>
    </mrow>
    <msup>
      <mfenced>
        <mrow>
          <mn> 1 </mn>
          <mrow>
            <mo> + </mo>
            <mn> 2 </mn>
          </mrow>
        </mrow>
      </mfenced>
      <mn>2</mn>
    </msup>
  </mrow>
</mfrac>
</math>

```

Figura 22. Exemplo de aplicação do MathML

A figura 22 acima representa um modelo simples de implementação do MathML demonstrando a simplicidade de implementação. O código da figura 22 ao ser interpretado pelo navegador gera a expressão matemática em forma de texto. A saída do código gera a expressão representada na figura 23:

$$x = \frac{\sqrt{\frac{162}{2}}}{(1+2)^2}$$

Figura 23. Saída no browser da aplicação exibida na figura 22

Ao se observar a figura 22 temos na linha dois a declaração raiz do MathML, isso indicará ao *browser* o tipo de código(MathML) que ele deverá interpretar. Alguns elementos contidos no código MathML são descritos na tabela abaixo:

Tabela 8. Alguns elementos MathML e suas respectivas descrições

Elemento	Descrição
<mi></mi>	constante matemática
<mo></mo>	operador matemático.
<mn></mn>	número
<mrow></mrow>	agrupa elementos MathML
<mfrac></mfrac>	fração matemática (será necessário dois elementos <mrow></mrow> declarados dentro da mesma)
<msqrt></msqrt>	símbolo da raiz quadrada
<msup></msup>	potência(base, expoente)
<mfenced></mfenced>	parênteses

Mais informações sobre o MathML podem ser encontradas no seguinte link: <http://www.w3.org/TR/REC-MathML/> - visão geral sobre o MathML.

## 5.2 SVG – Scalable Vector Graphics

SVG – *Scalable Vector Graphics* – é uma linguagem desenvolvida pela W3C que permite o desenvolvimento de gráficos bidimensionais utilizando um código escrito em XML. As imagens desenvolvidas nessa linguagem podem ter seu tamanho alterado sem perda da qualidade. É possível agrupar objetos (linhas, polígonos e imagens), aplicar estilos e inserir códigos em javascript para realizar animações.

Atualmente, essa linguagem encontra-se subdividida em conjuntos, sendo que cada um deles possui objetivos próprios. Um desses subconjuntos é o *Mobile SVG*, que traz especificações para o desenvolvimento de gráficos destinados a dispositivos móveis. Esse subconjunto, por sua vez, encontra-se subdividido em *SVG Tiny*, aplicado a dispositivos com limitados recursos de processamento gráfico, como celulares e em *SVG Basic*, utilizado em dispositivos que possuem maior capacidade de processamento gráfico, como PDAs (*Personal Digital Assistants*).

A figura 24 mostra um código de uma imagem em SVG (desenho.svg). O elemento <svg> possui alguns atributos como o xmlns, utilizado para especificar o tipo

do arquivo em desenvolvimento, *width* e *height* para definir a largura e altura da imagem, respectivamente.

```
<svg xmlns="http://www.w3.org/2000/svg" width="200" height="200">
  <circle cx="100" cy="100" r="90" stroke="#000" stroke-width="10"
fill="#FD0" />
  <path stroke="#000" stroke-width="10" d="M 80,60 L 80,120" />
  <path stroke="#000" stroke-width="10" d="M 120,60 L 120,120" />
  <path stroke="#000" stroke-width="10" fill="none" d="M 40,120 C 50,180
150,180 160,120" />
</svg>
```

Figura 24. Desenho.svg

Ainda observando a figura 8, nota-se a presença de um dos elementos primitivos do SVG, o <circle> (círculo). Outros elementos primitivos são: <rect> (retângulo) e <line> (linha). O elemento circle possui diversos atributos:

- cx: posição horizontal do centro do círculo;
- cy: posição vertical do centro do círculo;
- r: raio;
- stroke: cor da borda em hexadecimal;
- stroke-width: largura da borda;
- fill: cor do preenchimento em hexadecimal;

Após as especificações do círculo, são definidas três formas livres, através do <path>. Esse elemento forma um desenho constituído por diversos nós interligados. Os atributos stroke e stroke-width funcionam assim como no elemento <circle>. Nos dois primeiros elementos <path>, o atributo d começa com o M para indicar a localização (coordenadas) do primeiro nó. Depois disso, há o L, indicando que a ligação com o próximo nó ocorrerá através de uma linha. Após o L estão as coordenadas do segundo nó. O terceiro <path> apresenta o atributo d com uma diferença: ao invés de uma linha entre os pontos, existe uma Curva de Bézier (curva definida matematicamente que possui alguns pontos de controle). As coordenadas dos pontos de controle e o ponto final são definidos após o C. Quando esse arquivo é executado em um browser, obtém-se a saída apresentada na figura 25.

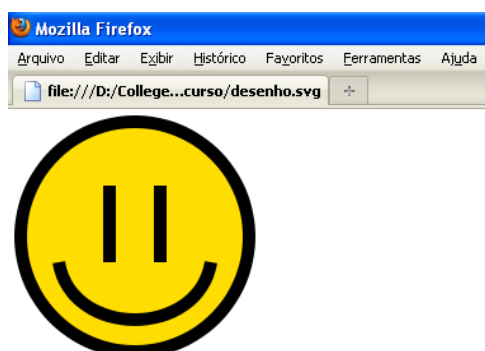


Figura 25. Apresentação do desenho.svg no browser

O SVG pode ser utilizado para gerar aplicações gráficas para web e desktop, inclusive promovendo uma maior interação com os usuários. Atualmente, já é aplicado no desenvolvimento de imagens para GIS (*Geographic Information System*), mapeamento e design de gráficos utilizados em empresas de construção civil e telecomunicações. Mais informações a respeito do SVG podem ser encontradas nos seguintes links: <http://www.w3.org/Graphics/SVG/About.html> - visão geral do SVG; <http://www.w3.org/TR/SVG11/> - especificações para o SVG 1.1; <http://www.w3.org/TR/SVGMobile/> - especificações para o *Mobile SVG*.

### 5.3 DOCX

Docx é uma extensão de um documento da Microsoft Word 2007, que faz parte do pacote da MS Office 2007. Essa extensão foi criada para compartilhar e abrir documentos mesmo que esses não tenham sido editados pelo mesmo software do arquivo original.

O doc necessitava de um programa compatível para abri-lo, esse foi um grande problema durante os anos de 1980 a 1990, pois os usuários eram obrigados a guardar seus documentos em um formato de texto simples perdendo sua formatação original. Esse novo formato pode ser visualizado por diversas plataformas não precisando utilizar um software da Microsoft para abrir-lo, facilitando a visualização de documentos compartilhados. Essa extensão pode ser interpretada por um descompactador de arquivos, abrindo a pasta de palavras e escolhendo o document.xml que pode ser executado em qualquer editor de texto.

Outra grande vantagem de fazer uma atualização para docx é o menor tamanho do arquivo. Geralmente um padrão carta com algumas centenas de palavras tem o tamanho de apenas 10kb usando docx, bem menor que o formato anterior com 100kb utilizando doc. Isso porque um docx é um arquivo zipado contendo pastas com arquivos XML, otimizando o envio desses por email.

## 6. Conclusão

Percebe-se com o estudo de XML que temos uma poderosa linguagem de programação para Web sendo apoiado por muitas outras ferramentas de apresentação dos dados, como XSLT. Pode-se entender XML basicamente como um conteúdo em formato de texto, hierarquicamente estruturado. É uma linguagem de marcação que pode introduzir novas possibilidades e trazer melhor integração entre dados e usuários.

Pelo fato de existir grande volume de dados na internet e usuários mais exigentes, há sempre a necessidade de estabelecerem-se parâmetros, regras e padrões para a utilização do XML, fazendo com que esta seja uma linguagem cada vez mais eficiente e que atenda as necessidades dos usuários.

O XML é uma linguagem bastante flexível, e por ter essa característica é facilmente adaptada a qualquer área, permitindo a criação de diversos vocabulários, na área de matemática, por exemplo, temos o MathML que apresenta fórmulas matemáticas, o SVG que é utilizada para descrever de forma vetorial desenhos e gráficos bidimensionais, de forma estática, dinâmica ou animada ou o GML (*Geography Markup Language* ou Linguagem de Marcação Geográfica) que serve como uma modelagem de sistemas geográficos, assim como muitas outras.

Um usuário com média experiência em desenvolvimento de conteúdo para a Web, por já conhecer HTML, ao ver o XML pela primeira vez, terá bastante facilidade de entendê-la.

Os navegadores de internet reconhecem o HTML mesmo que sua sintaxe esteja fora dos padrões, o XML é um mais rígido quanto a esse erro, uma página só é apresentada se toda sua sintaxe estiver completamente correta e dentro dos padrões definidos.

Portanto, observa-se que o XML é uma linguagem clara e legível tanto para máquinas quanto para as pessoas, suporta uma grande variedade de aplicações, é de fácil construção e muito flexível variando de acordo com as necessidades de aplicação.

## 7. Bibliografia

(SOUZA, 2007) Souza, Jackson Gomes. **Tópicos especiais I – XML** (Notas de Aula) Disponível em: <http://www.ulbra-to.br/DownloadArquivo.aspx?idArquivo=98dcdd73-279d-4578-9edd-8d5863c04526>. Acesso em agosto de 2009.

(W3C, 2000) **W3C Recommendation**. Extensible Markup Language (XML) 1.0 (2<sup>a</sup> ed). Disponível em <http://www.w3.org/TR/REC-xml>. Acesso em Agosto 2009.

(W3C, 1999b) **W3C Recommendation**. XML Path language (XPath) v. 1.0. Disponível em <http://www.w3.org/TR/xpath>. Acesso em Agosto 2009.

(W3C, 2001a) **W3C Recommendation**. XML Linking Language (XLink) v. 1.0. Disponível em <http://www.w3.org/TR/xlink>. Acesso em Agosto 2009.

(W3C, 2001b) **W3C Recommendation**. Scalable Vector Graphics (SVG) 1.0 Specification. Disponível em <http://www.w3.org/TR/SVG>. Acesso em Agosto 2009.

(W3C, 2001c) **W3C Recommendation**. Extensible Stylesheet Language (XSL) v. 1.0. Disponível em <http://www.w3.org/TR/xsl/>. Acesso em Agosto 2009.





# Persistência de Dados em Java com JPA e Toplink

Vinicius Teixeira Dallacqua

Curso de Tecnologia em Sistemas para Internet  
Instituto Federal de Educação, Ciência e Tecnologia - IFTO  
AE 310 Sul, Avenida LO 05, s/n, Plano Diretor Sul, Palmas-TO  
dev.webx@gmail.com

***Resumo.** Neste artigo serão apresentadas algumas características da persistência Java com sua API JPA unida com o framework TopLink para executar e fazer o controle dos comandos de persistência.*

**Palavras chave:** Java, Persistência, JPA, TopLink.

## 1. Introdução

Em qualquer linguagem de programação e para, quase, qualquer tipo de aplicação temos a necessidade de armazenar, recuperar, editar e remover dados de alguma base de dados. Para isso cada linguagem oferece soluções e implementações das mesmas estratégias básicas.

Neste contexto, persistência é um conceito fundamental no desenvolvimento de aplicações. Se um sistema de informação não for capaz de preservar dados quando for “desligado”, o sistema não terá um uso prático. Quando falamos de persistência, na grande maioria das vezes nos referimos a respeito do armazenamento de dados em um banco de dados relacional. Neste caso, tradicionalmente usamos a linguagem SQL (*Structured Query Language*).

A SQL possui duas divisões principais: DDL (*Data Definition Language*, ou Linguagem de Definição de Dados) e DML (*Data Manipulation Language*, ou Linguagem de Manipulação de Dados). A primeira é utilizada para criar e alterar e estrutura dos elementos dos bancos de dados, usando comandos como *CREATE*, *ALTER* e *DROP*. Já a DML é usada para manipular e recuperar dados por meio de comandos como *SELECT*, *INSERT*, *UPDATE* E *DELETE*.

### 1.1. Usando SQL com Java

Quando trabalhamos com banco de dados com Java passamos os comandos SQL para a base de dados pela API *Java Database Connectivity (JDBC)*. Tendo o SQL sido gerado e incorporado no código Java programaticamente (a mão) ou criado em tempo de execução por um código Java, usamos a API *JDBC* para preparar os parâmetros de nosso código SQL para ser então executado, interpretado pelo banco de dados e então recuperado em forma de *result set*. Estas são tarefas de baixo nível no acesso de dados.

### 1.2. ORM – *Object/Relational Mapping*

ORM ou *Object/Relational Mapping (Mapeamento Objeto/Relacional)* é uma técnica que aborda o mapeamento de banco de dados em objetos de entidades de persistência. Uma solução ORM consiste das seguintes partes:

- Uma API para prover as operações básicas (também conhecidas como CRUD, inglês para Cadastro, Recuperação, Alteração, Exclusão) em objetos de classes persistentes.
- Uma linguagem ou API para especificar as consultas que se referem a classes e propriedades de classes.
- Uma facilidade para a especificação de metadados de mapeamento.
- Uma técnica para que a implementação ORM interaja com objetos transicionais para prover checagens, associações, e outras funções de otimização.

Estamos usando o termo *full* ORM para incluir qualquer camada de persistência onde SQL é automaticamente gerado de um mapeamento baseado em metadados.

## 1.2. Mapeamento total de objeto

Mapeamento total de objeto é o nível de ORM que iremos usar. Este nível suporta uma modelagem sofisticada de objetos: composição, herança e polimorfismo. A camada de persistência implementa uma camada transparente de persistência; classes persistentes não herdam de nenhuma classe básica ou tem que implementar uma interface especial. Estratégias eficientes de busca (*lazy*, *eager* e *prefetching*) e estratégias de *cache* são implementadas transparentemente da aplicação.

## 2. EJB 3 + JPA

Uma grande melhoria da versão 2.1 para a versão 3 do EJB (*Enterprise JavaBeans*) (EJB, 2009) foi a adição da JPA (*Java Persistence API*) (JPA, 2009), que é uma especificação da Sun Microsystems com a finalidade de simplificar o modelo de persistência de entidades mantidas em bancos de dados relacionais e a forma de manipulação destas entidades, além de adicionar possibilidades que não eram possíveis no EJB 2.1. É importante destacar que, apesar da JPA ter sido lançada como parte da especificação EJB 3, ela pode ser utilizada mesmo em projetos que não utilizem EJB.

A JPA lida com a forma que os dados relacionais são mapeados para objetos Java (“entidades de persistência”), a forma que estes objetos são armazenados em um banco de dados relacional para que eles possam, então, ser acessados quando necessário. Em adição para simplificar o modelo de entidade de persistência, a JPA padroniza o mapeamento objeto-relacional, permitindo que diferentes implementações compatíveis possam ser usadas, como são os casos das bibliotecas TopLink (Oracle, 2009) e Hibernate (Hibernate, 2009).

### 2.1. *Entity Beans* (Beans de Entidade)

Um bean de entidade representa dados em meio de armazenamento (como um banco de dados relacional). Cada bean de entidade corresponde a uma tabela em um banco de dados, e cada instancia de um bean corresponde a uma coluna desta tabela. Beans de entidade não são limitados para representar bancos de dados relacionais, podendo representar dados de outros tipos de armazenamentos de dados. No entanto, no contexto deste trabalho iremos nos focar em mapeamento de bancos de dados relacionais.

A ênfase aqui é a palavra “representa”. Um bean de entidade não é parte de um banco de dados relacional, entretanto contem dados que são carregados do banco de

dados no momento apropriado, e dados que são armazenados de volta ao banco de dados no tempo apropriado.

Um bean de entidade pode ser compartilhado, isto é, múltiplos clientes podem usar do mesmo bean de entidade. Ele oferece a vários clientes a habilidade de acessar o mesmo dado que este bean represente. Obviamente, que é aqui que a gerencia de transações (e seu controle de integridade) providenciado pelo container de aplicações Java é importante. Em adição, os beans de entidade são persistentes. Isso significa que bean de entidade existe mesmo após que a aplicação que o usa termine sua execução. Um bean de entidade poderá, também, ter relacionamentos com outros beans de entidade, assim como uma tabela em um banco de dados relacional.

Um bean de entidade pode gerenciar sua própria persistência, ou deixar o container de aplicações Java gerenciá-la. Com o bean gerenciando sua própria persistência, o código do bean de entidade também inclui declarações de SQL para acesso da base de dados. Quando o container é quem gerencia, o próprio container gera automaticamente as chamadas de acesso ao banco de dados.

Para criar um bean de entidade precisamos apenas criar uma classe e usar as anotações apropriadas. A classe é codificada como POJO (*Plain Old Java Object*). Uma classe de um bean de entidade deve parecer com o código apresentado na Figura 1.

```
@Entity public class Employee {
    private Long id;
    private String name;
    private String position;
    private double salary;
    private List teams;

    public PlayerBean() {}

    @id(generate=AUTO) public Long getPlayerId() {
        return id;
    }

    public String setPlayerId(Long id) {
        this id=id;
    }

    public String getName() {
        return name;
    }

    public String setName() {
        this name= name;
    }

    ...

    public Collection getTeams() {
        return teams;
    }

    public Collection setTeams(Collection teams) {
        this teams=teams;
    }

    ...
}
```

**Figura 1. Exemplo de Bean de Entidade**

A anotação `@Entity` indica que esta classe é um bean de entidade persistente na tabela *Employee*. Como não definimos por anotação o nome da tabela, o container entenderá que o nome da tabela é o mesmo da classe. Poderíamos indicar o nome da tabela com a anotação `@Table (name="nomeDaTabela")` caso o nome da classe não for igual ao nome da tabela que ela representa.

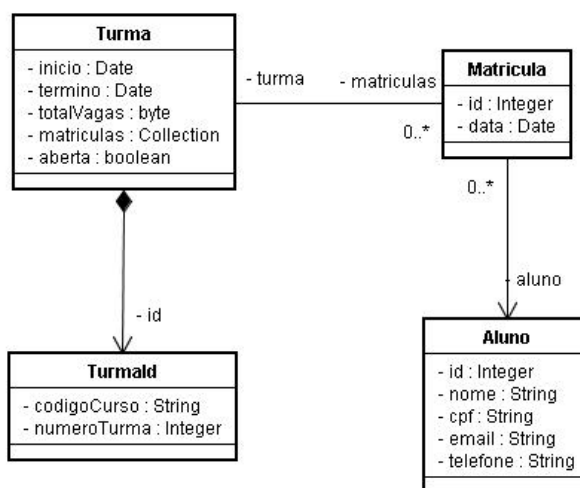
A anotação `@Id` é usada para indicar que temos uma chave primária no atributo especificado. O atributo `generate=AUTO` determina ao container usar uma estratégia para gerar a chave primária que é mais apropriada para o banco de dados usado pela aplicação. Poderíamos indicar a estratégia de geração da chave primária através da anotação `@GeneratedValue`, como por exemplo:

```
@Id @GeneratedValue(strategy=GenerationType.IDENTITY) private Long id;
```

É importante notar que as anotações sempre vêm antes dos atributos, ou métodos, alvo (ou declaração da classe no caso da anotação `@Entity`).

### 3. Exemplo de Mapeamento de Entidades

Para auxiliar no entendimento do mapeamento de entidades iremos usar como base o diagrama de classes ilustrado na Figura 2.



**Figura 2: Diagrama de Classes com as Entidades a serem Mapeadas**

Para representação da entidade Aluno, é definido o bean apresentado na Figura 3.

```
@Entity
@NamedQueries(value={
    @NamedQuery(name="alunoByCpf", query="select a from Aluno a where a.cpf = :cpf"),
    @NamedQuery(name="alunoByNome", query="select a from Aluno a where a.nome like :nome")
})
public class Aluno implements Serializable {

    @Id @GeneratedValue(strategy=GenerationType.IDENTITY) private Integer id;
    @Column(name="") private String nome;
```

```

private String cpf;

private String email;

private String telefone;
// ... getters e setters
}

```

**Figura 3: Mapeamento da Entidade Aluno**

Na classe Aluno temos como novidade o uso da anotação `@NamedQueries` que organiza um array de `@NamedQuery`. A `@NamedQuery` traduz consultas JPA para uma consulta SQL. A linguagem usada para criar uma `@NamedQuery` é a JPA Query Language (JPQL). Esta consulta será referenciada em qualquer lugar pelo nome estabelecido. Pode ser forçada a substituição dos parâmetros (identificadores precedidos por “:”) antes de executar a consulta.

Para representação da entidade Turma, é definido o bean apresentado na **Figura 4.**

```

@Entity
@NamedQueries({
    @NamedQuery(name="turmasByCurso", query="select t from Turma t
where t.id.codigoCurso = :codigoCurso"),
    @NamedQuery(name="turmaUltimoNumero", query="select
MAX(t.id.numeroTurma) from Turma t where t.id.codigoCurso =
:codigoCurso")
})
public class Turma implements Serializable {

    @EmbeddedId private TurmaId id;
    @Temporal(TemporalType.DATE) @Column(nullable=false) private Date
inicio;
    @Temporal(TemporalType.DATE) private Date termino;
    private byte totalVagas;

    @OneToMany(mappedBy="turma", cascade=CascadeType.ALL,
fetch=FetchType.EAGER)
    private Collection<Matricula> matriculas;

    private boolean aberta;

    //.. getters e setters
}

```

**Figura 4: Mapeamento da Entidade Turma**

Na classe Turma temos a anotação `@EmbeddedId`, que é usada para identificadores compostos. Assim como os atributos da classe turma, os atributos de `TurmaId` são considerados persistentes e correspondem a colunas na tabela turma.

A anotação `@Temporal` define atributos persistentes de data/hora. O atributo “`TemporalType.DATE`” da anotação define que serão armazenados apenas valores de data (dia, mês, ano).

A anotação `@Column(nullable=false)` define que a coluna deverá receber algum valor em uma inserção de dados, ou seja, não aceita passagem de valores nulos ao executar um INSERT.

Como a tabela Turma tem um relacionamento com a tabela Matricula, mapeamos este relacionamento com a anotação `@OneToMany` (que é o relacionamento representado no diagrama de entidade relacionamento acima) que define o relacionamento 1:n em um atributo do tipo Collection (que pode também ser do tipo List). O atributo “mappedBy” da anotação define qual o atributo da classe Matricula participa deste relacionamento. O atributo “cascade=CascadeType.ALL” obriga que todas as operações que acontecem com o objeto turma sejam refletidos (cascadeados) com os objetos matriculas associados a turma em questão na operação executada. O atributo “fetch=FetchType.EAGER” da anotação faz com que, ao recuperar uma turma, sejam também recuperadas as matriculas associadas a esta turma.

Para representação da entidade Matricula, é definido o bean apresentado na **Figura 5.**

```
@Entity
public class Matricula implements Serializable {

    @Id @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer Id;
    @ManyToOne private Aluno aluno;
    @ManyToOne private Turma turma;
    @Temporal(TemporalType.TIMESTAMP)
    @Column(nullable=false)
    private Date data;

    //.. getters e setters
}
```

**Figura 5: Mapeamento da Entidade Matricula**

A anotação `@ManyToOne` define a outra parte do relacionamento 1:n entre Turma (1)-(n) Matricula. O atributo “TemporalType.TIMESTAMP” para a anotação `@Temporal` define que o atributo em questão poderá receber dados do tipo data e hora.

Para representação da entidade TurmaId, é definido o bean apresentado na **Figura 5.**

```
public class TurmaId implements Serializable {
    private String codigoCurso;
    private Integer numeroTurma;
    //.. getters e setters
}
```

**Figura 6: Mapeamento da Entidade TurmaId**

#### 4. O Arquivo de Configuração Persistence.xml

O arquivo persistence.xml contém as configurações para acesso à base de dados, entre outras como, por exemplo, o dialeto utilizado e o provedor de persistência (no nosso caso o provedor de persistência será o TopLink, mas poderia ser outro como Hibernate,

pois a JPA sozinha não persiste dados na base de dados, apenas mapeia as entidades e as regras de persistência). Um exemplo de um persistence.xml funcional seria como o apresentado na Figura 7.

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0"
xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
  <persistence-unit name="MinhaPU" transaction-type="RESOURCE_LOCAL">
    <provider>oracle.toplink.essentials.PersistenceProvider</provider>
    <class>com.webx.escola.domain.Aluno</class>
    <class>com.webx.escola.domain.Matricula</class>
    <class>com.webx.escola.domain.Turma</class>
    <class>com.webx.escola.domain.TurmaId</class>
    <exclude-unlisted-classes>true</exclude-unlisted-classes>
    <properties>
      <property name="toplink.jdbc.user" value="root"/>
      <property name="toplink.jdbc.password" value="123"/>
      <property name="toplink.jdbc.url"
value="jdbc:mysql://localhost:3306/escola"/>
      <property name="toplink.jdbc.driver"
value="com.mysql.jdbc.Driver"/>
      <property name="toplink.logging.level" value="FINE"/>
    </properties>
  </persistence-unit>
</persistence>
```

**Figura 7: Exemplo de arquivo persistence.xml**

Na propriedade `toplink.jdbc.user` definimos o nome do usuário para acesso ao banco de dados. Na propriedade `toplink.jdbc.password` definimos a senha para acesso ao banco de dados. Na propriedade `toplink.jdbc.url` definimos a url, jdbc, de acesso ao banco de dados. Na propriedade `toplink.jdbc.driver` definimos o driver, necessário, para acesso ao banco de dados. Na propriedade `toplink.logging.level` definimos que tipo de log as consultas e acessos ao banco de dados devem gerar.

## 5. Provedor de Persistência - TopLink

O TopLink é um dos possíveis provedores de persistência para a JPA. Ele provê alta performance e escalabilidade, produtividade para o desenvolvedor e flexibilidade. O TopLink trabalha em conjunto com qualquer banco de dados, incluindo bancos de dados não-relacionais, e com qualquer servidor de aplicação.

O TopLink provê também recursos como:

- Um rico framework que provê opções de fácil uso para formular queries dinâmicas e sofisticadas. Onde os desenvolvedores podem definir queries usando qualquer expressão, como EJB QL, SQL e *Stored Procedures*.
- Um framework de transação que provê uma transação a nível de objetos. O framework de transação suporta gerenciamento direto das transações, e também gerenciamento externo das transações através de JTA/JTS.

- Um sistema avançado de cache que melhora a performance do sistema, ao não requerer que os dados mais recentemente ou mais frequentemente acessados sejam recuperados do banco de dados, mas sim da memória do servidor.

## 6. Referências Bibliográficas

- (JPA, 2009) Sun Microsystems. Java Persistence API. Disponível em <<http://java.sun.com/javaee/technologies/persistence.jsp>>. Acesso em: 30/10/2009.
- (EJB, 2009) Sun Microsystems. Enterprise JavaBeans Technology. Disponível em <<http://java.sun.com/products/ejb/>>. Acesso em: 30/10/2009.
- (Oracle, 2009) Oracle Technology Network. Oracle TopLink. Disponível em: <<http://www.oracle.com/technology/products/ias/toplink/index.html>>. Acesso em: 30/10/2009.
- (Oracle, 2009) Oracle Technology Network. Oracle TopLink – Anotações JPA. Disponível em: <<http://www.oracle.com/technology/products/ias/toplink/jpa/resources/toplink-jpa-annotations.html>>. Acesso em: 30/10/2009.
- (Oracle, 2009) Oracle Technology Network. Oracle TopLink. Disponível em: <[http://download.oracle.com/docs/cd/B31017\\_01/web.1013/b28221/cfgdepds005.html](http://download.oracle.com/docs/cd/B31017_01/web.1013/b28221/cfgdepds005.html)>. Acesso em: 30/10/2009.
- (Oracle, 2009) Oracle Technology Network. Oracle TopLink Overview. Disponível em: <<http://www.oracle.com/technology/products/ias/toplink/quicktour/feature.htm>>. Acesso em: 30/10/2009.
- (Hibernate, 2009) Hibernate. Relational Persistence for Java and .NET. Disponível em: <<https://www.hibernate.org/>>. Acesso em: 30/10/2009.



# Qualidade de Software

**André Mesquita Rincon**

Instituto de Informática/Universidade Federal de Goiás (UFG)  
Goiânia – GO – Brasil

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas/Fundação  
Universidade do Tocantins (UNITINS)  
Caixa Postal 173 – 77.020-122 – Palmas – TO – Brasil

andrerincon@inf.ufg.br

***Resumo.** Este artigo apresenta os conceitos básicos relacionados à qualidade do processo e de produtos de software utilizando, para isso, os modelos CMMI e MPS.BR e as normas ISO 12207, ISO 15504, ISO 9126 e ISO 14598. Além disso, este trabalho faz um breve comparativo entre algumas características dos softwares de “prateleira” e softwares “personalizados”.*

**Palavras chave:** Processo de software, CMMI, MPS.BR, qualidade de produto.

## 1. Introdução

Nas últimas décadas o software tem estado cada vez mais presente no cotidiano das pessoas. Ele é encontrado nos aparelhos de telefonia celular, nos fornos de microondas, nos carros, nas máquinas digitais e até nos porta-retratos. Atualmente, pode-se considerar que há uma dependência das características e serviços oferecidos por sistemas computadorizados. Em um estudo de REED (2000), encontra-se que se alguns sistemas de uso global deixarem de funcionar, aproximadamente 40% da população mundial sofrerão as conseqüências.

Em contrapartida à grande importância que o software tem tido no cenário mundial, estudos comprovam que a grande maioria dos projetos de software não atende aos objetivos traçados. De acordo com esses estudos, isso é decorrente da falta de processos adequados nas organizações em que eles são desenvolvidos.

Decorrente disso, desde o final das décadas de 80, passando por meados da década de 90 e chegando ao início deste século, os esforços para o desenvolvimento de modelos para maturidade dos processos de software tem tido uma constante evolução em todo o mundo. Principalmente através do SEI – *Software Engineering Institute* (SEI, 2009a) que publicou o SW-CMM – *Capability Maturity Model for Software* em 1987 e o CMMI – *Capability Maturity Model Integration* em 2002 e da ISO/IEC – *International Organization for Standardization / International Electrotechnical Commission* (ISO, 2009) que publicou as normas ISO/IEC 15504 e ISO/IEC 12207 nos anos 90.

Seguindo essa evolução, em maio de 2005 foi publicada a primeira versão do Guia Geral do programa para Melhoria de Processo do Software Brasileiro (MPS.BR) que consiste em uma iniciativa coordenada pela SOFTEX – Sociedade para Promoção da Excelência do Software Brasileiro (SOFTEX, 2009a) em parceria com universidades e grupos de pesquisa. Um dos principais objetivos do MPS.BR é criar e difundir os

modelos de qualidade para as empresas brasileiras e possibilitar que a grande massa de empresas (as de pequeno e médio porte) possa atingir maturidade em seus processos.

As iniciativas do SEI, da ISO e da SOFTEX em relação aos modelos de qualidade para o processo de software baseia-se na premissa de que um software terá qualidade, se o seu processo for bem realizado. Mas além dos modelos e normas direcionados ao processo de software, há também as normas ISO/IEC 9126 com um conjunto de características da qualidade de produto de software e a ISO/IEC 14598 que apresenta um processo de avaliação de produto de software.

Para ilustrar a relação entre os modelos e normas que abordam qualidade de software, a figura 1 apresenta um comparativo entre software de “prateleira” e software “personalizado”.

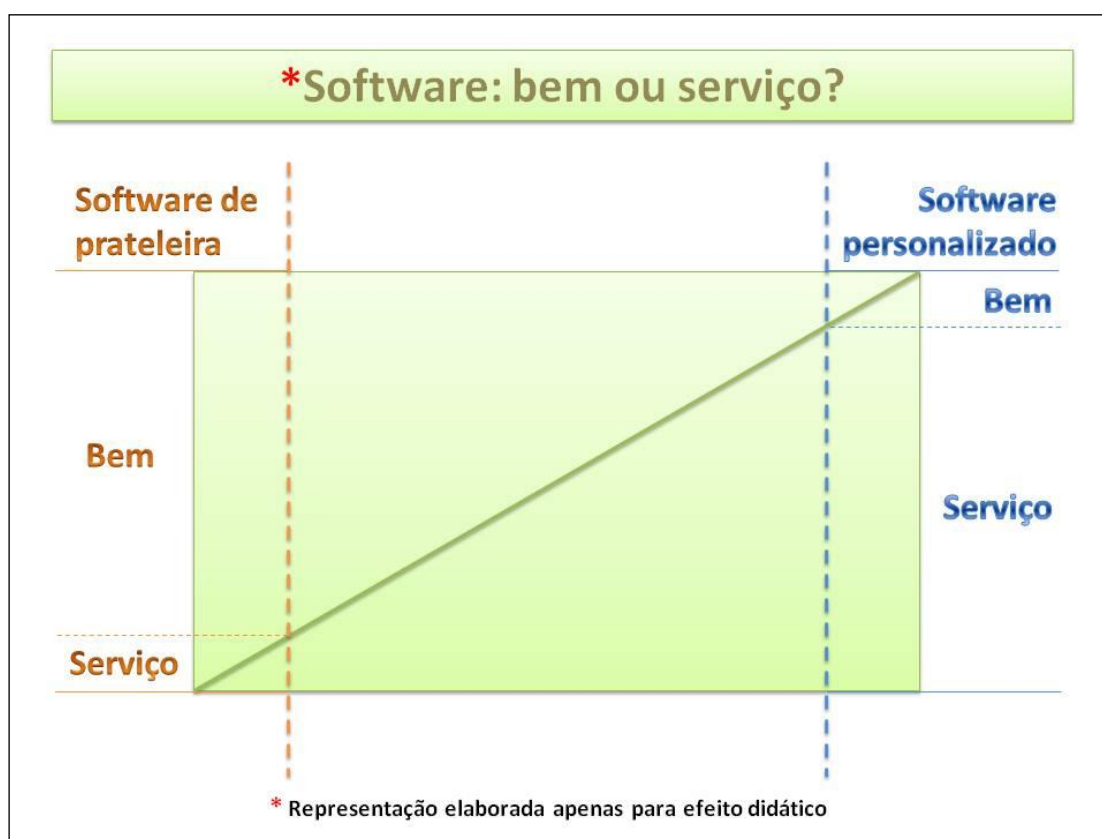


Figura 1. Software: bem ou serviço?

O título da figura 1 é considerado um antigo e importante questionamento da indústria de software: afinal, software é considerado um bem ou um serviço? Antes de dar início ao comparativo citado acima, vale ressaltar que esse artigo não pretende esgotar as discussões sobre o assunto ou sequer trazer uma resposta “fechada” para tal questão, mas apenas traçar uma relação entre essa natureza do software com alguns aspectos relacionados à qualidade.

No lado **esquerdo** da figura 1, tem-se a representação do software de prateleira. Considerando que esse tipo de software seja aproximadamente 90% um bem e apenas 10% um serviço, pois quando se adquire um produto dessa natureza, além do “pacote instalável” o cliente adquire também alguns serviços (como: instalação, suporte técnico, manutenções corretivas), pode-se sugerir ao cliente que ele se baseie nas normas ISO,

relacionadas à qualidade do produto (9126 e 14598), para nortear o processo de *validação* de um produto que está sendo adquirido.

Em contrapartida, no lado **direito** da figura 1, encontra-se a representação do software personalizado – aquele que é desenvolvido exclusivamente para um determinado cliente. Neste caso, a relação entre bem e serviço se inverte. Agora, esse software pode ser considerado 10% bem e 90% serviço. Ou seja, a preocupação com o processo de desenvolvimento do software deve passar a ser também do cliente, pois a qualidade do processo poderá ter um impacto na qualidade do produto. Exemplo dessa preocupação está em agências do governo e empresas que atuam com software de segurança crítica que exigem de seus fornecedores certificações em modelos de qualidade como o CMMI e MPS.BR. As seções seguintes apresentam detalhes sobre modelos e normas relacionados à qualidade de software.

## **2. Qualidade do processo de software**

Quando se fala em qualidade ou melhoria do processo de software, é inevitável a associação com o CMMI. No Brasil, esse fenômeno também tem ocorrido com outra sigla, o MPS.BR. No entanto, antes de apresentar os detalhes e comparações entre o CMMI e o MPS.BR, é importante citar as normas ISO/IEC 12207 e ISO/IEC 15504 que são as bases para criação desses modelos.

### **2.1. ISO 12207 e ISO 15504**

A Norma ISO/IEC 12207 - Processos de Ciclo de Vida de Software - tem como principal objetivo, o estabelecimento de uma estrutura comum para os processos de ciclo de vida de software como forma de ajudar as organizações a compreenderem todos os componentes presentes na aquisição e fornecimento de software e assim, conseguirem firmar contratos e executarem projetos de forma mais eficaz (Machado, 2006).

Ela foi publicada como norma internacional pela ISO em agosto de 1995 e em 1998 foi publicada a sua versão brasileira que tem o mesmo nome que a internacional, somente acrescida das iniciais NBR. Em 2002 e 2004, foram feitas atualizações na norma, em que foram inseridas algumas melhorias que geraram as Emendas 1 [ISO/IEC AMD 12207, 2002] e 2 [ISO/IEC AMD 12207, 2004] da ISO/IEC 12207. Já em 2008, ela passou por uma reformulação completa na sua estrutura.

Sobre a Norma ISO/IEC 15504, pode-se dizer informalmente, que ela é dividida em duas (Salviano, 2006):

Uma é a ISO/IEC 15504 que serve como um framework para modelos e métodos de avaliação (e melhoria) de processo. Um usuário típico desse framework é um grupo que queira desenvolver um modelo de capacidade de processo, um método de avaliação de processo e/ou uma abordagem para melhoria de processo. Exemplos de usuários desse framework são o SEI, que desenvolve os modelos de capacidade do CMMI e os métodos de avaliação SCAMPI (SEI, 2009c) e o MPS.BR, que desenvolve o modelo de capacidade MR-MPS e o método de avaliação MA-MPS (SOFTEX, 2009c).

A outra 15504 é o modelo ISO/IEC 15504-5, que é um modelo de capacidade de processo (definido segundo o framework da ISO/IEC 15504) para a engenharia de

software, baseado nos processos definidos na ISO/IEC 12207. Este modelo 15504-5 tem como usuário típico uma organização intensiva em software que queira realizar um programa de melhoria de processo de software e/ou avaliar a capacidade de seus processos de software.

## 2.1. O CMMI

Em março de 2002 ocorreu a publicação do CMMI, sigla que, em português, significa Modelo Integrado de Maturidade da Capacidade. No contexto de software, a maturidade da capacidade está relacionada ao processo de desenvolvimento do software. Já a palavra “integrado” está relacionada ao fato do CMMI ter a finalidade de integrar os modelos do antigo CMM em uma estrutura coerente e alinhada.

O modelo CMM foi desenvolvido pelo SEI que consiste em uma organização ligada à universidade *Carnegie Mellon*. Segundo Pessoa (2005), o desenvolvimento desse modelo foi financiado pelo Departamento de Defesa Americano, com o objetivo de se estabelecer um padrão de qualidade para software desenvolvido para as forças armadas.

Com o tempo, foram publicados outros modelos com finalidades específicas como engenharia de sistemas e aquisição. No entanto, quando eles eram aplicados em conjunto em uma mesma organização, encontravam-se algumas inconsistências, por isso, houve a necessidade de se integrar os modelos em uma versão consistente. Em 2006, foi lançada a nova versão do CMMI, conhecida como CMMI 1.2 que não apresenta mudanças radicais, mas sim pequenos ajustes e melhorias oriundos de várias sugestões de mudanças da comunidade.

### 2.1.1. Representações do modelo

A versão 1.2 do CMMI manteve a representação denominada **contínua**. Essa representação foi considerada uma novidade que a primeira versão do CMMI trouxe em relação ao CMM, pois na versão não integrada, existia apenas a representação tradicional por **estágios**. Segundo Pessoa (2005), a representação contínua foi incorporada para que o modelo se tornasse compatível à Norma ISO 15504. A figura 2 ilustra a representação contínua e a por estágios.

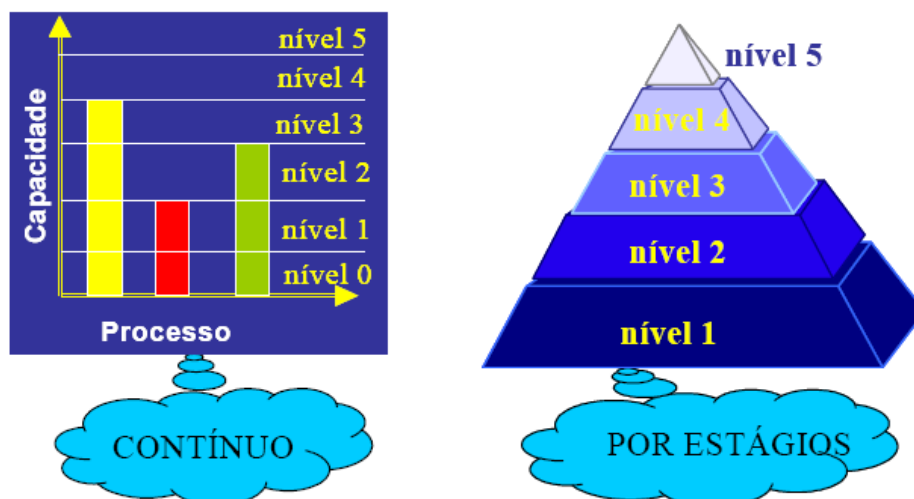


Figura 2. As duas representações do CMMI (Pessoa, 2005)

O CMMI por estágios não apresenta mudanças em relação ao CMM. Ou seja, a idéia é ir implementando os processos conforme a seqüência dada pelo modelo CMMI-DEV (SEI, 2009b). Já a representação contínua possibilita que uma organização tenha sua capacidade avaliada por processos e que cada um tenha níveis diferentes de capacidade.

Segundo Pêsoa (2005), para entender o significado do termo capacidade pode-se pensar na capacidade que alguém possui para realizar um trabalho: quanto maior a capacidade do processo, melhor deve ser o resultado obtido. Quando uma organização é avaliada levando-se em consideração a capacidades de seu processo, o resultado será seu perfil de capacidade conforme representado na figura 3.

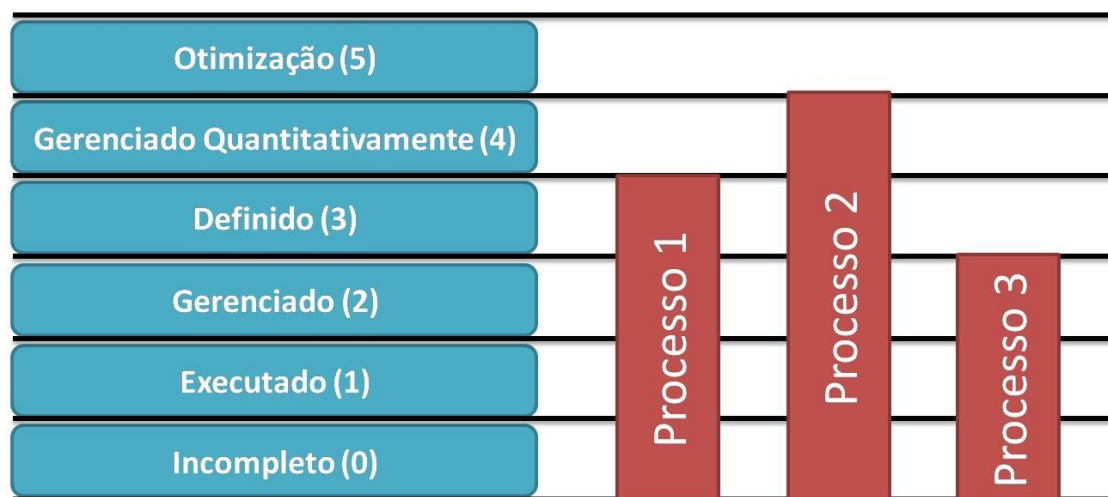


Figura 3. Exemplo de perfil de capacidade de uma organização

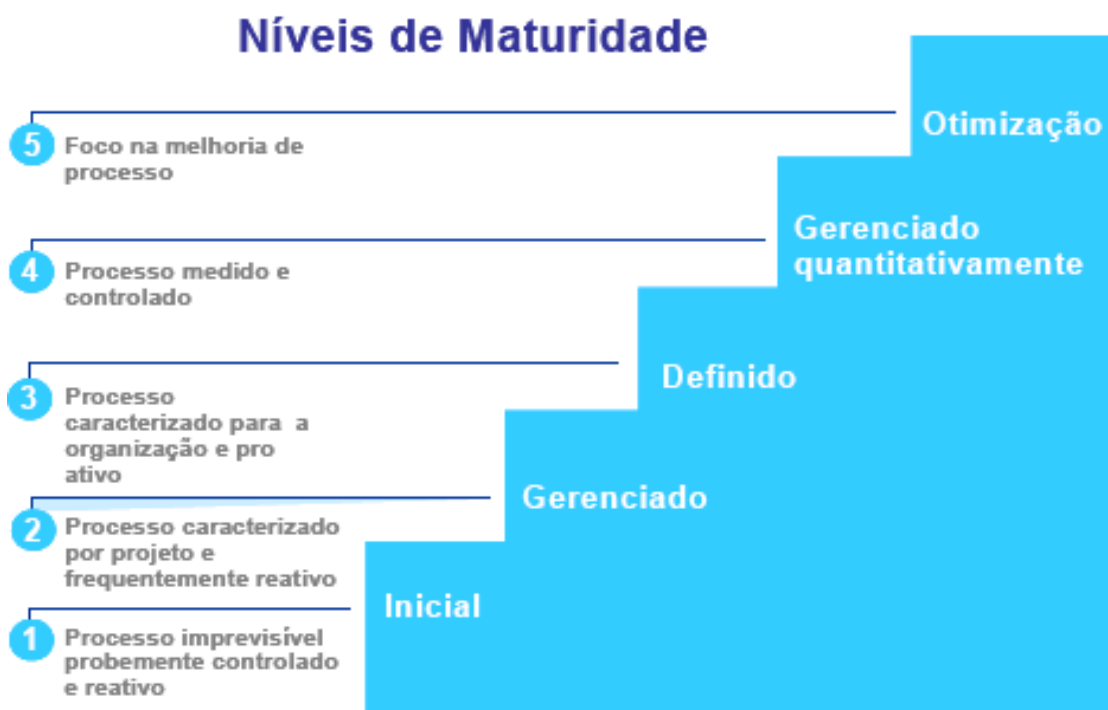
No exemplo acima, o perfil da capacidade da organização foi avaliado e o resultado foi que o processo 1 está no nível 3, o processo 2 no nível 4 e processo 3 no nível 2. Como se pode notar na figura 3, os níveis de capacidade do processo variam entre 0 a 5. Em Pêsoa (2005), encontra-se a descrição sobre o que representa cada nível no modelo contínuo, são eles:

- Nível 0 – incompleto: o processo não é executado ou é executado parcialmente;
- Nível 1 – executado: o processo satisfaz as metas específicas da área de processo;
- Nível 2 – gerenciado: o processo é executado e também planejado, monitorado e controlado para atingir um objetivo (em projetos individuais, grupos ou processos isolados);
- Nível 3 – definido: o processo é gerenciado, adaptado de um conjunto de processos padrão da organização;
- Nível 4 – gerenciado quantitativamente: o processo é definido, controlado utilizando estatística ou outras técnicas quantitativas;
- Nível 5 – otimização: o processo é gerenciado quantitativamente para a melhoria contínua do desempenho do processo.

Conforme foi citado anteriormente, no modelo por estágios uma organização deve implementar os processos de acordo com uma seqüência pré-estabelecida. Neste

caso, diferentemente da representação contínua, a organização é avaliada por um conjunto de processos de acordo com o nível que ela estiver pleiteando. Os níveis são denominados níveis de maturidade e variam de 1 a 5.

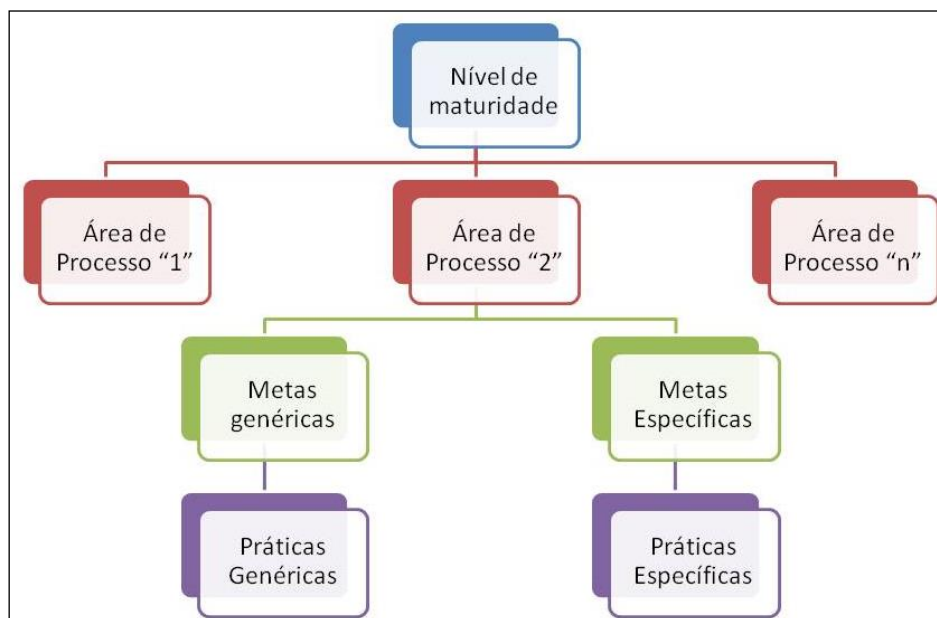
Pêsoa (2005) sugere que, para entender o significado do termo maturidade pode-se pensar na maturidade de um profissional à medida que evolui em sua carreira: o aprendizado teórico, aliado a vivências práticas em diversos projetos e diversas situações, cria uma experiência e conhecimento para esse profissional que o tornam maduro, analogamente, o mesmo foi concebido no modelo de maturidade de software com a diferença que a maturidade é da organização. A figura 4 representa os níveis de maturidade do CMMI.



**Figura 4. Os níveis de maturidade do CMMI (Pêsoa, 2005)**

Conforme ilustra a figura 4, o CMMI, no modelo por estágios, é composto por 5 níveis de maturidade. Do ponto de vista do CMMI, pode-se considerar que toda empresa que ainda não passou por uma implementação de melhoria de processos, está no nível 1. Sendo assim, para se tornar mais madura em relação ao seu processo de desenvolvimento, primeiramente ela deve implementar os processos do nível 2 e solicitar uma avaliação por uma instituição credenciada. Ela deverá realizar esse procedimento até que obtenha o nível 5, que é o mais alto na escala do CMMI.

Cada nível está dividido em Áreas de Processo. A figura 5 ilustra a estrutura que cada Área de Processo possui dentro do documento que descreve o modelo CMMI-DEV.



**Figura 5. Estrutura das Áreas de Processos do CMMI**

Conforme citado anteriormente, para que uma organização possa evoluir de um nível para o outro no CMMI, ela deve implementar um conjunto de processos pré-definidos. Para isso, ela deve cumprir o que é descrito nas metas e práticas genéricas e nas metas e práticas específicas de cada Área de Processo. O documento oficial do modelo, o CMMI-DEV, traz detalhadamente quais são as metas e práticas específicas de cada Área de Processo.

Em uma avaliação do CMMI, que utiliza o modelo SCAMPI, os avaliadores irão buscar Evidências Objetivas de que cada meta e prática (genérica e específica) estão implantadas na organização. Sendo assim, para implementá-las com sucesso, uma organização deve ter conhecimento suficiente sobre o que é “cobrado”, em termos de evidências objetivas, para que uma empresa possa obter o nível pleiteado. Geralmente as organizações contratam consultorias especializadas no assunto, pois elas terão mais experiências sobre quais evidências devem ser geradas para que se possa cumprir o que é solicitado em cada Área de Processo.

## **2.2. O MPS.BR**

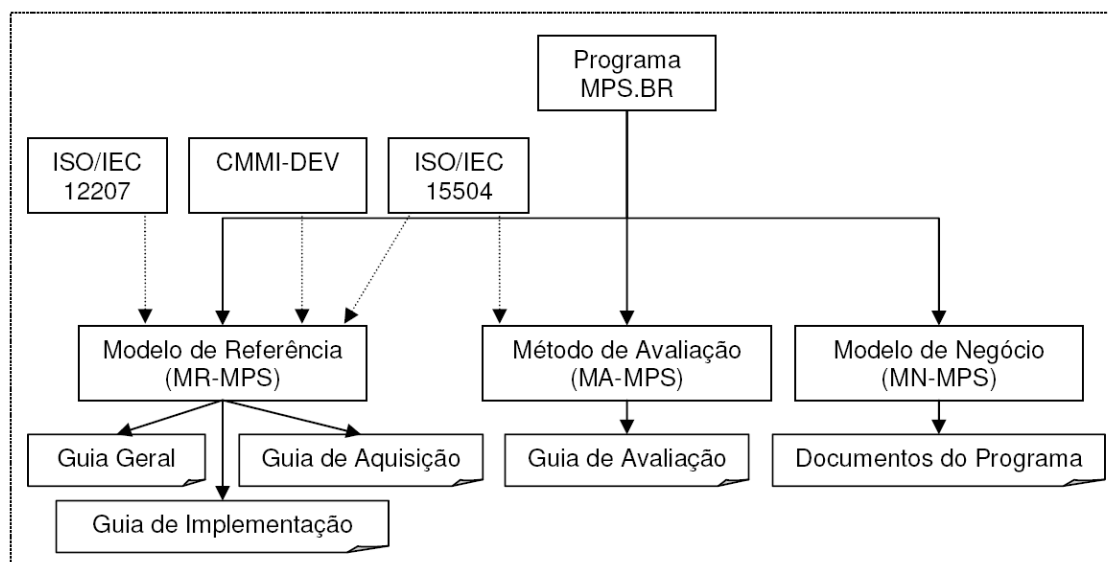
Em maio de 2005 foi publicada a primeira versão do Guia Geral do programa para Melhoria de Processo do Software Brasileiro (MPS.BR) que apresentava a descrição geral do MPS.BR e detalhava o Modelo de Referência (MR-MPS). Um ano depois, em maio de 2006, uma nova versão (a 1.1) trazia algumas modificações. Em 2007, a versão 1.2 acrescentou os processos relacionados à reutilização.

A versão atual do modelo está na sua versão 2009. Além da mudança na forma da nomenclatura (1.1, 1.2 e agora 2009), ela acrescentou alguns processos (como, por exemplo, Gerência de Portfólio de Projetos no nível B) e excluiu outros (como, por exemplo, Análise de Causas de Problemas e Resolução no nível A).

Segundo SOFTEX (2009b), as iniciativas deste programa buscam que ele seja adequado ao perfil de empresas com diferentes tamanhos e características, públicas e privadas, embora com especial atenção às micro, pequenas e médias empresas. Também

se espera que o MPS.BR seja compatível com os padrões de qualidade aceitos internacionalmente e que tenha como pressuposto o aproveitamento de toda a competência existente nos padrões e modelos de melhoria de processo já disponíveis.

O MPS.BR está estruturado em três componentes: Modelo de Referência (MR-MPS), Método de Avaliação (MA-MPS) e Modelo de Negócio (MN-MPS) e para sua descrição utilizam-se documentos em formato de guias. A figura 6 apresenta os componentes do MPS.BR.



**Figura 6. Componentes do MPS.BR. Fonte: SOFTEX (2009b)**

A figura 6 apresenta os componentes do MPS.BR e os documentos utilizados para a descrição de cada um deles. O MR-MPS contém os requisitos que os processos das unidades organizacionais devem atender para estar em conformidade com o MR-MPS. Ele contém as definições dos níveis de maturidade, processos e atributos do processo que estão descritos no Guia Geral (SOFTEX, 2009b).

Além do Guia Geral, o MR-MPS ainda possui outros dois guias: o de aquisição e o de implementação que, de acordo com SOFTEX (2009b), são, respectivamente: um documento complementar destinado a organizações que pretendam adquirir software e serviços correlatos; e um guia, composto de sete partes, que descreve como implementar cada um dos níveis do MR-MPS.

Já o Guia de Avaliação contém o processo e o método de avaliação (MA-MPS), os requisitos para os avaliadores líderes, avaliadores adjuntos e Instituições Avaliadoras (IA) (SOFTEX, 2009b).

Por último, o Modelo de Negócio (MN-MPS) descreve regras de negócio para implementação do MR-MPS pelas Instituições Implementadoras (II), avaliação seguindo o MA-MPS pelas Instituições Avaliadoras (IA), organização de grupos de empresas para implementação do MR-MPS e avaliação MA-MPS pelas Instituições Organizadoras de Grupos de Empresas (IOGE), certificação de consultores de aquisição e programas anuais de treinamento por meio de cursos, provas e workshops MPS.BR (SOFTEX, 2009b).

O MPS.BR não possui a perspectiva continuada do CMMI e da ISO 15504, apenas a versão estagiada. Ele é composto por sete níveis que vão do G ao A e possuem



correspondência com a versão por estágios do CMMI. A tabela 1 representa os níveis de maturidade e as áreas de processo do MPS.BR e sua equivalência com os níveis e áreas de processo do CMMI.

**Tabela 1 – Níveis de maturidade do MPS.BR e sua equivalência com as áreas de processo do CMMI**

CMMI		MPS.BR	
Inovação Organizacional	5	A	“Atributos do processo”
Análise de Causa e Solução de Problemas			
Desempenho do Processo Organizacional	4	B	---
Gestão Quantitativa de Projeto			Gerência de Projetos (evolução)
Gestão de Risco	3	C	Gerência de Riscos
---			Desenvolvimento para Reutilização
Análise de Decisão			Gerência de Decisões
Verificação		D	Verificação
Validação			Validação
Solução Técnica			Projeto e Construção do Produto
Integração de Produto			Integração do Produto
Desenvolvimento de Requisitos			Desenvolvimento de Requisitos
Gestão Integrada de Projeto			Gerência de Projetos (evolução)
---			Gerência de Reutilização
Treinamento Organizacional	E	Gerência de Recursos Humanos	
Definição do Processo Organizacional		Definição do Processo Organizacional	
Foco no Processo Organizacional		Aval. e Melhoria do Processo Organizacional	
Medição e Análise	2	F	Medição
Garantia da Qualidade de processo e Produto			Garantia da Qualidade
Monitoramento e Controle de Projeto			Gerência de Portfólio de Projetos
Gestão de Configuração			Gerência de Configuração
Gestão de Contrato com Fornecedor		Aquisição	
Gestão de Requisitos		G	Gerência de Requisitos
Planejamento de Projeto			Gerência de Projetos

Na tabela 1, é possível notar que o MPS.BR possui uma divisão dos processos de forma diferente do CMMI. Segundo o SOFTEX (2009b), o fato de o MR-MPS possuir mais níveis em relação ao CMMI é justamente para possibilitar uma

implementação mais gradual e mais acessível financeiramente para as pequenas e médias empresas. Note que nem todas as áreas de processo têm correspondentes nos dois modelos e, em alguns casos, as nomenclaturas não são iguais, no entanto, quando uma organização atinge um nível do MPS.BR, ela está apta a atingir o nível correspondente no CMMI (por exemplo, uma organização que é MPS.BR nível F, teoricamente também está apta a se tornar CMMI nível 2).

Sobre a estrutura do MPS.BR, pode-se dizer, a grosso modo que os atributos do processo são como as metas e práticas genéricas do CMMI e que os resultados esperados de cada processo são as metas e práticas específicas. Vale ressaltar que tal comparação é apenas para efeitos didáticos de comparação entre os modelos.

### 3. Qualidade do produto

Em Villas Boas (2007), encontra-se que a avaliação de produto de software tem sido uma das formas empregadas por organizações que produzem ou adquirem software para obtenção de maior qualidade nestes produtos, sejam eles produtos completos ou partes a serem integradas num sistema computacional mais amplo. Segundo ele, para que a avaliação seja mais efetiva é importante que se utilize um modelo de qualidade que permita estabelecer e avaliar **requisitos de qualidade** (ISO 9126) e também que o **processo de avaliação** (ISO 14598) seja bem definido e estruturado. A figura 7 ilustra o processo de avaliação da ISO 14598.

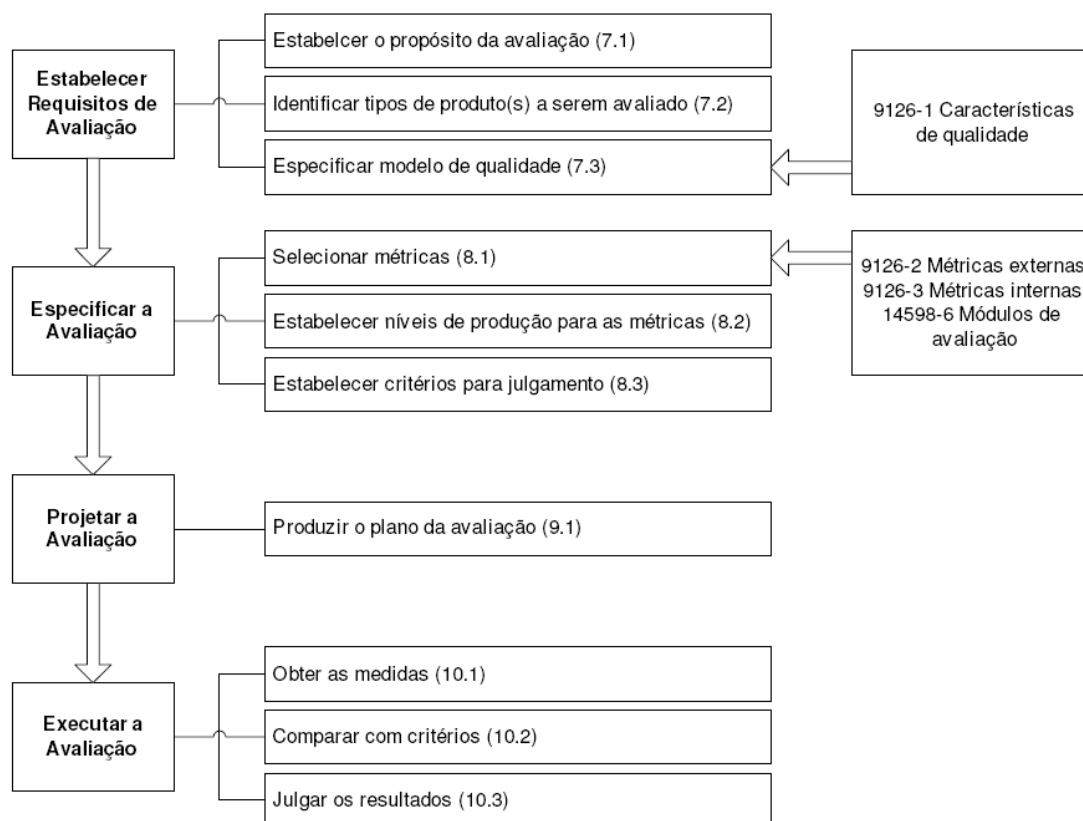
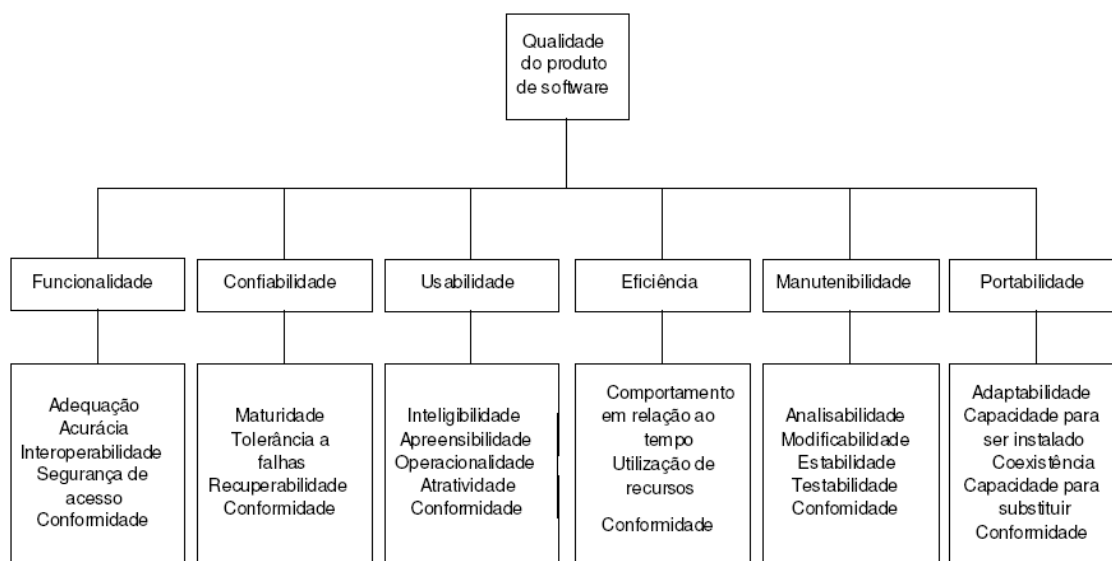


Figura 7. Processo de Avaliação da ISO 14598 (Villas Boas, 2007)

A norma ISO/IEC 14598 apresenta detalhes do processo de avaliação de produto de software. Na norma estão as informações necessárias para que se possa realizar uma avaliação utilizando-se do processo apresentado na figura 7. Pode-se notar que a norma

ISO/IEC 9126 é utilizada dentro do processo de avaliação fornecendo as características de qualidade e métricas para avaliação. A figura 8 apresenta as características de qualidade da ISO/IEC 9126.



**Figura 8. Características de qualidade da ISO 9126 (Villas Boas, 2007)**

O modelo de qualidade nesta parte da série ISO/IEC 9126 categoriza os atributos de qualidade de software em seis características (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade) as quais são, por sua vez, subdivididas em subcaracterísticas. As subcaracterísticas podem ser medidas por meio de métricas internas e externas. Cada característica e subcaracterística do software, que influenciam a característica de qualidade, possui uma definição. Para cada característica e subcaracterística, a capacidade do software é determinada por um conjunto de atributos internos que podem ser medidos (Villas Boas, 2007).

#### 4. Considerações finais

A separação entre qualidade do processo e qualidade do produto, como apresentada neste artigo, geralmente é utilizada como recurso didático para apresentação de modelos e normas, pois na prática, é um tanto quanto complicado fazer tal separação. Na implantação de processos de qualidade, uma organização, provavelmente, irá estudar normas as relacionadas à qualidade do produto para instituir seus processos de VV&T (verificação, validação e teste).

A principal contribuição deste trabalho está em apresentar conceitos iniciais de modelos de normas relacionados à qualidade de forma que o leitor possa ter a base para aprofundar seus estudos sobre o assunto. Além disso, vale reforçar um ponto que foi citado durante o trabalho: a contratação de consultoria especializada para implantação de processos de software em uma organização que visa certificação em modelos como CMMI ou MPS.BR pode minimizar consideravelmente o tempo/custo dessa implantação.

#### 5. Referências Bibliográficas

(ISO, 2009) ISO, International Organization for Standardization – Site Oficial. Disponível em <<http://www.iso.org/iso/home.htm>>. Acesso em setembro de 2009.

- (Machado, 2006) Machado, Cristina Ângela Filipak. Definindo processos do ciclo de vida de software usando a norma NBR ISO/IEC 12207 e suas ementas 1 e 2. Lavras: UFLA/FAEPE, 2006.
- (Pêsoa, 2005) Pêsoa, Marcelo Schneck de Paula. Modelo Integrado de Maturidade da Capacidade de Processo. Lavras: UFLA/FAEPE, 2005.
- (REED, 2000) Reed, K. Software engineering – a new millennium? IEEE Software, jul.-ago. 2000.
- (Salviano, 2006) Salviano, Clênio Figueiredo. Melhoria e Avaliação de Processo de Software com o Modelo ISO/IEC 15504-5:2006. Lavras: UFLA/FAEPE, 2006.
- (SEI, 2009a) SEI, Software Engineering Institute – Site Oficial. Disponível em <<http://www.sei.cmu.edu/>>. Acesso em setembro de 2009.
- (SEI, 2009b) SEI, Software Engineering Institute – CMMI for Development (CMMI-DEV). Disponível em <<http://www.sei.cmu.edu/cmmi/tools/dev/>>. Acesso em setembro de 2009.
- (SEI, 2009c) SEI, Software Engineering Institute – Standard CMMI Appraisal Method for Process Improvement (SCAMPI), Version 1.1: Method Definition Document. Disponível em <<http://www.sei.cmu.edu/library/abstracts/reports/01hb001.cfm>>. Acesso em setembro de 2009.
- (SOFTEX, 2009a) SOFTEX, Sociedade para a Promoção da Excelência do Software Brasileiro – Site Oficial. Disponível em <<http://www.softex.br/>>. Acesso em setembro de 2009.
- (SOFTEX, 2009b) SOFTEX, Sociedade para a Promoção da Excelência do Software Brasileiro – MPS.BR – Melhoria de Processo do Software Brasileiro: Guia Geral versão 2009. Disponível em <[http://www.softex.br/portal/mpsbr/\\_guias/](http://www.softex.br/portal/mpsbr/_guias/)>. Acesso em setembro de 2009.
- (SOFTEX, 2009c) SOFTEX, Sociedade para a Promoção da Excelência do Software Brasileiro – MPS.BR – Melhoria de Processo do Software Brasileiro: Guia de Avaliação versão 2009. Disponível em <[http://www.softex.br/portal/mpsbr/\\_guias/](http://www.softex.br/portal/mpsbr/_guias/)>. Acesso em setembro de 2009.
- (Villas Boas, 2007) Villas Boas, André Luiz de Castro. Qualidade e Avaliação de Produto de Software. Lavras: UFLA/FAEPE, 2007.

# VPN – *Virtual Private Network*

Andrew Silva de Lima, Madianita Bogo, Ricardo Marx Costa Soares de Jesus

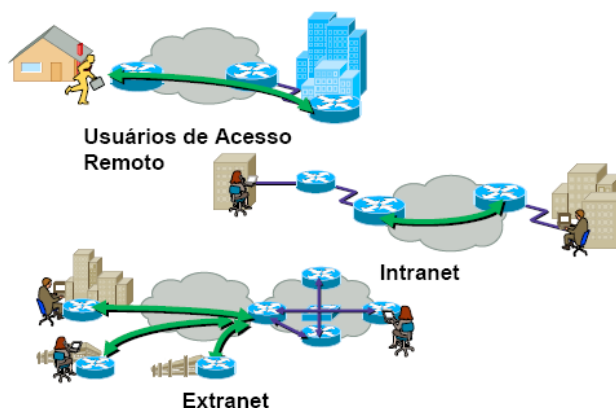
Cursos de Sistemas de Informação e de Redes de Computadores  
Centro Universitário Luterano de Palmas (CEULP)  
Caixa Postal 160 – 77054-970 – Palmas – TO – Brasil  
{andrew, madia, rmarx}@ulbra-to.br

**Resumo.** Uma VPN consiste em construir uma rede privada sobre uma rede pública, como a Internet, em que os pontos finais se comunicam através de túneis. Como os dados trafegam por uma rede pública aberta, é importante utilizar meios para garantir a segurança na comunicação, como, autenticação e criptografia. Os protocolos IPSec(*Internet Protocol Security*) e SSL(*Secure Sockets Layer*) são muito utilizados para garantir segurança na comunicação de uma VPN.

## 1. VPN – *Virtual Private Network*

Northcutt (2002) conceitua VPN como “uma sessão de rede protegida formada através de canais desprotegidos, como a Internet”. Desta forma, é possível construir uma rede com módulos processadores situados em pontos distintos, ligados por uma conexão lógica privativa, chamada de túnel. Esse túnel é construído sobre uma rede pública sem a necessidade de utilizar *links* dedicados, o que reduz os custos na comunicação.

Existem várias formas de configurar uma VPN. A figura 1 apresenta os principais tipos, que são: Intranet VPN, extranet VPN e VPN para Acesso Remoto (ABREU, 2006).



**Figura 1.** Tipos de VPN, retirada de (MORAES, 2004)

As VPNs para Acesso Remoto permitem que os usuários se conectem de forma segura às redes corporativas mesmo estando em um local fisicamente distante, por exemplo, funcionários externos de uma empresa podem acessar a mesma com um *notebook*. A Intranet VPN tem o objetivo de interligar redes locais de uma mesma

empresa, por exemplo, ligar a rede da matriz de uma empresa às redes de suas filias. A Extranet VPN é utilizada interligar a rede de uma empresa às redes de parceiros como, por exemplo, fornecedores e clientes, ou seja, interligar várias redes privadas com soluções distintas.

Dentre os cuidados que devem ser considerados ao configurar uma VPN, alguns específicos para cada tipo, o principal é a garantia de segurança. Isso se deve ao fato dos dados trafegarem por uma rede pública aberta, geralmente a Internet. Assim, qualquer VPN deve conter (CHIN, 2006):

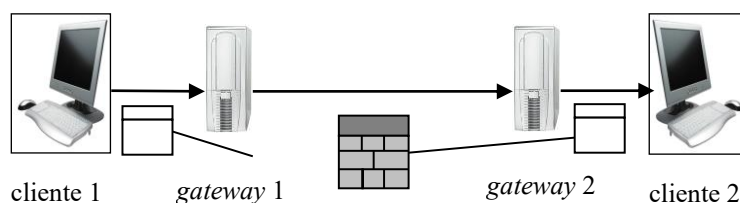
- Autenticação: consiste em identificar os agentes da comunicação, para que o acesso seja restrito às pessoas autorizadas.
- Criptografia: consiste em codificar os dados, para que as informações não possam ser lidas, caso sejam interceptadas por pessoas não autorizadas.
- Tunelamento: consiste em encapsular os dados para trafegar sobre a rede pública, sendo que os dados devem ser codificados antes do encapsulamento.

Existem várias formas de garantir a segurança na comunicação entre os pontos finais de uma VPN. Dois protocolos muito utilizados para esse fim são o IPSec e o SSL.

## 2. IPSec

O objetivo do protocolo IPSec é oferecer serviços de segurança a nível de camada de rede. Desse modo, todas as aplicações de rede situadas em camadas de nível superior podem utilizar a segurança IPSec sem qualquer necessidade de alteração (STALLINGS, 1999). Basicamente, os serviços oferecidos são integridade, autenticação e confidencialidade, os quais são disponibilizados em dois modos de operação: modo transporte e modo túnel. Dentre esses modos, o ideal para a implementação de VPNs é o modo túnel, pois provê segurança ao pacote IP por completo (NORTHCUTT, 2002).

O nível de segurança dada ao pacote IP é configurável, conforme os serviços que serão oferecidos. Para VPNs, convencionalmente utiliza-se IPSec em modo túnel com confidencialidade ou com confidencialidade e autenticação. O arranjo mais comum para o estabelecimento de VPNs via IPSec consiste no tunelamento de gateway para gateway, o qual ocorre de maneira transparente para os clientes VPN, pois apenas o gateways necessitam estar configurados (NORTHCUTT, 2002). A figura 2 ilustra o funcionamento do modo túnel com confidencialidade.



**Figura 2.** Ilustração do modo túnel com confidencialidade.

Na figura 2 o cliente 1 comunica-se com o cliente 2 enviando os pacotes IP para o gateway 1. Ao receber um pacote, o gateway 1 codifica e coloca esse pacote na área de dados em um novo pacote IP, que também possui um novo cabeçalho. Este novo pacote é enviado ao gateway 2, que realiza o trabalho inverso, isto é, decodifica o pacote IP original e encaminha-o ao cliente 2. Ainda que um atacante intercepte o canal de comunicação entre os gateways 1 e 2, não será possível descobrir o conteúdo dos pacotes.

### **3. SSL**

Diferentemente do IPSec, o SSL é um protocolo que provê segurança a nível de camada de transporte. O SSL foi desenvolvido para fornecer conexões TCP seguras a protocolos de camadas superiores, oferecendo os serviços de integridade, autenticação e confidencialidade (STALLINGS, 1999). Assim, qualquer protocolo de camada superior pode ser executado sobre SSL, como, por exemplo: HTTPS (HTTP sobre SSL), SMTPS (SMTP sobre SSL) e POP3S (POP3 sobre SSL).

A estrutura do SSL permite sua aplicação também para a construção de VPNs, embora não tenha sido originalmente desenvolvido para esse fim. Para tanto é necessária a existência de clientes e servidores SSL destinados a tal tarefa. Nessas VPNs o tráfego é criptograficamente codificado e redirecionado para uma porta que faz o encaminhamento do mesmo ao destino, constituindo o túnel SSL (NORTHCUTT, 2002).

### **6. Referências Bibliográficas**

Abreu, Luiz Henrique. “Arquitetura MPLS para Formação de VPN.” Disponível em <http://www.si.uniminas.br/TFC/monografias/Monografia%20Luiz%20Henrique%20de%20abreu.pdf>. Acessada em novembro de 2006.

Chin, Liou Kuo. “Rede Privada Virtual – VPN”. Disponível em <http://www.rnp.br/newsgen/9811/vpn.html>. Acessado em novembro de 2006.

Moraes, Alexandre F. “Reduzindo Custos de Telecom com VPNs”. Disponível em [www.sucesues.org.br/eventos/127/Reduzindo\\_custos\\_de\\_Telecom\\_com\\_VPNs\\_com\\_Alexandre\\_Moraes\\_15\\_03\\_2004.pd](http://www.sucesues.org.br/eventos/127/Reduzindo_custos_de_Telecom_com_VPNs_com_Alexandre_Moraes_15_03_2004.pd). Acessado em novembro de 2006. Publicado em março de 2004.

Northcutt, Stephen et. al. Desvendando Segurança em Redes. Rio de Janeiro: Campus, 2002.

Stallings, William. “Cryptography and networks security: principles and practice”. 2.ed. New Jersey: Prentice Hall, 1999.





## **ARTIGOS**



# Comparativo entre protocolos de gerenciamento de mobilidade

**HELDER CLEBER ALMEIDA PEREIRA**

Mestrando em Engenharia Elétrica da UNB (Universidade de Brasília)

Professor do Instituto Federal de Tecnologia do Tocantins – Palmas

*helder@etfto.gov.br*

**Resumo:** *MIP (Mobile Internet Protocol), mSCTP (Mobile Stream Control Transmission Protocol) e SIP (Session Internet Protocol) são os protocolos com maior tendência a serem utilizados para prover a mobilidade de aplicações dos usuários em um ambiente pulverizado de diferentes tecnologias de comunicação (celular, IEEE 802.11, WiMax e etc). Neste artigo, serão apresentados os aspectos positivos e negativos desses protocolos de gerenciamento de mobilidade.*

**Palavras chave:** protocolo de gerenciamento de mobilidade, handover/handoff.

**Introdução:** Com a popularização das redes wireless de variadas tecnologias (IEEE 802.11, celular, WiMax) e com a tendência de prover acesso à rede e serviços a qualquer hora, em qualquer lugar possibilitando a movimentação de terminais/laptops/palmtops ou outros dispositivos do gênero sem interrupção do serviço/sessão estabelecida é o grande desafio deste e dos próximos anos. O uso de uma rede totalmente IP é uma tendência forte para prover o uso desses diferentes meios de transmissão de maneira transparente para as aplicações, utilizando como base o padrão TCP/IP como meio padronizador. Para tanto existem disponibilizadas, de acordo com esse padrão, o gerenciamento de mobilidade da camada de rede (MIPv4 – Mobile Internet Protocol), o gerenciamento de mobilidade da camada de transporte (mSCTP – Mobile Stream Control Transmission Protocol) e o gerenciamento de mobilidade da camada de aplicação (SIP – Session Internet Protocol). Esses protocolos de gerenciamento de mobilidade serão comparados neste artigo seus aspectos positivos e negativos e sua adequação para as necessidades atuais.

## 1. Gerenciamento de Mobilidade MIP

No cenário apresentado através da figura 1, existe o MN (Mobile Node) que atua como o cliente que irá fazer uma chamada ao CN (Correspondente Node). Para cada Nodo existe dois tipos de endereço, o HoA(Home of Address) e o CoA(Care of Address), sendo que o CoA é utilizado quando o Nodo se encontra numa rede estrangeira.

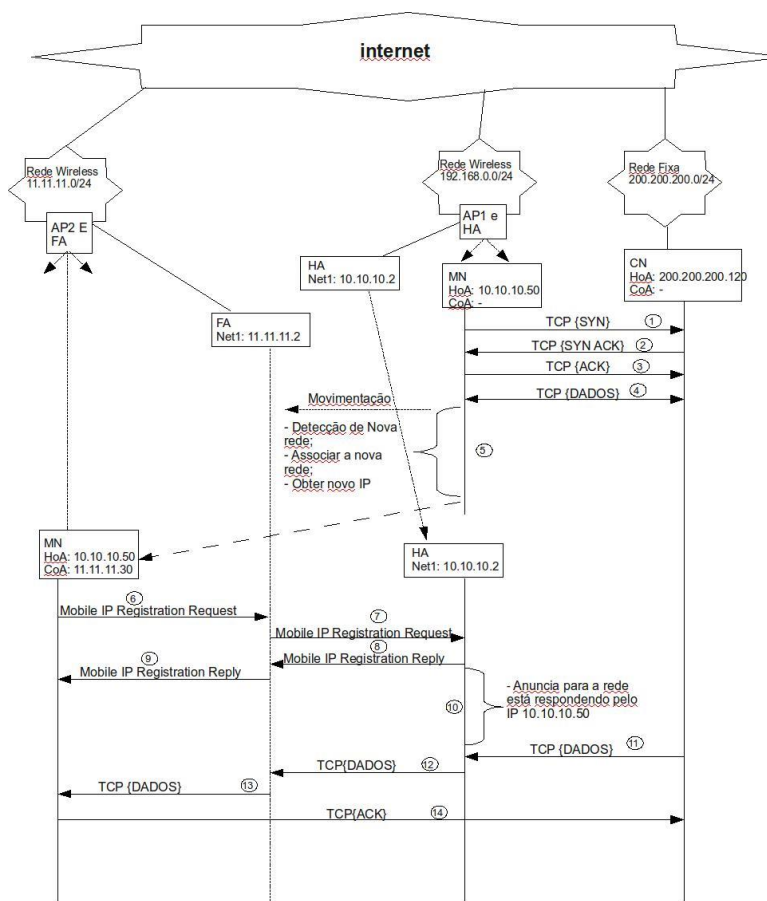
Para compor o cenário ainda temos o AP1 e AP2 que são Access Points pertencente a duas sub-redes diferentes. Em cada sub-rede existe um HA(Home Agent) e FA(Foreign Agent) que funcionam também como servidores de atribuição de endereços.

Quando a conexão TCP é aberta do MN para o CN é iniciada a transferência de dados, normalmente como ocorreria em um ambiente sem o MIP. Logo depois o MN inicia um movimento em direção ao AP2.

A partir de um determinado momento que o MN permanece na área de influência do AP2 em algum instante irá ocorrer o handoff do MN do AP1 para o AP2, nesse meio tempo deverá ocorrer a detecção, associação ao AP2 e obtenção do novo endereço IP. Esse novo endereço IP obtido pelo MN passará a representar o CoA (Care of Address), e uma vez obtido esse novo IP o MN o registrará em seu HA para que possa dar continuidade à sessão existente, no nosso caso, uma conexão TCP.

Quando houver ocorrido o cadastramento do novo endereço IP(11.11.11.30), pelo qual o MN responde no momento, o HA por sua vez anunciará em sua rede que a partir deste momento estará respondendo de agora em diante pelo endereço IP de registro do MN(no caso o IP 10.10.10.50). Na prática toda mensagem endereçada para o MN(10.10.10.50) será capturado pelo HA e tunelado ao FA para que possa ser entregue ao CoA do MN. Existe um mecanismo para contornar esse roteamento triangular denominado otimização de roteamento mas para funcionar necessita alterar a pilha IP do CN.

A partir das fases acima, toda mensagem enviada ao MN para o CN é feita de maneira direta sem nenhuma interferência do MIP, já as mensagens oriundas do CN devem ser tuneladas do HA via FA para que possam chegar ao MN.



**Figura1 – Gerenciamento de mobilidade utilizando o MIP**

## **1.2 - Vantagens e Desvantagens do Uso de Gerenciamento de Mobilidade com MIP**

Vantagens elencadas pelo uso do MIP para prover mobilidade:

- Consegue prover mobilidade tanto em aplicações com conexões TCP como UDP;
- Otimização de roteamento já implementado pelo MIP;
- Possui gerenciador de localização;

Desvantagens elencadas pelo uso do MIP para prover mobilidade:

- Como não é considerado um protocolo de uso geral dificilmente se encontrará instalado em terminais além de requerer uma estrutura de rede pré -instalada;
- Existem duas implementações separadas, uma para IPv4 e outra para IPV6;
- Como utiliza o processo de hard handover, a mobilidade é propensa à perda de pacotes ou no mínimo um delay significativo nessa transição [1];
- Não é recomendado para ambientes de micro-mobilidade [2];
- Para tráfego de tempo real tem uma tendência grande de perdas ou delay significativos;
- Roteamento triangular é um grande empecilho para aplicações de tempo real [1] [3];
- Necessidade de roteadores intermediários próprios para MIP para prover roteamento de pacotes;
- O uso do MIP requer mudança na pilha IP [2];;
- Faz uso de dois IP's quando em redes estrangeiras;
- Utiliza o reencapsulamento de pacotes IP aumentando cerca de 20B que também contribui para o aumento do delay de transmissão em ambientes com tunelamento [2];

## **2 - Gerenciamento de Mobilidade mSCTP**

No cenário apresentado através da figura 2, existe um MN (Mobile Node), CN (Correspondente Node) e dois Access Point (AP1 e AP2). O MN possui duas interfaces sendo que a Net1 (interface de rede 1) se encontra dentro da sub-rede 192.168.0.0/24, possuindo IP 192.168.0.10, já a segunda interface (Net2), não possui IP atribuído a ela (ND -Não definido). O CN possui apenas uma interface, o qual se encontra no momento na rede 10.10.10.0/24 possuindo o IP 10.10.10.50.

Num primeiro momento existe a abertura de uma associação (mesma idéia de conexão no TCP) entre MN e CN. Após a troca de alguns dados, o MN começa a se movimentar em direção a outro Access Point (no caso AP2).

Na abertura de uma associação o primeiro chunk que obrigatoriamente deve ser enviado é o INIT, onde o emissor(MN) informa a sua lista de IPs que utiliza no momento. O receptor (CN) ao receber o chunk INIT, após processamento sem erros, responde com um INIT ACK por onde também informa o endereços IPs que usa no momento.

Ao receber o INIT ACK, o MN o processa e envia de volta o chunk COOKIE ECHO. O CN ao receber o COOKIE ECHO, o mesmo será processado e como resposta será

enviado o chunk COOKIE ACK, concluindo dessa maneira a abertura de uma associação.

Empacotar dados de usuário junto ao chunk COOKIE ECHO no mesmo pacote SCTP pode ser feito pelo MN, mas isso dependerá se a implementação da pilha SCTP suporta. Claro que se houver envio de dados ocorrerá confirmação através de chunk SACK.

A partir do momento que foi concluída a associação, dados começam a ser trocados entre as partes e logo em seguida o MN inicia o movimento em direção ao AP2.

Quando o MN entrar na área de influência do AP2, começará a ocorrer uma negociação para que a interface Net2 do MN obtenha um IP do AP2 (é presumido que existe um serviço de DHCP na rede do AP2).

Ao obter o novo endereço IP, o MN o informa ao CN através do envio de um chunk ASCONF para que o mesmo considere o novo endereço como parte da associação, podendo dessa maneira enviar e receber dados por ele. O CN confirma a adição do novo IP com o envio do chunk ASCONF ACK, e só a partir deste momento é que o novo endereço IP passará a fazer parte da associação.

Os dados continuam a ser enviados pelo MN através do IP 192.168.0.10. Em um determinado momento o MN envia um chunk ASCONF para o CN informando para que deve alterar o endereço primário para o envio de dados, de 192.168.0.10 para 11.11.11.30. O CN confirma a alteração enviando de volta o chunk ASCONF ACK.

Em outro determinado momento o MN envia outro chunk ASCONF para o CN pedindo que se delete da associação o endereço IP 192.168.0.10, pois ele passará a não ser mais usado pelo MN. A confirmação da realização da solicitação é feita através do envio de um chunk ASCONF ACK.

Por questão de clareza a troca de cada mensagem com chunks ASCONF foram feitas de maneira separada, mas em um processo normal eles poderiam ter sido realizadas utilizando um único chunk ASCONF contendo várias estruturas TLVs (Entenda por estruturas TLV uma gama de pacotes de controles que possuem a mesma estrutura de cabeçalho, T(Type) L(Length) e V(Value)).

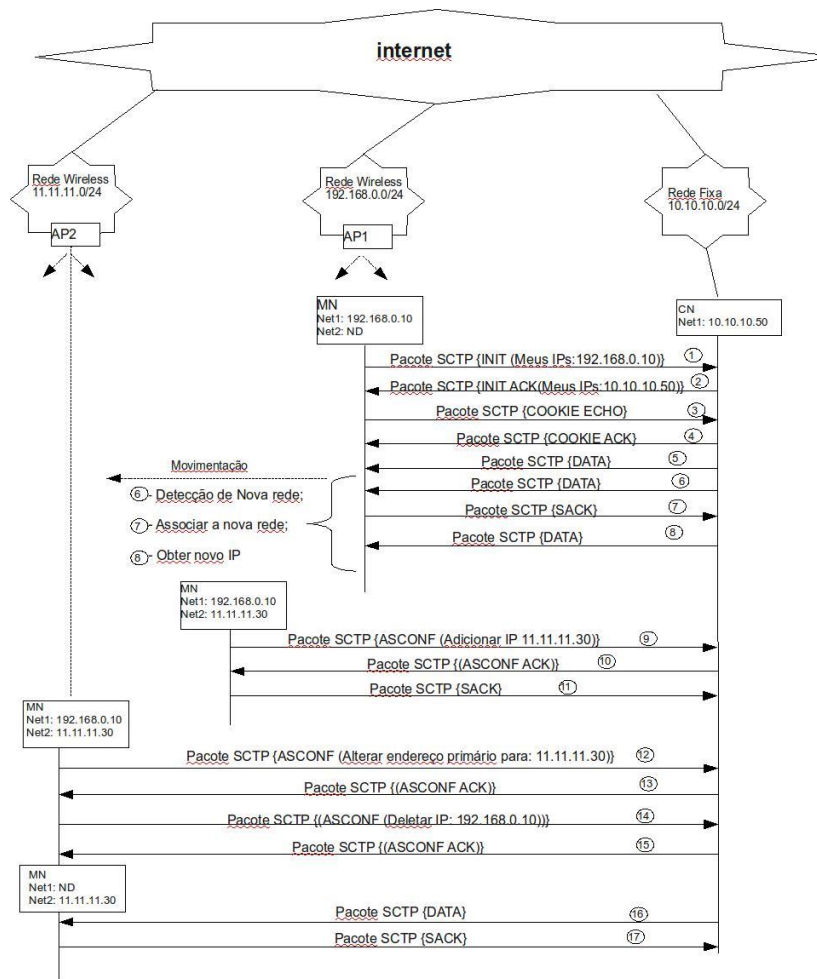


Figura 2 – Gerenciamento de mobilidade utilizando o SCTP

## 2.1 - Vantagens e Desvantagens do Uso de Gerenciamento de Mobilidade com mSCTP

Vantagens elencadas pelo uso do SCTP para prover mobilidade:

- Por ser um protocolo de transporte de uso geral, sua tendência é que seja naturalmente instalado em todas as máquinas juntamente com o sistema operacional;
- Não precisa instalar nenhum tipo servidor/agente na rede, como é o caso do MIP ou SIP;
- Como também é um protocolo de sinalização possui muita informação para possíveis tomadas de decisões de handoff;
- Possui suporte nativo tanto para ipv4 como ipv6;
- Pode proporcionar um handover sem perda de dados e sem interrupção de serviço pelo uso do método de soft handover;

Desvantagens elencadas pelo uso do SCTP para prover mobilidade:

- Implementações da pilha SCTP, seguindo a RFC4960(2007) e RFC 5061(2007) só existe em sistemas operacionais FreeBSD;
- Segundo [4] e [5], o SCTP não é apto para realizar handover com tráfego de tempo real;
- Conforme em [6] as implementações do SCTP ainda estão imaturas e propensas a interpretações errôneas de implementação;
- O controle das rotas em nível de estação SCTP tem que ser feito pela aplicação;
- Não possui um gerenciador de localização, conseqüentemente não consegue realizar uma associação quando as duas partes a serem envolvidas são móveis;

### **3 - Gerenciamento de Mobilidade SIP**

No cenário apresentado através da figura 3, temos um MN (Mobile Node), CN (Correspondente Node), dois Access Point (AP1 e AP2) e dois servidores SIP instalados na mesma máquina (Registrar e Redirect).

O MN envia uma mensagem SIP Invite para um servidor Redirect tentando obter o endereço IP do CN desejado. O servidor Redirect informa o endereço IP do CN, sendo confirmado o recebimento da resposta por parte do MN pelo envio da mensagem SIP ACK.

Uma vez que o MN tenha obtido o endereço IP do CN desejado é enviado uma mensagem SIP Invite ao CN. Ao receber a mensagem SIP Invite e processá-la o CN envia uma resposta 180 Ringing informando dessa maneira ao MN que uma solicitação para abertura de sessão chegou e que foi disparado um sinal sonoro ou uma mensagem visual para usuário do CN.

Logo em seguida o CN envia a mensagem SIP 200 OK, informando dessa maneira ao MN que aceita a abertura de sessão desde que o MN concorde com as diretrizes imposta pela mensagem SIP 200 OK. Concordando com a diretrizes impostas pela mensagem SIP 200 OK o MN envia em resposta a mensagem SIP ACK confirmando que aceita as condições para abertura de sessão SIP e está pronto para enviar/receber dados.

A partir do momento que a sessão SIP foi estabelecida e que se iniciou a transferência de dados o MN começa a se movimentar em direção ao AP2, sendo que isso ocorre paralelamente a transferência de dados. Em certo ponto da movimentação o MN começa a fazer o handoff para o AP2, sendo que a partir desse momento ele fica sem comunicação com o CN. Esse é o momento crítico onde provavelmente haverá perda de dados ou no mínimo um delay significativo na chegada dos dados. Esse delay é variável e dependente da condição de negociação da rede.

Uma vez que o MN tenha obtido um novo IP do AP2 (é presumido que este dispositivo também é um servidor DHCP), ele automaticamente enviará uma nova mensagem SIP conhecida como re-INVITE, possuindo a mesma identificação da sessão anteriormente estabelecida, para que a sessão seja reconfigurada com um novo endereço IP. Ao receber a mensagem SIP re-INVITE, após processamento, o CN retornará uma



mensagem SIP 200 OK. Assim que o MN receber a mensagem SIP 200 OK ele retornará uma mensagem SIP ACK, após isso a sessão está pronta novamente para enviar/receber dados de usuários.

Após o restabelecimento da sessão do MN para com o CN, ou mesmo em paralelo ou até mesmo antes, uma mensagem SIP Register é enviada para o Servidor SIP de Registro para que se atualize a localização do MN para que outros usuários possam localizá-lo a qualquer momento. O Servidor SIP Register confirma a atualização do cadastro enviado em resposta uma mensagem SIP 200 OK.

### **3.1 - Vantagens e Desvantagens do Uso de Gerenciamento de Mobilidade com SIP**

Vantagens do uso do SIP para gerenciamento de mobilidade:

- Já possui significativa estrutura instalada na internet, sendo utilizado como protocolo de sinalização padrão para gerenciamento de sessões multimídia de tempo real a partir das redes 3G [7] [8] [9] [10];
- Como o SIP é um protocolo da camada de aplicação, sua funcionalidade é transparente para as camadas mais baixas, por exemplo, funciona independentemente da rede trabalhar com IPv4 ou IPv6 [11];
- Como é um protocolo de camada de aplicação não exige nenhuma alteração na pilha IP, nem do lado do cliente, nem do lado do servidor;
- É um protocolo escalável com o uso de Servidores SIP Redirect [11] [12];
- Para aplicações de tempo real é um dos mais promissores [13];
- O SIP suporta nativamente mobilidade pessoal, também é capaz de suportar mobilidade de terminal, sessão e serviço [14];
- Possui nativamente um gerenciador de mobilidade;

Desvantagens do uso do SIP para gerenciamento de mobilidade:

- Dependendo do tipo de serviço a ser utilizado, geralmente esse protocolo requer servidores intermediários como pré-requisito de operação;
- Em uma proposta de handover em ambiente de múltiplos tipos de rede, poderá haver um esquema de computação extensa para poder trabalhar com cada tipo de arquitetura de comunicação ou no mínimo um framework para poder fornecer informações para tomadas de decisões de quando realizar o handover [15]
- Realiza handoff com perda de dados, ou no mínimo um delay significativo [10];
- Não é recomendado para ambientes de handover horizontal;
- Para tráfego de tempo real tem uma tendência grande de perdas ou delay significativos;
- A aplicação tem que se preocupar de controlar as tabelas de roteamento cada vez que ocorre um handoff;

- Não consegue manter sessão em ambientes de handover quando utilizando como transporte o protocolo TCP [11] [13] [16];
- Não é considerado próprio para micro-mobilidade devido ao restabelecimento da comunicação após o handoff, pois requer uma troca de sinalização fim-a-fim, resultando em um longo delay e um alto overhead [17] [18];
- Visto que o SIP é um protocolo de camada de aplicação, delays são inerentes de camada em camada [7] [19];
- Mensagens de controle SIP geralmente são maiores do que mensagens tipo MIP ou SCTP [5];

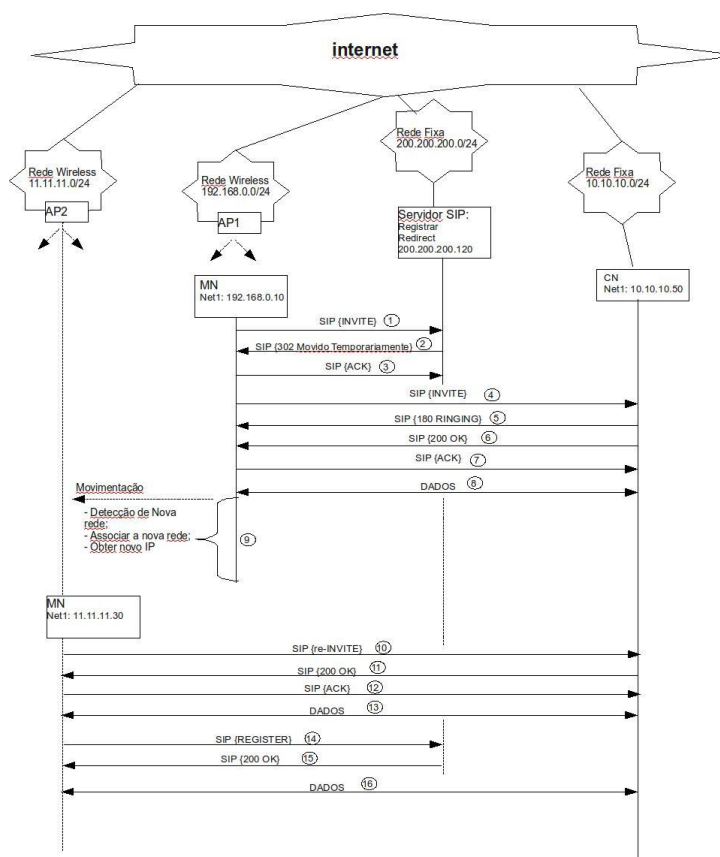


Figura 3 – Gerenciamento de mobilidade utilizando o SIP

**CONCLUSÕES:** Nenhum dos protocolos de gerenciamento de mobilidade estudado se mostrou pleno para atender todas as expectativas de tráfego de tempo real como de tráfego de tempo não real. Uma junção que talvez traga bons frutos é o uso do SIP com uma variação do protocolo SCTP, chamado PR-SCTP, o qual faz com que o protocolo SCTP seja mais parecido com o UDP, se adaptando dessa maneira ao tráfego de tempo real. Mas isso será trabalho para futuras pesquisas.

## Referências Bibliográficas

- [1] Paul Stalvig. Introdução ao protocolo de controle de fluxo de transmissão (sctp), a próxima geração do protocolo de controle de transmissão (tcp). Site Web - <http://www.f5networks.com.br/pdf/white-papers/introducao-ao-stream-control-transmission-protocol-sctp-wp.pdf> Acessado em 10/04/2009.
- [2] L.N. Scheiner. Análise de Desempenho do Protocolo SIP na Sinalização de Chamadas de Voz sobre IP. 2005.

- [3] K.H. Grace, D. Pecelli, and J.D. D'Amelia. Improving Multi-homed SCTP Mobile Communication Performance. Technical Papers, The MITRE Corporation, 2006.
- [4] J. Rosenberg, H. Schulzrinne, and G. Camarillo. The stream control transmission protocol(SCTP) as a transport for the session initiation protocol (SIP). Technical report, RFC 4168, Internet Engineering Task Force, 2005.
- [5] X.L. Wang and VCM Leung. Applying PR-SCTP to transport SIP traffic. In IEEE Global Telecommunications Conference, 2005. GLOBECOM'05, volume 2, 2005.
- [6] L. Budzisz, R. Ferrus, K.J. Grinnemo, A. Brunstrom, and F. Casadevall. An analytical estimation of the failover time in SCTP multihoming scenarios. In Wireless Communications and Networking Conference (WCNC), 2007.
- [7] N. Banerjee, A. Acharya, and SK Das. Seamless SIP-based mobility for multimedia applications. IEEE Network, 20(2):6–13, 2006.
- [8] M. Cardenete-Suriol, J. Mangués-Bafalluy, M. Portoles-Comeras, M. Requena-Esteso, and M. Gorricho. VoIP performance in SIP-based vertical handovers between WLAN and GPRS/UMTS networks. In IEEE International Conference on Communications, 2007. ICC'07, pages 1973–1978, 2007.
- [9] D.P. Kim, S.J. Koh, and Y.J. Kim. A scheme of primary path switching for mobile terminals using SCTP handover. WSEAS Transactions on Communications, 6(3):432, 2007.
- [10] K. Oberle, S. Wahl, A. Sitek, and B. Alcatel-Lucent. Enhanced Methods for SIP based Session Mobility in a Converged Network. Mobile and Wireless Communications Summit, 2007. 16th IST, pages 1–5, 2007.
- [11] R. Good and N. Ventura. A multilayered hybrid architecture to support vertical handover between IEEE802. 11 and UMTS. In Proceedings of the 2006 international conference on Wireless communications and mobile computing, pages 257–262. ACM New York, NY, USA, 2006.
- [12] L.N. Scheiner. An'alise de Desempenho do Protocolo SIP na Sinalizac, ̃ao de Chamadas de Voz sobre IP. 2005.
- [13] Q. Wang and M.A. Abu-Rgheff. Mobility management architectures based on joint mobile IP and SIP protocols. IEEE Wireless Communications, 13(6):68–76, 2006.
- [14] N. Banerjee, A. Acharya, and SK Das. Seamless SIP-based mobility for multimedia applications. IEEE Network, 20(2):6–13, 2006.
- [15] T. Al Mosawi, D. Wisely, and H. Aghvami. A Novel Micro Mobility Solution Based on Media Independent Handover and SIP. In 2006 IEEE 64th Vehicular Technology Conference, 2006. VTC-2006 Fall, pages 1–5, 2006.
- [16] N. Seta, H. Miyajima, L. Zhang, H. Hayashi, and T. Fujii. All-SIP Mobility: Session Continuity on Handover in Heterogeneous Access Environment. In IEEE 65th Vehicular Technology Conference, 2007. VTC2007-Spring, pages 1121–1126, 2007.
- [17] D. Vali, S. Paskalis, A. Kaloxylas, and L. Merakos. An efficient micro-mobility solution for SIP networks. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), volume 6, pages 3088–3092, 2003.
- [18] A. Ezzouhairi, A. Quintero, and S. Pierre. A New SCTP mobility scheme supporting vertical handover. In IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2006.(WiMob'2006), pages 205–211, 2006.
- [19] DS Nursimloo and HA Chan. Integrating fast mobile IPv6 and SIP in 4G network for real-time mobility. In Networks, 2005. Jointly held with the 2005 IEEE 7th Malaysia International Conference on Communication., 2005 13th IEEE International Conference



# Comportamento de personagens em jogos de computador

Deise Miranda Borges<sup>1</sup>, Rafael Gonçalves Barreira<sup>1</sup>, Jackson Gomes de Souza<sup>1</sup>

<sup>1</sup>Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas  
(CEULP/ULBRA)

Avenida Teotônio Segurado 1501 SUL – Caixa Postal 160 – Palmas – TO – Brasil

{infodeise, leafarneo, jackson.souza}@gmail.com

***Resumo.** Na atualidade existem várias categorias de jogos computacionais que atendem a vários estilos, oferecendo uma característica em comum: interação jogo-jogador. Um dos responsáveis por essa interação são os personagens controlados pelo jogo que atuam de forma inteligente para trazer mais realidade no que tange ao seu comportamento.*

**Palavras chave:** personagem, jogo, comportamento, inteligência.

## 1. Introdução

Os jogos de computadores são softwares de entretenimento que devido ao seu constante crescimento vem focando mais na interatividade e jogabilidade do jogo, na tentativa de trazer oponentes e aliados cada vez mais inteligentes. Os jogos de computador têm um caráter multidisciplinar no seu processo de desenvolvimento, no qual envolve diversas áreas da computação. De um modo geral o processo de desenvolvimento de jogos envolvem os seguintes módulos:

- Roteiro – é onde é escrito a estória do jogo, personalidade dos personagens, época em que o jogo irá se passar, características do ambiente do jogo, etc. Esta parte é uma das partes fundamentais de um jogo, pois é com um bom roteiro que se podem convencer os investidores da potencialidade do produto.
- Produção de Áudio – Neste módulo é realizada a criação da trilha e efeitos sonoros do jogo.
- Game Design – este módulo tem como principal função a conceituação artística do jogo. Nele é mostrado as principais características dos cenários, esboço dos personagens, descrição das texturas fundamentais, mapas e descrição das fases.
- Modelagem 3D – é o setor responsável por criar objetos geométricos e terrenos das fases, ou seja, tem como função fazer a modelagem e animação 3D do jogo.
- Programação – é o setor onde será feita toda a implementação do jogo, e um dos módulos que mais vem tendo desafios, em relação a áreas de inteligência artificial, na busca por técnicas que tornem os oponentes ou aliados cada vez mais inteligentes.

A motivação para investi em técnicas de inteligência artificial é apresentada por Karlsson (2005), que diz que os usuários dos jogos estão esperando jogos, ou mais especificamente personagens de jogos que se comportem de forma que estejam na qualidade e credibilidade esperada. Associando ainda ao fato de que o mercado de jogos é um dos que mais movimenta dinheiro da área de entretenimento.

Neste trabalho será apresentado a definição e o que houver relacionado ao personagem dos jogos de computador, como também mostrar técnicas que permitam elaborar um personagem controlado pelo jogo que tenha um comportamento inteligente.

## **2. Personagens em jogos de computador**

Os *non-player character* (NPC), são os personagens que não são controlados pelo jogador. Eles existem nos jogos multiplayer, vários jogadores, e singleplayer, só há um jogador, sendo que é no estilo singleplayer que ele ganha mais destaque, pois é o adversário do jogador.

Para que um jogo possa se tornar mais realístico em termo do comportamento dos personagens, os NPCs devem ter um certo nível de “inteligência”, antigamente essa “inteligência” era implementada através de regras por meio da estrutura *if-then-else*, mas além de tornar mais complexo a mudança em algum comportamento do personagem ainda trazia uma precariedade em termos de comportamentos mais complexos. Desta maneira identificou-se que para se ter comportamentos mais complexos, capaz de interagir de forma mais real e atrativa, dentro de um ambiente era necessário buscar novas técnicas que fizessem com que os NPCs tivessem a capacidade de agir de forma racional e aprender com suas ações.

Os NPCs possuem algumas características que estabelecem se ele é inteligente, como: ser autônomo, reativo, proativo e social. A seguir será explicado agentes autônomos e agentes emocionais hedonistas.

### **2.1. Agentes Autônomos**

Um agente autônomo é o que consegue operar com completa autonomia, decidir como relacionar os dados obtidos com ações de modo que seus objetivos sejam atingidos com sucesso (MAES, 1995). Assim pode-se dizer de uma forma mais clara que agentes autônomos são capazes de perceber o ambiente através dos seus sensores e operar com completa autonomia, sem qualquer intervenção, buscando sempre atingir seus objetivos da melhor forma possível.

Um agente autônomo é capaz de agir e sentir de acordo com seus propósitos em um ambiente, possuindo autonomia, pró-atividade, reatividade, aprendizado, ação propositada e adaptabilidade (WOOLDRIDGE & JENNINGS, 1995). Um agente autônomo pode se autoprogramar para adquirir o conhecimento que ele precisa para ajudar ou atrapalhar - no caso de um NPC adversário - um usuário. O agente autônomo é implementado com um conhecimento inicial mínimo e a partir daí vai aprendendo com o comportamento do usuário e dos outros agentes, desta forma quando ele estiver em alguma situação parecida com alguma que esteja na sua base de conhecimento ele irá tomar decisões baseado na sua experiência.

### **2.2. Agentes Emocionais Hedonistas**

O Hedonismo é uma teoria ou doutrina filosófico-moral que coloca como maior valor, o prazer e a satisfação individual em primeiro lugar (Gomes, 2007). Dessa maneira,

agentes hedonistas agirão e tomarão decisões baseados no prazer e na dor que as decisões trarão, mas como se trata de agentes emocionais hedonistas o prazer e a dor serão transformados em emoções, no qual cada emoção irá codificar um sub-objetivo no agente. Sendo assim a ação do agente e a sua tomada de decisão será baseada de acordo com o objetivo que foi codificado pela emoção.

A idéia de usar agentes emocionais hedonistas para a abordagem de comportamentos de personagens está na característica de considerar emoções como base para tomada de decisão. Um personagem que tenha emoções, ou simula uma, consegue se aproximar de um comportamento de um outro jogador, e não como um simples personagem de computador.

### **2.3. Comportamento inteligente**

As definições de NPC, agentes autônomos e hedonistas, têm um ponto em comum: buscam atingir objetivos. Para se alcançar um objetivo é necessário haver estratégias para isso, no caso dos agentes hedonistas usa a emoção para esse fim. Um NPC para atingir seu objetivo, tanto o objetivo específico no jogo - ser o adversário em um jogo de tiro - quanto o geral, traz uma qualidade para o jogo na questão de interação com o jogador, que deve seguir uma estratégia, um plano, sendo mais eficiente e/ou aceitável possível.

Ter uma boa estratégia não define um NPC como “inteligente”, pois de que adiantaria um personagem - como em um jogo de tiro - criar a melhor estratégia possível se ele não consegue se mover até o ponto e executa-la. Nesse momento surge outro fato para auxiliar na definição de comportamento inteligente do personagem, a movimentação. Nos jogos em que há uso de NPC, geralmente o ambiente possui obstáculos, e baseado na definição de um agente, o NPC deve interagir com o ambiente e realizar sua estratégia o levando em consideração. Então é possível dizer que um comportamento inteligente para NPC pode ser definido com a estratégia desenvolvida por ele para atingir seu objetivo e a forma de chegar ao seu alvo e executar a estratégia.

## **3. Técnicas de Comportamento de personagens**

Para o desenvolvimento do comportamento dos NPCs existem diversas técnicas de IA, no qual há algumas que são muito utilizadas e outras que começaram a ser utilizadas recentemente. Nesse artigo serão apresentadas as técnicas: máquina de estados finitos, sistema baseado em regras, lógica *fuzzy*, algoritmo A\*, algoritmos genéticos e redes neurais.

### **3.1. Máquina de estados finita**

A técnica de máquina de estados finita (*Finite State Machines* - FSMs) consiste na utilização de um conjunto de estados e um conjunto de transição entre esses estados, sendo que essas transições geralmente são eventos dentro de um jogo, como um jogador entra no campo de visão do NPC (KARLSSON, 2005). Os estados possíveis refletem as ações a serem tomadas pelo NPC baseado nos eventos do jogo. Por ser uma abordagem

simples, é uma das mais utilizadas até hoje. Um dos primeiros jogos a usar FSMs foi o jogo de estratégia em tempo real Herzo Wei (GALDINO, 2007).

Um exemplo de FSMs para jogos de tiro em primeira pessoa é apresentado na figura 1, no qual há um estado em que o NPC deve ficar parado, e quando um inimigo entra em seu campo de visão ele entra em modo de perseguição. Durante a perseguição do inimigo, caso este se encontre no alcance do NPC, o personagem começa a atacar. O comportamento fica executando ações até atingir o estado inicial, que no caso abaixo seria o NPC ficar parado.

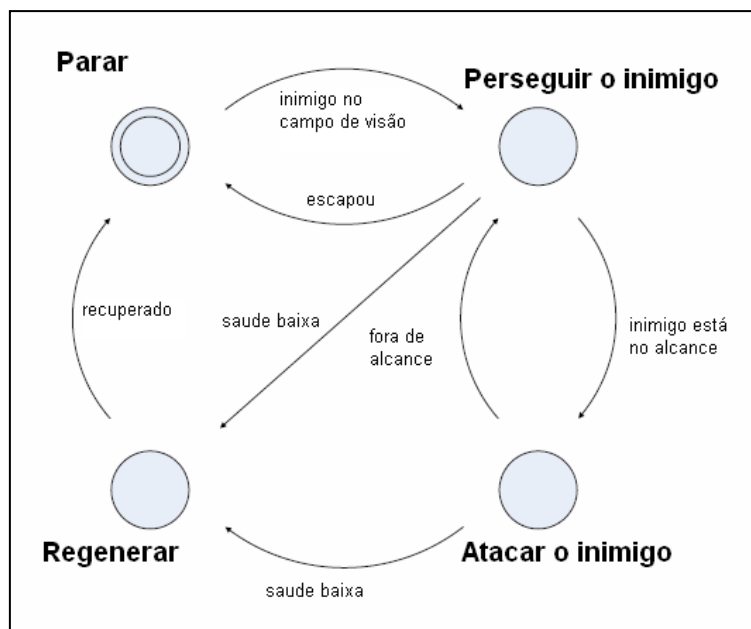


Figura 1: diagrama dos estados e transições de um jogo.

A simplicidade da criação de uma FSMs está na quantidade de estados possíveis. Com o aumento do número de estados deve-se prever todas as transições possíveis, aumentando assim a complexidade da FSMs. Outra desvantagem é a repetição de ações, pois a FSMs sempre executa uma mesma ação para a determinada transição.

### 3.2. Sistemas baseados em regras

Um sistema baseados em regras (*Rule Based Systems* – RBSs) é composto por um conjunto de variáveis e um conjunto de regras sobre essas variáveis. Tem como vantagem representar o conhecimento de como as pessoas pensam, permitindo também modelar conhecimentos complexos, devido ao fato de poder ser baseado em objetivo (KARLSSON, 2005).

As variáveis representam características inerentes ao comportamento do jogo, que no caso de um jogo de tiro seria a distância do alvo, pontos de vida do NPC, pontos de vida do inimigo. As regras são representadas por condicionais (*if-then-else*) dessas variáveis e o conseqüente representa a ação a ser tomada.



No caso do jogo de tiro em primeira pessoa, uma regra seria: se os pontos de vida forem maior que 30% e pontos de vida do inimigo forem menor que 40% então inicia a perseguição. Uma outra regra é se os pontos de vida forem menor que 70% e o inimigo estiver distante então o NPC deve se regenerar.

Apesar de ser mais preciso do que uma FSM e mais fácil de implementar, um RBSs apresenta algumas desvantagens, tal como a necessidade de espaço de armazenamento e poder de processamento maior (KARLSSON, 2005).

### 3.3. Lógica fuzzy

Na lógica clássica há dois valores: o verdadeiro e o falso. As FSMs e RBSs e outras técnicas usam dessa lógica para representar o conhecimento. Mas o ser humano tem noções de imprecisão, como “os pontos de vidas estão um pouco baixo”. O objetivo da lógica *fuzzy*, ou difusa, é representar essas variações ou valores intermediários entre o verdadeiro e o falso (KARLSSON, 2005).

Em jogos o uso de lógica *fuzzy* é associado às FSMs formando as *Fuzzy* FSMs (FuSMs). Uma FuSMs é uma máquina de estados finitas que os valores das transições são valores *fuzzy*. Suas vantagens incluem adaptabilidade e versatilidade que as máquinas de FSM não possuem, no entanto tem uma desvantagem que é também relacionada à quantidade de estados possíveis, junto com as características heurísticas da lógica fuzzy pode haver um consumo alto de memória (KARLSSON, 2005).

Um dos primeiros jogos a usar FuSMs foi o jogo Unreal Tournament (tiro em primeira pessoa), para fazer com que os inimigo (KARLSSON, 2005). Um exemplo, baseado na FSM apresentado na seção 4.1, seria na transição de parado para perseguição, o NPC irá perseguir, além de esta no campo de visão, se for “muito corajoso”, e em caso contrario ele irá fugir para encontra os demais membros da equipe.

### 3.4. algoritmo A\*

O algoritmo A asterisco, ou algoritmo A estrela, é um algoritmo de busca de caminho, pois como foi apresentada anteriormente, a movimentação junto com a estratégia é que define o comportamento inteligente do NPC. O algoritmo A\* é um algoritmo de busca de caminho (*path-finder*) que consegue encontrar, se existir, o caminho entre dois pontos em um ambiente, sendo que esse ambiente, ou mapa, é manipulado com um conjunto de pontos onde pode ficar um personagem, a posição (KARLSSON, 2005).

A base do algoritmo esta na equação:

$$f(n) = g(n) + h(n) \quad (1)$$

Onde  $f(n)$  é função que conta o valor do caminho,  $g(n)$  é custo de movimentação desde o ponto inicial até o ponto  $n$ , e  $h(n)$  é a distancia de  $n$  até o ponto final, sendo que  $h(n)$  não contem o valor exato e sim uma tentativa de previsão, é nesse valor que se encontra a heurística do algoritmo (LESTER, 2005).

Durante a execução do algoritmo, os possíveis caminhos são testados até que se chegue ao final ou não consiga encontrar o caminho. A seleção do menor caminho está no menor valor de  $f(n)$  (LESTER, 2005).

Apesar de ser facilmente implementado, a utilização desse algoritmo em jogos pode apresentar problemas, pois em um jogo com vários personagens o gasto em processamento se torna alto. O problema não está só na quantidade de instâncias do algoritmo que são utilizadas, mas também na existência de ambientes dinâmicos, onde há outras entidades que se movem (KARLSSON, 2005).

### **3.5. Algoritmos genéticos**

Os Algoritmos Genéticos (AG) baseiam-se na evolução natural das espécies. A unidade de informação trabalhada por esse algoritmo é o cromossomo, que são as características a serem analisadas pelo AG. Inicialmente é gerado um conjunto de cromossomos, as possíveis soluções. O próximo passo, e um dos mais importantes é a avaliação dos cromossomos, pois é nessa parte que o cálculo de um AG termina ou continua. São selecionados então os cromossomos mais adaptados e os demais são descartados. O passo seguinte é realizar os cruzamentos, que envolve ou não outro cromossomo selecionado, e a mutação nos cromossomos selecionados na etapa anterior. Com os novos cromossomos gerados volta ao passo de seleção, e no final do processo tem como resultado uma solução mais eficiente (ROSA, 2008).

Para se utilizar os algoritmos genéticos devem-se tratar os dados de entrada para ficarem na forma de um cromossomo, uma forma de normalização que não segue as características originais resultará em uma solução que não segue as características originais. Quando os parâmetro de configuração do comportamento do NPC possuir interações que são difíceis de prever - ao contrario de um comportamento simples que se use FSM - o AG terá uma grande utilidade (KARLSSON, 2005).

### **3.6. Redes neurais**

Redes neurais ou Redes Neurais Artificiais (RNA) pode ser definida como:

“É um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que tem a propensão natural para armazenar conhecimento experimental e torna-lo disponível para o uso.” (HAIKIN, 2001, p. 28).

Esses componentes de processamento de informação são chamados de neurônios, e possui uma estrutura análoga ao neurônio natural, pois possui um conjunto de entradas, o que seria os dendritos, um somador, o núcleo da célula, e uma função de ativação, o axônio (HAIKIN, 2001). Uma RNA é composta por um conjunto de neurônios e ligações entre eles. O conhecimento é produzido e armazenado nos pesos atribuídos a essas ligações (KARLSSON, 2005).

A semelhança de uma RNA com a Rede Neural Natural (RNN), não se limita em relação à estrutura e organização, mas também a capacidade de aprender, que é um processo onde os pesos das conexões são ajustados e são adaptados para o ambiente no qual a RN está inserida. De acordo com a forma de que os pesos são modificados é um

tipo de aprendizado diferente. Dentre os existentes podemos citar dois: o aprendizado supervisionado, em que o aprendizado se dá pela avaliação da saída por um professor, e este, que possui um conjunto de saídas esperadas, define como será o ajuste dos pesos. E o aprendizado não supervisionado, em que não há um professor e nem um conjunto de saídas definidas, e sim uma condição a ser alcançada, deixando a rede ser responsável por adaptar a essa condição. Baseado nos tipos de aprendizado há seis usos de redes neurais, sendo que duas, reconhecimento (tomada de decisão baseado em uma decisão já tomada) e controle (controle do comportamento físico), já podem ser aplicados a jogos (KARLSSON, 2005).

As RNAs, por sua semelhança com as RNNs, também trabalham com informações simples, que geralmente são valores entre 0 e 1 inclusive. Então um dos principais preocupações a se trabalhar com redes neurais reside em mapear as características a serem analisadas em dados que sejam entendíveis pela RNA.

#### **4. Considerações finais**

O uso de NPC em jogos é de muita importância, pois ele é uma maneira de buscar uma maior interação com o jogador. O objetivo desse trabalho foi apresentar uma visão sobre personagens em jogos de computador e como o comportamento deles é desenvolvido. O estudo sobre como é realizado a criação dos NPCs mostra um fato do desenvolvimento de jogos, as várias áreas da informática, sendo que na área relativa à programação percebe-se que diversas das técnicas de Inteligência Artificial existentes na literatura.

Durante os estudos desse trabalho iniciou-se o desenvolvimento de personagens, com o objetivo de testar as técnicas. Como trabalho futuro é a possibilidade de desenvolver personagens para todos os tipos existentes, abrindo caminho assim para o desenvolvimento em um jogo completo.

#### **5. Referências bibliográficas.**

GALDINO, Carlos Henrique Silva. **Inteligência artificial aplicada no desenvolvimento de jogos de computador**. Out. 2007. Disponível em: <[http://www.programadoresdejogos.com/trab\\_academicos/carlos\\_galdino.pdf](http://www.programadoresdejogos.com/trab_academicos/carlos_galdino.pdf)>.

GOMES, A. R. (2007). **Vocabulário de Filosofia**. Disponível em: <<http://ocanto.esenviseu.net/lexh.htm>>.

KARLSSON, Börje Felipe Fernandes. **Um Middleware de Inteligência Artificial para Jogos Digitais**. Dissertação de Mestrado, 2005. Departamento de informática. Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro. 126 p.

HAIKIN, Simon. **Redes Neurais: princípios e prática.**; trad. Paulo Martins Engel. 2ª ed. Porto Alegre: Bookman, 2001.

JENNINGS, N., SYCARA, K., e WOOLDRIDGE, M. **A Roadmap of agent research and development**. 1998.

LESTER, Patrick. **A\* Pathfinding for Beginners**. Jul. 2005. Disponível em:  
<<http://www.policyalmanac.org/games/aStarTutorial.htm>>.

MAES, P.: **Modeling Adaptive Autonomous Agents. Artificial Life**, an Overview. C. Langton (ed.). MIT Press. Cambridge MA. (1995).

ROSA, Thatiane de Oliveira; LUZ, Hellen Souza. **Conceitos básicos de Algoritmos Genéticos: Teoria e Prática**. In: X Encontro de Estudantes de Informática do Tocantins, 2001, Palmas, **Anais**. Palmas: CEULP/ULBRA, 2008. p. 79-90

WOOLDRIDGE, M.; JENNINGS, N. R. **Agent Theories, Architectures, and Languages: a Survey**. In: WOOLDRIDGE, M.; JENNINGS, N. R. (eds.) **Intelligent Agents**. Berlin, Springer-Verlag, 1995.

# **CompWeb: Um Repositório de Componentes de Software Acessível por Web Services REST**

Leandro Oliveira Ferreira, Michael Schuenck

Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas  
(CEULP/ULBRA)

Caixa Postal 160 – 77.054-970 – Palmas – TO – Brasil

lefer\_fsa@gmail.com, michael@ulbra-to.br

***Resumo:** A busca por componentes de software é uma das tarefas mais onerosas para desenvolvedores de software se estes não possuírem à sua disposição uma ferramenta de fácil acesso onde eles possam encontrar componentes úteis aos seus projetos. Neste contexto, este trabalho tem como objetivo apresentar um repositório Web de componentes denominado CompWeb. Desenvolvido utilizando a tecnologia Ruby on Rails, com uma interface em HTML e suporte a outras interfaces na forma de Web Services REST, o CompWeb é um repositório de fácil acesso em que os desenvolvedores poderão pesquisar, recuperar e submeter componentes, de diferentes linguagens e plataformas, na forma de uma ou mais versões.*

**Palavras chave:** reuso de software, repositório, componentes.

## **Introdução**

A evolução dos processos de desenvolvimento de software tem caminhado para a redução do esforço de produção e manutenção através do reuso de software. O reuso de software é uma abordagem em que o principal objetivo é evitar o retrabalho no desenvolvimento de um novo projeto, utilizando trabalhos anteriores, fazendo com que as soluções já desenvolvidas sejam imediatamente implementadas em novos contextos (Holanda e Souza, 2006).

A partir de produtos de software preexistentes, o reuso propõe o desenvolvimento de sistemas de modo que semelhanças entre requisitos e arquitetura de diversos sistemas possam ser mais bem exploradas (Holanda, 2001). Com a utilização do reuso e identificação dessas semelhanças, empresas podem obter melhores resultados no desenvolvimento de softwares em termos de tempo de desenvolvimento e custo. Além disso, a cada reuso de uma unidade de software, sua validade e eficiência são verificadas, fazendo com que sua qualidade seja aumentada continuamente (Holanda, 2001).

Neste contexto, o desenvolvimento de sistemas baseados em componentes tem se destacado como uma das principais abordagens de reaproveitamento. Nesta abordagem, há a necessidade de existir o compartilhamento de componentes, fato motivador da existência de diversos repositórios, que são o elo entre o desenvolvedor e utilizador de componentes de software (Tizzei, 2007). Tais ferramentas permitem armazenamento e busca com o propósito de auxiliar no reuso de componentes, e, assim, melhorar as metas de qualidade, custo e produtividade (Oliveira, 2008). Uma vez que

esses componentes armazenados são bem organizados, divulgados e gerenciados, essa ferramenta torna-se um poderoso instrumento na política do reuso.

Porém, muitos repositórios foram desenvolvidos com deficiências como falta de uma descrição de alto nível dos componentes e inexistência de uma busca eficiente, além do problema dos repositórios serem desenvolvidos, geralmente, adotando abordagens locais e centralizadas, limitando a acessibilidade dos mesmos (Schuenck, 2005).

Neste contexto, este trabalho apresenta um sistema denominado CompWeb (CompWeb, 2009), que é um repositório Web de componentes de software com suporte a interfaces HTML (*Hypertext Markup Language*) e *Web Service REST (Representational State Transfer)*. Este último tipo de interface é uma característica importante do CompWeb, pois possibilita maior acessibilidade ao repositório, uma vez que, desta forma, as buscas por componentes podem ser embutidas em quaisquer outras ferramentas, como ambientes de desenvolvimento. Este é o caso de um cliente para o CompWeb desenvolvido na forma de plugin para a ferramenta Eclipse, que também será apresentado neste trabalho.

No CompWeb podem ser compartilhados componentes de quaisquer plataformas ou linguagens. Assim, componentes de software que seguem quaisquer modelos de componentes (EJB, .NET, JavaBeans, COM, etc) e até mesmo arquivos de script como JavaScript, Python e Ruby, poderão ser submetidos, compartilhados e reutilizados. Juntamente com esses, poderão ser registradas as documentações referentes aos componentes. O CompWeb também dispõe de um sistema de versionamento, permitindo o envio de novas versões de um componente possibilitando sua maturidade, além de um mecanismo de interação do utilizador com o desenvolvedor de componentes através de *feedbacks* que possam relatar possíveis problemas ou até mesmo melhorias para o componente.

O restante deste documento está dividido da seguinte maneira: a Seção 2 descreve os repositórios de componentes; a Seção 3 traz o CompWeb propriamente dito; a Seção 4 descreve o plugin que foi desenvolvido para acessar a interface Web Service REST do CompWeb, na seção 5 traz os trabalhos relacionados e na Seção 6 temos as considerações finais.

## **REPOSITÓRIOS de componentes**

O modelo de desenvolvimento baseado em componentes de softwares (DBCS) é uma técnica que otimiza o processo de desenvolvimento de software porque permite a reutilização de componentes criados em outros projetos além de possibilitar que vários grupos possam trabalhar em partes diferentes de uma aplicação em construção simultaneamente (Brechó, 2009).

Atualmente, boa parte dos softwares disponíveis no mercado foi desenvolvida em blocos monolíticos, formados por partes interrelacionadas, onde esses relacionamentos não têm uma interface bem definida, dificultando a reutilização desses blocos (Braga, 2000).

O DBCS propicia o reuso de software e tem como base a construção e reutilização de componentes interoperáveis, com interfaces bem definidas, evidenciando

os tipos de relacionamentos permitidos por estes componentes, reduzindo a complexidade no desenvolvimento, assim como os custos, através da reutilização de componentes (Brechó, 2009). A cada erro constatado, um componente é melhorado gradativamente, propiciando um desenvolvimento de software com mais qualidade. Sendo assim, um mecanismo de armazenamento e busca desses componentes é bem usual para os desenvolvedores no processo, muitas vezes exaustivo, de localização dos componentes adequados a alguma necessidade de um software em desenvolvimento (Schuenck, 2005).

Dentro dessa perspectiva, um repositório de componentes surge com o intuito de organizar e facilitar a busca de componentes, que nem sempre estão centralizados e organizados. Um repositório será considerado, para o propósito desse trabalho, uma ferramenta onde componentes de software serão armazenados e classificados de modo a facilitar a busca e recuperação dos mesmos.

No entanto, algumas outras funcionalidades podem facilitar e, com isso, incentivar o seu uso. De acordo com (Ezran, 1999) Serão apresentadas algumas funcionalidades de um repositório:

Identificação e descrição: para identificar e descrever um componente de software são necessárias algumas características tais como nome, domínio, palavras-chave, dentre outras que os identificam e os diferenciam dos demais componentes de software que compõem esse mesmo repositório.

Inserção: um repositório deve permitir que usuários autorizados insiram novos componentes de software ou ainda, novas versões de componentes já cadastrados.

Exploração do catálogo: aos usuários de um repositório deve ser permitido que explorem a lista de componentes para que possam conhecer e analisar as características dos componentes disponíveis.

Pesquisa textual: um repositório deve permitir que seus usuários façam pesquisas mais específicas na descrição dos componentes. Como resultados da pesquisa, serão obtidos um ou mais componentes que satisfaçam as condições desejadas.

Recuperação: após a identificação do componente desejado, um repositório deve permitir que seus usuários recuperem esse componente para que possam posteriormente utilizá-lo num processo de reuso.

Histórico: é importante para o gerenciamento de um repositório que ele possa armazenar informações de uso, modificações, criação e exclusão de cada um dos ativos disponíveis. Essas informações devem montar uma base histórica que facilitará a análise e a reutilização dos mesmos.

Controle de acesso: um repositório pode adotar uma política de segurança para que determinadas funcionalidades só estejam acessíveis a pessoas autorizadas. Por exemplo, pode-se definir uma política de segurança onde a pesquisa e utilização de componentes sejam públicas, já o envio de componentes e de versões de componentes seja restrito aos usuários cadastrados no sistema.

Gerenciamento de versões: um repositório pode conter várias versões de um mesmo componente e, sendo assim, é recomendável que haja algum mecanismo para controlar essas versões e estabelecer o relacionamento entre elas.

Com a crescente motivação para o reuso de componentes de software no processo de desenvolvimento, diversas empresas estão investindo na construção de repositórios com suporte ao reuso. Alguns para uso interno, outros com propósitos comerciais (Holanda e Souza, 2006).

## O REPOSITÓRIO CompWeb

A idéia da construção de um repositório surgiu após estudos de outros repositórios e a detecção de funcionalidades que esses não dispunham, como diferentes meios de acesso e de maneira fácil, possibilidades de diferentes tipos de busca e uma forma simples de submissão de componentes. Além disso, empresas têm dificuldades em lidar com o reuso corporativo, especialmente se existem diferentes equipes de desenvolvimento e ainda mais se elas estão distribuídas geograficamente. Outro importante ponto é a utilização de reuso em modelos de melhoria de processos de software como é o caso do nível E (parcialmente definido) do MPS.BR (Melhoria de Processo de Software Brasileiro), que define o processo de Gerência de Reutilização, cujo propósito envolve gerenciar o ciclo de vida de ativos reutilizáveis (Softex, 2009), fomentando mais a reutilização e, em conseqüência, a utilização de repositórios de componentes.

Neste contexto, foi desenvolvido o CompWeb, um repositório de componentes de software com suporte a interfaces HTML e *Web Service* REST, que possibilita o armazenamento e busca de componentes, desenvolvido utilizando a tecnologia Ruby on Rails. O CompWeb também propõe um sistema de controle de versões que possa facilitar o versionamento de componentes armazenados.

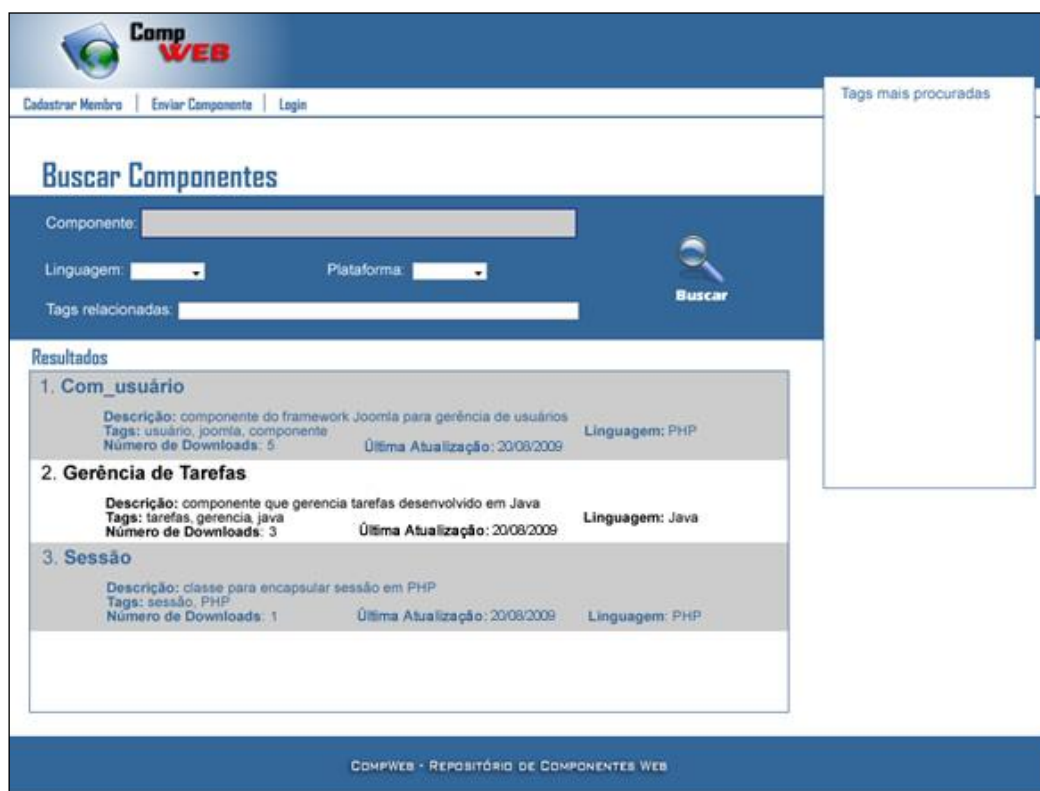
O CompWeb pode ser implantando tanto em uma intranet quanto na Internet, uma vez que a sua plataforma permite. No primeiro caso, traz grandes benefícios como um ambiente centralizado para armazenamento e busca de componentes, com diferentes tipos de acesso, dentro do próprio ambiente corporativo, restringindo o acesso a apenas os membros da empresa. No segundo caso, é interessante ressaltar, principalmente o incentivo ao desenvolvimento colaborativo, onde desenvolvedores poderão receber *feedbacks* de utilizadores de componentes sobre o funcionamento desses, além dos demais benefícios citados anteriormente.

O CompWeb disponibiliza um sistema de busca por nome do componente, *tags* relacionadas, linguagem e plataforma. É importante destacar que sistemas de busca com *tags* têm se popularizado e sido adotados por vários sites como o registrador de favoritos delicias.com (delicias, 2009) entre outros, como forma de classificação de conteúdo, facilitando as buscas posteriores. A **Figura 8** demonstra o sistema de busca do CompWeb através da interface HTML.

O utilizador de componentes poderá buscar componentes de diversas formas como, por exemplo, especificando linguagem de programação ou *tags* (separadas por espaço em branco) relacionadas com o componente que ele deseja. Vale ressaltar que o utilizador de componentes não necessariamente precisar ser membro ou ter sido autenticado no sistema para poder buscar algum componente, facilitando assim o acesso.

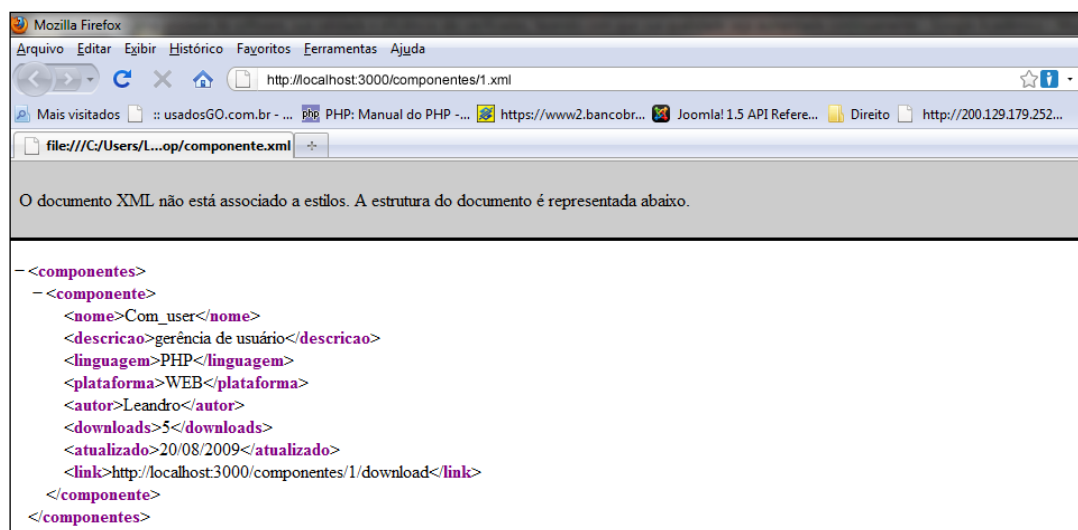
O resultado de uma busca no CompWeb disponibiliza também informações como o número de downloads e data de atualização dos componentes. A data de atualização se refere à data em que a versão mais recente do componente foi submetida.





**Figura 8 - Sistema de Busca do CompWeb**

É possível o acesso ao CompWeb através de interfaces Web Services REST para busca e downloads de arquivos como demonstra a **Figura 9**.



**Figura 9 - Resultado de uma busca utilizando a interface Web Service REST**

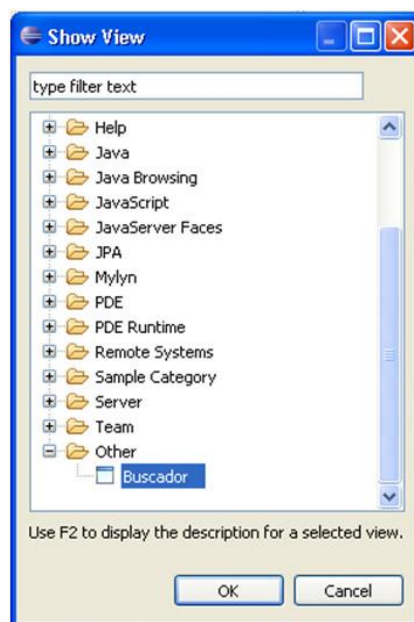
A **Figura 9** demonstra o resultado de um consulta a interface Web Service REST. Pode-se observar que a URL (*Uniform Resource Locator*) informada termina com “.xml”, o que indica que o servidor retornará um recurso do tipo XML. Cada tag do XML representa uma informação do componente, como por exemplo a tag “nome”, que trará o nome do componente, a tag “descricao” que conterá a descrição, a tag

“linguagem” que apresentará a linguagem de programação em que foi desenvolvido o componente, a tag “plataforma” que diz qual a plataforma do componente, a tag “autor” que diz quem é o autor do componente, a tag “downloads” que diz o número de downloads que pode servir como qualificador de um componente para o utilizador, a tag “atualizado” traz a data da ultima atualização do componente que, assim como a tag “downloads”, é um importante dado para que o utilizador possa qualificar o componente, e o link para baixar o componente. É importante destacar que essa URL poderá ser acessada por qualquer ferramenta com acesso Web incorporado que permita o tratamento dos dados. Isso poderá ser visto na seção seguinte onde será descrito o *plugin* da ferramenta Eclipse, que foi desenvolvido para demonstrar a utilização da interface Web Service REST.

## O Plugin PARA O AMBIENTE ECLIPSE

O cliente foi desenvolvido em forma de *plugin* da ferramenta Eclipse (Eclipse, 2009) com o objetivo de facilitar e otimizar a busca de componentes por parte dos desenvolvedores, e para exemplificar a utilização das interfaces Web Service REST do CompWeb, utilizando a linguagem Java. Este *plugin* foi criado como uma “*view*”, que é um dos pontos de extensão do Eclipse. Uma *view* no contexto Eclipse é uma subjanela que pode ser arrastada e colocada em qualquer área dentro da ferramenta. A escolha do Eclipse se deve ao fato de ele ser uma das ferramentas mais populares para auxílio ao desenvolvimento de software em Java e em outras linguagens de programação, já que suas extensões permitem que inúmeras linguagens sejam suportadas. Outro fator que contribuiu para a escolha do Eclipse é a sua arquitetura modular, que permite a fácil criação de novos plugins.

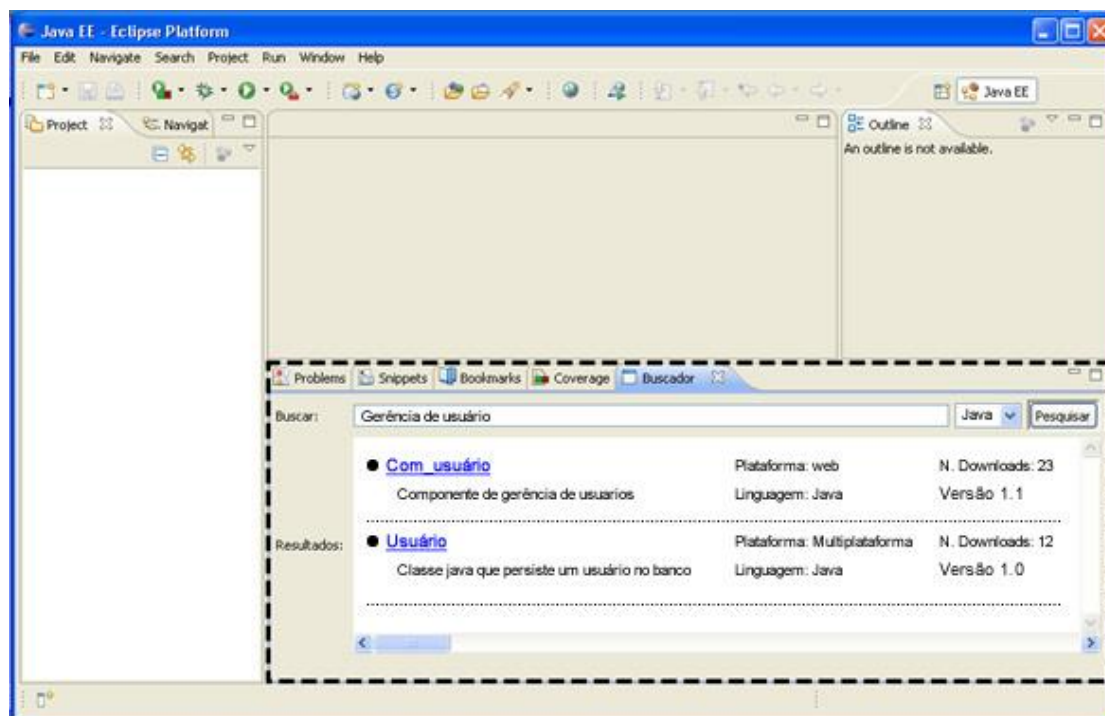
Para utilizar o *plugin*, depois de instalá-lo, basta acessar o menu *window*→*show*→*view*→*other* e, na janela de *show View* do Eclipse, ativar a *view* Buscador como mostra a **Figura 10**.



**Figura 10 - ativando a view buscador no Eclipse**

Depois de ativada a *view* Buscador, o *plugin* será exibido na parte inferior da janela do Eclipse, como mostra a **Figura 11**.

Para fazer uma busca por componentes, basta informar o nome, *tags* ou descrição do componente desejado dentro do campo “Buscar”, além da linguagem desejada, selecionando-a no *dropdown* ao lado e clicar no botão “Pesquisar”. O resultado da busca aparecerá na área de resultados.



**Figura 11 - Janela do Eclipse com a view de busca**

A busca é efetuada de modo simples, bastando ao utilizador informar o nome, linguagem ou as *tags*, separadas por um espaço em branco, relacionadas com o componente que deseja e clicar no botão “Pesquisar”. Serão exibidos na tela de resultados os componentes relacionados com os dados informados juntamente com os respectivos links para download. A URL que o Web Service REST responderá será <http://leandro-note/componentes/busca>. Assim, o utilizador de componentes poderá buscar e baixar componentes do seu próprio ambiente de desenvolvimento de maneira simples e rápida.

## Trabalhos Relacionados

Nos próximos parágrafos serão abordados alguns repositórios estudados, com uma breve descrição de suas principais características que serviram de base para construção do CompWeb.

O framework arquitetural ComponentForge proposto por Oliveira (Oliveira 2006) para apoiar o DBCS distribuído. O framework oferece suporte à adição, busca, certificação e negociação de componentes com uma arquitetura orientada a serviços. O compartilhamento dos componentes é feito através de um esquema de nomes

hierárquico similar ao DNS (Domain Name Service). Os nomes são organizados em uma árvore hierárquica onde os nós intermediários são zonas ou domínios e as folhas são os bens (para Oliveira, bens são artefatos de software, como código executável e fonte). As zonas representam os produtores desses bens. Elas são subdivididas em domínios que facilitam o gerenciamento dos bens. Entretanto

O Digital Assets Manager (DAM, 2009) é um exemplo de repositório comercial que suporta versionamento e permite a integração com ambientes de desenvolvimento. Diferente de muitos repositórios acadêmicos, o DAM fornece informações interessantes não somente para desenvolvedores e integradores de componentes como também para os gerentes de desenvolvimento. Por exemplo, uma de suas métricas de reutilização de componentes avalia quanto dinheiro foi economizado com a reutilização de componentes. Entretanto, o modelo de dados do DAM não oferece facilitadores para desenvolver ferramentas de apoio para integradores e construtores de componentes.

O ambiente Ragnarok (Christensen, 1999) é um sistema de gerência de configuração, com suporte a elementos de arquitetura. Ao incluir uma nova versão de um componente, a arquitetura é considerada, e todos os componentes relacionados são por sua vez versionados. Outra característica do Ragnarok é a possibilidade de rastrear a evolução da arquitetura do sistema. Entretanto o Ragnarok se preocupa mais com a evolução da arquitetura e não oferece alguns requisitos básicos como pesquisar componentes, integração com o ambiente de desenvolvimento ou um sistema de *tags*.

O ambiente MAE (Roshandel, 2004) é utilizado para gerenciamento da evolução da arquitetura de sistemas e não de componentes como o DAM e o ComponentForge. O modelo do sistema arquitetural é o ponto central do MAE. Este modelo de sistema arquitetural integra conceitos de gerência de configuração (GCS) de sistemas e de arquitetura. O resultado desta integração é o suporte à evolução, às variações e às opções disponíveis pelos elementos da arquitetura. Contudo, o MAE não possui algumas características comuns aos repositórios de unidades de reuso como pesquisa e integração com ambiente de desenvolvimento.

## **Considerações Finais**

É notório que o desenvolvimento baseado em componentes de software tem crescido muito, fazendo com que empresas em geral necessitem de repositórios de componentes de fácil acesso com diferentes opções de busca. O CompWeb tenta sanar essas necessidades oferecendo características e funcionalidades como facilidade de uso e acesso, envio de múltiplas versões de componentes, busca e download de componentes com diferentes opções de busca.

O CompWeb traz um interface de fácil acesso e simples de usar. Por exemplo, para fazer uma busca não é necessário qualquer tipo de cadastro, apenas informar os parâmetros desejados, pois é comum que desenvolvedores tenham a tendência de desenvolver a uma solução a partir do “zero” devido à dificuldade no acesso ou a necessidade de realizar cadastros para efetuar o download de um componente.

A busca é outro fator importante, afinal se ela não for eficiente o desenvolvedor verá com um complicador a utilização de um repositório e não como um benefício. Por isso o CompWeb traz um sistema de busca que várias opções, onde poderá ser

informado não só o nome do componente, mas também, a linguagem, plataforma ou tags para que a localização do componente seja mais eficiente.

É importante ressaltar também o feedback entre utilizador e desenvolvedor de componentes disponível pelo CompWeb. Essa funcionalidade incentiva a maturidade do componente, uma vez que o desenvolvedor poderá corrigir ou melhorar seus componentes utilizando também informações dadas pelo utilizador do seu componente.

A acessibilidade por Web Services REST acrescentou muito valor ao CompWeb devido à interoperabilidade proporcionada, possibilitando a comunicação entre aplicações clientes de diferentes plataformas e linguagens ao repositório. Neste sentido, também foi desenvolvido um cliente em forma de plugin da ferramenta Eclipse que permite que buscas e downloads de componentes sejam realizadas a partir do próprio ambiente de desenvolvimento, o que agiliza ainda mais a busca por componentes.

É proposto, como trabalho futuro, um sistema que integre o CompWeb a um sistema de gerenciamento de versões (SGV) como o Subversion (SVN, 2009), Git (Git, 2009), entre outros. A intenção deste sistema é tornar o processo de publicação de novas versões automatizada e transparente para o desenvolvedor de componentes. Para tanto, serão utilizadas API's para acesso aos sistemas de controle de versão. Além disso, será explorado o conceito de *tags* para que assim, os componentes possam, automaticamente, ser publicados no CompWeb e se tornarem disponíveis para download. No contexto de sistemas de controle de versões, *tags* são mecanismos que identificam uma revisão em um projeto como uma versão publicável.

## Bibliografia

(Braga, 2000) Braga, R.; Werner, C.; Mattoso, M.; - Uma infra-estrutura de Reutilização baseada em Modelos de Domínio- João Pessoa, outubro 2000, pp.366-369.

(Brechó, 2009) – Site do Laboratório de Engenharia de Software – Equipe de Reutilização de Software. Disponível em: [http://reuse.cos.ufrj.br/site/pt/index.php?option=com\\_content&task=view&id=37&Itemid=47](http://reuse.cos.ufrj.br/site/pt/index.php?option=com_content&task=view&id=37&Itemid=47). Acesso em 09 de maio de 2009.

(Ezran, 1999) Ezran, Michel, et al.. Practical Software Reuse: The Essencial guide.1999.185 p (Holanda, 2001) - Holanda, Caroline B. S.; Walcelio Melo; Clarissa Angélica de A. de Souza.- Proreuso: um repositório de componentes para Web dirigido por um processo de reuso - XV Simpósio Brasileiro de Engenharia de Software - 2001 - Rio de Janeiro, RJ, Brasil.

(Holanda e Souza, 2006) Holanda, Caroline Batista Spencer, Souza, Clarissa Angélica de Almeida de, Walcélio L. Melo. ProReuso: Um Repositório de Componentes para Web Dirigido por um Processo de Reuso - XV Simpósio Brasileiro de Engenharia de Software

(Oliveira, 2008) Oliveira, João Paulo Freitas de. X-CORE: um serviço de repositório distribuído e compartilhado de componentes de software – João Pessoa, 2008.

(Schuenck, 2005) - Michael Schuenck, Carla Tanure, José Jorge L. Dias Jr., Sindolfo Miranda, Yuri Negócio, Glêdson Elias - Análise de Repositórios de Componentes para a Identificação das Soluções mais Adotadas - COMPOSE - Departamento de Informática - Universidade Federal da Paraíba.

(Tizzei, 2007) - Uma infra-estrutura de suporte à evolução para repositórios de componentes - redação da Dissertação apresentada para a Banca Examinadora antes da defesa da dissertação. – Campinas 2 de março de 2007 – SP – Brasil

(SVN, 2009) - Site do Subversion – disponível em: <http://subversion.tigris.org/> - acesso em 27 de setembro de 2009.

(GIT, 2009) - Site do Git – disponível em: <http://git-scm.com/> - acesso em 27 de setembro de 2009.

(delicious, 2009) - Site do delicious.com – disponível em: <http://delicious.com/>- acesso em 27 de setembro de 2009.

(Oliveira 2006) João P.F. de Oliveira, Michael Schuenck, and Glêdson Elias. Componentforge: Um framework arquitetural para desenvolvimento distribuição baseado em componentes. In Workshop de Desenvolvimento Baseado em Componentes - Recife, 2006.

(DAM, 2009) Cit. Digital assets manager. [<http://www.digitalassersmanager.com.br>]. Acessado em 05 de setembro de 2009.

(Christensen, 1999) Christensen, Henrik Baerbak. The Ragnarok Architectural Software Configuration Management Model. Nordic Journal of Computing, 1999.

(Roshandel, 2004) Roshandel, Roshanak.; André van der Hoek, Marija Mikic-Rakic, and Nenad Medvidovic. Mae - a system model and environment for managing architectural evolution. ACM Transactions on Software Engineering and Methodology, 2004.

(Softex, 2009), “MPS.BR: Melhoria de Processo do Software Brasileiro”, Guia Geral v. 1.2, Disponível em: <http://www.softex.br/mpsbr>. Acessado em 12 de outubro de 2009.

(Eclipse, 2009) Site do Eclipse – disponível em: <http://eclipse.org/>- acesso em 27 de setembro de 2009.

(CompWeb, 2009) CompWeb : Repositório de Componentes Web – trabalho apresentado disciplina de Estágio Supervisionado do curso de Sistemas de Informação – CEULP/ULBRA – Palmas-To, 2009.

# Controle de uma Estufa Agrícola Utilizando Controlador Lógico Programável

Nathália Amaral Carvalho, Héverton de Melo Bezerra, Antonio Jeronimo Netto, Marcelo Lisboa Rocha

Curso de Ciência da Computação – Centro Universitário UNIRG  
Avenida Antônio Neves da Silva S/N, Pq. das Acácias, Gurupi - TO, CEP: 77400-000

nathalliacomp@gmail.com, hevertonmb@yahoo.com.br, ajnetto@yahoo.com.br,  
marcelolisboarochoa@yahoo.com.br

## Resumo

*Neste trabalho, foi realizado um estudo, partindo da necessidade de uma maior precisão e duração na influência benéfica exercida pelas variáveis passíveis de controle em uma estufa. Buscou-se conciliar o Controlador Lógico Programável (CLP) juntamente com a teoria de controle do Controlador Proporcional Integral Derivativo (PID) com a finalidade de beneficiar a cultura de eucalipto. Destaca-se que para alcançar uma menor perda com relação às mudas, fez-se necessário o estudo sobre o cultivo do eucalipto em estufas. Assim consegue-se compreender como se deve comportar o ambiente onde será cultivado, podendo determinar as variáveis controladas. Conclui-se que, com a integração da lógica PID ao CLP aplicada na estufa, tem-se uma maior precisão referente às variáveis ambientais a controlar. O projeto não se limita apenas na aplicação citada no presente trabalho, podendo ser aplicado em outras culturas.*

## 1. Introdução

O diferencial principal desse trabalho é exibir o controle da estufa utilizando-se os dois controladores, onde será agregado nas propriedades do CLP o segundo controlador, PID. As primeiras aplicações desse tipo surgiram por volta de 1975, apresentando desde então, vantagens imensuráveis. (ALQUATI e SANTOS, 2005).

Segundo a Associação Brasileira de Normas Técnicas (ABNT), CLP é um equipamento eletrônico digital com hardware e software compatíveis com aplicações industriais. E segundo a National Electrical Manufacturers Association (NEMA), CLP é um aparelho eletrônico digital que utiliza uma memória programável para o armazenamento interno de instruções para implementações específicas, tais como lógica, seqüenciamento, temporização, contagem e aritmética, para controlar, através de módulos de entradas e saídas, vários tipos de máquinas ou processos.

Um controlador PID, ou controlador de três termos terá uma saída para uma entrada de um erro. Num sistema de malha fechada, o sinal atuante de erro, que é a diferença entre o sinal de entrada e o sinal de retroação (sinal de saída), excita o controlador de modo a reduzir o erro e trazer o valor do sinal de saída para o valor desejado. Uma característica necessária de um sistema de malha fechada é a realimentação. (OGATA, 1970 ed. 1998, pág.178)

Utilizando-se malha fechada, pode-se inserir o Proporcional-Integral-Derivativo (PID), o qual é o tipo de controlador mais utilizado em indústrias, na arquitetura de

processamento de um segundo, o CLP. Alguns fabricantes já realizam essa inserção ainda na fabricação dos mesmos.

A linguagem que será utilizada para implementar o software será a LADDER.

O fato de o Tocantins ser um estado que apresenta um número favorável de produtores rurais, horticultores, agrônomos e também de haver poucos aplicativos para determinado controle, contribuiu na escolha da estufa, visto que se julgou necessário a construção de um controlador, o qual atuará satisfatoriamente pouco dependente de interferências humana, além de estar incentivando e colaborando com uma maior exploração de hortifrutigranjeiros, vegetações, entre outros.

O controlador de estufa é uma solução que visa um melhor desenvolvimento de mudas, visto que a mão-de-obra humana é muito sujeita a falhas, a automação firma o objetivo de evitar erros simples e retirar a responsabilidade do homem no controle de certos fatores. Logo, o objetivo geral do trabalho é projetar um sistema de automação integralizando a lógica PID ao controlador CLP em malha fechada para uma maior precisão na influência exercida nas variáveis a serem controladas em uma estufa de eucalipto.

## ***2. Controle e automação por Controlador Lógico Programável (CLP) e Proporcional Integral Derivativo(PID)***

“Sistema automático de controle pelo qual os mecanismos verificam seu próprio funcionamento, efetuando medições e introduzindo correções, sem a necessidade da interferência do homem.” (OGATA,1998), ou seja, o homem passa a supervisionar a execução da tarefa almejada, “desresponsabilizando-no” da ação-prática.

O CLP atua satisfatoriamente na efetuação de tarefas designadas a automatizar, fato esse contribuinte para expandi-lo em áreas extremamente distintas, apresentando inúmeras vantagens: maior confiabilidade, facilidade de programação, manutenção, redução de gastos para alteração de hardware, etc.

A automação pode ocorrer em duas formas básicas: Sistema de comando (malha aberta) e Sistema de controle (malha fechada).

Para a realização desse controlador é necessário que ocorra realimentação, portanto, do controle em malha fechada, que permite entre outras características: aumentar a precisão do sistema, rejeitar o efeito de perturbações externas, melhorar a dinâmica do sistema e, eventualmente, estabilizar um sistema naturalmente instável em malha aberta e diminuir a sensibilidade do sistema a variações dos parâmetros do processo, tornando o sistema robusto.

O CLP trabalha com SCAN (leitura constante) de informações, como pode ser visto na Figura 1. Logo, a cada leitura é repetido esse ciclo de execução descrito por tais etapas (SOUZA,2004). Já o controlador PID tem como base, também, o controle em malha fechada. Segundo OGATA (OGATA, 1998), nos dias atuais mais de 90% dos controladores utilizados em indústrias fazem uso do PID.

Os fabricantes do CLP que realizam a implementação da função PID a sua lógica deve atentar-se quanto à configuração das constantes, visto que um simples erro pode comprometer o desempenho do sistema e acarretar um mau funcionamento do processo. (OLIVEIRA, 2002)





Figura 12 – Scan de leitura

De acordo com o diagrama da Figura 2, interpreta-se que: Primeiramente a entrada é obtida do processo e em seguida é direcionada a conversão A/D, onde os valores que forem analógicos irão tornar-se digitais e os digitais manterão se iguais. Em seguida os valores serão encaminhados ao controlador, o qual trabalha com a lógica do PID, sofrerão modificações e, caso haja dados cuja saída necessite ser analógica, serão reconvertidos D/A encaminhando-se a processo. Nota-se que a saída corresponderá à nova entrada.

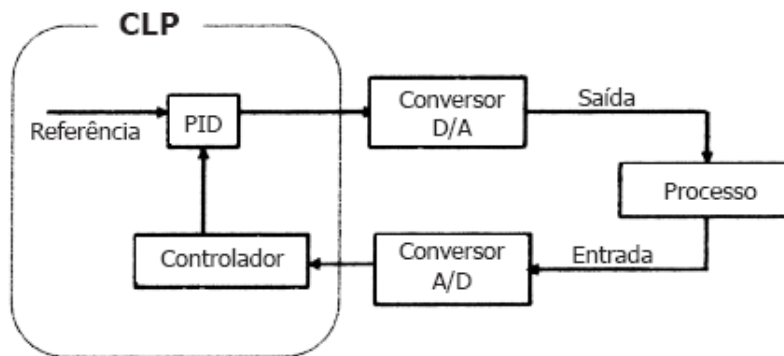


Figura 13 - Diagrama de funcionamento do PID ao CLP

O sinal de controle gerado pelo controlador PID é a sintonia no controlador: o ganho proporcional  $K$ , o tempo integral  $T_i$  e o tempo derivativo  $T_d$ .

$$u(t) = K(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt})$$

A Figura 3 resume como a lógica da Proporcional, Integral, Derivativa trabalha com as correções.

<b>P</b>	CORREÇÃO PROPORCIONAL AO ERRO	Conforme cresce o erro entre o valor real e o desejado cresce também a correção a ser aplicada ao processo
<b>I</b>	CORREÇÃO PROPORCIONAL AO PRODUTO ERRO x TEMPO	Uma correção mais intensa é aplicada a erros pequenos, porém que existem há muito tempo
<b>D</b>	CORREÇÃO PROPORCIONAL À TAXA DE VARIAÇÃO DO ERRO	Caso o erro esteja variando muito rápido, de modo a evitar oscilações, essa taxa deve ser reduzida

Figura 14 – Correção PID

### 3. Automação (CLP/PID) na Agricultura Brasileira

A utilização de estufa na agricultura, veja Figura 4, se justifica devido aos períodos climáticos adversos que prejudicam consideravelmente as plantações, gerando prejuízos a agricultores. Logo, as mudas estarão protegidas de fatores diversos: calor, baixa umidade do ar, frio, temporais, entre outros. Essa proteção resulta no aumento da produtividade, no melhor desenvolvimento do plantio, na possibilidade de colheitas precoces e no melhor controle sobre doenças e pragas, acarretando maior lucro.

A estufa automatizada contribuirá no controle de escassez de nutrientes e alimentos provocados por razões meteorológicas, como iluminação, temperatura e umidade.

A escolha do eucalipto deu-se ao fato de ser uma das plantas cultivadas que apresentam mais rápido crescimento é o Eucalipto, cujas plantações no Brasil ultrapassam os 3.800.000 hectares que os somando aos outros países, atinge mais de 18 milhões de hectares.

A planta, originada da Austrália, apresenta mais de 700 espécies diferentes. (EUCAGEN, 2009)



Figura 15 - Estufa (ZANATA FLORESTA,2009)

### 4. Resultados

As figuras apresentadas nesta seção, representam um sistema de automação em linguagem LAD, que integra a lógica do controlador PID ao CLP em malha fechada, para uma maior precisão na influência exercida das variáveis controláveis em uma estufa de eucalipto.

Inicialmente, na Figura 5, são representadas as entradas analógicas, a comparação e o resultado da comparação pela verificação do erro, que será descrito na seqüência. Primeiramente tem-se uma entrada analógica denominada AI1 sendo um contato analógico normalmente aberto, na qual é simulado o um sensor que registra o valor da variável do ambiente que vai ser ampliada pela função especial SF001 *SensAMP*, onde, é configurado o tipo do sensor para 0 ... 10 v, com o parâmetro Gain (ganho) de 0,10 para se ter uma faixa de medição de 0 a 100, de modo a ter uma faixa de mensuração próxima a real tanto para a temperatura que é medida entre 0 – 100°C, quanto para a umidade, que é medida entre 0 – 100%. Após ser amplificado, o sinal é

enviado pelo SF001 *SensAMP*, para a segunda entrada (AY) do Comparador Analógico, que é a função especial SF003 *Propband*, essa função também recebe o valor a ser comparado em sua primeira entrada (AX) pela função especial SF002 que contém o valor a ser comparado, o qual é denominado Set-point (Valor Almejado). No Set-point configura-se no parâmetro offset, o valor que deseja ser alcançado, enviando-o para a função SF003 pela entrada AX. Na função SF003 *Propband* é configurado o sensor para 0... 10 v e também, configura-se o parâmetro Gain (ganho) que será multiplicado pelo resultado da diferença entre as entradas AX e AY, onde o resultado dessa expressão é referenciado como tempo para o Temporizador Proporcional na função especial T006 *PropT* na Figura 21.

Primeiramente tem-se uma entrada analógica denominada AI1 que terá o seu valor amplificado por *SensAMP*, tem-se também o valor almejado (*SF002* Setpoint).

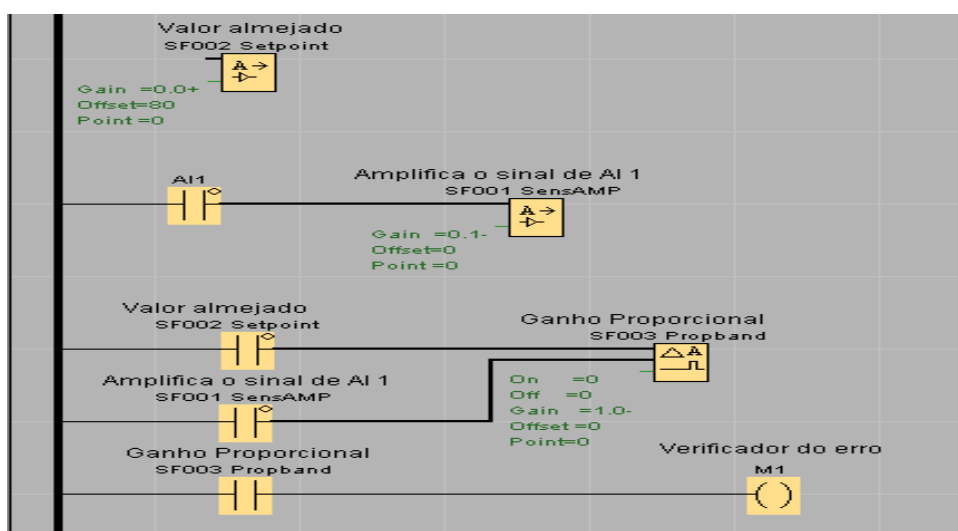


Figura 16 - Representação da comparação entre as entradas analógicas

No ganho proporcional será indicado o valor do ganho *Gain* e esse será multiplicado pela diferença entre o valor atual e o setpoint, resultando no valor do erro.

Conforme representado na Figura 17, é mostrada a verificação do erro e se ele for maior que zero, entrará no tempo integral.

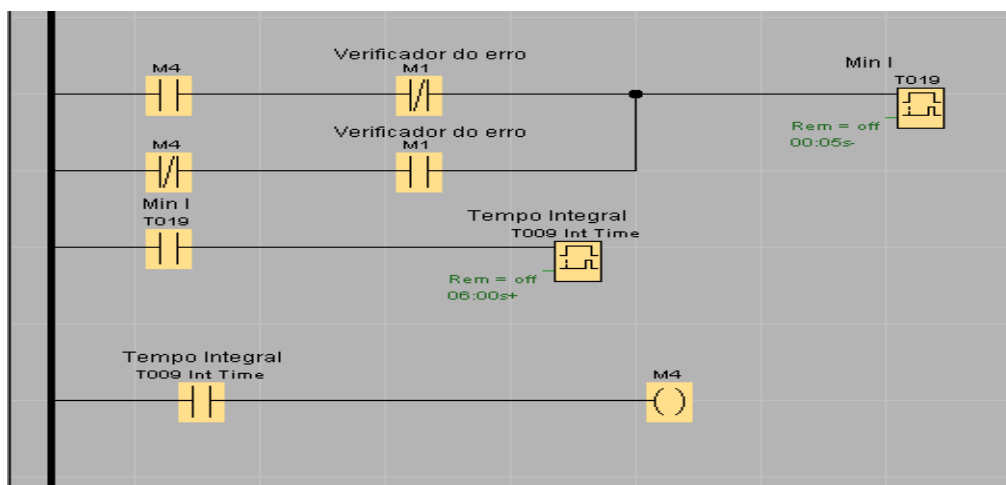
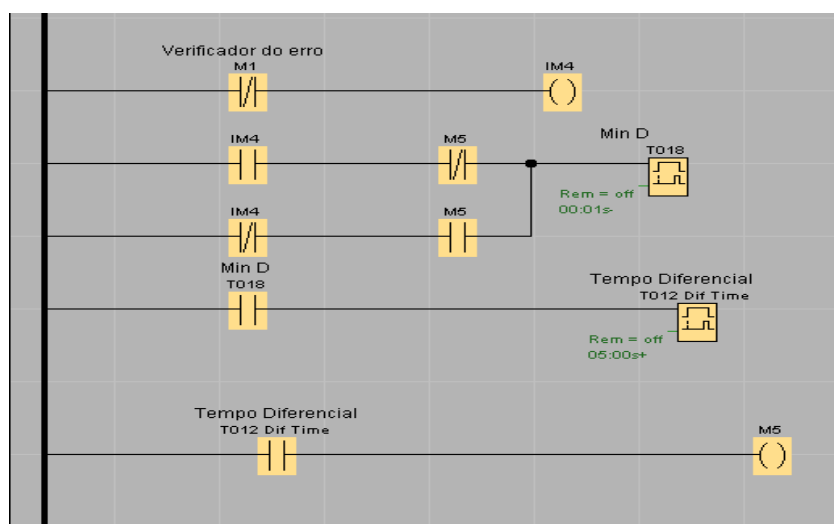


Figura 17 - Representação do Tempo Integral

Caso o resultado da diferença entre AX e AY do comparador SF003 seja maior que zero, significa dizer que é enviado um sinal lógico para o verificador de erro M1, sendo assim não será ativada a bobina IM4, pois para acionar IM4 o verificador M1 deverá enviar o sinal lógico de nível 0, sendo assim será ativada a implementação XOR com M1 e M4 que leva o sinal para o temporizador T019. Na implementação XOR é feita a comparação entre suas entradas, caso elas sejam diferentes a saída será sinal de nível lógico 1, sendo assim, é enviado o sinal lógico para os blocos On/Delay T019 e B009 (retardo de ligamento), com os quais são configurados e percorrido o tempo Integral. Ao término do tempo integral é enviado um sinal lógico para a bobina M4, que envia um sinal para a implementação XOR (M1, M4) e para o contato normalmente aberto M4 da Figura 19, o sinal que vai para o XOR (M1, M4), é para reiniciar o tempo integral caso o M1 ainda esteja enviando o sinal lógico de nível 1, o sinal que vai para M4 na Figura 19 é feita uma comparação OR com o M5, se aceita a condição, enviará um sinal ao Contador C008 *STORE C* e a cada vez que o C008 é acionado via tempo integral, é acrescentado 1 ao contador, o valor do C008 é o que define o tempo no Temporizador Integral/Diferencial na função especial T010 *INTDIF T* na Figura 21.

Conforme representado na Figura 18, é mostrada a verificação do erro e se ele for menor ou igual à zero, o IM4 será ativado pelo contato normalmente fechado do M1 e entrará no tempo diferencial.

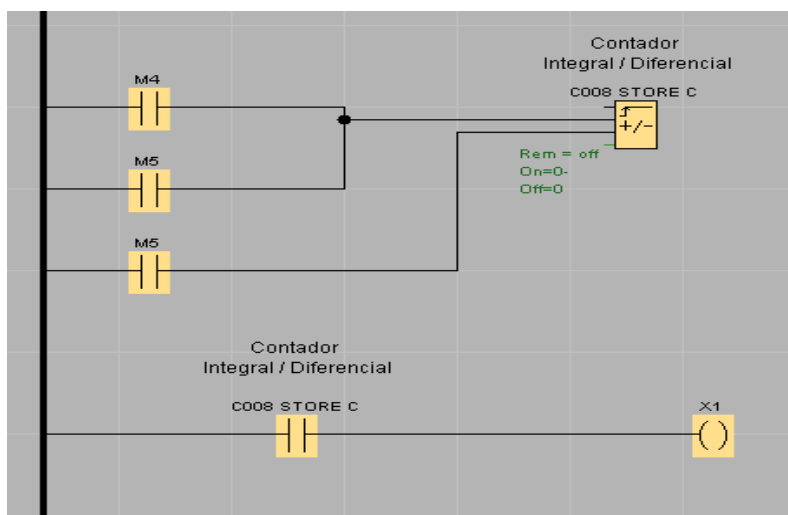


**Figura 18 - Representação do Tempo Diferencial**

Caso o resultado da diferença entre AX e AY do comparador SF003 seja menor ou igual à zero, significa dizer que, é enviado um sinal lógico de nível 0 para o verificador de erro M1, sendo assim não será ativada a comparação XOR (M1, M4) da Figura 17, mas será ativada o IM4, pois o verificador M1 contato normalmente fechado, deverá ser ativado como mostrado Figura 18, será ativada a entrada a comparação XOR (IM4, M5), nessa caso a saída seja de nível lógico 1, então é enviado o sinal lógico para as funções de retardo de ligamento que são configurados e percorrido o tempo diferencial, são eles T018 e T012. Ao término do tempo diferencial é enviado o sinal lógico para M5, que envia um sinal ao XOR (IM4, M5), ao M5 da Figura 19 tanto para o OR (M4, M5) quanto para a entrada Dir do contador C008 na, o sinal do XOR (IM4, M5) é para reiniciar o tempo diferencial caso o IM4 ainda esteja enviando o sinal lógico de nível 1, esse mesmo sinal é enviado para M5 da Figura 19, fazendo com que a cada vez que o C008 é acionado por suas entradas Cnt e Dir após o tempo diferencial é

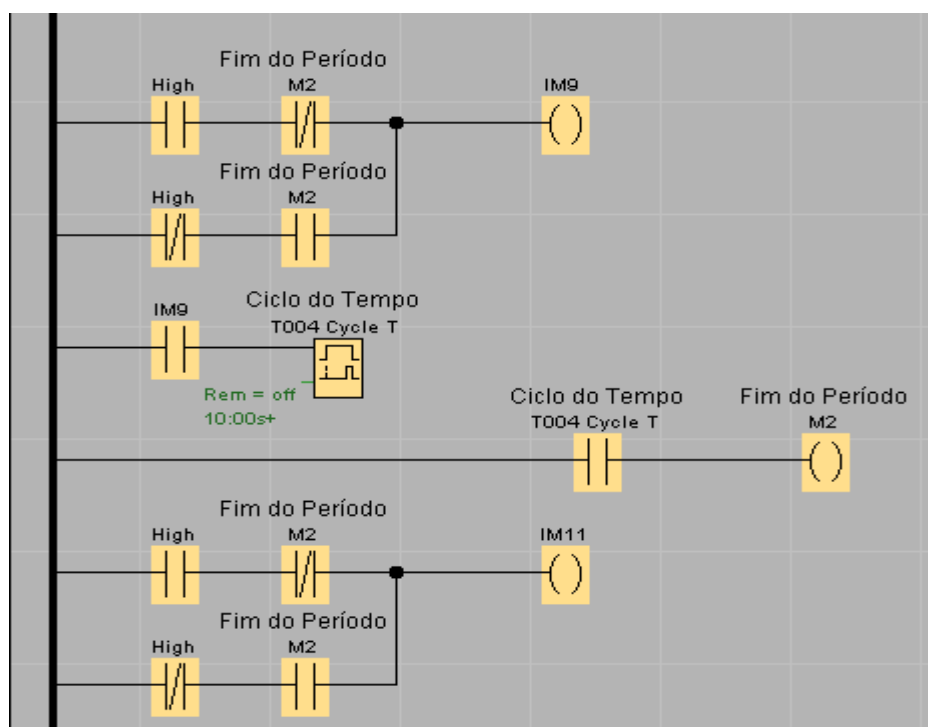
decrementado 1 do contador, o valor de C008 é o que define o tempo no Temporizador Integral/Diferencial T010 *INTDIF T* na Figura 21.

A Figura 21 é mostra o contador sendo ativado pela ação integral (M4) e diferencial (M5). Conforme já explicado anteriormente ao ativar o contato normalmente aberto M4 o contador é incrementado e se ativado o contato normalmente aberto M5 o contador será decrementado. O valor do Contador Integral/Diferencial é transformado em tempo para o temporizador Integral / Diferencial.



**Figura 19 - Representa a Contador Integral/Diferencial**

A representação da Figura 20 mostra o ciclo total do tempo de acionamento da saída. Esse tempo é subtraído pelos tempos proporcional e integral/diferencial.

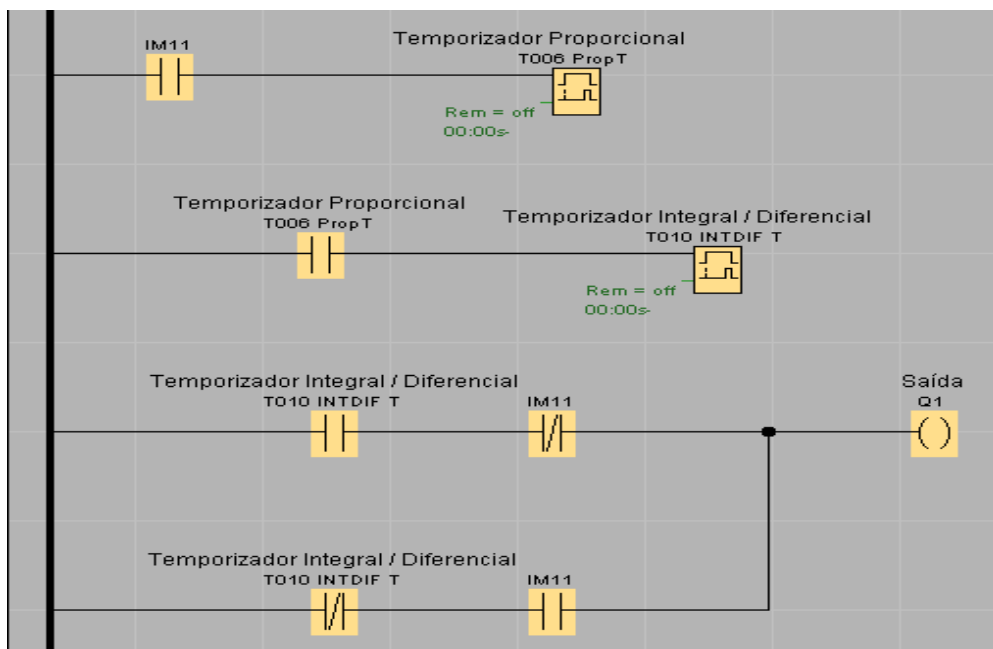


**Figura 20 - Representa o Ciclo do tempo de acionamento da saída**

A princípio, é feita uma comparação XOR (High, M2) onde se for aceita a comparação, a bobina IM9 é acionada ativando o contato normalmente aberto que está

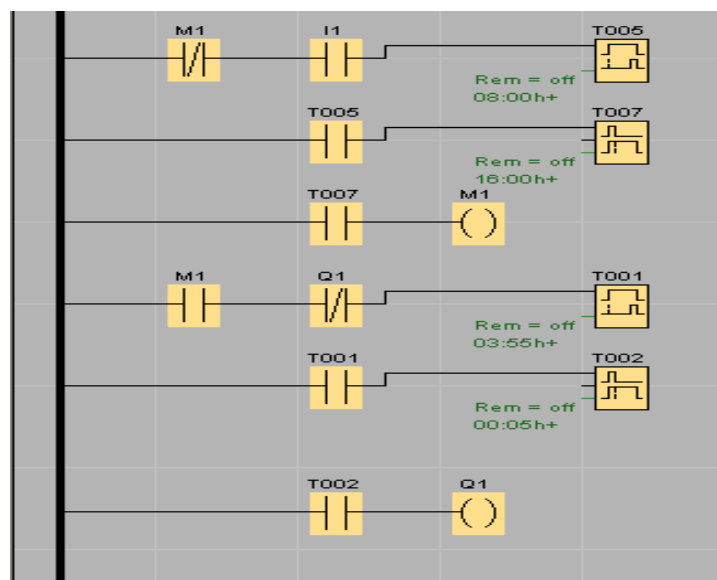
conectado ao Temporizador T004 *Cycle T* com isso inicia-se o ciclo do tempo de acionamento da saída. Após o tempo acabar é o M2 que significa fim do ciclo do tempo, então é enviado um sinal para o comparador XOR (High, M2) que reiniciará o T004 *Cycle T* e enviará o sinal para a bobina IM11.

Na Figura 21 tem-se a representação da soma dos tempos de T006 *PropT* + T010 *INTDIF T* e ativação da saída Q1.



**Figura 21 - Representa o acionamento da saída após a soma dos tempos PID.**

Após acionar o contato normalmente aberto IM11, o temporizador proporcional T006 é ativado. Após a espera de T006, é ativado o temporizador Integral/Diferencial T010. Decorrido o tempo de T010 é feita à comparação XOR (T010, IM11) para que o ciclo de tempo de acionamento seja reiniciado. Observa-se que a saída é ativada por IM11 e interrompida pelo T010. Na Figura 22, tem-se a implementação da irrigação automática.



**Figura 22 - Implementação da Irrigação**

A irrigação, segundo (FREITAG,2007), irá ocorrer quatro vezes ao dia sendo que serão 5 minutos de irrigação para quatro horas de espera, sendo que não poderá ocorrer irrigação entre 11h00min e 15h00min. Sendo assim a irrigação será das 06h55min as 07h00min, das 10h55min as 11h00min, das 14h55min as 15h00min e das 18h55min as 19h00min.

Caso a condição AND (M1,I1) seja aceita, então o T005 On/Delay de 8h00min é ligado fazendo com que se tenha 8 horas de espera no sistema de irrigação e logo após a espera, é ativado o T007 Off/Delay de 16h00min, assim começa a contar as horas de trabalho do sistema, como se inicia com 4h00min de espera, essa espera é somada às 8h00min de espera anterior tornando 12h00min de espera sem qualquer irrigação e 12h00min com quatro irrigações de 0h05min. Concluindo então que serão 12 horas de espera e 5 minutos de trabalho de 3h55min em 3h55min formando assim 12 horas com trabalho.

Quando T007 Off/Delay é ativado primeiramente ativa-se T001 On/Delay de 3h55min, logo após essa espera é ativado o T002 Off/Delay de 0h05min que manda o sinal que aciona a irrigação. Ao término dessa operação é emitido um sinal ao comparador AND (M1, Q1), caso este M1 estiver enviando sinal, significa que irá refazer toda a operação de 3h55min de espera e 0h05min de irrigação, caso o M1 não esteja enviando sinal então é reativado o T005 On/Delay de 8h00min.

## 5. Conclusão

A respeito dos objetivos propostos para o projeto, conclui-se que das variáveis analisadas identificaram-se duas como principais: temperatura e umidade. Aqui, fez-se o estudo sobre os princípios de funcionamento do Controlador Lógico Programável (CLP) com ênfase no funcionamento, estrutura, linguagem de programação, estudou-se também sobre a teoria de controle e automação, pois ela deu suporte para os princípios básicos do projeto desenvolvido.

Com a ferramenta LOGO!SOFT Comfort Version V6.0.17, foi realizada a simulação da realimentação com entradas analógicas (sensores) e saídas (atuadores) formando uma malha fechada de modo a “forçar” o ambiente da estufa ser alterado pelo controlador.

Implementou-se a lógica do controlador PID no simulador CLP para ter uma maior precisão na influência sobre as variáveis passíveis de controle na estufa de eucalipto. Implementou-se também um esquema de irrigação para manter as mudas sempre aguadas nos horários adequados.

Como encerramento recomenda-se para trabalhos futuros uma implementação física do projeto, podendo também adicionar um sistema de monitoramento em tempo real, usando aparelhos como, por exemplo: celulares, *smartphones*, computadores de mão, etc. para receber mensagens com dados sobre a estufa controlada.

Recomenda-se também a análise e projeto dos ganhos do controlador para melhorar o desempenho no controle das variáveis.

## Referências

**ALQUATI, Mário e SANTOS, José Eli dos**, Apostila de Automação, Fundação Universidade Federal do Rio Grande do Sul Colégio Técnico Industrial, 2005.

**ARACRUZ**, online. Site Aracruz - Celulose de Eucalipto - Disponível na internet via WWW.URL:<[http://www.aracruz.com.br/show\\_prd.do?act=stcNews&menu=true&id=107&lastRoot=16&lang=1/](http://www.aracruz.com.br/show_prd.do?act=stcNews&menu=true&id=107&lastRoot=16&lang=1/)> Acesso em: abril/2009

**FREITAG, Ângela Simone**, Freqüências de irrigação para Eucalyptus grandis e Pinus elliottii em viveiro, Universidade Federal de Santa Maria Santa Maria, 2007.

**OGATA, K.** Engenharia de Controle Moderno. Ed. LTC, Rio de Janeiro, 1998.

**OLIVEIRA, Júlio César P.**, Controlador Programável, Makron Books, 2002

**SOUZA, Geraldo Teles de**, Controle e Automação Industrial, Escola Técnica Estadual Pedro Ferreira Alves, Julho 2004.

**ZANATA FLORESTA**, online. Site Zanata Floresta - Modelos de estufa. Disponível na internet via WWW. URL: <[http://ibge.gov.br/home/presidencia/noticias/noticia\\_visualiza.php?id\\_noticia=1259&id\\_pagina=1](http://ibge.gov.br/home/presidencia/noticias/noticia_visualiza.php?id_noticia=1259&id_pagina=1)> Acesso em: junho/2009



# Desenvolvimento da Modelagem de uma Ferramenta para Gerenciar aplicações SaaS

**Valéria Pereira Mota, Fernando Luiz de Oliveira, Isabel Pires da Silva, Roberto Mayer**

Sistemas de Informação/Centro Universitário Luterano de Palmas  
Av. Gov. Siqueira Campos, 1501 Sul, Caixa Postal nº 160 - CEP 77054-970  
vpmota@gmail.com, nandoluiz.br@gmail.com, belpires@gmail.com, rocmayer@gmail.com

**Resumo.** *Software as a Service (SaaS) é um modelo utilizado como estratégia para a distribuição de software como serviço. Este novo padrão poderá transformar a forma como as pessoas constroem, vendem, compram e utilizam software. SaaS agrega várias vantagens aos cliente e aos fornecedores, tais como: implementação rápida, menor custo inicial, menor custo de manutenção, facilidade de uso, risco reduzido, segurança, contrato entre fornecedor e cliente (Service Level Agreements-SLA), entre outras. Este artigo apresenta a fase inicial de uma modelagem para um sistema de gerenciamento de aplicações SaaS. Além disso, descreve serviços importantes como escalabilidade, segurança e SLA para a implementação de SaaS.*

**Palavras chave:** Software as a Service, Gerenciamento de Aplicações SaaS, Comercialização de Software.

## 1. Introdução

*Software as a Service (SaaS)* é uma tendência na indústria de software, que se caracteriza por ser um modelo de negócio que oferece às organizações o acesso centralizado de suas informações, por meio da hospedagem e acesso pela internet, a um custo menor em relação ao modelo de software tradicional. Esse novo modelo de negócios provoca mudanças fundamentais no desenvolvimento de software, em direção a um modelo centrado na prestação de serviços. Além disso, esse modelo traz vários benefícios aos clientes, tais como: redução de custo, agilidade, acessibilidade, escalabilidade, flexibilidade e continuidade. Desta forma, o fornecedor possibilita a redução das despesas que o cliente teria que arcar ao trazer para sua empresa os serviços que antes faziam parte do ambiente do cliente (licenciamento do software, treinamento para a utilização, servidor para hospedar a aplicação, entre outros), possibilitando o benefício de utilizar o sistema, sem que haja elevados custos iniciais.

Neste modelo, há várias formas de pagamento, pelas quais não se cobra o licenciamento da propriedade intelectual, e sim o uso dos recursos do software. Em alguns casos, o uso pode ser gratuito, e isso ocorre quando o fabricante do software tem outras formas de obter receita (anúncios, por exemplo). Assim, o fornecimento da aplicação de SaaS é feito por quem desenvolve o software, ou por meio de um intermediário, que objetiva juntar as ofertas de SaaS de terceiros e anunciar em sua página local, a fim de vender os serviços existentes no software. Portanto, o cliente não compra a licença do software, mas apenas o serviço, sendo o pagamento feito, geralmente, por meio de assinatura, continuando com o fornecedor do software a

responsabilidade do gerenciamento total da aplicação. Dentro deste contexto, este artigo apresenta uma modelagem inicial de um sistema de gerenciamento de SaaS, sendo o objetivo principal o de modelar um sistema para gerenciar e distribuir vários tipos de sistemas.

## 2. Software as a Service (SaaS)

Segundo Chong & Carraro (2006, pag. 3), SaaS pode ser definido como "software desenvolvido como um serviço hospedado e acessado por meio da internet". Com base nesta informação, a principal diferença do SaaS é o local em que os algoritmos da aplicação estão localizados, como é desenvolvido e acessado. Esse serviço inclui uma diversidade de serviços e softwares que atende vários critérios do cliente por meio da internet. Assim, no modelo SaaS, o fornecedor passa a hospedar o software em um provedor externo. Consequentemente, a responsabilidade com infraestrutura e gerenciamento passa a ser de quem produziu/produz a aplicação. Porém, acredita-se que o custo é reduzido por meio da economia de escala. Um fornecedor de SaaS com um número grande de clientes significa que cada cliente é considerado responsável por somente um percentual do custo de um servidor. De acordo com Chong & Carraro (2006, pag.7), “um aplicativo de SaaS de linha de negócios em um *farm*<sup>1</sup> com cinco servidores com carga balanceada poderá dar suporte a 50 clientes de tamanho médio, o que significa que cada cliente seria responsável por somente um décimo do custo de um servidor.”

Melo (2007, pag. 20) apresenta o processo de desenvolvimento da aplicação de software como serviço, e enfatiza que as fases de desenvolvimento e testes são as mesmas em relação com o desenvolvimento tradicional, apresentando como diferença o local de onde a aplicação será operada. Analisando a figura 1, é possível verificar que no modelo de software como serviço a fase de operação do sistema passa a ser depois das fases de desenvolvimento e teste, diferentemente do modelo tradicional (parte laranja da imagem), que ainda teria que gastar tempo e custo com instalação, qualidade e treinamento. No modelo SaaS, essa passa a ser última fase, porque o software é operado na máquina, em um ambiente controlado pelo fornecedor. O usuário acessa normalmente a internet utilizando o software que foi disponibilizado pelo servidor, reduzindo pela metade os passos deste ciclo. Na medida em que houver demandas por melhorias e a equipe de desenvolvimento as realizar, automaticamente a operação fará uso da versão mais recente a cada execução. Com isso resolve-se o problema de distribuição do software.

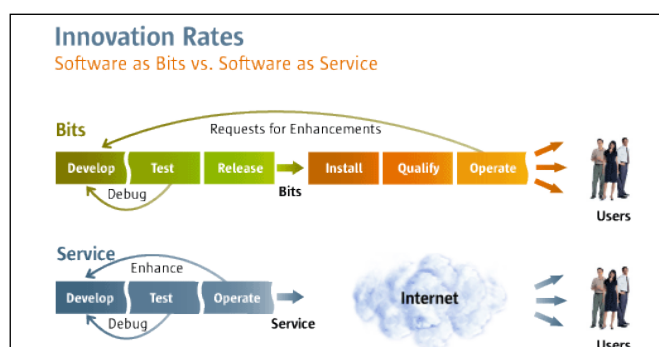


Figura 23 - Modelo de aplicação de Software tradicional e SaaS (MELO, 2007)

<sup>1</sup> Farm são várias unidades de servidores.

No modelo de desenvolvimento de software tradicional, a fase de aquisição ocupa muito o tempo da empresa na escolha de um software, porque além de esperar os módulos a serem implementados, há também o custo com testes, que é, praticamente, o mesmo que de uma instalação completa. Depois da avaliação do software, geralmente o cliente solicita uma customização da aplicação, que são adaptações específicas para o ambiente do cliente, alguma funcionalidade ou forma de operação da empresa que não coincida com a implementação original do software. Este é um problema que se constata com a maioria das empresas nacionais de software, as quais afirmam ter produtos, mas quando a customização é feita na mesma linguagem de programação em que foi desenvolvido o software originalmente, gera muita mão de obra em trabalho de customização. Quando se fala de software como serviço, isso passa a ser impossível, porque a execução do software se dá na máquina da empresa que fornece a aplicação. Se for feita uma modificação no código fonte, certamente vai repercutir para todos os usuários de todas as redes. Por isso é que se deve mudar a forma de fazer esse trabalho, o que se pode chamar de “configuração”. O desenvolvedor tem que ter a capacidade de prever, dentro do conjunto de clientes que se quer atender, quais são as “customizações” que poderão ocorrer e ter a previsão dentro do software já feito para que, conforme os usuários solicitem, a ação se resume a uma configuração. Essas “personalizações” podem ser desde algo simples como uma opção de menu que marca ou desmarca ou, funcionalidades mais complexas. Isso exige um bom nível de qualificação dos profissionais de desenvolvimento que vão implementar o software, o que pode ser um problema para as empresas produtoras. Se na empresa não há profissionais com este perfil, é necessário investir em qualificação profissional para melhorar a qualidade dos softwares produzidos.

Além disso, na aquisição do SaaS, a empresa compradora vai precisar de um fornecedor confiável, incluindo contratos e serviços de segurança, os clientes normalmente exigem o que se chama de SLA's. Trata-se de um acordo entre o fornecedor de serviços e um cliente sobre nível de serviços, onde o fornecedor se compromete a ressarcir a empresa que comprou o serviço pelo tempo durante o qual a aplicação não vai estar disponível. Desta forma, ao se projetar um software para ser comercializado no modelo SaaS, deve-se ter a preocupação com uma série de questões novas como, por exemplo, estender o modelo de segurança tradicional para garantir a confidencialidade das informações, disponibilidade em 99,99% da aplicação na web de acordo com SLA, modelagem dos dados, migração dos dados, escalabilidade e alocação dos recursos. Todas essas áreas devem ser bem entendidas pelo profissional de TI, pois são áreas que tiveram algumas mudanças e são indispensáveis para desenvolver um SaaS de qualidade. A seguir será apresentado parte da modelagem da ferramenta de gerenciamento de aplicações SaaS.

### **3. Modelagem do Gerenciador de Aplicações SaaS**

O objetivo do gerenciador é que ele seja o responsável pela administração dos recursos comuns a todas as aplicações SaaS no controle de (acessos, logs, contas de usuário, controle de contratos, controle de sistemas, entre outros). E para fazer a comunicação do gerenciador com as aplicações acredita-se que será necessário criar uma camada de serviço de todas as funcionalidades do sistema de gerenciamento SaaS, esses serviços

podem ser por exemplo, publicados como WebService, e a comunicação entre eles será feita por meio de interfaces de serviço utilizando SOA.

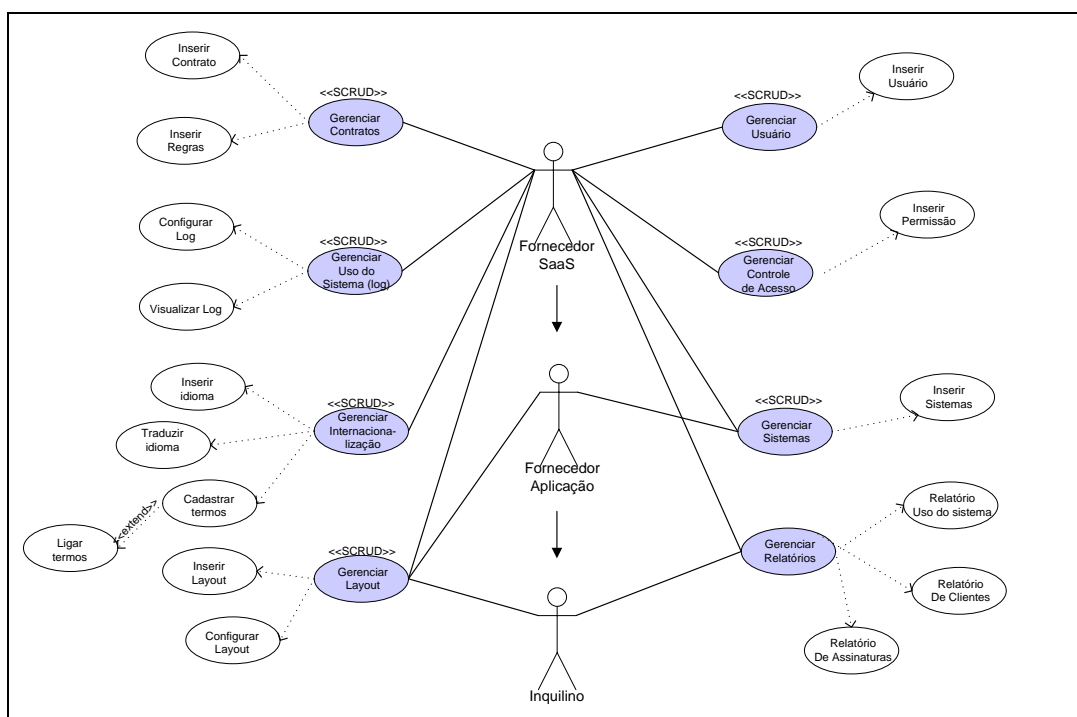
Para desenvolver a modelagem, inicialmente, para desenvolver a modelagem do gerenciador de aplicações SaaS, foi necessário identificar os atores envolvidos nas atividades. Para isto, foi necessário realizar diversas entrevistas com profissionais que estudam e acompanham o crescimento de SaaS. Posteriormente, foi feita uma coleta de requisitos abrangente para descobrir as principais funcionalidades do sistema e, após isso, foi desenvolvido o diagrama de caso de uso para representar o modelo do negócio do gerenciador SaaS. Abaixo (Tabela 1) segue a especificação dos atores envolvidos no sistema.

**Tabela 1** – Descrição dos atores do sistema de gerenciamento SaaS

Atores do sistema	
1 - Fornecedor SaaS	Empresa que fornece o ambiente SaaS.
2-Fornecedor da Aplicação	Empresa fornecedora de aplicações que utiliza o ambiente SaaS.
3 – Inquilino	Cliente que utiliza aplicações do ambiente SaaS, disponibilizadas por determinado fornecedor (2).
4 – Usuário Final	Usuário que acessa as funcionalidades do sistema do inquilino.

Obs:  
Se 1) atua apenas como fornecedor de ambiente SaaS, seu único cliente é 2).  
Se 1) atua também como revenda de 2). Seu único cliente é 3).  
Se 1) atua como fornecedor de ambiente SaaS e revenda de 2), seus clientes são 2) e 3) de maneira geral sempre vai ter usuários para 2) e 3).  
Se 1) atua apenas como fornecedor de ambiente SaaS, ele só gerencia usuários tipo 2). Nas outras situações ele gerencia usuários dos tipos: 2) e 3).

O diagrama de caso de uso (Figura 2), representa o modelo de negócios do gerenciador SaaS. Esta modelagem está sendo desenvolvida separadamente do modelo da aplicação. Isto porque a empresa fornecedora pode disponibilizar vários sistemas e, caso seja necessário incluir outros sistemas no modelo de negócios, a estrutura será totalmente aproveitada, sem a necessidade de nenhuma alteração. A ideia é ter uma base de dados genérica para gerenciar sistemas e fornecedores para qualquer aplicativo. O Diagrama de Casos de Uso será apresentado abaixo.



**Figura 24 – Caso de uso Gerenciador de Sistemas SaaS**

A tabela 2, apresentada a seguir, lista os requisitos iniciais que deverão ser desenvolvidos para que os objetivos deste trabalho sejam atingidos.

**Tabela 2 – Tabela de levantamento de requisitos**

Levantamento de Requisitos		
<b>Gerenciar Controle de Acesso</b> <ul style="list-style-type: none"> <li>Inserir permissão de sistema</li> <li>Alterar permissão de sistema</li> <li>Remover permissão de sistema</li> <li>Buscar permissão de sistema</li> </ul>	<b>Gerenciar Usuário</b> <ul style="list-style-type: none"> <li>Cadastrar usuário</li> <li>Alterar usuário</li> <li>Remover usuário</li> <li>Buscar usuário</li> </ul>	<b>Gerenciar Sistemas</b> <ul style="list-style-type: none"> <li>Inserir sistema</li> <li>Alterar sistema</li> <li>Remover sistema</li> <li>Buscar sistema</li> </ul>
<b>Gerenciar Uso do sistema (tabela de log)</b> <ul style="list-style-type: none"> <li>Configurar Log</li> <li>Visualizar Log</li> </ul>	<b>Gerenciar contratos</b> <ul style="list-style-type: none"> <li>Inserir contrato</li> <li>Alterar contrato</li> <li>Remover contrato</li> <li>Finalizar Contrato</li> <li>Buscar Contrato</li> </ul>	<b>Gerenciar internacionalização</b> <ul style="list-style-type: none"> <li>Inserir idioma</li> <li>Remover idioma</li> <li>Configurar idioma</li> <li>Cadastrar termos</li> <li>Validar interface</li> <li>Traduzir interface</li> </ul>
<b>Gerenciar Layout</b> <ul style="list-style-type: none"> <li>Inserir layout</li> <li>Remover layout</li> <li>Alterar layout</li> </ul>	<b>Gerenciar Relatórios</b> <ul style="list-style-type: none"> <li>Gerar Relatório Uso do sistema</li> <li>Gerar Relatório de assinaturas</li> <li>Gerar Relatório de fornecedores</li> <li>Gerar Relatório por filtro</li> </ul>	

O aumento da terceirização na área da TI vem provocando uma nova alteração no relacionamento entre os compradores de softwares e fornecedores de serviços por meio da aplicação entre contratos e acordos Service Level Agreement (SLA). Com esse acordo, os clientes passam a ter garantias de serviço de qualidade, prazo e estrutura oferecidos por seus fornecedores. Uma SLA é um acordo entre o fornecedor de serviços e um cliente sobre nível de serviços, onde o fornecedor se compromete a ressarcir o cliente pelo tempo em que a aplicação não estiver disponível. Há dois tipos de serviços de fundamental importância para a implantação de um SLA. De acordo com Ligocki (2008, p. 39), esses serviços são referentes “à própria criação do contrato e um aplicativo de auxílio para geração do SLA”. O contrato deve ser o mais detalhado possível, com o intuito de atender e favorecer tanto o cliente quanto o fornecedor. O tipo de cobrança deve ser cadastrado no sistema utilizando a funcionalidade inserir regras na tabela contrato. Basicamente um contrato deve ter a descrição do serviço, o acordo do nível de serviço, a descrição de todos os itens, a disponibilidade do sistema no ar, a disponibilidade dos dados caso o cliente queira trocar de fornecedor. Enfim, uma série de itens para deixar claro para ambas as partes (fornecedor) e (cliente).



Serviço: (Descrição do Serviço, ex: Fornecimento de sistema de advocacia pela web).

Acordo de Nível de Serviço entre (Empresa X) e (Cliente X), referente ao serviço descrito acima fornecido através do (contrato n.º XXX) de (data XX/XX/XXXX), nas seguintes condições:

(descrever os índices de forma sucinta)

Este acordo firmado entre as partes tem validade a partir de sua assinatura, enquanto o contrato de prestação de serviço for vigente entre as partes, e representa o acordo comum entre elas para reger o nível de eficiência dos serviços oferecidos.

A mínima alteração em eficiência, uso ou qualquer outra informação, somente terá validade após a alteração, e assinatura, deste acordo entre as partes e de comum acordo.

As partes abaixo assinadas aceitam os termos aqui descritos e oferecidos, como regra de fornecimento, aceitando estes indicadores como validadores para penalidades e multas em caso de falhas no fornecimento.

Assinam:

Fornecedor: \_\_\_\_\_

Cliente: \_\_\_\_\_

Data: XX/XX/XXXX

Exclusões e Limites de uso:

**Figura 25** – Modelo de contrato SLA

No aspecto da escalabilidade, à medida que o usuário utilizar a aplicação, o espaço no banco de dados pode ficar pequeno e o desempenho lento. Existem algumas técnicas para minimizar isto como, por exemplo, utilizar a replicação dos dados para tratar a escalabilidade. Replicar dados significa criar uma cópia do banco de dados para

outro local, e manter uma cópia sincronizada com a original. É uma técnica que possibilita aumentar a disponibilidade da base de dados. Pode ser usada com uma aplicação tradicional ou com SaaS indistintamente. Escolheu-se o tipo replicação para demonstrar a escalabilidade utilizando a ferramenta SQL Server 2005. Esta é uma ferramenta simples para configurar a replicação dos dados.

Foi feito um pequeno teste de um banco com 2 tabelas para implementar uma replicação de dados do tipo *Merge Replication* e como utilizá-la usando uma base de dados *SQL Server Compact Edition*. Inicialmente é necessário criar a base de dados com suas respectivas tabelas e seus relacionamentos. Foi criado o banco chamado ‘ORIGEM’, nesta base havia duas tabelas: uma ‘INQUILINO’ e outra ‘SISTEMA’ como mostra a figura 3 abaixo.

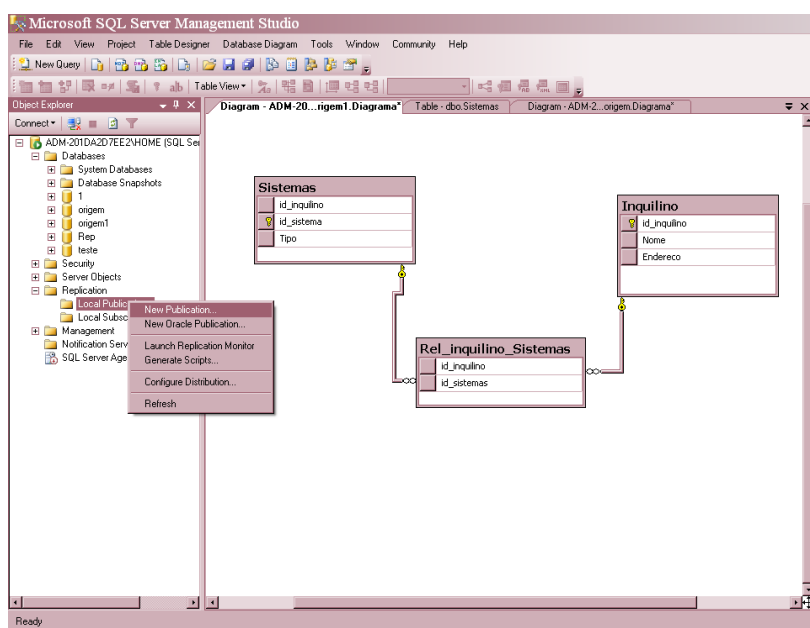


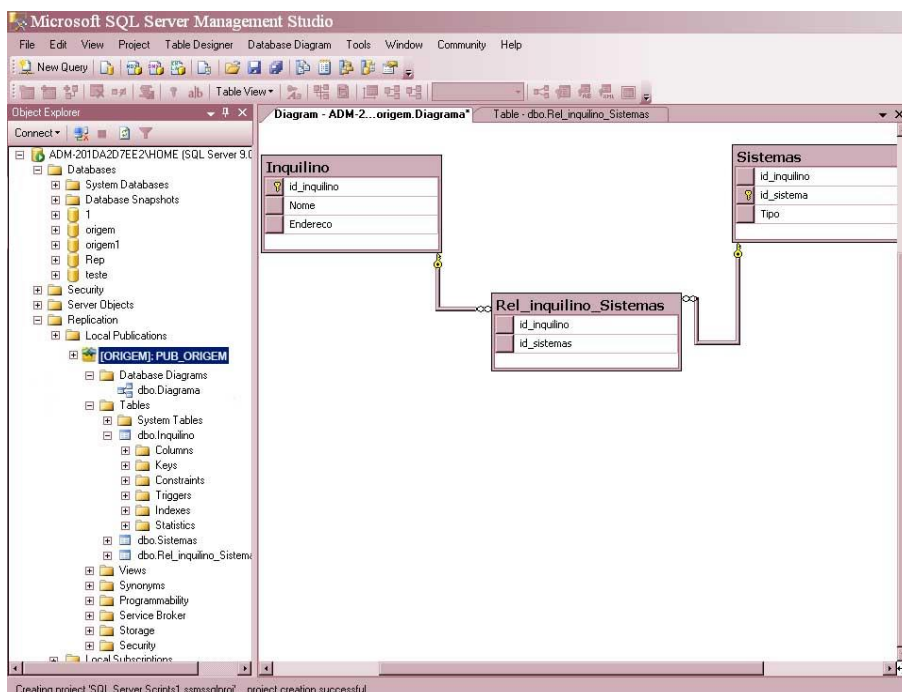
Figura 26 – Replicação de dados utilizando SQL Server

O SQL irá solicitar qual servidor será feito a publicação para distribuir suas replicações. Após fazer todas as configurações para a replicação, é iniciado o processo de publicação. Deve-se escolher o banco de dados para replicar os dados, neste caso a base é “ORIGEM”. Faz-se necessário escolher qual tipo de replicação (*Snapshot*, *Transactional* ou *Merge Replication*) que o SQL Server oferece. Para o teste foi escolhido o tipo *merge replication*. Cada um serve para determinado tipo de problema, como por exemplo:

- O *Snapshot Replication*: faz uma fotocópia do banco de dados e compartilha a fotocópia com os seus usuários, esse processo é demorado e consome vários recursos do servidor. Não é recomendado para bases de dados que sofrem muitas alterações.
- O *Transactional Replication*: é considerada uma replicação maleável em relação a Snapshot, nesse tipo de replicação o agente gerencia a base esperando por alterações e transmite apenas essas alterações para os usuários, de maneira otimizada e rápida.

- *Merge Replication*: nesta replicação, o usuário e o editor efetuam alterações independentes na base de dados, as duas bases podem trabalhar desconectadas. No momento que as bases se conectarem novamente o agente *merge agent* verifica as alterações feitas na publicação e em cada um dos usuários, mescla as alterações de maneira que os dados fiquem consolidados.

Após várias configurações no SQL Server, a replicação do banco foi criada com sucesso (figura 5).



**Figura 27** – Banco replicado

Além de poder ser oferecida escalabilidade aos inquilinos, a segurança dos sistemas deve ser eficiente, sendo necessário conter serviços de segurança tais como: do sistema operacional, de aplicativos, de autenticação, de comunicação, de monitoramento e gerenciamento, em vários níveis, além de certificação digital. Todos esses serviços são importantes para impedir o procedimento não autorizado de aplicativos que protegem os recursos contra uso não-autorizado; garantir que o usuário seja de fato a pessoa que afirma ser; garantir que as informações transmitidas pela rede não sejam modificadas; monitorar as tentativas de violação da segurança; melhorar continuamente as técnicas de segurança de acordo com possíveis ameaças; gerenciar usuários com vários tipos de controle de acesso e implementar certificação de acordo com a norma ISO.

Existem várias possibilidades para a implementação desses serviços de segurança tais como a utilização do protocolo Kerberos, que adiciona um nível maior de segurança à rede. Ou utilizar o Hasp SRM SaaS Pass. O serviço fundamenta-se em uma aplicação cliente instalada no computador do usuário final que comunica pela internet com a aplicação do fornecedor do serviço. Os serviços de login e o módulo de Web Service estão disponíveis com o HASP SEM sem qualquer custo adicional para testes e



se encontra disponível no site <[www.aladdin.com/hasp/solutions/saas-solution.aspx](http://www.aladdin.com/hasp/solutions/saas-solution.aspx)>. Outro serviço utilizado para cenários de autenticação, autorização e controle de acesso é a Hol da Enterprise Library, que oferece uma variedade de blocos de aplicação e bibliotecas para auxiliar nas questões de criptografia, acesso a dados, logging, auditoria, segurança de autorização, entre outros (BANAGOURO, 2009, on-line).

Enfim, são diversas opções utilizadas para implantar serviço de segurança em uma organização, diante de tantas possibilidades, pode ficar difícil para os desenvolvedores escolher uma opção em relação às questões de segurança. Portanto, é viável que as organizações desenvolvam o documento de políticas de segurança para analisar melhor todas as questões antes de aderir a alguns serviços de segurança.

Também começam a surgir no mercado plataformas como o Google, Microsoft Azure, Force, Zooho etc. Essas plataformas são data centers que funcionam como provedores de computação na nuvem e oferecem aos desenvolvedores a possibilidade de criar e rodar aplicativos no ambiente cloud como: hospedagem da aplicação, ambiente de desenvolvimento, serviço de segurança, escalabilidade etc. Segundo Mckendrick (2008, pag. 1), este mercado tem sido mais utilizado entres as empresas de menor porte, mas a tendência é que outras organizações aderem cada vez mais os serviços de computação nas nuvens por causa da redução do custo dos softwares, isso tem atraído as empresas maiores. De acordo com Mckendrick (2008, pag. 8), os gestores de TI estão seguros em relação ao nível de segurança de dados oferecido pelas plataformas, diz ser tão bom ou melhor que muitos serviços que outros departamentos de TI poderiam oferecer.

O software a ser modelado no decorrer deste trabalho se encaixa no tipo de software como serviço “horizontal”. De acordo com Melo et al. (1997, p. 112), esse tipo de software é de uso geral, a distribuição geralmente é feita em “larga escala e a preferência dos consumidores se dá pela marca e reputação das empresas”. Geralmente, esses softwares servem para qualquer segmento de mercado.

#### **4. Considerações Finais**

Observou-se que quando se projeta um software para ser comercializado no modelo SaaS, deve-se ter a preocupação com uma série de questões novas como, por exemplo, estender o modelo de segurança tradicional para garantir a confidencialidade das informações, oferecer uma disponibilidade de 99,99% da aplicação na web de acordo com SLA, planejar a modelagem dos dados, prover mecanismos de migração dos dados, garantir a escalabilidade e a alocação dos recursos. Todas essas áreas devem ser bem entendidas pelo profissional de TI, pois são áreas são indispensáveis para desenvolver um SaaS de boa qualidade. O objetivo do sistema, quando implementado, é dispor de uma ferramenta que administre um modelo de entrega de software em larga escala, envolvendo controle de acesso de usuários, gerenciamento de sistemas, gerenciamento SLA, configuração de interface e internacionalização. Em que a empresa fornecedora se responsabilize pela infra-estrutura, manutenção e suporte do produto, disponibilizando-o como um serviço, por meio da Web. A modelagem desse sistema está em seu estado inicial, porém, percebe-se a complexidade envolvida no desenvolvimento de uma ferramenta com esta finalidade. Será necessário criar uma camada de serviço de todas as funcionalidades de gerenciamento SaaS e publicar esses serviços como *WebServices*

para qualquer sistema poder utilizar. E a comunicação entre os sistemas poderá ser feita através de interfaces de serviço utilizando SOA – (*Service Oriented Architecture*).

Para trabalhos futuros sugere os seguintes trabalhos a seguir:

- Promover a implementação de todas as funcionalidades que serão descritas na modelagem e, com o desenvolvimento do sistema, o fornecedor terá possibilidades de administrar vários sistemas e também ter a possibilidade de comercializar o sistema diretamente com fornecedores de SaaS, com a isso, a distribuição de software poderá ser feita em larga escala.
- Após a implementação do sistema, inserir o sistema EVA Escritório Virtual de Advocacia no gerenciador de sistemas SaaS a fim de testar a comunicação das funcionalidades do gerenciador com o sistema EVA.
- No mercado atual não existem muitas ferramentas para gerenciamento de SLA, e a maioria das vezes é necessária a intervenção humana para obter o resultado dos relatórios da prestação de serviços (LIGOCKI, 2008, p. 40). Então, sugere-se que seja feita uma modelagem, a fim de promover o desenvolvimento do serviço de gerenciamento de SLA, a fim de suprir as necessidades do mercado.
- Promover, também, a possibilidade de ser gerado o desenvolvimento de uma solução de segurança referente ao controle de acesso, usuários, autenticação, certificação entre outros, que satisfaça as necessidades os fornecedores, inquilinos e usuários finais.
- É viável, também, fazer um estudo sobre escalabilidade, incluindo replicação de dados e particionamento nos diversos tipos de banco de dados, SQL Server, MySQL, Postgree, entre outros.

## 5. Referências Bibliográficas

(CHONG & CARRARO, 2006), CHONG, Frederick, CARRARO, Gianpaolo. Architecture Strategies for Catching the Long Tail. ©2009 Microsoft Corporation. Disponível em: <[http://www.cistrattech.com/whitepapers/MS\\_longtailsaas.pdf](http://www.cistrattech.com/whitepapers/MS_longtailsaas.pdf)>. Acesso em: 8 de março de 2009.

(CHONG, CARRARO & WOLTER, 2006) CHONG, Frederick, CARRARO, Gianpaolo, WOLTER Roger. Multitenant Data Architecture. Microsoft Corporation. Disponível em: <<http://www.wrappedapps.com/files/729.pdf>>. Acesso em: 10 de março de 2009.

(BANAGOURO, 2009, online) BANAGOURO, Michel. Hands-On Labs for Enterprise Library 4.1. Disponível em: <<http://sharpcode.com.br/blogs/mbanagouro/archive/2009/04/21/hands-on-labs-for-enterprise-library-4-1-march-2009.aspx>>. Acesso em: 26/09/2009.

(MCKENDRICK, 2008, online) MCKENDRICK, Joe. The 10 Paradoxes of Today's Data Centers. Disponível em: <<http://www.dbta.com/Authors/3538-JoeMcKendrick.htm>>. Acesso em: 25/09/2009.

(MELO et al, 2007) MELO, Cássio A., ARCOVERDE, Eduardo F., MORAES, Éfrem A., PIMENTEL, João H., FREITAS, Rodrigo. Software como Serviço: Um Modelo de Negócio Emergente. Centro de Informática – Universidade Federal de Pernambuco (UFPE). Publicado em 2007. Disponível em: <<http://www.cin.ufpe.br/~jhcp/publica/jhcp-saas.pdf>>. Acesso em: 19 de março de 2009.

# Estimativa de Tamanho de Software Utilizando APF e a Abordagem NESMA

Werley Teixeira Reinaldo, Cristina D’Ornellas Filipakis

Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas  
(CEULP/ULBRA)  
Av. Gov. Siqueira Campos, 1501 Sul, Caixa Postal 160 - 77054-970 - Palmas-TO  
{reinaldotx, filipakis}@gmail.com

**Resumo.** *Este trabalho tem como objetivo realizar a estimativa de tamanho de um software utilizando a técnica APF e o método de contagem indicativa da NESMA. A APF mede o tamanho do software de acordo com o ponto e vista do usuário e sobre uma perspectiva não tecnológica. A contagem indicativa visa oferecer o tamanho do software logo nas fases iniciais de concepção do sistema, em que as características ainda não estão bem definidas. O software a ser mensurado é o Sistema de Dados Estatísticos, uma ferramenta para difusão de conhecimento sobre aspectos físicos, demográficos e sociais dos municípios do estado do Tocantins.*

**Palavras chave:** contagem, estimativa, pontos de função

## 1. Introdução

Quando o software é planejado, estimativas de esforço, prazo (cronograma) e custo (orçamento) devem ser realizadas. A estimativa de software é executada no desenvolvimento e na manutenção de sistemas de informação e possui forte impacto na gestão do projeto, pois erros em estimativas (subestimar ou superestimar) podem causar transtornos à organização em termos de recursos alocados, perda de tempo e orçamento, ou mesmo tornar o projeto inviável.

O processo de estimativa de projetos de software envolve quatro atividades, em que é necessário estimar o tamanho do produto a ser desenvolvido, o esforço a ser empregado para sua implementação, a duração do projeto e o custo à organização. Segundo Hazan (2009, p. 26), o responsável pelas estimativas deve analisar os requisitos para garantir a qualidade do produto e então estimar o tamanho do projeto de software. O próximo passo é calcular o esforço necessário e então derivar as estimativas de prazo e custo com base nas estimativas de tamanho. Desta forma, a partir do cálculo do tamanho do projeto, é possível calcular todas as outras estimativas, de forma a identificar as necessidades de recursos financeiros e de pessoal, fazer compensações de custo, produtividade, qualidade e quantificar o impacto das mudanças no projeto de desenvolvimento.

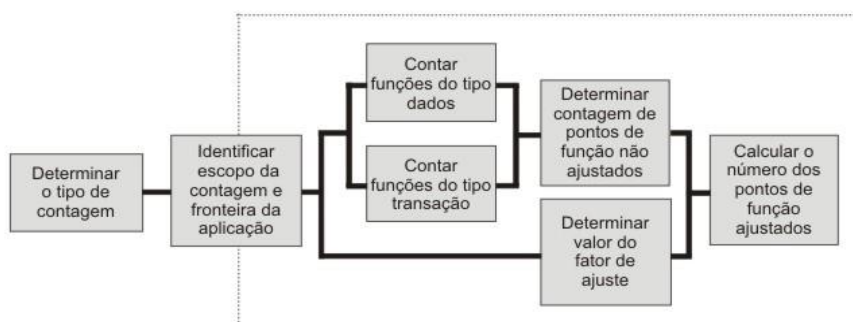
Há várias técnicas para mensurar o tamanho de um sistema, entre elas, a Análise de Pontos de Função – APF. A APF é uma medida padronizada e mantida por um Grupo Internacional de Usuários de Pontos de Função, o IFPUG. Juntamente com as abordagens propostas pela Associação Holandesa de Métricas – NESMA, a APF permite estimar o tamanho do sistema nas fases iniciais do projeto, em que as funcionalidades ainda não estão bem definidas. Assim, neste trabalho será realizada a contagem estimada em pontos de função do Sistema de Dados Estatísticos do Tocantins.

## 2. Análise de Pontos de Função – APF

A APF é a técnica que mede o tamanho funcional de um software, a partir de uma perspectiva funcional e independente da tecnologia adotada. É independente da linguagem, dos métodos de programação, hardware ou plataforma utilizada. Ou seja, a contagem de pontos de função de um sistema permanece constante em qualquer local que seja contado, diferenciando apenas a quantidade de esforço necessário para desenvolvê-lo. Ponto de função (PF) é a unidade de medida desta técnica.

Assim como o metro quadrado (m<sup>2</sup>) é uma unidade de medida de comprimento utilizado, por exemplo, na engenharia civil, para verificar a quantidade necessária de cerâmica por metro quadrado de um determinado espaço, os PF são unidades de medida de tamanho de software que podem ser utilizados para permitir a criação de métricas para verificar a produtividade e qualidade do produto de software. A partir de PF é possível medir o prazo, o esforço (quantidade de pessoas necessárias para desenvolver o produto) e o custo do desenvolvimento de um software.

O processo de contagem de PF envolve sete etapas. As duas primeiras fases abordam os tipos de contagem e o escopo da aplicação que será contada. As quatro etapas seguintes referem-se à contagem de requisitos funcionais e não funcionais. A última fase refere-se ao cálculo final em pontos de função da aplicação que está sendo mensurada. A figura 1 apresenta a visão geral do processo de contagem de pontos de função.



**Figura 1** - Visão geral do processo de contagem de pontos de função (VAZQUEZ, SIMÕES e ALBERT, 2003, p. 43)

### 2.1. Processo de Contagem

O primeiro passo do processo de contagem de pontos de função é a determinação do tipo de contagem. A contagem pode ser de três tipos: contagem de um projeto de desenvolvimento, contagem de um projeto de melhoria e contagem de uma aplicação. A contagem de um projeto de desenvolvimento é utilizada para dimensionar o tamanho de um projeto de desenvolvimento de uma nova aplicação. O projeto de melhoria ou manutenção é utilizado para dimensionar o tamanho em uma aplicação já existente. A contagem de uma aplicação é realizada sobre o sistema após ser finalizado e entregue.

Segundo Andrade (2004, p. 14) a fronteira da aplicação separa o projeto a ser contado dos usuários e das aplicações externas do domínio do projeto. Assim, da mesma forma que um lote não pode ser medido sem as fronteiras que delimitam sua área, a medição funcional de um sistema não pode ser calculada antes da determinação de seus limites.

O escopo da contagem define quais funções serão incluídas na contagem, se abrangerá um ou mais sistemas ou parte dele (VAZQUEZ, SIMÕES e ALBERT, 2003, p. 57). A finalidade do escopo é definir o que será medido, identificando quais funções serão incluídas no processo de contagem. O escopo de uma aplicação pode abranger todas as funcionalidades disponíveis, apenas as utilizadas pelo usuário ou algumas específicas, como relatórios ou transações cadastrais.

Uma vez estabelecidos os limites e o escopo da contagem é necessário classificar as funções do tipo dado. As funções do tipo dados representam as funcionalidades proporcionadas ao usuário pelo sistema e são classificadas em:

- **Arquivo Lógico Interno (ALI)** – é um grupo lógico de dados ou informações de controle sob o ponto de vista do usuário, cuja manutenção é feita pela aplicação (ANDRADE, 2004, p. 14). São exemplos de ALI, tabelas de banco de dados e arquivos de configurações mantidos na fronteira da aplicação;
- **Arquivo de Interface Externa (AIE)** – é um grupo lógico de dados referenciado na aplicação cuja responsabilidade pela manutenção é de outra aplicação (ANDRADE, 2004, p. 14). Assim, um AIE não é mantido pela aplicação que está sendo contada, é apenas referenciado. Um AIE é na realidade um ALI de outra aplicação.

A diferença entre Arquivo Lógico Interno e Arquivo de Interface Externa é que o AIE não é mantido pela aplicação que está sendo contada, é apenas referenciado ou lido por ela.

Após a identificação das funções do tipo dado é necessário identificar as funções do tipo transação. As funções do tipo transação representam as funcionalidades proporcionadas ao usuário para o processamento da aplicação e são classificadas em:

- **Entrada Externa (EE)** – é um processo elementar que processa dados ou informações de controle vindas de fora da fronteira da aplicação. São exemplos de EE transações que recebam dados externos utilizados na manutenção de ALI, janelas que permitam adicionar ou alterar registros e o processamento em lote de atualização em bases cadastrais. Em geral, possuem nomes como incluir, excluir, alterar, gravar, exportar e etc;
- **Consulta Externa (CE)** – é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. São exemplos de CE informações que são apresentadas aos usuários na forma de relatórios ou simplesmente consultas. Para ser considerada uma CE, nenhum ALI ou AIE pode ser alterado e não pode haver uma lógica de processamento adicional necessária para apresentar os dados, como por exemplo, cálculos matemáticos;
- **Saída Externa (SE)** – da mesma forma que a consulta externa, é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação. Porém estes dados são apresentados através de uma lógica de processamento adicional, utilizando obrigatoriamente cálculos, fórmulas ou criando dados derivados a partir da recuperação das funções do tipo dados. São exemplos de SE informações em formato gráfico e relatórios com totalização de valores.

A partir da identificação de cada função, seja do tipo dado ou transação, é necessário verificar a complexidade de cada uma delas e o quanto em PF ela contribui para cálculo do tamanho total do produto a ser mensurado.

## 2.2. Complexidade e Contribuição das Funções

Após identificar as funções do tipo dado (ALI e AIE) e as funções do tipo transação (EE, CE e SE), é necessário verificar a complexidade (alta, média e baixa) de cada uma delas com base na quantidade de tipos de dados, tipos de registros ou arquivos referenciados.

Um tipo de dado (TD) é um campo único identificável pelo usuário e não repetível, como por exemplo, um elemento de uma tabela. O tipo de registro (TR) é um conjunto de tipos de dados que são divididos em opcionais e obrigatórios. Um arquivo referenciado (AR) é qualquer ALI ou AIE lido ou mantido por uma função do tipo transação.

Assim, para identificar a complexidade das funções do tipo dados é necessário identificar os tipos de dados de cada campo não repetido e reconhecido pelo usuário e também os grupos de dados opcionais e obrigatórios reconhecidos. A tabela 1 apresenta a complexidade funcional dos ALI e AIE de acordo com quantidade de tipos de dados e tipos de registros.

**Tabela 1** - Complexidade funcional dos ALI e AIE (ANDRADE, 2004, p. 15)

	Tipo de Dados (TD)			
	Quantidade	< 20	20 – 50	> 50
Tipos de Registros (TR)	1	Baixa	Baixa	Média
	2 – 5	Baixa	Média	Alta
	> 5	Média	Alta	Alta

Para identificar a complexidade das funções do tipo transação é necessário identificar os arquivos referenciados e os tipos de dados identificados. As tabelas 2 e 3 apresentam a classificação das complexidades das funções do tipo transação.

**Tabela 2** - Complexidade para entradas externa (EE), baseado em Andrade (2004, p. 16)

	Tipos de Dados (TD)			
	Quantidade	< 5	5 – 15	> 15
Arquivos Referenciados (AR)	< 2	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	> 2	Média	Alta	Alta

**Tabela 3** - Complexidade para saídas externas (SE) e consultas externas (CE), baseado em Andrade (2004, p. 17-18)

	Tipos de Dados (TD)			
	Quantidade	< 6	6 – 19	> 19
Arquivos Referenciados (AR)	< 2	Baixa	Baixa	Média
	2 – 3	Baixa	Média	Alta
	> 3	Média	Alta	Alta

Após a determinar as complexidades das funções é necessário calcular a quantidade de pontos de função para cada complexidade (baixa, média e alta)

identificada. A tabela 4 apresenta a contribuição de cada tipo função de acordo com sua complexidade.

**Tabela 4** - Contribuição dos pontos de função das funções do tipo dados e transação

<b>Tipo de Função</b>	<b>Baixa</b>	<b>Média</b>	<b>Alta</b>
<b>Arquivo Lógico Interno</b>	7 PF	10 PF	15 PF
<b>Arquivo de Interface Externa</b>	5 PF	7 PF	10 PF
<b>Entradas Externas</b>	3 PF	4 PF	6 PF
<b>Saídas Externas</b>	4 PF	5 PF	7 PF
<b>Consultas Externas</b>	3 PF	4 PF	6 PF

Assim, por exemplo, um ALI identificado com quatro tipos de dados (TD) e um tipo de registro (TR) possui uma complexidade baixa e contribui com sete pontos de função. Já um EE com dez tipos de dados (TD) e dois arquivos referenciados (AR) possui uma complexidade média e contribui com quatro pontos de função.

Após a contagem das funções do tipo dados e as funções do tipo transação obtém-se, para cada uma delas, os pontos de função não ajustados. Pontos de função não ajustados refletem a funcionalidade específica, passível de contagem, fornecida ao usuário pela aplicação (FATTOCS, 2006, ONLINE). Após a contagem de pontos de função não ajustados, é necessário determinar o fator de ajuste da aplicação, para então obter o cálculo final de pontos de função.

### 2.3. Fator de Ajuste e Cálculo Final

O valor do fator de ajuste é baseado em 14 características gerais de sistemas, como comunicação de dados, processamento, volume de transações, facilidade de mudanças e outras. Essas características afetam o sistema de uma forma geral, e influenciam a complexidade do software e o seu tamanho. Para isso, é dado um nível de influência (peso) para cada uma delas, que vai de zero (0), em que não possui nenhuma influência, até cinco (5), com grande influência. A tabela 5 apresenta características gerais de sistemas a serem avaliadas.

**Tabela 5** - Características gerais do sistema (MECENAS, 2009, p. 34)

<b>Características Gerais do Sistema - CGS</b>	
01 - Comunicação de Dados	08 - Atualização On-line
02 - Processamento Distribuído	09 - Complexidade do Processamento
03 - Performance	10 - Reutilização
04 - Configuração Altamente Utilizada	11 - Facilidade de Instalação
05 - Taxa de Transações	12 - Facilidade de Operação
06 - Entrada de Dados On-line	13 - Múltiplas Localidades
07 - Eficiência do Usuário Final	14 - Facilidade de Mudanças

Para o cálculo do fator de ajuste, devem ser somados os níveis de influência de cada característica e utilizar a seguinte equação (HAZAN, 2005, p.15): **Fator de Ajuste = (Somatório dos Níveis de Influência das CGS \* 0,01) + 0,65.**

Após a obter o fator de ajuste, já é possível calcular os pontos de função ajustados de um sistema entregue, através da multiplicação dos pontos de função não ajustados com o fator de ajuste encontrado.

Enfim, com a contagem de tamanho em pontos de função é possível fazer estimativas de prazo, esforço e custo. Porém, quando os requisitos ainda não estão bem definidos não é possível identificar todas as funções, e conseqüentemente mensurar o tamanho do software com precisão. Desta forma, é necessário utilizar métodos auxiliares para mensurar o tamanho antes do início do desenvolvimento, para que possam ser estimados os recursos, o tempo e custo.

### **3. Abordagem NESMA**

A NESMA – *Netherlands Software Metrics Association* é a associação holandesa de métricas de software e é o maior grupo de usuário de pontos de função da Europa (NESMA, 2009, ONLINE). Algumas abordagens utilizadas pela NESMA são muito utilizadas pelos usuários do IFPUG, como por exemplo, as técnicas de estimativa de tamanho de software, que possibilitam medir um produto desde o início do processo, mesmo não possuindo as funções do tipo dado, transação e características gerais do sistema definidas. Desta forma, a NESMA propõe três tipos de contagem de pontos de função: a contagem indicativa, contagem estimada e a contagem detalhada.

A contagem indicativa oferece um cálculo estimado da quantidade de pontos de função de um software, sem a necessidade de conhecer em detalhes o modelo de negócios do sistema. É utilizada na fase inicial da proposta de desenvolvimento, quando há apenas um modelo preliminar de dados. Os elementos utilizados para a contagem indicativa são os Arquivos Lógicos Internos – ALI e os Arquivos de Interface Externa – AIE. Assim, cada ALI identificado contribui com 35 pontos de função e cada AIE identificado contribui com 15 pontos de função.

A contagem estimada de pontos de função pode ser utilizada em sistemas quando não há uma precisão do nível de complexidade das funções existentes. Assim, é utilizada nas fases iniciais do ciclo de vida do sistema, quando não existem dados detalhados do negócio e que há apenas informações preliminares do funcionamento das funções de processamento do sistema. Os elementos utilizados para a contagem estimada são os ALI, AIE, EE, CE e SE.

Para o cálculo de pontos de função através da contagem detalhada é necessário obter dados detalhados dos processos e do modelo de dados, como descrição de telas e relatórios ou mesmo um protótipo do sistema (ALBUQUERQUE e LACERDA, 2005, p. 81). A contagem detalhada utiliza as funções do tipo dado (ALI e AIE) e do tipo transação (EE, CE e SE).

### **4. Resultados e Discussões**

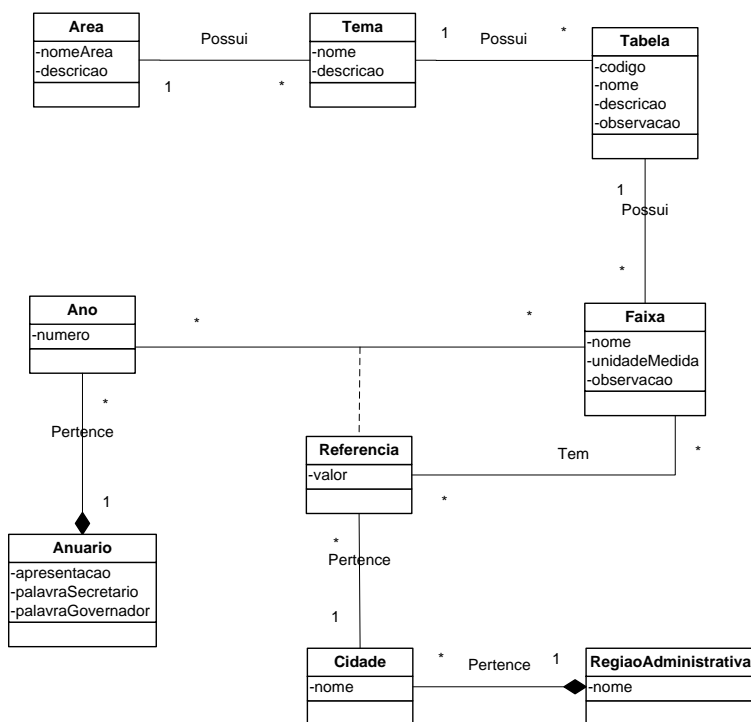
Para o desenvolvimento deste trabalho foram realizadas pesquisas bibliográficas sobre as principais métricas de tamanho de software utilizadas. Houve um destaque em pontos de função devido à crescente utilização desta técnica pela indústria de software,



principalmente pela utilização desta em contratos e licitações públicas (HAZAN, 2009, p. 26). Após isso, foi necessário escolher um projeto de software que possibilitasse mensurar o seu tamanho.

O Sistema de Dados Estatísticos do Estado do Tocantins foi o software escolhido para ser mensurado. O software é uma ferramenta para difusão de conhecimento sobre os aspectos físicos, demográficos e sociais de cada município e região do estado. Como o sistema ainda está na fase de concepção, não é possível realizar a medição utilizando apenas a contagem segundo o IFPUG, pois não existem dados suficientes para identificar as funções do tipo dado, funções do tipo transação e características gerais do sistema desejadas. Desta forma, para estimar o tamanho do software será necessário utilizar a contagem indicativa proposta pela NESMA, pois estima o tamanho do projeto quando não existem outras informações além das funções do tipo dado.

O Sistema de Dados Estatísticos consiste em uma ferramenta web que visa difundir o conhecimento sobre os aspectos físicos, demográficos e sociais dos municípios do estado do Tocantins. Atualmente estes dados são apresentados pela Secretaria do Planejamento - SEPLAN, órgão responsável pelas informações, de duas formas: através de um anuário em forma de um livro e na internet como vários arquivos do tipo PDF. Porém, as duas formas não permitem realizar consultas sobre os dados, dificultando a comparação entre valores e o acesso a determinados dados de um município. A figura 2 apresenta o modelo conceitual do sistema a ser mensurado, contendo todos os conceitos e associações necessárias para calcular o tamanho do sistema.



**Figura 2** - Representação do modelo conceitual do Sistema de Dados Estatísticos do Estado do Tocantins.

Os dados do anuário são divididos em áreas, temas e tabelas. As áreas são representadas pela demografia, agricultura, saúde, seguranças e etc. Para cada área existem vários temas, como por exemplo, na área agricultura, frutas, grãos e características do produtor. Cada tema possui várias tabelas com informações para cada ano, como por exemplo, no tema grãos, a colheita de arroz de sequeiro, o café, o milho e etc. As informações são divididas por regiões administrativas e por municípios. Cada tabela possui várias faixas com seus valores, a tabela grãos, por exemplo, apresenta as seguintes faixas: área colhida, produção total e rendimento médio. Já a tabela turmas no ensino médio apresenta a quantidades de turmas na rede pública, particular, federal e, ainda, o total para cada região administrativa. Algumas tabelas possuem faixas que são resultantes de cálculos através de outras faixas, como por exemplo, a faixa rendimento médio, que é resultado da divisão da produção total em quilogramas (kg) com sua produção em hectare (ha). A figura 3 apresenta o modelo de apresentação dos dados no anuário.

2.4 - ÁREA COLHIDA, PRODUÇÃO E RENDIMENTO MÉDIO DE SOJA DE SEQUEIRO, SEGUNDO AS REGIÕES ADMINISTRATIVAS E MUNICÍPIOS DO TOCANTINS - 2006

Regiões Administrativas	Área colhida (ha)	Produção (t)	Rendimento Médio (kg/ha)
TOTAL DO ESTADO	297.300	678.377	2.282
REGIÃO I - ARAGUATINS	200	400	2.000
São Bento do Tocantins	200	400	2.000
REGIÃO III - TOCANTINÓPOLIS	3.570	8.925	2.500
Darcinópolis	3.570	8.925	2.500
REGIÃO V - ARAGUAÍNA	9.330	23.481	2.517
Nova Olinda	6.000	15.300	2.550
Palmeirante	3.000	7.500	2.500
Santa Fé do Araguaia	200	330	1.650
Wanderlândia	130	351	2.700

**Figura 3** – Representação de informações da tabela área colhida, produção e rendimento médio de soja de sequeiro do ano de 2006 (SEPLAN, 2009, ONLINE)

Para calcular o tamanho utilizando a APF e a abordagem contagem indicativa da NESMA é necessário apurar a quantidade de ALI e AIE. Como o Sistema de Dados Estatísticos não acessa qualquer outro sistema e não faz consulta a dados armazenados fora da fronteira de sua aplicação, o mesmo não possui AIEs. Quanto aos ALIs, devem ser identificáveis pelo usuário e mantidos dentro da fronteira da aplicação. Assim, os ALIs identificados segundo o modelo conceitual (Figura 2) são:

✓ **Área e Tema** – de acordo com o modelo conceitual, uma AREA possui pelos menos um TEMA, e este tema está relacionado com apenas uma AREA. Assim, AREA é uma entidade independente<sup>1</sup> e é contado como um único ALI. Quanto ao TEMA, segundo os requisitos do negócio, não pode ser excluído, apenas remanejado para outra AREA correspondente. Assim, TEMA possui significado para o usuário, sendo considerada uma entidade independente de AREA. Desta forma, TEMA também é um único ALI autônomo;

✓ **Tabela** – um TEMA possui várias TABELAS, e estas estão relacionadas com apenas um TEMA. Segundo a visão do usuário, a entidade TABELA possui todas as informações necessárias de determinado tema e área em um determinado ano. Assim, as entidades FAIXA e REFERENCIA não têm significado para o negócio a menos que seja associada

<sup>1</sup> Uma entidade independente é aquela que tem significado para o negócio mesmo sem a presença de outra entidade (MECENAS, 2009, p. 56).

a ocorrências da entidade TABELA. Portanto, as entidades FAIXA e REFERENCIA formam juntamente com TABELA um único arquivo lógico;

✓ **Anuário** – as dados do anuário são correspondentes a vários anos, em que um ANUARIO possui pelo menos um ANO, e cada ANO está associado a apenas um anuário. A entidade ANO é dependente da entidade ANUARIO e não possui significado para o usuário. Assim, ANUARIO e ANO são tratados como um único arquivo lógico autônomo;

✓ **Cidade e Região Administrativa** – uma REGIÃO ADMINISTRATIVA é formada por várias entidades CIDADE, e cada CIDADE está associada a apenas uma região. Segundos os requisitos de negócio, uma CIDADE não pode ser excluída, apenas redistribuída para outra REGIÃO ADMINISTRATIVA, conforme a necessidade. Desta forma, ambas as entidades são independentes e cada uma é contada como um único arquivo lógico autônomo.

A partir da análise do modelo conceitual e a das regras de identificação de funções do tipo dados da APF, tem-se 6 ALIs (Área, Tema, Tabela, Anuário, Cidade e Região Administrativa) e nenhum AIE. Para o cálculo indicativo de tamanho segundo a NESMA, conta-se 35 pontos de função para cada ALI identificado, desta forma o sistema possui 210 pontos de função estimados.

Enfim, com tamanho em pontos de função é possível medir o que o usuário solicita e recebe, facilitando o processo de comunicação entre os gestores da organização e a empresa de desenvolvimento, devido à fundamentação de tamanho dada pela APF. Com pontos de função, o cliente saberá o porquê da contagem de pontos de cada funcionalidade requisitada, devido às regras serem de simples assimilação e independentes de plataformas de linguagem e banco de dados. Pontos de função podem ser utilizados também para fundamentar uma negociação contratual, em que cada ponto de função equivale a um valor, em que o cliente paga por cada ponto de função desenvolvido e entregue. Assim, há sempre a necessidade de analisar o propósito da contagem de PF, seja em mensurar o tamanho de um módulo entregue, para calcular o valor a ser pago, ou estimar um tamanho, para servir de base para calcular o tempo ou esforço necessário à finalização do projeto.

## 5. Considerações Finais

Após do cálculo do tamanho do sistema, já é possível iniciar as outras fases do processo de estimativas. A partir da dimensão do produto, pode-se verificar o esforço necessário para desenvolvê-lo, de acordo com a base histórica de projetos da empresa, em que são comparados o tamanho de produtos anteriores e os recursos computacionais e humanos utilizados. Com o esforço, tamanho e com base no histórico da empresa, chega-se a estimativa de entrega de módulos do produto. Enfim, com o esforço necessário definido e com o cronograma acertado, realiza-se a estimativa de custo que é entregue ao cliente. A partir do cronograma é realizado o acompanhamento das estimativas realizadas com o intuito de melhorar o processo de desenvolvimento, identificando falhas e corrigindo erros em estimativas, para que as próximas medições obtenham os resultados mais próximos do real.

Assim, utilizando a APF, como técnica de medição de tamanho de software e, a abordagem de estimativa da NESMA, com a contagem indicativa de tamanho, a

empresa de desenvolvimento consegue estimar o custo do software fundamentado por uma métrica internacionalmente aceita, que é o ponto de função. Ainda, consegue nas fases iniciais do projeto, identificar necessidades e problemas que possam surgir, melhorando o processo de desenvolvimento e os custos relativos à sua implementação.

A próxima fase deste trabalho é realizar as estimativas de esforço, prazo e custo do Sistema de Dados Estatísticos do Estado do Tocantins. Para realizar estas estimativas será utilizado o COCOMO II (*Constructive Cost Model II*), um modelo que permite estimativas para custo, esforço, prazo e tamanho de equipe de desenvolvimento a partir da dimensão do produto, que neste caso, é o tamanho em pontos de função estimado do sistema.

## 6. Referências Bibliográficas

- ANDRADE, Edméia Leonor Pereira de. **Pontos de Casos de Uso e pontos de Função na Gestão de Estimativa de Software Orientado a Objetos**. Dissertação (Mestrado em Gestão do Conhecimento e Tecnologia da Informação). 2004. Universidade Católica de Brasília, Brasília, 132 p. Disponível em <<http://www.bfpug.com.br/Artigos/UCP/Tese%20Edmeia.zip>>. Acesso em 10 de maio de 2009.
- ALBUQUERQUE, Jones Oliveira; LACERDA, Isac Mendes. **Uma Experiência na Medição Automática de Pontos de Função**. Disponível em <[http://www.proqualiti.org.br/revista/revista\\_ProQualiti\\_maio2005.pdf](http://www.proqualiti.org.br/revista/revista_ProQualiti_maio2005.pdf)>. Acesso em 08 de agosto de 2009.
- FATTOCS. FATTO Consultoria e Sistemas. Glossário sobre Análise de Pontos de Função. 2006. Disponível em <<http://www.fattocs.com.br/download/glossario.pdf>>. Acesso em 01 de junho de 2009.
- HAZAN, Cláudia. **Análise de Pontos de Função** - Uma aplicação nas estimativas de tamanho de Projetos de Software. Engenharia de Software Magazine. Rio de Janeiro, n. 02, p. 25-31, 2009. Disponível em <[http://kameha.devmedia.com.br/site/engsoft/02/ESM02\\_baixa.zip](http://kameha.devmedia.com.br/site/engsoft/02/ESM02_baixa.zip)>. Acesso em 22 de abril de 2009.
- HAZAN, Cláudia; FUKS, Hugo; LUCENA, Carlos José Pereira. **Avaliação do Tamanho Funcional de Ferramentas de E-learning**. Pontifícia Universidade Católica do Rio de Janeiro - PUC. Maio de 2005. Disponível via FTP. URL: <ftp://ftp.inf.puc-rio.br>. Diretório: pub/docs/techreports/05\_16\_hazan.pdf. Acesso em 25 de junho de 2009.
- MECENAS, Ivan. **Análise de Pontos de Função: Estudo Teórico, Crítico e Prático**. Rio de Janeiro: Alta Books, 2009. 224p.
- NESMA. Netherlands Function Point Users Group. Disponível em <<http://www.nesma.nl/section/nesma/>>. Acesso em 06 de setembro de 2009.
- SEPLAN. Secretaria do Planejamento. Disponível em <[http://www.seplan.to.gov.br/seplan/br/index2.php?area=estatico&id\\_m=123](http://www.seplan.to.gov.br/seplan/br/index2.php?area=estatico&id_m=123)>. Acesso em 22 de setembro de 2009.
- VAZQUEZ, Carlos Eduardo, SIMÕES, Guilherme Siqueira, ALBERT, Renato Machado. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento e Projetos de Software**. São Paulo: Érica, 2003. 230 p.

# **Análise de Gerenciamento de Projeto de Software Utilizando Metodologia Ágil XP e Scrum: Um Estudo de Caso Prático**

**Márcia Savoine, Lucyano Martins, Mayton Rocha, Cirlene dos Santos**

Curso de Sistema de Informação – ITPAC – Inst. Toc. Presidente Antônio Carlos  
Av. Filadélfia, 586 – Setor Oeste – CEP: 77.816-540 – Araguaína – TO

savoine@yahoo.com.br, lucyano@gmail.com, mayton@gmail.com, cirlene\_rs@yahoo.com.br

**Resumo.** *Este artigo relata um estudo de caso sobre projeto de software utilizando as metodologias ágeis eXtreme Programming e Scrum. Aborda-se o uso das melhores práticas de ambas metodologias em uma equipe de desenvolvimento de software; onde esta, evoluiu do paradigma tradicional para o ágil. Isto ocorreu devido às necessidades dos modelos de negócios atuais, tais como: crescimento da equipe, melhoria na qualidade do código implementado e exigências do mercado de software pela excelência do produto. São levantadas as principais características das metodologias tradicionais e ágeis, fazendo um comparativo entre elas.*

**Palavras chave:** Metodologia Ágil, eXtreme Programming, Scrum.

## **1. Introdução**

Os problemas enfrentados pelas empresas de desenvolvimento de software, como: os atrasos na entrega do projeto, produtos de baixa qualidade, aumento significativo dos custos muito além do esperado; são fatores causados pela falta de gerenciamento nos processos que envolvem a produção de software, que muitas vezes, estão ligados diretamente há uma coleta de dados realizada de forma incompleta.

O desenvolvimento de software tem falhas na entrega e nos valores entregues. Essas falhas têm impactos econômicos e humanos enormes. É necessário achar uma maneira de desenvolver software com qualidade e entregas freqüentes. (BECK, 2004)

Com a globalização a concorrência tornou-se muito acirrada; com isso, a tolerância à falhas, atrasos, *softwares* obsoletos e projetos cancelados estão cada vez menores, e as empresas precisam se enquadrar nesse novo contexto para poderem se destacar neste cenário.

Os métodos ágeis têm-se destacado nesse contexto de mudanças, oferecendo respostas rápidas a esse novo ambiente de desenvolvimento, onde os requisitos são mutáveis, não estão totalmente claros ou não são totalmente esclarecidos; mas exige-se a de entrega do produto com valor tangível. Com todos esses problemas envolvendo a produção de software, faz-se necessário a mudança dos métodos utilizados no desenvolvimento, através da adesão a um novo paradigma, eliminando conceitos que muitas vezes atrapalham o andamento do projeto.

Este artigo contempla um estudo das metodologias ágeis, explicando suas principais características, e realiza-se um comparativo entre as mesmas, incluindo um estudo de caso prático. Discutem-se os resultados obtidos deste *case* e são apresentadas às conclusões na última sessão.

## 2. Metodologias Ágeis

Os métodos ágeis de desenvolvimento de *softwares* surgiram com a necessidade de tornar o desenvolvimento de software mais leve, flexível a mudanças, sem o aumento exponencial dos custos em contrapartida dos métodos tradicionais que desperdiçavam muito tempo em análise e planejamento.

A partir de fevereiro de 2001, desenvolvedores de *software* de várias partes do mundo, insatisfeitos com as técnicas e métodos de desenvolvimento de sistemas usados até o momento, resolveram criar uma aliança chamada de *Agile Software Development Alliance*, mais conhecida como *Agile Alliance*.

Estes profissionais das diversas áreas de formação com pontos de vista diferentes sobre os modelos e métodos de desenvolvimento de software em comum, criaram um manifesto para encorajar melhores meios de desenvolvedor software (AMBLER, 2004).

Este documento, definido como o **Manifesto Ágil ou Agile Manifesto** para as novas metodologias de desenvolvimento de software; foi criado por vários desenvolvedores, mas teve como idealizadores Kent Beck, Ken Schwaber, Martin Fowler e Jim Highsmith. E, abrange um conjunto de princípios, que definem critérios para o processo de desenvolvimento de software ágil. Tais princípios ágeis são:

- ◆ Indivíduos e iterações são mais importantes que processos e ferramentas;
- ◆ Software em funcionamento mais que documentação abrangente;
- ◆ Colaboração com o cliente mais que negociação de contratos;
- ◆ Responder a mudanças mais que seguir um plano; ou seja, mesmo havendo valor nos itens à direita, valoriza-se mais os itens à esquerda.

As metodologias ágeis têm conseguido maior espaço devido principalmente a mudanças no perfil dos projetos de *software*. As abordagens tradicionais são pouco propícias às mudanças para atender as necessidades do projeto. No entanto, a realidade atual é composta por alterações a todo o momento e requisitos em constante evolução; ou seja, projetos mutantes, por isso surge à necessidade de metodologias que se adaptem a esse paradigma.

### 2.1 Metodologia Ágil eXtreme Programming

Os fundamentos da metodologia *eXtreme Programming - XP*; ou ainda Programação eXtrema, teve início na década de 80, tomando como base o desenvolvimento em *SmallTalk*, que já naquela época praticava o desenvolvimento com programação pareada, *refactoring*, testes constantes, desenvolvimento iterativo, que hoje são utilizados nas práticas da XP.

Segundo Beck (2004), XP é uma metodologia leve para equipes de tamanho pequeno a médio, que desenvolve software em face a requisitos vagos que modificam rapidamente. É uma maneira leve, eficiente, de baixo risco, flexível, previsível, científica e divertida de desenvolver software.

A Programação eXtrema possui como característica importante ser adaptativa, usada principalmente quando se tem um ambiente no qual os requisitos funcionais do

sistema não estão totalmente claros, ou não são conhecidos pela equipe de desenvolvimento; e quando os mesmos são mutáveis ao longo do ciclo de vida do projeto.

É uma metodologia que encoraja a equipe a enfrentar as mudanças, como algo natural, sendo importante se adaptar aos contratemplos que acontecem no projeto, contornando assim as dificuldades do modo mais simples possível sem perder o foco básico: produzir um produto de qualidade, almejando excelência.

A satisfação do cliente é uma preocupação constante na XP, pois entregar um produto de qualidade em tempo hábil que satisfaça as exigências e expectativas tanto do cliente como da equipe de desenvolvimento é o objetivo principal. Esta metodologia segue rigorosamente quatro princípios de fundamental importância para seu embasamento: a *comunicação*, o *feedback*, a *simplicidade* e a *coragem*.

A Programação eXtrema enfatiza o uso de 12 práticas, utilizadas de forma harmônica e complementar para a entrega de um software de alta qualidade, que são mostradas na tabela 1, apontados seus pontos fortes e fracos.

**Tabela 1. Características da metodologia ágil XP.**

Referência	Pontos Fortes	Pontos Fracos
1	Reuniões diárias: tiram-se dúvidas de estórias complexas, planejam-se as iterações do dia, discutem-se as dificuldades encontradas.	Não indicada para sistemas complexos, pois a análise não é detalhada.
2	Escopo do projeto não bem definido: os requisitos não são totalmente conhecidos ou são muito vagos.	Não indicada para equipes muito grandes, pois o escopo da equipe tem que ser bem definido, e as pessoas, geralmente, estão separadas geograficamente.
3	Foco na codificação, sendo que a documentação fica para segundo plano.	Não foca muito a gerência do projeto, o que interessa a equipe são os resultados, o código rodando.
4	Cliente presente é a chave para o sucesso.	Não é recomendada em ambientes onde os clientes não pode estar presente.
5	Flexível a mudanças encoraja as pessoas a enfrentarem com naturalidade.	Não é recomendado em projetos que o cliente exige uma extensa documentação do software.
6	Simplicidade: executar tarefas com simplicidade e qualidade, desenvolver funcionalidades realmente útil ao usuário.	Não recomendada em equipes que possuem resistências às práticas como <i>refactoring</i> , programação pareada, etc.
7	Iterações curtas, com isso o <i>feedback</i> do usuário é rápido e constante, realimentando o desenvolvimento das funcionalidades.	Não indicada em projetos em que o cliente não possa participar integralmente do desenvolvimento.
8	Os testes fazem parte do ciclo de desenvolvimento, aumentando assim a qualidade do software.	Se o <i>feedback</i> entre o cliente e a equipe não for constante, ou houver uma demora de respostas entre às partes, a XP não será produtiva.
9	Integração diária do código produzido.	Quando houver resistência em modificar o código já concluído, com o objetivo de torná-lo simples, estruturado.
10	Código coletivo: todos são responsáveis, e aptos a realizares melhorias, caso necessário.	Não é recomendada em ambientes de trabalhos exaustivos, a XP enfatiza o trabalho de 8 horas diárias.
11	Padronização do código: evitar problemas na hora da refatoração e da manutenção;	Não indicada para projetos que necessitem de ciclos de iterações longos;
12	Refatorar: simplificar o código, sem alterar sua funcionalidade.	A XP não é uma metodologia flexível em relação as suas práticas, que devem ser seguidas rigorosamente.

Segundo Beck (2004), as práticas apóiam umas às outras. O ponto fraco de uma é compensado pelos pontos fortes das outras. As peças individuais são simples. A riqueza vem da interação entre as partes.

## 2.1 Metodologia Ágil Scrum

A utilização da metodologia *Scrum* está crescendo muito ultimamente no mundo de desenvolvimento de *software*. Isto, por ser uma metodologia empírica, adaptativa, inovadora, usada no desenvolvimento de sistemas de modo incremental, onde os requisitos podem sofrer mudanças durante o processo de fabricação do produto.

É uma metodologia com o foco no gerenciamento da equipe, preocupada na organização dos processos, no modo como as atividades devem ser executadas, deixando a cargo dos participantes do projeto escolher a melhor maneira de concluir com sucesso essas etapas.

Os requisitos do software são levantados e organizados em uma lista de funcionalidades, chamada *product backlog*, que contém todo o escopo do projeto definido juntamente com o cliente, e suas respectivas prioridades de desenvolvimento, sendo que o *product backlog* deve ser constantemente revisado, validado e atualizado.

O desenvolvimento é dividido em iterações, chamado de *sprints*, de duração entre duas a quatro semanas. Cada *sprint* é composta de uma lista de tarefas, funcionalidades com prioridades retiradas do *product backlog*, chamada de *sprint backlog*. Definido as atividades a serem realizadas em cada *sprint*, a equipe foca o desenvolvimento de mais um ciclo que se repete ao longo do projeto. Após o encerramento das *sprints* é realizada a *sprint retrospective*, uma das mais importantes práticas dentro da *Scrum*, que são discutidos os pontos positivos e os negativos. A figura 1 ilustra o ciclo de vida da *Scrum*.



Figura 1. Ciclo de vida da *Scrum* - Fonte: PEREIRA, 2007.

Durante a fase das *sprints*, acontecem reuniões diárias, de no máximo 15 minutos chamadas *daily scrum meetings*, com todos os integrantes da equipe. Onde se coloca em pauta perguntas sobre as atividades realizadas no dia anterior. No encerramento de cada ciclo da *sprint*, ocorre a apresentação das tarefas concluídas, também conhecidas como *sprint review*, onde o grupo apresenta os resultados atingidos e as funcionalidades 100% prontas e testadas a toda a equipe.

Os papéis da *Scrum* são: *Product Owner*, *ScrumMaster* e o *TeamMember*. Esta metodologia pode ser usada em equipes pequenas e grandes, mas recomenda-se quando for um grupo com mais de dez pessoas, que faça à subdivisão do mesmo. Então, com o



projeto dividido em partes, que todos trabalhem paralelamente e de forma sincrônica. Segue a tabela 2 indicando os pontos fortes e fracos da metodologia.

**Tabela 2. Características da metodologia ágil Scrum.**

Referência	Pontos Fortes	Pontos Fracos
1	Metodologia destinada ao caos, onde não se tem todos os requisitos do sistema; ou eles, não estão totalmente definidos.	Não indicada para sistemas complexos, pois a análise não é detalhada.
2	Flexibilidade para mudanças rotineiras que acontecem freqüentemente durante as fases do projeto.	Não indicada para projetos que exigem uma vasta documentação.
3	O cliente faz parte da equipe em tempo integral.	Não indicada onde os clientes não tenham disponibilidade de tempo e onde são resistentes a troca de informações.
4	Reuniões diárias que guiam o andamento do projeto.	Não indicada para equipes que não fazem uso das reuniões diárias, sendo de extrema importância para o sucesso de um projeto.
5	Ciclos de desenvolvimentos curtos e constante, conhecidas como <i>sprints</i> .	Não indicada para projetos que necessitem de ciclos de iterações longos.
6	Comunicação entre os membros da equipe é freqüente.	Não recomendada em ambientes onde as tarefas são impostas as pessoas, pois as mesmas têm liberdade em escolher tarefas que tenham afinidades de desenvolvimento.
7	Revisões das funcionalidades realizadas acontecem no final de cada ciclo.	Projetos que o líder não divide responsabilidades com a equipe, não tendo efetiva troca de idéias com o restante do grupo, a Scrum não será produtiva; já que, defende trabalho colaborativo, não apenas feito pelo líder.
8	Pode ser aplicada tanto para equipes pequenas e grandes, e também para projetos de pequeno e de grande porte.	Definindo-se a <i>sprint</i> , nenhuma mudança poderá ser acrescentada na mesma; adições de novos requisitos são feitos na próxima <i>sprint</i> .
9	Elaboração de cronogramas para a fixação de datas e horas trabalhadas.	Não indicada em projetos que não há a presença do cliente, nesta metodologia se tem o papel do <i>product owner</i> que representa a figura do cliente.
10	Testes acontecem em cada fase das <i>sprints</i> .	Equipe tem resistência em realizar testes na fase de desenvolvimento, a Scrum não é recomendada.

### 3. Estudo de Caso Utilizando as Metodologias XP e Scrum

O estudo de caso foi realizado em uma equipe de desenvolvimento com 15 colaboradores, que se constituiu para trabalhar em um projeto de *software* de gestão de ensino, em uma fábrica de *software* que já desenvolvia outros produtos.

Após diversas reuniões e estudos de mercado, a equipe chegou ao consenso de que seria necessário mudar a atitude em relação ao desenvolvimento de *software*, pois hoje o mercado traz novos conceitos e novas necessidades. Fez-se a análise do paradigma atual e propôs-se a adoção de práticas ágeis, pois este seria o primeiro projeto de desenvolvimento ágil e os requisitos não se apresentavam com exatidão.

#### 3.1 Práticas Ágeis Utilizadas no Projeto

A tabela 3 mostra a compilação das melhores práticas selecionadas entre as metodologias XP e Scrum para serem utilizadas no projeto de gestão de ensino. Dividiram-se as práticas por metodologia e foco, abordando o nome pela qual a prática é

conhecida, a metodologia que utiliza a mesma, uma descrição de sua utilização dentro do estudo de caso e a justificativa da escolha das práticas para o projeto.

**Tabela 3. Práticas ágeis das metodologias XP e Scrum Aplicadas no estudo de caso.**

PRÁTICA	ORIGEM	DESCRIÇÃO	JUSTIFICATIVA
<b>Práticas Focadas no Desenvolvimento</b>			
<b>Código Coletivo</b>	<b>XP</b>	Código fonte disponível para qualquer desenvolvedor a qualquer momento.	Tornou-se necessário adotar essa prática devido à necessidade da implementação ser feita em colaboração, ou seja, vários desenvolvedores no mesmo projeto.
<b>Refatoração</b>	<b>XP</b>	A refatoração está presente no projeto devido ao código coletivo, quando um desenvolvedor visualiza uma possibilidade de melhoria, isto é feito e documentado no próprio código, criando o histórico dessas alterações.	A coletividade do código também gera responsabilidades a todos os desenvolvedores de perceberem possibilidades de melhoria.
<b>Simplicidade</b>	<b>XP</b>	Todo o código produzido é elaborado com o máximo de simplicidade para que possa ser economizado tempo de produção e entendimento da estrutura do código.	A utilização da simplicidade justifica-se pelo fato do código ser coletivo; ou seja, todos precisam entender o que foi implementado.
<b>Foco na Codificação</b>	<b>XP</b>	A documentação tem a função de orientar o desenvolvedor as necessidades do cliente, porém não é o foco da equipe de analistas desenvolver documentação.	A prioridade da equipe é software rodando com base nos princípios ágeis; porém, a documentação não é dispensada.
<b>Integração Contínua</b>	<b>XP</b>	Toda a produção do projeto é organizada no servidor automaticamente pela ferramentas utilizadas, o código produzido pelos desenvolvedores, documentação pelos analistas, e requisitos pelos clientes e gerentes.	O desenvolvimento em colaboração exige que tudo esteja centrado em um único lugar permitindo assim que essas informações sejam repassadas a todos da equipe.
<b>Testes</b>	<b>XP</b>	Após o término de cada <i>sprint</i> semanal ou quinzenal é repassado o código compilado em executáveis para o departamento de testes, após a conclusão dos testes os desenvolvedores trabalham em rodízio para correção de <i>bugs</i> .	Para evitar a entrega de produtos com um número elevado de erros para o cliente faz-se necessário estabelecer uma rotina de testes.
<b>Programação Pareada</b>	<b>XP</b>	O desenvolvimento em pares ocorre quando uma funcionalidade requer conhecimento de vários desenvolvedores; sendo assim, estabelece um par de desenvolvedores responsável para implementar essa funcionalidade.	A necessidade de desenvolver em pares ocorre quando o domínio sobre o problema não é total por somente uma pessoa, faz-se assim a junção de pares para a proposta do desenvolvimento em duplas.
<b>Práticas Focadas no Gerenciamento</b>			
<b>Product Backlog</b>	<b>SCRUM</b>	Volume de documentação inicial do projeto que foi extraída na fase de análise de requisitos pelos analistas contém as informações sobre o escopo do projeto, ficando no portal do projeto e, é acessível por todos os membros da equipe.	Essa documentação engloba a estrutura fundamental do projeto é composta pela lista de todos os módulos do sistema com a descrição de suas funcionalidades que foram estabelecidas juntamente com o cliente.

<i>Sprints</i>	SCRUM	As iterações têm duração de duas semanas, que é a divisão do desenvolvimento, que por sua vez, compõem a lista de funcionalidades a serem desenvolvidas.	Os <i>sprints</i> são como o plano a ser seguido o mapa para a equipe possa se orientar no que desenvolver durante o prazo do ciclo.
<i>Sprints Backlog</i>	SCRUM	No planejamento de cada <i>sprint</i> retira-se as funcionalidades a serem desenvolvidas durante aquela fase; é repassado para o papel e exposta no quadro atual da equipe para que seja fácil o acesso e visível a todos.	Durante cada <i>sprint</i> é feita à lista dessas funcionalidades a serem desenvolvidas para que seja possível acompanhar as tarefas da equipe.
<i>Sprint Retrospective</i>	SCRUM	Quando fecha-se uma <i>sprint</i> é feita uma retrospectiva para que os problemas encontrados pela equipe sejam discutidos, isto para não ocorrer no futuro, e também os acertos para servirem como base a todos.	Essa prática torna-se fundamental à medida que o projeto é desenvolvido, para que a equipe possa crescer com os erros e acertos.
<i>ScrumMaster</i>	SCRUM	O gerente do projeto, líder da equipe, tem papel fundamental na parte gerencial do projeto, auxilia toda a equipe, resolve problemas que possa estar prejudicando o andamento das tarefas, exige dos colaboradores, delega responsabilidades, marca reuniões e em contato direto com os patrocinadores.	A liderança máxima na equipe em relação aos participantes da linha de produção é de fundamental importância, pois o seu papel principal é facilitar o trabalho dos membros da equipe.
<i>Daily Scrum Meeting</i>	SCRUM	Pequenas reuniões diárias com a participação de todos os integrantes da equipe, para que possa ser dado sempre o status do projeto, e também para que o <i>ScrumMaster</i> possa resolver qualquer problema que impeça a equipe de desenvolver o <i>sprint</i> dentro do tempo hábil.	As reuniões são de grande importância para o projeto, pois é onde as informações sobre o status atual do projeto é dada para todos, também o líder <i>ScrumMaster</i> exerce o seu papel de facilitador das tarefas da equipe.

Pode-se observar a diferença de foco entre as metodologias; enquanto a XP é voltada para as práticas de implementação, a Scrum foca no gerenciamento e planejamento e; também, como tratar com a parte estratégica do projeto.

Apresenta-se na Tabela 4 as melhores práticas aplicadas neste estudo de caso; porém, algumas práticas previstas nas metodologias ficaram fora do projeto devido a empecilhos impostos pela empresa que subsidia a equipe de desenvolvimento.

**Tabela 4. Práticas ágeis não utilizadas no projeto**

PRÁTICA	ORIGEM	JUSTIFICATIVA	IMPORTÂNCIA
Cliente sempre presente	XP	Apesar de existir consultores o projeto foi iniciado sem patrocinador específico o que impediu a presença constante desse ator de papel fundamental no início do projeto.	Com o cliente sempre presente é possível ter sempre em mãos os requisitos diretos da fonte; ficando mais simples para a equipe entender os requisitos propostas.
Semana de 40 horas	XP	Os membros da equipe do projeto também atuam em outros projetos desenvolvidos pela empresa que acolhe a equipe; portanto, os desenvolvedores têm semanas de 44 horas trabalhando em expedientes de segunda a sexta-feira de 8 horas e aos sábados de 4 horas.	Trabalhar com colaboradores sempre motivados e descansados é um dos princípios do manifesto ágil, permite extrair do colaborador o máximo do seu poder criativo, fundamental para desenvolver-se software.

### 3.2. Ferramenta Ágil para Gerenciamento das Práticas Ágeis

Para tornar possível a utilização das práticas ágeis, utilizou-se uma ferramenta que promove a colaboração, a qualidade, a previsibilidade e o controle, permitindo também a integração de todas as equipes envolvidas no projeto, trazendo recursos que oferecem ao gerente do projeto a possibilidade de entender a aplicação dos pilares do desenvolvimento ágil.

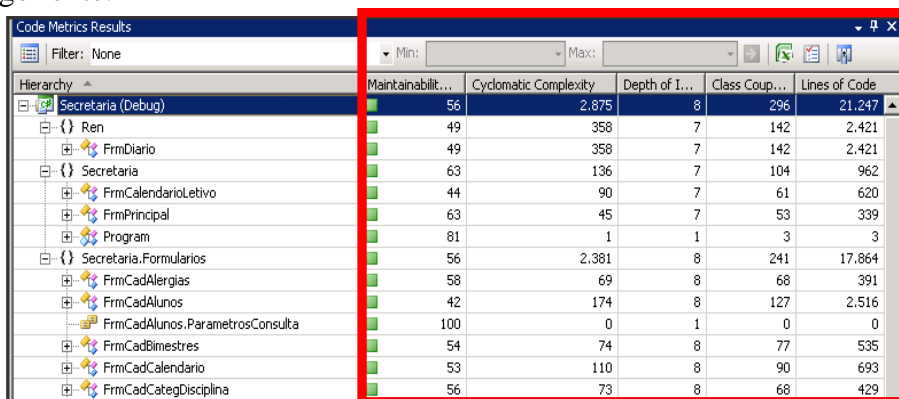
## 4. Resultados Obtidos

Os resultados obtidos foram classificados com foco nos resultados de desenvolvimento e nos resultados gerenciais. São descritos nos próximos itens.

### 4.1 Resultados de Desenvolvimento:

#### a) Qualidade do Código Fonte

Na qualidade do Código Fonte: esses resultados partiram das práticas de implementação da metodologia XP. Observou-se que o código fonte produzido no projeto atende critérios de qualidade estabelecidos em métricas padrões da ferramenta; essa qualidade pode ser comprovada com a opção *Code Analysis*. Na figura 2 mostra uma avaliação de parte do código fonte.



Hierarchy	Maintainability Index	Cyclomatic Complexity	Depth of Inheritance	Class Coupling	Lines of Code
Secretaria (Debug)	56	2.875	8	296	21.247
{ } Ren	49	358	7	142	2.421
{ } FrmDiario	49	358	7	142	2.421
{ } Secretaria	63	136	7	104	962
{ } FrmCalendarioLetivo	44	90	7	61	620
{ } FrmPrincipal	63	45	7	53	339
{ } Program	81	1	1	3	3
{ } Secretaria,Formularios	56	2.381	8	241	17.864
{ } FrmCadAlergias	58	69	8	68	391
{ } FrmCadAlunos	42	174	8	127	2.516
{ } FrmCadAlunos.ParametrosConsulta	100	0	1	0	0
{ } FrmCadBimestres	54	74	8	77	535
{ } FrmCadCalendario	53	110	8	90	693
{ } FrmCadCategDisciplina	56	73	8	68	429

Figura 2. Análise do código fonte.

Os valores das colunas da figura 2 mostram os itens:

- **Maintainability Index** - mostra a facilidade de manutenção do código, quanto maior o valor indicado mais fácil à manutenção.
- **Cyclomatic Complexity** – demonstra o quanto o código apresenta complexidade, quanto menor o valor menos complexo é a codificação.
- **Depth of Inheritance** – mostra a quantidade de herança da classe, quanto menor o valor, melhor será para esse campo.
- **Class Coupling** – avalia a quantidade de referências de outras classes, o valor menor é o melhor neste item.
- **Lines of Code** – aproximadamente o número de linhas de código do executável.

Justifica-se o alcance dessa qualidade devido às práticas ágeis voltadas para codificação como: simplicidade, refatoração e coletividade do código. Porém, o gerenciamento do projeto proporcionado pela ferramenta é foco da metodologia Scrum; e este acompanhamento das etapas desenvolvidas, é função do *ScrumMaster*. A figura 3 mostra o acompanhamento realizado pelo *ScrumMaster* na ferramenta.

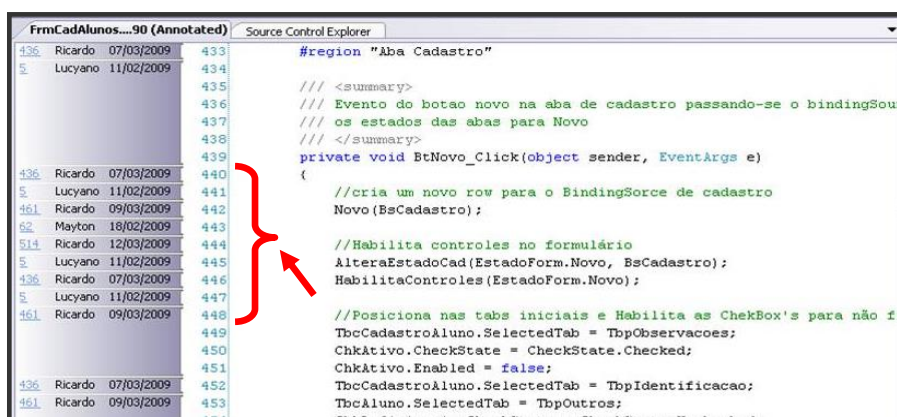


Figura 3. Annotated mostra a refatoração e a coletividade do código-fonte, indicando qual usuário e em qual data fez a alteração.

### b) Integração Contínua

A integração contínua ocorre de forma automática através da opção *Source Control* da ferramenta. Com o uso dessa prática é possível unir o trabalho da equipe de forma coesa, permitindo assim, além de centralização a armazenagem segura.

O ganho de tempo com o uso da prática de integração contínua não pode ser mensurado de forma precisa; pois anteriormente, esse processo era totalmente manual e gastavam-se várias horas para que o gerente do projeto reunisse todo o trabalho produzido. Isto, atualmente é feito de forma automática e imperceptível pelos membros da equipe. Porém; pode-se fazer uma estimativa de que o ganho de tempo na equipe é de aproximadamente 48 horas por mês (sendo 2 horas diárias de perda), totalizando 1 mês de tempo para a equipe em produtividade.

## 4.2 Resultados Gerenciais:

### a) Cumprimentos de Prazos

Foi possível medir a eficiência no cumprimento das datas pré-estabelecidas, após a utilização das práticas de planejamento e delegação de tarefas conforme as iterações e com estabelecimento de prazos a serem cumpridos pela ferramenta. Na Tabela 5 são mostrados os dados obtidos a partir do primeiro *sprint* de funcionalidades desenvolvidas dentro e fora do prazo.

Tabela 5 – Dados de itens desenvolvidos dentro e fora do prazo.

Descrição	Quantidade	Média da Quantidade
Dentro do prazo previsto	38	58%
Fora do prazo previsto	28	42%
<b>TOTAL</b>	<b>66</b>	<b>100%</b>

### b) Resultados dos Testes

Analisaram-se os resultados dos testes com base nos *Work Items* (itens de trabalho) desenvolvidos durante a iteração, ao final foram realizados os testes considerando cada funcionalidade para verificar quantos e quais itens seriam necessários serem revistos pelos desenvolvedores. Esses resultados são apresentados na Tabela 6 abaixo.

Tabela 6. Resultados dos testes aprovados e a serem revistos pelos desenvolvedores.

Descrição	Quantidade	Média da Quantidade
Itens Aprovados	45	68%
Itens Revistos	21	32%
<b>TOTAL</b>	<b>66</b>	<b>100%</b>

Observou-se que o índice de revisão foi baixo, devido a gerência anteriormente aplicada aos itens e; também as técnicas de desenvolvimento proporcionarem um código com maior qualidade e, então consequentemente as funcionalidades atenderam as necessidades propostas.

## 5. Conclusão

A padronização dos processos de desenvolvimento de software é essencial para o alcance do sucesso; sem o mesmo, o desenvolvimento comercial fica inviável, ou tende ao fracasso. O mercado de *software* está cada vez mais competitivo, exigindo produtos de qualidade, equipes dinâmicas, cliente sempre presente; sendo de fundamental importância uma metodologia que possa proporcionar a construção de artefatos aplicáveis a soluções de problemas corriqueiros.

Os resultados obtidos com a utilização das metodologias ágeis foram fundamentais para o sucesso do projeto e da equipe de desenvolvimento, causando impactos determinantes para o alcance dos objetivos propostos; tais como: a mudança de conceito de desenvolvimento tradicional para o ágil, alcançando as exigências de mercado e excelência dos produtos desenvolvidos.

Como mostrado nesse trabalho, o uso das metodologias ágeis *XP* e *Scrum* não é aplicável em todas as situações, e frequentemente requer uma adaptação específica. Estas adaptações mostram os aspectos positivos em relação a flexibilidades desse tipo de metodologia, apresentando um leque de soluções sugeridas a serem usadas em cada problema encontrado.

A equipe absorveu com naturalidade o uso das práticas da metodologia *XP*, não seguindo a risca todas elas; mas acolhendo as que se adequavam melhor na situação atual da empresa. O uso da ferramenta de gerenciamento foi imprescindível para o alcance dos objetivos iniciais, auxiliando o *ScrumMaster* a mensurar todo o trabalho realizado pela equipe e a demonstrar os resultados a todos os participantes do projeto, onde sem esta mudança de paradigma seria um pouco mais árdua e complicada de se conseguir.

Contudo, pode-se notar que o foco desse tipo de metodologia não está na análise de riscos, nem tão pouco em uma documentação detalhada, deixando a desejar em projetos grandes, e com complexidade um pouco mais elevada, que necessitam de um planejamento mais aprofundado. Assim, o grande desafio das metodologias ágeis é encontrar formas de se detalhar a análise de riscos sem torná-las pesadas; mantendo acima de tudo a flexibilidade, que é uma das grandes características que as diferenciam das demais.

## 6. Referências Bibliográficas

- AMBLER, Scott W. Modelagem Ágil: Práticas eficazes para a Programação eXtrema e o Processo Unificado. In: Porto Alegre, Bookman, 2004.
- BECK, Kent. Programação Extrema (XP) explicada: acolha as mudanças. In: Porto Alegre: Bookman, 2004.
- PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. Entendendo Scrum para Gerenciar Projetos de Forma Ágil. Mundo PM. v. 1.8, p3-11, 2007.

# Implementando maturidade e agilidade em uma fábrica de software através de Scrum e MPS.BR nível G

Fernando Szimanski<sup>1</sup>, Jones Albuquerque<sup>2</sup>, Felipe Furtado<sup>2</sup>

<sup>1</sup>CRIATIVA tecnologia - 77.410-020 - Gurupi - TO - Brasil

<sup>2</sup>C.E.S.A.R - Centro de Estudos e Sistemas Avançados do Recife - 50.030-390 - Recife - PE - Brasil

{fszimanski, jones.albuquerque}@gmail.com, felipe.furtado@cesar.org.br

**Abstract.** *Introduce agile concepts into software processes looking a balance between flexibility and maturity is an alternative capable of improving the quality of products and, therefore, increased competitiveness in the market. This paper presents the extension of the agile method Scrum in the process areas of MPS.BR level G applied in a small software factory. This is useful for organizations that have the intention to adopt agile practices in the software process maintaining compatibility with the MPS.BR.*

**Key-words:** Agile methodologies, Scrum, MPS.BR, Project Management, Requirements Management.

**Resumo.** *Introduzir conceitos ágeis em processos de software buscando um equilíbrio entre agilidade e maturidade é uma alternativa capaz de promover a melhoria da qualidade dos produtos e, conseqüentemente, aumento da competitividade no mercado. Este trabalho apresenta a extensão da metodologia ágil Scrum para as áreas de processo do MPS.BR nível G aplicada em uma pequena fábrica de software. A extensão realizada neste trabalho pode contribuir de forma relevante para organizações que têm o propósito de adotar práticas ágeis no seu processo de software mantendo a compatibilidade com o MPS.BR.*

**Palavras-chave:** Metodologias Ágeis, Scrum, MPS.BR, Gerenciamento de projetos, Gerenciamento de requisitos.

## 1. Introdução

Ao final dos anos 90, como reação às formas tradicionais de desenvolvimento de software, que baseado em estudos realizados pela indústria e academia REF8 (2004) REF11 (2006) REF12 (2008), foi apontada como a principal responsável por falhas encontradas em projetos de software, acompanhamos o surgimento de várias metodologias ágeis, como: *Adaptive Software Development, Crystal, Dynamic Systems Development, eXtreme Programming (XP), Feature Driven Development (FDD) e Scrum* REF3 (2006).

Neste sentido, organizações que procuram melhoria em seus processos de produção através de modelos e *frameworks* como *Capability Model Integration (CMMI)*, *Control Objectives for Information and related Technology (COBIT)*, *Information Technology Infrastructure Library (ITIL)*, entre outros, agora também acreditam que introduzir conceitos ágeis em seus processos de desenvolvimento, buscando um equilíbrio entre agilidade e maturidade, é uma alternativa capaz de

promover a melhoria da qualidade dos seus produtos e, conseqüentemente, aumento da competitividade no mercado REF5 (2008).

O desenvolvimento ágil e os modelos e padrões de qualidade de software tradicionais são vistos freqüentemente como contraditórios, pois se tem o raciocínio que modelos e padrões são muito burocráticos e o desenvolvimento ágil é *ad-hoc* REF3 (2004). Na verdade existem desafios na integração entre os dois, embora o esforço possa valer à pena, pois ao final pode-se obter qualidade no produto através da união de maturidade e agilidade.

Inserido neste contexto, este trabalho tem o objetivo de propor uma estratégia para extensão do *Scrum* segundo as áreas de processo do guia MPS.BR nível G. Este estudo se inicia com o mapeamento entre o *Scrum* e o MPS.BR avaliando os resultados esperados do guia segundo as práticas do *Scrum* e partir do mesmo a extensão é realizada através da adição de práticas complementares para satisfazer o guia. Ao final é gerado um novo processo de desenvolvimento para uma fábrica de software buscando avaliar os primeiros resultados obtidos com o uso desta abordagem.

Além do capítulo introdutório, este trabalho está organizado da seguinte forma: na Seção 2 é apresentada uma visão geral do MPS.BR; na Seção 3 apresenta-se o *Scrum*; na Seção 4 é descrito o mapeamento entre o *Scrum* e o MPS.BR; na Seção 5 é proposta a extensão de práticas complementares para o atendimento do guia MPS.BR; na Seção 6 é realizada a adaptação da extensão proposta em uma fábrica de software; a Seção 7 apresenta-se o estudo de caso com os primeiros resultados obtidos com o uso desta nova abordagem. Finalmente, através da Seção 8, apresenta-se conclusão deste trabalho.

## 2. O MPS.BR

O MPS.BR tem como objetivo definir um modelo de melhoria e avaliação de processo de software, preferencialmente para as micro, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e também, ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software REF7 (2007).

O Modelo de Referência MR-MPS define sete níveis de maturidade, que são uma combinação entre processos e capacidade de processos, tais como: A (Em Otimização); B (Gerenciado Quantitativamente); C (Definido); D (Largamente Definido); E (Parcialmente Definido); F (Gerenciado); G (Parcialmente Gerenciado). Para cada um destes sete níveis de maturidade foi atribuído um perfil de processos e de capacidade de processos que indicam onde a organização tem que concentrar o esforço para melhoria de forma a atender os objetivos de negócio REF7 (2007)

O nível G de maturidade do MPS.BR (Parcialmente Gerenciado) é composto pelos processos de Gerência de Projetos e Gerência de Requisitos satisfazendo os atributos de processo AP 1.1 e AP 2.1.

## 3. O Scrum

A metodologia ágil *Scrum* foi criada em 1996 por Ken Schwaber e Jeff Sutherland e destaca-se das demais metodologias ágeis pela maior ênfase dada ao gerenciamento do projeto REF6 (2008).

Trata-se de uma abordagem empírica focada nas pessoas para ambientes em que os requisitos surgem e mudam rapidamente, resultando em uma abordagem que



reintroduz as idéias de flexibilidade, adaptabilidade e produtividade REF2 (2002). Ela se baseia em princípios como: equipes pequenas, requisitos pouco estáveis ou desconhecidos e iterações curtas. Divide as iterações em intervalos de tempos de, no máximo 30 dias, também chamadas de *Sprints*.

Segundo REF2 (2009) o Scrum define para sua estrutura iterativa incremental três papéis principais para diferentes responsabilidades: *Scrum Master*, *Product Owner* e o *Team* e se o seu ciclo de desenvolvimento é baseado em três fases principais: *Pregame*, *Game* e *Postgame* REF1 (2009).

#### 4. Mapeando o Scrum para as áreas de processo do MPS.BR nível G

Para cada área de processo do MPS.BR nível G foi realizada uma análise detalhada entre os resultados esperados do MPS.BR e as práticas encontradas no Scrum, classificando cada resultado esperado de acordo com os critérios estabelecidos na Tabela 1.

Tabela 3. Critérios para classificação das áreas de processo.

CLASSIFICAÇÃO		CRITÉRIO
NS	Não Satisfeito	Não há evidências da prática na metodologia
PS	Parcialmente Satisfeito	Há evidências da prática na metodologia, embora a prática não esteja plenamente atendida.
S	Satisfeito	A prática está totalmente atendida na Metodologia.

O mapeamento detalhado apresentando e pontos que o Scrum satisfaz ou não os resultados esperados do MPS.BR nível G (Gestão de projetos e Gerenciamento de requisitos) por ser encontrado em REF9 (2006).

O processo de gestão de projetos (GPR) é composto por dezessete resultados esperados e o resumo da análise para este processo é apresentado na Tabela 2.

Tabela 4. Resumo do mapeamento entre Scrum e área de GPR.

ÁREA DE PROCESSO	MPS.BR - Nível G		SCRUM	
	ABREV.	OBJETIVOS	CLASSIFICAÇÃO	RESUMO DAS PRÁTICAS
GESTÃO DE PROJETOS (GPR)	GPR1	Escopo do trabalho	SATISFEITO	Elaboração da Visão do Produto e Definição dos Itens de <i>Backlog</i> (IBL)
	GPR2	Dimensionamento de tarefas e produtos de trabalho	PARCIALMENTE SATISFEITO	Estimativas através de <i>Story Points</i>
	GPR3	Modelo e as fases do ciclo de vida do projeto	SATISFEITO	Iterativo incremental. Fases: <i>Pregame</i> , <i>Game</i> e <i>Postgame</i>
	GPR4	Estimativa de esforço e o custo baseado em dados históricos ou referências técnicas	PARCIALMENTE SATISFEITO	Retorno sobre o Investimento (ROI), Estimativas através de <i>Story Points</i> e divisão de IBL's em tarefas
	GPR5	O orçamento e o cronograma do projeto	PARCIALMENTE SATISFEITO	Planejamento de <i>Sprints</i> e Estimativas através de <i>Story Points</i>
	GPR6	Riscos do projeto	SATISFEITO	<i>Daily Meetings</i> , <i>Impediment Backlog</i> e

				responsabilidades do <i>Scrum Master</i>
	GPR7	Planejamento de recursos humanos	SATISFEITO	Definição dos Recursos humanos (Time, <i>Product Owner</i> , <i>Scrum Master</i> ) baseado no perfil
	GPR8	Planejamento das tarefas, os recursos e o ambiente de trabalho	SATISFEITO	Itens adicionais ao <i>Product Backlog</i> e Tarefas da <i>Sprint (Task)</i>
	GPR9	Planejamento de dados relevantes do projeto (armazenamento, privacidade e segurança)	NÃO SATISFEITO	PRÁTICA NÃO MENCIONADA NO SCRUM
	GPR10	Planos para a execução do projeto	SATISFEITO	Visão do Produto, <i>Product Backlog</i> , <i>Sprint Backlog</i> , <i>Sprint planning 1 e 2</i> e <i>Daily meeting</i>
	GPR11	Avaliação de viabilidade de atingir as metas do projeto	SATISFEITO	<i>Sprint Planning (1 e 2)</i> e Definição do objetivo da iteração na elaboração do <i>Sprint Backlog</i>
	GPR12	Revisão e compromisso com o Plano do Projeto	SATISFEITO	<i>Sprint Planning (1 e 2)</i> e <i>Daily meeting</i>
	GPR13	Monitoramento d progresso do projeto	SATISFEITO	<i>Product Burndown</i> , <i>Sprint Burndown</i> , <i>Daily Review</i> e <i>Retrospective Meeting</i>
	GPR14	Gerenciamento do envolvimento das partes interessadas no projeto	SATISFEITO	<i>Daily Scrum Meeting</i> , <i>Sprint Review Meeting</i> e <i>Sprint Retrospective Meeting</i>
	GPR15	Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento	SATISFEITO	<i>Sprint Review Meeting</i>
	GPR16	Identificação e registros de problemas	SATISFEITO	<i>Daily Scrum Meeting</i> e <i>Impediment Backlog</i>
	GPR17	Ações para corrigir e prevenir desvios em relação ao planejado	SATISFEITO	<i>Daily Scrum Meeting</i> , <i>Impediment Backlog</i> e <i>Retrospective Meeting</i>

O processo de gerenciamento de requisitos (GRE) é composto por cinco resultados esperados e o resumo da análise para este processo é apresentado na Tabela 3.

**Tabela 3. Resumo do mapeamento entre Scrum e área GRE**

MPS.BR - Nível G			SCRUM	
ÁREA DE PROCESSO	ABREV.	OBJETIVOS	CLASSIFICAÇÃO	RESUMO DAS PRÁTICAS

GERENCIAMENTO DE REQUISITOS (GRE)	GRE1	Entendimento dos requisitos	SATISFEITO	Visão do produto, Definição de Itens de <i>Backlog</i> e elaboração das tarefas da <i>Sprint</i>
	GRE2	Aprovação de requisitos	SATISFEITO	Aprovação de requisitos através do <i>Business Value (BV)</i> , Priorização de requisitos (ROI) e <i>Sprint Planning (1 e 2)</i>
	GRE3	A rastreabilidade bidirecional entre os requisitos e os produtos de trabalho	NÃO SATISFEITO	PRÁTICA NÃO MENCIONADA NO SCRUM
	GRE4	Revisões em planos e produtos de trabalho do projeto	SATISFEITO	<i>Daily Scrum Meeting</i> e <i>Sprint Review Meeting</i>
	GRE5	Gerenciamento de mudanças nos requisitos	SATISFEITO	<i>Sprint Planning (1 e 2)</i> , <i>Daily Scrum Meeting</i> e <i>Sprint Review Meeting</i>

#### 4.1. Resultado do mapeamento

Este comparativo entre o *Scrum* e as áreas de processo do MPS.BR nível G, teve por objetivo mapear o alinhamento e a conformidade entre eles. A partir deste mapeamento descobrimos que o *Scrum* não satisfaz totalmente o nível do G do MPS.BR conforme apresentado na Figura 1.

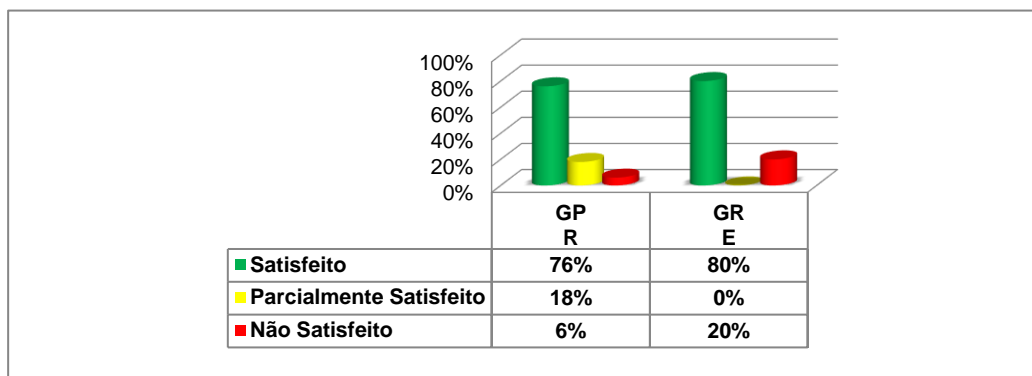


Figura 28. Resultado Geral da Avaliação.

O resultado obtido no mapeamento se deve a diversos fatores, ou seja, há algumas lacunas e atividades ausentes no *Scrum* que são apresentadas na extensão realizada neste trabalho (seção 5).

### 5. Estendendo o Scrum para o MPS.BR nível G

Para as lacunas e atividades ausentes no *Scrum* denominadas “GAP”, ou seja, os resultados esperados classificados como “**Parcialmente Satisfeito**” e “**Não satisfeito**”, algumas soluções complementares foram adicionadas ao processo conforme descrição na extensão seguinte:

#### 5.1. GAP1 - lacunas na definição de um método apropriado de complexidade ou tamanho para as tarefas e os produtos de trabalho do projeto, impactando diretamente o GPR2.

Para realização da estimativa das tarefas e os produtos de trabalho foi definido a combinação de estimativas por *Story Points* com a técnica por complexidade *Planning Poker*, que consiste em um jogo de cartas numeradas com a série de *Fibonacci*.

## **5.2. GAP2 – ausência de estimativa do esforço e o custo para a execução das tarefas e dos produtos de trabalho com base em dados históricos ou referências técnicas**

Para definição dos custos e esforço serão analisados dados históricos organizacionais mantidos através em um documento que contempla informações de projetos anteriores.

## **5.3. GAP3 – ausência de definição do orçamento**

Para definição do orçamento será utilizada a política de negociação da empresa em questão. Este documento fornece orientações sobre os procedimentos a serem observados na negociação dos projetos de software da empresa.

## **5.4. GAP4 - ausência de privacidade e segurança no acesso às informações**

A proposta é armazenar as informações coletadas no projeto utilizando a política de segurança, acesso e armazenamento das informações de projetos. Este documento define a estratégia para implementação nos projetos da empresa no que se refere à infraestrutura (confiabilidade, segurança e estabilidade) e nível operacional (níveis de usuários, responsabilidades, limites e controle de versões).

## **5.5. GAP5 - ausência de rastreabilidade entre os requisitos e produtos de trabalho**

Para a rastreabilidade vertical cada item de *Backlog* (IBL) será inicialmente identificado, então, na criação do *Sprint Backlog* cada tarefa estará associada a um IBL e os artefatos de engenharia estarão associados ao IBL correspondente. Portanto, a rastreabilidade vertical pode ser distribuída em cada artefato gerado, como por exemplo, o código implementado.

## **6. Adaptando a extensão em processo de uma fábrica de software**

Para validação deste trabalho, foi implementada a extensão proposta na CRIATIVA tecnologia, que é uma empresa de tecnologia que cria produtos e serviços utilizando tecnologia da informação (TI). Atualmente a empresa conta com cerca de trinta colaboradores, atendendo diversos clientes e atuando em três áreas de negócio: fábrica de software, treinamentos e prestação de serviços técnicos em equipamentos eletrônicos.

### **6.1. Histórico de melhoria de processos da CRIATIVA**

Os trabalhos em melhoria de processo de software na CRIATIVA iniciaram cerca de três anos após a sua fundação (2003), onde após um crescimento significativo de sua base de clientes, a empresa começou a apresentar diversos problemas relacionados ao processo de produção de software. Baseado neste contexto, foi realizada a modelagem de um processo de desenvolvimento baseado no guia MPS.BR nível G REF9 (2006), onde no período inicial pode-se notar melhorias, no entanto, após alguns meses de utilização, alguns fatores impactaram negativamente na empresa em relação ao processo que foi definido.

Em seguida foi proposto o desenvolvimento de um novo processo de *software* REF10 (2009) que proporcionasse maturidade para a organização e que ao mesmo tempo tivesse seu foco nos princípios do manifesto ágil.

## 6.2 O novo processo de software da criativa tecnologia

A partir de uma perspectiva de gerenciamento e baseado no ciclo de desenvolvimento iterativo incremental definido por REF6 (2008) para o framework *Scrum*, chegou-se a um processo composto por 3 (três) fases: *Pregame*, *Game* e *Postgame*, apoiadas por atividades de Monitoramento e Controle conforme ilustrado na Figura 2.

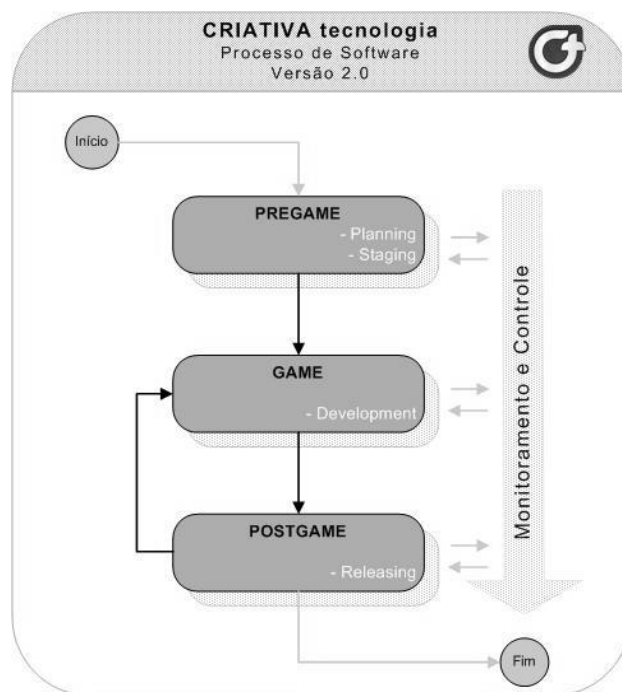


Figura 29. Processo de Software Macro da CRIATIVA tecnologia.

Com o objetivo de proporcionar o entendimento para os colaboradores da empresa em questão, para todas as fases deste novo processo foram elaborados fluxos de trabalho, apresentando suas etapas, objetivos, conceitos chave e quadros de estruturação de cada atividade com seus respectivos papéis, entradas, saídas e artefatos.

**a) Fase PREGAME:** Esta fase tem como objetivos delimitar o escopo do projeto, verificar a viabilidade econômica e eliminar riscos a partir de uma arquitetura estável. A mesma foi dividida em duas sub-fases: *PLANNING E STAGING*.

### a.1) Conceitos chave da sub-fase *Planning*

A criação da Visão do Produto é de responsabilidade do *Product Owner*, porém deve-se frisar que o *Scrum* não estimula a documentação desnecessária, mas é importante que toda a equipe tenha entendimento do produto que está se tentando alcançar. Este documento deve ser curto e acessível por todos.

O *Product Backlog* é outro artefato de responsabilidade do *Product Owner* (PO) mas também é atualizado pelo time, em comum acordo com o PO. Pode conter requisitos funcionais, não funcionais e outras questões.

### a.2) Conceitos chave da sub-fase *Staging*

Esta sub-fase é uma continuação da sub-fase inicial *Planning*, mas com o foco voltado à definição de algumas atividades, como: organização da infra-estrutura para

desenvolvimento do projeto, definição da arquitetura, definição do plano de comunicação, elaboração do *Release Plan*, realiza da reunião *kick-off*.

**b) Fase GAME:** esta fase tem como objetivos criar releases do produto podendo obter versões funcionais do *software*. Nesta direção, o time realiza algumas atividades como: dividir os Itens de *Backlog* (IBL) que foram selecionados para a *Sprint* em tarefas, realizar estimativas, priorizar e analisar a viabilidade dos IBL's. Durante as reuniões diárias, as tarefas são selecionadas para cada membro do time, o compromisso é obtido e uma parte do produto é desenvolvida com base na arquitetura definida e com ênfase no gerenciamento de custos, recursos e qualidade.

### **b.1) Conceitos chave da fase *Game***

Esta fase se inicia com as reuniões da *Sprint* que são divididas em dois níveis, *Sprint Planning 1* e *Sprint Planning 2*. A *Sprint Planning 1* é uma reunião que tem por objetivo diversas atividades, tais como: verificação de dados históricos organizacionais, priorização de IBL da *Sprints*, seleção de IBL que irão compor a *Sprint* e elaboração da *Sprint Backlog*. A *Sprint Planning 2* é a segunda reunião de planejamento com participação do time e *Scrum Master* para realização de atividades como: definição de tarefas (*Task*) necessárias para realização da *Sprint* e segundo nível de estimativa em horas, divisão das tarefas da *Sprint* entre a equipe, análise da capacidade produtiva da equipe, atualização do *Task Board*;

Ao final da *Sprint Planning 1* e 2, a obtenção de compromisso e revisões no planejamento foram realizadas por todos os envolvidos.

A partir disto é executada a *Sprint*, que consiste no desenvolvimento das tarefas definidas para cada IBL, com o objetivo de obter um produto potencialmente entregável.

Durante a execução da *Sprint* é realizada a atividade *Daily Scrum*, que é uma reunião diária onde o time monitora o progresso da execução das atividades e procura identificar os impedimentos encontrados.

Posteriormente, o *Scrum Master* trabalha no sentido de elaborar ações para resolver tais impedimentos e disseminar a solução para o time. Atualizações na *Sprint Backlog* e *Product Backlog* são realizadas para controlar o progresso das *Sprints*.

**c) Fase POSTGAME:** esta fase tem como objetivos apresentar o produto ao cliente para validação, realizar uma reunião para melhoria do processo e avaliar a capacidade produtiva do time.

### **c.1) Conceitos chave da fase *POSTGAME***

A reunião de revisão da *Sprint*, chamada de *Sprint Review Meeting* fornece algumas visões para todos os envolvidos, tais como: visibilidade se a meta da *Sprint* foi alcançada, apresentação de resultados, Inspeção de funcionalidades para possíveis mudanças e adaptações e a reunião de retrospectiva (*Sprint Retrospective Meeting*).

Ao final desta fase caso haja nova *Sprint*, ou seja, caso ainda exista a necessidade de serem desenvolvidas outras funcionalidades, novos ciclos são realizados iniciando-se pela fase *GAME*.

Ao final, informações coletadas no projeto são armazenadas para utilização em projetos futuros formando a base de dados históricos organizacionais.

## 7. Estudo de Caso

Visando avaliar os efeitos da aplicação da extensão proposta neste trabalho, foi realizado o estudo de caso na empresa em questão em projetos de *software* selecionados com base na semelhança das características de domínio, escopo, arquitetura, tempo, custo e equipe, através das métricas que mais estavam impactando negativamente nos projetos, tais como: entregas no prazo, satisfação do cliente e quantidade de *Bugs*.

Na avaliação de entregas no prazo foi possível observar que houve um pequeno desvio em relação à capacidade de produção do time devido ao fato de não existir projetos anteriores que utilizaram este novo processo. Mas apesar de não conseguir completar 100% das funcionalidades planejadas, o novo processo conseguiu demonstrar que as entregas dentro do prazo do processo anterior em relação a esta nova abordagem, apresentam uma melhoria de 42,50%.

Para medir a satisfação dos clientes, foi aplicado um questionário em relação a diversos aspectos, tais como: comunicação, qualidade do produto, gerenciamento de mudanças, entregas no prazo, dentre outros. Para avaliar o processo foram utilizados três indicadores, sendo que para cada questão, o questionário disponibilizou orientações de como ela deve ser avaliada. Foi possível observar que no projeto que utilizou o novo processo de desenvolvimento obteve aumento de 57,14% das respostas “atendem totalmente” no questionário, proporcionando uma grande diminuição nos valores para as respostas que atendem parcialmente (57,14%) e manutenção da ausência de respostas que não atendem o cliente.

Para medir a quantidade de *Bugs*, foram coletados os dados na ferramenta *Bug Tracker* da empresa e pode-se observar que o Projeto que utilizou o novo processo teve um número menor de *Bugs* que o processo anterior.

O estudo de caso realizado apresentou os resultados obtidos através da implementação da proposta de melhoria do processo de software deste projeto. Através das métricas apresentadas anteriormente, pode-se observar os efeitos de agilidade e maturidade em relação à abordagem anterior.

Alguns aspectos nos projetos que utilizaram o novo processo podem ser ressaltados, tais como:

- O quadro de tarefas (*Taskboard*) proporcionou uma grande visibilidade do andamento das tarefas;
- A participação da equipe nas decisões do projeto proporcionou comprometimento e motivação na realização das tarefas diárias;
- Através das reuniões diárias pode-se monitorar a produção do time, promover *feedback* e remover impedimentos que viessem a atrasar o bom andamento do desenvolvimento da *Sprint*.

## 8. Conclusão

No mapeamento entre o *Scrum* e o guia MPS.BR nível G, descobriu-se que ele não cobre totalmente os resultados esperados do guia, mas pode ser um ponto de partida, pois proporciona uma base fundamental para empresas que estão iniciando melhoria de processos, principalmente as que dispõem de poucos recursos como as micro e pequenas empresas.

O gerenciamento de requisitos através de abordagens ágeis proporcionou uma forma alternativa e eficaz de gerenciar requisitos e solicitações de mudanças durante o desenvolvimento do projeto evitando o retrabalho.

A abordagem de gerenciamento ágil de projetos foi uma alternativa capaz de proporcionar a inovação, diminuição do tempo de entrega dos projetos, desenvolvimento de produtos de trabalho que agregam valor real para o cliente e resultados para a empresa.

O novo processo proposto demonstrou um avanço inicial no sucesso de um projeto de software desenvolvido pela empresa, mas como se encontra ainda em fase de implantação, o monitoramento se faz necessário para obtenção da melhoria contínua e obtenção de melhores resultados.

## **9. Referências Bibliográficas**

- (REF1, 2003) Adm, Advanced Development Methods. Scrum Methodology – Incremental, Iterative Software. Development from Agile Processes.
- (REF2, 2002) Beedle, M., and Schwaber, K.. Agile Software Development With Scrum. New Jersey, Books.
- (REF3, 2006) Boehm, B.. “A View of 20th and 21st Century Software Engineering.” ICSE06. Shanghai, China, ACM.
- (REF4, 2004) Davis, C, et al.. An Agile Approach to Achieving CMMI. [http://www.agiletek.com/images/AgileTek/pdf/an\\_agile\\_approach\\_to\\_achieving\\_cm mi.pdf](http://www.agiletek.com/images/AgileTek/pdf/an_agile_approach_to_achieving_cm mi.pdf) (acesso em 27 de Dezembro de 2008).
- (REF5, 2008) Playfair, K. “When 'General Agile' Isn't Enough - Why Scrum Wins in the Enterprise.” Agile Journal. <http://www.agilejournal.com/content/view/808/111/> (acesso em 29 de Dezembro de 2008).
- (REF6, 2008) Schwaber, K.. Agile Project Management With Scrum. Redmond, Microsoft Press.
- (REF7, 2007) Softex. MPS.BR, Melhoria de Processo de Software Brasileiro - Guia Geral V1.2. Rio de Janeiro, Softex.
- (REF8, 2004) Standish, The Standish Group International. (2004).. “The Chaos Report.” Standish Group. [secure.standishgroup.com/reports/reports.php?rid=500](http://secure.standishgroup.com/reports/reports.php?rid=500).
- (REF9, 2006) Szimanski, F.. Implementando MPS.BR nível G na melhoria do processo de software em uma pequena empresa. Simpósio Mineiro de Sistemas de Informação. Lavras, UFLA.
- (REF10, 2009) Szimanski, F. Extensão do Scrum segundo as áreas de processo do MPS.BR nível G. Dissertação de mestrado. Recife, CESAR.
- (REF11, 2006) GLASS, Robert L. “The Standish Report: Does It Really Describe a Software Crisis?” COMMUNICATIONS OF THE ACM, Agosto de 2006.
- (REF12, 2008) MCCONNELL, Steven C. “Classic Mistakes Enumerated.” Web Site Steve Mcconnell. 1996. <http://www.stevemccconnell.com/rdenum.htm> (acesso em 20 de Abril de 2008).



# Interfaces Dinâmicas e Adaptativas para Celulares

Flavio Henrique Cardoso de Freitas, Silvano Maneck Malfatti

Faculdade Católica do Tocantins (FACTO)  
Palmas – TO – Brasil

{flaviohcf,silvanomalfatti}@hotmail.com

***Resumo.** Devido ao fácil acesso e às diferentes formas de utilização do celular, torna-se necessário que a interface utilizada para acessar as funcionalidades do sistema seja mais amigável e dinâmica, de maneira que se adapte às necessidades do usuário. Este trabalho propõe um modelo de interface que dinamicamente apresenta as funcionalidades de um celular. O principal objetivo do modelo de interface proposto é se adaptar ao perfil do usuário no intuito de oferecer uma melhor usabilidade e agilidade na execução de tarefas.*

**Palavras chave:** celular, interface, usabilidade.

## 1. Introdução

O mercado de celulares cresceu muito nos últimos anos, aumentando cada vez mais o número de usuários que possuem pelo menos um aparelho. Esta tendência pode ser confirmada em números, onde de acordo com o portal G1 (2009) até o final de 2009 duas em cada três pessoas terão um aparelho móvel.

Os celulares já deixaram a um bom tempo de serem apenas ferramentas destinadas à comunicação, passando a incorporar aplicativos de alta tecnologia com diversas funcionalidades como aplicações multimídia, tocadores de áudio e vídeos de alta resolução, câmeras fotográficas, GPS (*Global Positioning System*), acesso à Internet e até mesmo TV. O iPhone da empresa APPLE (2009) é um bom exemplo dessa evolução.

Devido aos avanços tecnológicos os celulares estão cada vez mais cobiçados por um público bastante heterogêneo, que abrange tanto homens e mulheres, desde crianças até idosos. Por esse motivo, garantir a usabilidade, disponibilizando aparelhos fáceis de usar e que agradem os usuários, é uma das maiores preocupações das empresas que desenvolvem celulares.

Este trabalho apresenta um modelo de interface dinâmico para acesso ao sistema de dispositivos móveis, denominado JInterfaceDin e desenvolvido em J2ME (*Java 2 Micro Edition*). A principal característica do modelo de interface proposto é a sua adaptação de acordo com o perfil do usuário no intuito de oferecer um mecanismo simples e que possibilite a agilidade na execução de tarefas mais rotineiras.

O trabalho encontra-se organizado da seguinte maneira: na seção 2 é discutida a implicação da usabilidade em dispositivos móveis; na seção 3 é apresentada a arquitetura do J2ME bem como os motivos que levaram à escolha dessa plataforma; na seção 4 são apresentados os conceitos adotados no projeto JInterfaceDin; por fim são apresentados os resultados obtidos e trabalhos futuros.

## 2. Usabilidade em Dispositivos Móveis

O termo usabilidade foi definido pela primeira vez pela norma ISO/IEC 9126 em 1991, sendo caracterizado como a capacidade que um sistema interativo oferece a seu usuário, em determinado contexto de operação, para a realização de tarefas de maneira eficaz, eficiente e agradável.

A preocupação com a usabilidade em sistemas interativos surgiu em paralelo com o desenvolvimento de novas tecnologias. Observando a evolução dos celulares é possível notar que os primeiros modelos não possuíam interfaces gráficas. Porém, à medida que os aparelhos foram se modernizando, o teclado foi sendo reduzido e telas e interfaces gráficas foram ganhando espaço. Atualmente, existem modelos em que o teclado foi excluído totalmente dando lugar às telas sensível ao toque. A Figura 1 apresenta um comparativo entre um celular moderno com um dos primeiros modelos de celular lançados.



Figura 1. Comparativo entre a interface de um celular atual com um dos primeiros modelos.

Essa tendência reflete a necessidade das interfaces atuais para celulares atenderem aos princípios de usabilidade que, segundo NIELSEN (1993), são definidos por atributos como facilidade de aprendizado, eficiência de uso, facilidade de memorização, baixa taxa de erros, consistência e flexibilidade.

Em celulares, a interface deve ainda prezar por questões intrínsecas aos dispositivos móveis como memória restrita, limitada capacidade de processamento, bem como a forma como a informação será apresentada em uma tela com tamanho reduzido sem que haja uma sobrecarga da área de visualização.

Além disso, o tempo de acesso e o custo de alguns serviços são fatores críticos para usuários móveis. O acesso a um site WAP é cobrado do usuário, assim como a transferência de dados via web. Nesse sentido, após acessar uma aplicação remota, deve-se minimizar o número de cliques e telas necessárias para executar uma determinada tarefa.

Outra questão importante refere-se à facilidade de navegação, tendo em vista que a capacidade limitada das telas, interrupções frequentes e possível falta de atenção podem fazer com que o usuário fique perdido no sistema. Neste sentido, a utilização de menus complexos compostos por muitos submenus deve ser substituída por menus mais simples.

É preciso também notar que apesar do celular de hoje incorporar dezenas de aplicativos, sua finalidade ainda é permitir a comunicação. Sendo assim, a interação móvel pode ser interrompida a qualquer momento por diversas razões que vão desde uma chamada, falhas na conexão ou descarregamento da bateria. Portanto, o sistema deve oferecer mecanismos para que o usuário possa continuar a sua tarefa após o retorno de alguma situação especial.

Para ABREU (2004), o excesso de recursos e informações pode atrapalhar e confundir os usuários inexperientes. Geralmente, usuários inexperientes não tem necessidade de utilizar os recursos avançados do aparelho, mas apenas suas funcionalidades básicas como agenda, despertador, gerenciador de mensagens e discador. Este problema aumenta à medida que os fabricantes de celulares adicionam cada vez mais tecnologias aos dispositivos sem considerar que dependendo do perfil do usuário tais recursos não serão utilizados, pois para isto seria necessário que o mesmo recebesse treinamento.

Em uma pesquisa realizada por SANTOS (2007) concluiu-se que um dos principais motivos para os usuários trocarem seus aparelhos é a busca por novas tecnologias. Entretanto, quando os entrevistados foram questionados sobre o que mudariam na interface do aparelho atual, a maioria respondeu que adicionaria atalhos para os comandos mais utilizados. Além disso, quando os usuários foram questionados sobre o que achavam dos aparelhos atuais, a maioria respondeu que os celulares têm muitos recursos que não são utilizados. Estes resultados mostram que grande parte dos usuários troca de aparelho mais por uma questão de *status* do que por necessitar dos novos recursos que estão sendo agregados aos aparelhos mais recentes.

No intuito de amenizar os problemas descritos torna-se necessária a busca por interfaces gráficas para celulares que atendam aos preceitos definidos pelas normas de usabilidade e, dessa forma, possam melhorar o processo de interação entre usuário e dispositivo. Portanto, se a tendência são aparelhos celulares incorporarem cada vez mais tecnologias, deve ser oferecido aos usuários mecanismos que possibilitem a configuração da interface do sistema no sentido de que os recursos não utilizados não precisam ser apresentados.

### 3. A Plataforma J2ME

A plataforma J2ME é uma versão reduzida da linguagem Java destinada ao desenvolvimento de aplicações para dispositivos móveis. Nessa plataforma, a implementação da máquina virtual é bastante enxuta, o que significa que muitos dos recursos da linguagem Java não estão disponíveis em razão da limitação computacional de dispositivos móveis como celulares.

A arquitetura do J2ME é baseada nos conceitos de configuração e perfil que funcionam sobre a KVM (*Kilobyte Virtual Machine*), como mostra a Figura 2.

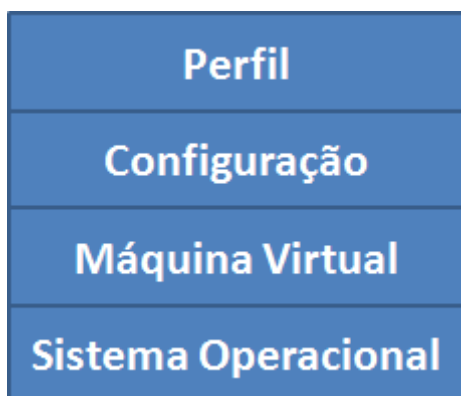


Figura 2. Arquitetura de camadas da J2ME.

A primeira camada da arquitetura J2ME é composta pela máquina virtual Java, que foi desenvolvida especialmente para dispositivos com pouca capacidade computacional e que possuam processadores de 16 ou 32 bits.

A camada de configuração implementa o conjunto de classes que fornecem as funcionalidades básicas para a programação de acordo com os recursos oferecidos pelo dispositivo. Atualmente existem dois tipos de configuração, a CDC (*Connected Device Configuration*) e a CLDC (*Connected Limited Device Configuration*).

Segundo JOHNSON (2008), a CDC é destinada a dispositivos móveis com maior capacidade computacional, que possuam processadores de 32 bits e pelo menos 2 MB de memória RAM. A CLDC por sua vez, é destinada a dispositivos que possuem grandes restrições no que se refere à capacidade de processamento, memória e resolução gráfica.

A última camada corresponde ao perfil, que nada mais é do que uma extensão da camada de configuração e que complementa os recursos oferecidos pela J2ME. Através dos recursos implementados pelo perfil é possível criar aplicações gráficas, realizar a persistência de dados e estabelecer conexões. Assim como ocorre com as configurações existem basicamente dois tipos de perfil no J2ME, o PP (*Personal Profile*) e o MIDP (*Mobile Information Device Profile*). É importante salientar que cada perfil está associado a somente uma das configurações.

O PP é o perfil mais amplo e está associado ao CDC para oferecer recursos como suporte à *applets* e interfaces no estilo AWT (*Abstract Window Toolkit*). O MIDP está associado ao CLDC, sendo assim é indicado para equipamentos mais simples e

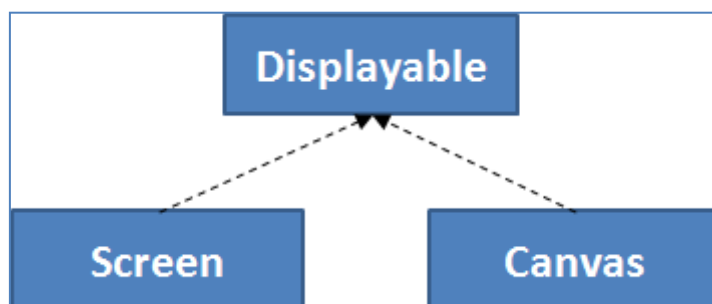
com interfaces gráficas menos sofisticadas, como a grande maioria dos celulares encontrados no Brasil.

Graças a esta arquitetura multicamadas, o J2ME pode garantir a principal característica da linguagem Java na qual após uma aplicação ter sido codificada ela possa ser executada em diversos modelos de aparelhos que fazem parte do mesmo subconjunto de dispositivos com configurações definidas pelo CDC ou CLDC. Este é um dos principais motivos pelo qual a plataforma J2ME foi escolhida para o desenvolvimento do projeto JInterfaceDin.

Para SANTOS (2007), o usuário se confunde ao se deparar com a falta de padrões encontrados no desenvolvimento de interfaces destinadas a celulares. Desta forma, a portabilidade do Java garante que um mesmo conceito ou estilo possa ser mantido para aparelhos de diferentes marcas.

Outras vantagens do J2ME incluem a sua gratuidade e programação orientada a objetos em uma linguagem de alto nível que comporta uma série de classes já consagradas no desenvolvimento de aplicações *desktop*. Além disso, sempre existe a possibilidade de integração entre as aplicações móveis J2ME com outras arquiteturas do Java, como a J2SE e J2EE.

Em termos de desenvolvimento de interface, o perfil MIDP escolhido para este projeto, oferece duas formas para o desenvolvimento de aplicações baseadas em componentes visuais, como mostra a Figura 3.



**Figura 3.** Hierarquia de classes oferecidas pelo perfil MIDP para o desenvolvimento de interfaces.

A classe *Displayable* representa um conjunto de objetos visuais que podem ser apresentados na tela do dispositivo. No ramo esquerdo da árvore estão os objetos do tipo *Screen* que oferecem componentes visuais de alto nível e prontos para serem utilizados. Tais componentes incluem áreas de texto, listas de opções e formulários nos quais é possível inserir outros componentes visuais como campos para entrada de informações, rótulos, data, hora, imagens e barras de progressão. Apesar da programação de interfaces por meio de objetos da classe *Screen* ser rápida e fácil, ela é pouco flexível para a elaboração de interfaces customizadas.

No ramo direito da árvore estão os objetos do tipo *Canvas* que representam uma área de desenho onde o programador pode renderizar primitivas gráficas como linhas, quadrados, círculos, imagens e textos. Ao criar uma classe que herda os atributos de *Canvas*, o desenvolvedor tem acesso a um objeto do tipo *Graphics* que permite a

criação de interfaces personalizadas. O modelo de interface descrito na próxima seção foi desenvolvido com base em objetos da classe *Canvas*.

## 4. O Projeto JInterfaceDin

O projeto JInterfaceDin consiste no desenvolvimento de um modelo de interface que define a forma como será realizada a interação e o controle dos recursos e aplicações oferecidos pelos celulares atuais. As soluções empregadas foram baseadas em um conjunto de idéias obtidas a partir dos problemas relatados na seção 2 e consistem dos seguintes itens:

- Sistema de interface dinamicamente adaptável ao modelo do aparelho
- Atalhos dinâmicos
- Menu customizável
- Acessibilidade via som
- Manual eletrônico

### 4.1. Interface Adaptável

Com o intuito de amenizar o problema que envolve a falta de padronização das interfaces entre diferentes aparelhos, a interface do JInterfaceDin tende a se adaptar ao modelo do aparelho. Para tanto são considerados parâmetros como tamanho da tela, resolução, se o visor é colorido ou monocromático e se suporta a renderização de imagens.

Todos estes parâmetros são analisados por um gerenciador de layout no momento em que o sistema é inicializado para que sejam feitas as adaptações necessárias conforme mostra a Figura 4.

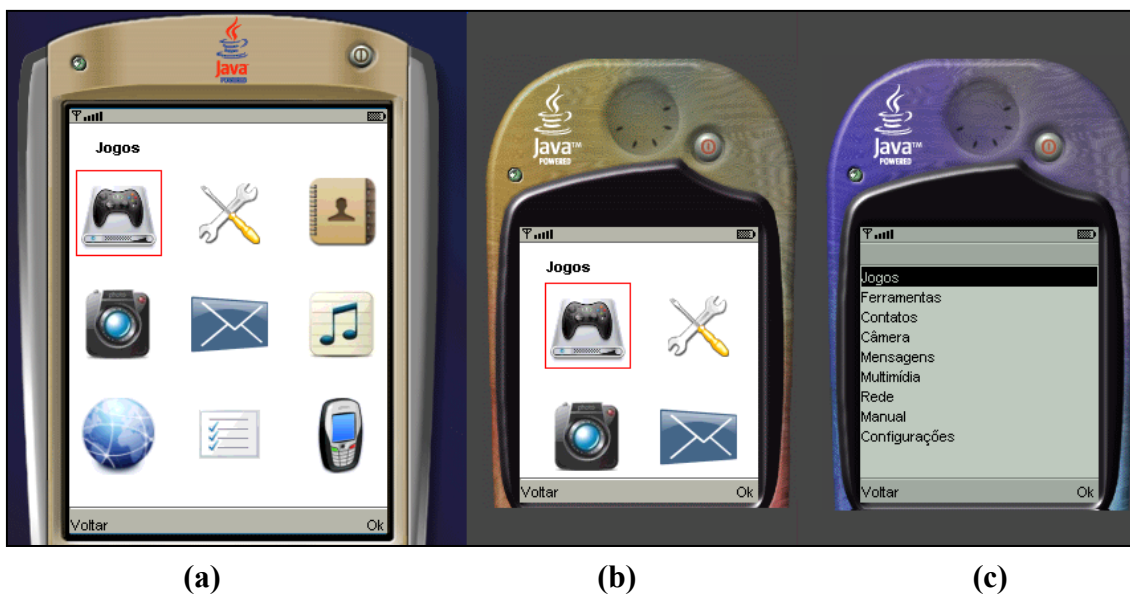


Figura 4. Sistema JInterfaceDin sendo executado em diferentes modelos de celulares.

Como é possível notar na Figura 4, o JInterfaceDin adaptou-se a diferentes modelos de celular, levando em conta o tamanho da tela, conforme os modelos (a) e (b) apresentados na figura, ou ainda, transformando imagens em itens para aparelhos

monocromáticos e que não suportam a renderização de imagens, como apresentado no modelo (c) da figura.

Apesar de não ser possível manter o mesmo padrão de telas entre visualizadores coloridos e monocromáticos sem suporte a imagens, a disposição dos itens e o fluxo das telas é o mesmo, o que preserva o modelo mental que o usuário tem da interface caso venha a trocar de aparelho.

#### 4.2. Atalhos Dinâmicos

Um dos recursos que agiliza o processo de interação em dispositivos móveis é a disponibilização de atalhos para as tarefas mais acessadas. Entretanto, nos modelos de celulares encontrados atualmente que suportam essa funcionalidade, os comandos de atalho são fixos ou necessitam serem configurados pelo usuário.

No projeto JInterfaceDin os atalhos são dinamicamente selecionados com base no total de acessos realizados pelo usuário a uma determinada operação. Esse recurso favorece não só usuários com perfil diferente, como também um mesmo usuário em diferentes situações. Por exemplo, um usuário no período de férias tende a acessar mais a câmera e os jogos do que a agenda de contatos, a qual é muito utilizada no período de trabalho. A Figura 5 apresenta o sistema de atalhos do JInterfaceDin adaptado a diferentes situações.

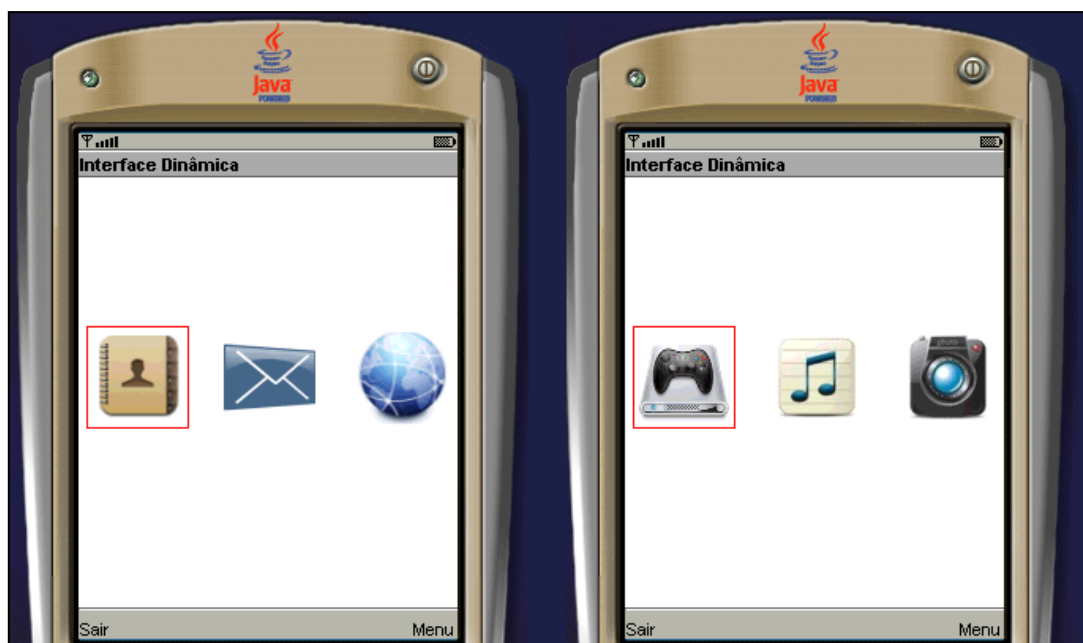


Figura 5. Atalhos dinamicamente selecionados conforme o contexto de uso.

#### 4.3. Menu Customizável

Este recurso foi inserido com o objetivo de diminuir a complexidade da interface através da redução do número de opções oferecidas pelo menu principal. Desta forma, o usuário poderá excluir do menu aqueles itens que não utiliza no dia-a-dia.

Além disso, os itens que não estiverem no menu principal também não concorrem para a geração de atalhos dinâmicos. A Figura 6 apresenta uma configuração personalizada para o menu principal.

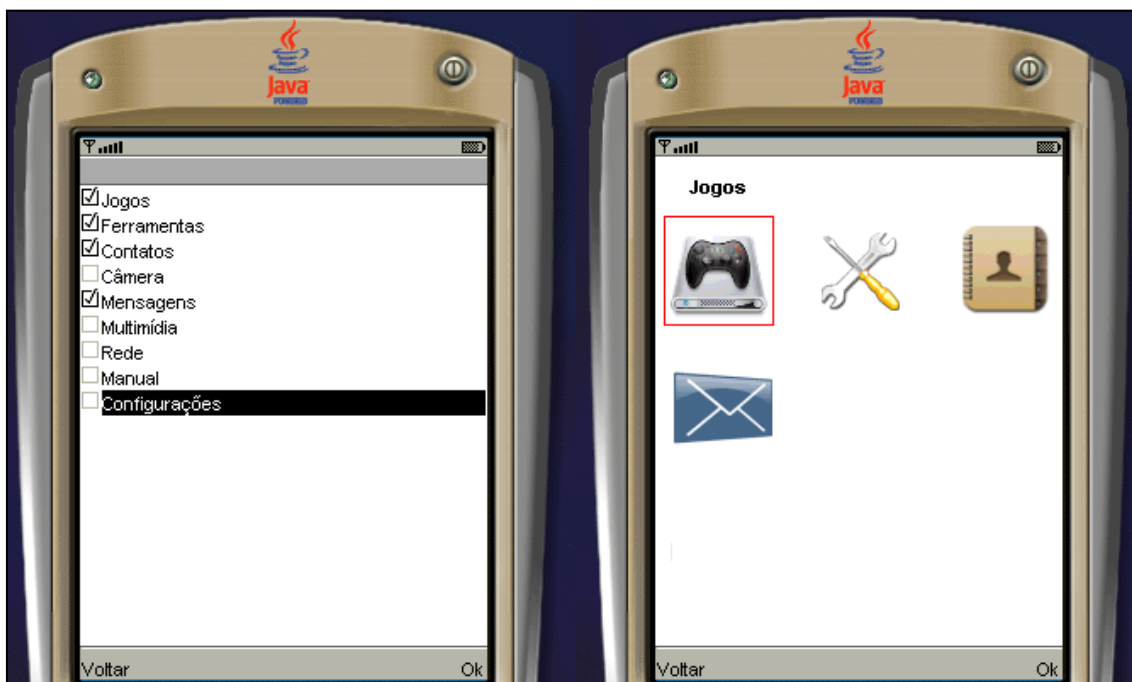


Figura 6. Personalização do menu principal.

#### 4.4. Acessibilidade

Um dos principais problemas em customizar dinamicamente a interface refere-se à questão da acessibilidade, principalmente no caso de usuários com deficiência visual. Para amenizar este problema foi incorporada ao JInterfaceDin uma opção de configuração que habilita um locutor para que sejam narradas as interações que ocorrem com o aparelho.

Para o desenvolvimento deste recurso utilizou-se um programa sintetizador de voz para converter em sons os comandos disponíveis na interface. No momento da inicialização do sistema os sons são carregados e passam a ser emitidos quando o recurso for habilitado.

#### 4.5. Manual Eletrônico

Muitas questões referentes aos recursos do aparelho e forma de manuseio podem ser encontradas no manual de instruções impresso fornecido pelo fabricante. Entretanto, são poucas as pessoas que levam o manual junto consigo no dia-a-dia.

Para amenizar este problema foi incluído ao JInterfaceDin um manual eletrônico que permite ao usuário realizar consultas no intuito de resolver possíveis dúvidas. Porém, as explicações oferecidas pelo manual são bastante sucintas tendo em vista que também foram gerados arquivos de áudio para as mesmas.



## 5. Conclusões e Trabalhos Futuros

Com a popularização dos celulares e o surgimento de novas tecnologias a cada dia, as interfaces utilizadas em celulares tendem a se tornar mais complexas, sem considerar que o consumidor esteja preparado para utilizar tais recursos.

Neste cenário, as interfaces adaptativas podem ser uma saída para a criação de sistemas que melhor se adaptem ao perfil do usuário. Este trabalho apresentou algumas idéias desenvolvidas com o intuito de minimizar os problemas que estão surgindo devido ao excesso de recursos incorporados aos dispositivos móveis.

Como trabalhos futuros pretende-se realizar um estudo de caso com grupos de usuários para avaliar o impacto das idéias adotadas. Para que esta próxima etapa possa ser realizada será necessário desenvolver metodologias de avaliação especiais para usuários móveis, tendo em vista que o contexto no qual o dispositivo é utilizado torna-se um importante parâmetro de avaliação durante os testes.

## 3. Referências Bibliográficas

(G1, 2009) Portal G1. Disponível em:

<http://g1.globo.com/Noticias/Tecnologia/0,,MUL1331059-6174,00-MUNDO+TERA+BILHOES+DE+ASSINATURAS+DE+CELULAR+NO+FIM+D+O+ANO.html>. Acesso em 14/10/2009.

(APPLE, 2009) Disponível em <http://www.apple.com/br/iphone>. Acesso em: 14/10/2009.

(NIELSEN, 1993) NIELSEN, Jakob. Usability Engineering. San Francisco: Morgan Kaufmann, 1993.

(ABREU, 2009) ABREU, Leonardo Marques; Moraes, Anamaria de. Usabilidade de telefones celulares com base em critérios ergonômicos. Rio de Janeiro, 2005. 294p. Dissertação de Mestrado - Departamento de Artes & Design, Pontifícia Universidade Católica do Rio de Janeiro.

(SANTOS, 2007) SANTOS, Robson; FREITAS, S.; SOUSA, C. Usabilidade de aparelhos celulares: um estudo de consumidores de classes C e D. In: 4o Congresso Internacional de Pesquisa em Design Brasil, 2007, Rio de Janeiro.

(JOHNSON, 2007) JOHNSON, Thienne M. Java para Dispositivos Móveis - Desenvolvendo Aplicações com J2ME. 1. ed. São Paulo: Novatec, 2007.



# Proposta de um Algoritmo Genético para o Problema de Árvore de Steiner Euclidiana Tridimensional

Marcelo Lisboa Rocha<sup>1</sup>, Wellington Gomes Ribeiro<sup>1</sup>

<sup>1</sup>Departamento de Informática – Centro Universitário UNIRG  
Av. Antônio Neves da Silva S/N, Parque das Acácias, Gurupi - TO, CEP: 77400-000

marcelolisboarochoa@yahoo.com.br, eltinguitarra@hotmail.com

**Resumo.** *A Árvore de Steiner Euclidiana é uma tecnologia muito utilizada para resolver vários problemas no mundo atual. Tendo como objetivo, otimizar ligações entre pontos existente, no intuito de melhorá-las para que tenham um menor custo, obtendo este resultado com um menor tempo. Foram utilizadas algumas tecnologias como o Algoritmo Genético, para melhoramento das ligações como soluções. Mostra um ganho de desempenho com a aplicação da Busca Local sobre o Algoritmo Genético. Com a utilização destas tecnologias foi possível se chegar a um resultado aceitável comparado ao algoritmo que hoje é o melhor para resolver tais problemas.*

**Palavras chaves:** *AGPASE 3D – Algoritmo Genético. Árvore de Steiner Euclidiana 3D. Otimização Combinatória.*

## 1. Introdução

“O Problema de Árvore de Steiner Euclidiana (PASE) aplicado em um ambiente 3D é um problema da classe NP-difícil” (GAREY and JOHNSON, 1979), que torna inviável a obtenção da solução ótima e a garantia de que a mesma é ótima, dado o esforço necessário para realização da tarefa e a complexidade da mesma.

O PASE é um problema onde sua função principal é mapear ligações em um determinado ambiente, a partir de pontos fixos e, fazendo com que estas ligações sejam as melhores possíveis (menor distância entre os pontos) e sua estratégia é colocar pontos intermediários que sirvam como apoio para diminuir o custo (tamanho) das arestas e assim gerar uma Árvore de Steiner Mínima (ASM). Este problema é aplicado no  $R^n$  (espaço multidimensional onde o  $n$  é o número de dimensões), neste caso específico  $n = 3$  (tridimensional – 3D).

Na Figura 1, está sendo apresentado um exemplo de aplicação usando uma Árvore Geradora no  $R^2$  (plano, duas dimensões) sem pontos de Steiner, onde se pode ver a forma como os pontos são ligados. Já na Figura 2, é apresentado o mesmo exemplo, mas utilizando pontos de Steiner, onde se pode verificar a forma de ligação das arestas e vértices.

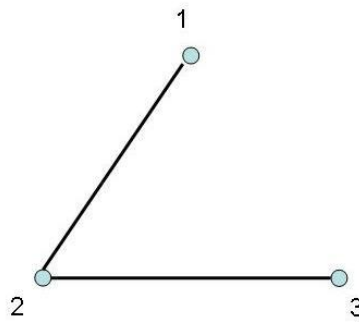


Figura 30 - Árvore com 3 pontos obrigatórios ( $P=3$ ) no  $R^2$ .

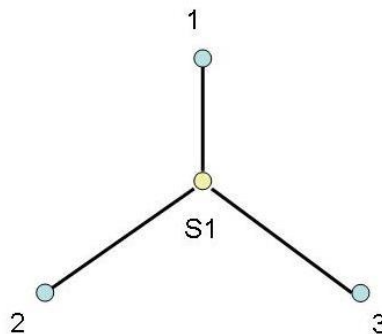


Figura 31 - Árvore de Steiner com 3 pontos obrigatórios ( $P=3$ ) e um pontos de Steiner ( $S1$ ) aplicado no  $R^2$ .

Supondo que as distâncias Euclidianas entre os pontos fixos (em azul) é 1, o custo/tamanho da Árvore Geradora Mínima (AGM) apresentada na Figura 1 é “2”. Já o custo/tamanho da Árvore de Steiner Euclidiana Mínima (ASEM) apresentada na Figura 2 é “1,754”, comprovando que a utilização de pontos extras (pontos de Steiner em amarelo) produz uma árvore de tamanho menor que o da AGM.

Existem diversas aplicações tanto práticas quanto teóricas que fazem uso ou podem ser modeladas como uma Árvore de Steiner Euclidiana 3D (ROCHA et al., 2007), (SMITH, 1998) e (OLIVEIRA, 2005). Assim sendo, faz-se importante o desenvolvimento e utilização de métodos eficientes aplicados ao problema em questão.

Contudo, de acordo com (ROCHA et al., 2007) e (FAMPA and ANSTREICHER, 2008), atualmente existem apenas algumas heurísticas que encontram soluções razoáveis para instâncias de tamanho médio e dois métodos exatos aplicáveis apenas a instâncias pequenas.

O Algoritmo Genético (AG) é uma técnica baseada na evolução natural dos seres vivos. Com ele, buscam-se novas soluções e combinações que possibilitem o melhoramento genético. Ele é utilizado para gerar possíveis resultados cada vez melhores para o problema em questão. Para isso é aplicado os operadores genéticos *Crossover* e *Mutação* para melhoramento dos resultados.

Será aplicada a técnica de Busca Local para trabalhar sobre os resultados encontrados pelo Algoritmo Genético, no intuito de melhorar os mesmos. Serão também apresentados os resultados que foram obtidos neste trabalho e será feita uma comparação com um algoritmo existente na literatura atual, que é considerado como

sendo o melhor trabalho existente, para que se possa comparar o desempenho obtido com esta aplicação.

## 2. Referencial Teórico

Nesta primeira parte serão expostos os algoritmos, as ferramentas e as tecnologias que foram utilizadas para o desenvolvimento deste trabalho. Serão apresentadas as tecnologias como: o Espaço Euclidiano, a Árvore de Steiner, e o Algoritmo Genético, que foram o foco das ferramentas utilizadas neste trabalho.

### 2.1. Espaço Euclidiano

“A Distância Euclidiana é uma das medidas de similaridade entre comunidades mais utilizadas na prática” (GAUCH, 1982). De acordo com BROWER e ZAR (1977), quanto menor o valor obtido pela Distância Euclidiana entre duas comunidades mais próximas elas se apresentam em termos de parâmetros quantitativos por espécie.

Segundo BROWER e ZAR (1977), a Distância Euclidiana é obtida pela seguinte expressão:

$$DE = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

DE = Distância Euclidiana

$x_i$  = Coordenada/Posição na dimensão  $i$  para o ponto  $x$ ;

$y_i$  = Coordenada/Posição na dimensão  $i$  para o ponto  $y$ ;

$n$  = Número de dimensões.

Com a aplicação do Calculo Euclidiano, que trabalha com foco na otimização das distâncias entre os pontos, levando em consideração a posição de cada ponto, poderá se chegar a um resultado aceitável.

### 2.2. Árvore de Steiner Euclidiana

Na Figura 3 pode ser visto um exemplo de uma árvore natural comparada a uma árvore do mundo computacional. Uma diferença encontrada é que, para o mundo computacional, a árvore é expressa de cabeça para baixo, começando do nó raiz e indo até o mais baixo nível que são as folhas (BOAVENTURA NETTO, 2001).

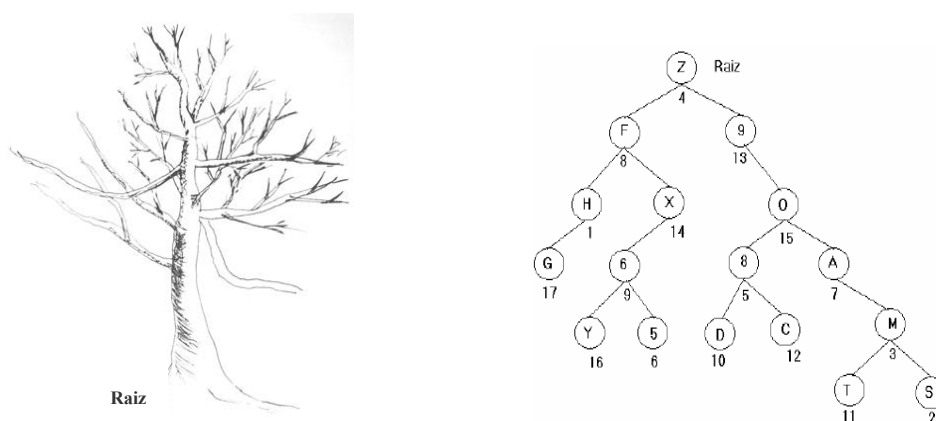
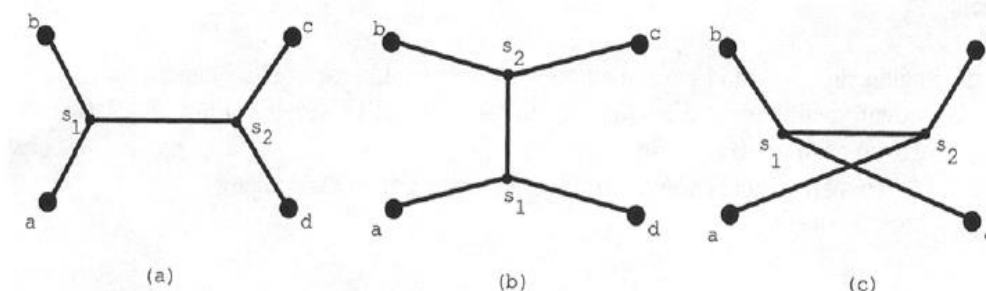


Figura 32 - Árvore natural e árvore computacional.

O problema da Árvore de Steiner Euclidiana, segundo (HERRING, MICHAEL, 2004), é classificado como NP-Completo, tornando impossível de se resolver manualmente e bastante difícil de obter a solução ótima em um tempo computacional aceitável fazendo uso de métodos exatos, mesmo com uma execução computadorizada.

Para que uma árvore seja considerada de Steiner, ela precisa ter pontos intermediários (pontos de Steiner) que irão ligar os pontos fixos (para diminuir o custo das ligações) construindo a Árvore de Steiner Mínima - ASM. Cada árvore tem um número máximo de pontos de Steiner que podem ser inseridos na mesma. Quando o número máximo de pontos auxiliares é inserido, a Árvore de Steiner Euclidiana é classificada como uma Árvore de Steiner Euclidiano cheia. Para que a árvore seja considerada como cheia, ela precisa ter  $(n - 2)$  pontos auxiliares (Steiner) inseridos na mesma, onde  $n$  é o número de pontos fixos existentes (ZACHARIASEN, M., 1998).

Na Figura 4 são apresentadas três possíveis resultados de ligações entre os mesmos quatro pontos fixos com a utilização de Pontos de Steiner de uma Árvore, deixando a mesma com o limite máximo, *cheia*.



**Figura 33 - Árvore de 4 pontos fixos e com 2 pontos de Steiner em três possibilidades (WINTER, 1997).**

A aplicação prática do PASE é interessante não só em um ambiente bidimensional, mas muito importante e funcional em um ambiente tridimensional (3D). Uma aplicação interessante é no projeto de uma rede de dutos em uma mina subterrânea, composta por vários túneis que se conectam internamente formando uma espécie de cidade 3D no subsolo.

### **2.3. Aplicações Práticas do Problema da Árvore de Steiner Euclidiana**

Algumas aplicações que podem ser modeladas como um PASE é: configuração molecular (bioquímica), (SMELYANSKIY, et al 2005), (OLIVEIRA, N. V. 2005) e projeto de redes (elétrica, minas, dutos, etc.). A seguir, estará especificado com mais detalhes a configuração molecular (bioquímica) e projeto de redes.

#### **\* Aplicação do PASE na Configuração Molecular**

OLIVEIRA em, (OLIVERA 2005) propôs em sua tese de doutorado um algoritmo denominado APOLONIO, baseado no algoritmo de Smith (SMITH 1992), como também realizou um estudo do problema de Árvore Mínima de Steiner Euclidiana Tridimensional (PASET), relacionando a função razão de Steiner ao problema de configuração de energia mínima de macromoléculas de interesse biológico. A seguir, na Figura 5, tem-se um exemplo de como a estrutura de Árvore de Steiner Euclidiana Tridimensional aparece em algumas configurações moleculares.

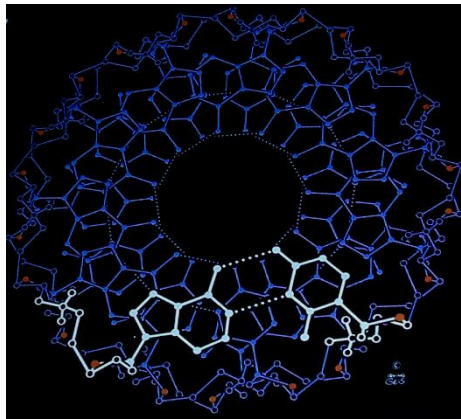


Figura 5 - Forma menos hidratada do DNA (A-DNA) (OLIVEIRA, N 2005).

### \* Aplicação do PASE no Projeto de Rede

No caso da necessidade de realização do projeto de uma rede de tamanho mínimo em uma estrutura tridimensional, pode-se criar um modelo baseado no Problema da Árvore de Steiner Euclidiano Tridimensional (PASET), onde os pontos fixos são os pontos da rede que se deseja conectar. Assim sendo, resolvendo-se o PASET, obtém-se a rede de custo mínimo que interliga estes pontos fixos, fazendo uso, se necessário de pontos extras de conexão (pontos de Steiner). Na Figura 6 tem-se um exemplo de uma rede de dutos de ventilação, onde os círculos brancos são os pontos fixos e os círculos pretos são os pontos de Steiner.

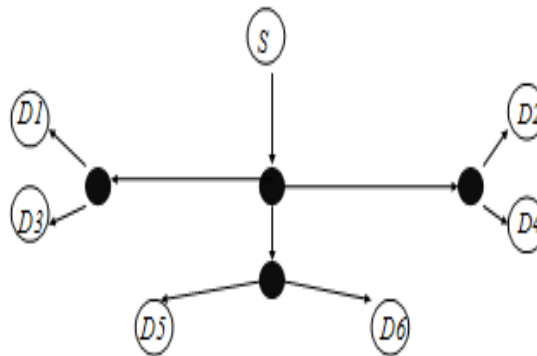


Figura 6 - Rede de dutos de ventilação (ROCHA 2007).

## 2.4. Algoritmo Genético

Segundo (RESENDE, 2003), o Algoritmo Genético é uma classe de procedimentos com um conjunto de passos distintos e bem especificados, no qual cada um destes passos possui muitas possíveis variações.

“Quanto melhor um indivíduo se adaptar ao seu meio ambiente, maior será sua chance de sobreviver e gerar descendentes.” (DARVIN, 1859).

## 2.5. Busca Local

“A Busca Local é um algoritmo utilizado para resolver problemas de otimização. Ela procura encontrar o seu melhor estado de acordo com uma função objetivo” (RUSSEL e NORVING, 2007).

A Figura 7 mostra um exemplo de aplicação da Busca Local em uma população. É considerada completa quando encontra um objetivo e ótima sempre quando encontra o mínimo/máximo global.

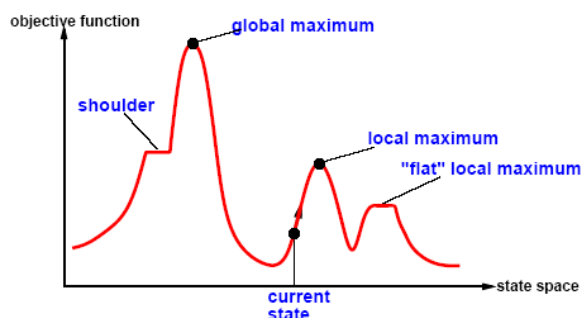


Figura 7 - Aplicação da Busca Local. Fonte (RUSSEL e NORVING, 2007).

### 3. Desenvolvimento

Para esta aplicação foi utilizado instancias, que tem como conteúdo as coordenadas tridimensional, X, Y e Z, para os pontos fixos utilizados, como o objetivo de, a partir destes pontos, começar a inserir novos pontos na Árvore de Steiner Euclidiana 3D na idéia de diminuir estas ligações existentes. Estas instâncias foram utilizadas para todos os testes que serão mostrados no tópico de Resultados.

Foi implementado um Algoritmo Genético com todas as devidas adaptações para que fosse possível diminuir o custo das ligações entre os pontos fixos de uma árvore, fazendo evoluções no intuito de se chegar a bons resultados.

Para seguir as normas para construção da Árvore de Steiner Euclidiana, para cada posição (i) de um cromossomo poderia ter qualquer valor inteiro  $A_i$  sendo:  $1 \leq A_i \leq 2 \cdot i + 1$ . Esta formula é que determina a quantidade de pontos de Steiner que cada intervalo da árvore pode ter.

De cada topologia foi calculado o *Fitness*  $F(i)$ ; a *Aptidão*  $A(i)$  e também a *Aptidão Final*  $A(i) \text{ Final}$ .

Para a *Aptidão Final*, é colocado os  $M$  indivíduos da população em ordem decrescente por  $A(i)$ , onde  $M$  é o número de indivíduo da população. O calculo para cada indivíduo  $i$  é:

$$A(i) \text{ Final} = \min + \frac{(\max - \min) \cdot (i - 1)}{(M - 1)}$$

Neste trabalho a técnica de Busca Local foi aplicada após a finalização do AG. Este método trabalha com a melhor solução obtida pelo AG com o intuito de, se possível, melhorá-la ainda mais.

O problema abordado neste trabalho tem como característica possuir uma vizinhança muito grande. Neste caso, a utilização do processo de busca local com exploração de toda a vizinhança, exige um grande esforço computacional. De modo a amenizar este problema, adotou-se a estratégia a seguir. A cada passo do loop principal, sortear uma porcentagem de vizinhos a serem visitados para procurar melhorar o valor encontrado, o número mínimo de vizinhos teria que ser igual a 1. Continuando o processo, é sorteado outro valor que se enquadrasse nas regras do tamanho de pontos de



Steiner para a árvore, para o campo do vizinho no intuito de encontrar um valor diferente do atual e não igual a zero.

Após, eram feitos os calculados com o mesmo e se o ele se tornasse melhor que o atual era feita a troca, caso contrário era dado continuidade até encontrar um melhor ou terminar as operações com os vizinhos.

#### 4. Testes e Resultados Computacionais

Os resultados obtidos se deram após a execução do código do AG para o PASE 3D em um computador com o processador Pentium Core Duo, contendo 1GB de memória RAM DDR2 e sistema operacional Windows XP Professional.

Os menores tempos de execução gastos para encontrar bons resultados foram sem a aplicação da Busca Local, mas os melhores resultados encontrados foram com a Busca Local, aumentando assim o tempo.

Foram executadas 180 instâncias divididas em 12 conjuntos, sendo cada conjunto com um número de pontos fixos diferentes, que são: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 250 e 500. Assim, cada conjunto contém 15 arquivos, cada um com dados (posições dos pontos fixos) diferentes. Sobre esses pontos fixos representados é que foram inseridos os pontos de Steiner.

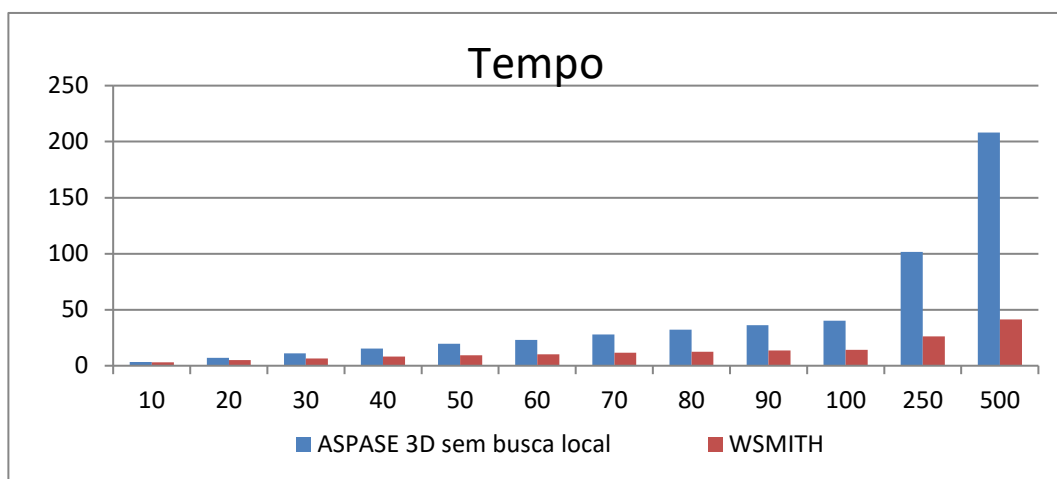


Gráfico 1 – Menores tempos gasto com o AGPASE 3D comparado ao WSMITH.

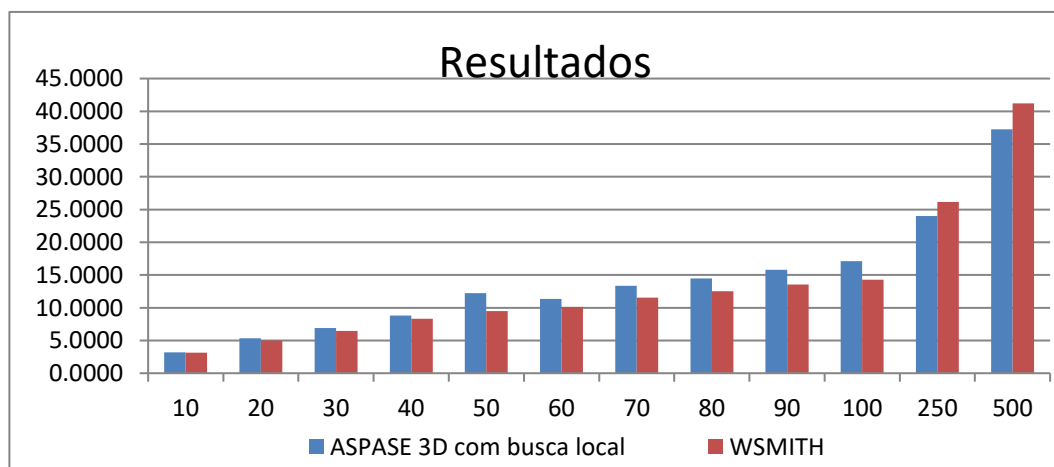


Gráfico 2 – Menores resultados obtidos com o AGPASE 3D comparado ao WSMITH.

Dos gráficos apresentados, é possível observar que o AG proposto consome um maior tempo de execução. Contudo, apresenta melhores resultados quando se apresentam instâncias de maior dimensionalidade (250 e 500 pontos) que são as de real interesse.

## 5. Conclusão

Pode-se observar que os resultados obtidos pelo AGPASE 3D com Busca Local ficaram bem próximos e equivalentes ao WSMITH, chegando até a superar-lo nas instancias maiores, embora o tempo gasto pelo AGPASE 3D, mesmo sem a Busca Local, ainda foi bem maior que o WSMITH para instâncias médias e grandes.

Com a aplicação da Busca Local, observou-se que o tempo aumentou razoavelmente, mas os resultados sofreram uma grande melhora após aplicar esta técnica, fazendo com que o AGPASE 3D superasse o WSMITH nas instancias maiores como 250 e 500.

É uma questão de trocar uma parte do tempo para obter um melhor resultado.

Dos testes computacionais realizados, pode-se verificar o potencial do AGPASE 3D com a adoção da técnica de Busca Local, para a resolução do Problema da Árvore de Steiner Euclidiana 3D.

## 6. Referencias

- BOAVENTU RA NETTO, P. O. Grafos: teoria, modelos, algoritmos. 2. ed. e ampl. São Paulo: Edgard Blücher, 2001.
- BROWER, J.E.; ZAR, J.H. Field & laboratory methods for general ecology. 2.ed. Dubuque: Wm. C. Brown Publishers, 1977. 226p.
- DARWIN, C (1859). On the Origin of Species by Means for Natural Selection. London: Jhon Murray.
- FAMPA, M. H. C. and ANSTREICHER, K. (2008) "An improved algorithm for computing Steiner minimal trees in Euclidean d-space", *Discrete Optimization* 5(2), pp. 530-540.
- GAREY, M. R. and JOHNSON, D. S. Computers and Intractability: A Guide to the Theory of NP-completeness. W. H. Freeman, San Francisco, 1979.
- GAUCH, H.G. Multivariate analysis in community ecology. Cambridge: Cambridge University Press, 1982. 298p.
- HERRING, Michael: The Euclidean Steiner Tree Problem. Denison University 2004, <[www.denison.edu/academics/departments/mathcs/herring.pdf](http://www.denison.edu/academics/departments/mathcs/herring.pdf)>.
- OLIVEIRA, N. V., *O Problema de Steiner e a Estrutura das Biomacromoléculas*. Tese de D.Sc., COPPE-UFRJ, Rio de Janeiro, Brasil, 2005.
- REZENDE, Solange Oliveira (Coord.). Sistemas inteligentes: fundamentos e aplicações. Barueri, SP: Manole, 2003.
- ROCHA, M. L.; BHAYA, A.; MONTENEGRO, F.; MACULAN, N.. A GRASP Heuristic for the Tridimensional Euclidean Steiner Tree Problem. In: Metaheuristic International Conference, 2007, Montreal. Seventh Metaheuristic International Conference, 2007.

- RUSSEL, S. e NORVING, P.; Inteligência Artificial. Junho, 2007.
- SMITH, W. D. (1992): How to Find Steiner Minimal Trees in Euclidean d-space. *Algorithmica*, v.7, n. 2/3, pp.137-177.
- SMELYANSKIY, M., SKEDZIELEWSKI, S., DULONG, C. “Parallel computing for large-scale optimization problems: Challenges and solutions”. *Intel Technology Journal*, 9(2):151-163, May 2005. ISSN 1535-766X.
- SMITH, J.M. “Steiner Minimal Tree in  $E^3$ : Theory, Algorithms and Applications”. In: *Handbook of Combinatorial Optimization*, v.2, pp. 397-470. Kluwer Academic Publishers, 1998.
- ZACHARIASEN, M. Algorithms for Plane Steiner Tree Problems. *Ph.D. Thesis*, Department of Computer Science, University of Copenhagen, 1998.



# **Proposta de um Jogo Educativo para Simular o Corpo Humano baseado em Agentes Emocionais Hedonistas**

**Deise Miranda Borges, Douglas Neves de Jesus, Luane Gomes Cunha,  
Naara Soares Negre, Fernando Luiz de Oliveira**

Curso de Sistemas de Informação – Centro Universitário Luterano do Brasil  
(CEULP/ULBRA)

Caixa Postal 160 – 77.054-970 – Palmas – TO – Brasil.

{infodeise, douglasnj01, luaneg, naarasoares, nandoluiz.br}@gmail.com

***Resumo.** Este artigo tem como objetivo demonstrar as etapas para o desenvolvimento de um jogo computacional educativo utilizando como técnicas de Inteligência Artificial os agentes inteligentes, mais especificamente agentes emocionais hedonistas em comportamento autônomo. Com a utilização dessa técnica procura-se quebrar o grande desafio da área de jogos educativos, que seria trazer mais realidade e entretenimento para um jogo computacional escolar. Inicialmente serão apresentados os conceitos de agentes inteligentes, agentes autônomos e agentes emocionais hedonistas. Procura-se também destacar o papel do agente emocional hedonista no jogo proposto, demonstrando que diferencial o agente trará para o jogo.*

**Palavras chave:** Agentes Emocionais, Jogos Educativos, Simulador Humano.

## **1. Introdução**

Com a inserção de computadores no ambiente escolar os métodos didáticos tradicionais estão sofrendo algumas mudanças e, por isso, tenta-se buscar novas metodologias educacionais para atingir uma maior integração e flexibilidade do ensino, além de tornar o conteúdo mais atraente para os alunos estudarem e se interessarem. Dentro deste contexto, os jogos computacionais se tornaram uma arma poderosa para o ambiente escolar, onde o jogo tornou-se mais do que uma forma de lazer e entretenimento, sendo utilizado também para propagar conhecimento. Assim, os jogos educativos estão sendo introduzidos como uma ferramenta auxiliar no ambiente de aprendizagem, no qual apresentam um mundo imaginário para ser explorado pelo aluno. Além de ensinar, um jogo educativo tem como objetivo aumentar o interesse e a motivação do aluno pelos estudos através da integração do material didático, multimídia e computadores.

Segundo Jenson & Castel [2002], o maior desafio dos jogos com propósitos educacionais é oferecer para o aprendiz um ambiente que propicie a imersão onde os usuários queiram estar, explorar e aprender da mesma forma que os aprendizes fazem nos jogos computadorizados comerciais. Desta forma, o objetivo deste artigo é demonstrar as etapas para o desenvolvimento de um jogo computacional educativo baseado em técnicas de Inteligência Artificial (IA), utilizando agentes inteligentes emocionais hedonistas. Este jogo será utilizado como uma ferramenta de auxílio ao professor, sendo uma forma diferente de ministrar aulas na área da biologia.

Nas próximas seções serão apresentados os conceitos necessários para o desenvolvimento do trabalho, assim como a modelagem e a proposta do jogo.

## 2. Agentes Inteligentes

Um agente inteligente, segundo Wood [1994], é um sistema computacional que está situado em algum ambiente, e que é capaz de ações autônomas neste ambiente visando atingir seus objetivos propostos. Com base neste conceito Russel & Norvig [1995] definem que agentes inteligentes são qualquer entidade que perceba um ambiente através de sensores, executem tarefas de processamento de informações e de conhecimento, e atue sobre o ambiente através de atuadores.

Para um agente ser considerado inteligente ele deve ser flexível, que segundo Jennings [1988] flexibilidade envolve:

- **Autonomia:** modo a agir sem qualquer tipo de intervenção, possuindo controle sobre suas ações e estado interno;
- **Reativo:** Perceber seu ambiente e responder de forma oportuna as mudanças que nele ocorrem.
- **Proativo:** Não devem apenas responder as mudanças do ambiente, devem ser oportunistas, ter iniciativa em prol de seus objetivos.
- **Social:** Capacidade de interagir quando necessários com outros agentes, sejam eles artificiais ou humanos, a fim de completar suas soluções ou ajudar a completar as soluções de outros agentes.

De uma forma mais simples, um agente inteligente deveria tomar qualquer ação possível para cada possível sequência de percepção, onde ele baseia-se nas evidências fornecidas pela sequência de percepção e nos conhecimentos prévios que ele tenha adquirido através da experiência. A seguir será explicado sobre dois tipos de agentes inteligentes que serão utilizados para o desenvolvimento do jogo.

### 2.1. Agentes Autônomos

Um Agente Inteligente Autônomo é o que consegue se operar com completa autonomia, decidir por si só como relacionar os dados obtidos com ações de modo que seus objetivos sejam atingidos com sucesso [MAES, 1995]. As principais características de um agente autônomo segundo Russel & Norving [1995] são: Comportamento determinado pela experiência, Conhecimento Inicial e Habilidade de aprender.

Pode-se dizer que um agente é autônomo quando ele toma decisões baseado na sua experiência, onde o agente começa com um conhecimento inicial e o vai aumentando de acordo com o que ele vai passando utilizando desta forma a sua habilidade de aprender.

Agentes autônomos em jogos de computadores são de extrema importância para trazer mais realidade ao jogo, pois é o sistema de controle inteligente que deve decidir as ações do agente para poder simular com realismo o comportamento de um oponente humano. Segundo Bradshaw [1997]; Murch & Johnson [1999]; Franklin & Graesser [1996]; Wooldridge & Jennings [1995], um agente autônomo é capaz de agir e sentir de acordo com seus propósitos em um ambiente, possuindo autonomia, pró-atividade, reatividade, aprendizado, ação propositada e adaptabilidade.

Um dos problemas recorrentes dos agentes autônomos inteligentes é a implantação em ambientes complexos, como os jogos de computadores, no qual os

agentes devem agir de maneira que simule o máximo possível a realidade tentando aproximar-se de um comportamento humano. Para tentar atingir o mais próximo do comportamento humano este artigo usará metáforas emocionais para a implementação dos agentes.

## **2.2. Agentes Emocionais Hedonistas**

O hedonismo é uma teoria ou doutrina filosófico-moral que coloca como maior valor, o prazer e a satisfação individual em primeiro lugar [GOMES, 2007]. “Um agente que possua uma vida afetiva é um agente com emoções rudimentares e ações induzidas por emoções. Adicionalmente os agentes são capazes de raciocinar sobre acontecimentos emocionais que ocorreram em outras interações”, [Elliot, 1992]. Assim a decisão que o agente irá tomar será baseada no prazer e na dor que ele poderá ter, utilizando o conceito do hedonismo. O comportamento do agente deve visar diminuir a dor presente e minimizar a expectativa de dor futura, ou seja, buscando sempre o prazer atual e futuro.

Para a implementação de um agente autônomo utilizando um agente emocional hedonista, deve-se inicialmente definir quais emoções estarão presentes e como essas emoções serão calculadas. Além disso, devem ser definidas prioridades, para que caso ocorra conflito de emoções o agente saiba qual é a emoção que tem maior prioridade. Utilizando este conceito de agentes emocionais hedonistas, será feita a implementação do agente autônomo que navegará no jogo educativo. As metáforas emocionais que serão utilizadas são: medo, fome e curiosidade.

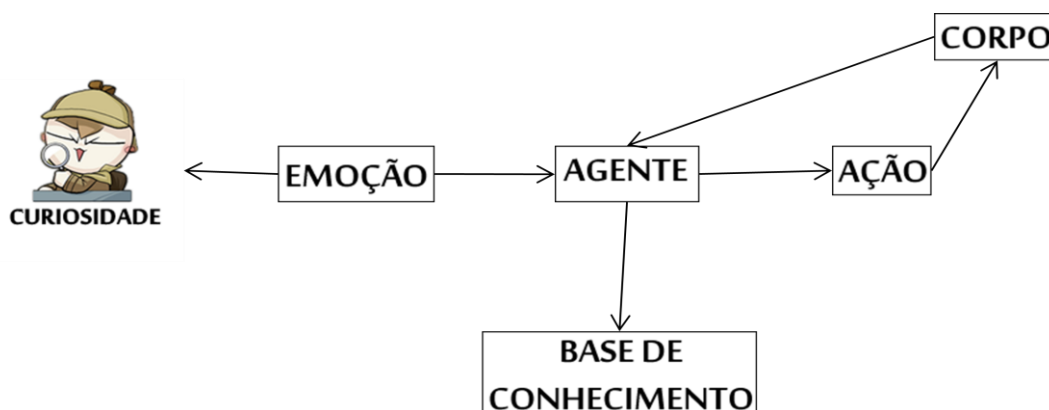
## **3. Jogo Educacional: Simulador Humano**

O Simulador Humano tem como proposta ser um jogo *multiplayer* e *single player* que poderá ser jogado em rede. O jogo será dividido por times, no qual serão chamados, temporariamente, de ‘time da doença’ e ‘time da saúde’. O ‘time da doença’ será o responsável por causar doenças, inserir vírus, e qualquer ação que danifique o corpo humano, o seu objetivo é fazer com que o corpo fique debilitado até que chegue a morte. Já o ‘time da saúde’ será responsável por alimentar o corpo humano além de ter que combater qualquer corpo estranho que for inserido, o objetivo deste time é fazer com que o corpo fique em perfeito estado de saúde e consequentemente derrote todas as ameaças. Cada time iniciará o jogo com uma quantidade X de recursos, que irão variar de acordo com o mapa escolhido. Esses recursos serão utilizados para poder realizar ações como, por exemplo, inserir um vírus no corpo humano. Ao ganhar um “round” o time vencedor adquire mais recursos que o outro time, podendo gastar esses recursos com mais ações.

O Simulador Humano irá possuir quatro mapas que representarão as fases evolutivas do corpo humano, que serão: criança, jovem, adulto e idoso. Desta forma cada mapa terá o seu nível de dificuldade, onde dependendo do mapa pode favorecer a um dos times. No mapa jovem, o ‘time da saúde’ irá iniciar com mais recursos que o outro time, pois um corpo novo é bastante saudável e possui, naturalmente, um sistema imunológico melhor. Já o idoso e a criança são mais vulneráveis a doenças e possuem uma saúde um pouco mais frágil, favorecendo assim o time da doença. O mapa adulto é o único que favorece igualmente ambos os times, pois está em um estado neutro do ciclo de vida humano.

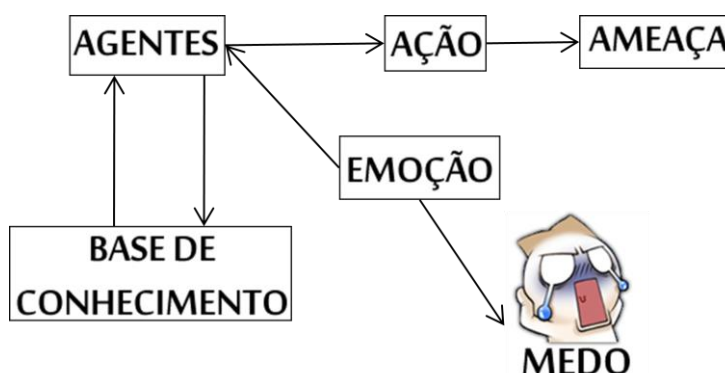
Ao iniciar o jogo, o ‘time da doença’ e/ou ‘da saúde’, dependendo do mapa escolhido pelos jogadores, terá alguns recursos habilitados e outros desabilitados. O corpo humano estará saudável e a partir das ações realizadas pelos times o corpo continua saudável ou vai se debilitando. Antes de inserir alguma ameaça os jogadores do time da doença devem fazer uma breve pesquisa pra saber qual seria o melhor vírus ou doença a ser inserida para atingir o órgão que ele queira, ou até mesmo o corpo todo. Da mesma forma, o ‘time da saúde’ também deverá fazer uma pesquisa para entender a doença e descobrir qual remédio, vitaminas ou alimentos que ele deve inserir no corpo para que ele possa ficar saudável e tenha forças para combater as ameaças.

Nas figuras 1 e 2 é representado o fator curiosidade e o fator medo trabalhando para combater uma ameaça ao corpo humano.



**Figura 1:** Curiosidade

Na figura 1 acima é representada a curiosidade de um agente. Primeiramente, é escolhida qual emoção o agente irá priorizar, logo após o agente irá realizar uma ação como, por exemplo, verificar se o corpo está saudável ou não. Nesta fase, o corpo disponibiliza seu estado atual e o agente atualiza a base de conhecimento com as informações obtidas.



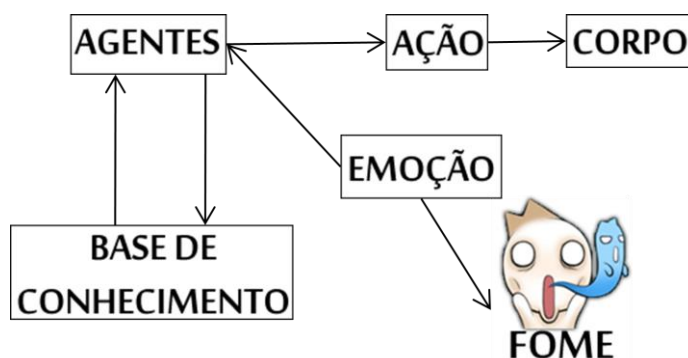
**Figura 2:** Medo

A figura 2 demonstra o Fator Medo. Os agentes consultam a base de conhecimento buscando informações do estado do corpo. A base de conhecimento



retorna que foram encontradas ameaças que podem danificar o corpo. Por fim, o Agente prioriza a emoção Medo e realiza uma ação para combater a ameaça.

Após expulsar todas as ameaças, o corpo sente-se fraco, necessitando de alimentos, nesse momento entra a fome, que pode ser combatida por nutrientes já armazenados no corpo ou ainda inseridos pelo usuário.



**Figura 3:** Fome

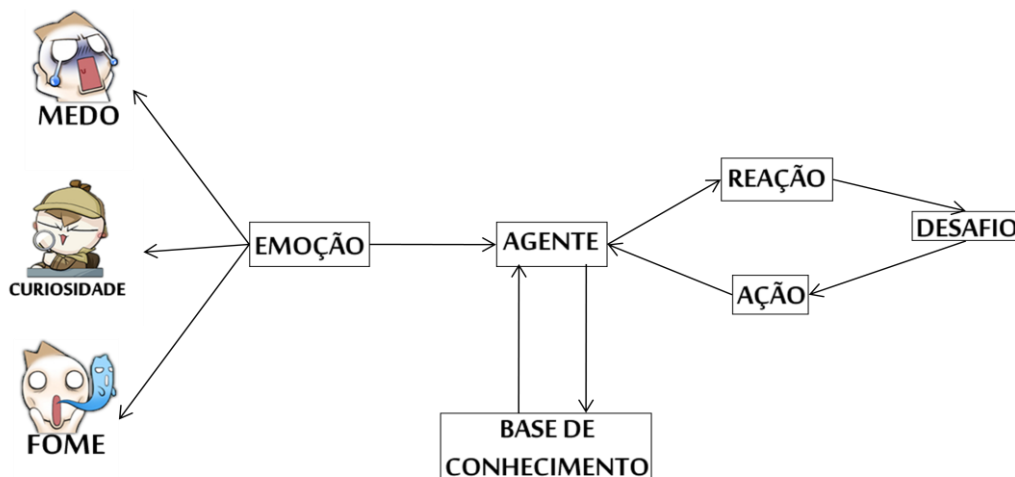
A figura 3 acima demonstra o Fator Fome. Assim como o Fator Medo, os agentes consultam a base de conhecimento em busca de informações do estado do corpo. A base de conhecimento retorna que o corpo está debilitado e o Agente prioriza a emoção Fome. A seguir será apresentada a modelagem do Agente Emocional Hedonista.

#### **4. Modelagem do Agente**

O objetivo do jogo é simular o funcionamento do corpo humano o mais próximo da realidade. Neste jogo o usuário poderá simular diversas ações sobre o corpo humano, tais como: inserir vírus ou combatê-los, alimentar o corpo humano, entre outros. De acordo com as ações do jogador, o corpo humano, que possuirá os agentes autônomos, terá reações de acordo com a decisão tomada pelo agente emocional, que será baseada nas metáforas emocionais. As codificações das emoções serão da seguinte forma:

- Medo: avalia o quanto uma situação pode significar risco para a integridade do agente. De acordo com o medo sentido, o agente deve priorizar uma ação.
- Fome: avalia as condições energéticas. Se o agente estiver com fome ele procura por alimentos.
- Curiosidade: ação para aumentar o conhecimento do agente sobre o ambiente e procurar ameaças.

O agente autônomo também agirá de modo informativo, ou seja, de acordo com o ambiente ele dará dicas ao usuário sobre a situação atual do corpo, por exemplo, se o nível de açúcar no sangue estiver alto, ele pode informar ao usuário que excesso de açúcar pode indicar diabetes.



**Figura 4:** Visão geral de um agente autônomo utilizando a emoção para a tomada de decisão.

Na figura acima, um desafio surge no corpo (como um vírus, por exemplo) e o agente deve tomar uma determinada ação. Por exemplo, o agente consulta a base de conhecimento para verificar a situação atual do corpo; a partir do estado físico obtido pela base, o agente irá verificar qual emoção que lhe proporcionará mais prazer, gerando uma reação.

As emoções terão um nível de prioridade, que serão codificadas inicialmente no agente, caso ocorra um conflito de emoções, o agente poderá tomar a decisão baseado na emoção prioritária. Desta forma, as emoções serão classificadas em prioridade como:

1. Fome: o agente priorizará a energia do corpo humano, dando alerta para o jogador que o corpo humano precisa se alimentar corretamente.
2. Curiosidade: o agente deve sempre estar buscando conhecer cada vez mais o ambiente e assim aumentar o seu conhecimento sobre o mesmo, além de procurar por ameaças.
3. Medo: o agente deve evitar que ocorram eventos indesejáveis ao corpo humano.

Portanto, para que o agente autônomo aja de forma independente e de forma mais eficaz, ele será implementado utilizando o conceito de agentes emocionais hedonistas, no qual o sistema inteligente do agente autônomo fará a tomada de decisões baseado nas emoções.

Para a implementação dos agentes autônomos pretende-se utilizar a *toolkit* chamada DirectIA, que permite desenvolver agentes que aprendem através do contato com o ambiente e com as ações do usuário. Desta forma, tenta-se facilitar o desenvolvimento dos agentes, para que se possa ter um melhor resultado e rapidez na implementação do agente autônomo.

## 5. Considerações Finais

Foi apresentado neste artigo o conceito do agente emocional hedonista que, baseado em metáforas computacionais para emoção, procura gerar um comportamento mais incrementado dos agentes autônomos. Demonstrou-se também os agentes autônomos

que é a criatura artificial que dá autonomia ao sistema, as duas técnicas juntas formam as propostas para resolver o problema inicial, como tratar a autonomia de um agente em um ambiente complexo como os jogos para computadores e fazer com que os jogos educativos tragam mais realidade e entretenimento para seus usuários.

O estudo realizado até o momento permitiu visualizar a futura possibilidade de implementação de um jogo educacional que simule o corpo humano, através da utilização de agentes inteligentes, que podem interagir com o ambiente. Outro motivo pela escolha foi o fato de possuírem autonomia, o que possibilita maior realidade ao jogo, já que este tenta simular o corpo humano. Inicialmente, utilizou-se três metáforas emocionais: medo, fome e curiosidade, mas a idéia é codificar cada possível objetivo do agente em emoções diferentes, tentando implantar no agente quase todas as emoções sentidas pelo ser humano.

Por fim, a implementação do jogo possibilitará um maior aproveitamento em sala de aula, devido a sua grande interação com os estudantes trazendo também abordagem de conceitos referentes à saúde e de como o corpo humano reage as ameaças.

## 6. Referências Bibliográficas

GOMES, A. R. . **Vocabulário de Filosofia**. Disponível em: <<http://ocanto.esenviseu.net/lexh.htm>>. Acesso em: 18 de outubro de 2009.

JENSON, JENNIFER; CASTEL, SUZANNE. **Serious Play: Challenges of Educational Game Design** In: American Research Association Annual Meeting in New Orleans. Louisiana: AERA, 2002. Disponível em: <<http://edtech.connect.msu.edu/Searchaera2002/viewproposaltext.asp?propID=5573>>. Acesso em: 18 de outubro de 2009.

Jennings, N.; Sycara, K.; and Wooldridge, M. 1998. **A Roadmap for Agent Research and Development**. Disponível em: <<http://www.springerlink.com/content/p321417jr14k2123/>>. Acesso em: 18 de outubro de 2009.

Maes, P.; **Modeling Adaptive Autonomous Agents**. In C. Langton (Ed.) *Artificial Life: an Overview*. Cambridge: MIT Press, 1995. Disponível em: <<https://eprints.kfupm.edu.sa/51317/1/51317.pdf>>. Acesso em: 18 de outubro de 2009.

RUSSELL, Stuart J.; NORVIG, Peter.; **Artificial Intelligence: a Modern Approach**. New Jersey: Prentice Hall, 1995. 932 p.

WOOD, A.; **Towards a Medium for Agent-Based Interaction**. School of Computer Science, The University of Birmingham, 1994.

WOOLDRIDGE, M.; JENNINGS, N.R.; **Intelligent agents: Theory and practice**. The Knowledge Engineering Review, 1995.

Elliot, C.; **The Affective Reasoner: A Process Model of Emotions in a Multi-agent System**. Disponível em: <<http://condor.depaul.edu/~elliott/ar/ftp/elliott-thesis.pdf>>. Acessado em: 18 de outubro de 2009.



# Uma Abordagem No Uso De Rede De Energia Elétrica Para Transferência De Dados

Timote Lima da Silva<sup>1</sup>

<sup>1</sup>Ciência da Computação – Fundação Unirg

Av. Antônio Neves Silva S/N, Pq. das Acácias, CEP: 77400-000 – Gurupi – TO – Brasil

tim\_lima@yahoo.com.br

**Resumo.** *Este artigo aborda uso da rede de energia elétrica tecnologia PLC (Power Line Communication) para transferência de dados. A rede elétrica já possui infra-estrutura instalada na maioria das residências brasileiras tornando instalação simples e barata para usuários residenciais*

*Objetivo é estudar e mostrar como a rede elétrica pode ser útil e disponibilizar ferramentas com tecnologia do sistema domotico (integração automatização residencial) que trarão mais comodidade e acessibilidade envolvendo equipamentos eletrônicos, como computadores, câmeras vigilância e TV digital.*

## PALAVRAS-CHAVE

PLC, Sistema domótico, rede domiciliar, redes com fio, redes sem fio.

## 1. Introdução

Estudo sobre a tecnologia Power Line Communication (PLC), que permite transmissões de sinais por onda portadora em redes de distribuição de energia, atuando como opção de conectividade para transferência de dados, além dos sistemas com fios cabo UTP, sem fio IEEE 802.11 wireless, cabos coaxiais das operadoras de TV por assinatura, integrando na prática da domotica.

Este trabalho visa difundir este novo conceito de transmissão de dados em lugares onde não há opções economicamente e fisicamente viáveis para usuários de internet.

Para conseguir objetivo proposto será feito estudo sobre projetos feitos, mostrando exemplos de Países onde tecnologia é usada com sucesso, e estimativa de uso.

Uso da tecnologia PLC foi escolhido devido a ser uma tecnologia com abrangência favorável em qualquer lugar onde haja energia elétrica e possibilidade de instalação de equipamentos e infra-estrutura favorável.

Segundo a APTEL (Associação de Empresas Proprietárias de Infraestrutura e de Sistemas Privados de Telecomunicações), o investimento na tecnologia PLC é inevitável em função do contexto das redes inteligentes (smart grid) e o que pode ocorrer é uma antecipação deste investimento por parte das empresas de energia. “A evolução tecnológica das redes terá impacto na melhoria da oferta de energia e apenas o excedente da capacidade será usado para a comercialização de outros serviços, como banda larga, por exemplo.

Mas para isso viabilizar-se, as empresas precisam vislumbrar um retorno, como a possibilidade de vender o excedente para empresas de telecomunicações. Para isso, até admite-se repassar uma parte deste ganho para a redução tarifária de energia elétrica”, afirma o presidente da entidade Pedro Jatobá. “O que não pode ocorrer é as distribuidoras de energia elétrica serem meras fornecedoras de infraestrutura”, completa ele.[1]

Este artigo tem como objetivo principal a realização de um estudo sobre as diversas tecnologias de redes para uso residencial, levando-se em consideração suas vantagens, desvantagens e tendências. Os resultados deste estudo demonstram a necessidade de integração das diferentes tecnologias de rede, formando redes híbridas.

## **2.Comparação Entre Tecnologias - Tipos De Redes**

As redes domiciliares abordadas podem ser classificadas em:

### **2.1) Redes Com Fio (representada pela Ethernet);**

É uma rede de difusão de barramento com controle descentralizado, em geral operando em velocidades de 10 Mbps a 10 Gbps. Os computadores em uma rede Ethernet podem transmitir sempre que desejam; se dois ou mais pacotes colidirem, cada computador aguardará um tempo aleatório e fará uma nova tentativa mais tarde. Meio mais usado para transferência de dados atualmente[2].

### **2.2) Redes Sem Fio (e.g. Bluetooth, IEEE 802.11);**

Rede de computadores sem a necessidade do uso de cabos – sejam eles telefônicos, coaxiais ou ópticos – por meio de equipamentos que usam radiofrequência (comunicação via ondas de rádio) ou comunicação via infravermelho.

Embora as redes sem fios e a computação móvel frequentemente tenham uma estreita relação, elas não são idênticas. Algumas vezes, até mesmo os computadores portáteis podem estar conectados por fios. Por exemplo, se um viajante conecta um notebook à tomada de telefone em um quarto de hotel, ele tem mobilidade sem precisar utilizar uma rede sem fio.[2]

A Tabela 1 a seguir mostra um resumo dos padrões para WLANs IEEE 802.11.

Padrão	Frequência	Taxa máxima da camada física	Método de transmissão	Área de cobertura (indoor)
802.11	2.4GHz	2Mbps	FHSS/DSSS	
802.11a	5GHz	54Mbps	OFDM	~ 30m
802.11b	2.4GHz	11Mbps	DSSS	~50m
802.11g	2.4GHz	54Mbps	OFDM	~30m

**Tabela 1: Comparativo entre os padrões de WLAN IEEE 802.11**

### **2.3) Outros Tipos de Redes (e.g. PLC - HomePlug),**

#### **2.3.1) PLC (Power Line Communications)**

É a tecnologia que utiliza uma das redes mais utilizadas em todo o mundo: a rede de energia elétrica. A idéia desta tecnologia não é nova. Ela consiste em transmitir dados e voz em banda larga pela rede de energia elétrica. Como utiliza uma infra-estrutura já disponível, não necessita de obras em uma edificação para ser implantada.

A PLC trabalha na camada 2 do modelo ISO/OSI, ou seja, na camada de enlace. Sendo assim, pode ser agregada a uma rede TCP/IP (camada 3) já existente, além de poder trabalhar em conjunto com outras tecnologias de camada 2.[3]

Alguns protocolos como o HomePlug 1.0 utilizam criptografia DES de 56 bits. Os dados estão sempre em rede local porque esta tecnologia não ultrapassa a caixa elétrica da casa. Contém de fato muito mais segurança do que o Wi-Fi, que pode ser visível pelos vizinhos e que necessita uma identificação por usuário e senha.

#### **2.3.2) HomePlug 1.0**

O padrão HomePlug 1.0 foi desenvolvido em 2001, pela HomePlug PowerLine Alliance e utiliza algumas técnicas para superar a hostilidade do ambiente PLC.

A comunicação de dados confiáveis sobre o meio hostil PLC requer técnicas eficazes de FEC (Forward Error Correction), entrelaçamento, detecção de erro e ARQ (Automatic Repeat Request), bem como, um protocolo de controle de acesso ao meio (MAC) robusto.

A técnica de multiplexação OFDM (Orthogonal Frequency Division Multiplexing) do HomePlug 1.0 foi especialmente projetada para ambientes de linhas elétricas. Ela utiliza 84 subportadoras igualmente espaçadas na banda de frequência entre 4,5 e 21 MHz. Considerando todos os possíveis parâmetros de transmissão, a camada física do HomePlug pode oferecer taxas de transmissão que variam de 1 a 14 Mbps.

O OFDM apresenta excelente resultado com relação a efeitos de dispersão do sinal ao longo do tempo, provendo baixa interferência entre canais (ICI - Interchannel Interference). A funcionalidade de Prefixo Cíclico<sup>1</sup>, proporcionada pelo OFDM, em conjunto com as técnicas diferenciais de modulação DBPSK

(Differential Binary Phase Shift Keying) e DQPSK (Differential Binary Phase Shift Keying) é utilizada para eliminar, por completo, a necessidade de equalização 2.

Problemas com ruído impulsivo 3 são solucionados através da utilização de FEC e entrelaçamento de dados. A carga (payload) a ser transmitida através da tecnologia de acesso ao meio HomePlug 1.0 usa as técnicas Viterbi e Reed-Solomon FEC [4].

## **2.4) Redes Mistas (e.g.LONWorks)**

Diferentemente de alguns dos padrões citados anteriormente, as redes mistas que serão apresentadas neste trabalho, já foram desenvolvidas especificamente com o intuito de serem utilizadas em redes domiciliares.

As redes mistas se caracterizam por apresentar em sua própria arquitetura, opções de comunicação a partir de diferentes meios de transmissão, como par trançado, fibra óptica, rádio-freqüência e outros. Dentre os padrões mais conhecidos, destacam-se LONWorks.

### **2.4.1 LONWorks**

A plataforma LONWorks foi desenvolvida pela Echelon Corporation, em 1988, para dispositivos de rede, que operam sob diversos meios de comunicação [5]. É mais popularmente utilizado na automação de determinadas funções de uma residência, como iluminação e HVAC (Heating, Ventilating and Airconditioning), também referenciado como controle climático.

A pilha LONWorks implementa o modelo completo de sete camadas OSI e consiste em um protocolo de comunicação, denominado LonTalk e formalizado como ANSI/EIA-709, um controlador dedicado(Neuron Chip) e uma ferramenta de gerenciamento de rede [6].

O EIA-709 suporta uma variedade de diferentes meios de comunicações e de diferentes topologias para redes cabeadas. Para dar suporte à interoperabilidade, um número de perfis de canais de comunicação é definido: existem canais para par trançado (twisted pair - TP), powerline (PL) e fibra óptica (fiber optic -FO).

Para o par trançado é utilizado um mecanismo de acesso ao meio singular, denominado predictive ppersistent CSMA. Neste mecanismo, quando um serviço multicast é utilizado, uma predição sobre a carga futura da rede pode ser feita. O tempo da fase de decisão é modificado conforme for necessário. Dessa forma, o aumento da taxa de colisão com o aumento da carga é suavizado.

Isto ajuda a assegurar uma taxa de pacote mínima aceitável, mesmo quando existe uma grande quantidade de carga.

## **3. Sistemas Domóticos**

Fora do contexto de sistemas domóticos, maioria dos serviços é planejada para operar de forma autônoma sem interagir com outros serviços. Tipicamente, a única conexão externa de um serviço é apenas a fiação elétrica. As redes domiciliares permitem que



esses serviços (e.g. gerenciamento de energia, controle de temperatura, segurança) sejam integrados.

Um sistema domótico (Domotic System) ou domiciliar (Home System) engloba uma rede de comunicação pertencente a uma casa com uma possível conexão à redes externas a mesma (e.g. Internet). A rede domiciliar pode gerenciar serviços, sensores, atuadores, comutadores, controladores e interfaces com o usuário para criar um novo serviço. A rede domiciliar também pode prover uma infra-estrutura para a distribuição de dados multimídia [7].

Contudo, uma remodelagem sobre os serviços é necessária para que as empresas possam explorar os produtos em potencial para sistemas domiciliares e possam fornecer serviços significantes para os usuários finais. A maior mudança é a criação de serviços que combinem múltiplos serviços, sensores e painel de controle de usuário para conseguir produtos que operem de maneira transparente. Transparência significa que o usuário se beneficia dos novos serviços, sem perceber a complexidade interna necessária para implementar os mesmos.

## 4. Vantagens, Desvantagens no Uso PLC

### 4.1 Vantagens

Uma das grandes vantagens do uso da PLC é que, por utilizar a rede de energia elétrica, qualquer "ponto de energia" pode se tornar um *ponto de rede*, ou seja, só é preciso plugar o equipamento de conectividade (que normalmente é um modem) na tomada, e pode-se utilizar a rede de dados.

Pode – se citar seguintes funcionalidades:

1. Tomada elétrica única de alimentação elétrica, voz e dados
2. Sem necessidade de obras nem cabos adicionais, instalação rápida e simples, sem obras
3. Equipamento de ligação modem PLC “ plug & play “, auto instalável
4. Ligação permanente de dados (ativa 24 horas dia)
5. Permite continuar o fornecimento de eletricidade sem nenhum tipo de problemas
6. Transmissão de voz IP
7. Domótica (gestão de todos os recursos habitacionais) e automatização integral
8. Tele-vigilância e Segurança
9. Televisão Interactiva

### 4.2 Desvantagem

Uma das grandes desvantagens do uso da PLC (ou **BPL**), é que qualquer "ponto de energia" pode se tornar um ponto de interferência, ou seja, todos os outros equipamentos que utilizam radiofrequência, como receptores de rádio, telefones sem

fio, alguns tipos de interfone e, dependendo da situação, até televisores, podem sofrer interferência. A tecnologia usa a faixa de frequências de 1,7MHz a 30MHz, com espalhamento de harmônicos até frequências mais altas. Outra desvantagem é o facto de ser *half-duplex* sem esquecer que é um sistema de banda partilhada. Estas duas características fazem com que o débito seja reduzido em comparação com outras tecnologias. Em alguns países, existem movimentos contra a sua instalação.[8]

## 5. Interferência do PLC na rede elétrica

A PLC não interfere em nenhum eletrodoméstico, pois as frequências utilizadas por ela não são usadas por nenhum eletrodoméstico, podendo conviver sem problemas com os outros equipamentos. No entanto, parte da banda de rádio de onda média - 1,7 a 3 MHz - e toda a onda curta - 3 a 30 MHz - ficam completamente prejudicadas e inutilizáveis. Outros equipamentos podem causar interferências em uma rede PLC, como motores de escova e os dimmers de luz. Entre os motores domésticos, destacam-se os secadores de cabelos, aspiradores e as furadeiras elétricas. Além desses, chuveiros elétricos também podem interferir no PLC.

Outro ponto importante da PLC é a conexão com equipamentos bloqueadores de frequência (filtros de linha) e os equipamentos isoladores (estabilizadores) ou que sejam alimentados por fontes chaveadas (*no-breaks*): os equipamentos PLC não podem ser ligados nestes.

No caso dos *no-breaks*, a saída da rede elétrica é isolada, e nos filtros de linha as altas frequências são bloqueadas, o que impossibilita o funcionamento da rede.

A interferência da PLC ocorre na faixa de 1,6 a 30 MHz, principalmente nos equipamentos de comunicação das forças armadas e no controle de tráfego aéreo. O fato é que estudos devem ser feitos para que exista uma recomendação para o uso da PLC em locais próximo a aeroportos, unidades militares, portos, etc. O uso irregular da PLC

## 6. Equipamentos

Os principais equipamentos presentes em redes PLC são:

Modem (PNT): Usado para a recepção e transmissão dos dados, o modem é instalado em um host (estação de trabalho, servidor, etc.) que é ligado à tomada de energia. Ele realiza a comunicação com o Demodulador Repetidor (PNR).

Demodulador Repetidor (PNR): Esse equipamento provê acesso direto do usuário do sistema InDoor para o sistema Outdoor. Cada residência tem um, e este se comunica com o Concentrador Mestre (PNU).

Concentrador Mestre (PNU): Controla o sistema Outdoor e interconecta uma Célula de Energia (Power Cell) à rede do backbone. Geralmente esta localizada no transformador. Deste ponto em diante a comunicação pode ser feita pela operadora de telecomunicações.

## 6.1 Multiplexação e Modulação

A tecnologia PLC utiliza a técnica de modulação de sinais OFDM (Orthogonal Frequency Division Multiplexing), pode utilizar as modulações por QAM (Quadrature Amplitude Modulation) e PSK (Phase Shift Modulation).[8]

## 7. Segurança

Toda comunicação do PLC é criptografada. Alguns protocolos como o HomePlug 1.0 utilizam criptografia DES de 56 bits. Os dados estão sempre em rede local porque esta tecnologia não ultrapassa a caixa elétrica da casa. Contém de fato muito mais segurança do que o Wi-Fi, que pode ser visível pelos vizinhos e que necessita uma identificação por usuário e senha.

## 8. Conclusão

A tecnologia PLC e redes mistas é uma área que está em ampla expansão. Diferentemente da automação industrial, a automação residencial é considerada de menor escala e possui ênfase na integração dos sistemas de entretenimento e gerenciamento de funções domiciliares.

Padrões que não necessitam de encomenda, ou seja, já estão disponíveis no mercado.

O meio de comunicação para integrar o sistema domótico é bastante diversificado, assim como a quantidade de aplicações que podem ser desenvolvidas para este cenário.

Aplicações de segurança, iluminação, gerenciamento de energia, sistema de áudio e vídeo, dentre outras, deverão ser capazes de funcionar em diversos meios de comunicação, como infravermelho, rádio, par trançado e PLC.

Devido esta grande diversidade, muitos padrões para redes com fio, sem fio, e mistas surgiram, cada um com vantagens e desvantagens singulares, com o intuito de atender as exigências dos sistemas domiciliares.

As redes com fio, representadas pela Ethernet, possuem a grande vantagem da estabilidade e da segurança, contrapondo com a idéia de que a instalação de novos fios em uma casa pode ser custosa para o usuário. Já as redes sem fio, representadas pelos padrões IEEE 802.11x, HomeRF, Bluetooth, têm sua grande vantagem na mobilidade fornecida simultaneamente com a conectividade na rede domiciliar.

Contudo, uma das principais desvantagens das redes sem fio é o fato de algumas tecnologias possuírem dispositivos que operam na mesma banda de frequência ISM, causando interferência na comunicação entre os mesmos.

Outros tipos de redes, representadas pelas redes PLC, HomePlug, apresentam uma inferioridade inerente ao meio de comunicação: a rede de distribuição elétrica ou telefônica não foi projetada com o propósito de comunicação e apresenta grande nível de atenuação e impedância. Por outro lado, diferentemente das redes com fio, PLC não necessitam de uma nova infra-estrutura para se implementada e isto faz com que essas redes também tenham um grande atrativo para o sistema domiciliar.

As redes mistas aparentam ser uma ótima solução no cenário das redes domiciliares, onde há diversos dispositivos interagindo entre si, sobre diferentes meios de comunicação. Contudo, além de serem capazes de se comunicar através de par trançado e rádio-freqüência, por exemplo, os dispositivos necessitam se comunicar e interagir com dispositivos de diferentes tecnologias, caracterizando o que se denomina de redes híbridas.

Dessa forma, as redes híbridas representam a estrutura plausível para aplicações em redes domiciliares, uma vez que conseguem agregar as vantagens das diversas tecnologias disponíveis e minimizar as desvantagens das mesmas. Assim, cada padronização pode contribuir com sua característica singular e oferecer mais e melhores opções para o usuário do sistema domótico controlar e executar suas aplicações na rede domiciliar.

## 9. Referencia

[1] Fator Brasil, Portal, “Aptel defende direito de concessionária de energia explorar tecnologia PLC”, [http://www.revistafator.com.br/ver\\_noticia.php?not=82348](http://www.revistafator.com.br/ver_noticia.php?not=82348), acesso em 10/09/09.

[2] Tanenbaum, Andrew S. “Redes de computadores 4º Ed.”

[3] Ed Hare, W1RFI “Calculated Impacto f PLC in Stations Operating in the amateur radio services” <http://www.arrl.org/tis/info/HTML/plc/files/C63NovPLC.pdf>, acesso em 14/10/09.

[4] LIN, Y (2003) A comparative performance study of wireless and power line networks.

In: IEEE Communications Magazine, Vol 41, Edição 4, Abril de 2003

[5] KASTNER *et al* (2005) **Communication Systems for Building Automation and Control**. In: Proceedings of The IEEE, Vol. 93, No. 6, Junho de 2005, pp. 1178-1203

[6] CHEMISHKIAN, S (2002) **Building smart services for smart home**

In: 2002 IEEE 4<sup>th</sup> International Workshop on Networked Appliances, pp. 215 - 224

[7] WACKS, K (2002) Home Systems Standards: Achievements and Challenges.

In: IEEE Communications Magazine, Volume 40, Edição 4, Abril de 2002, pp. 152-159.

[8] WIKIPÉDIA(2009) “PLC” <http://pt.wikipedia.org/wiki/PLC>, acesso em 14/10/09.

# **União Rede Social e Marketing Viral para ampliação da concorrência em processo seletivo (vestibular) de uma Instituição de Ensino Superior**

Heitor de Sousa Miranda, Parcilene Fernandes de Brito, Fabiano Fagundes  
Curso de Sistemas de Informação - Centro Universitário Luterano de Palmas  
(CEULP/ULBRA)  
Teotônio Segurado 1501 SUL - 77054-970 - Palmas - TO - Brasil  
heitordesousa@gmail.com, {pfb, fagundes}@ulbra-to.br

**Resumo.** *Este artigo tem como objetivo apresentar a união de Rede Social e Marketing Viral para ampliação da concorrência em processo seletivo (vestibular) de uma Instituição de Ensino Superior. Serão apresentados conceitos sobre Rede Social e Marketing Viral visando apresentar como os dois podem ser usados em um âmbito virtual. Também são apresentadas etapas da criação da Rede Social, tais como a sua estrutura, formas de acesso e a de divulgação para o público-alvo. Por fim, o funcionamento e os efeitos do marketing aplicado a rede são apresentados.*

**Palavras-chave:** Redes Sociais, Marketing Viral, Portal Acadêmico.

## **1 Introdução**

As redes sociais virtuais estão sendo exploradas cada vez mais, aumentando a quantidade de domínios nos quais podem ser aplicadas. As redes sociais pessoais, onde uma pessoa se cadastra e conhece novas pessoas, ou interage com pessoas já conhecidas, foram as primeiras a surgir, fazendo com que houvesse uma maior integração entre internautas do mundo inteiro.

Com a popularização da internet, a quantidade de internautas aumentou, logo, a internet começou a ganhar mais credibilidade, possibilitando que houvesse sistemas que tratassem de coisas sérias, ou seja, possibilitou a criação de novas formas de negócios, aumentando, assim, a produtividade e a abrangência de uma organização.

No momento em que essa confiança se torna realidade no contexto virtual, a interação entre pessoas, de forma profissional, se torna cada vez mais facilitada. Se existe um nível de confiança entre o sistema e as pessoas que fazem uso do mesmo, conseqüentemente vai existir confiança para essas pessoas manterem um relacionamento profissional.

As redes sociais possibilitam que exista um grande poder de influência entre as pessoas envolvidas devido ao contato direto das mesmas. Essa influência aumenta ou diminui através dos mecanismos presentes na rede social, pois, o uso desses mecanismos influi diretamente na reputação da pessoa que a usa. Esses mecanismos podem ser mensagens, grupos que a pessoa faz parte, interesses pessoais e/ou

profissionais etc. Obviamente, essas ferramentas não são as únicas que vão definir um nível de influência que uma pessoa tem em relação à outra.

Um aliado forte de uma rede social é o marketing viral. A troca de influência dentro de uma rede social é abundante, seja ela virtual ou real. Há quem diga que “somos produto do nosso ambiente” ou “uma laranja podre estraga o saco todo”, apesar de não ser tão simples assim. O marketing viral, como o nome já diz, visa agir como um vírus, onde os hospedeiros são os usuários da rede social. Um bom exemplo de uso dessa técnica é o serviço de e-mail Hotmail ([www.hotmail.com](http://www.hotmail.com)). Sempre que um usuário desse serviço envia algum e-mail, no rodapé da mensagem aparece alguma propaganda. No entanto, dessa forma não existe algum tipo de filtragem entre o usuário remetente e o destinatário.

A criação de uma rede social em um portal de uma universidade pode ajudar com aumento na procura das inscrições no vestibular, fazendo uso de marketing viral na rede. Como foi dito, uma rede social tem como objetivo integrar um grupo de pessoas que têm o mesmo interesse. Mas como a integração de alunos de uma universidade aumentaria o número de inscritos no vestibular? Neste artigo será explicado como seria feita a implantação de uma rede social no portal universitário e, também, como o marketing viral deveria agir para alcançar tal objetivo.

## **2 Rede Social**

Rede social pode ser definida como um conjunto de relações interpessoais concretas que vinculam indivíduos a outros indivíduos (BARNES, 1987). Ou seja, rede social é o nome que se atribui a um conjunto de pessoas que estão ligadas entre si devido a interesses em comum.

A utilização de redes sociais no âmbito virtual é uma das formas mais eficazes no que diz respeito à facilidade da comunicação. Não apenas isso como, também, o aumento da troca de informações entre pessoas. Essas informações podem ser referentes a qualquer contexto ou a contextos, depende do objetivo da rede.

Redes como *Orkut* ([www.orkut.com](http://www.orkut.com)), *MySpace* ([www.myspace.com](http://www.myspace.com)) e *Facebook* ([www.facebook.com](http://www.facebook.com)) prezam mais pela interação pessoal entre seus usuários, através de recados, comunidades, amigos etc. Porém, isso não significa que essas redes apenas sirvam para isso, até porque os usuários são livres para criarem seus perfis, podendo dar mais ênfase em dados pessoais e/ou profissionais.

Redes como *LinkedIn* ([www.linkedin.com](http://www.linkedin.com)) e *Ikwa* ([www.ikwa.com.br](http://www.ikwa.com.br)) são, digamos, “orientadas a contexto”, pois, são redes que prezam por interações profissionais entre seus usuários e que foram criadas com esse objetivo. Essas redes têm ligações com empresas e universidades visando uma integração entre pessoas que querem saber das oportunidades de empregos, fazer cursos etc. Vale ressaltar que essas redes “orientadas a contexto” não se limitam a apenas perfis profissionais. Um bom exemplo de rede social dessa categoria é a chamada *Last.fm* ([www.last.fm](http://www.last.fm)), onde seus usuários, através de *plugins*, postam as músicas que estão escutando em tempo real (caso estejam ligados a internet).

É importante ressaltar que redes sociais estão presentes em qualquer ambiente que existam pessoas e, também, formas para que as mesmas possam se comunicar.

O principal objetivo de uma rede social é proporcionar interações entre os usuários. Essas interações, dependendo de seus conteúdos, podem exercer certo tipo de influência nos usuários. E essa influência pode ser utilizada para diversos fins. Na próxima seção será feita uma abordagem sobre *Marketing Viral*, que tenta agir como um vírus fazendo uso dessa influência.

### **3 Marketing Viral**

A troca de influência em uma rede social é algo que constantemente acontece, seja por indicação individual (um usuário, baseado no seu conhecimento sobre o outro usuário, recomenda alguma coisa para outro usuário), por indicação coletiva (um grupo através de, por exemplo, uma enquete, recomenda o uso de determinada coisa para muitos usuários) etc.

*Marketing viral* pode ser entendido como uma forma de difusão de informações sobre um produto através das pessoas que compõem uma rede (LESKOVEC, 2007). Em outras palavras, *marketing viral* é um método que visa difundir informações sobre produtos dentro de uma rede social, através dos seus usuários.

A rede social ajuda nesse tipo de *marketing* devido a sua divisão de interesses, ou seja, existem locais da rede para diferentes contextos. Na maioria das redes é adotado o modelo de “comunidades” em que, cada uma delas, tem um tema, logo, as pessoas que participam de uma comunidade compartilham o interesse por esse tema. Com a existência dessa divisão, o *marketing* pode ser direcionado para apenas pessoas que são potenciais clientes.

### **4 Rede Social + Portal Acadêmico**

Levando em consideração uma estrutura idêntica ao do portal do CEULP/ULBRA ([www.ulbra-to.br](http://www.ulbra-to.br)) é possível fazer a implantação de uma rede social. Para explicar melhor o processo de anexação da rede ao portal, cada processo será dividido por tópicos. São eles: localização da rede no portal e formas de acesso, estrutura da rede, divulgação da rede e formas de cadastro.

#### **4.1 Localização da rede no portal e formas de acesso**

Um link no *menu* “Institucional” (que se encontra na página inicial) poderia ser criado para encaminhar o usuário para a rede, sendo este o principal *link*. Nas páginas individuais dos cursos também seriam criados *links* para o redirecionamento para a rede, porém, com uma diferença, esses *links* direcionariam o usuário direto para página principal do curso na rede. A estrutura da rede será explicada ainda. Caso existam escolas associadas (explicado mais à frente), é interessante que exista uma ligação entre os dois portais (universidade e escola), para que o acesso seja mais abrangente. Os usuários podem, também, acessar a rede pelo seu endereço, sem ter que passar por outras páginas antes.

A imagem abaixo mostra de forma geral as formas de acesso:



**Figura 34. Formas de Acesso.**

A Figura 1 apresenta os caminhos que os usuários podem tomar para chegar à rede. O banco de dados da rede social vai precisar de dados referentes aos cursos, professores e alunos, como será explicado na seção 4.4, por isso existirá uma dependência parcial.

Deveria existir uma colaboração entre idealizadores desse empreendimento e desenvolvedores do portal acadêmico assim como uma integração entre o sistema já existente e o novo. Poderia ser feito algo totalmente independente, para facilitar a implantação, porém, é interessante a idéia de integração, apesar de levar mais tempo e ser mais complexo de ser desenvolvido.

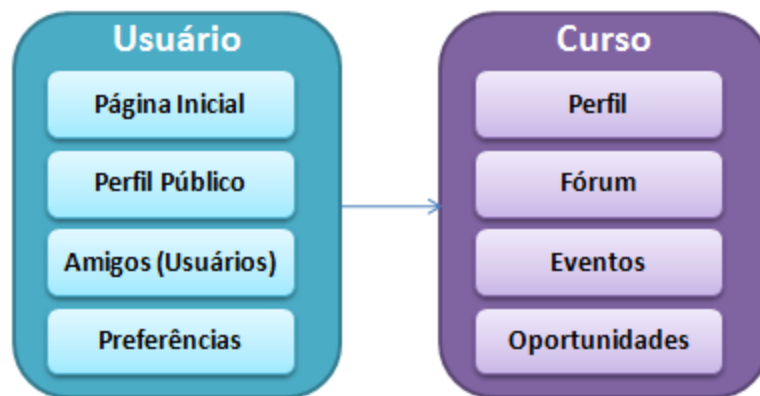
## 4.2 Estrutura da rede

A estrutura da rede seguirá a estrutura do portal, no que diz respeito a sua categorização de cursos. Será dividida pelo número de cursos oferecidos pela instituição. Cada curso terá seu espaço na rede, visando um direcionamento ao usuário de acordo com o seu interesse.

A rede irá proporcionar informações sobre cada curso de uma maneira menos “formal”. Muitas vezes, quando existe contato direto com pessoas envolvidas a algum domínio, é mais simples a forma de suporte a dúvidas sobre o domínio, pois, apenas um texto com a descrição de um curso não é o suficiente para esclarecer as dúvidas de uma pessoa. A presença de professores e alunos na rede é imprescindível nesse caso, pois, são eles os responsáveis pelo auxílio às pessoas que querem se graduar em algum curso.

Inicialmente a estrutura da rede contará com páginas pessoais dos usuários, visando à criação do perfil, e com páginas dos cursos, onde será realizada a maior parte das interações dos usuários. A imagem abaixo ilustra a estrutura:





**Figura 35. Estrutura da Rede.**

O conjunto de dados da criação do perfil das pessoas cadastradas vai abranger opiniões, interesses pessoais e profissionais, experiência em áreas, funções que exercem (além de estudante) etc. Como é uma rede social, é necessário que haja uma ampla visão dos usuários.

#### **4.3 Divulgação da rede**

A divulgação da rede aconteceria de três formas:

- dentro da universidade: após a criação da rede, poderia ser feita uma palestra na universidade para justificar a sua criação, seu objetivo, para demonstrar sua utilização etc. Antes disso, poderia ser postada uma notícia no portal da universidade sobre a palestra e a rede.
- fora da universidade: algumas instituições têm escolas associadas, como o CEULP/ULBRA. Com essa ligação poderia ser criados *banners* para serem mostrados nas escolas, pois, os estudantes de ensino médio fazem parte de um grupo alvo para a criação da mesma, como será visto posteriormente. A idéia de fazer uma palestra, também, pode ser usada nesse modo de divulgação.
- página de inscrição do vestibular: como o objetivo da rede é integrar alunos interessados no vestibular e alunos já matriculados na instituição, a página de inscrição no vestibular pode ser uma grande aliada para a divulgação da rede. Quando uma pessoa acessa a página do vestibular, ela se depara com um formulário que deve ser preenchido para concretizar sua inscrição. Nesse formulário a pessoa faz a escolha (ou, as escolhas) de cursos que deseja cursar. Nessa página pode existir uma imagem ou um texto fazendo uma breve explicação sobre a rede, explicando o que ela proporciona e o seu objetivo. Assim a pessoa já poderá se cadastrar na rede e já estar vinculada aos grupos que correspondem às opções de cursos escolhidos pela mesma.
- panfletos de divulgação do vestibular: são criados panfletos para divulgar o vestibular, sendo assim, poderia ser inserida uma propaganda da proposta da rede social. Os dados sobre a proposta seriam poucos, devido à falta de espaço, por isso,

teria que ser inserido um endereço *web* (de preferência) para orientar a pessoa sobre onde obter mais informações.

#### **4.4 Formas de Cadastro**

Os usuários da rede serão os alunos e professores da instituição, pessoas inscritas no vestibular e pessoas interessadas em ingressar em uma carreira acadêmica. A princípio, os cadastros à rede serão restritos a convites de pessoas já cadastradas.

No caso da divulgação em escolas, o cadastro seria aberto aos alunos da escola associada. Alunos e professores da instituição estariam vinculados diretamente a rede através de seus CGUs, no caso da ULBRA.

Os dados de cadastro serão, em síntese, os dados da inscrição no vestibular, no caso de recém inscritos no vestibular, adicionando apenas informações de acesso (*login* e senha). Para alunos de escolas, os dados serão mais genéricos (nome, data de nascimento, cursos de interesse, informações de acesso, etc). Não serão apenas esses dados que farão parte do perfil do usuário, como dito anteriormente.

### **5 Rede Social + Marketing Viral + Portal Acadêmico**

Como foi dito anteriormente, essa rede social tem como objetivo aumentar a procura no vestibular da instituição. Para isso, foi estabelecido que a rede iniciará suas atividades dentro da universidade, visando informar todos os acadêmicos de sua existência, para depois expandir sua abrangência. Claro que esses dois processos acontecerão simultaneamente, pois, não faz sentido disponibilizar um serviço para um público que não é o principal alvo do mesmo.

Para facilitar o entendimento de como irá funcionar o marketing e os seus possíveis efeitos, foram divididas as seguintes etapas do processo: seleção informal de alunos e professores, funcionamento e efeitos do marketing.

#### **5.1 Seleção informal de alunos e professores**

Nessa etapa, cada coordenador de cada curso fará uma chamada entre os alunos e professores a fim de obter colaboração dos mesmos, para participarem da rede social. Foi dito que todos os acadêmicos serão vinculados automaticamente a rede (no caso da ULBRA), porém, isso não quer dizer que todos irão aderir à idéia. Por isso, é importante que seja feita uma seleção de alunos que estejam interessados em ajudar pessoas a sanarem suas dúvidas em relação aos cursos. Vale ressaltar que o relacionamento entre aluno e curso dentro da rede é de 1 pra “n”, ou seja, um aluno não ficará preso à apenas um curso, pois, o aluno pode apresentar interesse em mais de uma área.

Esta etapa tem como objetivo estabelecer uma base de alunos que tenham interesse em ajudar. É indispensável essa base, pois, ela influi diretamente no sucesso da rede e do marketing.

#### **5.2 Funcionamento e efeitos do marketing**

O funcionamento do marketing é bastante simples devido a sua dependência com os envolvidos na rede social. A rede social reúne pessoas com interesses em comum que é de obter informações sobre uma instituição e seu processo de seleção. Além de prover informações sobre o vestibular, também servirá como um serviço que vai tirar dúvidas sobre os cursos oferecidos, acarretando em uma familiarização com alunos e

professores da instituição, ou seja, vai existir uma maior integração da parte interna da instituição e parte externa que possa vir, ou não, fazer parte da instituição.

A questão da restrição de cadastros pode agir como um filtro. Quando existe um cadastro aberto, o sistema fica mercê de cadastros inúteis, ou seja, cadastros que não irão trazer resultados positivos em relação ao objetivo do sistema, podendo até prejudicar o mesmo. Um filtro de cadastro ajudará a rede no sentido de apenas aceitar pessoas que, de fato, querem participar e que acreditam que a rede trará benefícios para elas mesmas.

A divulgação da rede nas escolas associadas deve ser feita com muito cuidado, devido à importância desse nicho. Alunos de ensino médio, muitas vezes, não conseguem decidir a carreira profissional que querem seguir. Por isso, a idéia de existir uma relação entre escola e faculdade pode ser crucial para essa decisão. Com isso, pode ser observada a magnitude de existir um mecanismo que faça ultrapassar a barreira entre dois tipos de instituições.

A maioria dos alunos que saem da escola quer ingressar em uma faculdade. Tendo isso em vista, pode-se concluir que se houver um modo de obter atenção desse público, que é o alvo do sistema, haverá mais possibilidade de êxito na oferta de cursos, no preenchimento de vagas e, especialmente, na qualidade dos candidatos, já que tal ação conduzirá a uma maior procura no vestibular, o que acarretará numa maior concorrência por curso. O marketing viral funcionaria de forma que os alunos pudessem usufruir de um ambiente voltado para seu interesse, visando esclarecimento e integração entre duas partes.

## **6 Considerações finais**

O uso de rede social pode ser um forte aliado na integração de pessoas e na difusão de informações. As redes estão presentes onde existe interação entre pessoas. Onde há interação, existe abundante criação e troca de informação. A rede cria a base para pessoas se relacionarem, que é o passo inicial para a troca de informações e influência.

O *marketing* viral usado em uma rede faz desta uma fonte de grande potencial comercial, pois, o nível de abrangência do *marketing* fica por conta de seus usuários, podendo atingir o meio virtual e o real.

Um portal universitário concentra muitas informações que podem, também, ser de interesse externo, ou seja, não só dos professores e alunos da instituição. Por intermédio de uma rede social, através do *marketing* viral, essas informações podem ser do conhecimento de mais pessoas. Portanto, a união dessas ferramentas pode ser de grande ajuda para levar um pouco da universidade para pessoas que têm interesse em ingressar em uma vida acadêmica.

## **7 Referências Bibliográficas**

BARNES, J. A. Redes sociais e processo político. In: FELDMAN-BIANCO, Bela (Org.). Antropologia das sociedades contemporâneas: métodos. São Paulo: Global Universitária, 1987. p. 159-189.

LESKOVEC, J., Adamic, L. A., Huberman, B. A., The dynamics of viral marketing, ACM Transactions on the Web (TWEB), vol. 1, ed. 1, n. 5, 2007.



# Uso do Netkit no Ensino de Roteamento Estático

Autor(es)

Nyl Marcos Soares Barbosa, Moisés Lima dos Anjos, Madianita Bogo

Curso de Sistemas de Informação – Centro universitário Luterano de Palmas  
(CEULP/ULBRA)

Teotônio Segurado 1501 Sul – Caixa Postal 160 – 77054970 – Palmas –TO -Brasil

{nyl1702, mozart08, madianitab}@gmail.com

***Resumo.** Para a compreensão de alguns conceitos das disciplinas da área de Informática é fundamental a associação entre teoria e prática. As disciplinas da área de Computadores abordam vários conceitos que, para serem associados à prática, necessitam da configuração de ambientes de rede com vários componentes. Oferecer uma estrutura física adequada para esse estudo acarretaria em um custo financeiro muito alto para as instituições. Uma solução satisfatória para esse tipo de problema é a utilização de emuladores que possibilitem criar ambientes virtuais de rede que funcionem de forma semelhante a ambientes reais, como o netkit. Esse artigo apresenta um ambiente de rede virtual, configurado no netkit, que foi utilizado para o ensino do roteamento estático, na disciplina de Redes de Computadores I, do curso de Sistemas de Informação do CEULP/ULBRA.*

**Palavras chave: netkit, roteamento estático, emulador.**

## 1. Introdução

Em algumas disciplinas da área de Informática é fundamental que os professores associem as teorias estudadas à utilização prática das tecnologias que utilizam os conceitos apresentados.

As disciplinas da área de Redes de Computadores abordam vários conceitos de comunicação que são difíceis de ser compreendidos pelos alunos caso o funcionamento não seja visto na prática. O problema é, para que se tenha uma relação adequada entre teoria e prática, é necessário criar ambientes de rede que tenham vários componentes de *hardware*: computadores; concentradores, como hubs e *switches*; cabos ou *access-point* para a ligação; entre outros. Dessa forma, oferecer uma estrutura física adequada para esse estudo prático acarretaria em um custo financeiro muito alto para as instituições de ensino. Adicionalmente, seria muito difícil uma instituição criar vários ambientes, de forma que todos os alunos teriam que usar o mesmo ambiente para seus estudos.

Uma solução satisfatória para esse tipo de problema é a utilização de emuladores que possibilitem criar ambientes virtuais de rede que funcionem de forma semelhante a ambientes reais. Uma solução gratuita e simples de ser utilizada é o netkit, que é um emulador que possibilita a reprodução do comportamento e das funcionalidades de redes de computadores reais, pequenas ou grandes, criando máquinas virtuais que formam um cenário de rede em um único computador. Com o netkit, além dos alunos poderem configurar um ambiente desde o início, para aprender a configurar redes, podem-se criar laboratórios, que são ambientes de rede pré-configurados, que podem ser

utilizados para que os alunos compreendam conceitos em ambientes complexos, sem se preocupar com a configuração destes.

Pensando nisso, utilizou-se o netkit no ensino do roteamento estático, na disciplina de Redes de Computadores I do curso de Sistemas de Informação do CEULP/ULBRA. Para isso, além de projetar um ambiente de rede adequado, que possuísse os componentes mínimos necessários para a compreensão do tema e que não fosse muito complexo, foi realizada a implementação de um laboratório com a configuração deste ambiente. A seguir serão apresentadas uma visão geral sobre o netkit, a arquitetura do ambiente de rede projetado e o processo de criação e utilização do laboratório que configura este ambiente.

## 2. Netkit

O Netkit é um *software* livre, criado por um grupo de pesquisadores da Universidade de Roma, que permite emular, em um único *host*, cenários de rede com diversos níveis de complexidades, visando reproduzir no ambiente virtual as características do ambiente de rede real.

O netkit é constituído de dois componentes principais: um conjunto de scripts e ferramentas que facilitam a criação de cenários de rede complexa formados por vários dispositivos virtuais; e um conjunto de experiências virtuais prontas para o uso, chamadas de laboratórios, que podem ser usados para estudar os ambientes de rede típicos (RIMONDINI, 2007, *on-line*). Os ambientes são criados sobre a plataforma Linux, distribuição Debian.

Vale ressaltar que, apesar de possibilitar a criação de ambientes de redes complexos, a utilização do Netkit não requer um grande conhecimento por parte do utilizador, já que sua utilização é muito simples.

O netkit foi criado visando auxiliar nas pesquisas e no ensino de conteúdos relacionados da área de Redes de Computadores, tendo em vista que é difícil pensar no aprendizado sem que o aluno realize atividades práticas sobre ambientes de rede completos e que criar ambientes reais tem custos elevados. Por exemplo, para que o roteamento seja visualizado na prática, sem a utilização do netkit, seriam necessários, no mínimo três computadores, sendo que um deles precisaria de duas placas de redes, além do cabeamento necessário para a interconexão. Considerando que o ideal seria cada aluno, ou dupla de alunos, pudesse ter seu próprio ambiente, demandaria um custo financeiro e de espaço muito elevados. A emulação dos cenários possibilita oferecer suporte a disciplinas de redes de computadores, com custo muito baixo.

Com a utilização do Netkit, o professor dará a seus alunos a oportunidade de ter um contato com a prática dos seus próprios experimentos, cada um poderá criar seu próprio ambiente, com *hosts* e roteadores, e poderá fazer as configurações desejadas no ambiente virtual. A mesma configuração feita em um *host* do ambiente virtual poderá ser feita em um *host* de um ambiente real, facilitando muito o aprendizado do aluno e dando a este a oportunidade de aprender na prática o que usará no mercado de trabalho.

Um recurso importante do Netkit, que pode auxiliar na prática do ensino de redes de computadores, é a possibilidade de criação de laboratórios. “Laboratório é o nome que é associado a cenários de rede pré-configurados” (RIMONDINI, 2007, *on-line*). Assim, é possível a criação de um conjunto de máquinas virtuais pré-

configuradas, formando um ambiente de rede completo, que podem ser iniciadas ou paradas simultaneamente utilizando um único comando.

A utilização de laboratórios pode ter grande impacto no aprendizado da disciplina de redes de computadores. Por intermédio deles é possível criar um ambiente com diversas máquinas, entre elas roteadores, que possibilite aos alunos analisar o funcionamento dos ambientes de rede sem ter o trabalho de realizar todas as configurações.

O netkit fornece vários laboratórios prontos que vão desde a criação de ambientes de protocolos de roteamento, como RIP, aos serviços de nível de aplicação, como DNS (Roma TRE, 2009, *on-line*).

A seguir será apresentada a arquitetura de um ambiente de rede projetado para que os alunos compreendam o funcionamento do roteamento estático, que foi utilizado na disciplina de Redes de Computadores I, do curso de Sistemas de Informação do CEULP/ULBRA.

### 3. Ambiente implementado – Roteamento Estático

Antes de iniciar a configuração de um ambiente de rede ou a criação de um laboratório, é importante que seja desenhada a estrutura da rede a ser emulada, pois esta fornecerá todas as informações necessárias para uma correta configuração e para que os alunos compreendam o que estão utilizando na prática.

O ambiente criado para o estudo de roteamento estático, que é apresentado na figura 1, consiste em dois roteadores e três hosts. O ambiente possui dois domínios de colisão (**A** e **B**), além de uma rede virtual entre a máquina real e o roteador virtual (router 1). Para que um ambiente real com essas características fosse implantado, seriam necessários os seguintes equipamentos: 5 computadores (três *hosts*, dois roteadores (um com duas interfaces de rede e outro com três) e dois *switches* ou *hubs*).

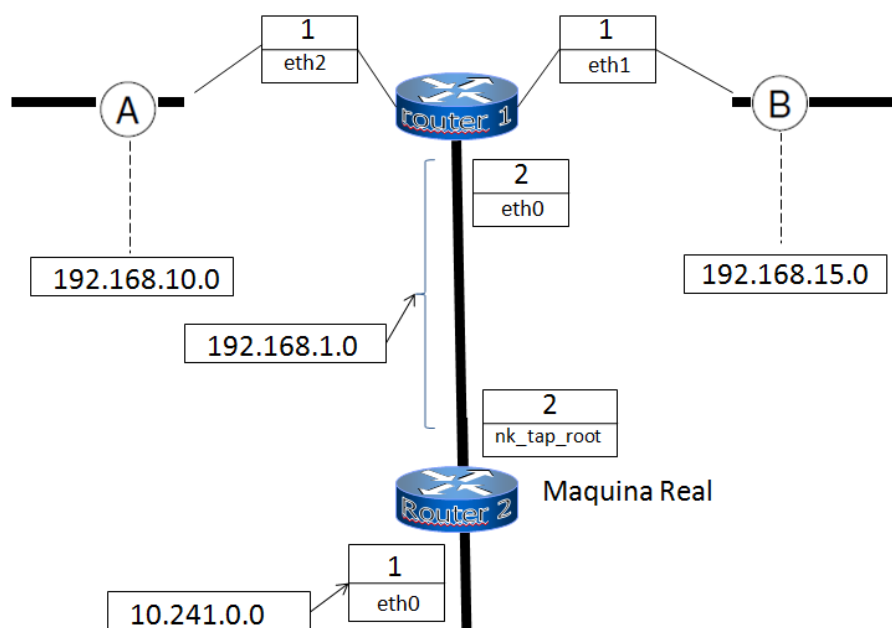


Figura 1: Ambiente Implementado

Como pode ser visto na figura 1, o roteador 1 (**router 1**), que é uma máquina virtual, realiza a ligação do ambiente virtual com a máquina real (**router 2**), permitindo assim que todo o ambiente obtenha acesso à internet. Na máquina real (**router 2**), que está ligada com a rede real **10.241.0.0**, é criada uma interface de rede virtual (**nk\_tap\_root**), com a qual se comunicará com o ambiente de rede virtual emulado. A rede do domínio de colisão **A** é a **192.168.10.0**, a rede do domínio de colisão **B** é a **192.168.15.0** e a rede que interliga a máquina real com o ambiente virtual é a **192.168.1.0**.

No domínio de colisão **A** foram configurados dois *hosts* e no **B** um *host*. Cada *host* virtual possui uma configuração própria de rede, igual a de um *host* real.

Existem duas formas de se utilizar esse ambiente para o ensino do roteamento. Os alunos podem fazer a configuração manual de cada uma das máquinas virtuais, para aprender os comandos necessários para configurar e testar uma rede, já que a mesma configuração é a que deve ser realizada nas máquinas reais. Também, o professor pode criar um laboratório que possibilita a execução de todo ambiente usando apenas um comando, para que os alunos executem e compreendam o funcionamento, sem se preocupar com a parte da configuração do ambiente.

A seguir, é apresentada a criação e a utilização do laboratório implementado para criar o ambiente de rede apresentado na figura 1.

#### 4. Laboratório: Ambiente de Roteamento Estático

Para criar o laboratório do ambiente projetado deve-se criar o arquivo **lab.conf**, que conterá a descrição do laboratório e a configuração da rede, além de poder ter outras configurações adicionais, como, por exemplo, a quantidade de memória a ser usada para cada máquina virtual. A figura 2 apresenta o conteúdo do arquivo **lab.conf**, criado para configurar o ambiente de rede representado na figura 1.

Como pode ser observado na figura 2, o arquivo **lab.conf** contém o nome das máquinas virtuais que serão criadas, associados aos seus respectivos domínios de colisão: **r1** (router 1) faz parte dos domínios de colisão **A** e **B**, **pc1** e **pc2** fazem parte do domínio de colisão **A** e o **pc3** faz parte do domínio de colisão **B**. O comando **tap**, utilizado na definição da interface de rede **0** do **router1**, é que possibilita a comunicação da máquina real com as máquinas do ambiente virtual.

```
LAB_DESCRIPTION="Ensinando roteamento estático com netkit"
LAB_VERSION="0.1"
LAB_AUTHOR="Soares Nyl, Lima Moises"
LAB_EMAIL="nyl1705@gmail.com , moztart08@gmail.com"
LAB_WEB=http://www.ulbra-to.br/

r1[0]=tap,192.168.1.2,192.168.1.1
r1[1]=A
r1[2]=B

pc1[0]=A
pc2[0]=A
pc3[0]=B
```

Figura 2: Conteúdo do arquivo lab.conf



Em seguida, deve-se realizar a configuração do arquivo **.startup**, um para cada máquina virtual, que conterà as ações a serem realizadas quando o laboratório for iniciado. A figura 3 apresenta o conteúdo do arquivo **router1.startup**, criado para configurar o **router 1**, do ambiente de rede representado na figura 1. As ações configuradas foram: nos três primeiros comandos, configurar as interfaces de rede; e no último comando, configurar o servidor DNS que será utilizado.

```
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
ifconfig eth1 192.168.10.1 netmask 255.255.255.0 up
ifconfig eth2 192.168.15.1 netmask 255.255.255.0 up
echo "nameserver 200.199.229.66" >> /etc/resolv.conf
```

**Figura 2: Conteúdo do arquivo router1.startup**

Pode-se configurar ainda o arquivo **lab.dep**, que deve ser feito se houver a necessidade de se inicializar as máquinas virtuais em paralelo. Nesse caso, é importante informar as dependências, quando existirem, entre as máquinas virtuais. Por exemplo, se uma máquina virtual for usar um serviço que outra máquina irá inicializar, existirá uma dependência da primeira em relação à segunda. Na construção do laboratório esse arquivo não foi configurado, pois não se constatou tal necessidade.

Por fim, há o diretório **\_test**, que pode ser utilizado para realizar testes com o laboratório criado. Para isso, basta inserir *scripts* para executar nas máquinas virtuais que se deseja testar, que podem conter comandos como *ping* ou *traceroute*. Essa bateria de testes pode ser usada para gerar uma assinatura que, posteriormente, poderá ser comparada com uma outra gerada em uma diferente plataforma, a fim de verificar se o laboratório irá funcionar corretamente nesta. Não se fez uso deste recurso, haja vista que não houve necessidade de executar o laboratório em outras plataformas.

A seguir é apresentado como executar o laboratório implementado e é mostrado o ambiente criado na máquina real com essa execução.

#### **4.1. Execução do Laboratório**

Para iniciar a execução de um laboratório, é necessário apenas digitar o comando **lstart**, dentro do diretório no qual os arquivos de configuração estão localizados. A figura 4 mostra os terminais criados, um para cada máquina virtual (**r1**, **pc1**, **pc2** e **pc3**), quando foi executado o laboratório apresentado anteriormente. A figura 4 foi inserida apenas para que se tenha uma visão do ambiente de trabalho criado, o que é possível mesmo sem visualizar o conteúdo de cada terminal.

A máquina virtual **r1** possui três interfaces de redes, sendo que a interface **eth0** é configurada na mesma sub-rede da interface virtual que foi criada na máquina real com o comando *tap*. As outras interfaces de rede são configuradas nas redes de seus respectivos domínios de colisão, como mostra a figura 5, que apresenta o resultado da execução do comando **ifconfig**, no terminal de **r1**. O comando retorna as configurações de todas as interfaces de rede configuradas.

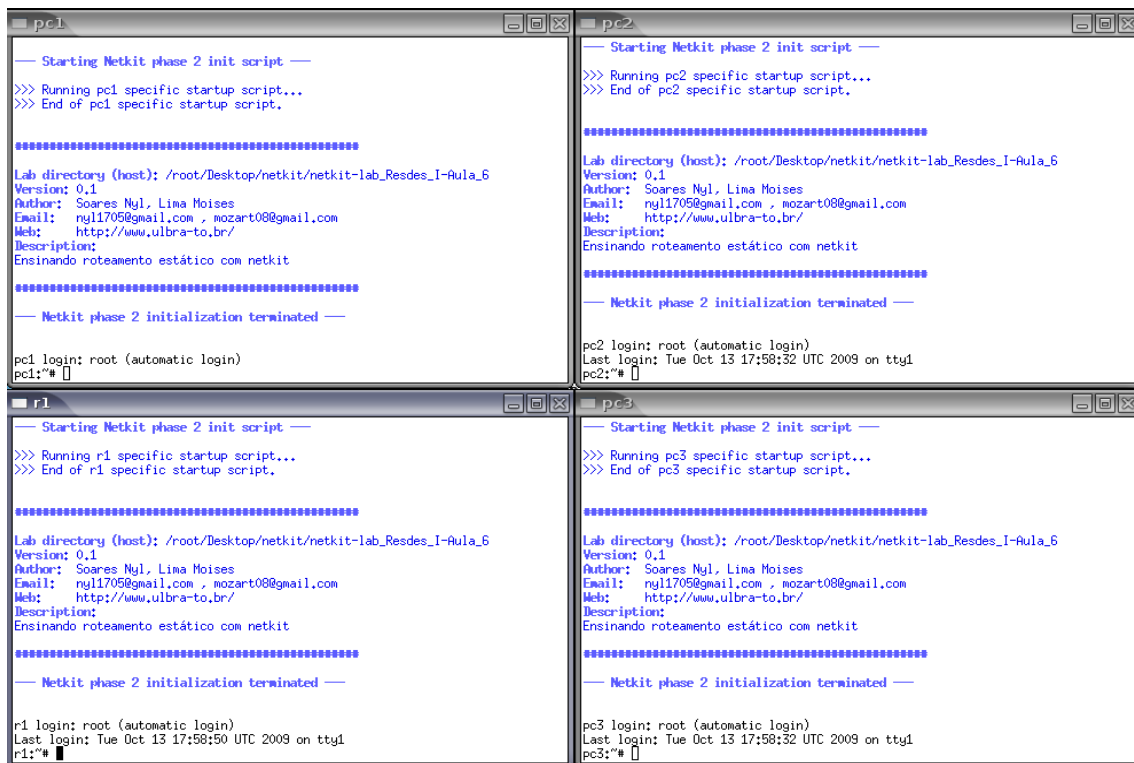


Figura 4: Terminais das máquinas virtuais criadas

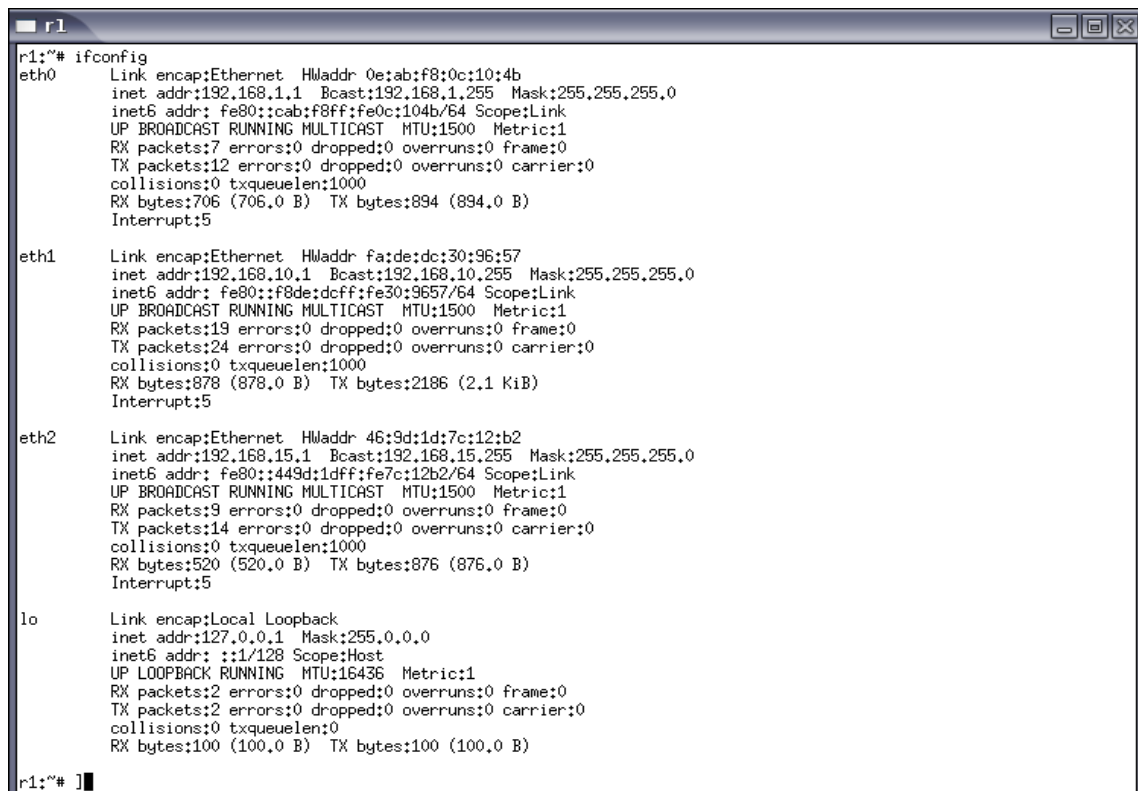
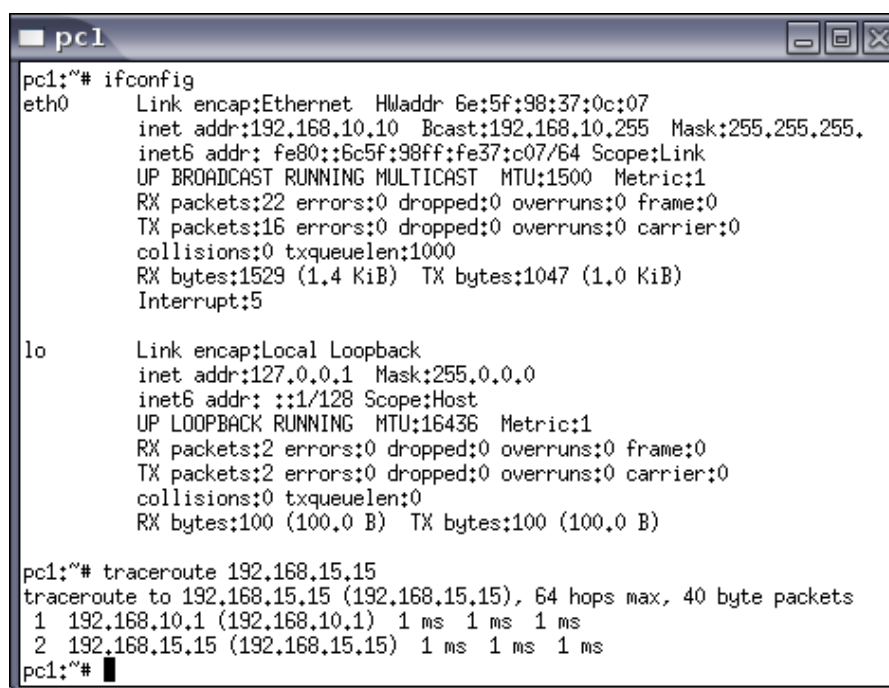


Figura 5: Interfaces configuradas em r1 (router1 da figura 1)

As máquinas virtuais **pc1** e **pc2** pertencem ao domínio de colisão **A**, e **pc3** pertence ao domínio **B**. A figura 6 mostra a configuração de interface de rede e a execução do comando “*traceroute 192.168.15.15*” no terminal da máquina virtual **pc1**.



```
pc1:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 6e:5f:98:37:0c:07
          inet addr:192.168.10.10  Bcast:192.168.10.255  Mask:255.255.255.
          inet6 addr: fe80::6c5f:98ff:fe37:c07/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:22 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1529 (1.4 KiB)  TX bytes:1047 (1.0 KiB)
          Interrupt:5

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:100 (100.0 B)  TX bytes:100 (100.0 B)

pc1:~# traceroute 192.168.15.15
traceroute to 192.168.15.15 (192.168.15.15), 64 hops max, 40 byte packets
 1  192.168.10.1 (192.168.10.1)  1 ms  1 ms  1 ms
 2  192.168.15.15 (192.168.15.15)  1 ms  1 ms  1 ms
pc1:~# █
```

Figura 5: Retorno dos comandos *ifconfig* e *traceroute*, executados no **pc1**

O comando *traceroute* apresenta o caminho que uma mensagem percorre para chegar ao seu destino. O comando executado retornou o caminho que uma mensagem percorreu para sair de **pc1** (192.168.10.10) e alcançar **pc3** (192.168.15.15). Como pode ser observado na figura 6, a mensagem que sai de **pc1** (domínio de colisão A) é encaminhada para **r1** (192.168.10.1) e este encaminha a mensagem para **pc3** (domínio de colisão B), já que **r1** é o roteador que faz a ligação do domínio de colisão **A** com o domínio de colisão **B**.

Assim como foi executado esse comando, é possível executar vários comandos que para auxiliar o aluno a compreender os conceitos relacionados ao roteamento estático, como o comando *route* para ver as tabelas de roteamento dos roteadores e dos *hosts* virtuais.

## 5. Conclusões

O netkit pode auxiliar muito no ensino de conceitos relacionados a área de Redes de Computadores, facilitando a associação entre a teoria e a prática. O professor tem várias possibilidades: utilizar um dos laboratórios prontos disponíveis na página do netkit ([www.netkit.org](http://www.netkit.org)); criar laboratórios com ambientes projetados para situações específicas; ou projetar ambientes e deixar que os alunos configurem estes desde o início, para, além de compreender os conceitos, aprendam a configurar as redes.

O netkit, como descrito nesse artigo, foi utilizado para o ensino do roteamento estático. A sua utilização mostrou-se satisfatória, já que em um único laboratório, vários alunos puderam trabalhar individualmente sobre um ambiente de rede completo,

executando em único computador. Com isso, o conceito de roteamento foi assimilado com mais facilidade.

Apesar de inicialmente ter sido projetado com fins educacionais, com o tempo o netkit foi saindo do cenário acadêmico e entrando no meio profissional, pois para os administradores de redes é interessante saber antecipadamente se determinadas configurações terão os resultados esperados quando forem implementadas, sem a necessidade de gastar com *hardware* antes de ter a certeza do sucesso do projeto. Desta forma, os alunos que utilizarem o netkit para o seu aprendizado, poderão utilizá-lo no mercado de trabalho.

## **6. Referências Bibliográficas**

- (Rimondini, 2007) Rimondini Massimo, Emulation of Computer network with Netkit. 2007. Disponível em <http://www.netkit.org/publications/netkit-tr.pdf>. Acesso em agosto de 2009. Publicado em janeiro de 2007.
- (Roma TRE, 2009) University of Roma Tre - Computer Networks Research Group. Labs and Lecture Slides. Disponível em <http://www.netkit.org/labs.html>. Acesso em setembro de 2009.

# Utilização de Agentes Inteligentes no desenvolvimento de um Jogo RPG para dispositivos móveis

Heitor de Sousa Miranda, Fernando Luiz de Oliveira

Curso de Sistemas de Informação - CEULP/ULBRA

Teotônio Segurado 1501 SUL - 77054-970 - Palmas - TO - Brasil

{heitordesousa, nandoluiz.br}@gmail.com

**Resumo.** *Este trabalho tem como objetivo apresentar o desenvolvimento de um protótipo de um jogo RPG para dispositivos móveis. Para tanto, serão apresentados alguns conceitos sobre inteligência artificial, abordando especificamente o uso da técnica de agentes inteligentes, assim como uma breve abordagem sobre jogos para aparelhos celulares, apresentando a tecnologia usada para esse tipo de desenvolvimento. Por fim, são apresentados resultados da implementação do protótipo do jogo, bem como a utilização da técnica de agentes inteligentes para dar um caráter inteligente a alguns elementos do jogo.*

**Palavras-chave:** Inteligência Artificial, Agentes Inteligentes, Jogos.

## 1 Introdução

As tecnologias usadas na criação de jogos evoluem continuamente, fazendo com que a qualidade dos jogos eletrônicos seja cada vez mais aperfeiçoada. Esse avanço possibilitou que a experiência do jogador não fosse limitada a uma máquina, ou seja, possibilitou ao jogador novas plataformas e novos meios de jogar. Nos primórdios, o jogador era limitado a usar uma máquina, cuja única utilidade era suportar jogos. Atualmente, devido a essa evolução, os jogos estão presentes em muitos dispositivos eletrônicos, como, por exemplo, computadores, celulares, MP3 players, etc.

Porém, apenas a utilização de tecnologias capazes de melhorar os elementos gráficos e a interação homem-computador, entre outras características, não é o suficiente para cativar o jogador. Todo jogo tem como objetivo satisfazer o jogador de forma que ele se sinta atraído, desafiado e, ao mesmo tempo, entretido. A utilização de Inteligência Artificial (IA) fez com que os jogos se tornassem, de certa forma, mais inteligentes, capazes de fazer com que os jogos atendessem a essas necessidades do jogador.

Um dos principais objetivos da IA é possibilitar a criação de sistemas autônomos, ou seja, sistemas que tenham a capacidade de tomar decisões visando alcançar um objetivo, o qual deve ser definido durante a sua implementação. A intervenção da Inteligência Artificial fez com que uma gama de novas aplicações fossem descobertas, como na robótica, mineração de dados (*Data Mining*), análise de imagens, jogos, etc; possibilitando não só essa descoberta, mas também fazendo com que os sistemas que já existiam melhorassem suas funcionalidades.

No começo do desenvolvimento de jogos eletrônicos, a programação de IA era mais usualmente conhecida por “programação de jogabilidade”, pois não havia nada de

inteligente sobre os comportamentos exibidos pelos personagens controlados pelo computador (SCHWAB, 2004). Esses personagens podem ser entendidos como elementos do jogo, animados ou inanimados, pré-definidos na criação do jogo, que fazem interação com outros personagens e que, muitas vezes, fazem parte da história principal do jogo.

O objetivo deste trabalho é apresentar o desenvolvimento de um jogo para dispositivos móveis, expondo como a Inteligência Artificial pode ser utilizada (mais especificamente a técnica de Agentes inteligentes) para embutir certa inteligência aos elementos do jogo. Também será feita uma abordagem teórica sobre IA, a técnica utilizada no trabalho e uma descrição do tipo de jogo que foi o alvo do trabalho. Por fim, é apresentado o resultado da utilização da técnica em um protótipo do jogo, descrevendo a plataforma (linguagem e ferramenta) e a modelagem do agente inteligente.

## **2 Inteligência Artificial**

A Inteligência Artificial tem como uma de suas principais características a capacidade de possibilitar a criação de sistemas que, de alguma forma, apresentem um índice de autonomia para realização de operações, os quais seriam provenientes de sua forma inteligente de agir, de acordo com o seu objetivo. O estudo da computação que torna possível perceber, raciocinar e agir (WINSTON, 1982).

Os estudos de IA se iniciaram após o final da Segunda Guerra Mundial, com um artigo de Alan Turing, matemático inglês. Apesar de naquela época não existirem computadores avançados, já se pensava em técnicas para que pudessem ser criados computadores que apresentassem algum nível de inteligência. Redes Neurais Artificiais é um exemplo de técnica que foi pensada nos primórdios dos estudos voltados para a IA.

Novos problemas surgem de acordo com o aumento da abrangência dos tipos de *softwares* criados. Conseqüentemente, para novos problemas, novas soluções devem ser exploradas. A IA possui flexibilidade para a resolução de problemas, pois existem técnicas específicas para determinados tipos de problema.

Neste trabalho será abordada a utilização da técnica Agentes Inteligentes na criação de um jogo. Esta técnica foi escolhida devido a sua facilidade de implementação e sua grande eficácia quando aplicada em jogos.

## **3 Agentes Inteligentes**

Até a década de 80, os problemas eram resolvidos através da inteligência humana e de modo individual. Com o passar dos anos e o aumento na complexidade dos problemas, tornou-se claro a necessidade da utilização de algoritmos na construção de sistemas que pudessem auxiliar a inteligência humana.

Um agente é qualquer entidade capaz de perceber o mundo exterior através de sensores e agir neste mundo empregando atuadores (RUSSELL & NORVIG, 2004). Ou seja, um agente funciona de modo a agir de acordo com suas percepções sobre o ambiente em que está empregado. A seguir são apresentadas características de um agente:

- autonomia: capacidade de um agente atuar em determinado ambiente de forma independente, sem a intervenção do homem ou de outros agentes, podendo planejar,

monitorar e avaliar as ações a serem tomadas, sem perder o controle do seu estado interno. Uma das definições mais familiares é a de Holec (1981) que, segundo ele, autonomia é “a habilidade de se responsabilizar pela própria aprendizagem”.

- pró-atividade: capacidade de um agente tomar decisões sobre um determinado ambiente, de forma orientada para atingir os seus objetivos, ou seja, é poder reagir aos estímulos externos, tomando a iniciativa quando apropriado.
- reatividade: capacidade de resposta dos agentes em detectar as mudanças no ambiente no qual estão inseridos, fazendo com que sejam tomadas medidas antecipadamente a essas mudanças.
- sociabilidade: capacidade de analisar a forma de relacionamento dos trabalhos em grupos com os demais agentes ou com humanos, em um mesmo ambiente, sempre no intuito de alcançar seus objetivos.

De acordo com essas características é possível concluir que um agente inteligente se relaciona diretamente com o seu meio e com os elementos presentes nesse meio, visando alcançar seus objetivos (executar ações a partir de monitorações).

#### **4 Role-Playing Game**

Um *Role-Playing Game* (RPG), ou, traduzindo, Jogo de Interpretação de Papel, é um tipo de jogo em que o jogador, para jogar, cria um personagem com características de sua preferência, como, cor de cabelo, estatura, peso, cor de pele, cor da roupa, etc. Nem sempre o jogo permite um alto nível de detalhamento na criação do personagem.

Como todo jogo, um RPG, obrigatoriamente, deve possuir uma história. Tal história pode ser entendida como um *script* no qual o jogador deverá conhecer para poder cumprir os objetivos do jogo. Alguns jogos, devido a sua complexidade, podem apresentar diferentes histórias a serem seguidas, aumentando assim a diversidade de aventuras nas quais o jogador poderá “viver”.

Basicamente, o jogador entra em um mundo virtual com esse personagem e "vive" as aventuras dentro desse mundo, finalizando *quests* (missões), derrotando monstros, fazendo amizades e interagindo com personagens de outros jogadores (no caso de um *RPG Online*), criando *pets* (animais de estimação), comprando e vendendo itens, etc. Não é obrigatório existirem todas essas características em um RPG, pois, depende da história do jogo.

Em jogos, seja RPG ou não, existem personagens que não são controlados por jogadores, ou seja, são elementos nativos do jogo, pré-programados para executar determinadas funções. São chamados de *Non-Player Character* (NPC). Esses personagens, geralmente, estão presentes para dar suporte ao jogador e/ou para dificultar a vida do personagem (no caso de monstros ou algum elemento de caráter inimigo). São essenciais devido a sua importância, muitas vezes, para o entendimento da história central do jogo, pois, alguns, podem servir para exibir mensagens cruciais para o progresso da aventura do personagem. Esses personagens, normalmente, são criados utilizando IA, pois, devem apresentar inteligência para poderem interagir com o ambiente e com o personagem do jogador.

Um RPG se difere de um jogo qualquer devido a sua flexibilidade de direções a serem tomadas durante o progresso do jogo. A história principal existe, entretanto, ela pode ser seguida ou não, vai depender do jogador e da própria flexibilidade do jogo. Além disso, devido a essa flexibilidade, o jogo pode apresentar finais alternativos, ou seja, durante a jornada do personagem, podem ser feitas escolhas que afetarão o seu destino de forma significativa. É, de fato, uma vida, porém, virtual.

Na próxima seção será apresentada a plataforma na qual o protótipo do jogo irá ser desenvolvido e executado, assim como a ferramenta utilizada no seu desenvolvimento.

## 5 Plataforma e ferramenta utilizada na criação do jogo

O jogo foi criado usando a tecnologia Java ME (ou J2ME) devido à sua simplicidade e, poder na criação de jogos. J2ME é uma plataforma da *Sun Microsystems* que usa a linguagem Java para o desenvolvimento de aplicativos para dispositivos móveis. Pode ser visto como uma simplificação do J2SE (*Standard Edition*) com algumas modificações para possibilitar que Java rode em dispositivos com capacidade limitada (celulares, *smartphones*, etc).

Como ambiente de desenvolvimento foi usada a ferramenta *NetBeans*, que possui um suporte a tecnologia Java ME, facilitando o seu desenvolvimento. Esta ferramenta também pertence a *Sun*, porém, não foi criada por essa empresa. A vantagem de poder utilizar essa ferramenta para desenvolvimento móvel é que, além de sua organização, ela permite programação visual, ou seja, *drag-and-drop* (arrastar e soltar), o que facilita bastante o processo de criação, aumentando a produtividade e rapidez em que as aplicações são desenvolvidas.

### 5.1 Construtor de Jogos

O *NetBeans* possui um módulo de construção de desenvolvimento sobre a plataforma J2ME voltada, apenas, a jogos, chamada de “Construtor de Jogos”. Neste módulo estão presentes três elementos de criação de jogos, que são: cenas, camadas xadrez e *sprite*. Cada um desses conceitos serve para separar os elementos dos jogos, melhorando a organização interna do jogo, facilitando a utilização dos mesmos na hora de alocá-los no, de fato, jogo. É importante ressaltar que esse é apenas um modelo organizacional dos recursos do jogo, não é algo que deve ser seguido à risca para que o jogo funcione. Como foi dito, é apenas uma forma que desenvolveram para organizar esses elementos. A seguir é explicado cada um destes três conceitos.

- *Sprites*: são elementos que possuem “vida” dentro do jogo (personagem, por exemplo). Nesse módulo, a criação de *sprites* é muito facilitada, pois, permite a animação dos personagens de forma simples. Cada *sprite* criado gera um objeto do tipo *Sprite* e vetores de *int* (vai depender da quantidade de animações criadas). Um *Sprite* é gerado a partir de uma imagem, assim como uma camada xadrez.
- Camadas Xadrez: são elementos responsáveis por representar áreas de um jogo. O uso desse recurso é importante para que exista uma separação de elementos internos do jogo, como, por exemplo, um rio ou uma casa. Este elemento também possibilita tratamento de colisão entre elementos (*sprites* e camadas xadrez). Uma camada xadrez gera um objeto do tipo *TiledLayer*.



- Cenas: são elementos que fazem uma indexação de camadas xadrez, ou seja, essas camadas são fragmentos de uma cena. A utilização de cenas é muito interessante pelo fato de possibilitar a interação entre todos os elementos do jogo. Uma cena gera um objeto do tipo *LayerManager* que recebe objetos do tipo *TiledLayer*.

## 6 Descrição do Cenário e dos Personagens do Jogo

Como dito anteriormente, o protótipo do jogo foi criado para ser executado por celulares que suportam a tecnologia Java ME. Apesar da rápida evolução dos dispositivos móveis, é necessário um cuidado em relação aos recursos utilizados na criação de aplicativos para essa plataforma. Isso se deve ao fato desses dispositivos não possuírem um alto nível de memória disponível e de capacidade de processamento.

Para criação deste jogo foram usados os recursos disponíveis pelo Construtor de Jogos do *NetBeans*, são eles: camadas xadrez, *sprites*, e uma única cena. A figura abaixo ilustra o cenário do jogo, onde os personagens irão interagir.

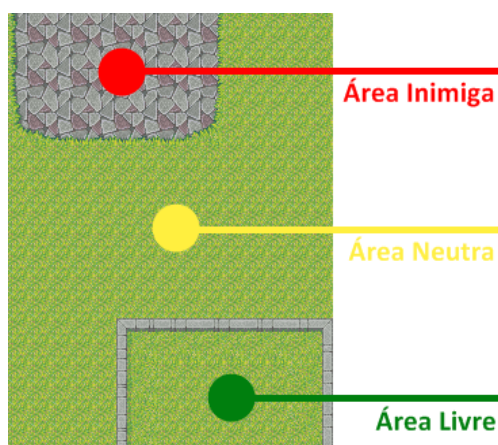


Figura 36. Cenário.

Como mostrado na figura, existe uma cena que contém três diferentes áreas (que são camadas xadrez). A seguir é apresentada a descrição de cada uma dessas camadas.

- Área Inimiga: camada xadrez que representa o território inimigo. Também é a área responsável pela interação do inimigo com o personagem. Será mais detalhado em outra seção.
- Área Neutra: camada xadrez que representa o território neutro, onde o personagem pode andar livremente. Em apenas uma situação (que será explicada em outra seção) o personagem não poderá andar livremente por esta área.
- Área Livre: camada xadrez que representa o território do personagem. Não haverá perseguição enquanto o personagem estiver nessa área.

A figura abaixo mostra os personagens do jogo:



Figura 37. Personagens.

O usuário (jogador) do jogo terá controle do personagem, podendo apenas fazê-lo andar para cima, baixo, esquerda e direita. O personagem terá acesso livre a todo o cenário do jogo, podendo andar em qualquer parte do cenário. O inimigo, será um NPC, não será controlado pelo jogador, será apenas o elemento que irá responder às ações do herói, ou seja, o inimigo é personagem que irá agir a partir do monitoramento do seu ambiente. Em outras palavras, o inimigo é a entidade que terá um agente inteligente vinculado, definindo seu comportamento de acordo com o comportamento do personagem. Na seção a seguir será explicado o que o agente irá monitorar e as suas ações de acordo com o monitoramento.

## 7 Funcionamento do Jogo

O jogo ainda está em sua versão preliminar e o objetivo central é integrar o conceito de agentes inteligentes ao jogo. Por isto, o funcionamento do jogo é simples, pois se trata apenas de uma perseguição do inimigo ao personagem. A figura abaixo apresenta o fluxo básico dos eventos do jogo:

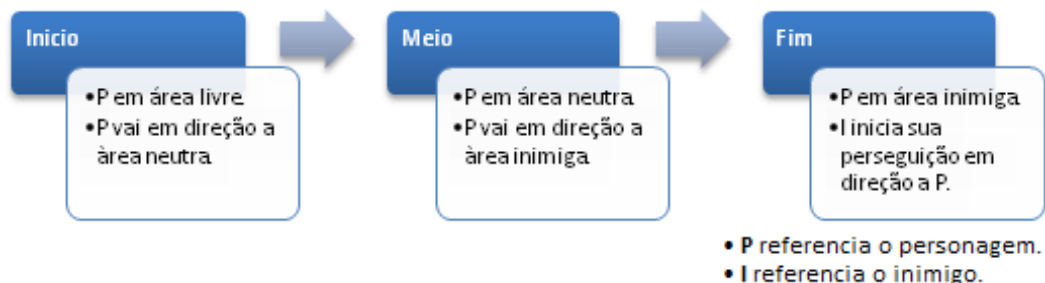


Figura 38. Fluxo básico dos eventos.

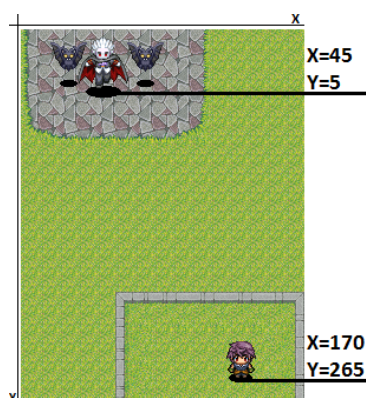
O “fim” apresentado na figura não é o fim propriamente dito, pois, o jogo simplesmente não acaba depois que o personagem invade a área inimiga. Como foi dito, este é apenas o fluxo básico dos eventos. Após o personagem invadir a área hostil, ele pode: permanecer parado enquanto o inimigo vai a seu encontro. Isso ocasionará no fim do jogo, pois, o inimigo alcançou o personagem.; ou fugir do inimigo, retornando para a área livre. Quando o personagem retorna a esta área, o inimigo volta para seu território. A seguir é feita uma explicação detalhada sobre a monitoração feita nas ações do personagem.

### 7.1 Monitoramento das ações do personagem

O monitoramento do jogo é feito durante seu tempo de execução. A monitoração não é feita apenas quando o personagem faz um movimento, mas durante toda a execução do jogo. Isto porque os métodos de monitoração já implementam a ação designada para cada situação, ou seja, monitora e age imediatamente (se as condições forem satisfeitas). Caso contrário, o inimigo só iria se movimentar quando o personagem se movimentasse.

O cenário do jogo é baseado em um plano cartesiano, pois é um jogo 2D, ou seja, possui apenas duas dimensões. O início da contagem das coordenadas x e y se

inicia na região superior esquerda. A figura abaixo mostra as coordenadas dos personagens quando o jogo é iniciado.



**Figura 39. Coordenadas dos personagens.**

Note que todos os personagens possuem uma sombra. As coordenadas apresentadas, na verdade, são as coordenadas da sombra dos personagens. A sombra é o alvo da monitoração do agente, porque a sombra é que vai determinar se o personagem está, ou não, em uma determinada área. Considere as duas situações que são apresentadas na figura:



**Figura 40. Situações.**

Vale ressaltar que em ambas as situações o corpo do personagem se encontra dentro da área inimiga, entretanto, não quer dizer que ocorre invasão nas duas situações. A invasão só acontece quando a sombra do personagem, de fato, estiver dentro da área inimiga, pois é a sombra que determina se os pés do personagem estão presentes no território inimigo. Esse problema se deve ao fato do jogo ser 2D e, também, ao formato do cenário. Portanto, para resolver esse pequeno impasse, foi criado o elemento sombra com o intuito de aumentar o nível de detalhamento do jogo.

### **7.3 Situações de perseguição**

O propósito do jogo é fazer com que o inimigo inicie uma perseguição em direção ao personagem quando este invadir seu território. Como foi dito na seção 7.1 (ver fluxo básico dos eventos), o personagem invade a área inimiga e o inimigo inicia a perseguição. Esse é o fluxo básico para a iniciação de uma perseguição.

Na seção 6, onde é feita uma explicação dos territórios, foi dito que o personagem, em uma única situação, não poderia andar livremente na área neutra. Essa situação ocorre quando o personagem retorna a sua área livre após estar sendo perseguido pelo inimigo. Nesse momento o inimigo se encontra na área neutra, pois leva um tempo para ele retornar a seu território. Durante esse tempo, a área neutra passa a ser um território hostil, pois, o inimigo entra em um estado de alerta que o faz

monitorar esta área. Portanto, se, o personagem resolver sair de sua área livre nesse intervalo de tempo, o inimigo iniciará outra perseguição. Não será uma nova perseguição porque o inimigo não retornou a sua área. Os estados do inimigo são:

- Neutro: quando ele está em seu território, esperando a invasão do personagem.
- Em perseguição: quando ele está perseguindo o personagem.
- Em alerta: quando ele está retornando para seu território e se encontra na área neutra.

Esses estados não existem de forma explícita no código do agente, apesar de, ao ver o jogo em execução, ser possível perceber alguma coisa referente ao estado do inimigo. É só uma forma de melhor explicar o monitoramento e as ações tomadas.

#### 7.4 Estrutura interna da aplicação

Durante o desenvolvimento de aplicativos para aparelhos móveis, deve-se ter um cuidado em relação à quantidade de classes (quantidade de classes, processamento desnecessário em métodos, etc) e recursos (imagens, sons, etc). Para a criação desse jogo foram criadas três classes, cada uma com sua particularidade. A figura abaixo apresenta a estrutura interna do jogo:



Figura 41. Estrutura interna do jogo.

- *PrincipalMidlet*: classe responsável por rodar o aplicativo no aparelho. Em outras palavras, um *Midlet* é a aplicação de celular feita em Java. Nessa classe é criada uma instância da classe *Jogo*.
- *Jogo*: classe responsável pela toda parte lógica da aplicação. Essa classe possui todos os métodos e atributos responsáveis pelo funcionamento do jogo e do agente inteligente. É interessante ressaltar que, essa classe, é uma extensão da classe *GameCanvas*, que é uma classe voltada para a criação de jogos em Java ME. Por fim, essa classe faz o uso dos recursos indexados pela classe *Engine*.
- *Engine*: a classe *Engine* é responsável por fazer as importações dos recursos usados na classe *Jogo*. Os personagens e as camadas xadrez são feitas baseadas em imagens.
- *Recursos*: é o pacote de itens usados para criar o jogo (apenas imagens, pois o jogo não possui som).

A criação do agente é baseada em métodos (de monitoração e ação) e atributos (de controle) que, juntos, são responsáveis pela monitoração das ações do personagem e pela tomada de decisão. O principal método do agente se chama “*monitorarEagir()*”, pois é responsável por perceber as alterações no ambiente e tomar decisões a partir disso. Nesse método são chamados métodos auxiliares que retornam valores (*true* ou

*false*) para saber em qual área os dois personagens estão, pois, assim, ações poderão ser tomadas. Além dos métodos, existe uma variável chamada “invadiu” (do tipo *boolean*) que recebe *true*, quando o personagem invade a área do inimigo, ou *false*, quando o jogo é iniciado e quando uma perseguição chega ao fim.

## **8 Considerações Finais**

A criação de jogos é uma tarefa que exige tempo e dedicação para ser executada com sucesso. Além disso, é preciso se ter conhecimento, matemático e de programação. Mais do que isso, é preciso ter criatividade para que seja feito algo que sirva de fonte de entretenimento. A inteligência artificial pode ser usada para que sejam criados jogos que sejam mais fiéis a realidade, no que envolve sua parte lógica, refletindo, assim, na experiência do jogador.

Um agente inteligente foi criado e usado neste trabalho para “dar vida” a um NPC, possibilitando que este personagem pudesse perceber alterações no ambiente e, conseqüentemente, agir de acordo com essas percepções. A técnica não apresentou alto nível de complexidade em sua implementação, pois, além do problema (aqui apresentado) ser simples, não existe uma forma de implementação a ser seguida.

O protótipo não foi testado em aparelhos celulares, apenas em emuladores do pacote de desenvolvimento. Porém, na continuidade do trabalho, o mesmo deverá ser testado e será necessário definir os requisitos mínimos dos aparelhos, assim como a versão do Java, resolução da tela, etc.

O trabalho terá continuidade no sentido de aumentar o número as ações dos personagens, adicionando novas habilidades. Será considerada a possibilidade do aumento da área do jogo, melhorando o cenário e adicionando novos elementos, possibilitando o estudo de outros recursos disponíveis pela linguagem.

## **9 Referências Bibliográficas**

- HOLEC, Henri. *Autonomy in Foreign Language Learning*. Oxford: Pergamon, 1981.
- RUSSEL, S., NORVIG, P. “*Inteligência Artificial*”. Rio de Janeiro: Elsevier. 2004.
- SILVA, F. C. L. *Agentes Inteligentes em Jogos de Computador*. [online] Disponível na Internet via WWW. URL: <http://www.ime.usp.br/~leliane/MAC5701/2005-1oSem/PlanosMonografias/Filipe.pdf>. Arquivo capturado em 2 de junho de 2009.
- SCHWAB, Brian. *AI Game Engine Programming*. Hingham: Charles River Media. 2004.
- WINSTON, P. H. *Artificial Intelligence*. Addison-Wesley (Third edition), 1992.



# UTILIZAÇÃO DO *CARD SORTING* NA IMPLEMENTAÇÃO DE UM SISTEMA DE *BALANCED SCORECARD* PARA INSTITUIÇÕES DE ENSINO SUPERIOR

Rafael Silvestre de Souza Vilela<sup>1</sup>, Katia Moraes de Carvalho<sup>1</sup>, Fabiano Fagundes<sup>1</sup>

<sup>1</sup>Curso de Sistemas de Informação – Centro Universitário Luterano de Palmas  
(CEULP/ULBRA)

Caixa Postal 160 – 77.054-970 – Palmas – TO – Brasil

{rsv1983, anjokatia, thilfa} @gmail.com

**Resumo.** *Este artigo tem por objetivo apresentar a utilização da técnica de Card Sorting ao avaliar a usabilidade durante o desenvolvimento de uma proposta para um sistema de Balanced Scorecard voltado para Instituições de Ensino Superior e que se baseará nos conceitos de Arquitetura de Informação para planejar, organizar e estruturar a informação.*

## 1. Introdução

As Instituições de Ensino Superior (IES) buscam uma melhora significativa nos cursos que são oferecidos para a sociedade, para que isso seja um diferencial em relação às demais instituições. Contudo, apenas elaborar projetos pedagógicos modernos e eficientes, grades curriculares atualizadas, planos estratégicos superficiais para consolidação dos resultados e auto-avaliações, não está sendo satisfatório, devido à grande dificuldade para colocá-los em prática.

Com o afloramento da necessidade de várias empresas, tanto do setor acadêmico quanto de outros setores da cadeia produtiva, em tirar do papel os seus planos estratégicos, foi proposto por Robert S.Kaplan, David P.Norton e com auxílio de várias empresas, o *Balanced Scorecard* (BSC), metodologia que traduz a missão e a visão das empresas em um conjunto abrangente de medidas de desempenho que serve de base para um sistema de medição e gestão estratégica. Esta metodologia fornece um novo referencial para a descrição da estratégia mediante a conexão de ativos tangíveis e intangíveis em atividades criadoras de valor.

Somente ter as informações através do BSC acessando por meio de um *website* não é o suficiente; é necessário organizar e estruturar a informação para que os *stakeholders* possam entender com maior clareza o conteúdo oferecido. Para tornar a informação do *scorecard* neste caso mais compreensível são utilizados os conceitos de Arquitetura de Informação e, para desenvolver arquiteturas baseadas na forma como as pessoas classificam e agrupam as informações, utilizou-se o método de *Card Sorting*.

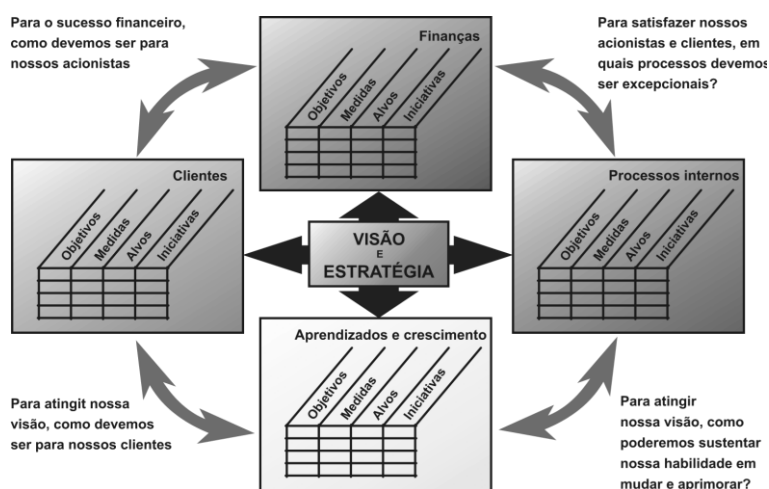
## 2. *Balanced Scorecard*

O *Balanced Scorecard* foi criado em 1990 através de um estudo idealizado e patrocinado pelo instituto Nolan Norton e realizado, tendo como líder do estudo David Norton e como consultor acadêmico Robert Kaplan, junto a diversas empresas (KAPLAN & NORTON, 1997, p.7), motivado principalmente pela observação da forma arcaica com que o desempenho empresarial estava sendo avaliado, apoiado

basicamente por indicadores contábeis e financeiros. Após analisar e discordar deste cenário, foi proposta uma nova metodologia de gestão empresarial com o intuito de desenvolver um modelo de medição de desempenho que pudesse equilibrar os objetivos de curto e longo prazo, entre medidas financeiras e não-financeiras, entre indicadores de tendência e ocorrências e entre as perspectivas internas e externas de desempenho (KAPLAN & NORTON, 1997, p.7).

Com o auxílio desta metodologia procura-se realizar a tradução da visão e da missão da organização para um conjunto abrangente de medidas de desempenho, que servirão de base para um sistema de medição e gestão da estratégia. O BSC preserva, ainda, os indicadores financeiros e contábeis tradicionais, permitindo conciliar com os indicadores necessários futuros, e realiza a medição do desempenho organizacional através de quatro perspectivas equilibradas (ilustradas na figura 1):

- **Financeiro:** perspectiva que realiza a demonstração da execução da estratégia para constatar se está contribuindo para a melhoria dos resultados financeiros e contábeis;
- **Do cliente:** perspectiva que avalia se a proposição de valor da empresa para os acionistas e clientes está produzindo os resultados angariados em termos de satisfação, conquista de novos clientes, retenção de clientes, lucratividade de clientes e participação de mercado da empresa;
- **Dos processos internos:** perspectiva que identifica se os principais processos de negócio definidos na cadeia de valor da empresa estão contribuindo para a geração de valor percebido pelos clientes, que façam com que estes clientes sejam atraídos e retidos nos segmentos-alvo de mercado, e na forma de atingir os objetivos financeiros que irão satisfazer os acionistas;
- **Do aprendizado e do crescimento:** esta perspectiva verifica se a aprendizagem, a obtenção de novos conhecimentos e o domínio de competências no nível do indivíduo, do grupo e das áreas de negócio está desempenhando o papel de viabilizadores das três perspectivas acima, utilizando-se preferencialmente de três fontes principais: pessoas, sistemas e procedimento organizacionais.



**Figura 42 - A estrutura do *Balanced Scorecard* oferece a tradução da estratégia em termos operacionais (BATISTELLA, 2002)**



### 3. Arquitetura de Informação

Atualmente as informações que transitam pelos *websites* e *softwares* estão crescendo vertiginosamente, porém muitos usuários se queixam da dificuldade na navegabilidade através dos hipertextos devido à grande quantidade de informação. Para ajustar ambientes conflituosos como este, é utilizada a Arquitetura de Informação, Iraque objetiva planejar, organizar e estruturar a informação para torná-la mais compreensível ao usuário.

O *Information Architecture Institute* é uma organização empenhada em difundir e promover a prática da Arquitetura de Informações e a define como (INFORMATION ARCHITECTURE INSTITUTE, 2009, *online*): uma concepção estrutural de ambientes com informações compartilhadas; a arte e a ciência de organizar e rotular *websites*, *intranets*, comunidades *online* e *softwares* de apoio à usabilidade e a localização; uma prática emergente focada em trazer princípios da arquitetura para o meio digital.

É perceptível a existência de vários conceitos sobre Arquitetura de Informação, devido aos aspectos que a englobam, como Informação; Estruturação; Organização e Rotulação; Facilidade para encontrar informações; e Arte e Ciência. Segundo Souza (2008) para melhorar a compreensão sobre AI é necessário entender alguns conceitos:

- **Informação:** a gerência dos dados e do conhecimento é um dos importantes fatores na Arquitetura de Informação. No processo da AI, a informação representa tanto a entrada quanto a saída sendo que a diferença está em como a mesma é trabalhada tendo como foco o objetivo de torná-la clara e mais acessível aos usuários;
- **Estruturação, Organização e Rotulação:** estruturação é o que os arquitetos fazem de forma geral, ou seja, definir níveis e espaços para determinados contextos e relacioná-los entre si. Organizar envolve agrupar aqueles componentes em categorias significativas e distintas. Rotular representa as categorias e a série de ligações a elas associadas;
- **Facilidade de encontrar informação:** este é um fator crítico para que se obtenha mais usabilidade, independente do sistema. A principal idéia é que o design de sistemas deve auxiliar as pessoas a encontrarem o que desejam;
- **Arte e ciência:** algumas disciplinas como Engenharia da Usabilidade estão auxiliando a trazer um método científico para a análise das necessidades dos usuários e para o processo de busca das informações. A prática da Arquitetura de Informação deve considerar fatores como experiência, intuição e criatividade, mesmo existindo outras metodologias de apoio ao desenvolvimento de *websites*, *interfaces* e *softwares*.

#### 3.1. Componentes da Arquitetura de Informação

Segundo Souza (2008) a Arquitetura de Informação encontra-se dividida em quatro sistemas independentes:

- **Sistema de Organização:** define como a informação é agrupada, podendo ser organizada através dos esquemas Geográfico, Alfabético e Cronológico;
- **Sistema de Navegação:** define a navegabilidade, através de *links* e hipertextos.

- **Sistema de Rotulação:** estabelece a forma como a informação será apresentada, ou seja, o rótulo dado a uma classe de determinado conjunto com características semelhantes;
- **Sistema de Busca:** permite realizar uma busca por meio de palavras-chave e expressões, e assim obter respostas.

A figura 2 mostra a estrutura de um *website* e os sistemas que compõem a Arquitetura de Informação.



Figura 43 - Página de um *website* contendo os sistemas de Arquitetura de Informação

## 4. Técnica para Avaliação da Usabilidade

Após apresentar os conceitos dos quatro sistemas que formulam a arquitetura de informação do *website*, o próximo passo consiste em avaliar estes sistemas para verificar a necessidade de implementá-los. A avaliação destes conceitos envolve a resposta a três perguntas (REIS, 2007, p.141): atende as necessidades de negócio? é viável técnica e operacionalmente? é fácil de usar?

As respostas quanto às duas primeiras perguntas são obtidas com os profissionais da empresa envolvidos na fase de Pesquisa<sup>1</sup> do projeto. Como na maioria das vezes são estes os profissionais que definem a proposta de valor do *site*, os requisitos e as diretrizes para implementação, então são os mais capacitados para avaliar se a arquitetura da informação proposta atenderá as expectativas de negócio e se é viável técnica e operacionalmente (REIS, 2007, p.141).

Quanto à terceira pergunta, “é fácil de usar?”, é sugerido que sejam realizadas avaliações de usabilidade utilizando técnicas da Interação Humano-Computador (REIS, 2007, p.141). Sendo assim, será apresentada a técnica de *Card Sorting* para avaliar a usabilidade utilizada em projetos de Arquitetura de Informação.

### 4.1. Card Sorting

*Card Sorting* é um método de teste com usuários que visa explorar a forma como eles agrupam uma série de itens. Este método consiste na entrega ao usuário de uma pilha de

<sup>1</sup> É uma fase da proposta de metodologia de projetos definida por Rosenfeld e Morville (SOUZA, 2008, p.46) cujo objetivo é capturar o contexto e as informações para observar as necessidades tanto da empresa quanto do público-alvo que a empresa (cliente) deseja atingir.

cartões, em que cada cartão representa um conteúdo do *website*, e é solicitado ao usuário para que os organize.

A sua finalidade é gerar a estrutura global da informação ou para avaliar uma estrutura que já existe; sendo assim, a técnica é indicada para ajudar no projeto dos sistemas de organização e rotulação. De acordo com Reis (2007), esta técnica pode ser usada para definição destes sistemas quanto também para validá-los.

Essa técnica é bastante eficiente e eficaz para se testar com usuários as propostas de *design* de *interface* computacional. Entretanto, não é apropriada para identificar as necessidades, os comportamentos e as informações que o usuário necessita, por ter o foco na avaliação de uso da *interface*.

*Card Sorting* é também um método barato e confiável, pois encontra padrões na forma como os usuários desejam encontrar as informações ou as funcionalidades do *site*. Esses padrões dos usuários são frequentemente referenciados como modelos mentais. Segundo Spencer e Warfel (2004), ao compreender os usuários através do modelo mental é possível aumentar a localização do que se procura que por sua vez, torna o produto mais fácil de usar.

#### 4.1.1. Utilizando-se do *Card Sorting*

Recomenda-se a utilização do *Card Sorting* quando o objetivo for: desenvolver um novo *website*; desenvolver uma nova área em um *website*; ou redesenhar um *website*.

O *Card Sorting* não é uma avaliação técnica, não vai mostrar o que está errado no *website* e não cria estrutura de informação. Na verdade, ele complementa outras atividades, tais como: análise das informações, análise das tarefas envolvidas no processo e avaliação contínua da usabilidade. E se torna mais eficaz quando concluída (SPENCER & WARFEL, 2004, *online*):

- a pesquisa sobre o que os usuários precisam fora do *site*;
- o conteúdo da auditoria (funcionalidade), inventário de *sites* existentes ou uma lista detalhada sobre um novo *site*. Com relação a um *site* existente, é imprescindível que o conteúdo do inventário seja examinado com atenção, para somente incluir o conteúdo necessário aos usuários.

Com isto, constata-se que o *Card Sorting* representa um facilitador na identificação da nomenclatura mais adequada para determinados grupos de informação, porém se torna desafiador para se usado contra alguns tipos de informações devido a complexidade e a variedade de agrupamentos em alguns casos.

#### 4.1.2. Métodos do *Card Sorting*

Existem dois métodos principais para a realização do *Card Sorting*:

- **Abertura do *Card Sorting*:** são dadas as cartas aos participantes mostrando o conteúdo do *website* sem nenhum agrupamento pré-estabelecido. Então, os participantes são convidados a classificar os cartões em grupos e depois são definidos os grupos. A abertura do *Card Sorting* é útil quando novas estruturas de informação são criadas e em estruturas de *sites* já existentes;
- **Fechamento do *Card Sorting*:** são dadas cartas aos participantes mostrando o conteúdo do *site* criado e também um conjunto inicial de grupos. Os participantes são convidados a relacionar estes cartões com os grupos. O fechamento do *Card*

*Sorting* é útil quando for adicionado um novo conteúdo a uma estrutura já existente ou para obtenção de comentários adicionais após a Abertura do *Card Sorting*.

Para este sistema foi adotado o método de Abertura, pois os usuários serão os responsáveis pela criação das categorias e também porque proporciona mais identificação do público-alvo com os grupos de informações disponíveis no *website*.

## 5. Modelo de Análise de Resultados do *Card Sorting*

Após a aplicar este método, para análise dos resultados foi utilizado um modelo de planilha que possui o propósito de auxiliar na geração da taxonomia final do *website* e também facilitar a análise dos testes aplicados com os usuários.

Este modelo de planilha foi customizado pelo consultor americano de Arquitetura de Informação Joe Lamantia (LAMANTIA, 2003, *online*). Através deste modelo é possível visualizar: a relação das categorias em que cada cartão aparece, a quantidade de vezes que este cartão aparece para cada categoria e o percentual de cada cartão em cada categoria; conforme será apresentado na próxima seção juntamente com o passo a passo necessário para utilização desta planilha.

## 6. Aplicação do *Card Sorting*

Para aplicar o *Card Sorting* foram escolhidos os requisitos funcionais como sendo os cartões a serem agrupados pelos usuários, conforme demonstrado na tabela 1, pois serão estas as funcionalidades que os usuários acessarão via sistema.

**Tabela 5 - Requisitos Funcionais**

Requisito	Funcionalidade	Requisito	Funcionalidade
<b>Usuários</b>	Cadastrar Usuário;	<b>Indicadores</b>	Cadastrar Indicador;
	Buscar Usuário;		Buscar Indicador;
	Alterar Usuário;		Associar Indicador;
	Excluir Usuário; e		Alterar Indicador;
	Visualizar Usuário.		Excluir Indicador; e Visualizar Indicador.
<b>Organizações</b>	Cadastrar Organização;	<b>Mapa Estratégico</b>	Cadastrar Mapa Estratégico;
	Buscar Organização;		Buscar Mapa Estratégico;
	Alterar Organização;		Alterar Mapa Estratégico;
	Excluir Organização; e		Excluir Mapa Estratégico;
	Visualizar Organização.		Visualizar Mapa Estratégico;
			Associar Objetivo;
			Desassociar Objetivo;
			Associar Indicador;
			Desassociar Indicador;
			Associar Iniciativa; e
	Desassociar Iniciativa.		
<b>Missão e Visão</b>	Cadastrar Missão e Visão;	<b>Coletando Dados</b>	Coletar Dados.
	Buscar Missão e Visão;		

	Alterar Missão e Visão;		
	Excluir Missão e Visão; e		
	Visualizar Missão e Visão.		
<b>Perspectivas</b>	Cadastrar Perspectiva;	<b>Desvios de Desempenho</b>	Registrar Desvio de Desempenho.
	Buscar Perspectiva;		
	Alterar Perspectiva;		
	Excluir Perspectiva; e		
	Visualizar Perspectiva.		
<b>Objetivos</b>	Cadastrar Objetivo;	<b>Desempenho</b>	Monitorar Desempenho.
	Buscar Objetivo;		
	Alterar Objetivo;		
	Excluir Objetivo; e		
	Visualizar Objetivo.		

Após a definição de que os requisitos funcionais representam o conteúdo do sistema de BSC, desta forma foi iniciado o processo de criação dos cartões, escrevendo os nomes de cada item em cartões de cor branca, que no caso representa o conteúdo a ser categorizado, conforme demonstrado na figura 3.

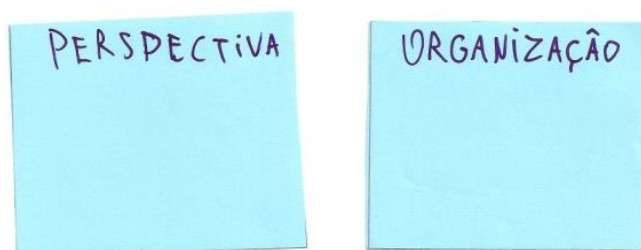


**Figura 44 - Exemplos dos Cartões Criados**

Com isso, o *Card Sorting* foi aplicado em quatro possíveis usuários desta proposta de sistema de *Balanced Scorecard* destinado às IES. E com a criação de um cartão para cada funcionalidade, foram então criados 45 cartões que possuíam a cor branca, os quais antes de serem entregues para o teste foram embaralhados. Devido à escolha do método de Abertura, de acordo com o informado na seção Métodos do *Card Sorting*, não foi estabelecido inicialmente nenhum agrupamento.

Depois de aplicado o *Card Sorting* com estes usuários percebeu-se que foram criados nove grupos, representados pelos cartões de cor azul conforme figura 4, sendo que os itens foram agrupados do mesmo modo por todos. E divergindo somente no item de Coleta de Dados, pois um usuário achou que este deveria pertencer ao grupo

Integrações, enquanto que os outros três acharam que não haveria categoria que se encaixasse com este item. Enfim, como estes três usuários não realizaram o agrupamento da funcionalidade Coleta Dados, esta informação foi colocada no único grupo criado para a mesma.



**Figura 45 - Categorias Criadas pelos Usuários**

Com a finalização da aplicação do *Card Sorting*, foi necessário traduzir para uma leitura mais adequada o resultado deste teste, por isso foi utilizado o modelo de Lamantia (2003), cujo primeiro passo é inserir os cartões e o número sequencial que foi colocado atrás de cada cartão, conforme demonstrado na tabela 2.

**Tabela 6 - Cartões e seus Números**

Título do Cartão	Número do Cartão	Título do Cartão	Número do Cartão
Cadastrar Usuário	1	Excluir Objetivo	24
Buscar Usuário	2	Visualizar Objetivo	25
Alterar Usuário	3	Cadastrar Indicador	26
Excluir Usuário	4	Buscar Indicador	27
Visualizar Usuário	5	Associar Indicador	28
Cadastrar Organização	6	Alterar Indicador	29
Buscar Organização	7	Excluir Indicador	30
Alterar Organização	8	Visualizar Indicador	31
Excluir Organização	9	Cadastrar Mapa Estratégico	32
Visualizar Organização	10	Buscar Mapa Estratégico	33
Cadastrar Missão e Visão	11	Alterar Mapa Estratégico	34
Buscar Missão e Visão	12	Excluir Mapa Estratégico	35
Alterar Missão e Visão	13	Visualizar Mapa Estratégico	36
Excluir Missão e Visão	14	Associar Objetivo	37
Visualizar Missão e Visão	15	Desassociar Objetivo	38
Cadastrar Perspectiva	16	Associar Indicador	39
Buscar Perspectiva	17	Desassociar Indicador	40
Alterar Perspectiva	18	Associar Iniciativa	41
Excluir Perspectiva	19	Desassociar Iniciativa	42
Visualizar Perspectiva	20	Coletar Dados	43
Cadastrar Objetivo	21	Registrar Desvio de Desempenho	44
Buscar Objetivo	22	Monitorar Desempenho	45
Alterar Objetivo	23		

Devido à utilização do método Abertura foi necessário padronizar as categorias criadas pelos usuários, pois existiu um caso em que o nome da categoria criada pelos quatro usuários divergia, entretanto possuía a mesma finalidade. Com esta padronização foi possível seguir até o segundo passo do modelo: inserção das categorias nas colunas e inserir o código de cada cartão de acordo com a categoria informada pelo usuário, conforme a tabela 3, que demonstra, a título de ilustração, os dados de dois usuários.

**Tabela 7 - Categorias X Cartões**

Usuário	Organização	Missão e Visão	Objetivo	Iniciativa	Indicador	Perspectiva	Mapa Estratégico	Desempenho	Integração
1	6	11	16	21	26	32	29	44	43
2	7	12	17	22	27	33	37	45	
3	8	13	18	23	28	34	38		
4	9	14	19	24	30	35	39		
5	10	15	20	25	31	36	40		
							41		
							42		
1	6	11	16	21	26	32	37	44	
2	7	12	17	22	27	33	38	45	
3	8	13	18	23	28	34	39		
4	9	14	19	24	29	35	40		
5	10	15	20	25	30	36	41		
					31		42		

Com a categorização dos cartões e auxiliado por este modelo de análise desta técnica de teste proposto por Lamantia (2003) foi atingido o objetivo do *Card Sorting*, que é auxiliar no desenvolvimento da arquitetura baseada na forma como o público-alvo categoriza os assuntos, conforme ilustrado na tabela 4. Nesta estrutura percebe-se que todos menos o cartão 29 possuem somente uma categoria, isto ocorreu porque quatro usuários classificaram este cartão para duas categorias distintas.

**Tabela 8 - Estrutura Global da Informação**

Título do Cartão	Número do Cartão	# Categorias Com Este Cartão	Média Por Categoria	Título do Cartão	Número do Cartão	# Categorias Com Este Cartão	Média Por Categoria
Cadastrar Usuário	1	1	100%	Excluir Objetivo	24	1	100%
Buscar Usuário	2	1	100%	Visualizar Objetivo	25	1	100%
Alterar Usuário	3	1	100%	Cadastrar Indicador	26	1	100%
Excluir Usuário	4	1	100%	Buscar Indicador	27	1	100%
Visualizar Usuário	5	1	100%	Associar Iniciativa	28	1	100%

Cadastrar Organização	6	1	100%	Alterar Indicador	29	2	50%
Buscar Organização	7	1	100%	Excluir Indicador	30	1	100%
Alterar Organização	8	1	100%	Visualizar Indicador	31	1	100%
Excluir Organização	9	1	100%	Cadastrar Mapa Estratégico	32	1	100%
Visualizar Organização	10	1	100%	Buscar Mapa Estratégico	33	1	100%
Cadastrar Missão e Visão	11	1	100%	Alterar Mapa Estratégico	34	1	100%
Buscar Missão e Visão	12	1	100%	Excluir Mapa Estratégico	35	1	100%
Alterar Missão e Visão	13	1	100%	Visualizar Mapa Estratégico	36	1	100%
Excluir Missão e Visão	14	1	100%	Associar Objetivo	37	1	100%
Visualizar Missão e Visão	15	1	100%	Desassociar Objetivo	38	1	100%
Cadastrar Perspectiva	16	1	100%	Associar Indicador	39	1	100%
Buscar Perspectiva	17	1	100%	Desassociar Indicador	40	1	100%
Alterar Perspectiva	18	1	100%	Associar Iniciativa	41	1	100%
Excluir Perspectiva	19	1	100%	Desassociar Iniciativa	42	1	100%
Visualizar Perspectiva	20	1	100%	Coletar Dados	43	1	100%
Cadastrar Objetivo	21	1	100%	Registrar Desvio de Desempenho	44	1	100%
Buscar Objetivo	22	1	100%	Monitorar Desempenho	45	1	100%
Alterar Objetivo	23	1	100%				

## 7. Considerações Finais

Através da aplicação do método de *Card Sorting* foi possível avaliar a usabilidade utilizada no desenvolvimento da proposta de um sistema de *Balanced Scorecard*. E constatar que a criação das categorias e o agrupamento destas foram praticamente idênticos nos testes realizados com todos os usuários, isto ocorreu porque existe uma semelhança nos nomes das funcionalidades dos requisitos funcionais.

Quanto à contribuição deste teste na arquitetura do sistema de *Balanced Scorecard*, esta foi decisiva neste quesito, pois auxiliou na criação da estrutura organizacional que agora está direcionada de acordo com o modelo mental do público-alvo. No que diz respeito à Arquitetura de Informação, o *Card Sorting* também norteou os sistemas de Organização e Rotulação.

Desta forma, é possível dizer que o que era esperado com a aplicação do referido método foi atingido, porque foram criadas as categorias e conseqüentemente a forma de



agrupamento dos cartões nestes grupos. Porém, mesmo atingindo os principais objetivos um item poderia ter sido mais bem trabalhado, que foi a parte de explicação do sistema que passou pelo *Card Sorting*. Isto ocorreu porque não foram apresentados, durante as seções, documentos que auxiliariam na explanação do sistema e no entendimento do usuário, o que acarretou em dúvidas triviais que poderiam ter sido mais bem explicadas se existisse no momento uma ilustração. Este problema origina-se sobretudo pela ausência de usuários contumazes da ferramenta BSC para serem alvo da aplicação do *Card Sorting*.

Por fim, com a disseminação da Arquitetura de Informação é necessário entender o acesso intuitivo a informação dos usuários, e com a utilização do *Card Sorting* contribui-se para entender este acesso e também cooperar na criação de estruturas organizacionais otimizadas e direcionadas ao modelo do usuário.

## 8. Referências Bibliográficas

BATISTELLA, João Paulo G., **Balanced Scorecard como Ferramenta Gerencial da Estratégia**. 2002. Conceptia Ltda. Disponível em: <<http://www.conceptia.com.br/artigos/artigo-2002-03-27.html>>. Acesso em 29 mar. 2008.

*Information Architecture Institute, Learning Information Architecture*. Disponível em: <<http://iainstitute.org/en/>>. Acesso em: 15 mai. 2009.

KAPLAN, Robert S.; NORTON, David P., **A Estratégia em Ação: Balanced Scorecard**. Rio de Janeiro: Elsevier, 1997.

LAMANTIA, Joe, **Analyzing Card Sort Results with a Spreadsheet Template**. Publicado em: 26 ago. 2003. Disponível em: <[http://www.boxesandarrows.com/view/analyzing\\_card\\_sort\\_results\\_with\\_a\\_spreadsheet\\_template](http://www.boxesandarrows.com/view/analyzing_card_sort_results_with_a_spreadsheet_template)>. Acesso em 22 set.2009.

REIS, Guilherme Almeida dos. **Centrando a Arquitetura de Informação no usuário**. 2007. 250f. Monografia (Mestrado em Ciência da Informação) – ECA/USP, São Paulo.

SOUZA, Igor F., **Arquitetura da Informação na Criação de um Portal Acadêmico**. 2008. Dissertação (Graduação em Sistemas de Informação). Centro Universitário Luterano de Palmas, Palmas/TO.

SPENCER, Donna; WARFEL, Todd. **Card Sorting: a definitive guide**. Publicado em 07 abr. 2004. Disponível em: <[http://www.boxesandarrows.com/view/card\\_sorting\\_a\\_definitive\\_guide](http://www.boxesandarrows.com/view/card_sorting_a_definitive_guide)>. Acesso em 23 mai.2009.



## Índice Onomástico