

Revista

INICIA

Revista de Iniciação Científica da FAI – Centro de Ensino Superior em Gestão,
Tecnologia e Educação.

Publicação anual

Número 6, Ano 2006

FAI – Centro de Superior em Gestão, Tecnologia e Educação

Santa Rita do Sapucaí – MG – Brasil

Outubro de 2006

Revista Inicia, n. 6

Santa Rita do Sapucaí: FAI – Centro de Ensino Superior em
Gestão, Tecnologia e Educação, 2006

Anual

ISSN 1806-8278

1. Administração. 2. Informática. 3. Educação

FAI – Centro de Ensino Superior em Gestão, Tecnologia e
Educação.

EXPEDIENTE

Revista de Iniciação Científica de Informática, Administração e Educação da FAI
Ano 06 - Número 2006
ISSN 1806-8278

Diretor

Prof. Aldo Ambrósio Morelli

Conselho Editorial

Prof^ª. Adicinéia Aparecida Oliveira
Prof. Benedito Márcio B. Magalhães
Prof^ª. Eunice Gomes de Siqueira
Prof. Fábio Gavião Avelino de Mello
Prof. José Cláudio Pereira
Prof^ª. Valéria Santos Paduan Silva

Jornalista Responsável

Ana Maria Beraldo - Mtb MG 05054 JP

Assessora de Marketing

Prof^ª. Sandra Carvalho dos Santos

Bibliotecária

Elen Maria Ferreira Terra - CRB6/1890

Projeto Gráfico

Prof^ª. Adicinéia Aparecida Oliveira

Tiragem

Impressa - 300 exemplares
CD ROM - 75 exemplares

Endereço para correspondência

Faculdade de Administração e Informática
Av. Antônio de Cássia, 472
Jardim Santo Antônio
Cep: 37.540-000
Santa Rita do Sapucaí - MG
Brasil
E-mail: inicia@fai-mg.br

Fundação Educandário Santarritense

Presidente: Pe. Vonilton Augusto Ferreira

Vice-Presidente: Prof. Benedito Márcio Barbosa Magalhães

1º Tesoureiro: Prof. João Teles de Souza

2ª Tesoureira: Profª Fátima Cecília Seguro de Carvalho

1ª Secretária: Profª Maria Luiza Campos do Amaral Moreira

2º Secretário: Prof. José Cláudio Pereira

FAI – Faculdade de Administração e Informática

Diretor: Prof. Aldo Ambrósio Morelli

Vice-Diretor: Prof. Benedito Márcio Barbosa Magalhães

Colégio Tecnológico Delfim Moreira

Diretora: Profª Maria Luiza Campos do Amaral Moreira

Vice-Diretora: Maria Aparecida Cássia de Gruiter

Editorial

Como parte da comemoração pelos 35 anos da Faculdade de Administração e Informática (FAI), apresentamos a sexta edição da Revista INICIA, que é um veículo exclusivamente dedicado à publicação de artigos de alunos dos cursos de graduação da FAI e do Instituto Superior de Educação (ISE).

Esta edição da Revista INICIA contém quatro artigos que são fruto do trabalho de bolsistas e orientadores, participantes do Programa de Bolsas de Iniciação Científica e Tecnológica Institucional da FAI e FAPEMIG. Os demais artigos foram desenvolvidos como atividade discente de disciplinas específicas dos seus respectivos cursos. Completam esta edição, os resumos dos projetos de conclusão de curso dos alunos de Ciência da Computação de 2005.

Gostaríamos de convidar todos os membros da comunidade da FAI e do ISE para contribuírem com o crescimento da Revista INICIA, seja como autores, leitores ou colaboradores.

Entrem em contato conosco e participem!

Boa leitura a todos.

Conselho Editorial

SUMÁRIO

Artigos	Pág.
Programa de boas práticas de fabricação para indústrias de alimentos Hiordan Moura Cardoso e Durval Teixeira	8
Resolução de problemas de suporte à decisão utilizando um pacote comercial de programação linear Mário Luiz Costa de Andrade, Herick Ferreira Souza e Fábio Gavião A. de Mello	13
Um laboratório virtual de algoritmos e estruturas de dados Vanice Ribeiro Pinto e Roberto de Souza Porto	18
Alan turing e o modelo de máquina universal Soraia Garcia Rosa Botelho e Eunice Gomes de Siqueira	23
Uma configuração do Linux e samba como controlador primário de domínio para clientes Windows Rogério Tomassoni de Araújo Junior e Flávio Neri Rodrigues	27
A utilização da Biblioteca Allegro para desenvolvimento de aplicações multimídia Luana Pereira de Lima, e Túlio Vitor Machado Faria	33
Introdução à plataforma Java Micro Edition: configurações e perfis Alexandre Ferraz Luz e Karl Pascher Wolf	38
Resumos de projetos de conclusão de curso de Sistemas de Informação de 2005	
A abordagem de projeto centrado no uso André Luís Alves Dias	44
Engworks printers driver for Windows Carol Gaudino Palma, Fabrício Fraga Ferreira, Márcia Cristina da Silva, Rafael Noronha Tavares Gomes e Tânia Noberto Couto	45
GAIA - Sistema de Gestão Ambiental Alisson Lopes, Carlos Eduardo Moreira de Mello Borges, Gisela de Lima Coelho, Gisele Anézio Felipe e Gustavo Pires dos Santos	46
Sistema de controle de arquivos multimídia para web: SCAM2WEB Ednardo David Segura, Juarez Monteiro Villela Vitor, Leandro Gomes Silveira, Pedro de Faria Costa e Ruhelinton Perre de Lima	47

Sistema de software para apoio ao método de avaliação scampi: e-web scampi	47
Camillo Alessandro Camillo, Eduardo do Carmo Lopes, Pedro Henrique Begalli Rezende, Rodrigo Duarte Beraldo e Ronaldo Vitorino Vilela	
Sistema de controle laboratorial - SISLAB	48
Sandro Retori Pinto	
System Mobile	49
Carolina Pereira Mângia, Christiane Martins Costa, Daiani Vaz Moreira, Mariana Pereira Osório e Paula Junqueira Paiva	

PROGRAMA DE BOAS PRÁTICAS DE FABRICAÇÃO PARA INDÚSTRIAS DE ALIMENTOS

Hiordan Moura Cardoso

Bolsista de Iniciação Científica da FAPEMIG
hiordan@hotmail.com

Prof. Durval Teixeira

Orientador
professordurval@uol.com.br

Faculdade de Administração e Informática

Resumo - Este artigo tem o objetivo de apresentar um programa de qualidade que auxilie as pequenas e médias empresas de alimentos no atendimento das normas de Boas Práticas de Fabricação (BPF) da Agência Nacional de Vigilância Sanitária (ANVISA). Tais normas visam assegurar que produtos alimentícios saudáveis sejam produzidos.

Abstract - This work has the objective to introduce a quality program that helps small and medium business of goods follow the norms of Good Manufacturing Practices (GMP) of Agência Nacional de Vigilância Sanitária (ANVISA). Such norms aim to assure that healthy food products are produced.

Palavras-chave - Qualidade, ISO 9000:2000, Boas práticas de fabricação.

Keywords - Quality, ISO 9000:2000, Good Manufacturing Practices.

1. HISTÓRIA DAS BOAS PRÁTICAS DE FABRICAÇÃO

As Boas Práticas de Fabricação (BPF) é um termo originado do Inglês *Good Manufacturing Practices (GMP)*, usado para designar um conjunto de ações e critérios que objetiva, especialmente, assegurar a qualidade de produtos e serviços que lidam diretamente com a manipulação de alimentos ou produtos farmacêuticos (ANVISA, 2006).

As BPF surgiram durante um congresso americano realizado em 1938, porém, a primeira regulamentação foi publicada apenas em 1969, sendo direcionada aos produtos alimentícios, apresentando a importância de se criar uma legislação específica para o setor.

No Brasil, as BPF são estabelecidas por Leis e Portarias da Agência Nacional de Vigilância Sanitária (ANVISA), que envolvem os produtos e processos de domissanitários, produtos de higiene pessoal, cosméticos, alimentícios e farmacêuticos. São elas:

- Lei 8.078/90 - Código de Defesa do Consumidor;
- Portaria 58 – BPF para as indústrias de sementes;
- Portaria 348 – BPF para as indústrias produtoras de artigos de higiene pessoal, cosméticos e perfumes;
- Portaria 326 – BPF para as indústrias produtoras de alimentos – objeto deste trabalho;
- Portaria 134 – BPF para as indústrias produtoras de medicamentos.

2. O PROGRAMA BPF - FAI

Com base nas portarias e regulamentações estabelecidas pela ANVISA, o Centro de Desenvolvimento e Pesquisa (CDP) da Faculdade de Administração e Informática (FAI) resolveu desenvolver um programa de qualidade que prestasse apoio às empresas alimentícias, no atendimento dessas

normas, e auxiliassem-nas no controle e garantia da qualidade. Ele foi intitulado de Programa de Boas Práticas de Fabricação para indústrias de alimentos da FAI (BPF-FAI).

As normas ISO 9000:2000 definem qualidade como sendo: “Grau no qual um conjunto de características inerentes satisfaz a requisitos”. As definições da ISO são extremamente úteis para o desenvolvimento deste programa que, apesar de ser extremamente prático, exige fundamentos teóricos.

Segundo as normas da ISO 9000:2000 existe uma diferença clara entre controle da qualidade e garantia da qualidade. Controle da qualidade é definido como sendo: “Parte da gestão da qualidade focada no atendimento dos requisitos da qualidade”. Ela também define garantia da qualidade como sendo: “Parte da gestão da qualidade focada em prover confiança de que os requisitos da qualidade serão atendidos”.

Baseados na ferramenta ISO 9000:2000 e nas portarias da ANVISA, desenvolveu-se um programa de qualidade único que, além de promover o atendimento às normas das BPF, promove o desenvolvimento interno, melhora as condições de trabalho e auxilia na consolidação de uma cultura voltada para a qualiprodutividade.

O programa BPF-FAI é constituído pelos módulos apresentados a seguir.

2.1 Palestra sobre a importância das Boas Práticas de Fabricação

Infelizmente, muitos pequenos e médios empresários do ramo de alimentos desconhecem as normas e portarias das BPF. Por isso, criou-se um módulo exclusivo para apresentar essas portarias e para apresentar os benefícios do programa BPF-FAI.

2.2 Treinamento para o repasse de conceitos sobre o programa 5s

Neste segundo módulo, são repassados os conceitos sobre o programa de qualidade 5s para colaboradores da empresa ou para um comitê da qualidade em formação.

O programa 5s é um método prático e simples que tem o objetivo de obter um local de trabalho ordenado, limpo e saudável, constituindo uma estrutura ideal para a implantação de outros programas de qualidade mais complexos. O termo 5s se origina de palavras do idioma japonês que começam com a letra "S". Cada senso é explicado a seguir (TEIXEIRA, 2005).

Seiri – o primeiro senso, traduzindo genericamente, significa descarte do que não apresenta utilização. As vantagens oferecidas por esse senso são:

- liberação de equipamentos e documentos inúteis;
- redução dos tempos de procura;
- melhor visualização do local de trabalho.

Seiton – o segundo senso, traduzindo genericamente, significa ordenação e organização. As vantagens oferecidas por esse senso são:

- rapidez e facilidade na busca de documentos e objetos;
- redução da perda de tempo;
- eliminação de desperdícios;
- controle do uso de equipamentos e documentos necessários;
- facilidade de comunicação;
- evita rupturas organizacionais;
- reduz o risco de acidentes.

Seiso – o terceiro senso, traduzindo genericamente, significa limpeza. As vantagens oferecidas por esse senso são:

- higiene no local de trabalho;
- eliminação de fontes geradoras de sujeiras.

Seiketsu – o quarto senso, traduzindo genericamente, significa higiene. As vantagens oferecidas por esse senso são:

- melhoria do ambiente de trabalho;
- melhoria das áreas comuns;
- condições de trabalho favoráveis à saúde.

Shitsuke – o quinto senso, traduzindo genericamente, significa disciplina. As vantagens oferecidas por esse senso são:

- cumprimento natural dos procedimentos;
- disciplina moral e ética;
- cultivo de bons hábitos;

- efetivação da administração participativa;
- garantia da qualidade de vida.

2.3 Treinamento para formação do Comitê da Qualidade e formatação da lista de checagem

Neste módulo são apresentados os conceitos para formatação da lista de checagem (ou *check list*) e para auxiliar as empresas que não possuem um comitê da qualidade a formarem um.

É importante a existência de um comitê de qualidade em organizações, independentemente de seu tamanho, pois ele é o responsável pela melhoria de processos e pela melhoria contínua.

Sua posição na estrutura organizacional geralmente encontra-se em *Staff* do Diretor Geral Executivo.

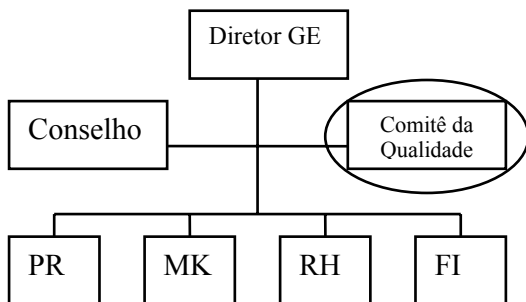


Figura 1. Posição do Comitê da Qualidade

O comitê da qualidade pode ser traduzido em um grupo de estudos avançados para melhoria. Deve possuir um representante de cada setor da empresa a fim de formar um grupo homogêneo.

O comitê tem um papel fundamental no uso de ferramentas de melhoria, como o *Brainstorming*.

2.4 Treinamento para lançamento do dia “D”

Neste módulo são repassados todos os conceitos necessários para o lançamento do dia “D” e toda expectativa que deve ser criada, assim como as regras que deverão ser seguidas.

No dia “D”, a empresa se une com um único objetivo, limpeza. Os diretores,

gerentes e todos os funcionários trabalham unidos, utilizando todos os conceitos do programa 5s. Assim que uma área da empresa está completamente limpa e arrumada, os responsáveis pelo setor se deslocam para ajudar os amigos de trabalho que ainda não terminaram a sua área. No final do dia, toda a empresa estará unida em um único setor terminando a limpeza. Sobre o dia “D”:

- é o dia oficial do lançamento do programa 5s na empresa;
- deve ter a participação de todos;
- poderá receber outros nomes, variando conforme a criatividade de cada empresa;
- poderá ser feito em vários dias, mas o ideal é que seja feito num único dia, para fixar melhor o conceito;
- recomenda-se uma atividade de descontração (final festivo) para o resgate de conceitos, troca de experiências, registro de depoimentos, etc;
- seu maior objetivo é vivenciar, na prática, as teorias do programa 5s e fixar melhor os seus conceitos básicos.

O dia “D” traz a união para dentro da empresa, faz com que os funcionários criem um sentimento de dependência, o que ajuda na mudança para uma cultura de melhoria contínua e de qualidade.

2.5 Treinamento para formação de auditores para o programa 5s e a auditoria

A existência de auditorias é essencial para a manutenção de qualquer programa de qualidade. Elas garantem que as normas do programa estão sendo seguidas (GIL, 1994). Este módulo prepara os auditores do programa 5s para usar de forma correta a lista de checagem.

No início do programa de auditorias são executadas três auditorias de treinamento, com objetivo de preparar os funcionários e ajustar a lista de checagem. Os modelos de auditoria usados são apresentados a seguir:

- auditoria interna programada: esse modelo de auditoria possui data e hora para ser executada, e os auditores são

membros da organização a ser auditada;

- auditoria interna não-programada: esse modelo de auditoria também possui data e hora para ocorrer, porém essas informações não são de conhecimento da empresa, apenas do grupo de auditores da organização;
- auditoria externa programada: a organização é auditada por grupos especializados que não fazem parte da empresa;
- auditoria externa não programada: a organização é auditada por grupos especializados que não fazem parte da empresa e acontece em data indeterminada.

Geralmente, nas primeiras auditorias ocorrerá uma diferença entre a auditoria programada e a não programada. Isso é natural, pois a cultura da qualidade ainda não está consolidada. Com o passar do tempo, essa diferença diminui apresentando, assim, um amadurecimento do programa (TEIXEIRA, 2005).

2.6 Treinamento para o repasse de conceitos sobre as BPF

Este módulo consiste no repasse das normas de BPF para o comitê de qualidade. Ele abrange os seguintes assuntos:

- Introdução
- Boas Práticas de Fabricação
- História das BPF
- Objetivos das BPF
- Legislação
- Principais definições
- Controle, garantia e gerenciamento da qualidade
- Controle da qualidade
- Garantia da qualidade
- Estrutura da documentação
- Materiais, processo e produtos acabados
- Materiais
- Procedimento para inspeção e recebimento
- Identificação
- Controle de não conformidades

- Instalações da inspeção de recebimentos
- Armazenamento
- Processo
- Procedimentos de processo
- Pessoal
- Higiene pessoal
- Controle de qualidade durante o processo
- Ações corretivas e preventivas
- Equipamentos
- Instalações e ambiente
- Limpeza e sanitização
- Objetivo da higienização
- Fatores básicos para a higienização
- Etapas dos procedimentos de higienização
- Processo de higienização
- Monitoramento da higiene e sanitização
- Requisitos para a implantação dos procedimentos de higienização
- Causas possíveis dos fracassos dos procedimentos de higienização
- Controle de pragas
- Produto acabado
- Controle de qualidade do produto acabado
- Rastreabilidade
- Controle de não conformes
- Armazenamento
- Abrangência das BPF
- Monitoramento das BPF
- Auditorias de manutenção
- Reclamação de consumidores
- *Recall*

2.7 Treinamento para a consolidação de Procedimentos Operacionais Padrão-POPs

Procedimentos operacionais padrão são um conjunto de informações organizadas, constituídas pelo conhecimento organizacional a fim de otimizar operações e processos. É de fundamental importância para a análise e melhorias de processos, conduzindo a empresa, de forma natural, ao caminho da qualidade total (GALVÃO e MENDONÇA, 1997; TEIXEIRA, 2005).

Neste módulo são repassados todos os conceitos sobre padronização de procedimentos, processos, documentos e operações para que o comitê da qualidade trabalhe nas melhorias.

2.8 Treinamento para reciclagem de auditores para o programa BPF

Neste módulo é realizado o treinamento dos auditores para a adaptação da auditoria do programa 5s para auditoria de BPF.

3. CONCLUSÃO

O programa BPF-FAI passou por testes em empresas alimentícias da região do sul de Minas Gerais, com eventuais ajustes. Os resultados do programa foram satisfatórios e as melhorias no ambiente fabril, escritórios, estoques de materiais são visíveis e sentidos por todos da organização.

A cultura das empresas mudou e o conceito de qualidade está na mente de todos. O comprometimento pessoal de cada um nas empresas tornou-se um exemplo.

No início do processo de implantação de um programa de qualidade, como o BPF-FAI, encontra-se resistência a mudanças. O simples fato de que os colaboradores devem usar toca de proteção e uniforme para garantia da não-contaminação de alimentos já traz um transtorno fora do comum. Mas, com os métodos corretos, é possível tornar essa resistência em comprometimento e garantir a qualidade dos produtos produzidos e, principalmente, a satisfação e a saúde dos consumidores.

AGRADECIMENTOS

Agradeço ao excepcional mestre, professor e orientador, Durval Teixeira, responsável pela metodologia Programa de Desenvolvimento Organizacional e Gestão pela Qualidade Total com foco para

Administração por Resultado (PDO/GQT-APR), o apoio e conhecimento que vem me transmitindo ao longo desses anos. Agradeço a bolsa de iniciação científica concedida pela FAPEMIG, que está propiciando a realização deste trabalho.

Hiordan

REFERÊNCIAS BIBLIOGRÁFICAS

AGÊNCIA NACIONAL DE VIGILÂNCIA SANITÁRIA - ANVISA. **Boas práticas de fabricação – BPF**. 2006. Disponível em: <http://www.anvisa.gov.br/alimentos/bpf.htm>

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO 9000:2000 - Sistema de gestão da qualidade – fundamentos e vocabulários. Rio de Janeiro, 2000.

GALVÃO, Célio; MENDONÇA, Mauro. **Fazendo acontecer na Qualidade Total**. 1 ed. Rio de Janeiro: Ed. Qualymark, 1997.

GIL, Antônio de Loureno. **Auditoria da Qualidade**. 1 ed. São Paulo: Ed. Atlas,1994.

TEIXEIRA, Durval. **Programa 5s serviços**. Pouso Alegre, 2005.

RESOLUÇÃO DE PROBLEMAS DE SUPORTE À DECISÃO UTILIZANDO UM PACOTE COMERCIAL DE PROGRAMAÇÃO LINEAR

Mário Luiz Costa de Andrade

Bolsista de Iniciação Científica da FAPEMIG
lecomario@hotmail.com

Herick Ferreira Souza

Bolsista de Iniciação Científica da FAPEMIG
herickf@yahoo.com.br

Prof. Fábio Gavião Avelino de Mello

Orientador
gavião_tsi@uol.com.br

Faculdade de Administração e Informática

Resumo – Este artigo mostra como formular problemas de suporte à decisão e como resolvê-los através de pacotes de otimização comerciais os quais utilizam técnicas de Programação Linear e de Otimização Combinatória embutidas. O início dos estudos nesta área e sua importância para a construção de sistemas de suporte à decisão também são apresentados. O processo completo é ilustrado através de um exemplo conhecido como o “dilema do atleta” que começa pela análise do problema, passa pela fase de criação de um modelo matemático e é resolvido por intermédio do pacote de software comercial XPRESS da Dash Optimization.

Abstract – Abstract – The primary purpose of this work is to present readers on how to make models and solve decision support systems using commercial packages that employ linear programming and combinatorial optimization. A brief historical perspective is presented in the beginning of this work. The whole process is illustrated using a very simple and known problem called the Athlete Dilemma. First, a description and analysis of this problem is presented following a step-by-step sequence of building its mathematical model. Finally, it is showed how to solve the problem using XPRESS package from Dash Optimization..

Palavras-chave – Otimização Combinatória, Programação Linear, Pesquisa Operacional.

Keywords – Combinatorial Optimization, Linear Programming, Operational Research.

1. INTRODUÇÃO

Alguns países, no decorrer da Segunda Grande Guerra Mundial, necessitaram de soluções para a melhoria de seus armamentos e das suas técnicas operacionais militares, que propiciaram um grande impulso nas técnicas de programação linear (PINA, 2006).

Em 1947, George B. Dantzig pesquisou e desenvolveu uma técnica de solução para resolver problemas de Pesquisa Operacional, batizando-a de “método *Simplex*”. Este método seqüencial de otimização pode ser empregado tanto para problemas de maximização como de minimização. Dantzig é considerado o “Pai da Programação Linear”.

Os algoritmos de programação linear são poderosos e muito utilizados em otimização, além de serem ideais para quem necessita de “otimilidade” comprovada.

A programação linear tem como objetivo maximizar ou minimizar uma função definida sobre certo domínio finito de valores. Em geral, ela se resume em encontrar dentre todos os valores possíveis aqueles que

satisfaçam a função pedida, de modo a maximizar ou minimizar um valor.

Um sistema de suporte à decisão visa apoiar a tomada de decisão especialmente no que diz respeito ao tratamento de grandes números de variáveis envolvidas no problema. Como exemplo de uso, pode-se citar o caso da Universidade do Chile que desenvolveu um desses sistemas para a JUNAEB, empresa que faz a entrega de merenda escolar para todas as escolas públicas do Chile. Com o sistema, houve uma economia de US\$ 40 milhões por ano, que depois pôde ser revertida para a melhoria da qualidade da merenda (ILOG, 2006).

2. IDENTIFICAÇÃO DO PROBLEMA

Segundo Goldberg, Pacca e Luna (2005) para demonstrar como funciona a otimização combinatoria pode-se usar o exemplo conhecido como o “dilema do atleta”.

Um atleta se sente atraído pela prática de dois esportes: natação e ciclismo. Sabe-se, por experiência, que a natação exige um gasto de mensalidades do clube e de deslocamento até a piscina que pode ser expresso em um custo médio de R\$ 3,00 por seção de treinamento de até 1 hora. Já o ciclismo, mais simples, acaba custando cerca de R\$ 2,00 por seção de treinamento de até 2 horas.

O orçamento do atleta restringe seus gastos em treinamento em R\$ 70,00 no total. Seus afazeres de aluno de graduação na universidade lhe dão liberdade de empregar, no máximo, 30 horas mensais para o treinamento e 80.000 calorias para os esforços físicos. Sabe-se que cada seção de natação consome, em média, 1.500 calorias, enquanto cada etapa ciclística despende 1.000 calorias. Considerando que o atleta não tem predileção por nenhum dos dois esportes em particular, o problema consiste em descobrir quantas seções dos dois esportes, o atleta deverá praticar de forma a maximizar a soma das seções.

3. ANÁLISE DOS DADOS

O exemplo dado refere-se a um modelo de programação de atividades, que

pode ser resolvido utilizando a programação linear. No exemplo, o objetivo é maximizar o número de seções totais de treinamento envolvendo ambas as atividades.

O primeiro passo para a solução de um problema de suporte à decisão é o conhecimento dos dados e como eles se relacionam. Veja o Quadro 1 com os dados do exemplo.

	Natação	Ciclismo	Disponibilidade
Custo	R\$ 3,00	R\$ 2,00	R\$ 70,00
Calorias	1.500 cal	1.000 cal	80.000 cal
Tempo	1 h	2 h	30 h

Quadro 1. Valores individuais e restrições de custo, calorias e tempo por modalidade

3.1. Formulação das matrizes do problema

O próximo passo é a colocação dos dados num formato matricial adequado (GOLDBARG, PACCA e LUNA, 2005).

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{33} \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 1500 & 1000 \\ 1 & 2 \end{bmatrix}$$

Figura 1. Matrizes operacionais

No lado esquerdo da Figura 1 vê-se a representação literal da matriz operacional do problema e no lado direito os respectivos valores de cada variável. A primeira linha é construída com o custo de cada modalidade (R\$3,00 e R\$2,00). A segunda linha com as calorias (1500 cal e 1000 cal) de cada uma. A terceira linha contém o tempo em horas por cada seção de treinamento em cada modalidade (1h e 2h).

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Figura 2. Matrizes de variáveis de decisão

Na Figura 2 vêem-se as incógnitas do problema que são variáveis representando o número de seções de 2 horas por modalidade que deve fazer parte da solução final. Na primeira linha se coloca a variável referente ao número de seções de natação e na segunda

linha a variável referente à modalidade ciclismo.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 70 \\ 80.000 \\ 30 \end{bmatrix}$$

Figura 3. Matrizes de restrições do problema

A Figura 3 mostra a matriz de restrições do problema. Na primeira linha, coloca-se o gasto máximo permitido pelo atleta (R\$70,00). Na segunda linha, o máximo de calorias que o atleta pretende gastar no total (80.000 cal). E, na terceira linha, o tempo disponível em horas para o treinamento (30 h).

4. FORMULAÇÃO MATEMÁTICA DO PROBLEMA

4.1 Escolha das variáveis de decisão

A escolha das variáveis de decisão é importante, pois é ela que mostrará o número máximo de seções que o atleta vai fazer em cada modalidade. Para tal, escolhe-se a variável x_1 para representar o número de seções de natação e x_2 para representar o número de seções de ciclismo.

4.2 Elaboração da função objetivo

A função objetivo tem como finalidade mostrar o resultado máximo ou mínimo de uma determinada função. Por exemplo, maximizar o lucro de uma empresa ou minimizar seus custos.

Para obter o máximo de seções de treinamento, é necessário somar o número de seções a serem praticadas. Para isso, usa-se a seguinte função objetivo:

$$Z = \text{Max } \{f(x) = x_1 + x_2\}$$

4.3 Formulação das restrições tecnológicas (equações de restrição)

Conforme foi visto no problema, existem algumas restrições que precisam ser obedecidas. São elas:

a) equação associada aos recursos financeiros do atleta:

$$3,00x_1 + 2,00x_2 \leq 70,00$$

onde: 3 é o gasto que o atleta tem com natação; 2 é o gasto que ele tem com o ciclismo; x_1 e x_2 são o número de seções de cada modalidade. A soma tem que ser menor ou igual a R\$70,00 que é o orçamento disponível pelo atleta.

b) equação associada à disponibilidade de energia:

$$1.500x_1 + 1.000x_2 \leq 80.000$$

onde: 1.500 é o total de calorias despendida pelo atleta por seção de natação; 1.000 calorias é a energia despendida pelo atleta por seção de ciclismo; estes valores são multiplicados pelos números de seções de cada esporte e não podem ultrapassar o total de calorias disponível que é de 80.000 cal.

c) equação associada à disponibilidade de tempo:

$$x_1 + 2x_2 \leq 30$$

sabe-se que cada seção de natação exige 1 hora de treino e que cada seção de ciclismo exige 2 horas de treino, assim deve-se multiplicar esses valores pela quantidade de seções de cada esporte encontradas no final e a soma deve respeitar o valor máximo de 30 horas.

d) restrição de não negatividade:

tal restrição serve para que os valores das incógnitas nunca sejam negativos, pois não há números de seções negativas.

$$x_1 \geq 0, x_2 \geq 0$$

5. MODELO MATEMÁTICO FINAL

É possível formular uma expressão genérica de todas as fórmulas.

$$\sum_{j=1}^2 a_{ij} \cdot x_j \leq b_i \quad (i=1,\dots,3)$$

Essa fórmula geral é construída com base nas restrições impostas pelo exemplo. Ela pode ser representada com as matrizes anteriores, como mostra a Figura 4.

$$\begin{bmatrix} 3,00 & 2,00 \\ 1500 & 1000 \\ 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 70,00 \\ 80.000 \\ 30 \end{bmatrix}$$

Figura 4. Matrizes com a fórmula geral

6. RESOLUÇÃO

Como os problemas de otimização combinatoria possuem muitas variáveis, torna-se inviável a resolução desse tipo de problema de maneira manual. Para isso, podem ser utilizados pacotes comerciais de software, como o Xpress (DASH, 2006) ou o CPLEX (ILOG, 2006). Neste artigo, embora o número de variáveis seja pequeno e o método de resolução fácil de se implementar, é mostrado o uso de um destes pacotes comerciais, no caso o pacote Xpress.

O primeiro passo para utilizar um pacote comercial qualquer é transformar o modelo matemático do problema em algo que o software entenda. O Xpress possui uma linguagem de programação de modelos denominada Mosel. Basta escrever o modelo matemático em Mosel para que o Xpress entenda e otimize-o.

Para se criar o código na linguagem Mosel é preciso seguir alguns passos. Primeiro, deve-se criar o nome do modelo: *model* será o indicativo do nome do modelo e *end-model* vai finalizar o programa. Todas as outras instruções ficarão dentro desse bloco.

model Atleta

·
·
end-model

Em seguida, coloca-se a palavra *uses*, que mostrará qual biblioteca de otimização o programa vai usar. Neste caso, o programa

usará somente a biblioteca *mmxprs*, referente a Programação Mista Inteira.

uses "mmxprs"

Logo após, precisa-se criar as variáveis x_1 e x_2 . Para isto, coloca-se a palavra *declarations* que indica o início das declarações das variáveis, em seguida as variáveis escolhidas, no caso x_1 e x_2 , e o tipo é *mpvar* para informar que se trata de variáveis escalares comuns. Fecha-se esse bloco com a palavra *end-declarations*.

declarations

x1, x2: mpvar

end-declarations

Após declarar as variáveis, devem-se apresentar as restrições tecnológicas do problema. Agora, usam-se as fórmulas criadas anteriormente. O nome *Custo* é apenas um rótulo para o nome da primeira restrição, assim como *Tempo*, *Calorias*, *Negatividade1* e *Negatividade2*.

*Custo:= 3,00*x1 + 2,00*x2 <= 70,00*
*Tempo:= x1 + 2*x2 <= 30*
*Calorias:= 1500*x1 + 1000*x2 <= 80000*
negatividade:= x1 >= 0
negatividade1:= x2 >= 0

Deve-se informar que x_1 e x_2 são variáveis inteiras, de acordo com a formulação do problema, pois o atleta não poderá fazer uma parte de uma seção de treinamento.

x1 is_integer

x2 is_integer

O próximo passo é escrever a função objetivo que deverá maximizar o número de seções de treinamento do atleta. Para isso, utiliza-se a palavra *maximize*, que é um comando da linguagem Mosel. Em seguida escreve a função que, no caso, é $x_1 + x_2$.

maximize(x1 + x2)

Neste ponto da execução, o Xpress já tem a resposta de quantas seções o atleta

deverá fazer. Para mostrar o resultado em tela, utiliza-se o comando *write* ou *writeln*. O comando *getobjval* é utilizado para obter o valor final da função objetivo.

```
writeln("Resultados:\n")
writeln("Valor da função objetivo é: ",
getobjval)
```

O comando *getsol* é utilizado para acessar o valor que está contido dentro da variável entre os parênteses.

```
writeln("\nFaça ", getsol(x1), " seções de
natação e ", getsol(x2), " seções de ciclismo")
```

O exemplo da Figura 5 mostra o código implementado na linguagem Mosel.

```
model Atleta
uses "mmxprs"
declarations
  x1, x2: mpvar
end-declarations
Custo:= 3,00*x1 + 2,00*x2 <= 70,00
Tempo:= x1 + 2*x2<= 30
Calorias:= 1500*x1 + 1000*x2<= 80000
Negatividade1:= x1 >= 0
Negatividade2:= x2 >= 0
x1 is_integer
x2 is_integer
maximize( x1 + x2 )
writeln("Resultados:\n")
writeln("Valor da função objetivo é: ", getobjval)
writeln("\nFaça ", getsol(x1), " seções de natação e
", getsol(x2), " seções de ciclismo.")
end-model
```

Figura 5. Programa em Mosel

Concluído o programa em Mosel, o próximo passo é a execução do otimizador. Para tal, basta acionar o comando F5 na interface gráfica do Xpress. O resultado apresentado para o problema foi o de 20 seções de natação e 5 seções de ciclismo.

7. CONCLUSÃO

Este artigo apresentou um exemplo de um problema de suporte à decisão, onde um atleta estava indeciso entre quantas seções de ciclismo e de natação deveria fazer. Foi mostrado como modelar matematicamente o

problema e depois como implementar sua solução através do uso de um pacote de software comercial de otimização.

Os modelos de otimização expõem, em termos matemáticos, como resolver problemas da melhor forma possível, podendo assim reduzir gastos, aumentar lucro, melhorar a produtividade, o desempenho e a qualidade. Assim, as empresas podem se valer dos sistemas de suporte à decisão para obter uma vantagem competitiva no mercado.

AGRADECIMENTOS

Agradecemos a FAI por propiciar recursos para o desenvolvimento de nossas pesquisas. Agradecemos também a FAPEMIG pela oportunidade de estarmos aumentando nossos conhecimentos através da Iniciação Científica.

REFERÊNCIAS BIBLIOGRÁFICAS

DASH OPTIMIZATION. **Help Online do Xpress-IVE Student Licence**. 2006. Disponível em: <http://www.dashoptimization.com>. Acesso em: 08 jun 2006.

GOLDBARG, M. C.; PACCA, H.; LUNA H. P. **Otimização Combinatória e Programação Linear: Modelos e Algoritmos**. 2 ed. Rio de Janeiro: Editora Campus, 2005. p. 1–89.

ILOG. **Changing the rules of business**. 2006. Disponível em: <http://www.ilog.com/download/docs/SS-Univers.Chile2003.pdf>. Acesso em: 24 mai. 2006.

PINA, J. C. **Programação Linear**. 2006. Disponível em: <http://www.ime.usp.br/~coelho/mac315>. Acessado em: 24 abr. 2006.

UM LABORATÓRIO VIRTUAL DE ALGORITMOS E ESTRUTURAS DE DADOS

Vanice Ribeiro Pinto

Bolsista de Iniciação Científica da FAPEMIG
vaniceribeiro@yahoo.com.br

Prof. Roberto de Souza Porto

Orientador
rsporto@fai-mg.br

Faculdade de Administração e Informática

Resumo – Esse artigo apresenta o Laboratório Virtual de Algoritmos e Estruturas de Dados (LAED), uma ferramenta complementar ao estudo de algoritmos e estruturas de dados. O LAED é uma ferramenta que demonstra alguns conceitos e técnicas ministradas nas disciplinas de algoritmos e estruturas de dados dos cursos de Computação e Informática.

Abstract – This work introduces a Virtual Laboratory of Algorithms and Data Structures (LAED) as a complementary tool to the study of algorithms and data structures. LAED is a tool that approaches some subjects given in the disciplines of algorithms and data structure of the computing courses..

Palavras-chave – Algoritmos, estruturas de dados, ferramentas de aprendizagem.

Keywords – Algorithms, data structures, learning tools.

1. INTRODUÇÃO

O aumento da importância da informação nas organizações e o crescente avanço tecnológico requerem profissionais que sejam capazes de compreender e aplicar estratégias para buscar soluções eficazes e eficientes, aliadas às habilidades de criatividade, inovação e flexibilidade (SANTOS e COSTA, 2005)

Tais estratégias e habilidades podem ser desenvolvidas pela metodologia aplicada pelos docentes na formação dos futuros profissionais, em especial, nos cursos da área

de Computação e Informática, que exigem constante aperfeiçoamento e inovação.

Um fator importante que leva a propor uma inovação no processo metodológico de ensino, principalmente nas disciplinas que abordam algoritmos e estruturas de dados, é a dificuldade dos discentes iniciantes de discernir e compreender esses conceitos, que são considerados como complexos e abstratos para eles.

Assim, uma ferramenta computacional demonstrativa que ajude a transformar processos abstratos em concretos, procurando facilitar a compreensão dos tópicos ensinados, pode ser um bom recurso didático.

O Laboratório Virtual de Algoritmos e Estrutura de Dados (LAED) é uma ferramenta de demonstração desses conceitos, de forma interativa e animada, o que facilita a interação do aluno com o objeto de ensino – o computador.

Com ela, espera-se que o aluno saia da posição de observador passivo e comece a visualizar o leque de situações reais que explicam a necessidade do uso das técnicas de programação.

Este artigo está organizado da seguinte forma: a seção 2 apresenta a justificativa da construção de um software dessa modalidade; a seção 3 apresenta o LAED em termos de informações de desenvolvimento e temas abordados; a seção 4 relata resumidamente os projetos pesquisados que visam o ensino de algoritmos e estruturas de dados. Por fim, a seção 5 apresenta a conclusão.

2. JUSTIFICATIVAS

O conhecimento de técnicas de programação para o desenvolvimento de softwares é um requisito básico para os profissionais da área. Por isso, as disciplinas que ensinam tais técnicas são indispensáveis nos cursos de bacharelado da área de Computação e Informática (CELES, CERQUEIRA e RANGEL, 2004).

Segundo Rodrigues (2002, *apud* PETRY, 2005 p. 15) a reprovação na disciplina de Algoritmos varia de 40% a 50% dos alunos que a cursam em todo o Brasil. Segundo o autor, isto deve provocar uma reflexão por parte dos professores que estão preocupados com a qualidade no processo de ensino e aprendizagem dos alunos.

A didática utilizada pelo professor para apresentar o conteúdo introdutório da disciplina também é um fator fundamental no processo de ensino de algoritmos, pois para todo problema requer uma boa capacidade de abstração do aluno. O aluno, normalmente, tem dificuldade em entender os diversos e novos conceitos referentes à execução de tarefas pelo computador (RODRIGUES, 2002 *apud* PETRY, 2005, p. 15).

Os índices de reprovação revelam que realmente existem dificuldades no aprendizado de algoritmos e estruturas de dados. Por isso, os professores das disciplinas precisam trabalhar essas dificuldades e ajudar a desenvolver o raciocínio lógico e abstração nos alunos.

Uma das maiores dificuldades encontradas no processo de ensino e aprendizagem de algoritmos é entender o raciocínio desenvolvido pelos alunos. Mesmo em uma situação ideal, onde existe a figura de um mediador que possa acompanhar individualmente cada um deles, é difícil acompanhar cada etapa da solução apresentada. (KOSLOSKY, 1999 *apud* PETRY, 2005, p. 15).

Nos livros de algoritmos e estrutura de dados, os autores, além da linguagem em pseudocódigo, procuram detalhar e ilustrar com exemplos estáticos as teorias explicadas.

Esses modelos estáticos, representados por desenhos, vistos repetitivamente tornam-se pouco compreensíveis devido à falta de comunicação entre os objetos de ensino, no caso o livro didático, e o ensinado – o aluno (GARCIA, REZENDE e CALHEIROS, 1997).

É nesse contexto que o software demonstrativo sintoniza perfeitamente com o aprendiz. O software é dotado de cores e animações, o que enriquece o poder de comunicação e interação entre os dois.

Diante da oportunidade de unir a teoria à visualização gráfica dinâmica e interativa, o LAED vem contribuir para a melhoria das metodologias de ensino utilizadas no ambiente acadêmico.

O LAED utilizado como um recurso didático inserido dentro da metodologia do professor poderá auxiliar os alunos nos estudos dessas disciplinas.

3. O LAED

Para se chegar a uma aplicação que incorpore os recursos de visualização gráfica, dinâmica e interativa, faz-se necessária desenvolvê-la utilizando-se de uma linguagem de programação que seja rica em termos de componentes visuais e, de preferência, que seja portátil, isto é, que permita a execução em múltiplas plataformas.

A título de exemplo, uma pesquisa mostra que a grande maioria dos softwares desenvolvidos para esse fim e que estão disponíveis no mercado é operável em múltiplas plataformas. A linguagem utilizada por eles é Java, já que ela possui características que se encaixam no escopo pré-definido. A ferramenta mais utilizada é a Netbeans IDE 5 (SUN, 2006) por facilitar o uso de recursos visuais e dar suporte ao desenvolvimento na linguagem Java. Visto as características dessa poderosa ferramenta aliada a uma linguagem de programação que oferece excelentes recursos, foi determinada a sua utilização para a codificação do LAED.

A seguir, são apresentados os conceitos e técnicas que o aluno tem contato ao utilizar o LAED:

- algoritmos de ordenação: entende-se por ordenação, classificar dados em uma ordem particular, como crescente ou decrescente. São demonstrados seis algoritmos de ordenação: *Bubble Sort*, *Quick Sort*, *Merge Sort*, *Shell Sort*, *Insertion Sort* e *Selection Sort*;
- algoritmos de busca: envolve verificar se uma determinada chave de pesquisa encontra-se num conjunto determinado de dados. Os seguintes algoritmos de busca são apresentados: Busca Serial, Busca Seqüencial, Busca Binária, Busca Direta Indexada, Busca Direta Transformada e *Hashing*;
- listas encadeadas: coleções de registros (nós) encadeados;
- pilhas: definidas como estruturas do tipo *Last In, First Out (LIFO)*, cuja representação pode ser seqüencial ou encadeada;
- filas: definidas como estruturas do tipo *First In, First Out (FIFO)*, cuja representação pode ser seqüencial ou encadeada.
- árvores binárias: caracterizadas como estruturas compostas por nós que respeitam uma determinada hierarquia.
- grafos: são definidos, basicamente, como estruturas compostas por nós e linhas que representam suas relações.

Além das ilustrações interativas das técnicas, são mostrados exemplos práticos fazendo uso de determinadas especialidades algorítmicas. O LAED também disponibiliza as teorias referentes aos algoritmos e as estruturas de dados.

As Figuras 1 e 2 apresentam algumas telas do LAED.

4. PROJETOS RELACIONADOS

Nesta seção são apresentados alguns projetos existentes que influenciaram positivamente na construção do LAED, com os respectivos comentários de seus autores.

- Tutorial de Estrutura de dados (TED)

O sistema de Tutorial apresenta os algoritmos das estruturas de dados, ao mesmo tempo em que demonstra a sua implementação lógica através de exemplos visuais permitindo ao aluno acompanhar todos os passos simultaneamente ao que é executado (DELLAZZANA, 2004).

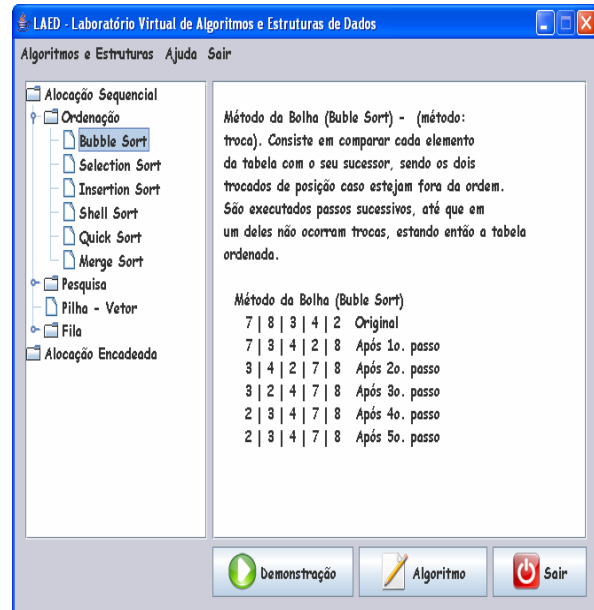


Figura 1. Tela principal

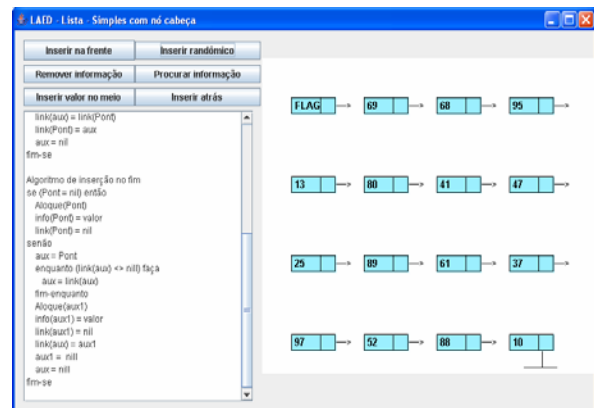


Figura 2. Tela de demonstração de lista simples com nó cabeça

- Estruturas de Dados Dinâmicas Lineares (EDDL)

A discussão que tem sido levada a cabo nos últimos anos acerca das estratégias de ensino das estruturas de dados e tipos de dados abstratos utilizados em programação influenciou a estruturação do programa EDDL e a concepção das suas principais funcionalidades, no sentido de serem tidas em conta aos seguintes componentes: 1) uma

abordagem introdutória aos conceitos ou especificação dos tipos de estruturas; 2) as técnicas de implementação; 3) a utilização dessas estruturas na resolução de problemas. O programa contempla esses três componentes e pode ser utilizado quer como ferramenta introdutória quer como ferramenta de revisão, exercitação e auto-avaliação. (AZUL E MENDES, 1998).

- *Animation of Data Structures and algorithms* (ASTRAL)

Diante da oportunidade de associar o ensino teórico de algoritmos e estruturas de dados a uma disciplina de laboratório de software, decidimos verificar se um acréscimo no nível de participação do aprendiz no processo de animação resultaria em ganho ainda maior na aprendizagem. Nossa tentativa visou incorporar à fase de implementação realizada pelos estudantes algumas chamadas a rotinas gráficas de alto nível para visualização tanto das estruturas de dados quanto das atuações dos algoritmos implementados para manipulação delas [...]. O processo de animação foi feito utilizando-se funções e procedimentos, sem alterações à linguagem ou utilização de pré-processadores. Em um nível acima do sistema operacional, temos uma camada que agrupa rotinas de interesse geral para a construção de animações que obedecem ao nosso modelo, denominada *camada básica*. Fazem parte desta camada o encapsulamento de tópicos como manipulação de janelas de texto, a visualização e a comunicação com o usuário da aplicação assim como o tratamento de eventos e o gerenciamento de objetos gráficos. (GARCIA, REZENDE e CALHEIROS, 1997).

- Treinamento Baseado em Computador – Algoritmo e Estruturas de dados (TBC-AED)

Este projeto está servindo como referência principal para a elaboração do LAED.

Quando ao desenvolvimento, o *TBC-AED* apresenta os seguintes temas: Busca em Vetor (Busca Binária), Métodos de Ordenação (*Select Sort*, *Insert Sort*, *Bubble Sort*, *Merge Sort* e *Quick Sort*), Estruturas de Alocação Estática e Dinâmica de Memória (Lista, Fila e Pilha) e Árvores (Árvore Binária de Busca) (SANTOS e COSTA, 2005).

Nas experiências realizadas com projetos já existentes como o ASTRAL e o TBC-AED, os resultados obtidos com o uso

das ferramentas pelos alunos foram positivos. Os alunos melhoraram o interesse e o desempenho durante as aulas. Como consequência, houve aumento no número de aprovações nas disciplinas de algoritmos e estrutura de dados, garantindo para o mercado profissionais melhores preparados (GARCIA, REZENDE e CALHEIROS, 1997; SANTOS e COSTA, 2005).

5. CONCLUSÃO

Neste artigo procurou-se mostrar as justificativas para se investir em ferramentas computacionais para fins educacionais na área de Computação e Informática, principalmente, para apoiar o ensino e a aprendizagem de algoritmos e estruturas de dados.

Dentro do escopo da questão de melhorias dos métodos aplicados para formação de profissionais capacitados, o LAED fica como contribuição.

O software LAED contém módulos agregados com uma interação simples, o que traz para o aprendiz maior usabilidade em sua utilização. Assim, espera-se que o LAED seja um recurso didático para os professores, de tal modo que promova mudanças que interfiram positivamente no entendimento e transmissão das técnicas de algoritmos e estrutura de dados, a fim de melhorar o desempenho dos discentes.

AGRADECIMENTOS

Agradecemos a FAPEMIG por fomentar a bolsa de estudo de iniciação científica e a FAI por proporcionar comodidade e tranquilidade no desenvolvimento do estudo.

REFERÊNCIAS BIBLIOGRÁFICAS

AZUL, A. A.; MENDES, A. J. EDDL: um programa didático sobre estruturas de dados dinâmicas lineares. In: SIMPÓSIO DE INVESTIGAÇÃO E DESENVOLVIMENTO DE SOFTWARE EDUCATIVO, n. 3, Portugal, 1998. **Anais**. Disponível em: <http://www.minerva.uevora.pt/simposio/>

comunicacoes/Mendes/Eddl.html. Acesso em: 02 ago 2006.

CELES, W.; CERQUEIRA, R.; RANGEL, J.L. **Introdução a Estrutura de Dados**: com técnicas de programação em C. Rio de Janeiro: Ed. Campus, 2004.

DELLAZZANA, F. **TED - Tutorial de Estruturas de Dados**. Universidade do Vale do Itajaí, 2004. Disponível em <http://www.tutorialdeestruturadedados.8m.com/>. Acesso em: 02 ago 2006.

GARCIA, I. C.; REZENDE, P. J.; CALHEIROS, F. C.. Astral: Um Ambiente para Ensino de Estruturas de Dados através de Animações de Algoritmos. **Revista Brasileira de Informática na Educação**, n. 01, 1997. Disponível em: <http://www.inf.ufsc.br/sbcie/revista/nr1/garcia.htm>. Acesso em: 02 ago 2006.

SANTOS, P. R.; COSTA, X. A. H. **TBC-AED e TBC-AED/WEB**: um desafio no ensino de Algoritmos, Estruturas de Dados e Programação. 2005. Disponível em: <http://www.weimig2005.unis.edu.br/artigos/TBC-AED.pdf>. Acesso em: 18 ago 2006.

PETRY, P. G. **Um sistema para o ensino e aprendizagem de algoritmos utilizando um companheiro de aprendizagem colaborativo**. Dissertação (mestrado) – Programa de pós-graduação em Ciência da Computação, UFSC, Florianópolis, 2005. Disponível em: <http://150.162.90.250/teses/PGCC0717.pdf>. Acesso em: 5 out 2006.

SUN MICROSYSTEMS. **NetBeans IDE 5.0**. 2006. Disponível em: <http://www.sun.com>. Acesso em: Jan. 2006.

ALAN TURING E O MODELO DE MÁQUINA UNIVERSAL

Soraia Garcia Rosa Botelho

Bolsista de Iniciação Científica da FAPEMIG
soraiaagr@yahoo.com.br

Profa. Eunice Gomes de Siqueira

Orientadora
eunice@fai-mg.br

Faculdade de Administração e Informática

Resumo - O legado de contribuições de Alan Turing para algumas das áreas da Ciência é extenso e fundamental. Este artigo apresenta um breve relato sobre a sua atuação na área de Computação e Informática, com destaque para o modelo de máquina de Turing.

Abstract - The legacy of contributions of Alan Turing to some fields of Science is large and fundamental. This paper introduces his Science Computer researches, and specially, the Turing Universal Machine.

Palavras-chave - Máquina universal, máquina de Turing, computação.

Keywords - universal machine, Turing Machine, computer science.

1. INTRODUÇÃO

Alan Mathison Turing nasceu em 23 de junho de 1912, em Londres, filho de Julius Mathison e Ethel Sara Turing.

Graduou-se em Matemática na Universidade de Cambridge, em 1934, continuando seus estudos na Universidade de Princeton (1936-1938), nos Estados Unidos, onde obteve seu PhD em lógica matemática, sob a orientação do professor americano Alonzo Church.

Embora tenha vivido pouco mais de 40 anos, Turing foi um dos mais brilhantes matemáticos de seu tempo e deixou contribuições fundamentais para vários outros campos da Ciência, como Lógica, Criptografia, Inteligência Artificial, Ciência da Computação e Biologia.

2. UM BREVE RELATO SOBRE A ATUAÇÃO DE TURING NA ÁREA DE COMPUTAÇÃO E INFORMÁTICA

Em 1936, então com 24 anos de idade, Turing consagrou-se como um dos maiores matemáticos de seu tempo, quando construiu “a conceituação matemática para a noção de algoritmos, segundo os passos que um ser humano dá quando executa um determinado cálculo” (CHAGAS, 2002).

No artigo “*On computable Numbers with an Application to the Entscheidungsproblem*”, Turing explicitamente baseado em uma metáfora de um “computador” – na época significava uma pessoa que realizava cálculos segundo um método efetivo – ocupado com operações previamente determinadas, idealizou uma máquina abstrata que poderia ser instruída para realizar qualquer tipo de cálculo. Era, portanto, o modelo de uma máquina de propósito geral ou universal que ficou conhecida com a máquina de Turing (mT).

Segundo Britton (1992, *apud* CHAGAS, 2002):

No momento, quase ninguém percebera que a descoberta descrita naquele obscuro artigo iria conduzir a uma arrancada mundial no campo tecnológico, embora seu jovem autor, Alan Mathison Turing, já estivesse na busca de uma máquina que simulasse os processos humanos de conhecimento.

A máquina de Turing teve importância fundamental no desenvolvimento das áreas de computabilidade, teoria dos autômatos formais e análise de algoritmos. A distinção entre hardware e software por meio do

conceito de máquina universal é considerada como um dos triunfos intelectuais do Século XX (BITTENCOURT, 2001; LOVE, 2004; SETZER, 2006). A Seção 3 deste artigo descreve alguns dos conceitos relacionados à mT.

Durante a Segunda Guerra Mundial (1939 - 1945), Turing trabalhou para o governo britânico, na Government Code and Cypher School (GC&CS), em Bletchley Park, Buckinghamshire, Inglaterra. Ali, ele ajudou a projetar máquinas eletromecânicas para decifrar as comunicações de rádio alemãs, cuja codificação era produzida por um sistema chamado de Enigma. Estima-se que em 1942, até 39.000 mensagens eram decodificadas a cada mês, graças às máquinas eletromecânicas (COPELAND, 2000).

Por algum tempo essas máquinas se mostraram eficientes, mas o alto comando alemão modificou o sistema de codificação de mensagens, tornando-as mais complexas e difíceis de serem decodificadas exigindo um sistema mais eficiente para esse trabalho. Em 1942, a necessidade de decifrar as mensagens de forma mais rápida possível, dada a situação de guerra, levou Turing e os demais cientistas da GC&CS a participarem do projeto do Colossus, o primeiro computador eletrônico e digital completamente funcional (ainda sem o conceito de programa armazenado). F.H. Hinsley, historiador oficial da GC&CS, estimou que a Segunda Guerra na Europa teve seu tempo reduzido em pelo menos em dois anos, graças à atuação dos cientistas e da máquina Colossus (COPELAND, 2000).

Em 1950, em seu famoso artigo "*Computing Machinery and Intelligence*", Turing fez previsões precisas sobre o que seria necessário para um computador se passar por um ser humano em uma conversa e lançava as bases para a Inteligência Artificial. Turing sugeriu que ao invés de se perguntar "se as máquinas poderiam pensar", a pergunta deveria ser "se elas poderiam ser aprovadas em um teste comportamental de inteligência". O "Teste de Turing", basicamente, consiste em fazer um ser humano, sem contato visual com o outro lado, descobrir se quem responde às suas perguntas, introduzidas via teclado, é uma outra pessoa

ou uma máquina. A máquina será aprovada no teste se pelas respostas, for impossível a alguém distinguir qual interlocutor é a máquina e qual é a pessoa (RUSSEL e NORVIG, 1995; BITTENCOURT, 2001).

Em 1951, Turing foi nomeado membro da Royal Society do Reino Unido, uma instituição destinada à promoção da excelência na Ciência.

Turing faleceu em Manchester, Inglaterra, no dia 7 de junho de 1954, vítima de suicídio, após comer uma maçã envenenada com cianureto (HODGES, 1992).

A Association for Computing Machinery (ACM), a partir de 1966 e a cada ano, concede o "Prêmio Turing" para uma pessoa selecionada por suas contribuições de natureza técnica feitas para a comunidade da Computação (ACM, 2006). Esse prêmio é conhecido como o "Prêmio Nobel da Computação". Alguns de seus ganhadores foram: Marvin Minsky (pelo trabalho com inteligência artificial, em 1969); Edgar Frank Codd (por sua fundamental contribuição para a teoria e prática dos sistemas gerenciadores de banco de dados, em 1981); Ole-Johan Dahl e Kristen Nygaard (idéias fundamentais para o surgimento da programação orientada ao objeto, por meio do projeto das linguagens de programação Simula I e Simula 67, em 2001).

3. MÁQUINA DE TURING (MT)

A mT é um mecanismo simples que formaliza a idéia da realização de cálculos, imitando o comportamento humano. Apesar de sua simplicidade, o modelo mT possui, no mínimo, o mesmo poder computacional de qualquer computador de propósito geral. Ela é abstrata porque se restringe apenas aos aspectos lógicos do seu funcionamento (memória, estados e transições) e não à sua implementação física (MENEZES, 2005).

A mT apresentada a seguir, é uma das várias variantes encontradas na literatura e está baseada em Gersting (2004) e Menezes (2005).

Uma máquina abstrata de Turing é constituída de três partes: fita, unidade de controle e função de transição (o programa).

a) Fita: usada simultaneamente como dispositivo de entrada, saída e memória de armazenamento. Sendo finita à esquerda e infinita à direita, dividida em células, cada uma armazenando um símbolo pertencente a um alfabeto ou o símbolo especial b (“branco”) ou marcador de início da fita.

b) Unidade de controle: reflete o estado corrente da máquina. Possui uma unidade de leitura e gravação (cabeça da fita), a qual acessa uma célula de cada vez e movimentase para a direita ou para a esquerda. Possui um número finito e pré-definido de estados.

c) Função de transição (ou programa): função que define o estado da máquina e comanda a leitura, gravação e o sentido do movimento da cabeça da fita. Dependendo do estado corrente da máquina e do símbolo lido, ela determina o símbolo a ser gravado, o sentido do movimento da cabeça da fita e o novo estado.

Podem-se descrever as ações de uma máquina particular através de um conjunto de quintuplas da forma (e, i, i', e', s) , onde:

- e indica o estado atual;
- i o símbolo que está sendo lido;
- i' denota o símbolo a ser gravado;
- e' denota o novo estado
- s denota o sentido do movimento da cabeça: E para a esquerda e D para a direita.

e	i	i'	e'	s
Estado atual	Símbolo lido	Símbolo gravado	Próximo estado	Movimento

A definição da mT é dada como segue (GERSTING, 2004):

sejam E um conjunto finito de estados e I um conjunto finito de símbolos para a fita (seu alfabeto), incluído o símbolo especial b . Uma máquina de Turing é um conjunto de quintuplas da forma (e, i, i', e', s) . $e, e' \in E$; $i, i' \in I$; $s \in \{D, E\}$ e duas quintuplas distintas nunca começam com os mesmos símbolos e, i .

A Tabela 1 apresenta um exemplo do funcionamento de uma mT que tem a função de agrupar os símbolos 1s submetidos como

entrada. A tabela tem 5 colunas que especificam: o estado em que a máquina se encontra (coluna 1); o símbolo corrente sob a cabeça de leitura/gravação da máquina (coluna 2); o símbolo a ser gravado $\{0,1\}$ (coluna 3); a transição para o próximo estado (coluna 4), que pode ser um estado final ou não; e finalmente a ação a ser realizada (coluna 5) mover a cabeça para a esquerda E ou mover a cabeça para a direita D .

O processamento da mT consiste na sucessiva aplicação da função de transição a partir do estado inicial e_0 e da cabeça posicionada mais à esquerda da fita até ocorrer uma condição de parada ou não.

TABELA 1. Um exemplo de mT

e	i	i'	e'	s
Estado atual	Símbolo lido	Símbolo gravado	Próximo estado	Movimento
0	0	0	1	D
0	1	1	0	D
1	0		Pára	
1	1	0	2	E
2	0	1	0	D
2	1		Pára	

Assim, ao ser submetida a seguinte sequência de símbolos a mT da Tabela 1:

11101110

após a entrada do último símbolo (0), a fita armazenará a seguinte sequência de símbolos:

11111100

Em Gersting (2004) e Menezes (2005) podem ser encontradas demonstrações e explicações detalhadas sobre a mT.

5. CONCLUSÃO

As contribuições de Turing foram fundamentais para a área de Computação e Informática. Ele inspirou termos hoje comuns como a máquina de Turing e o teste de Turing. Como matemático, ele aplicou o conceito de algoritmo aos computadores digitais, diferenciando o hardware do software. Suas pesquisas, por exemplo, entre

máquinas e a natureza humana, influenciaram a criação da Inteligência Artificial.

Espera-se que este artigo tenha incitado o leitor a procurar conhecer um pouco mais sobre a história da computação moderna e sobre o trabalho de Alan Turing. Especialmente, aqueles que estão na área, conhecendo a história de sua profissão, poderão se tornar mais críticos e reflexivos, tanto quanto ao impacto de sua atuação quanto no aspecto de valorização profissional.

AGRADECIMENTOS

Os autores agradecem a bolsa de iniciação científica concedida pela FAPEMIG e a FAI pela oportunidade de realização deste trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

ASSOCIATION FOR COMPUTING MACHINERY (ACM). **A.M. Turing Award**. 2006. Disponível em: <http://awards.acm.org/homepage.cfm>. Acesso em: 20 Ago. 2006.

BITTENCOURT, Guilherme. **Inteligência Artificial: ferramentas e teorias**. 2 ed. Florianópolis: Ed. da UFSC, 2001. 362p.

CHAGAS, Elza Figueiredo. O envolvimento da matemática com a criação dos computadores: um estudo de caso da lógica matemática à Máquina Universal de Turing. **Revista Millenium**. Portugal, n. 25, 2002. Disponível em: www.ipv.pt/millenium/Millenium25/25_28.htm. Acesso em: 06 jun. 2006.

COPELAND, Jack. **A brief history of computing**. 2000. Disponível em: www.alanturing.net. Acesso em: 04 jun. 2006.

GERSTING, Judith. **Fundamentos Matemáticos para a Ciência da Computação**. 5 ed. Rio de Janeiro: LTC, 2004.

HODGES, Andrew. **Alan Turing: the enigma**. Vintage Books, 1992.

LOVE, Peter. The legacy of Alan Turing. **IEEE Computing in Science & Engineering**, vol. 6, issue 4, pages 97-99, July/August, 2004.

MENEZES, Paulo Blauth. **Matemática Discreta para Computação e Informática**. 2 ed. Porto Alegre: Ed. Sagra Luzzato, 2005.

RUSSEL, Stuart; NORVIG, Peter. **Artificial Intelligence: a modern approach**. Prentice Hall, 1995.

SETZER, Valdemar W. **Alan Turing e a Ciência da Computação**. Disponível em: www.ime.usp.br/~vwsetzer. Acesso em 18 de Jul 2006.

TURING, A. **Computing machinery and intelligence**. Mind. Out. 1950. Disponível em: <http://www.abelard.org/turpap/turpap.htm>

TURING, A. **On computable numbers, with application to the Entscheidungsproblem**. Nov. 1936. Disponível em: <http://www.abelard.org/turpap2/tp2-ie.asp>.

UMA CONFIGURAÇÃO DO LINUX E SAMBA COMO CONTROLADOR PRIMÁRIO DE DOMÍNIO PARA CLIENTES WINDOWS

Rogério Tomassoni de Araújo Junior
jrtomassoni@yahoo.com.br

Flávio Neri Rodrigues
flavio.neri@gmail.com

Faculdade de Administração e Informática

Resumo – Samba é um servidor e um conjunto de ferramentas que tornam possíveis a comunicação entre máquinas, utilizando sistemas operacionais diferentes, como Linux e Windows. Aborda-se, neste artigo, como configurar o Samba como um Controlador Primário de Domínio ou *Primary Domain Controller* (PDC), de forma a torná-lo responsável pela centralização e autenticação de usuários em uma rede.

Abstract – Samba is a server and a set of tools that make the communication among machines that use different operating systems such as Linux and Windows. This article introduces how to configure Samba to function as a Primary Domain Controller (PDC) that is responsible for the user centralization and identification in a network.

Palavras-chave – Samba, controlador primário de domínio, autenticação de usuário.

Keywords – Samba, primary domain controller, user identification.

1. INTRODUÇÃO

O crescimento das redes locais tornou as atividades de administração uma tarefa árdua. Muitas ferramentas surgiram para auxiliar os administradores e, neste cenário, o Samba desponta como uma solução ideal para a centralização do sistema de autenticação.

A centralização do controle de usuários e de senhas facilita as atividades do administrador, pois permite a mobilidade dos

usuários e garante um modelo único de controle de credenciais.

O Samba é um servidor e um conjunto de ferramentas que permitem a comunicação entre redes Microsoft, Linux e Unix, por meio do protocolo *Server Message Block* (SMB)/*Common Internet File System* (CIFS). Com Samba é possível configurar servidores de arquivos e de impressão e compartilhar os demais recursos de hardware, como as unidades de CDROM.

Mostra-se neste artigo a configuração de um Controlador Primário de Domínio ou *Primary Domain Controller* (PDC) que centralizará todos os usuários e senhas em uma só estação, sendo esta a responsável pela autenticação dos usuários na rede (SAMBA, 2006).

Considerando que o leitor já possua uma distribuição Linux instalada e em funcionamento, abordam-se somente os procedimentos de instalação e configuração do Samba como PDC para clientes Windows. Esses procedimentos foram testados nas máquinas com sistemas operacionais Linux Slackware 10.2 Kernel 2.4.31 (SLACKWARE, 2006), Samba 3.0.23b e clientes Windows XP Professional versão 5.1 Service Pack 2.

2. INSTALAÇÃO E CONFIGURAÇÃO DO SAMBA

Nesta seção apresentam-se os procedimentos necessários para a instalação e configuração do Samba (ECKSTEIN, TS e COLLIER-BROWN, 2003; SAMBA, 2006).

2.1 Instalação do Samba

O primeiro procedimento é obter o pacote necessário para a instalação e configuração do Samba que se encontra disponível no site oficial do projeto: <http://us1.samba.org/samba>. Após o *download*, pode-se partir para a instalação e a configuração propriamente dita. Desempacote e descompacte o conteúdo do arquivo, com o seguinte comando:

```
tar -zxvf samba-3.0.23b.tar.gz
```

Depois desse comando, será criado um diretório com o nome `samba-3.0.23b`. Acesse esse diretório e, em seguida, o subdiretório *source*. Para instalar utilize:

```
./configure --prefix=/etc/samba  
make  
make install
```

Aguarde pelo fim da execução do comando e, caso nenhum erro aconteça, será criado um diretório no caminho especificado na *flag* `--prefix`. Neste exemplo, será criado em `/etc/samba`, podendo ser alterado para onde se queira.

2.2 Configuração do Samba

Para configuração é preciso editar o arquivo `smb.conf` que contém todas as variáveis de configuração do Samba. Por questão de segurança, faça uma cópia do arquivo original, utilizando o comando:

```
cp smb.conf smb.conf.orig
```

A partir de agora, pode-se editar o arquivo `smb.conf` com o editor de preferência e realizar a configuração básica para o funcionamento do Samba como PDC.

Na seção *global* definem-se as variáveis que afetam diretamente o comportamento do Samba.

workgroup = domínio ou grupo da qual a máquina fará parte. Esse domínio será utilizado para *logon* (na opção “Fazer logon em:”).

netbios name = nome da máquina. Se o Samba não conseguir resolver o nome do servidor pelo DNS, será utilizado o nome declarado nesta variável. Um procedimento útil é colocar o mesmo nome encontrado no `HOSTNAME`. Utilize o comando *hostname* para verificar o nome da máquina. Esse nome aparecerá no ambiente de rede.

server string = breve comentário sobre o servidor.

keepalive = esta opção é usada para verificar se existe algum cliente que não está em funcionamento. Utiliza-se o tempo em segundos, o valor padrão é 300 segundos que representa, por sua vez, 05 minutos. Para redes com poucas máquinas, 30 segundos é uma boa opção, se ajustada para 0 não será verificado.

deadtime = esta opção ajusta o número de minutos que o Samba esperará por um cliente inativo, antes de fechar a sua sessão. Um cliente é considerado inativo quando não possui arquivos abertos no servidor. O valor padrão é 0, para nunca encerrar a sessão.

logon script = especifica o *script* que será utilizado quando o usuário efetuar o *logon*. A opção `%U` faz com que o *script* possua o nome do usuário, ou seja, cada usuário tem um *script* `<nome do usuario>.[bat|cmd]`.

logon path = indica o diretório do servidor onde ficarão armazenados os perfis dos usuários. Especificar um *logon path*, sem caminho, evita que os dados do usuário sejam armazenados no servidor e que uma mensagem de erro apareça.

logon drive = força o mapeamento do diretório *home* do usuário com a identificação de unidade `j:` na máquina cliente. Pode ser trocado por qualquer outra letra que esteja disponível na máquina cliente. Usualmente, especificar uma letra mais próxima do fim do alfabeto garante que o mapeamento seja efetuado com sucesso.

domain logon = determina se será permitido o *logon* no Samba. Deve ser ajustado como *yes*.

os level = este valor é usado quando existe uma disputa entre máquinas servidoras. Especificar um valor alto garante que esse servidor vencerá a eleição, caso ela seja realizada, garantindo sua prevalência. Máquinas Windows usam o valor 32. O valor recomendado é 100.

preferred master = se o Samba poderá efetuar uma eleição para o *domain master*. Deve ser ajustada como *yes*.

domain master = especifica o servidor *master* do domínio. Deve ser ajustada como *yes*.

security = diz respeito ao modo de segurança que será utilizado. A opção *user* diz que será efetuada autenticação com base nos usuários, permitindo assim que apenas aqueles que têm uma conta servidora possam fazer *logon*.

socket options = são opções de comunicação com os clientes da rede. A configuração indicada é TCP_NODELAY.

passwd program = caminho para o arquivo binário responsável pela troca de senha, o padrão é /usr/bin/passwd %u, onde %u é o nome do usuário.

passwd chat = utilizada para trocar a senha a partir da estação cliente Windows, juntamente com o *passwd program*.

Na seção *netlogon* ficarão os arquivos de lote, como os *scripts* de *logon* dos usuários, ajustados na variável global *logon script*.

comment = breve comentário sobre essa seção, não sendo obrigatório.

path = caminho completo do diretório que conterà os *scripts*. Certifique-se de que ele exista. O administrador poderá criar o diretório no caminho onde desejar, basta informar nesta variável.

guest ok = define se será permitido o acesso de visitantes. Ajuste como *yes*.

browseable = define se esta seção será visível no ambiente de rede. Ajuste como *no*.

Salve o arquivo após a realização de todas as configurações. Para verificar se existem erros no arquivo de configuração *smb.conf*, utilize o comando:

```
testparm /etc/samba/smb.conf
```

Acostume-se a usar o comando *testparm*, pois ele tem grande eficiência. A seguir é apresentado um exemplo do *Smb.conf* já configurado (SAMBA, 2006).

```
[global]
workgroup = Dominio
netbios name = SPDC
server string = Samba PDC
keepalive = 300
```

```
deadtime = 10
logon script = %U.bat
logon path =
logon drive = j:
domain logons = yes
os level = 100
preferred master = yes
domain master = yes
security = user
socket options = TCP_NODELAY
passwd program = /usr/bin/passwd %u
passwd chat = "**New*UNIX*password*"
               "**Retype*new*UNIX*password*"
               %n\n"
*passwd:*all*authentication*tokens*updated
*successfully*
```

```
[netlogon]
comment = Servico de logon
path = /etc/samba/netlogon
guest ok = yes
browseable = no
```

3. CRIAÇÃO DO DIRETÓRIO *NETLOGON*

Para criar o diretório *netlogon* utilize o comando:

```
mkdir -p /etc/samba/netlogon
```

Ajuste na permissão de acesso:

```
chmod 755 /etc/samba/netlogon
```

4. CRIAÇÃO DAS CONTAS DE USUÁRIOS

É necessário criar as contas de usuário e de máquinas para que o objetivo seja atingido. Siga as instruções dadas a seguir, utilizando o usuário *root*.

Primeiramente cadastre o *root* no Samba:

```
smbpasswd -a root
```

Defina, localmente, os usuários que utilizarão o Samba. Sintaxe do comando:

```
useradd -m -d <diretório de home real do usuário> -s /bin/false <nome do usuário>
```

Ex:

```
useradd -m -d /home/junior -s /bin/false junior
```

Com o intuito de aumentar a segurança, não é fornecido um *shell* para o usuário, pois ele não necessitará acessar o servidor para executar comandos. Adicione o nome do usuário no Samba:

```
smbpasswd -a junior
```

Para dar e obter permissão de administrador no domínio é preciso criar um grupo e associá-lo ao grupo "*Domain Admins*", para isso, primeiro crie um grupo com o comando:

```
addgroup admin
```

Onde *admin* é qualquer nome do grupo a ser criado.

O próximo passo é associar o grupo criado ao grupo do Unix, para isso execute:

```
net groupmap modify ntgroup="Domain Admins" unixgroup=admin
```

Após esses comandos, basta adicionar os usuários que terão permissão de administrador (permissão total) ao grupo *admin*. Para adicionar um usuário, no momento de sua criação, basta colocar a *flag* `-g` seguida do grupo:

```
useradd -g admin -m -d /home/junior -s /bin/false junior
```

Para adicionar os usuários já existentes pode-se agir de dois modos. O primeiro consiste em alterar diretamente o arquivo *group* em */etc/group*

```
admin:x:504:junior,filho,neto
```

Nota-se que os usuários *junior*, *filho* e *neto* foram adicionados ao grupo *admin*. Para acrescentar os usuários usando este modo, coloque os nomes dos usuários separados por vírgula. A outra maneira consiste em utilizar o comando *usermod -G* que insere um usuário por vez. Por exemplo:

```
usermod -G admin junior  
usermod -G admin filho
```

```
usermod -G admin neto
```

Também pode-se colocar o usuário em mais de um grupo, separando-os por vírgulas, assim:

```
usermod -G admin,presidencia junior
```

Neste caso, o usuário *junior* fará parte do grupo *admin* e *presidência*.

5. CRIAÇÃO DAS CONTAS DE MÁQUINA

É necessário que se criem contas para as máquinas. Os nomes deverão ser os reais das máquinas que são encontrados em: Iniciar > Configurações > Painel de Controle > Sistema > Nome do Computador.

Por exemplo, a máquina a ser adicionada chama *depti01*:

```
useradd -d /dev/null -s /bin/false depti01$
```

Note que foi inserido o símbolo *\$* na conta de máquina para aumentar a segurança. Para aumentá-la ainda mais, pode-se “travar” a senha da conta:

```
passwd -l depti01$
```

Para adicionar a máquina no Samba, utilize o comando *smbpasswd*. Não é necessário colocar o símbolo *\$*, pois ela será identificada pela *flag* `-m` (*machine*).

```
smbpasswd -a -m depti01
```

Com todos os procedimentos executados corretamente, o servidor estará pronto para ser usado, após o reinício do Samba. Coloque-o para iniciar junto com o sistema operacional (verifique como fazer isso na sua distribuição).

6. CRIAÇÃO DO SCRIPT DE LOGON

Um exemplo de *script* de *logon* genérico para todos os usuários:

```
net use h: \\SPDC\publico /yes
net time \\SPDC /set /yes
notepad.exe
```

A primeira linha faz com que o diretório público (não esqueça de criar o compartilhamento) do servidor \\SPDC (nome configurado na variável global *netbios name*) seja mapeado como unidade h: na máquina cliente. A segunda linha sincroniza o horário do cliente com o horário do servidor SPDC (o servidor deve estar configurado, para isso ajuste a opção *time Server* para *yes*). A terceira linha executa o arquivo notepad.exe toda vez que o *logon* for efetuado (use o notepad para criar o arquivo, para manter as quebras de linha no padrão Windows).

```
net use \\nome_servidor\
nome_compartilhamento /opção
```

A opção define se será persistente ou não, ou seja, se ficará mapeado permanentemente ou não após o *logoff*.

Esse arquivo precisa ser salvo dentro de `/etc/samba/netlogon` com um nome seguido da extensão *bat*. A variável *logon script* na seção global deve fazer referência a esse arquivo. Por exemplo:

```
logon script = logon.bat
```

Para personalizar o arquivo para cada usuário basta criá-lo e salvá-lo com o nome de usuário e usar a opção %U na variável *logon script* da seção global. %U é uma variável que traz em seu conteúdo o nome do usuário que fez o login no servidor. Por exemplo:

```
logon script = %U.bat
```

Se o existir no servidor um usuário chamado tomassoni, logo deverá conter um arquivo denominado *tomassoni.bat* dentro do diretório netlogon.

7. CONFIGURAÇÃO DAS ESTAÇÕES CLIENTES WINDOWS

Antes de colocar as estações no domínio, é preciso alterar o registro do

cliente Windows. Para isso, dentro da pasta descompactada do Samba (samba-3.0.23b), existe o diretório docs/registry que contém um arquivo para alteração do registro para cada sistema operacional. Escolha o arquivo apropriado, copie-o para a estação cliente e depois o execute (foi feito um teste com o Windows XP Profissional sem efetuar essa alteração e funcionou corretamente).

Dando andamento à configuração, coloque a estação no domínio, para isso faça: Iniciar > Configurações > Pannel de Controle > Sistema > Nome do Computador > Alterar No campo “Domínio” coloque o nome do domínio criado. Este nome está na variável *workgroup* da seção *global*. Após a configuração correta será exibida uma mensagem de boas-vindas.

Após reiniciar a máquina, no momento do *logon*, escolha “Opções >>” e em “Fazer *logon* em:” escolha o domínio criado.

Finalmente, para verificar se o Samba está em execução, utilize o comando:

```
ps aux | grep samba
```

8. CONCLUSÃO

Além de ser gratuito para o usuário, o Samba tem a seu favor o modelo de código aberto, o que propicia o crescimento do projeto, uma vez que qualquer pessoa apta poderá contribuir com seus conhecimentos. Um outro ponto a favor é o poder de automação de um sistema Unix, que diminui drasticamente as tarefas dos administradores, pois aquelas corriqueiras podem ser executadas com o emprego mínimo de esforço.

O custo para a realização do trabalho exposto neste artigo é próximo do zero, uma vez que foram utilizados apenas softwares gratuitos e de código aberto.

Importante ressaltar que foi apresentada apenas uma das várias funcionalidades que podem ser aplicadas utilizando os recursos citados. Assim recomenda-se a leitura dos *sites* oficiais, pois acredita-se que novos valores consideráveis serão agregados ao conhecimento do leitor.

AGRADECIMENTOS

Agradeço primeiramente a Deus por me proporcionar a vida; a todos que me ajudaram e ainda ajudam durante a caminhada rumo à graduação no curso de Sistemas de Informação; especialmente à minha tia Roselene Tomassoni; aos meus poucos, mas grandes amigos de escalada que, por muitas vezes, tiveram que aceitar minha ausência devido aos estudos. A todos aqueles que direta ou indiretamente contribuem com a minha pessoa. E, claro, não poderia deixar de lembrar e agradecer aos desenvolvedores de softwares *free e opensource*.

Rogério

Primeiramente a Deus, pelo prazer da vida. À minha esposa Erika e meu filho Leonardo, pela paciência e apoio. A meu amigo Rogério, que compartilhou sua idéia, acreditou em meu trabalho e proporcionou horas de lazer durante esta jornada. À comunidade *opensource* que produziu as ferramentas utilizadas e aos amigos que fiz no curso de Sistemas de Informação da Faculdade de Administração e Informática de Santa Rita do Sapucaí -MG.

Flávio

REFERÊNCIAS BIBLIOGRÁFICAS

ECKSTEIN, Robert; TS, Jay; COLLIER-BROWN, David. **Using Samba**. 2 ed. O'Reilly&Associates: 2003. Disponível em: <http://www.faqs.org/docs/samba/toc.html>

SAMBA. **The Official Samba-3 HOWTO and Reference Guide**. 2006. Disponível em: <http://us5.samba.org/samba/docs/man/Samba-HOWTO-Collection>.

SLACKWARE. **The Slackware Linux Project**. 2006. Disponível em: <http://www.slackware.Com>

A UTILIZAÇÃO DA BIBLIOTECA ALLEGRO PARA O DESENVOLVIMENTO DE APLICAÇÕES MULTIMÍDIA

Luana Pereira de Lima
luanaelima@gmail.com

Túlio Vitor Machado Faria
tuliofaria@gmail.com

Faculdade de Administração e Informática

Resumo - O artigo apresenta a biblioteca Allegro, que pode ser usada com a linguagem C/C++ para o desenvolvimento de aplicações multimídia e de jogos. Com a Allegro, podem-se adicionar aos aplicativos, recursos como imagens, sons, temporizadores e entrada de dados via mouse e teclado. Para demonstrar essas funcionalidades utiliza-se o ambiente Dev C++.

Abstract - This work introduces Allegro Library, which can be used with C/C++ language, in the development of multimedia and game applications. Using this library, one can add resources such as images, sounds, timers, mouse and keyboard data entry. Dev C++ Environment is used to demonstrate some of these functionalities.

Palavras-chave – Linguagem C/C++, Allegro, multimídia, jogos.

Keywords - C/C++ Language, Allegro, multimedia, games.

1. INTRODUÇÃO

O objetivo deste artigo é demonstrar as funcionalidades da Allegro, biblioteca criada especialmente para o desenvolvimento de jogos e aplicações multimídia. Essa biblioteca de funções permite adicionar sons, imagens, tratamento de entrada de dados via mouse e teclado, temporizadores, entre outras funcionalidades, nas aplicações desenvolvidas em C/C++.

Neste artigo é mostrado como instalar a biblioteca e como utilizar as funções que permitem inserir imagens e sons, empregar a

técnica de *Double Buffering*, tratar eventos de teclado e de mouse e controlar o tempo por meio de temporizadores.

Espera-se propiciar ao leitor, condições de desenvolver uma aplicação simples com recursos multimídia, utilizando para isso a biblioteca Allegro e a Linguagem C.

2. A BIBLIOTECA ALLEGRO

A Allegro é uma biblioteca multi-plataforma de código aberto utilizada para a programação de jogos e de aplicações multimídia em geral. Foi desenvolvida, inicialmente, para ser utilizada com a plataforma Atari ST, mas depois do desaparecimento dessa, a biblioteca passou pelo Borland C, até ser adotada pelo DJGPP (ambiente com o compilador C/C++), onde a mesma cresceu com sua mesclagem entre código C e Assembly.

Ela é muito conhecida pela sua facilidade em adicionar entrada de dados via teclado, mouse e joystick, áudio, temporizadores e imagens, em programas C e C++, aliada à pluralidade de sistemas operacionais suportados, como DOS, Windows, BeOS, Mac OS X e vários sistemas Unix (ALLEGRO, 2005)

3. INSTALAÇÃO DA ALLEGRO NO AMBIENTE DEV C++

A instalação da biblioteca Allegro no ambiente Dev C++ tem a vantagem de poder ser realizada através do Gerenciador de Pacotes do próprio compilador. Para isso, é preciso obter o pacote de instalação que fica

disponível em <http://www.devpaks.org> e abri-lo no Dev C++, sendo a mesma automaticamente instalada.

4. CRIAÇÃO DE UM PROGRAMA COM A ALLEGRO

A criação de um programa que utiliza a Allegro no Dev C++ é simplificada com a abertura de um novo projeto. Para isso, devem-se escolher as opções Arquivo, Novo e Projeto; depois selecionar *Multimedia, Allegro Application (static)* e a linguagem que será utilizada, no exemplo, é a linguagem C. Será necessário também selecionar uma pasta para salvar o projeto e dar um nome para o arquivo de projeto. Logo após esses procedimentos, será criado automaticamente um arquivo contendo todo código para iniciar um aplicativo em C com a Allegro.

Escolhendo as opções Executar, Compilar & Executar, do menu principal, poderá ser visto o aplicativo em execução. Note que o mesmo foi aberto em uma janela, conforme mostra a Figura 1.

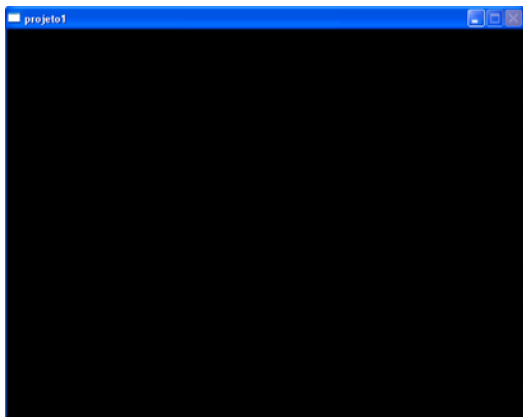


Figura 1. Janela do aplicativo em C com a Allegro

O código fonte, mostrado na Figura 2, foi criado automaticamente. Os comentários foram colocados para melhor entendimento das linhas de código.

5. USO DE IMAGENS

A carga e a utilização de arquivos de imagem na Allegro são feitas através de um ponteiro do tipo BITMAP. Para carregar a imagem do disco é utilizada a função *load_bitmap*. Depois de carregada a imagem para a memória é necessário exibi-la na tela. Para

isso, pode-se utilizar a função *draw_sprite(bitmap1, bitmap2, x, y)* que apenas desenha a imagem *bitmap2* em *bitmap1* na posição *x,y*.

```
#include <allegro.h>
/* inclui a biblioteca allegro.h */
void init();
void deinit();

int main() {
    init();

    while (!key[KEY_ESC]) {
        /* todo o jogo acontece em um laço
infinito, até que a tecla ESC seja pressionada. */
    }
    deinit();
    return 0;
}
END_OF_MAIN();

void init() {
    int depth, res;
    /* indica que será usada a allegro na aplicação */
    allegro_init();

    /* obtém a profundidade de cores usadas no PC de 8
a 32 bits */
    depth = desktop_color_depth();

    /* se depth contiver 0, define a mesma como 32 */
    if (depth == 0) depth = 32;

    /* define a quantidade de cores usadas na aplicação */
    set_color_depth(depth);

    /* seta o modo gráfico que vai ser usado */
    /* no caso,
    GFX_AUTODETECT_WINDOWED - detecta a
placa de vídeo automaticamente no modo windows
640, 480 - largura e altura da janela */
    res=
set_gfx_mode(GFX_AUTODETECT_WINDOWED,
640, 480, 0, 0);

    /* caso não consiga reconhecer a placa de vídeo
exibe uma mensagem de erro e encerra a aplicação */
    if (res != 0) {
        allegro_message(allegro_error);
        exit(-1);
    }

    install_timer();
    install_keyboard();
    install_mouse();
}
void deinit() {
    clear_keybuf();
}
```

Figura 2. Código fonte criado pelo Dev C++ com comentários

Pode parecer estranho desenhar uma imagem em outra, mas a Allegro interpreta a tela como uma imagem, que é definida como *screen*. Assim sendo, quando houver a necessidade de mostrar uma imagem na tela, basta desenhá-la na mesma.

Um detalhe interessante é que a Allegro permite que a imagem tenha fundo transparente. Para isso basta que ela possua a cor RGB (255,0,255), onde se deseja ter as áreas com transparência.

Um exemplo de como declarar, carregar e exibir uma imagem na tela é mostrado na Figura 3.

```
BITMAP *imagem;
imagem=load_bitmap("Fai.bmp",NULL);
draw_sprite(screen, imagem, 0, 0);
```

Figura 3. Exemplo de declaração e carga de um *Bitmap* com a Allegro

6. A TÉCNICA *DOUBLE BUFFERING*

A técnica de *Double Buffering* consiste em armazenar toda a tela em um ponteiro do tipo *BITMAP*, para depois escrevê-lo. Assim, caso haja a necessidade de escrever algo na tela, limpa-se o buffer com a função *clear* e inicia-se o processo de escrita novamente, sobrescrevendo a tela com o conteúdo do buffer. Essa técnica é utilizada em animações ou em trocas de desenhos na tela, para evitar que a mesma tenha variação na apresentação (WIKIPEDIA, 2006).

A Figura 4 mostra um exemplo de uma imagem que troca de posição na tela sem o uso de *Double Buffering*. Percebe-se que a tela, no segundo instante, fica sem nada desenhado, o que provoca um efeito de “pisque” da tela. A Figura 5 mostra o mesmo exemplo, porém já com o uso da técnica *Double Buffering*. Percebe-se que a tela, em nenhum instante, fica vazia, ou seja, em nenhum momento a mesma sofre o efeito de “pisca”.

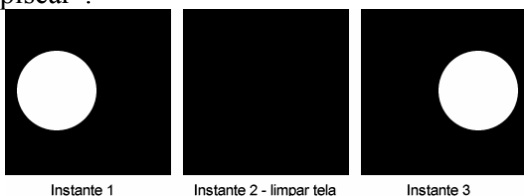


Figura 4. Exemplo de uma imagem trocando de posição, sem o uso de *Double Buffering*

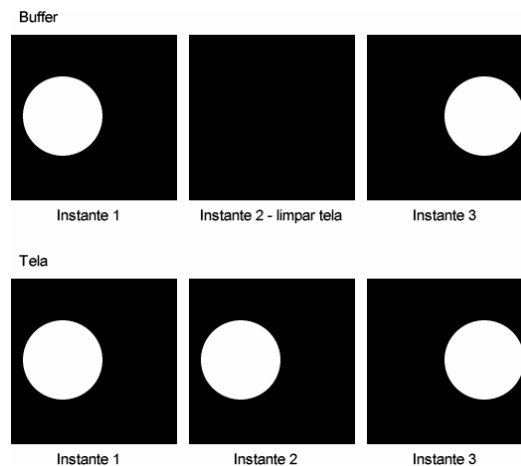


Figura 5. Exemplo de uma imagem trocando de posição, com o uso de *Double Buffering*

A Figura 6 mostra o código que realiza a movimentação de uma figura na tela, por meio das funções que implementam a técnica de *Double Buffering*.

```
BITMAP *buffer;
BITMAP *imagem;
buffer = create_bitmap(640,480);
/* usa a função create_bitmap para criar uma nova
imagem bitmap de 640 por 480 px */
imagem = load_bitmap("imagem.bmp",NULL);
int x = 0;
while(!key[KEY_ESC]){
    if (x==400){ x = 0; } /* evitar que a imagem saia
da tela */
    clear(buffer); /* limpa o buffer */
    draw_sprite(buffer, imagem, x, 0);
    draw_sprite(screen, buffer, 0,0);
    x ++;
}
```

Figura 6. Exemplo de código de uma animação com *Double Buffering*

7. USO DE FUNÇÕES DE ÁUDIO

A biblioteca Allegro permite o uso de vários formatos de sons, dentre eles, o MIDI e WAV. Para isso, deve-se inicialmente utilizar a função *install_sound* para obter a mesma configuração de som utilizada pelo computador onde o aplicativo for executado. Também, deve-se declarar um ponteiro do tipo *SAMPLE*, para o caso de um arquivo WAV ou do tipo *MIDI* para um arquivo MIDI.

Depois de declarados os ponteiros, os arquivos precisam ser carregados na memória,

com o uso das funções *load_midi* ou *load_wav*, de acordo com o tipo de arquivo.

A execução do som é feita através da função *play_midi* (*musica, loop*), que toca um arquivo MIDI uma única vez, caso o parâmetro *loop* seja *FALSE*, ou infinitas vezes caso seja *TRUE*. A função *play_sample*(*musica, vol, pan, freq, loop*) executa um arquivo WAV, com o volume determinado em *vol*, no canal *pan* ($0 \leq \text{pan} < 255$, sendo 0 canal esquerdo e 255 direito), na velocidade *freq* (sendo 1000 o valor original, caso o valor seja 2000, a música será executada com o dobro da velocidade). O parâmetro *loop* tem a mesma função já citada.

Na Figura 7 é mostrado o código fonte para carga e execução de uma música MIDI e uma música WAV.

```
MIDI *midi;
midi = load_midi("Greensleeves.mid");
play_midi(midi, TRUE);

SAMPLE *wav;
wav = load_wav("Beep.wav");
play_sample(wav, 100, 128, 1000, FALSE);
```

Figura 7. Código para execução de um MIDI e WAV

8. ENTRADA DE DADOS VIA TECLADO

A biblioteca Allegro facilitou o uso do teclado nas aplicações multimídia, sendo esse um dos meios de entrada mais utilizados em jogos.

Quando o projeto é criado, o teclado é automaticamente iniciado com a função *install_keyboard*. Depois de iniciado, para se saber se uma tecla foi pressionada, basta testar se o seu código está dentro do *array key*. A Figura 8 mostra um exemplo.

```
if (key[KEY_SPACE]){
    /* ação que será realizada caso o espaço seja
    pressionado */
}
```

Figura 8. Código para verificar se a barra de espaço foi pressionada

9. ENTRADA DE DADOS VIA MOUSE

A manipulação do mouse em Allegro pode ser feita após a iniciação com a função *install_mouse*.

Depois disso, pode-se capturar a posição do mouse na tela através das variáveis *mouse_x* e *mouse_y*. O cursor do mouse, por padrão, não fica visível na tela. Para torná-lo visível têm-se três maneiras. A primeira é através da função *show_mouse*, que exhibe o cursor padrão na tela; a segunda é através da função *set_mouse_sprite*, que define um ponteiro do tipo BITMAP como sendo o cursor do mouse e a terceira e última maneira, desenhando uma imagem na posição *mouse_x* e *mouse_y*.

Para saber se os botões do mouse foram pressionados existem duas maneiras. A primeira, usando *mouse_b & 1*, para o caso do botão esquerdo pressionado, e *mouse_b & 2* para o direito.

Na Figura 9 são apresentados os dois primeiros métodos para mostrar o cursor e como capturar o clique do mouse.

```
/* primeira maneira */
show_mouse(screen);

/* segunda maneira */
BITMAP *cursor;
cursor = load_bitmap("cursor.bmp", NULL);
set_mouse_sprite(cursor);

/* verificando se o mouse foi pressionado */
if (mouse_b & 1){
    /* ação para quando o cursor esquerdo foi acionado */
}
```

Figura 9. Exemplos de uso de funções do mouse

10. USO DE TEMPORIZADORES

Um recurso bastante interessante oferecido pela biblioteca Allegro é a possibilidade de adicionar temporizadores (*timers*) nas aplicações, de forma que uma função possa ser executada em intervalos de tempo definidos. Esse recurso permite criar vários itens em um jogo, como marcadores de tempo e animações, de forma independente da velocidade do computador. Para utilizar um temporizador, é necessário iniciá-lo com a função *install_timer*.

A Figura 10 exemplifica o uso de temporizadores e suas particularidades em um código comentado.

```
volatile int tempo = 0; /* declaração de uma variável
inteira tempo */
LOCK_VARIABLE(tempo); /* toda variável global
usada em uma função manipulada pelo Trimer deve ser
travada */
void relógio(){
    tempo++;
    /* toda função manipulada por Timers, não deve
retornar nenhum valor e não pode manipular a tela.
Além de ter que executar instruções simples para não
sobrecarregar o processamento da máquina e porque
timers também são usados em funções de música, por
exemplo. */
}
END_OF_RELOGIO(); /* indica o fim da função */

/* quando quiser que a função comece a ser executada
de tempos em tempos */
install_int(relógio,1000); /* executará a função relógio
de 1000 em 1000 milisegundos. A cada intervalo de
tempo, a variável tempo será incrementada em 1 */
```

Figura 10. Exemplo de uso de temporizadores

11. CONCLUSÃO

Procurou-se, neste artigo, mostrar as funções básicas da biblioteca Allegro que permitem a utilização de recursos de multimídia.

O desenvolvimento de jogos e aplicações multimídia é um trabalho complexo em linguagens como C e C++. Levando isso em consideração, a biblioteca Allegro facilita o processo de adicionar recursos como imagens, sons e temporizadores, que são de suma importância para esses aplicativos, permitindo que se diminua o seu tempo de desenvolvimento.

REFERÊNCIAS BIBLIOGRÁFICAS

ALLEGRO. **Official Site**. 2005. Disponível em: <http://www.allegro.cc>. Acesso em: 5 nov. 2005

WIKIPEDIA. **Double Buffering**. 2006. Disponível em: http://en.wikipedia.org/wiki/Double_buffering. Acesso em: 5 out. 2006.

INTRODUÇÃO À PLATAFORMA *JAVA MICRO EDITION*: CONFIGURAÇÕES E PERFIS

Alexandre Ferraz Luz
aleafl@yahoo.com.br

Karl Pascher Wolf
karl@uti.psi.br

Faculdade de Administração e Informática

Resumo – Este artigo tem como finalidade apresentar a plataforma *Java Platform Micro Edition* (J2ME) e as configurações e perfis para a programação de aplicativos utilizados, principalmente, em dispositivos móveis ou portáteis. As configurações foram criadas para que fosse possível integrar e suportar a variedade de produtos existentes no contexto dessa plataforma. Os perfis vieram para disponibilizar mais flexibilidade na classificação de cada dispositivo de acordo com a variação dos seus recursos.

Abstract – This work has as its purpose to show *Java Platform, Micro Edition* (J2ME) and its configurations and profiles for applications programming in a mobile device. The configurations have been created so that it is possible to integrate and deal with the variety of existing products in J2ME context. The profiles were brought up to provide more flexibility in the classification of each mobile device according to their resource variation.

Palavras-chave – Java, J2ME, MIDP, CLDC.

Keywords - Java, J2ME, MIDP, CLDC.

1. OS TIPOS DE PLATAFORMAS JAVA

O ponto de partida da linguagem de programação Java foi a *Java Platform, Standard Edition* (J2SE), com foco no desenvolvimento de aplicativos para estações de trabalho e *desktops*. A idéia que fundamentou a criação dessa linguagem foi que um aplicativo deveria ser escrito para ser executado em qualquer plataforma que

suportasse uma máquina virtual Java (*Java Virtual Machine* - JVM).

Desde 1995, quando a Linguagem Java foi criada nos laboratórios da Sun Microsystems, são evidentes as mudanças significativas no cenário da Computação. Com essas mudanças, a Linguagem Java teve seu alcance ampliado além dos computadores pessoais. Prova disso, é a *Java Platform, Enterprise Edition* (J2EE), que surgiu em 1997 com o intuito de fornecer suporte para aplicativos multiusuários em nível empresarial de larga escala.

Além das duas versões apresentadas, a última a ser incorporada à família de tecnologias Java foi a *Java Platform, Micro Edition* (J2ME), que possui como principal objetivo atender aos dispositivos móveis ou portáteis providos de memória, vídeo, processamento e conectividade limitados.

2. CARACTERÍSTICAS DA J2ME

A plataforma J2ME é destinada para o desenvolvimento de aplicativos para dispositivos móveis ou portáteis de uma forma geral, como *paggers*, aparelhos celulares, controles remotos, *personal digital assistant* (PDA), dentre outros. Com o intuito de tornar a interação com esses dispositivos dinâmica, a plataforma possui um conjunto de classes restrito e de fácil programação, um ambiente em tempo de execução que fornece uma execução segura e portátil, e é claro, possibilidade de acesso a conteúdos dinâmicos.

Para entender a finalidade dessa plataforma, basta compreender o quão impróprio seria utilizar interfaces de

programação de aplicativos (*Application Program Interface* - API) J2SE em dispositivos móveis (microdispositivos). Por exemplo, a tela de um aparelho celular não pode fornecer toda a funcionalidade disponível no *Abstract Window Toolkit* (AWT), que foi a primeira biblioteca de classes desenvolvida para interface gráfica de usuário. Desta forma, a tecnologia J2ME é utilizado para tratar das necessidades especiais de consumidores que ficam de fora da abrangência da J2SE e J2EE.

A seguir apresentam-se os conceitos de Configurações e Perfis e as razões pelas quais eles são tão importantes para o desenvolvimento na J2ME.

2.1. Configurações

O conceito de Configuração foi introduzido pela Sun para que fosse possível dar suporte a grande variedade de produtos inseridos no contexto J2ME.

Uma configuração define uma plataforma Java para uma faixa de dispositivos. Ela está intimamente vinculada a uma JVM. Na verdade, uma configuração define os recursos da linguagem Java e as bibliotecas básicas da JVM para esta configuração em particular (MUCHOW, 2004).

Cada aplicação tem uma configuração de acordo com a capacidade de memória, poder de processamento, limitações de vídeo e conectividade de rede de um determinado dispositivo.

A seguir estão as características típicas dos dispositivos dentro das duas configurações definidas (MUCHOW, 2004).

2.1.1. Configuração de Dispositivo Conectado (CDC)

As características típicas de um equipamento com CDC são:

- 512 Kilobytes (no mínimo) de memória para executar o Java;
- 256 Kilobytes (no mínimo) de memória para alocação em tempo de execução;

- Conectividade com rede, com largura de banda possivelmente persistente e alta.

2.1.2. Configuração de Dispositivo Conectado Limitado (*Connected Limited Device Configuration* - CLDC)

As características típicas de um equipamento com CLDC são:

- 128 Kilobytes (no mínimo) de memória para executar o Java;
- 32 Kilobytes para alocação de memória em tempo de execução;
- interface restrita com o usuário;
- baixo poder, normalmente alimentado por bateria;
- Conectividade de rede. Normalmente dispositivos sem fio com largura de banda baixa e acesso intermitente.

2.2. Perfis

Com intuito de dar mais flexibilidade na classificação dos dispositivos de acordo com a variação dos seus recursos, foi introduzido na plataforma J2ME o conceito de perfil.

Um perfil é uma possibilidade de extensão de uma configuração. Ele fornece as bibliotecas para um desenvolvedor escrever aplicativos para um tipo particular de dispositivo. Por exemplo, o perfil de dispositivo de informação móvel (*Mobile Information Device Profile* - MIDP), define APIs próprias para componentes, entrada e tratamento de eventos de interface com o usuário, armazenamento persistente, interligação em rede e cronômetros, levando em consideração as limitações de tela e da memória dos dispositivos. Trata-se de uma extensão da CLDC (MUCHOW, 2004).

Na seqüência deste artigo são detalhadas as informações sobre o MIDP bem como as diferenças entre as suas versões.

3. O PERFIL MIDP

O MIDP é um dos perfis mais utilizados nos dispositivos móveis,

representados em sua maioria pelos aparelhos celulares. Esse perfil encontra-se na versão 2.0, embora a maioria dos dispositivos tenha os recursos apenas para a versão 1.0.

O MIDP é um dos perfis que estão definidos para a CLDC. Para explicar o funcionamento desse perfil, a Figura 1 apresenta a sua arquitetura geral. No canto direito superior, estão as aplicações nativas que até o lançamento da plataforma J2ME eram as únicas aplicações existentes em um dispositivo móvel. Essas aplicações nativas são os programas padrão para configuração de um dispositivo, como para controle de volume, atualização de data, tipo do toque e não podem ser removidas ou atualizadas. Esses aplicativos específicos são os programas que foram feitos pelo fabricante do dispositivo para aproveitar o máximo de recursos que ele oferece.

No canto esquerdo superior estão os aplicativos MIDP. Estes são programas feitos na linguagem Java, mais especificamente J2ME, utilizando a configuração CLDC e o perfil MIDP. Já as classes específicas do dispositivo podem incluir opções como capacidade de responder às chamadas recebidas ou de pesquisar entradas em uma agenda telefônica. Ressalta-se que essas classes são específicas para o dispositivo e a sua utilização pode levar à perda de portabilidade.

A CLDC é instalada no software nativo e é a base para o perfil MIDP. Note que as aplicações MIDP têm acesso às bibliotecas CLDC e MIDP.

O perfil MIDP foi desenvolvido por um grupo de pessoas de várias entidades e é descrito em um documento chamado “A especificação MIDP” ou JSR-118 (JCP, 2006).

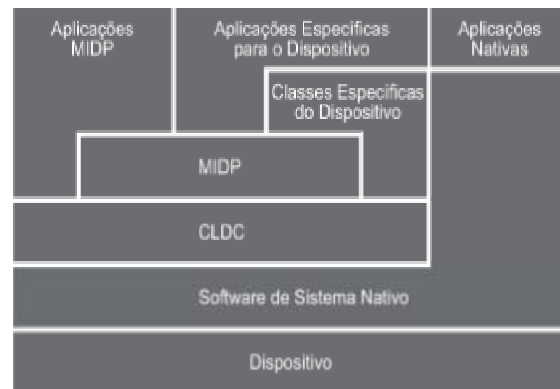


Figura 1. Arquitetura de um dispositivo MIDP
Fonte: SUN (2000).

3.1. Características do MIDP 2.0

Segundo a especificação MIDP versão 2.0 um dispositivo MID deve ter as seguintes características mínimas de hardware:

- Tela
 - Tamanho: 96x54 pixels.
 - Profundidade de cor: 1-bit.
 - Forma do pixel (*aspect ratio*): aproximadamente 1:1.
- Entrada
 - Um ou mais dos mecanismos de entrada seguintes: "teclado de uma mão", "teclado de duas mãos" ou tela sensível ao toque.
- Memória
 - 256 kilobytes de memória não volátil para os componentes MIDP.
 - 8 kilobytes de memória não volátil para dados persistentes criados pelas aplicações.
 - 128 kilobytes de memória volátil para o ambiente de execução do Java.
- Comunicação
 - Comunicação sem fio nos dois sentidos, possivelmente intermitente e com largura de banda limitada.

Segundo CARDOSO (2006), em relação aos requisitos de software, o MIDP assume apenas funcionalidades mínimas uma vez que, no nível dos dispositivos MID, há uma grande variedade de arquiteturas de

sistema. Os requisitos mínimos são os seguintes:

- um mecanismo para ler e escrever em memória não-volátil;
- acesso de leitura e escrita ao sistema de comunicação sem fios do dispositivo;
- um mecanismo que forneça uma base de tempo para ser usado nas *APIs* de temporizadores;
- capacidade mínima para escrever gráficos do tipo mapas de bits na tela;
- um mecanismo para capturar entrada do usuário;
- um mecanismo para gerenciar o ciclo de vida das aplicações do dispositivo.

Além dos requisitos mencionados, a especificação MIDP 2.0 (JCP, 2006) definem também outros que não podem ser facilmente incluídos nas categorias anteriores. Segundo a especificação, os dispositivos MIDP 2.0 são obrigados a:

- suportar *MIDlets* e *MIDlet suites* (descritas na seção 3.3) nas versões MIDP 1.0 e 2.0, ou seja, devem ser retro-compatíveis com a versão anterior do MIDP;
- implementar a especificação OTA (descrita na seção 3.4);
- fornecer as indicações visuais ao usuário em relação ao uso da rede;
- suportar o acesso a servidores HTTP 1.1 e conexões HTTP seguras (HTTPS);
- suportar transparência nas imagens PNG;
- suportar gerações de tons (biblioteca multimídia);
- suportar SP-MIDI (formato MIDI adaptado para dispositivos móveis).

3.2. Bibliotecas do MIDP

O MIDP oferece ao programador algumas bibliotecas de classes para o desenvolvimento de aplicações. Essas bibliotecas abrangem as seguintes áreas:

- Bibliotecas do núcleo

O perfil MIDP adiciona algumas classes e funcionalidades aos pacotes `java.lang` e `java.util` da CLDC.

- Ciclo de vida da aplicação

Classes que definem o ciclo de vida da aplicação e a forma como é controlada pelo dispositivo.

- Interface Gráfica

Classes para construir interfaces gráficas e para obter a entrada do usuário.

- Armazenamento persistente

Apesar de não existir o conceito de arquivo em MIDP, existem, no entanto, classes para ler e escrever dados num sistema de armazenamento persistente.

- Comunicação

O MIDP disponibiliza classes para ter acesso ao sistema de comunicação do dispositivo, o que possibilita, por exemplo, o uso de conexões HTTP.

- Multimídia

Classes para manipulação de recursos multimídia como geração de tons, notas MIDI, reprodução de arquivos de áudio e vídeo.

- Segurança

Biblioteca relacionada com certificados usados em ligações seguras.

A Tabela 1 mostra os pacotes Java definidos pelo perfil MIDP 2.0.

TABELA 1. Pacotes da API MIDP 2.0

Funções	Pacote
Bibliotecas do núcleo	<code>java.lang</code> <code>java.util</code>
Ciclo de vida da Aplicação	<code>javax.microedition.midlet</code>
Interface Gráfica	<code>javax.microedition.lcdui</code> <code>javax.microedition.lcdui.game</code>
Armazenamento persistente	<code>javax.microedition.rms</code>
Comunicação	<code>javax.microedition.io</code>
Multimídia	<code>javax.microedition.media</code>
Segurança	<code>javax.microedition.pki</code>

Fonte: CARDOSO (2006).

3.3. *MIDlets* e *MIDlet Suites*

As aplicações MIDP desenvolvidas são chamadas pela Sun de *MIDlets*. Uma *MIDlet* consiste em:

- uma classe que estende a classe `javax.microedition.midlet.MIDlet`
- outras classes necessárias à aplicação.

As *MIDlets* são distribuídas em arquivos Java (*Java Archive – JAR*). Várias *MIDlets* podem ser empacotadas num arquivo JAR, por meio do *MIDlet Suite*. Neste caso, as várias *MIDlets* podem compartilhar classes e recursos contidos no mesmo arquivo JAR.

3.3.1. Ciclo de vida de uma *MIDlet*

Uma *MIDlet* tem um ciclo de vida semelhante às *applets* (aplicações executadas em navegadores *web*). Existem três estados possíveis no ciclo de vida de uma *MIDlet*:

- inativa: a instância da *MIDlet* foi construída e está inativa;
- ativa: a *MIDlet* está ativa e em operação;
- destruída: a *MIDlet* foi terminada.

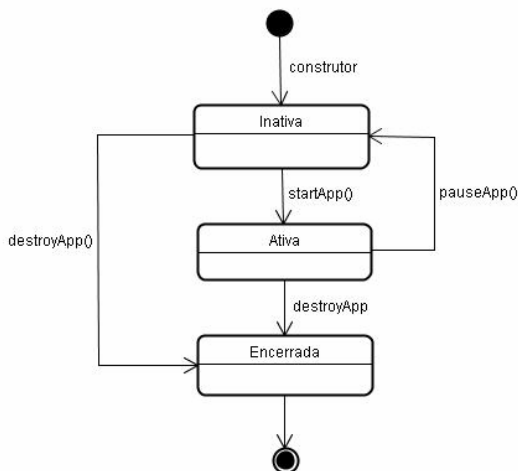


Figura 2. Diagrama de estados de uma *MIDlet*

O ciclo de vida das *MIDlets* é controlada pelo gerenciador de aplicações (*Application Management Software – AMS*) do dispositivo. O AMS é, basicamente, um programa (ou conjunto de programas) que gerencia a instalação, execução e remoção das aplicações MIDP no dispositivo.

A Figura 2 mostra o diagrama de estados de uma *MIDlet* e as transições entre os possíveis estados. Imediatamente depois de ser construída, a *MIDlet* encontra-se no estado inativo (*paused*). Em seguida, o AMS invoca o método `startApp()` e a *MIDlet* passa para o estado ativo. É nesse estado que a pessoa

pode interagir com a aplicação. Do estado ativo, a *MIDlet* pode passar para o estado inativo por imposição do AMS ou por opção da própria *MIDlet*. O AMS pode terminar a aplicação a pedido do usuário ou a pedido da *MIDlet* (CARDOSO,2006) .

3.4. Distribuição *Over The Air*

Uma das grandes vantagens do uso da plataforma J2ME em aparelhos celulares é a facilidade com que as pessoas podem instalar novas aplicações nos seus dispositivos.

A especificação MIDP (JCP, 2006) define uma forma padronizada para a distribuição e instalação de *MIDlets*, chamada de *Over The Air User Initiated Provisioning* (OTA).

De uma forma geral, a especificação OTA define que os dispositivos devem fornecer uma forma para a pessoa descobrir *MIDlets* que possam ser instaladas. Isso pode ser feito através do navegador residente no dispositivo ou através de uma aplicação própria. Através dessa aplicação, o usuário pode escolher a *MIDlet* que pretende instalar e iniciar o processo de instalação. A transferência dos arquivos é feita via HTTP 1.1.

O processo típico para distribuir uma *MIDlet* é o seguinte:

- colocar o arquivo JAR e o *arquivo Java Application Descriptor* (JAD) num servidor *web*;
- colocar um *link* para o JAD numa página HTML.

A instalação é feita acessando a página HTML com o navegador do dispositivo e ativando o link. O AMS irá descarregar um arquivo JAD e perguntar para a pessoa se pretende instalar a *MIDlet*. Em caso afirmativo, o arquivo JAR será descarregado (o endereço do JAR está definido num atributo do arquivo JAD), somente se o gerenciador de aplicações determinar que o dispositivo tem a capacidade para instalar e executar a *MIDlet* (a partir da descrição feita no JAD).

3.5.Diferenças entre o MIDP 1.0 e o MIDP 2.0

A versão 2.0 do perfil MIDP é pouco mais do que uma versão aumentada da versão 1.0 em relação às bibliotecas disponibilizadas.

As principais diferenças são:

- o mecanismo de distribuição OTA era apenas uma recomendação no MIDP 1.0 e na versão 2.0 passou a ser obrigatório;
- adição de bibliotecas multimídia para geração e reprodução de áudio e para reprodução de vídeo; no MIDP 1.0 era impossível gerar ou reproduzir som ou vídeo;
- a API de interface gráfica foi estendida com funções direcionadas para jogos;
- maior nível de segurança para acesso às funções sensíveis.

CONCLUSÃO

A plataforma J2ME facilita o desenvolvimento de aplicativos para dispositivos móveis ou portáteis. As configurações e os perfis criados tornaram possível a evolução das aplicações para esses dispositivos, como o caso do perfil MIDP 2.0 que utiliza a configuração CLDC. Esse perfil possibilita a utilização de interfaces em três dimensões, bem como a reprodução de sons e vídeo com uma maior qualidade. Esses e outros recursos permitem que as aplicações sejam cada vez mais providas de um maior nível de interatividade, segurança e portabilidade.

REFERÊNCIAS BIBLIOGRÁFICAS

CARDOSO, J. **Java Para Telemóveis-MIDP 2.0**. 2006. Disponível em: <http://livromidp.jorgecardoso.org/>. Acesso em: 29 mar. 2006.

JAVA COMMUNITY PROCESS. **Community Development of Java Technology Specifications**. JSR-000118 Mobile Information Device Profile 2.0. 2006. Disponível em: <http://jcp.org/aboutJava/>

communityprocess/final/jsr118/>. Acesso em: 28 mar. 2006.

MUCHOW, John W. **Core J2ME – Tecnologia e MIDP**. São Paulo: Ed. Pearson Makron Books, 2004.

SUN MICROSYSTEMS. **Mobile Information Device Profile 1.0a. Especificação**. 2000. Disponível em: <http://www.sun.com>.

RESUMOS DOS PROJETOS DE CONCLUSÃO DE CURSO DE SISTEMAS DE INFORMAÇÃO – TURMA 2005

A ABORDAGEM DE PROJETO CENTRADO NO USO

André Luís Alves Dias

Profa. Eunice Gomes de Siqueira
Orientadora

Resumo: Com uma maior preocupação do mercado em relação à usabilidade, um novo desafio para os desenvolvedores de software surgiu. Este trabalho tem como objetivo apresentar uma abordagem para o desenvolvimento de sistemas que garanta, em grande parte, a usabilidade desses por possuir técnicas simples e poderosas. A abordagem de projeto centrado no uso é baseada na obra Software For Use. A visão de sistemas de software como ferramentas é o cerne dessa abordagem, onde o foco do desenvolvimento deve ser o uso que o sistema terá e, conseqüentemente, quem o utilizará e a forma como isto será feito.

Palavras-chave: Projeto centrado no uso, usabilidade, ferramentas de software.

ENGWORKS PRINTERS DRIVER FOR WINDOWS

Carol Gaudino Palma

Fabrcio Fraga Ferreira

Mrcia Cristina da Silva

Rafael Noronha Tavares Gomes

Tnia Noberto Couto

Profa. Eunice Gomes de Siqueira
Orientadora

Resumo: Drivers de dispositivos so basicamente unidades de software que fazem a interface de comunicao entre o Sistema Operacional (SO) e os dispositivos de hardware conectados ao computador. Eles so verdadeiras partes do SO, sendo responsveis por permitir que os aplicativos acessem os dispositivos de hardware sem que o desenvolvedor desses aplicativos conheca os detalhes especficos dos mesmos. O ENGWORKS Printers Driver for Windows (EPD for Windows) e um projeto de pesquisa e desenvolvimento de um driver de dispositivo para a comunicao do SO Windows com as impressoras trmicas fabricadas pela empresa ENGWORKS Industrial Ltda.

Palavras-chave: drivers de dispositivos, drivers de impressora, impressoras trmicas.

GAIA - SISTEMA DE GESTÃO AMBIENTAL

Alisson Lopes

Carlos Eduardo Moreira de Mello Borges

Gisela de Lima Coelho

Gisele Anézio Felipe

Gustavo Pires dos Santos

Prof. Alexandre Franco de Magalhães
Orientador

Resumo: O projeto GAIA - Sistema de Gestão Comercial consiste no desenvolvimento de um sistema de software para setor de vendas de uma empresa que atua no ramo de telecomunicações. O GAIA é um aplicativo web que deve implementar os cadastros básicos sobre vendas, cálculo de bônus, cálculo de comissões e agendamento de compromissos. Ele também serve de apoio aos usuários no processo de tomada de decisão, através da geração de gráficos e relatórios com informações gerenciais.

Palavras-chave: desenvolvimento web, sistema de vendas, desenvolvimento de software.

SISTEMA DE CONTROLE DE ARQUIVOS MULTIMÍDIA PARA WEB: SCAM2WEB

Ednardo David Segura

Juarez Monteiro Villela Vitor

Leandro Gomes Silveira

Pedro de Faria Costa

Ruhelinton Perre de Lima

Prof. Alexandre Franco de Magalhães
Orientador

Resumo: Desenvolver software com qualidade deixou de ser um fator diferencial para tornar-se um requisito fundamental para as empresas desenvolvedoras. Este trabalho apresenta uma revisão sobre métodos e tecnologias para o desenvolvimento de aplicações web com qualidade, escalabilidade e segurança. Assim, são abordados os conceitos de padrões web e de projetos de software, as principais técnicas de invasão e as tecnologia Java Server Pages (JSP) e struts utilizadas para o desenvolvimento de aplicativos. Um sistema de software para controle de arquivos multimídia é utilizado como estudo de caso.

Palavras-chave: padrões de projeto, padrões web, Java server pages, desenvolvimento web.

SISTEMA DE SOFTWARE PARA APOIO AO MÉTODO DE AVALIAÇÃO SCAMPI: E-WEB SCAMPI

Camillo Alessandro Camillo

Eduardo do Carmo Lopes

Pedro Henrique Begalli Rezende

Rodrigo Duarte Beraldo

Ronaldo Vitorino Vilela

Prof. Nilson Sant'Anna
Orientador

Resumo: O projeto e-Web SCAMPI consiste no desenvolvimento de um sistema de software com a função de apoiar o método de avaliação Standard CMMI Aprasial Method Process Improvement (SCAMPI).

Palavras-chave: qualidade de software, CMMI, SCAMPI.

SISTEMA DE CONTROLE LABORATORIAL – SISLAB

Sandro Retori Pinto

Profa. Silvana Isabel de Lima
Orientadora

Resumo: O desenvolvimento tecnológico, principalmente da Informática, permite às empresas melhorias em seus processos de negócio, trazendo a possibilidade de maior produtividade e qualidade em seus serviços. Este projeto visa o desenvolvimento de um sistema de software - SISLAB capaz de atender às necessidades de um laboratório especializado em análises clínicas.

Palavras-chave: sistema de software, automatização de processos, controle laboratorial.

SYSTEM MOBILE

Carolina Pereira Mângia

Christiane Martins Costa

Daiani Vaz Moreira

Mariana Pereira Osório

Paula Junqueira Paiva

Profa. Eunice Gomes de Siqueira
Orientadora

Resumo: O projeto System Mobile consiste no desenvolvimento de um sistema de software para o ambiente acadêmico que atenda aos corpos docente e discente, aos coordenadores e bibliotecários de uma instituição de ensino. Por meio da utilização de dispositivos portáteis, como os aparelhos celulares, os alunos poderão consultar suas notas e frequências, reservar e renovar empréstimos de exemplares do acervo da Biblioteca. Os professores poderão requisitar materiais didáticos, consultar a situação do seu pedido, reservar e renovar empréstimos de exemplares do acervo da Biblioteca, além de enviar comunicados para a coordenação acadêmica. No ambiente da instituição, os coordenadores terão acesso às requisições dos materiais didáticos e aos comunicados enviados pelos professores para a instituição. Os bibliotecários poderão atender aos pedidos de reservas e às renovações de empréstimos de exemplares do acervo, enviadas pelos professores e alunos.

Palavras-chave: tecnologias móveis, mobilidade, sistemas de software.

Orientações para a apresentação de artigos

1. A Revista INICIA Publica artigos acadêmicos, monografias, resenhas de livros e artigos livres.
2. Poderão enviar trabalhos todos os alunos universitários matriculados nesta Instituição, dando prioridade a temas relacionados às áreas de Administração, Informática e Educação.
3. Todos os originais recebidos serão submetidos à apreciação do professor da disciplina subordinada e ao Conselho Editorial, que decidirá pela sua publicação. A Revista INICIA não se compromete a devolver os textos que não forem publicados, podendo fazê-lo por solicitação dos autores.
4. A Revista INICIA tem autonomia para fazer alterações nos originais, para adequá-los às normas de correção gramatical e ortográfica e às exigências de clareza, objetividade e concisão, tendo em vista torná-los acessíveis ao maior número possível de leitores.
5. A Revista INICIA não remunera os autores do texto publicado.
6. Os artigos e monografias não devem exceder a dez páginas incluindo referências bibliográficas, notas, tabelas e gráficos. Ambos devem ser acompanhados de resumo e abstract de no máximo 150 palavras cada um. As palavras chaves poderão ser de 3 a 5 no máximo.
7. Os artigos, monografias e resenhas devem ser formatados para folha A-4, fonte Times New Roman nº12, com espaçamento simples.
8. Os textos devem ser gravados em disquetes 3½, em Word for Windows 6.0 ou outro editor de texto compatível. O disquete deve vir acompanhado de cópia em papel.
9. Os originais devem conter as seguintes informações sobre o autor: instituição a que está vinculado, ano que está cursando e endereço eletrônico.
10. Os originais devem ser entregues sempre na primeira semana de agosto.
11. A Revista INICIA será divulgada na FAITEC.
12. Nas referências bibliográficas, que acompanham os artigos, as obras devem ser identificadas por autor e data, entre parênteses, no transcorrer do texto. No fim do artigo deve constar a referência bibliográfica completa. Se, por exemplo, for citada no texto a obra Raízes do Brasil, de Sérgio de Holanda Buarque, escrever (Holanda, 1935, pp. 73-75). Ao fim do texto deve aparecer:

HOLANDA, Sérgio Buarque de (1936) Raízes do Brasil, São Paulo, 5ª edição, Global Editora, 1969. Primeira edição, 1936.

A data escolhida para referência pode ser a da edição ou, de preferência, a da edição original, de forma a deixar clara a época em que a obra foi produzida. No caso, deve-se incluir ambas as datas, a da edição utilizada a da edição original.

O nome de jornais e revistas deve aparecer sempre em itálico e o nome dos artigos entre aspas. Os artigos devem ser identificados pelo volume, número e data do jornal ou revista. Por exemplo: Revista Inicia, “O Capital Privado e o Financiamento do Metrô”, Luís Marcelo Gallo, vol. 4 nº 15, agosto/setembro de 1998.

O conteúdo dos artigos é de exclusiva responsabilidade de seus autores. Todos os direitos editoriais são reservados. Nenhuma parte das publicações poderá ser reproduzida sem permissão prévia do conselho editorial ou sem contar com o crédito de referência.

A aceitação do trabalho para publicação implica na transferência de direitos do autor para a Revista INICIA, sendo assegurada a mais ampla divulgação da informação.