Rogério Rocha

UTILIZAÇÃO DA ROBÓTICA PEDAGÓGICA NO PROCESSO DE ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO DE COMPUTADORES

BELO HORIZONTE

Centro Federal e Educação Tecnológica de Minas Gerais - CEFET

2006

Rogério Rocha

UTILIZAÇÃO DA ROBÓTICA PEDAGÓGICA NO PROCESSO DE ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO DE COMPUTADORES

Dissertação apresentada ao curso de Mestrado em Educação Tecnológica da Diretoria de Pós-Graduação do Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG como requisito parcial à obtenção do título de Mestre em Educação Tecnológica.

Linha de Pesquisa III – Tecnologias da Informação e Educação

Orientador: Prof. Dr. José Wilson da Costa

BELO HORIZONTE

2006

O temor do senhor é o princípio da Sabedoria ...

Salomão

Bíblia Sagrada,

Livro de Provérbios

Capítulo 1 – Verso 7

Dedico esta dissertação à
minha amada esposa Kátia e
nossos queridos filhos
Samuel e Rebecca.

Agradecimentos

Ao Senhor nosso Deus, por tudo!

Ao professor José Wilson, que me fez compreender
o valor de se ter um orientador, na acepção mais nobre da palavra.

A UNA, através dos professores Johann, pró-reitor de graduação e Daniel Braga, coordenador interino do curso de Sistemas de Informação, pelo apoio recebido.

Ao professor Thiago Hofman e os alunos da turma de robótica, em especial a representante da turma, Flávia.

Minha equipe de apoio logístico, tecnológico, pedagógico e cinematográfico Célia e Tiago.

E especialmente, minha amada esposa Kátia, e nossos preciosos filhos Samuel e Rebecca, pelo amor paciente e compreensivo.

SUMÁRIO

RELAÇÃO DE FIGURAS

RELAÇÃO DE TABELAS

RESUMO

Este trabalho apresenta uma investigação da utilização da robótica pedagógica no processo de ensino-aprendizagem de programação de computadores. Seu objetivo é contribuir para a melhoria na qualidade do processo ensino-aprendizagem de conceitos relacionados à programação de computadores em virtude da problemática que envolve dificuldades na aprendizagem destes conceitos e que se reflete nos elevados níveis de reprovação nas disciplinas relacionadas a esta temática.

A pesquisa foi realizada por meio de um estudo de caso em uma instituição de ensino superior de Belo Horizonte na qual é ministrado um curso de sistemas de informação, com foco em uma disciplina que aplica a robótica pedagógica como instrumento de apoio ao processo de ensino-aprendizagem de programação de computadores. Foi adotada uma abordagem qualitativa que empregou como instrumentos de coleta de dados uma extensa observação de campo, entrevistas semi-estruturadas e pesquisa documental.

Os resultados verificados apresentam aspectos promissores, apesar das limitações observadas, na aplicação da robótica como mecanismo de fomento à interação entre os alunos e destes com o professor em um ambiente que favorece a construção do conhecimento. Finalmente é apresentada uma proposta para a utilização de maneira mais abrangente da robótica pedagógica.

Palavras-chave: Robótica Pedagógica; Linguagens de Programação de computadores; Construcionismo.

ABSTRACT

This paper presents a research of the application of the educational robotics in the teaching and learning processes of computer programming. It aims to contribute to the quality improvement of theses processes taking into consideration the difficulties in the learning of the concepts about the computer programming, resulting in high levels of fail in subjects related to this area at college.

The research was accomplished through a case study in a Brazilian college located in the city of Belo Horizonte. It is based on a subject of the course of Information Systems, in such college, in which the pedagogical robotics is used as an instrument to support the teaching and learning processes of computer programming.

A qualitative approach was adopted and the instruments for collecting data are extensive field observation, semi-structuralized interviews and documentary research.

The results show promising aspects in the application of robotics as an instrument to improve interaction of the students and between the students and the professor in a favorable environment for reaching the knowledge. In spite of this, some limitations were observed.

As a specific result a proposal for a more extensive use of the educational robotics is presented.

Key Words: Educational robotics; Computer programming languages; Construcionism.

1 INTRODUÇÃO

O presente trabalho tem como título a utilização da robótica pedagógica no processo de ensino-aprendizagem de programação de computadores.

O tema abordado foi a educação tecnológica, com ênfase nas tecnologias da informação e educação e seu problema central da pesquisa surgiu através da prática profissional do pesquisador em mais de 15 (quinze) anos de atuação no mercado da tecnologia da informação (TI) como programador, analista de sistemas, líder de equipes e gerente de desenvolvimento. Além da atuação docente em mais de 5 (cinco) anos lecionando disciplinas ligadas a TI em geral e a programação em particular, em instituições como a UNA e a PUC Minas, além da graduação em tecnologia de processamento de dados e a especialização em educação tecnológica. As observações realizadas durante esta trajetória tornaram possível identificar dificuldades na compreensão e aplicação de conceitos relacionados à programação de computadores em um razoável número de acadêmicos da área de tecnologia da informação.

Como conseqüência desta problemática, o objetivo deste trabalho foi contribuir para a melhoria na qualidade do processo ensino-aprendizagem de conceitos relacionados à programação de computadores, a partir da observação e discussão da aplicação de uma abordagem metodológica que utiliza a robótica pedagógica (RP). Esta abordagem propõe aulas em que os alunos são estimulados a discutir os resultados observados, retornando ao algoritmo e introduzindo melhorias em um ciclo de planejamento, execução e reflexão no qual o ator principal é o aluno e o aparato tecnológico atua como coadjuvante. Neste cenário, cabe ao professor fornecer as orientações para que o processo se desenvolva com eficácia, propondo correções de rota quando necessário. A partir destas premissas realizou-se uma observação nas interações ocorridas entre os alunos e o professor em contexto de sala de aula utilizando-se abordagem metodológica estudada.

Foi realizada ainda, uma reflexão sobre a aplicação da robótica pedagógica como artefato cognitivo no processo de ensino-aprendizagem de programação de computadores, tomando como base experimentos conduzidos por instituições de renome nacional e internacional como, por exemplo, o núcleo de informática aplicada à educação (NIED) mantido pela UNICAMP e o *Media Lab* mantido pelo *Massachussets Institute of Technology* (MIT).

A hipótese investigada foi que a introdução no ambiente da aula de um instrumento, o robô, que permita ao aluno observar o comportamento dos algoritmos (programas de computadores) desenvolvidos por ele e executados pelo robô e as interações deste com o ambiente em tempo real, pudesse favorecer outras formas de percepção e de estímulo, contribuindo efetivamente para a aprendizagem de técnicas de programação de computadores. As questões básicas de pesquisa foram: Quais os efeitos da introdução da robótica pedagógica em disciplinas de programação de computadores? Quais as abordagens metodológicas desejáveis na aplicação da robótica pedagógica? Qual o papel do professor neste processo?

Como fruto deste trabalho, a presente dissertação apresenta os seguintes capítulos:

No capítulo 2 o computador e sua arquitetura, são discutidos conceitos elementares da ciência da computação, a saber, o *hardware*, o *software* e o algoritmo, através de uma revisão histórica de sua gênese e processo evolutivo. A programação de computadores é apresentada, no capítulo 3, por meio de uma reflexão sobre o processo de desenvolvimento de programas de computador por meio das linguagens de programação e seus paradigmas. Também é apresentada uma discussão sobre o processo de ensino aprendizagem de programação de computadores. O quarto capítulo trata das concepções de aprendizagem, tomando como referência as teorias do conhecimento, o racionalismo, o empirismo e o interacionismo. Também é proposta uma reflexão sobre as teorias da aprendizagem tradicional, comportamentalista e interacionista. Na esfera de influência do interacionismo a discussão converge para o construtivismo e desagua no construcionismo de Papert, o referencial teórico básico deste trabalho de pesquisa.

A Robótica em geral e sua aplicação pedagógica em particular são discutidas no capítulo 5, onde os *kits MINDSTORMS LEGO* e sua metodologia típica de aplicação são apresentados, além de introduzir a questão das competições de robótica. O sexto capítulo trabalha as questões referentes à metodologia, a abordagem adotada, seus instrumentos, etapas e a delimitação do campo de pesquisa.

O capítulo de número 7 apresenta o estudo de caso e busca retratar e discutir, com profundidade e riqueza de detalhes as observações e propor uma reflexão à luz do referencial teórico adotado. Enquanto o oitavo e derradeiro capítulo apresenta as conclusões e considerações finais do pesquisador, como as intervenções e posturas que se apresentam, sob sua ótica, como adequadas.

Finalmente, como de praxe, são apresentadas as referências bibliográficas e os anexos.

2 O COMPUTADOR E SUA ARQUITETURA

2.1 O Computador

O termo "computador" usado no Português é uma tradução do Inglês "*computer*" que por sua vez refere-se a uma expressão em latim, "*computare"*, que até por volta do século XV era usada para denotar alguém que fazia cálculos.

O termo permaneceu associado à atividade humana até aproximadamente o meio do século XX, quando passou a ser aplicado a um dispositivo eletrônico e programável com capacidade de armazenar, recuperar e processar dados ROJAS & HASHAGEN (2000). Os computadores, de acordo com CORNACHIONE (2001), são equipamentos capazes de executar milhões, eventualmente bilhões, de instruções por fração de segundo. Esta capacidade é suportada por um conjunto de componentes eletrônicos, *hardware (HW)* que depende do gerenciamento dos programas de computador, *software (SW)*.

A figura 2 ilustra a empresa *North American Aviation* na cidade de *Los Angeles* no EUA no ano de 1962. A enorme quantidade de pessoas que se vê na figura 1 foi substituída na tarefa de computar por computadores.

De uso cotidiano, um computador é um equipamento eletrônico, já quase considerado um eletrodoméstico, geralmente associado a um monitor, um teclado e um mouse. Computadores podem ser utilizados para a digitação de textos, armazenamento de informações, processamento de dados, comunicação escrita ou falada ou para entretenimento. Em sentido mais amplo, um computador é qualquer equipamento ou dispositivo capaz de armazenar e manipular, lógica e matematicamente, quantidades numéricas representadas fisicamente.

2.2 História

No século XVIII foi criada uma maneira para que as máquinas de tecer produzissem padrões de cores diferentes representados através de cartões de papel perfurado, que eram tratados manualmente. Em 1801, Joseph Marie Jacquard (1752-1834) inventa um tear mecânico, com uma leitora automática de cartões. A máquina de tecer de Jacquard trabalhava tão bem que milhares de tecelões perderam o emprego com a automação, se rebelando e quase matando o inventor. CERUZZI (2003)

Charles Babbage (1792-1871), um professor de matemática de Cambridge, Inglaterra, tentava desenvolver uma máquina capaz de calcular polinômios por meio de diferenças. Enquanto projetava seu calculador diferencial, a idéia de Jacquard fez com que Babbage imaginasse uma nova e mais complexa máquina, o calculador analítico.

Esta máquina era extremamente semelhante ao computador atual. Sua parte principal seria um conjunto de engrenagens, o moinho, formando uma máquina de somar com precisão de 50 dígitos. As instruções seriam lidas de cartões perfurados. Babbage conseguiu, durante algum tempo, fundos para sua pesquisa, porém não conseguiu completar sua máquina no tempo prometido e não recebeu mais dinheiro. Hoje, partes de sua máquina podem ser vistas no Museu Britânico, que também construiu uma versão completa, utilizando as técnicas disponíveis na época. Junto com Babbage, trabalhou a jovem Ada Augusta, filha do poeta Lord Byron, conhecida como Lady Lovelace, ou Ada Lovelace. Ada foi a primeira programadora[1] da história, projetando a pedido de Babbage, programas para a máquina inexistente. Ada inventou os conceitos de sub-rotina, uma seqüência de instruções que pode ser usada várias vezes, *loop*, uma instrução que permite a repetição de uma seqüência de cartões, e do salto condicional, que permite saltar algum cartão caso uma condição seja satisfeita.

Ada Lovelace e Charles Babbage estavam avançados demais para o seu tempo, tanto que até a década de 1940, nada se inventou parecido com seu computador analítico. Até essa época foram construídas muitas máquinas mecânicas de somar, destinadas a controlar negócios (principalmente caixas registradoras) e algumas máquinas inspiradas na calculadora diferencial de Babbage, para realizar cálculos de engenharia (que não alcançaram grande sucesso).

Segundo ROJAS & HASHAGEN (2000), o primeiro computador eletro-mecânico foi construído por Konrad Zuse (1910–1995). Em 1936, esse engenheiro alemão construiu, a partir de relés que executavam os cálculos e dados lidos em fitas perfuradas, o Z-1. Foi na II Guerra Mundial que realmente nasceram os computadores atuais. A Marinha americana, em conjunto com a Universidade de Harvard, desenvolveu o computador Mark I, projetado pelo professor Howard Aiken, com base no calculador analítico de Babbage. O Mark I ocupava 120 m^3 aproximadamente, conseguindo multiplicar dois números de 10 dígitos em 3s.

---

1 Veja o item 2.4 “O Software ou programa de computador”

2.3 As gerações

Para fins didáticos, os computadores são divididos em gerações, como por exemplo, STALLINGS (2002), propõe o seguinte esquema:

- Primeira Geração, de 1945 a 1955, Válvulas. (Figura 3)
- Segunda Geração, de 1955 a 1965, Transistores. (Figura 5)
- Terceira Geração, de 1965 a 1980, Circuitos Integrados (CI). (Figura 6)

2.3.1 A primeira geração

Em segredo, o Exército Americano desenvolvia um projeto, chefiado pelos engenheiros
J. Presper Eckert e John Mauchy, cujo resultado foi o primeiro computador a válvulas, conforme figura 3. O *Eletronic Numeric Integrator And Calculator* (ENIAC) era capaz de fazer 500 multiplicações por segundo. Tendo sido projetado para calcular trajetórias balísticas, o ENIAC só foi terminado após o fim da guerra. ROJAS & HASHAGEN (2000)

Fonte: CAPRON e JOHNSON (2004)

No ENIAC, o programa era feito rearranjando a fiação em um painel[2]. Nesse ponto John von Newmann propôs a idéia que transformou os calculadores eletrônicos em "cérebros eletrônicos", modelar a arquitetura do computador segundo o sistema nervoso central. Para isso, eles teriam que ter 3 (três) características, a saber, codificar as instruções de uma forma possível de ser armazenada na memória do computador, von Newmann sugeriu que fossem usados uns e zeros, armazenar as instruções na memória, bem como toda e qualquer informação necessária à execução da tarefa e quando processar o programa, buscar as instruções diretamente na memória, ao invés de lerem um novo cartão perfurado a cada passo. Este é o conceito de "Programa Armazenado", cujas principais vantagens são: rapidez, e versatilidade. Assim, o computador programável que conhecemos hoje, no qual o programa e os dados estão armazenados na memória ficou conhecido como computador de von Newmann, conforme representado pela figura 4.

[2] Veja a figura 14 na página 26

Figura 4 – Arquitetura de von Newmann

Fonte: CORNACHIONE (2001, p.30)

2.3.2 A segunda geração

)

A indústria de computadores começou, nos anos 60, a projetar computadores usando transistores, figura 5, que eram menores e mais econômicos que as válvulas. Os computadores a válvulas da década de 40 eram máquinas imensas, caríssimas, instáveis (pois as válvulas se queimavam a uma taxa astronômica) e de capacidade computacional muito limitada; com a adoção de transistores, o computador começou a se tornar uma máquina viável.

O transistor foi inventado nos Laboratórios da *Bell Telephone* em dezembro de 1947 por Bardeen e Brattain e William Bradford Shockley, que foram laureados com o prêmio Nobel da Física em 1956. ROJAS & HASHAGEN (2000).

2.3.2 A terceira geração

Dois engenheiros americanos, Jack Kilby e Robert Noyce trabalhando independentemente concebem, em 1959 os primeiros circuitos integrados (CI). Componentes eletrônicos, formados por conjuntos de componentes isentos de interligações por fios os quais estariam montados e, inseridos em um bloco sólido. Este bloco pode ser constituído de múltiplas camadas de materiais que atuaram

como isoladores, retificadores, condutores e amplificadores, cujas funções elétricas são interligadas por um processo de corte de certas áreas das mesmas. ROJAS & HASHAGEN (2000).

Segundo CORNACHIONE (2001), a integração pode ser classificada da seguinte forma:

- Integração em PEQUENA escala (1965)

Até 100 transistores em um chip

- Integração em MÉDIA escala (1971)

De 100 a 3.000 transistores em um chip

- Integração em LARGA escala (1977)

De 3.000 a 100.000 transistores em um chip

- Integração em MUITO LARGA escala (1978)

De 100.000 a 100.000.000 transistores em um chip

- Integração em EXTREMEMANTE LARGA escala

Acima de 100.000.000 transistores em um chip

Em 1976 foram fabricadas pastilhas de silício com menos de 0,5 cm com capacidade para armazenar tudo o que havia no ENIAC. Antes de 1976 os componentes de informática tinham o preço elevado sendo acessível a pouquíssimas empresas existentes em países mais desenvolvidos, porém após a fabricação das pastilhas de silício (*chip*), houve uma queda súbita nos preços influenciando diretamente na tecnologia associada à informática e a robótica. ROJAS & HASHAGEN (2000).

O computador deve ser encarado como uma ferramenta de auxílio à resolução de problemas que podem ser de natureza cientifica, comercial ou qualquer outra. Os computadores contemporâneos têm como característica a incapacidade de raciocínio, porém são dotados de uma enorme capacidade de executar instruções com alta velocidade e precisão. Ao conjunto destas instruções, denomina-se programa. Um programa de computador é uma seqüência de instruções que descreve precisamente algumas operações a serem executadas pelo hardware. Cada programa deve ter um objetivo, sendo por seu intermédio que o computador é capaz de "entender" a tarefa a ser executada, ou seja, o programa age como interface entre homem e máquina.

2.4 O Software ou programa de computador

Um programa de computador ou *software (SW)* é composto por uma seqüência lógica de instruções, que é interpretada e executada por um processador. Um programa pode ser executado por qualquer

dispositivo capaz de interpretar e executar as instruções de que é formado. O dispositivo mais conhecido que dispõe de um processador é o computador. Existem outras máquinas programáveis, como telefone celular, máquinas de automação industrial, calculadora, etc.

Um programa é codificado através de uma linguagens de programação, ou instruções do processador. Qualquer computador moderno tem uma variedade de programas que fazem diversas tarefas. Segundo Guimarães e Lages (1985) o *software* contem as instruções que são executadas pelo hardware, possibilitando obter uma das peculiaridades dos computadores, sua capacidade de executar diversos tipos diferentes de tarefa, pela substituição do programa em execução. A título de ilustração, usando o mesmo computador um atarefado mestrando pode controlar seu orçamento pessoal, manter a lista do supermercado, solicitar uma pizza e principalmente redigir a dissertação, simplesmente pela substituição do programa que utiliza. Os programas de computadores são desenvolvidos por pessoas com conhecimentos técnicos em programação de computadores, os programadores ou desenvolvedores. Eles utilizam as linguagens de programação para criar os programas que serão aplicados para resolver os mais diversos tipos de problemas, daí a utilização do termo aplicativo como sinônimo para programa de computador ou software, O'BRIEN (2004).

2.5 O Algoritmo

Em muitos casos, antes de se codificar o programa de computador em uma linguagem de programação, o programador desenvolve um planejamento com vistas a refinar as soluções lógicas adotadas pelo programa, elaborando um algoritmo.

Um algoritmo é uma seqüência finita e não ambígua de instruções para solucionar um problema. Mais especificamente, em matemática, constitui o conjunto de processos, e símbolos que os representam, para efetuar um cálculo. ASCENCIO e CAMPOS (2002)

A palavra algoritmo tem origem no sobrenome, *Al-Khwarizmi*, do matemático persa do século IX, *Abu Ja'far Mohamed, ou Muhammad, ibn Musa al-Khwarizmi* (780 - 850), cujas obras foram traduzidas no ocidente cristão no século XII, tendo uma delas recebido o nome "*Algorithmi de numero indorum*", sobre os algoritmos usando o sistema de numeração decimal (indiano). UNIVERSITY OF ST ANDREWS (2006).

Os algoritmos podem repetir passos (fazer iterações) ou necessitar de decisões (tais como comparações ou lógica) até que a tarefa seja completada. Um algoritmo corretamente executado não irá resolver um problema se o algoritmo estiver incorreto ou não for apropriado ao problema. Este

exemplo ilustra um algoritmo, do ano 830, para a execução de uma subtração de números inteiros. Do número 12025 subtrair 3604. Observe a figura 7.

A operação era iniciada pela esquerda.

Operação I

De 12 tirando 3 restam 9;

Cancelamos os algarismos considerados

Escrevemos o resto obtido em cima do minuendo.

De 90 tirando 6 restam 84.

Operação II

A diferença obtida é escrita sobre o minuendo,

Os algarismos que formavam os termos de subtração aparecem cancelados.

Operação III

De 8425 tirando 4 restam 8421

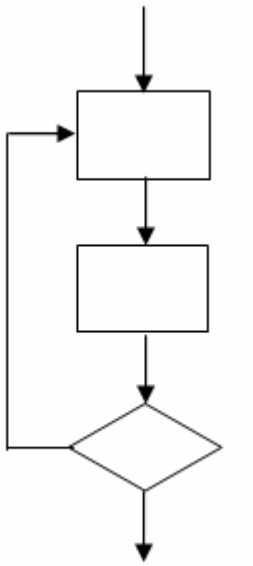
8.421 é a diferença entre os números dados (12.025 – 3.604). Era assim que *Al-Khwarizmi* um dos sábios mais notáveis do Século IX, realizava uma subtração de números inteiros. COMPANHIA DOS NUMEROS (2006).

Diferentes algoritmos podem realizar a mesma tarefa usando um conjunto diferenciado de instruções em mais ou menos tempo, espaço ou esforço do que outros. Um algoritmo não representa, necessariamente, um programa de computador, e sim os passos necessários para realizar uma tarefa.

Figura 8 – Fluxograma

Existem diversas formar de se representar um algoritmo, duas das mais difundidas entre os programadores são o fluxograma e a pseudo-linguagem, como o Portugol, uma combinação entre o Português e o Algol. Guimarães e Lages (1985, p.19)

Um fluxograma auxilia a explicar a seqüência de instruções em algoritmos e programas. Na figura 8, um retângulo representa uma instrução, uma seta indica a próxima instrução a ser executada e um losango indica uma condição que interfere no fluxo do algoritmo.



A Pseudo-linguagem é notação que se assemelha a uma linguagem de programação, mas que também possibilita ao programador concentrar-se no problema a ser modelado sem "se prender" a uma linguagem de programação específica.

Fonte: Autoria Própria

Um algoritmo pode ser executado por um ser humano ou por um computador. Para resolver um problema no computador é necessário que seja primeiramente encontrada uma maneira de descrever este problema de uma forma clara e precisa. É preciso que encontremos uma seqüência de passos que permitam que o problema possa ser resolvido de maneira automática e repetitiva.

Um algoritmo é, num certo sentido, um programa potencial, dizendo de outra forma, um programa é um algoritmo concretizado. No entanto, os programas são, à exceção dos menores, visualizados mais facilmente como uma coleção de algoritmos menores combinados de um modo único - da mesma forma que uma casa é construída a partir de componentes. Dessa forma, um algoritmo é uma descrição de como um computador pode ser levado a executar uma operação simples e específica, como, por exemplo, uma ordenação. Um programa, por outro lado, é uma entidade que na verdade implementa uma ou mais operações de forma que seja útil para as pessoas.

Considerando que cada computador possui um conjunto limitado de instruções que é capaz de executar e que tipicamente tais instruções são representadas por seqüências de zeros e ums, os chamados números binários (exemplo na figura 9), para serem executados por computadores, os algoritmos devem ser traduzidos para esta linguagem. A conversão direta das linguagens naturais (inglês, português, etc.) para a linguagem binária é muito complexa, daí a introdução de linguagens de programação, que posteriormente serão convertidas para a linguagem que pode ser diretamente interpretada pelo computador, o chamado código de máquina.

Figura 9 – Números Binários

```
00000000011100000001
```

```
00000001100100000011
00000010000100000110
00000011000000000000
00000100000000000000
```

Fonte: Autoria Própria

Os computadores estão sendo usados em uma infinidade de diferentes áreas, desde o controle de usinas nucleares à armazenagem de registros de talões de cheques pessoais. Por causa desta grande diversidade, linguagens de programação com metas muito diferentes tem sido desenvolvidas SEBESTA (2003, p.19).

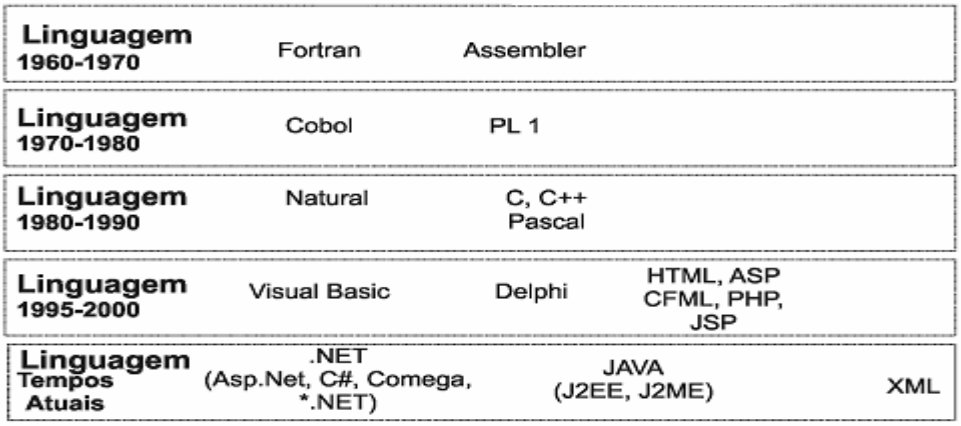
3 AS LINGUAGENS DE PROGRAMAÇÃO

Uma linguagem de programação é um método padronizado para expressar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Uma linguagem permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias. O código fonte (*source code*) é o conjunto de palavras escritas de forma ordenada, contendo instruções em uma determinada linguagem de programação.

A abstração é um conceito fundamental no projeto de linguagens de programação contemporâneas. De acordo com SEBESTA (2003) abstração significa a capacidade de definir e, depois, de usar estruturas ou operações complicadas de uma maneira que permita ignorar muitos dos detalhes inerentes ao *hardware*, concentrando-se na solução do problema através do *software*. Há diversas linguagens de programação disponíveis no mercado profissional de desenvolvimento de software e no meio acadêmico. Podem ser citadas “Basic”, “COBOL”, “Fortran”, “Pascal”, “C”, “C++”, “JAVA”, “C#”, “LOGO”, “LISP”, “PROLOG”, entre diversas outras, cada qual com suas aplicações e características peculiares. FARRER et al. (1989).

Um esquema da evolução das linguagens de programação de computadores é apresentado na fig.10

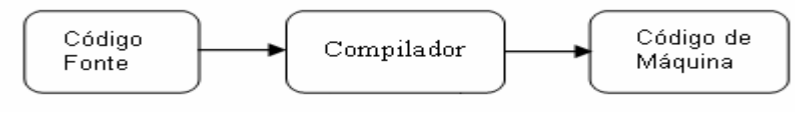
Figura 10 – Evolução das linguagens de programação de computadores



Fonte: Autoria Própria

O código fonte escrito em uma linguagem de programação pode ser convertido, ou traduzido, em código de máquina por compilação ou interpretação. O método utilizado para traduzir o código fonte em código de máquina, para só depois executar (ou rodar, como se diz no jargão da computação) o programa, é a compilação, representada na figura 11.

Figura 11 – Processo de compilação



Fonte: Autoria Própria

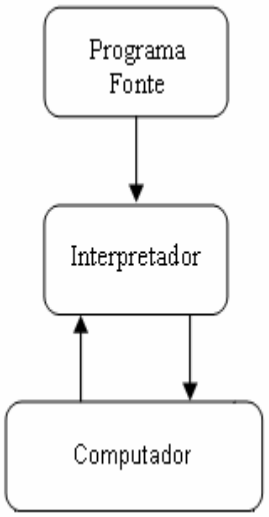
Um compilador, segundo MIZRAHI (1990, p.3) é um programa de computador que traduz programas fontes escritos em uma determinada linguagem de programação para programas em linguagem de máquina. Depois de compilado, este pode ser executado independente do compilador e do programa original. As linguagens Pascal e C são tipicamente compiladas. Um interpretador é um programa que executa outros programas escritos em alguma linguagem de programação, conforme demonstra a figura 12.

A execução de um programa interpretado é em geral mais lenta que o programa compilado. Por outro lado, o uso de programas interpretados permite que trechos de código possam ser trocados por novos facilmente, fazendo com que o programa fonte possa mudar durante sua execução.

Este é um dos grandes motivos de se usar programas interpretados em sistemas especialistas. Duas linguagens para as quais podemos encontrar interpretadores são *Lisp* e *Prolog*.

Figura 12

Processo de interpretação



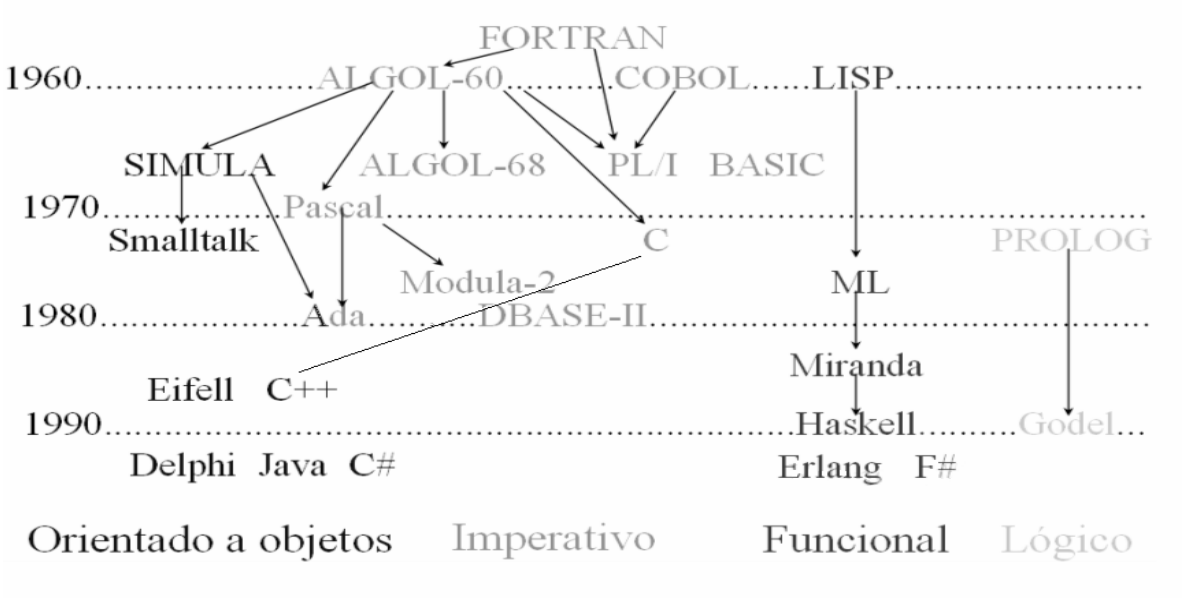
Fonte: Autoria Própria

Tomando como referência a visão de que os programas de computadores têm como finalidade a resolução de problemas, as diferentes formas de abordar tais problemas e de formular as respectivas soluções são conhecidas como paradigmas de programação, representados na figura 13.

3.1 Os paradigmas de programação

Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns. Segundo Papert (1991, p. 8) o paradigma é a maneira como o programador pensa sobre a tarefa.

Figura 13 – As linguagens e seus paradigmas



Fonte: BARANAUSKAS (1998)

No contexto educacional, segundo Baranauskas (1998), as linguagens de programação podem ser classificadas segundo a ótica dos "meios" diferentes onde os problemas são representados e resolvidos e esta abordagem leva a uma reflexão sobre metodologias de uso de linguagens de programação no contexto educacional.

Os paradigmas de programação e respectivos conceitos de base

- Paradigma imperativo:
  Conceitos: estado, atribuição, seqüenciação.
- Paradigma orientado a objetos:
  Conceitos: objeto, mensagem.
- Paradigma funcional:
  Conceitos: função, aplicação, avaliação.
- Paradigma lógico:
  Conceitos: relação, dedução.

3.1.1 O Paradigma Imperativo

O paradigma imperativo define programas centrados no conceito de estado modelado por variáveis, através de instruções que as manipulam. Também denominado procedural, por incluir sub-rotinas ou procedimentos como formas de estruturação dos programas produzidos. Este paradigma foi o primeiro a surgir, inspirado na arquitetura de von Newmann, e nos dias atuais, ainda é o dominante segundo WATT (1990).

Segundo o paradigma imperativo, programar o computador significa "dar-lhe ordens" que são executadas seqüencialmente. O conceito central para representação da solução do problema é o conceito de "variável" como uma abstração para uma posição de memória na máquina, para a qual se pode atribuir um valor. O fluxo de controle da execução pela máquina é ditado por sequenciação e por comandos de repetição. BARANAUSKAS (1998). Nos primórdios da utilização dos computadores, por volta de 1940-1950, a programação era feita em baixo nível, ou seja, diretamente nas instruções do hardware conforme ilustrado na figura 14. Esta abordagem tornava muito baixa a produtividade na produção de programas de computador, pois somente estavam aptos a desenvolver programas para um determinado tipo de máquina (arquitetura), aqueles profissionais que conhecessem profundamente o projeto daquela arquitetura computacional. Além do fato de que a detecção e correção de eventuais erros de codificação eram extremamente ineficientes. CAPRON e JOHNSON (2004).

Provavelmente tenha sido este o maior fator motivador para o desenvolvimento das chamadas linguagens de montagem e seus montadores. Além disso, as aplicações numéricas da época requeriam o uso de certas facilidades que não estavam incluídas no hardware das máquinas de então, (números reais, acesso a elementos de um conjunto por seu índice, por exemplo) surgindo daí a criação de linguagens de mais alto nível que suportassem tais recursos.
Podem ser citadas BASIC, COBOL, C e PASCAL, entre outras, como exemplos comerciais e científicos de linguagens de programação procedurais ou imperativas. A literatura especializada apresenta como a mais provável razão que poderia justificar crescimento das linguagens procedurais, o papel histórico do uso do computador em aplicações numéricas.

3.1.1.1 A Programação Estruturada

A programação estruturada, inserida no paradigma imperativo, estabelece uma disciplina de desenvolvimento de algoritmos que facilita a compreensão de programas através do número restrito de mecanismos de controle da execução de programas. DEITEL e DEITEL (2003, p.61)

Qualquer algoritmo, independentemente da área de aplicação, de sua complexidade e da linguagem de programação na qual será codificado, pode ser descrito por estes mecanismos básicos. O princípio básico de programação estruturada é que um programa é composto por blocos elementares de código que se interligam através de três mecanismos básicos, que são seqüência, seleção e iteração. Cada uma destas construções tem um ponto de início (o topo do bloco) e um ponto de término (o fim do bloco) de execução.

- A seqüência implementa os passos de processamento necessários para descrever qualquer programa. Por exemplo, um segmento de programa da forma
  "*faça primeiro a tarefa A e depois a tarefa B*";
- A seleção especifica a possibilidade de selecionar o fluxo de execução do processamento baseado em ocorrências lógicas. A principal forma é a construção "*se*", que permite representar fluxos da forma "*se a condição lógica x for verdadeira, faça a tarefa A; senão (isto é, se a condição x for falsa), faça a tarefa B.*";
- A Iteração permite a execução repetitiva de segmentos do programa. Na forma básica de repetição ou *loop*, uma condição lógica é verificada. Caso seja verdadeira, o bloco de tarefas associado ao comando é executado. A condição é então reavaliada; enquanto for verdadeira, a tarefa é repetidamente executada.

Discussões didáticas sobre a programação estruturada podem ser encontradas em KERNIGHAN e RITCHIE (1986), GUIMARAES e LAGES (1985), FARRER et al. (1989), MIZRAHI (1990) e DEITEL e DEITEL (2005).

3.1.2 O Paradigma Orientado por Objetos

A idéia básica do paradigma orientado a objetos é imaginar que programas simulam o mundo real: um mundo povoado de objetos. Dessa maneira, linguagens baseadas nos conceitos de simulação do mundo real devem incluir um modelo de objetos que possam enviar e receber mensagens e reagir a mensagens recebidas. Esse conceito é baseado na idéia de que no mundo real freqüentemente usam-se objetos sem precisar-se conhecer como eles realmente funcionam. Assim, programação orientada a objetos fornece um ambiente onde múltiplos objetos podem coexistir e trocar mensagens entre si. MIZRAHI (2006).

Na programação orientada por objetos (POO), as unidades de programa são objetos. Por exemplo, desde uma constante numérica até um sistema para manipular arquivos, são todos objetos. Mensagens possibilitam a comunicação entre os objetos e é através delas que uma operação de um objeto é requisitada. Um método especifica a reação de um objeto a uma determinada mensagem recebida correspondente àquele método.

Esses conceitos são exemplificados pela representação em POO da expressão 21 + 2. O objeto receptor é o número 21, para o qual é enviada a mensagem "+ 2". Essa mensagem, que se constitui do método “+” e do objeto 2, passa este objeto “2” para o método "+" que pertence ao objeto “21”. O código desse método usa o objeto 2 para construir um novo objeto, o 23".
SEBESTA (2003, p. 443).

Os conceitos mais elementares da POO são: a “Classe” que representa um conjunto de objetos com características afins. Uma classe define o comportamento dos objetos, através de métodos, e quais estados ele é capaz de manter, através de atributos. Exemplo de classe: Os seres humanos. O “Objeto” que é uma instância de uma classe. Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos. Exemplo de objetos da classe Humanos: João, José, Maria e a “Mensagem” que é uma chamada a um objeto para invocar um de seus métodos, ativando um comportamento descrito por sua classe. DEITEL e DEITEL (2003).

Outro conceito fundamental é a “Herança” que é o mecanismo pelo qual uma classe (sub-classe) pode estender outra classe (super-classe), aproveitando seus comportamentos (métodos) e estados possíveis (atributos).

Há Herança múltipla quando uma sub-classe possui mais de uma super-classe. Essa relação é normalmente chamada de relação "é um". Um exemplo de herança: Mamífero é superclasse de Humano. Ou seja, um Humano é um mamífero. A “Associação” que é o mecanismo pelo qual um objeto utiliza os recursos de outro. Pode tratar-se de uma associação simples "usa um" ou de um acoplamento "parte de". Por exemplo: Um humano usa um telefone. A tecla "1" é parte de um telefone. DEITEL e DEITEL (2003).

Deve-se também citar o “Encapsulamento” que consiste na separação de aspectos internos e externos de um objeto. Este mecanismo é utilizado amplamente para impedir o acesso direto ao estado de um objeto (seus atributos), disponibilizando externamente apenas os métodos que alteram estes estados. Exemplo: você não precisa conhecer os detalhes dos circuitos de um telefone para utilizá-lo. A carcaça do telefone encapsula esses detalhes, provendo a você uma interface mais amigável (os botões, o monofone e os sinais de tom) e o “Polimorfismo” que permite que uma referência de um tipo de uma super-classe tenha seu comportamento alterado de acordo com a instância da classe filha a ela associada. O polimorfismo permite a criação de super-classes abstratas, ou seja, com métodos definidos (declarados) e não implementados, onde a implementação ocorre somente nas sub-classes não abstratas. MIZRAHI (2006).

De acordo com SEBESTA (2003), o conceito de programação orientada a objetos tem suas raízes na SIMULA 67, mas não foi amplamente desenvolvido até que a evolução do Smaltalk, que alguns consideram a única linguagem puramente orientada a objeto.

Em relação ao paradigma procedural, a diferença é mais de metodologia quanto à concepção e modelagem do sistema. Grosso modo, uma aplicação é estruturada em módulos (classes) que agrupam estados (atributos) e operações (métodos) que atuam sobre os atributos. Classes podem ser estendidas e/ou usadas como tipos (cujos elementos são objetos). Tem tido muita aceitação comercial desde a segunda metade da década de 1990.

Exemplos de linguagens bem sucedidas no mercado coorporativo C++, Java, Object Pascal (Delphi), C#, entre outras. Discussões didáticas sobre a programação orientado por objetos, podem ser encontradas em MIZRAHI (2006), através da linguagem C++ e DEITEL e DEITEL (2003) através da linguagem Java.

Tanto o paradigma imperativo quanto o orientado por objetos foram concebidos com vistas a um uso eficiente de computadores baseados na arquitetura da von Newmann, o que é visto por alguns como uma restrição desnecessária ao processo de desenvolvimento de software e que tem fomentado a proposição de outras bases para projetos de linguagens. SEBESTA (2003).

3.1.3 O Paradigma Funcional

O paradigma funcional de programação surgiu com o desenvolvimento da linguagem de *List Processing* (Lisp) por John McCarthy em 1958. Lisp, vide exemplo na figura 15, foi projetada numa época em que só existia processamento numérico, para atender aos interesses dos grupos de Inteligência Artificial no processamento de dados simbólicos. BARANAUSKAS (1998)

Programação Funcional é um paradigma de programação que trata a computação como uma avaliação de funções matemáticas. Uma função, neste sentido, pode ter ou não ter parâmetros e um simples valor de retorno. Os parâmetros ou argumentos são os valores de entrada da função, e o valor de retorno é o resultado da função. A definição de uma função descreve como a função será avaliada em termos de outras funções. Por exemplo, a função *f(x) = x2 + 2* é definida em termos de funções de exponenciação e adição. Do mesmo modo, a linguagem deve oferecer funções básicas que não requerem definições adicionais. A principal característica do meio funcional é "imitar" o comportamento de funções. Assim, no meio funcional, o computador atua como uma máquina que avalia funções e o programa consiste da definição e composição de funções. Escrever um "programa", segundo esse paradigma, envolve definir funções e aplicação de funções para o problema em questão e o processo de "execução" pela máquina consiste em avaliar as aplicações das funções envolvidas. O computador assume o papel de uma "máquina funcional".

A programação funcional pode ser contrastada com a programação imperativa. Na programação funcional parecem faltar diversas construções freqüentemente consideradas essenciais em linguagens imperativas, como C ou Pascal. Por exemplo, em uma programação estritamente funcional, não há alocação explícita de memória, nem declaração explícita de variáveis. No entanto, essas operações podem ocorrer automaticamente quando a função é invocada; a alocação de memória ocorre para criar espaço para os parâmetros e para o valor de retorno, e a declaração ocorre para copiar os parâmetros dentro deste espaço recém-alocado e para copiar o valor de retorno de volta para dentro da função chamadora. COUSINEAU e MAUNY (1998).

Ambas as operações podem ocorrer nos pontos de entrada e na saída da função, então efeitos colaterais no cálculo da função são eliminados. Ao não permitir efeitos colaterais em funções, a linguagem oferece transparência referencial. Isso assegura que o resultado da função será o mesmo para um dado conjunto de parâmetros não importando onde, ou quando, seja avaliada. Transparência referencial facilita muito ambas as tarefas de comprovar a correção do programa e automaticamente identificar computações independentes para execução paralela. SEBESTA(2003).

Segundo COUSINEAU e MAUNY (1998) os laços, outra construção de programação imperativa, está presente através da construção funcional mais geral de recursão. Funções recursivas invocam-se a si mesmas, permitindo que uma operação seja realizada várias vezes. Na verdade, isso prova que laços são equivalentes a um tipo especial de recursão chamado recursão reversa. Recursão em programação funcional pode assumir várias formas e é em geral uma técnica mais poderosa que o uso de laços. Por essa razão, quase todas as linguagens imperativas também a suportam, sendo FORTRAN 77 e COBOL exceções notáveis. A figura 15 apresenta um exemplo de um programa em LISP, que define o fatorial de x.

Figura 15 – Exemplo de um programa em LISP, que define o fatorial de x

```
> (defun fact (x)
    (if (> x 0)
(* x (fact (- x 1)))
        1
      ) )

    FACT
> (fact 5)
    120
```

Fonte : SEBESTA (2003, p.552).

A literatura técnica especializada apresenta como as principais limitações para o paradigma funcional as implementações ineficientes e os mecanismos primitivos de entrada e saída e formatação. SEBESTA (2003, p.552).

3.1.4 O Paradigma Lógico

Programação em lógica é uma teoria que representa um modelo abstrato de computação sem relação direta com o modelo von Newmann de máquina. Prolog (*Programming in Logic*), surgiu no início dos anos 70, dos esforços de Robert Kowalski, Maarten van Emden e Alain Colmerauer e é a linguagem de programação desenvolvida em máquina seqüencial, que mais se aproxima do modelo de computação de programação em lógica. WATT (1990).

O enfoque do paradigma da programação em lógica para se representar um problema a ser resolvido no computador, consiste em expressar o problema na forma de lógica simbólica. Um processo de inferência é usado pela máquina para produzir resultados. BARANAUSKAS (1998). Uma das características principais das linguagens para programação em lógica é sua semântica declarativa. A idéia por trás dessa semântica é que existe uma maneira de determinar o significado de cada declaração que não depende de como a declaração seria usada para resolver o problema. Isto é, o significado de uma dada proposição num programa é determinado a partir da própria proposição, enquanto que, em linguagens imperativas a semântica de um comando requer informações que não estão contidas ou não estão explicitadas no comando.

No meio gerado pela programação em lógica, um programa não contém instruções explícitas à máquina. Em vez disso, ele estabelece "fatos" e "regras" sobre a área do problema como um conjunto de axiomas lógicos, que são "interpretados" como "programas".

Segundo o modelo de programar proposto por Prolog, o significado de um "programa" não é mais dado por uma sucessão de operações elementares que o computador supostamente realiza, mas por uma base de conhecimento a respeito de certo domínio e por perguntas feitas a essa base de conhecimento, independentemente. Dessa maneira, Prolog pode ser visto como um formalismo para representar conhecimento a respeito do problema que se quer resolver, de forma declarativa (descritiva). Existe por trás do programa uma máquina de inferência, em princípio "escondida" do programador, responsável por "encontrar soluções" para o problema descrito. Na prática, inclui características imperativas, por questão de eficiência, sendo tipicamente usado em prototipação em geral, sistemas especialistas e banco de dados.

Alguns pesquisadores, segundo SEBESTA (2003), reivindicam um grande número de vantagens no ensino de Prolog a jovens. Primeiro é possível fazer a introdução a informática usando o paradigma lógico. Isso tem como efeito colateral de ensinar lógica, o que pode resultar em pensamento e expressão mais claros. Pode ajudar os estudantes a aprender uma variedade de matérias, como, por exemplo, resolver equações matemáticas, lidar com a gramática de linguagens naturais e entender as regras e a ordem do mundo físico.

A atividade de programação de computadores é considerada uma atividade de resolução de problemas, onde as linguagens representam, através de seus diferentes paradigmas, os meios onde os problemas devem ser resolvidos.

Assim, resolver um problema nos paradigmas citados envolve "moldar" o problema segundo as "entidades" representativas de cada paradigma, comandos que são executados passo a passo pela máquina virtual, se o paradigma for o procedural, funções, que são aplicadas a certos argumentos e "retornam" valores se o paradigma for o funcional, objetos que se comunicam se o paradigma for orientado a objetos e proposições assumidas verdadeiras sobre determinado domínio de conhecimento, se o paradigma for o da programação em lógica. BARANAUSKAS (1998).

Programar nos diferentes paradigmas significa, portanto, representar, segundo modelos diferentes, a solução do problema a ser resolvido na máquina. Cada linguagem que suporta determinado paradigma representa, portanto, um "meio" onde o problema é "resolvido". Enquanto "meio de expressão" e de "comunicação" com a máquina, a linguagem e, indiretamente o seu paradigma, "moldam" a representação do problema a ser resolvido. Assim, na atividade de programar, mudar de paradigma significa muito mais do que conhecer as entidades sintáticas e semânticas da nova linguagem, o processo de pensamento também deve ser mudado, ajustando-se ao novo meio de representação do problema.

3.2 O Processo de ensino e aprendizagem de programação de computadores

Na sociedade da informação, novos conceitos são introduzidos em ritmo acelerado e demandadas novas habilidades e competências, tanto no ambiente empresarial quanto no acadêmico. Segundo CASTELLS (1999), a economia informacional (informacionalismo) apresenta-se de formas distintas dependendo do contexto, porém com características comuns, com sua lógica organizacional alicerçada no novo paradigma tecnológico. Esta nova lógica organizacional, que apresenta como marco histórico os anos 70, abriu as portas das organizações para a tecnologia da informação e um primeiro fruto desta nova lógica foi a mudança da abordagem de produção em massa para a produção flexível, que estimulou as grandes corporações a interagir em maior escala com as pequenas empresas. Também se manifestaram os novos métodos de gerenciamento, como, por exemplo, o *toyotismo* que visa reduzir incertezas, tornando o processo de produção mais flexível.

A estrutura social tem como seu ponto central o trabalho que é diretamente afetado pelo paradigma informacional. As teorias do informacionalismo (pós-industrialismo) usam como prova empírica a mudança do enfoque da produção de produtos para a prestação de serviços e a ascensão das funções administrativas e especializadas, além do crescente conteúdo de informações no trabalho nas economias mais avançadas, dentre de uma conformação multicultural.

As sociedades podem ser consideradas informacionais porque organizam seu sistema produtivo em torno de princípios de maximização da produtividade baseada em conhecimento, por intermédio do desenvolvimento e difusão da TI e pelo atendimento dos pré-requisitos para sua utilização (RH e infra-estrutura de comunicação) e por uma migração das atividades indústrias para a prestação de serviços, que se apresenta como uma expressão de difícil definição, pois o intangível assume novas formas, como por exemplo, softwares, mídias, etc.

Podem ser citadas como exemplos de corporações do ramo de software que desenvolvem e comercializam programas de computadores para solução de problemas científicos, comerciais, industriais e pessoais, a *ORACLE, SUN, IBM e MICROSOFT*, entre outras. Estes grandes *players* do mercado movimentam cifras anuais superiores ao PIB de muitos países em desenvolvimento.

A automação aumenta enormemente a importância do cérebro humano no processo de trabalho, pois o trabalhador atuante é o agente necessário a empresa em rede e a inovação agrega valor. Existem teorias que postulam que os empregos tendem a migrar de atividades rotineiras que podem ser automatizadas para atividades que exigem análise e os empregos tradicionais tendem a ser substituídos por relações flexíveis entre profissionais altamente especializados e corporações extremamente flexíveis. Este movimento que catalisa o processo de adaptação aos novos modos de organização do trabalho, onde o trabalhador "taylorizado" que se limitava à execução de tarefas prescritas vai sendo gradualmente substituído por um perfil mais autônomo de quem se espera a capacidade de tomadas de decisões, segundo FERRETI (1994), propiciou o surgimento de várias profissões, dentre elas, aquelas de desenvolvimento de programas de computador.

Neste cenário, ganha valor o processo de ensino e aprendizagem de programação de computadores. De acordo com STAHL (1991, p.5) o uso dos computadores está obrigando a repensar como se realiza a aquisição do conhecimento e a tratar o processo ensino-aprendizagem numa abordagem construtiva, na qual os alunos criam, exploram e integram conhecimento. Instrução em programação pode ser caracterizada como um pensamento de alta ordem, mas grande parte da energia de aprender a programar é gasto em aprender a sintaxe e estrutura das linguagens de programação e isso deve interferir na meta inicial de desenvolvimento de habilidades em resolução de problemas. JOHANSON (1988).

Segundo a estrutura curricular constante nas Diretrizes Curriculares do MEC, a matéria de programação faz parte da área de formação básica em Ciência da Computação, juntamente com as matérias "Computação e Algoritmos" e "Arquitetura de Computadores". Seu conteúdo abrange, além do ensino de linguagens de programação propriamente ditas, os conceitos, os princípios e os modelos de programação e o estudo de estruturas de dados e de métodos de classificação e pesquisa de dados. Conforme consta na caracterização da matéria programação, entendida como programação de computadores, ela é uma atividade voltada à solução de problemas. Nesse sentido, ela está relacionada com uma variada gama de outras atividades como especificação, projeto, validação, modelagem e estruturação de programas e dados, utilizando as linguagens de programação como ferramentas.

Ao contrário do que se apregoava há alguns anos, a atividade de programação deixou de ser uma arte para se tornar uma ciência, envolvendo um conjunto de princípios, técnicas e formalismos que visam o desenvolvimento de produtos de software bem estruturados e confiáveis. Cite-se, dentre estes, os princípios da abstração e do encapsulamento e as técnicas de modularização e de programação estruturada. Portanto, o estudo de programação não se restringe ao estudo de linguagens de programação. As linguagens de programação constituem-se em uma ferramenta de concretização de produto de software, que representa o resultado da aplicação de uma série de conhecimentos que transformam a especificação da solução de um problema em um programa de computador que efetivamente resolve aquele problema. SANTOS e COSTA (2005)

Durante o processo de ensino-aprendizagem de fundamentos de programação, nota-se que grande parte dos alunos apresenta dificuldades em assimilar as abstrações envolvidas. Esta disciplina tem um dos maiores índices de reprovação em todas as instituições de ensino brasileiras, o que torna ponto de reflexão por parte dos professores preocupados com a melhoria da qualidade no processo, ratificando a necessidade de alterações didáticas e metodológicas de apresentação. Um exemplo de abordagem alternativa que explora o lúdico pode ser encontrado em ROCHA (2003) que de forma indireta trabalha a teoria dos modelos mentais proposta por JOHNSON-LAIRD (1983).

Esta preocupação pode ser observada através de uma reflexão sobre trabalhos recentemente publicados sobre as pesquisas relativas ao processo de ensino-aprendizagem de programação no Brasil a partir da análise dos anais do maior evento nacional que envolve Educação e Informática, o *Workshop* de Educação em Informática (WEI) e do Simpósio Brasileiro de Informática na Educação (SBIE). Tanto o WEI quanto o SBIE fazem parte do conjunto de eventos anuais promovidos pela Sociedade Brasileira de Computação (SBC). RAPKIEWICZ e PEREIRA (2004).

É possível encontrar em trabalhos de publicação recente no âmbito nacional, como por exemplo, CHAVES DE CASTRO *et al* (2003) e DELGADO *et al* (2004), o registro da observação de um comportamento apático em muitos dos alunos que não conseguem desenvolver o raciocínio lógico necessário para o posterior desenvolvimento de programas, o que tende a levar a evasão e a reprovação. Segundo KOLIVER, DORNELES e CASA (2004) a questão chave no processo é justamente como motivar o aluno, fazê-lo tomar gosto pelo aprendizado e procurar superar suas dificuldades como a falta de habilidades matemáticas.

De acordo com NOBRE e MENEZES (2002), O modo "tradicional" de apresentar o conteúdo de tal disciplina se faz de forma expositiva concentrando-se na solução de problemas. O professor apresenta o conteúdo com o emprego de uma pseudo-linguagem[3], seguido por alguns exemplos e propõe exercícios para a turma. Para BORGES (2000), o modo "tradicional" não consegue facilmente motivar os alunos a se interessar pela disciplina, pois não é clara para os alunos a importância de certos conteúdos para sua formação.

RAPKIEWICZ e PEREIRA (2004) observam pela análise dos resultados quantitativos obtidos em sua pesquisa que a preocupação com o ensino de programação tem se demonstrado crescente para o WEI e estável para o SBIE e concluem que a comunidade de computação brasileira se mostra ciente e busca soluções referentes aos problemas do ensino-aprendizagem de programação de computadores.

No âmbito internacional duas das mais conceituadas organizações ,o *Institute of Electrical and Electronic Engineers* (IEEE)[4] e a *Association for Computing Machinery* (ACM)[5] concordam que estudo de algoritmos deve fornecer o discernimento da natureza intrínseca do problema e técnicas de solução independentes de qualquer paradigma[6], linguagem de programação ou hardware. IEEE & ACM (2001). E demonstram a relevância da discussão através de publicações como, por exemplo, HENDERSON (1987), BEER & CHIEL (1999), KLASSNER (2002), FAGIN (2003) e FLOWERS & GOSSET (2006).

---

[3] Veja o item "2.5 O Algoritmo"

[4] http://www.ieee.org.br

[5] http://www.acm.org/

[6] Veja o item "3.1 Os paradigmas de programação"

4 CONCEPÇÕES DE APRENDIZAGEM

4.1 As Teorias do Conhecimento

A aprendizagem humana difere do adestramento dos demais animais. O ato ou vontade de aprender é uma característica essencial do ser humano, pois somente este possui o caráter intencional, ou a intenção de aprender. O que é o conhecimento? De onde provêm os conteúdos da consciência? Pode o sujeito realmente conhecer o objeto? Quais são as formas possíveis de conhecimento?

Estas questões, segundo JAPIASSU (1975), formam o corpo temático que constitui a disciplina filosófica chamada de Teoria do Conhecimento ou Epistemologia do grego *epistem* (ciência) que trata dos problemas filosóficos relacionados à crença e ao conhecimento. Pode-se dizer que a epistemologia se origina em Platão, ele opõe a crença ou opinião ("*doxa*", em grego) ao conhecimento.

As respostas encontradas para as questões anteriores deram origem às concepções de aprendizagem que norteiam as práticas pedagógicas. Em toda prática docente está presente um conjunto de pressupostos filosóficos que determina os objetivos e os procedimentos educacionais, mesmo quando o professor não possui consciência explícita deles. A efetividade da ação docente, entretanto, está vinculada à capacidade do professor de compatibilizar sua visão de homem e de conhecimento aos objetivos de ensino e aos procedimentos pedagógicos adotados. Um mapeamento entre as concepções filosóficas sobre o conhecimento, as teorias psicológicas da aprendizagem e as práticas educacionais pode ser estabelecido. Em termos epistemológicos, o conhecimento é uma relação que se estabelece entre o homem, denominado de sujeito, e o mundo, real ou abstrato, denominado de objeto. A natureza da relação sujeito-objeto é explicada segundo paradigmas filosóficos distintos, como por exemplo, o Racionalismo, o Empirismo e o Interacionismo. OLIVEITA, COSTA E MOREIRA (2001).

As origens do racionalismo remontam à antiguidade. Platão, por exemplo, concebia a existência de um mundo perfeito, imaterial, chamado por ele de mundo das idéias, do qual o mundo real seria uma “cópia piorada”. Todo conhecimento humano teria sua preexistência no mundo das idéias, antes de existir no mundo real. Todo aprendizado seria uma espécie de recordação de uma idéia inata na mente. Na Idade Moderna, René Descartes distinguiu o corpo da alma, atribuindo a esta a função de ser sede da razão. Da mesma forma que Platão, Descartes admitia que ao nascer o homem possuía um conjunto de idéias inatas, capaz de orientar e organizar as observações trazidas pelos sentidos. JAPIASSU (1975).

O racionalismo considera que a razão é a fonte do conhecimento, ou seja, que o conhecimento provém do sujeito. Ao nascer, segundo os racionalistas, homem já possui a "caixa de ferramentas" e os "esquemas" do conhecimento. O papel da experiência sensorial é apenas o de despertar e ativar a consciência do conhecimento preexistente. BIGGE (1977).

Para os empiristas, todo conhecimento decorre da experiência sensorial. O homem, ao nascer, não teria nenhum conteúdo na consciência. Seria, segundo Locke, uma "tabúla rasa". Somente através dos sentidos que as experiências do mundo iriam sendo acumuladas na consciência e associadas umas às outras para dar origem ao conhecimento. Para John Locke, principal defensor desta concepção filosófica, "não existe nada na mente que não tenha passado antes pelos sentidos". Para ele, as idéias eram as unidades da mente, e as associações consistiam em combinações de idéias. As idéias complexas eram obtidas a partir de combinações de idéias simples através de operações mentais, tal como as combinações químicas produziam substâncias compostas a partir dos elementos químicos básicos. Segundo o paradigma empirista, a fonte do conhecimento está no objeto e o sujeito só pode adquiri-lo através dos sentidos. OLIVEIRA, COSTA e MOREIRA (2001).

O interacionismo surgiu como uma reação ao racionalismo e ao empirismo. A idéia central desta é que as qualidades de uma coisa derivam da relação com outras coisas. No ato de conhecer, a fonte do conhecimento não se encontra nem no sujeito nem no objeto, mas na interação de ambos. Ele não atribui ao sujeito nenhuma idéia inata, nem o objeto é considerado como possuidor de características imutáveis. Para os interacionistas, o conhecimento não é uma descrição literal do que existe fora do homem, é uma interpretação do mundo que o sujeito constrói a partir da percepção que desenvolveu do objeto. Uma mudança no campo perceptivo irá provocar uma mudança na interpretação do objeto. Sujeito e objeto mudam na medida em que muda a forma de interação entre eles. COUTINHO e MOREIRA (1992).

O fenômeno educacional é por sua própria natureza uma realidade complexa que envolve múltiplos aspectos, o técnico, o humano, o histórico, o cognitivo, o emocional, o sócio-político e o cultural. Cada teoria irá privilegiar um ou outro aspecto deste fenômeno multidimensional. Não se pode afirmar que exista uma teoria "certa", mas pode-se falar de uma teoria "adequada" para se atingir determinados objetivos educacionais, concebidos como necessários ao aprendizado.

4.2 As Teorias da Aprendizagem

Denomina-se de abordagem tradicional o conjunto de teorias da aprendizagem desenvolvidas ao longo da história, desde a Antiguidade, cujas concepções foram elaboradas pela introspecção e a especulação filosófica realizada pelos seus idealizadores, que se basearam em seus próprios processos mentais para explicar a aprendizagem. A abordagem tradicional considera que o homem possui uma mente ativa, imaterial, dotada de faculdades inatas que precisam ser exercitadas para ser fortalecidas, da mesma forma que os exercícios físicos desenvolvem a musculatura do corpo. A abordagem tradicional fundamenta-se, portanto, nos pressupostos do paradigma racionalista ao considerar que as faculdades mentais preexistam no ser humano como estruturas cognitivas acabadas. BIGGE(1977).

A aprendizagem, nesta visão, é reduzida a um treinamento mental em que a forma do estudo sobrepõe-se ao conteúdo estudado. A relação professor-aluno é vertical e individual, inexistindo relação aluno-aluno, nem formação de grupos. Há uma desvinculação entre o conteúdo aprendido e a sua aplicabilidade. Ao admitir a pré-formação das estruturas cognitivas, a visão racionalista do processo de aprendizagem acaba por absolver a escola de seu compromisso de garantir, dentro de certos limites, o desenvolvimento e a aprendizagem dos alunos. COUTINHO e MOREIRA (1992).

A abordagem comportamentalista fundamenta-se nos pressupostos filosóficos do empirismo e considera a experimentação planejada como a base do conhecimento. A aprendizagem seria o resultado da ordenação de experiências e eventos do universo, colocando-os em códigos simbólicos compreensíveis ao sujeito. Para os comportamentalistas, conhecer é a tentativa de descobrir a ordem natural das coisas. Como certos fatos se relacionam sucessivamente uns com os outros, o aprendizado consistiria em registrar os fatos observados experimentalmente e associá-los, estabelecendo cadeias de causa e efeito. Nesta abordagem, o sujeito é visto como um sujeito passivo, cuja consciência è moldada através da experiência com o mundo objetivo, considerado externo e acabado. A instrução programada criada por Skinner sintetiza estas estratégias. COUTINHO e MOREIRA (1992).

> Como exemplo da aplicação da concepção behaviorista (comportamentalista) de aprendizagem, Oliveira, Costa e Moreira (2001, p.21) relatam que a produção de softwares educativos (SE) que vem sendo freqüentemente influenciada por ela. Os títulos de SE disponíveis tanto no mercado brasileiro como no mercado internacional em geral caracterizam-se pela tentativa de se impedir que o aluno cometa erros, o que é feito normalmente por um *feedback* imediato, que ocorre em um tom muito desagradável para o aluno e que funciona regularmente como uma punição. Um exemplo disso seria a utilização do recurso de uma animação mostrando um enforcamento público em conseqüência dos erros cometidos pelo aluno no trabalho com o SE. Ainda que se proponha um caráter lúdico nesse tipo de programa esse se restringiria a incitar uma competição entre alunos o que em vez de propiciar uma experiência prazerosa, assumiria o papel de despertar meramente a emulação entre os usuários, sem a preocupação de lhes garantir um interesse genuíno pelo conteúdo que está sendo discutido. Uma opção para lidar com o erro do aluno seria a inclusão de estratégias interativas de ensino.

O ponto de partida da abordagem interacionista é a consideração de que o conhecimento não se encontra de forma absoluta nem no sujeito e nem no objeto. Ele é construído continuamente através da interação que se estabelece entre os dois elementos, sujeito e objeto, no ato de conhecer. O conhecimento humano não é só dependente de fatores genéticos ou ambientais, mas devem ser também levados em conta os fatores culturais que contextualizam a relação sujeito-objeto. Pode-se notar assim, os pressupostos interacionistas sobre os quais se fundamentam esta abordagem. Na concepção interacionista, ensinar é bem mais do que o simples ato de "passar" ou "transmitir o conteúdo". COUTINHO e MOREIRA (1992).

Diversos teóricos, como por exemplo, Piaget, Vygotsky e Wallon, entre outros, têm sinalizado que o centro do processo de aprendizagem não está naquele que ensina, mas sim, naquele que aprende. Sendo assim, "o ato de ensinar", não garante, necessariamente, para a pessoa que o recebe, "o fato de aprender". Neste contexto delineia-se a figura do professor mediador, que utiliza os ambientes informatizados de aprendizagem (AIA) como um substrato no qual apóia suas práticas pedagógicas, que podem usar as simulações e o trabalho em grupos como ponto de apoio para aqueles que apresentem maiores dificuldades, dentro da perspectiva na qual Vygotsky desenvolve o conceito de zona de desenvolvimento proximal (ZDP), que é a distância entre o nível de desenvolvimento real, que se costuma determinar através de testes psicológicos e que se refere à capacidade de solução independente de problemas, e o nível de desenvolvimento potencial, determinado através da solução de problemas sob orientação do professor, ou em colaboração com companheiros mais capazes. COSTA (2002, p. 55).

A constatação de que na atualidade o conhecimento é gerado em um contexto orientado a mudanças tecnológicas e culturais e que o mesmo se torna obsoleto rapidamente, mostra que este não é estático e nem absoluto. Tanto o objeto (o mundo) como o sujeito (o ser humano), transformam-se continuamente. O conhecimento é, portanto, o resultado de uma interação entre o sujeito e o objeto.

As abordagens tradicional e comportamentalista são inadequadas para tratar deste tipo de aprendizagem porque ambas são construídas com base no pressuposto de que o conhecimento é estático e absoluto. A abordagem interacionista, ao contrário, assume que o conhecimento é continuamente construído pelo sujeito pela interação com o seu contexto.

4.2.1 O Construtivismo de Piaget

No bojo do interacionismo, o construtivismo define a aprendizagem como um processo de troca mútua entre o meio e o indivíduo, tendo o outro como mediador. O aluno é um elemento ativo que age e constrói sua aprendizagem. Cabe ao professor instigar o sujeito, desafiando, mobilizando, questionando e utilizando os "erros" de forma construtiva, garantindo assim uma reelaboração das hipóteses levantadas, favorecendo a construção do conhecimento. Nesta concepção o aluno não é apenas alguém que aprende, mas sim o que vivencia os dois processos sendo ao mesmo tempo "ensinante" e "aprendente". COUTINHO e MOREIRA (1992).

A partir desta idéia nuclear, entretanto, a aceitação crescente dos princípios construtivistas, em psicologia e em educação originou, nos últimos anos, um enriquecimento e uma diversificação também crescentes de tais princípios, de maneira que, sob o rotulo genérico de "construtivismo", encontram-se atualmente enfoques teóricos e propostas de atuação muito diversos, quando não abertamente contraditórios. COLL *et al* (2004, p.107). Neste cenário fortemente influenciado pelo construtivismo, alguns pesquisadores têm ampliado a discussão sobre a concepção interacionista de aprendizagem, como por exemplo, Vergnaud e sua teoria dos campos conceituais.

De acordo com MOREIRA (2002), Vergnaud redireciona em sua teoria o foco piagetiano das operações lógicas gerais, das estruturas gerais do pensamento, para o estudo do funcionamento cognitivo do "sujeito-em-situação". Ele toma como referência o próprio conteúdo do conhecimento e a análise conceitual do domínio desse conhecimento, destacando, na teoria de Piaget, as idéias de adaptação, desequilibração e reequilibração como pedras angulares para a investigação em didática das ciências. Ele acredita que a grande pedra angular colocada por Piaget foi o conceito de esquema. A teoria dos campos conceituais foi desenvolvida também a partir do legado de Vygotsky. Isso se percebe, por exemplo, na importância atribuída à interação social, à linguagem e à simbolização no progressivo domínio de um campo conceitual pelos alunos.

As situações é que dão sentido ao conceito, elas é que são responsáveis pelo sentido atribuído ao conceito. Um conceito torna-se significativo em uma variedade de situações, mas o sentido não está nas situações em si mesmas, assim como não está nas palavras nem nos símbolos. O sentido é uma relação do sujeito com as situações e com os significantes. Mais precisamente, são os esquemas, ou seja, os comportamentos e sua organização, evocados no sujeito por uma situação ou por um significante (representação simbólica) que constituem o sentido dessa situação ou desse significante para esse indivíduo. Campo conceitual é, para ele, um conjunto informal e heterogêneo de problemas, situações, conceitos, relações, estruturas, conteúdos e operações de pensamento, conectados uns aos outros e, provavelmente, entrelaçados durante o processo de aquisição. VERGNAUD *apud* MOREIRA (2002).

4.2.2 O Construcionismo de Papert

Nos trabalhos de Papert, o construcionismo é apresentado como uma teoria da aprendizagem, em que o computador é incluído no contexto do mundo como fator de transformação cultural profunda. Em segundo lugar, a elaboração teórica tem sido produzida a partir de uma aplicação prática orientada efetivamente à educação, portanto, caminha no sentido inverso do desenvolvimento das teorias anteriores, no qual as aplicações em sala de aula surgiram posteriormente ao desenvolvimento teórico. Em terceiro lugar, é recente e se encontra em fase de estruturação. Suas origens remontam à década de 1960, com o trabalho desenvolvido por cientistas do *Massachusetts Institute of Tecnology* (MIT), liderados por Seymour Papert, matemático e pesquisador da área de Inteligência Artificial. Este trabalho consistiu no desenvolvimento de uma linguagem de programação chamada de LOGO. O objetivo do LOGO foi tornar o uso do computador acessível às crianças. A idéia defendida por Papert como premissa do seu trabalho é que o computador pode contribuir para o desenvolvimento dos processos mentais, não somente como instrumento, mas, mais essencialmente, de maneira conceitual, influenciando o pensamento. Isto porque são portadores de inúmeras idéias e de sementes de mudança cultural, que podem ajudar na formação de novas relações com o conhecimento de maneira a atravessar as tradicionais barreiras que separam a ciência dos seres humanos e os conhecimentos que cada indivíduo tem de si mesmo" PAPERT (1986).

Papert define o construcionismo como a sua reconstrução pessoal do construtivismo Piagetiano. Ele trabalhou pessoalmente com Piaget durante 4 (quatro) anos. Em essência pode-se afirmar que o construcionismo aceita as teses centrais do construtivismo piagetiano.

As diferenças estão colocadas por Papert da seguinte forma: o construcionismo enfatiza o papel do meio cultural no desenvolvimento enquanto o construtivismo não o considera relevante. Amplia o conceito de assimilação, no sentido de incluir o aspecto afetivo. Rejeita o sequenciamento dos estágios de desenvolvimento proposto por Piaget, em especial a supervalorização do pensamento formal, visto por Papert como impedimento direto à aprendizagem. PAPERT (1994). Em síntese, o construtivismo se preocupa mais em explicar como o conhecimento é construído, enquanto o construcionismo se preocupa também em criar ambientes de aprendizagem que possam produzir mudanças no intelecto. É exatamente sobre as diferenças citadas que o construcionismo elabora seus principais conceitos teóricos.

Na visão construcionista o homem é um construtor e como tal ele necessita "materiais" para sua obra. A aprendizagem será mais eficaz se o ambiente onde ela ocorre puder dispor de "materiais" que facilitem a experimentação e a construção de conceitos capazes de auxiliar na estruturação de novos conceitos. Papert chama esses ambientes de micromundos e os materiais, de objetos transitórios ou de objetos-para-pensar-com. A função destes objetos é fornecer um meio concreto para que um conhecimento imediato possa ser construído e ao mesmo tempo estabelecer uma base para nova aprendizagem. Assim são considerados os aspectos culturais. O aspecto afetivo da assimilação é incluído pelo conceito de aprendizagem sintônica. O termo sintonicidade, explica Papert, foi usado por Freud para "descrever instintos ou idéias que sejam aceitáveis ao ego, isto é, compatíveis com a integridade do ego e com as suas necessidades". A aprendizagem sintônica é aquela que ocorre quando o indivíduo se identifica com o objeto de estudo e se envolve afetivamente com a aprendizagem, porque sente prazer, orgulho em aprender e se torna responsável e ativo por ela. Assim, o conhecimento assimilado está relacionado não apenas ao fator cognitivo, mas também e principalmente ao aspecto afetivo. PAPERT (1986).

Segundo Papert, uma estratégia de aprendizagem eficaz consiste no desenvolvimento de projetos em grupos. Os projetos devem ser suficientemente abertos para permitir abordagens muito diferentes e ao mesmo tempo restritos o suficiente, para permitir que diferentes abordagens sejam comparadas. A idéia defendida por PAPERT (1994), aqui é a de que não são as regras de resolução que resolvem o problema, mas pensar sobre o problema que promove a aprendizagem. Além disto, a discussão de um problema com outra pessoa também contribui para promover a aprendizagem. Não existe um "método de ensino", porque isto pressupõe transmissão de conhecimentos e, "quando o conhecimento é distribuído em minúsculos pedaços, não se pode fazer nada, exceto memorizá-lo na sala de aula e escrevê-lo no teste". PAPERT(1994).

Papert critica assim a concepção tradicional da escola, que considera a inteligência como inerente ao ser humano, desnecessária e até impossível de ser desenvolvida. Ao contrário disto, ele afirma que só quando o conhecimento está integrado num contexto de uso pode-se ativá-lo e, ao corrigir sucessivamente as falhas de compreensão, realmente adquiri-lo. O professor, dentro da teoria construcionista, tem um papel não apenas técnico de promover a aprendizagem, planejando e coordenando as atividades desenvolvidas na forma de projeto, pelos alunos, mas também de ser um construtor do seu próprio conhecimento pedagógico. Isto só pode ocorrer, segundo Papert, se o professor também estiver envolvido sintonicamente com a atividade de aprendizagem em questão. Assim como Piaget afirmou que brincar é o trabalho das crianças, Papert afirma que é preciso desenvolver a idéia de que o trabalho deve ser o brinquedo dos adultos.

Esta perspectiva de desenvolvimento da aprendizagem irá, mais evidentemente, influenciar as discussões desenvolvidas neste trabalho. Contudo, não serão descartadas as contribuições das outras teorias de aprendizagem.

5 ROBÓTICA

5.1 Os Robôs

A construção de uma máquina com capacidade de efetuar o trabalho pesado ou perigoso dotada de força e inteligência além de total submissão é o arquétipo do escravo ideal. Esta fantasia torna-se próxima da concretização pelos robôs. O termo robô se origina da palavra Tcheca "*robota*", que significa "trabalho compulsório" e foi usado pela primeira vez em 1921 na peça teatral *R.U.R (Rossum's Universal Robots)* pelo novelista e escritor Tcheco Karel Capek. SALANT (1990, p. 1).

Segundo MARTINS (1993), a palavra robô vem sendo usada para se referir a máquinas que executam trabalhos repetitivos ou trabalhos que humanos acham difíceis ou indesejáveis. Pode-se exemplificar o uso da robótica em diversas áreas de conhecimento. Na engenharia, há os robôs que mergulham a grandes profundidades para auxiliar em reparos nas plataformas de petróleo, enquanto na medicina os robôs já auxiliam as cirurgias de alto risco.

Robôs são dispositivos mecânicos versáteis equipados com sensores e atuam sob o controle de um sistema computacional. Eles realizam tarefas executando movimentos em um espaço físico. Este espaço é povoado por vários objetos e está sujeito às leis da natureza. SALANT (1990).

A robótica é uma ciência da engenharia aplicada que é tida como uma combinação das tecnologias de máquinas operatrizes e ciência da computação GROOVER et al. (1988, p.23).

5.1.1 Histórico

Na época em que foram lançados, os robôs eram caros e acessíveis a pouquíssimas empresas existentes em países mais desenvolvidos, principalmente no Japão e nos Estados Unidos. No entanto, a partir de 1976 começaram a baixar de preço de maneira extremamente acelerada. O grande responsável por esta brutal redução de custos que ocorreu na informática e na robótica é a microeletrônica. Com o avanço desta disciplina, por exemplo, foi possível colocar toda a capacidade do ENIAC[7], o primeiro computador a válvula desenvolvido em 1950, em uma pastilha de silício e menos de 0,5 cm^2. Ressaltando que, com isso, obtém-se maior velocidade de processamento a um custo extremamente inferior.

[7] Veja o capitulo 2 – O Computador e sua arquitetura

Desta maneira os microprocessadores influenciaram diretamente a capacidade de todas as máquinas industriais, tendo impacto decisivo nas tecnologias associadas à robótica, permitindo que a capacidade de processamento de informações se multiplicasse de modo estrondoso, além de baratear o custo dos robôs, tornando-os mais acessíveis.

Os robôs, segundo SALANT (1990), são normalmente divididos em gerações:

1ª Geração:

São dispositivos que atuam como escravo mecânico do homem, pois realizam movimentos pré-programados.

2ª Geração:

São dispositivos programáveis que trabalham em função de seu ambiente, devem possuir a capacidade de reconhecer o objeto desejado e adaptar a melhor trajetória das suas partes móveis para obtê-la. A comunicação com o meio ambiente é feita através de sistemas de sensoriamento e identificação.

3ª Geração:

Robôs que empregam métodos informáticos avançados conhecidos como inteligência artificial e procedimentos de recepção multisensorial.

A primeira pessoa a usar o termo robótica foi o escritor Issac Asimov, que em 1942 em uma pequena história chamada "*Runaround*". Ele também estabeleceu as leis que deveriam governar o comportamento dos robôs, como o exemplo fictício da figura 16.

	1. *Um robô não pode ferir um ser humano ou, permanecendo passivo, deixar um ser humano exposto ao perigo.*
	2. *O robô deve obedecer as ordens dadas pelos seres humanos, exceto se tais ordens estiverem em contradição com a primeira lei.*
	3. *O robô deve proteger sua existência na medida em que esta proteção não estiver em contradição com a primeira ou a segunda lei.*
	4. *O robô não pode causar mal a humanidade nem permitir que ela própria o faça.*

Na Ficção Científica (*SCI-FI*), uma forma de ficção desenvolvida no século XX, que lida principalmente com o impacto da ciência, tanto verdadeira como imaginada, sobre a sociedade ou

os indivíduos, existem diversos robôs famosos como, por exemplo, o *R2-D2* da série guerra nas estrelas e o andróide da série perdidos no espaço.

5.1.2 Classificações

Segundo VIEIRA (1985), os robôs podem ser classificados quanto às suas aplicações e formas de trabalhar:

Manipuladores:

> Sistemas mecânicos multifuncionais, cujo sistema de controle permite ao homem governar o movimento de seus membros.

Robôs de Seqüência:

> Manipulador que funciona segundo uma seqüência e condições pré-estabelecidas de movimentos. Quando é possível alterar algumas características do ciclo de trabalho, os robôs se intitulam de seqüência variável.

Robôs de Aprendizagem:

> Limitam-se a repetir uma seqüência de movimentos, depois de sofrer a intervenção de um operador ou memorizador.

Robôs Inteligentes:

> Capazes de determinar seus próprios atos e de tomar decisões em tempo real utilizando-se de sensores.

De acordo com GROOVER *et al* (1988), outra classificação aceita refere-se ao sistema de acionamento:

Hidráulico:

> Utilizado em robôs de maior porte, este sistema propicia ao robô maior velocidade e força; em contrapartida, ele se soma ao espaço útil no piso requerido pelo robô.

Elétrico:

> Os robôs ocupam menor espaço e suas aplicações tendem para trabalho de precisão e repetição.

Pneumático:

> Robôs menores que se limitam a simples operações “de pega e põe”, com ciclos rápidos.

5.1.3 Cenário Mundial

O estudo "*2001 World Robotics Survey*", produzido pelas Nações Unidas, relata um estrondoso crescimento no mercado mundial de robôs, conforme pode ser observado na tabela 1.

Tabela 1 – Mercado Mundial de Robôs

	2001	**2004 (Previsão)**
Japão	389.000	447.000
Europa	198.000	306.000
América do Norte	90.000	116.000
TOTAL	750.000	975.000

Fonte: INOVACAOTECNOLOGICA.

O setor número um na utilização de robôs é o automobilístico. Tomando-se o caso da Itália e Alemanha, enquanto que na indústria em geral, para cada 10.000 trabalhadores, há cerca de 120 robôs na Alemanha e 95 na Itália, na indústria automobilística de ambos os paises, há um robô para cada 10 trabalhadores. Para a indústria como um todo, estes números são de 300 no Japão, 80 na Suécia, 60 na França e 50 na Espanha. O mercado de robôs no Brasil está defasado se comparado ao potencial de sua economia, pois, enquanto a Espanha possui em torno de 4.000 robôs industriais, o Brasil possui somente 550 unidades. A robotização iniciou-se em 1982 pela Volkswagen, sendo que atualmente é a GM que possui o maior número de robôs com cerca de 200 unidades na sua linha de montagem. Na versão mais atual do *World Robotics Survey*, publicada pela comissão econômica das nações unidas (ONU) para a Europa (UNECE) em 2004, a taxa de crescimento da utilização da robótica é mantida em forte ritmo.

O maior fornecedor para o mercado nacional é a ABB – *Asea Brow Bovery* com cerca de 70% do mercado, seguida da japonesa Fanuc. Os custos do robô nacional são compatíveis com os preços praticados a nível internacional, sendo que entre os fabricantes nacionais destacam-se a Vilares-Hitachi, a Metrixer e a Romi. Além da indústria automobilística, as áreas que mais usam os robôs são a indústria de alimentos e a petrolífera. Os centros de tecnologia que produzem as mais significativas pesquisas na área de robótica no Brasil são a UNICAMP, ITA/USP e a UFMG. VIEIRA (1995).

5.1.4 A Robótica na indústria

O robô industrial segundo a *Robotics Industries Association* (RIA)[8] é definido como "um manipulador multifuncional programável projetado para mover materiais, partes, ferramentas e outros dispositivos especiais usando movimentos variáveis programados para realização de uma variedade de trabalho" CAPELLI (2002). Os robôs usados hoje nas fábricas possuem sua base fixada ao piso, o corpo está fixado a base e o braço ao corpo. Na extremidade do braço está o punho que consiste em inúmeros componentes que lhe permitem orientação numa diversidade de posições. Os movimentos relativos entre os diversos componentes do corpo, braço e punho são proporcionados por uma série de vínculos e articulações que o mesmo possui envolvendo movimentos rotativos e deslizantes GROOVER et al. (1988, p.24).

A grande maioria dos robôs usados atualmente na indústria é de 2ª geração[9], sendo manipuladores de peças que funcionam em conjunto com outras máquinas, como robôs soldadores de montadoras de automóveis, como o exemplo da figura 17. As principais aplicações da robótica na indústria estão direcionadas para o manuseio de materiais, quando o robô transporta materiais ou peças entre locais ou em células de trabalho quando o robô contribui diretamente com o processo de manufatura, como por exemplo, soldagem ou pintura como também na inspeção, quando o robô contribui para o controle de qualidade do bem manufaturado.

5.1.5 O debate sobre os efeitos da Robótica na sociedade

A automação possibilita grandes incrementos na produtividade do trabalho, além de aumentar a produção, os equipamentos automatizados possibilitam uma melhora na qualidade do produto, uniformizando a produção, eliminando perdas e refugos. Ela também permite a eliminação de tempos mortos, ou seja, permite a existência de "operários" que trabalhem 24 horas por dia sem reclamar, o que leva a um grande crescimento na rentabilidade dos investimentos.

A automação permite ainda a flexibilização da produção nos moldes do *toyotismo* ou dos modelos italiano e sueco de produção, que prevêem a produção conforme as demandas do mercado, evitando que se produzam estoques de produtos de pouca aceitação pelos consumidores. Entretanto, como qualquer aspecto da civilização humana, existem também aspectos negativos. Para uma visão mais

---

8 http://www.roboticsonline.com/index.cfm

9 Para maiores esclarecimentos consulte o item 5.1.1Histórico.

abrangente deste debate, sugere-se a leitura de pensadores relacionados à Sociologia do Trabalho, como Braverman, Naville, Friedmann, A. Touraine, P. Zarifian, Tomasi, etc.

5.2 A Robótica pedagógica

O conceito que aparece na literatura como *educational robotics*, robótica pedagógica (RP) ou robótica educativa consiste basicamente na aprendizagem por meio da montagem de sistemas constituídos por modelos (robôs). Esses robôs são mecanismos que apresentam alguma atividade física, como movimento de um braço mecânico, levantamento de objetos, etc. Ao trabalhar ambiente de RP o objeto de programação passa a ser um dispositivo robótico construído pelos alunos. Este dispositivo passa a ser na verdade, um artefato cognitivo que os alunos utilizam para explorar e expressar suas próprias idéias, ou "um objeto-para-pensar-com", nas palavras de PAPERT(1986).

Com a Robótica disponibilizada dentro ou fora da sala de aula, encontra-se hoje o surgimento de uma grande e abrangente área de utilização da tecnologia a favor dos processos cognitivos. Isto se justifica ainda mais quando se constata a atrativa utilização não só do computador, mas dos mecanismos que o mesmo pode gerenciar, controlar, mover, dominar e conduzir.
OLIVEIRA (2004)

A partir dos anos 50 tiveram início os estudos sobre a possibilidade de se utilizar robôs como instrumento de apoio ao ensino, mas durante muito tempo o fator custo foi um severo limitador da sua aplicação prática no "chão da escola". Historicamente, as primeiras aplicações da RP foram baseadas no reaproveitamento de sucata como, por exemplo, no projeto da UFRGS[10] ou o Cyberbox[11].

Foi no final da década de 70, mais precisamente no ano de 1976, no MIT, que o matemático *Seymour Papert*[12] desenvolveu a linguagem de programação LOGO, voltada para a Educação. Seu trabalho é fundamentado na teoria de Piaget, que propõe um aprendizado baseado nas diferenças individuais, na reflexão sobre o próprio processo de aprendizagem e na lógica do pensamento. Inicialmente utilizou computadores de grande porte, fato que fez com que, até o surgimento dos

---

10 http://educadi.psico.ufrgs.br/centros/relators/messa/relat98.html

11 http://www.cyberbox.com.br

12 Veja o item "4.2.2 Construcionismo", deste trabalho

microcomputadores, o uso do LOGO ficasse restrito às universidades e laboratórios de pesquisa. PAPERT (1986).

Com a disseminação dos microcomputadores, o LOGO passou a ser adotado em muitas escolas. No período de 1983 até 1987 aconteceu uma verdadeira explosão no número de experiências, na produção de material de apoio, livros, publicações e conferências sobre o uso do LOGO. Esta abordagem durante muitos anos privilegiou a utilização da linguagem de programação LOGO e atualmente, por questões tecnológicas, tem se direcionado para a utilização de outras linguagens de programação como ROBOLAB, NQC (C) ou LEJOS (JAVA) e os Kits de construção de robôs têm sido os mecanismos mais utilizados para a aplicação da RP. KNUDSEN (1999)

O advento dos *kits* a baixo custo propiciou a condução de uma série de experimentos nesta temática, conforme descritos por FAGIN e MERKLE (2002), BEER e CHIEL (1999), WOLZ (2001), KLASSNER (2002) e KUMAR e MEEDEN (1998), entre outros. Diversos órgãos de pesquisa como, por exemplo, o *MIT Media Lab*[13] e o NIED (UNICAMP)[14], estão envolvidos na investigação do processo de construção do conhecimento no contexto educacional e o desenvolvimento de equipamentos e de software a serem utilizados com finalidades educacionais. Inúmeros projetos têm sido conduzidos para avaliar como a utilização desses equipamentos e dos softwares por alunos e professores possibilita a criação de uma metodologia de ensino-aprendizagem.

Essa metodologia propõe situações de aprendizagem em que o aluno constrói o seu conhecimento por intermédio de uso do computador. Pode-se ainda citar iniciativas atuais da aplicação da robótica com finalidades educacionais como por exemplo a Microsoft Robotics Studio[15] para a simulação de robôs via Internet, e o *RobotC*, uma linguagem de programação desenvolvida pelo *The National Robotics Engineering Center* (NREC)[16] da universidade Carnegie Mellon, em seu instituto de robótica, uma as maiores instituições de pesquisa sobre robótica em âmbito mundial.

A RP tem propiciado a alunos e professores uma maneira diferenciada de se trabalhar o aprendizado de conceitos a partir da montagem, automação e controle de dispositivos robóticos, via computador. Tipicamente, o processo de disseminação da robótica pedagógica inclui na sua metodologia a realização de oficinas de trabalho envolvendo professores e alunos. D'ABREU (2004)

---

13 http://www.media.mit.edu/

14 http://www.nied.unicamp.br/

15 http://msdn.microsoft.com/robotics/

16 http://www.rec.ri.cmu.edu/about/overview/index.htm

Segundo D'ABREU (2004), as etapas típicas destas oficinas são:

- A demonstração do funcionamento dos componentes eletrônicos, motores, sensores e lâmpadas;
- A formação de grupos de trabalhos;
- A montagem de dispositivos robóticos pelos grupos;
- O desenvolvimento dos programas de computador responsáveis pelo controle do robô;
- A discussão dos aspectos científicos e tecnológicos inerentes ao dispositivo robótico, em construção, com base nos conceitos curriculares que se pretende trabalhar;
- Os teste e a conclusão dos projetos;
- A apresentação dos projetos para os colegas participantes da oficina e demais convidados.

Durante o processo, os comportamentos inesperados não são tratados como erros dentro de uma visão reducionista binária (certo ou errado), mas como possibilidades a serem exploradas, buscando explicar o fenômeno através da reflexão individual e em grupos e posteriormente propondo alternativas por meio do aprimoramento do algoritmo. Um projeto de RP, além da construção do robô, exige a criação de um conjunto de programas que lhe permita a realização de tarefas especificas. A movimentação, o controle, a definição de trajetórias e a utilização dos sensores para interagir como meio são algumas das capacidades que o programa deve implementar no robô. Nessa atividade pode-se pensar na montagem do robô, na programação e nos conceitos que poderão ser trabalhados para que a atividade não tenha apenas um fim em si mesma, desprovida de conexão com as demais áreas de conhecimento.

A montagem do robô implica em primeiro lugar, definir a sua parte mecânica associando peças comuns com peças mecânicas (engrenagens, polias, cremalheiras, etc.) e elétricas (motores, lâmpadas e sensores). Assim, forma-se uma estrutura que dá forma ao robô, a parte mecânica permite o seu movimento. Com relação à programação esta é uma etapa importante do processo de desenvolvimento da atividade, pois é o programa que controlará o dispositivo dando-lhe "vida", movimento. No momento da programação define-se as funcionalidades do robô, respeitando-se os princípios físicos e mecânicos (força, atrito, peso, redução, transmissão de velocidade, etc.).

Esta abordagem permite ao aluno interagir com o concreto (robô) e o abstrato (programa) em um mesmo projeto, proporcionando a oportunidade de observar o comportamento de seu raciocínio executado em um artefato físico. Esta idéia aparece na literatura contemporânea através do conceito das chamadas “interfaces tangíveis”, que propõem aproximar o mundo físico do mundo digital ou virtual, de modo que este último possa ser manipulado por meio de interfaces físicas, diminuindo assim a distância que os separa. METOYER; XU & SRINIVASAN (2003). Apesar da área de interfaces tangíveis ser relativamente nova, ela já apresenta grandes trabalhos desenvolvidos. Como

exemplos, temos o *Topobo*, um jogo de montagem em 3D, motorizado, que permite ao usuário criar diferentes estruturas com formas de animais, nas quais pode gravar movimentos e executá-los repetidas vezes; e o i/o *Brush*, um pincel que permite que o usuário capture a cor, textura e forma de qualquer superfície em que o mesmo seja encostado e que, posteriormente desenhe com a textura que foi capturada. Ambos os trabalhos foram desenvolvidos pelo *Tangible Media Group* (TMG), no *MIT Media Lab*, disponíveis no endereço http://tangible.media.mit.edu.

Um outro aspecto importante a se considerar no uso do ambiente robótica pedagógica é a questão da interdisciplinaridade inerente a atividade de montar e controlar dispositivos robóticos via computador. Algumas condições da interdisciplinaridade evidenciadas na literatura servem de aporte ou fundamentos teóricos para esta afirmação. Estas condições são: mudança de postura, diálogo, cooperação, metodologia, dúvida e indagação, além de significação. Mudança de atitude do professor, que por sua vez refletirá na mudança de postura do aluno frente ao conhecimento. Tanto o professor quanto o aluno terão que mudar para compreender que o conhecimento não existe, a priori, pronto e acabado, faz parte do compromisso de ambos, participar na elaboração do mesmo. Diálogo e cooperação tanto entre os pares, quanto entre as disciplinas para que haja troca de idéias, dúvida e indagação que conduz à reflexão e à busca de uma teoria que fundamente a prática. A significação do dispositivo robótico deve contextualizar e dar sentido aos conceitos a serem apresentados. D'ABREU (2002)

Um ambiente típico de robótica pedagógica pressupõe a existência de professor, aluno e ferramentas que propiciam a montagem, automação e controle de dispositivos mecânicos. Nas palavras de D'ABREU (2004), os ambientes de aprendizagem baseado no uso de dispositivos robóticos tem possibilitado, de forma, simples, econômica, rápida e segura, disponibilizar recursos tecnológicos para a aprendizagem, não só de robótica, mas de ciências de uma maneira geral. Isso tem propiciado criar situações de aprendizagens onde os alunos podem, avaliar resultados, experimentar idéias e testar hipóteses, de uma forma rápida barata e segura.

A disseminação de uso desse ambiente em diferentes contextos de aprendizagem tem se constituído numa forma interessante de se trabalhar com tecnologias digitais onde tanto professores quanto os alunos podem de forma mais concreta manusear os conceitos científicos. Alunos e professores interagindo entre si e com essas ferramentas produzem novos conhecimentos caracterizando esse ambiente como um ambiente pedagógico que não existe a priori. Na montagem dos robôs pode-se utilizar peças mecânicas tais como rosca sem-fim, engrenagens, eixos, cremalheiras, correias dentadas etc., para montar estruturas mecânicas. Essas peças são devidamente acopladas respeitando-se alguns princípios da mecânica, da física, da matemática, da arquitetura, buscando

trabalhar os conceitos de uma forma interdisciplinar. Além disso, utilizam-se componentes elétricos como motores, sensores de luz, toque, temperatura, som, posição, lâmpadas que possibilitam o acionamento dos dispositivos a partir de comandos enviados pelo computador.

No cenário da RP, automatizar um robô significa criar uma interação, por intermédio do programa elaborado no computador, entre peças mecânicas e componentes elétricos obtendo, por exemplo, como resultado, os movimentos de um braço manipulador. Ou seja, precisa-se descrever para o computador, em termos de uma seqüência lógica devidamente estruturada, os comandos que acionam luzes, motores e sensores conectados a uma determinada máquina, para que, ao colocá-la em funcionamento, essa seqüência lógica represente, mais próximo possível, a máquina projetada e idealizada. Caso isso não ocorra, abre-se uma gama de possibilidades de depuração propiciando a interação com diversas áreas de conhecimento numa abordagem interdisciplinar. Os kits de montagem são compostos por peças diversas, motores e sensores controláveis por computador e softwares que permitam programar o funcionamento dos robôs montados. Dentre os principais fornecedores de Kits de robótica encontram-se o SYMPHONY[17], LEGO[18], LYNXMOTION[19], FISCHERTECHNIK[20] entre outros.

5.2.2 Competições de Robótica

O objetivo das competições de robótica é promover e divulgar o desenvolvimento e avanço tecnológico em robótica autônoma e mecatrônica, além de promover a interação de pesquisadores e estudantes de robótica por meio de competições e desafios em diferentes categorias. A *RoboCup*[21], uma competição iniciada em 1997, que promove um campeonato de futebol entre robôs projetados por alunos de diferentes instituições de ensino, conforme descrito em LUND & PAGLIARINI (1998) e ilustrado na Figura 19, representa um dos mais conceituados eventos mundiais de competição de robótica.

Vale ressaltar que já existe a *RobCup* Brasil[22], atualmente na terceira edição, um evento realizado juntamente com o *XXVI* Congresso da Sociedade Brasileira de Computação (SBC)[23] realizado em

17 http://www.symphony.com.br/robeduc.php

18 http://www.lego.com/eng/education/mindstorms

19 http://www.aliatron.com/lynxmotion/

20 http://www.fischertechnik.de/en/index.aspx

21 http://www.robocup.org/

22 http://jri.sorocaba.unesp.br/

Campo Grande, MS, entre 17 e 20 de julho de 2006. As Competições de Robótica 2006 são eventos associados à Jornada de Robótica Inteligente (JRI) e ao Encontro de Robótica Inteligente (EnRI). Trata-se de uma ação conjunta da SBC, Sociedade Brasileira de Automática (SBA), *IEEE* e *RoboCup,* para divulgar e promover os avanços tecnológicos na área de Robótica Móvel e Mecatrônica. A competição reúne três eventos distintos:

- A Competição IEEE Brasileira de Robôs encontra-se em sua IV edição e é patrocinada pelo IEEE *Latin American Robotics Council*. Seu objetivo é promover o desenvolvimento científico e tecnológicos nas áreas de Robôs Autônomos e Mecatrônica, na América Latina. Estas competições possuem grande parte dos desafios encontrados pela robótica móvel do mundo real, tais como, navegação em ambientes dinâmicos, exploração de ambientes, realização de procedimentos de contingência em ambientes perigosos ou insalubres, monitoramento ambiental, etc.
- A RoboCup Brasil, atualmente em sua terceira edição, é um projeto conjunto internacional para promover a IA, robótica e campos relacionados. Trata-se de uma tentativa de estimular a IA e a robótica inteligente pelo oferecimento de problemas padrão onde diversas tecnologias podem ser empregadas.
- A RoboCupJunior Brasil, realizada pela primeira vez no país, é uma iniciativa orientada à educação que tem por objetivo introduzir a robótica aos jovens estudantes de ensino fundamental, médio e técnico. Mais do que o desenvolvimento tecnológico, o foco neste liga é a educação e o estímulo ao surgimento de futuros pesquisadores.

Em Belo Horizonte, a maior competição de robótica é organizada pela Sociedade de Usuários de Informática e Telecomunicações (SUCESU). A SUCESU nacional foi criada em 1965 por um grupo de usuários que se reunia para debater e buscar soluções em conjunto para os seus problemas. As Regionais surgiram em diferentes anos. Em 14 de agosto de 1968, a SUCESU-MG foi fundada[24]. Anualmente a SUCESU realiza a INFORUSO, uma feira de tecnologia da informação e comunicação que em 2006, completa sua 22º edição[25].

Durante a INFORUSO SUCESU, é realizada a "Copa Robótica", que visa fomentar a criatividade de estudantes dos níveis fundamentais, médio, médio técnico e universitário no desenvolvimento de projetos tecnológicos, além de incentivar a formação de profissionais preparados para o mercado de trabalho. A competição é caracterizada pela disputa dos robôs desenvolvidos pelos alunos das diversas instituições inscritas e é composto de 02 (dois) níveis, o intermediário, indicado para

---

23 http://www.sbc.org.br/

24 http://www.sucesumg.org.br

25 http://www.sucesumg.org.br/inforuso2006/

equipes de instituições de ensino fundamental e médio e o avançado, indicado à equipes de instituições de nível superior.

5.2.3 Abordagem metodológica típica da RP

É embasado, principalmente, na teoria construcionismo de Papert, que a robótica firmou o seu alicerce na educação, passando a ter então uma denominação específica para este ramo, sendo, por alguns, chamada de robótica pedagógica e por outros, de robótica educacional. Porém tanto uma quanto a outra significam a mesma coisa, isto é, ambiente de aprendizagem que reúne materiais de diversos tipos e diversas peças em formatos e cores diferentes, motores e sensores controláveis por computador e softwares que permitam programar, de alguma forma, o funcionamento dos modelos montados.

A robótica, acima de tudo, constitui nova ferramenta que se encontra à disposição do professor, por meio da qual é possível demonstrar na prática muitos dos conceitos teóricos, às vezes de difícil compreensão, motivando tanto o professor como principalmente o aluno.

Na interdisciplinaridade a robótica pedagógica firma outro alicerce, pois a partir do momento em que se começa a trabalhar conceitos de várias disciplinas com um único objetivo, saem ganhando tanto o aluno quanto o professor.

Neste sentido, PAPERT (1986), afirma que dizer que estruturas intelectuais são construídas pelo aluno, ao invés de ensinadas por um professor não significa que elas sejam construídas do nada. Pelo contrário, como qualquer construtor, a criança se apropria, para seu próprio uso, de materiais que ela encontra e, mais significativamente, de modelos e metáforas sugeridos pela cultura que a rodeia.

5.2.4 A Metodologia LEGO

Segundo KNUDSEN (1999), a divisão educacional do grupo LEGO[26], uma indústria dinamarquesa de brinquedos, elaborou uma proposta pedagógica em parceria com renomados institutos de ensino e pesquisa, dentre eles o MIT, e está sendo utilizada nos Estados Unidos, na Europa e em mais de 50 países do mundo. Nessa proposta, os alunos se envolvem em atividades dinâmicas, que fazem parte de um projeto, no qual utilizando a experiência própria eles passam a ser protagonistas de sua

26 http://www.lego.com/education/default.asp

aprendizagem. A abordagem de projetos é baseada em situações-problema, por meio das quais eles aprenderão a trabalhar em equipe, a fim de encontrar soluções criativas e idéias novas para os problemas propostos. Esse cenário de desafios os equipará com uma importante base para a aquisição de várias competências, habilidades e qualidades pessoais, a fim de viver nesse novo mundo. Um cenário que conduz à aprendizagem, segundo esta proposta, é aquele em que os alunos são desafiados de maneira equilibrada, valorizando conhecimentos e habilidades prévias, e estimulados a avançar na exploração, pesquisa e solução dos novos problemas propostos. Nesse cenário, com equilíbrio entre habilidades e desafios, eles experimentam um alto grau de satisfação.

A Metodologia LEGO, KNUDSEN (1999), se propõe a possibilitar o desenvolvimento da criatividade, das relações entre as pessoas, do trabalho em equipe, da ética e da cidadania, permitindo ao professor praticar ações que desenvolvam nos alunos motivação, memória, linguagem, atenção, percepção, emoção. Essa metodologia contempla quatro fases:

- Contextualizar

  Nesta fase, estabelece-se uma conexão dos conhecimentos prévios, que o aluno possui, com os novos e insere-se uma atividade prática, podendo ser uma situação-problema relacionada com o mundo real.

- Continuar

  Nesta fase é proposto um novo desafio, estreitamente relacionado com o tema, estimulando os alunos a entrar em uma espiral de aprendizagem, na qual a cada nível valorizam-se os conhecimentos prévios, equilibrando assim a relação de habilidades e desafios.

- Construir

  Nesta fase, os alunos farão montagens relacionadas com a situação-problema proposta pela contextualização, ocorrendo nesse momento uma constante interação entre o abstrato e o concreto. O processo de construção física de modelos proporcionará um ambiente de aprendizagem fértil para o processo de mediação a ser realizado pelo professor, que negociará conflitos, ouvirá diferentes idéias e opiniões dos grupos para os mesmos problemas propostos e orientará quanto ao uso racional e efetivo da tecnologia.

- Analisar

  Nesta fase os alunos pensam sobre como as coisas funcionam, experimentando, observando, analisando, corrigindo possíveis erros e validando, assim, o projeto.

5.2.5 MINDSTORMS LEGO

O MINDSTORMS é o nome de uma das linhas de produtos vendido pela LEGO. O carro chefe dessa linha de produtos é o *Robotic Invention System* (RIS), que é um conjunto de peças (kit) que permite a construção de robôs. Com este kit é possível a construção de robôs de maneira simples e rápida através de um conjunto de peças e um conjunto de recursos que permitem a elaboração de programas que controlam estes dispositivos. O kit fornece diversos tipos de peças: engrenagens, eixos, roldanas, polias, rodas, motores, lâmpadas, sensores, etc.

Um sucesso de público no meio da robótica educativa, o kit do LEGO Mindstorms, vendeu mais de 80.000 unidades em 1998, seu ano de lançamento, segundo MINDELL (2000). O ambiente é um revolucionário conjunto de robótica que permite criar os modelos combinando peças LEGO que incluem blocos de montar, engrenagens, motores, sensores e principalmente o *Robotic Command Explorer* (RCX)[27], que é na verdade um pequeno computador encapsulado num tijolo LEGO que atua como controlador do robô.

O RCX foi desenvolvido pela LEGO em parceria pelo MIT com o formato de uma peça LEGO, conforme figura 20. Ele é baseado em um processador Hitachi H8/300 de 8 (oito) bits com arquitetura *RISC* operando a 6 (seis) MHz. A Hitachi é uma corporação japonesa do segmento de eletrônicos e seu *web-site* institucional está disponível, em inglês, através do endereço: *http://www.hitachi.com*

O RCX possui dois tipos de memórias, 16K de ROM interna, que como o próprio nome indica, não pode ser escrita. Esta é já vem programada de fábrica e não pode ser mudada e 512 bytes de RAM estática interna (SRAM) e um adicional de 32K SRAM. A RAM pode ser escrita e lida quantas vezes for necessário, porém ela precisa ter uma fonte de força. Isto significa que se as pilhas, 6 (seis) no formato AA no total, do RCX são retiradas, o conteúdo da RAM é apagado. Sob circunstâncias normais, as baterias preservam a informação armazenada na RAM. O RCX pode armazenar até 5 (cinco) programas e permite controlar até 3 (três) dispositivos de entrada (sensores) para interação com o meio e até outros três dispositivos de saída, onde podem ser ligados motores, lâmpadas, buzinas, etc. As três portas de saída, chamadas de A, B e C, que estão localizadas quase no meio do tijolo. Os atuadores do robô (motores ou luzes) podem ser conectados a estas portas. As três portas de entrada, chamadas de 1, 2 e 3 podem ser conectadas a variados tipos de sensores, permitindo ao RCX interagir com o ambiente em que se encontra, conforme ilustrado pela Fig. 21.

---

27 Este trabalho foi baseado na versão 1.0 do RCX

O RCX inclui uma pequena tela de cristal líquido. Esta tela apresenta informações úteis tais como o valor da leitura dos sensores e estado das portas de saída. O RCX é capaz de produzir alguns *bips* em diferentes freqüências. Existem 4 botões no RCX, através deles se permite selecionar programas, iniciá-los, pará-los. Pode-se também escolher visualizar o valor dos sensores conectados ou o estado das portas de saída. O RCX se comunica com o microcomputador através de uma conexão feita por infravermelho, também é possível estabelecer comunicação com outros tijolos RCX através desta conexão, conforme demonstra a figura 22.

O conjunto inclui também diversos tipos de sensores, que permitirão ao robô interagir com o ambiente. Entre eles merecem destaque:

Sensor de luminosidade: Composto por duas partes principais: o LED (uma pequena luz vermelha) e o leitor óptico. O leitor óptico irá analisar a intensidade luminosa que o LED esta refletindo sobre determinada superfície, esta intensidade será nomeada através de um número (x). Este sensor é muito utilizado para distinguir cores, pois cada cor irá refletir, uma intensidade luminosa diferente, podendo assim cada cor ter o seu número (x) dado pelo sensor.	

O sensor de luminosidade mede a intensidade luminosa de determinado objeto, se a programação foi feita num ambiente claro ela passara a não funcionar em um ambiente escuro, devido a variação da intensidade luminosa, num dia escuro cai o valor da intensidade refletida e irá acarretar na diminuição do valor numérico(x) dado.

	Sensor de toque: O sensor de toque é composto por um contato que quando é ativado (tocado) é enviada uma mensagem para o RCX.

Sensor de temperatura: Esse sensor tem como função enviar para o RCX a	Figura 25 Sensor de temperatura

temperatura em que está submetido, tanto em ºC como em ºF.	

	Sensor de rotação e ângulo: Esse sensor é utilizado para detectar a abertura de ângulo ou a RPM (Rotações por Minuto).

O sensor de rotação e ângulo é certamente o mais preciso sensor que existe no Kit da lego, pois ele não depende de fatores da natureza como o sensor de temperatura e o de luminosidade. Já o sensor de rotação é completamente dependente de apenas uma parâmetro a quantidades de voltas que o motor deu, não irá interferir a claridade da luz,a temperatura do ambiente ou a carga das pilhas, ao atingir o número de rotações estipulados pelo programa ele irá executar a função pedida sem erros.

Além destes sensores, o RIS oferece 700 (setecentas) peças LEGO podem ser utilizadas para construir o corpo do robô, alguns exemplos estão demonstrados através da figura 27.

Figura 27 – Exemplos de peças que compõe o kit da LEGO

Fonte: EDACOM

Um aspecto relevante a ser considerado é o fator custo, pois os kits baseados na tecnologia LEGO, como o ilustrado na figura 28, viabilizaram sua adoção, se comparados aos custos proibitivos das opções até então disponíveis, além da simplificação no processo de construção dos robôs, que podem ser montados em instituições sem infra-estrutura específica para a sua montagem.

É importante salientar que já existe uma nova geração de Kits, baseados no NXT, que atualiza e substitui o RCX. Porém, esta pesquisa foi conduzida tomando como referência a geração de Kits, baseados no RCX.

Figura 29 : NXT

Fonte: LEGO

5.2.6 PROGRAMAÇÃO DO RCX

É possível escrever programas que rodam no RCX utilizando linguagens de alto nível. Estes programas uma vez escritos podem ser carregados no RCX através de uma conexão feita por raios infravermelhos. Para tal, o conjunto é provido de um dispositivo que uma vez conectado a porta serial de um microcomputador realiza esta tarefa. Este dispositivo é chamado de torre IR (do inglês *Infrared Tower*).

A figura 30 apresenta o esquema básico de comunicação entre um microcomputador e o RCX. A construção de um robô utilizando RCX consiste basicamente de quatro passos:

1. Construir o corpo do robô;
2. Escrever um programa utilizando as ferramentas de software existente no microcomputador;
3. Carregar o programa no robô;
4. Executar o programa;

A programação é feita em um computador pessoal usando linguagens de programação que pode variar conforme o contexto, como por exemplo, a linguagem LOGO, a linguagem C (*NQC*) ou uma linguagem baseada em ícones chamada ROBOLAB.

O software do RIS inclui um ambiente que permite escrever programas que podem ser executados no RCX. Esta técnica é chamada de *cross-compiling*, o que significa escrever um programa num computador, porém este programa será executado em um outro. Neste caso, o programa é escrito num microcomputador e executado no RCX. Algo que deve ser enfatizar é que existem vários ambientes de programação para o RCX. O ambiente oficial de programação ROBOLAB, o qual faz parte do conjunto RIS, é apenas um deste ambientes. Uma vez escrito, em um computador, o programa deve ser carregado no RCX. O carregador de programas do RIS é um aplicativo que é executado a partir do microcomputador e transmite, através da torre IR o programa que o RCX irá executar. Escrever programas para o RCX envolve a passar por várias camadas de software, tanto no microcomputador como no próprio RCX. A figura 31 apresenta de modo geral dessas camadas.

Uma das primeiras coisas que devem ser realizadas antes de poder utilizar o RCX é carregar o *firmware*. O *firmaware* é, essencialmente, o sistema operacional do RCX. As rotinas da ROM sabem como carregar as rotinas do *firmware* a partir da porta da torre IR e armazená-las na RAM.
O *firmware* é capaz mais do que simplesmente processar as rotinas da ROM. Este pode reconhecer e responder as operações realizadas pelos botões do painel do RCX. E o mais importante, pode receber os programas do robô através da porta da torre IR e executá-los.

Embora a primeira vista o *firmaware* e os programas do robô pareçam à mesma coisa, isto não é verdade. O *firmware* é na verdade código de máquina do Hitachi H8. Em conjunto com o código de máquina do Hitachi H8 existente na ROM, o *firmware* define o sistema operacional do RCX. Este provê acesso às portas de entrada e saída do RCX. Esta provê também o meio que os programas de robô são carregados, armazenados, inicializados e parados.

Os programas de robô que são carregados no RCX não são código de máquina do H8. Eles são definidos num nível mais alto de programação e são chamados de *bytecodes*.
Desta forma, um código de máquina do H8 é bastante rudimentar, tal como "mova este valor para o registrador 1", por outro lado uma instrução em *bytecodes* é muito poderosa, tal como "ligue o motor 2 a potência máxima". O *firmware* interpreta o *bytecode* e executa a ação apropriada.

Existem ainda várias outras alternativas de linguagens de programação, como, por exemplo, o leJOS[28], um *firmware* alternativo para o RCX, que implementa uma máquina virtual Java (JVM), viabilizando a execução de programas escritos em Java nos robôs. LAVERDE (2006).

Legolog[29] é uma linguagem baseada no Prolog, enquanto o XS[30] é um dialeto da linguagem o Lisp, que foram desenvolvidas para permitir a pesquisa de inteligência artificial em Robôs construídos com o kit LEGO MINDSTORMS.

Um recurso referente ao RCX que merece menção especial é sua capacidade de trocar informações, em tempo real, com outros equipamentos do mesmo tipo que estejam a pequena distância. Esta capacidade permite utilizar vários controladores RCX trabalhando de forma cooperativa, expandindo sua capacidade e contornando muitas de suas limitações.

---

[28] http://lejos.sourceforge.net/index.html

[29] http://www.cs.toronto.edu/cogrobo/Legolog/

[30] http://www.xslisp.com/

Diferentemente do ROBOLAB que é um software proprietário comercializado pela LEGO, os programas NQC, leJOS, Legolog e XS, são softwares livres. Para uma discussão aprofundada os softwares livres recomenda-se a leitura de BASTOS (2006).

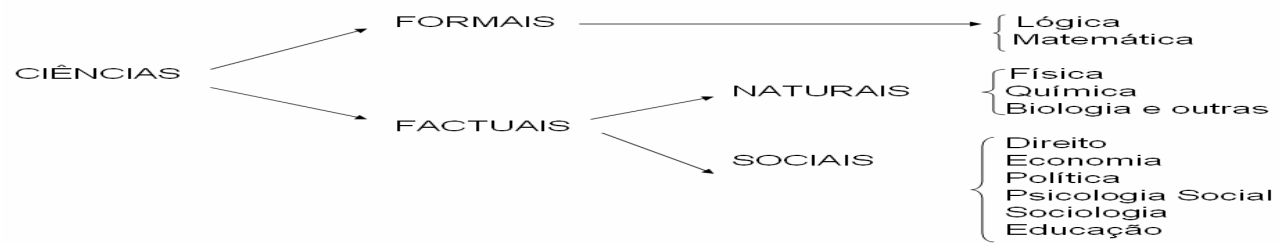
6 METODOLOGIA DE PESQUISA

6.1 O Conhecimento e a Ciência

A afirmação de que vivemos na era da informação e que o conhecimento é o ativo mais precioso às instituições e aos indivíduos é lugar comum em grande parte da literatura contemporânea[31]. Segundo LUCKESI (1996), o conhecimento é a explicação da realidade e decorre de um esforço de investigação para descobrir aquilo que está oculto, que não está compreendido ainda. Só depois de compreendido em seu modo de ser é que um objeto pode ser considerado conhecido. De acordo com LAKATOS e MARCONI (1991) pode-se distinguir quatro tipo de conhecimentos, a saber, o popular, o filosófico, o religioso e o científico. Dentre eles, aquele que baseia-se em fatos, é sistemático e que suas proposições ou hipóteses têm veracidade ou falsidade comprovadas através da experimentação é o cientifico. Em suma, não se sustenta sobre dogmas.

TRUJILLO *apud* LAKATOS & MARCONI (1991) conceitua a ciência como todo um conjunto de atitudes e atividades racionais, dirigidas ao sistemático conhecimento com objeto limitado, capaz de ser submetido à verificação, que diante das limitações da mente humana, exigem a fragmentação do real para que se possa atingir um de seus segmentos, resultando, desse fato, a pluralidade das ciências, que na figura 32 aparecem divididas entre formais e factuais.

Figura 32 – Tipos de ciência



BUNGE apud LAKATOS & MARCONI (1991, p.81)

Por ciências formais, entende-se o estudo das idéias abstratas, existentes apenas na mente humana, como por exemplo, o conceito de número, que não existem fora de nossa mente, ninguém pode vê-lo em sua forma, composição e essência. Por outro lado, por ciências factuais, entende-se, o estudo dos fatos que supostamente ocorrem no mundo, recorrem à observação e à experimentação para comprovar ou refutar suas hipóteses. Tanto as ciências formais, quanto as factuais, caracterizam-se pela utilização de métodos científicos, portanto não há ciência sem o emprego de métodos científicos.

[31] Veja o item 4.1 deste trabalho

6.2 A Ciência e o método científico

O método foi conceituado por JOLIVET (1979) como a ordem que se deve impor aos diferentes processos necessários para atingir um fim dado (...) é o caminho a seguir para chegar à verdade nas ciências, por NÉRICI (1978) como o conjunto coerente de procedimentos racionais ou prático-racionais que orienta o pensamento para serem alcançados conhecimentos válidos, e por CERVO e BERVIAN (2000) como a ordem que se deve impor aos diferentes processos necessários para atingir um dado fim ou um resultado desejado. Nas ciências, entende-se por método o conjunto de processos que o espírito humano deve empregar na investigação e demonstração da verdade.

Os antigos gregos pré-socráticos foram os precursores da investigação cientifica, primeiramente determinando que somente as questões da natureza eram passíveis de investigação. A partir de Sócrates e seus teoremas dialéticos há uma necessidade de investigação de questões tangentes da cotidianidade dos indivíduos fazendo assim, com que os problemas mais cotidianos viessem a fazer parte das investigações científicas. Por seu turno, Aristóteles determina que a lógica possa realizar tais indagações cotidianas se tivesse um modelo que pudéssemos seguir, ou seja, determinar um padrão que deve ser seguido por todos. Desta maneira, a necessidade de definição destes padrões se determinou entre os modelos dedutivo e indutivo, bem como os raciocínios logicamente aceitáveis. Tais procedimentos científicos foram aceitos e utilizados até os séculos XIV e XV. JOLIVET (1979)

Segundo CERVO e BERVIAN (2000), Galileu (1564-1642) foi um precursor teórico do método experimental, quando contradizendo os ensinamentos de Aristóteles, preconizou que o conhecimento íntimo das coisas deveria ser substituído pelo conhecimento de leis gerais que condicionam as ocorrências. O método proposto por Galileu Galilei pode ser rotulado de indução experimental, pois é a partir da observação de casos particulares que se propõe a chegar a uma lei geral. Francis Bacon (1561-1626), contemporâneo de Galileu, destaca serem essenciais à observação e a experimentação dos fenômenos reiterando que a verdade de uma afirmação só poderá ser proporcionada pela experimentação. Bacon propõe que sejam seguidos os passos:

- Realização de experimentos sobre o problema para que se possa observar e registrar, de forma sistemática, as informações coletadas;
- Analisar os resultados experimentais através da apresentação de hipóteses que sugiram explicações sobre as relações causais entre os fatos; (analisar uma ocorrência determinando a validade da mesma para ocorrências similares);

- Repetição dos experimentos em outros locais e ou por outros cientistas, com a finalidade de acumular novos dados que servirão para a formulação de hipóteses (outras ou as mesmas já formuladas). Teste das hipóteses com nova repetição experimental.

O grau de confirmação das hipóteses depende da quantidade de evidências favoráveis. Formulação de leis gerais para o fenômeno estudado, fundamentadas nas evidências experimentais obtidas com posterior generalização destas leis para os fenômenos similares ao que foi estudado. Portanto, na base do método proposto por Bacon estava a necessidade de um complemento para o método indutivo aristotélico e principalmente uma nova consciência de que para que ocorra uma pesquisa devemos fazer uso do "método das coincidências constantes", pois somente dizer que uma causa determina um efeito não era sustentação para determinação de uma ocorrência. Mas, sim que a constatação de que um fenômeno depende de varias ocorrência para que possamos determinar sua validade.

Foi com Descartes (1596-1650) que o procedimento científico teve sua fundamental evolução. Ele determina a importância de se buscar meios para a investigação cientifica e principalmente determinar que na sua totalidade não há condições passiveis para investigação de qualquer tipo de fenômeno ou fato. Deste modo, Descartes propõe um processo que se afasta em essência dos anteriores. Em vez de usar inferência indutiva, utiliza a inferência dedutiva (do geral para o particular). CERVO e BERVIAN (2000). A certeza somente poderá ser alcançada pela razão. As quatro regras clássicas de seu método são:

a) não aceitar jamais como verdadeiro uma coisa que não se reconheça evidentemente como verdadeira, abolindo a precipitação, o preconceito e os juízos subjetivos (EVIDÊNCIA);
b) dividir as dificuldades em tantas partes quantas for possível e necessário para resolvê-las (ANÁLISE);
c) conduzir ordenadamente o pensamento, começando pelos objetos mais simples e mais fáceis de conhecer até culminar com os objetos mais complexos, em uma seqüência natural de complexidade crescente (SÍNTESE);
d) realizar sempre discriminações e enumerações as mais completas e revisões as mais gerais, de forma a se ter certeza de nada haver sido omitido (ENUMERAÇÃO).

No caso das ciências factuais a análise e a síntese podem ser realizadas sobre os fatos e sobre os seres ou coisas materiais ou espirituais. A análise pode ser entendida como o procedimento que permite decompor o todo em suas partes constituintes, indo do mais para o menos complicado. Já com a síntese é feita a reconstituição do todo, após a análise preliminar (do simples para o complexo). Em ambos deve haver um procedimento gradual sem a omissão de etapas intermediárias. Nas ciências naturais a análise sempre precede a síntese. ALVES-MAZZOTI (1999).

6.3 O método científico e a abordagem

O debate quantitativo-qualitativo nas ciências sociais é ainda um debate aberto. Guba e Lincoln (1994) nos fornecem alguns elementos de diferenciação entre as duas abordagens. A investigação quantitativa atua em níveis de realidade na qual os dados se apresentam aos sentidos e tem com campo de práticas e objetivos trazer à luz fenômenos, indicadores e tendências observáveis. A investigação qualitativa trabalha com valores, crenças, hábitos, atitudes, representações, opiniões e se adequa a aprofundar a complexidade de fatos e processos particulares e específicos a indivíduos e grupos. A abordagem qualitativa é empregada, portanto, para a compreensão de fenômenos caracterizados por um alto grau de complexidade interna. Estes autores consideram que, do ponto de vista metodológico, não há contradição assim como não há continuidade entre investigação quantitativa e qualitativa. Ambas são de natureza diferente. Consideram ainda que, do ponto de vista epistemológico, nenhuma das duas abordagens é mais científica do que a outra. Ou seja, uma pesquisa, por ser quantitativa não se torna “objetiva” e, portanto, “melhor”. Da mesma forma, uma abordagem qualitativa em si não garante a compreensão em profundidade de um determinado fenômeno.

Em resumo, pode ser dito que ambas são de natureza diferenciada, não excludentes e podem ou não ser complementares uma à outra na compreensão de uma dada realidade. Se a relação entre elas não é de continuidade, tampouco elas se opõem ou se contradizem. Somente quando as duas abordagens são utilizadas dentro dos limites de suas especificidades é que podem dar uma contribuição efetiva para o conhecimento.

6.4 Metodologia da pesquisa

Este trabalho procurou seguir as características da investigação qualitativa, descritas por BOGDAN e BIKLEN (1994):

> "Na investigação qualitativa, a fonte direta de dados é o ambiente natural, constituindo o investigador o instrumento principal". Nesta abordagem não existe a simulação do ambiente em laboratório, mas a pesquisa ocorre diretamente no ambiente estudado.
>
> "A investigação qualitativa é descritiva".
>
> "Os investigadores qualitativos interessam-se mais pelo processo do que simplesmente pelos resultados ou produtos".
>
> "Os investigadores qualitativos tendem a analisar os seus dados de forma indutiva". O problema de pesquisa, o objeto a ser investigado define-se melhor enquanto o pesquisador levanta informações em campo.
>
> "O significado é de importância vital na abordagem qualitativa". O foco está no sentido, no ponto de vista das pessoas. Ao participar do cotidiano das pessoas, o investigador qualitativo busca entender as diferentes maneiras que diferentes grupos percebem um dado fenômeno.

Os métodos qualitativos e quantitativos não se excluem. Embora difiram quanto à forma e à ênfase, os métodos qualitativos trazem como contribuição ao trabalho de pesquisa uma mistura de procedimentos de cunho racional e intuitivo, capazes de apoiar uma melhor compreensão dos fenômenos estudados. Pode-se distinguir o enfoque qualitativo do quantitativo, mas não seria correto afirmar que guardam relação de oposição.

A preferência pelo uso do estudo de caso deve ser dada quando do estudo de eventos contemporâneos, em situações onde os comportamentos relevantes não podem ser manipulados, mas onde é possível se fazer observações diretas e entrevistas sistemáticas. O estudo de caso se caracteriza pela capacidade de lidar com uma completa variedade de evidências, documentos, artefatos, entrevistas e observações. YIN (1989, p.19).

Tomando como referência GIL (2002) quando afirma que o estudo de caso é uma inquirição empírica que investiga um fenômeno contemporâneo dentro de um contexto da vida real, este trabalho foi delimitado a uma instituição de ensino superior sediada em Belo Horizonte, envolvendo uma disciplina do oitavo período do curso de Sistemas de Informação. A pró-reitoria de graduação, a coordenação do curso e o professor da disciplina foram previamente consultados e aprovaram a condução do experimento. Esta abordagem metodológica pode ser comparada a uma abordagem quantitativa, tipicamente norte-americana, disponível em DEEPAK e MEEDEN (1998), FAGIN e MERKLE (2002) e KLASSNER (2002).

6.5 Delimitação do campo de pesquisa

O Centro Universitário UNA[32] é uma entidade particular de Educação, Ciência e Pesquisa, fundada em 20 de outubro de 1961 e reconhecida pelo Decreto Federal nº 74.455, de 26 de agosto de 1974. Inicialmente era denominada Instituto de Relações Públicas e em 1966, quando foi credenciada, passou a se chamar Faculdade de Ciências Administrativas. À época funcionava na residência de Afonso Pena II que hoje integra o patrimônio histórico de Minas Gerais. Em 1976, o Centro muda seu nome para Centro de Pesquisas Educacionais de Desenvolvimento de Recursos Humanos, CEPEDERH. Três anos depois são autorizados os cursos de Contabilidade e Comércio Exterior, reconhecidos em 1983 pelo Conselho Federal de Educação. Em 1987 foi autorizado o curso de Tecnologia e Processamento de Dados e em 1992 o curso de Economia. Em 1998 o Conselho Nacional de Educação autorizou o funcionamento do curso de graduação em Gestão de Hotelaria, Turismo e Lazer. Em 1991 a então Faculdade de Ciências Gerenciais foi transferida para a Rua Sapucaí onde permaneceu até 2003. Em 1996 o edifício da rua Aimorés ficou pronto. Em outubro de 2000 recebeu o título de Centro Universitário e em 2002 abriu seu terceiro campus no Buritis. No mesmo ano o curso de Direito foi autorizado pelo Ministério da Educação. Em 2003, passou por uma mudança em seu controle acionário que acabou por refletir em seu conceito e imagem como instituição de ensino superior. No ano seguinte, foram criados novos cursos nas áreas de Comunicação e Artes, Ciências Humanas e Ciências Biológicas e Saúde. Em 2004, o Centro Universitário inaugurou um novo campus na av. Afonso Pena, onde são oferecidos os cursos da UNATEC. E, em agosto de 2005, abriu seu quarto campus, o Liberdade. A UNA tem como missão fundamental a formação de profissionais para o mercado de trabalho, pautados pelo desempenho superior, comportamento ético e postura cidadã. Busca da excelência acadêmica e administrativa no lançamento de cursos em Minas Gerais e na adoção de práticas modernas de ensino, sempre alinhadas ao mercado.

---

[32] Informações reproduzidas do *site* oficial da instituição: http://www.una.br

Conta atualmente com um corpo discente de aproximadamente 10.000 (dez mil) alunos distribuídos entre os 22 (vinte e dois) cursos de graduação, reconhecimentos pelo MEC, e dos 15 (quinze) cursos de graduação tecnológica, ofertados através do Instituto UNA de tecnologia (UNATEC) e das faculdades de Ciências Biológicas e da Saúde, Ciências Humanas, Comunicação e Artes e Ciências Sociais Aplicadas (FCSA). No contexto da FSCA, encontra-se o curso de bacharelado em sistemas de informação (SI).

O curso de SI segundo o *site* da UNA, tem como meta desenvolver profissionais com espírito prático e empreendedor, senso de análise crítica, criatividade, talento gerencial, conhecimentos de informática e capacidade para trabalhar em equipe, itens fundamentais para se diferenciar no mercado de trabalho. Após 8 (oito) semestres confere ao egresso do título de Bacharel em Sistemas de Informação que, segundo a UNA, é o agente que analisa problemas, pesquisa e aplica os princípios de computação de acordo com as características de cada empresa, melhorando o planejamento e a tomada de decisões das organizações. De acordo com informações obtidas na página oficial da instituição, o curso de SI apresenta como diferenciais excelentes professores com titulação e experiência de mercado, laboratórios de informática com equipamentos avançados e softwares do mercado, curso com vivência prática com base nas linguagens tecnológicas mais modernas, parcerias com empresas como a Microsoft, Borland, Datasul, SUCESU, dentre outras e convênios de estágios com grandes empresas de mercado e bolsas para monitoria e iniciação científica. Finalmente, participação em campeonatos de robótica e eventos da Sociedade Brasileira de Computação (SBC), inclusive com premiações recentes.

O curso de SI foi reconhecido pela portaria, MEC, número 2.239 de 23.6.2005 e propõe as seguintes habilidades para os egressos: programação, análise e desenvolvimento de projetos de informática e administrativos, gerência e implantação de sistemas e recursos de informática nas organizações, racionalização de rotinas e fluxos de informação das organizações e desenvolvimento de aplicações de informática e criação de novos produtos e serviços. Uma cópia da grade curricular do curso de SI, está disponível do anexo 1 (um) deste trabalho.

As observações ocorreram durante as aulas da disciplina “Tópicos especiais em informática II”, do oitavo período do curso de Sistemas de Informação da Faculdade de Ciências Sociais Aplicadas do Centro Universitário UNA, com carga horária prevista de 72 horas.

Segundo seu programa, tem como objetivo permitir ao aluno vislumbrar a compreensão da robótica, sua vivencia e seus valores agregados e a ementa prevê a formalização dos termos robótica e robôs, a construções de robôs, sensorização, programação visual e programação avançada A metodologia descrita prevê aulas expositivas seguidas de debates, demonstração e aplicação dos conceitos científicos relacionados à robótica através do uso dos kits LEGO e a programação dos robôs construídos e a bibliografia básica adotada foi GROOVER et al (1988). Seguindo o regimento do centro universitário, o semestre foi dividido em duas etapas em que foram distribuídos 50 (cinqüenta) pontos em cada e será considerado aprovado o aluno que obtiver no mínimo 70 (setenta) pontos na soma das duas etapas. Estas informações foram obtidas do plano de ensino oficial da disciplina, elaborado pelo professor.

O estudo de caso ocorreu durante o primeiro semestre de 2006, no turno da noite, no campus Buritis, com 17 (dezessete) alunos matriculados. A maioria (15) dos alunos da população observada era do sexo masculino. A faixa etária situava-se entre 19 e 29 anos.

6.6 Etapas da pesquisa

A pesquisa de campo é aquela com o objetivo de conseguir informações e/ou conhecimentos acerca do problema, para o qual se procura uma resposta. Consiste na observação, sistemática e não-participante, de fatos e fenômenos tal como ocorrem espontaneamente, na coleta de dados a eles referentes e nos seus registros. LAKATOS & MARCONI (1991, p.186)

6.6.1 Etapas da pesquisa

Tabela 2 : Etapas da pesquisa

Etapa	Atividade	Alunos	Professor	Pesquisador
1	Entrevista semi-estruturada com o prof. da disciplina. (Expectativas)		X	X
2	Análise documental (histórico)			X
3	Entrevista semi-estruturada com alguns alunos (Expectativas e motivação)	X		X
4	**Observação da utilização da Robótica Pedagógica**			**X**
5	Entrevista semi-estruturada com o prof. da disciplina. (Evolução dos grupos)		X	X
6	Entrevista semi-estruturada com alguns alunos.	X		X

7	Análise e discussão dos resultados			X

Fonte: Autoria Própria

Em um primeiro momento será realizada uma entrevista semi-estruturada, que segundo LAKATOS & MARCONI (1991, p.196) é o instrumento por excelência da investigação social, com o professor da disciplina para um levantamento sobre o perfil da disciplina e da turma, bem como suas expectativas sobre a disciplina (etapa 1).

Será realizada pelo pesquisador, uma análise dos históricos escolares dos alunos matriculados na disciplina em questão, visando o mapeamento de desempenho nas disciplinas relacionadas a programação de computadores. (etapa 2).

A aproximação, no início do semestre, com os alunos envolvidos na pesquisa será conduzida através de entrevistas semi-estruturadas com alguns alunos selecionados aleatoriamente (etapa 3).

Durante o experimento os alunos irão interagir com os robôs e serão observados pelo pesquisador. (etapa 4)

Ao final do experimento, será realizada uma outra entrevista semi-estruturada com o professor da disciplina para um levantamento sobre a performance da turma durante o semestre (etapa 5)

As impressões dos alunos envolvidos na pesquisa serão obtidas em entrevistas semi-estruturados com o mesmo grupo de alunos entrevistados no início do semestre. (etapa 6).

Finalmente a Análise e discussão dos resultados será conduzida na etapa 7.

Indicadores a serem analisados:

- Impressões do professor da disciplina.
- Auto-avaliação dos alunos (amostra).
- Desempenho na avaliação formal do professor.
- Impressões do pesquisador.

7 ESTUDO DE CASO

Conforme descrito no item 6.4 – "Metodologia de pesquisa", neste trabalho foi adotada uma abordagem qualitativa, por opção do pesquisador. Neste tipo de estudo, segundo BOGDAN e BIKLEN (1994) deve-se considerar os seguintes aspectos, na investigação qualitativa, a fonte direta de dados é o ambiente natural, constituindo o investigador o instrumento principal, não existe a simulação em laboratório, mas a pesquisa ocorre diretamente no ambiente a ser pesquisado. A investigação qualitativa é descritiva e os investigadores qualitativos interessam-se mais pelo processo do que simplesmente pelos resultados ou produtos, ou seja, o significado é de importância vital na abordagem qualitativa. Considerando as peculiaridades do trabalho, optou-se por conduzir um estudo de caso, que de acordo com GOODE e HATT (1969) é considerado um tipo de análise qualitativa.

7.1 Descrição e análise do caso

Uma análise da descrição do curso, apresentada no item 6.5 "Delimitação do campo de pesquisa" deste trabalho, demonstra a relevância do desenvolvimento de programas de computadores em sua proposta.

Esta relevância fica evidenciada pela lista de habilidades listadas para seus egressos, onde a programação figura como a primeira e pelo conjunto de disciplinas diretamente relacionadas a esta habilidade, presentes em 7 (sete) dos 8 (oito) períodos do curso, conforme demonstra a Tabela 3.

Tabela 3 - Disciplinas relacionadas ao desenvolvimento de programas de computadores

Período	Disciplina(s)
1º	Introdução à lógica (IL) e Teoria da computação (TC)
2º	Técnicas de programação de computadores I (TP1)
3º	Técnicas de programação de computadores II (TP2)
4º	Técnicas de programação de computadores III (TP2)
6º	Inteligência Artificial (IA) e Computação Avançada (CA)
7º	Engenharia de software (ES)
8º	Tópicos Especiais em Informática II (RP)

Fonte: Autoria Própria

Logo no primeiro período, a disciplina "Introdução à lógica" trabalha os métodos e princípios usados para distinguir entre o raciocínio correto e o incorreto, uso de linguagens, falácias formais e informais, diagramas, tabelas verdade, notação simbólica, dedução de provas e indução. Esta disciplina introduz as noções fundamentais e técnicas da lógica formal que podem ser utilizadas em diferentes áreas e como suporte para outras disciplinas. A disciplina "Teoria da computação" introduz os conceitos discutidos nos capítulos "2-O Computador e sua Arquitetura" e "3-Programação de Computadores", deste trabalho, como por exemplo, arquitetura de *Von Newmann*, números binários, compilação, interpretação, algoritmos e Pseudo-linguagem, etc.

Durante o segundo período, a disciplina "Técnicas de programação de computadores I" discute o paradigma de programação imperativo[33] e suas principais ferramentas, como, por exemplo, variáveis, comando condicional, laços de repetição e de programação estruturada[34], como procedimentos e funções. Normalmente utiliza as linguagens "C" e "Pascal". No 3º período, a disciplina "Técnicas de programação de computadores II" trabalha a questão dos apontadores (ponteiros) e tratamento de arquivos, normalmente utilizando a linguagem de programação estruturada "C".

A disciplina "Técnicas de programação de computadores III", do 4º período, introduz a programação orientada por objetos (OOP), normalmente utilizando a linguagem de programação "C++" ou "Java"[35]. No 6º período na disciplina "Inteligência Artificial (IA)", são apresentados os paradigmas funcional[36] e lógico[37]. E a disciplina "Computação Avançada" introduz a discussão dos tipos abstratos de dados (ADT) como listas, filas, pilhas e árvores. Normalmente utilizando a linguagem de programação "C++" ou "Java". Aspectos como as práticas de desenvolvimento de software, análise de problemas de desenvolvimento de software, modelagem de soluções de problemas e métricas no desenvolvimento de *software* fazem parte da disciplina "Engenharia de software" do 7º período. O planejamento do curso prevê que o aluno deva atingir o oitavo e último período do curso com um razoável arcabouço teórico no que concerne a programação de computadores.

---

33 Imperativo — Veja o item 3..1.1 O Paradigma

34 Estruturada — Veja o item 3.1.1.1 A Programação

35 por Objetos — Veja o item 3.1.2 O Paradigma Orientado

36 Veja o item 3.1.3 O Paradigma Funcional

37 Veja o item 3.1.4 O Paradigma Lógico

Esta disposição das disciplinas, durante o curso, remete à teoria dos campos conceituais de Vergnaud, que propõe que o domínio de um campo conceitual não ocorre em alguns meses, nem mesmo em alguns anos. Ao contrário, novos problemas e novas propriedades devem ser estudados ao longo de vários anos para que os alunos progressivamente os dominem. De nada serve tentar contornar as dificuldades conceituais; elas são superadas na medida em que são encontradas e enfrentadas, mas isso não ocorre de um só golpe. VERGNAUD *apud* MOREIRA (2002).

A disciplina observada foi "Tópicos especiais em informática II", do último período do curso de Sistemas de Informação da Faculdade de Ciências Sociais Aplicadas do Centro Universitário UNA, com carga horária prevista de 72 horas aula. Na primeira etapa do semestre, as aulas foram ministradas na sala 306 que dispunha de 2 (dois) quadros brancos dispostos em "V" como principal recurso de apoio pedagógico para o professor.

Considerando as características do curso e da disciplina descritas no item 6.5 "Delimitação do campo de pesquisa" deste trabalho, e destacando a divisão do semestre em duas avaliações, foi favorecida uma abordagem em duas etapas, a saber, a primeira introduz os conceitos elementares da robótica e prioriza o tratamento dos aspectos físicos e mecânicos da montagem dos robôs como, por exemplo, estruturas e forças, alavancas (ponto de apoio, esforço e carga), engrenagens, rodas e eixos, polias e polias motorizadas, etc. Esta etapa funcionou como alicerce para a segunda, que priorizou a automação dos robôs, pela utilização dos sensores e programas desenvolvidos pelos alunos nas linguagens ROBOLAB e NQC.

De acordo com esta abordagem, foram discutidas três aulas que representam três momentos significativos do estudo em questão. O momento I contempla a primeira etapa do semestre, com ênfase na montagem dos robôs. Os momentos II e III contemplam a segunda etapa, com ênfase na programação, sendo que o momento II focado na utilização da linguagem ROBOLAB e o momento III focado na programação com o emprego da linguagem NQC.

7.2.1 Discussão do momento I – Montagem dos Robôs

Esta aula ocorreu no dia 28 de março de 2006, com o tema "Rodas e Eixos" e teve como desafio a construção de uma esteira rolante. Foi conduzida em sala, por se tratar de uma atividade de montagem dos "robôs" sem a programação, ao contrário dos outros momentos discutidos, que ocorreram em um laboratório. Foram utilizados kits MINDSTORMS LEGO[38].

Após as saudações e uma breve reflexão sobre as discussões anteriores, o professor iniciou a discussão do tema a ser abordado no dia, as rodas e os eixos. Ele tomou como ponto de partida para sua explanação a constatação de que engrenagens (rodas dentadas) adjacentes giram em sentidos opostos, conforme ilustrado pela figura 33.

Também foi relembrado, pelo professor, que as engrenagens são descritas pelo número dos dentes que têm. Uma engrenagem 24, por exemplo, tem 24 dentes.

A Figura 34 mostra 4 (quatro) exemplos engrenagens que acompanham o kit 9793 da LEGO, da esquerda para a direita, 40, 24, 16 e 8 dentes.

Após a breve fala inicial do professor, os alunos organizaram-se em grupos, referenciados como I, II, III e IV. O grupo I foi composto por 4 (quatro) alunos, o grupo II por 3 (três) alunos, o II por 5 (cinco) alunos e o IV por 5 (cinco) alunos, totalizando os 17 (dezessete) alunos presentes. Durante a pesquisa, os integrantes dos grupos tenderam a se manter fixos, com mudanças ocasionais. Quando questionados sobre este comportamento, a resposta mais comum, foi que os grupos eram formados em virtude de afinidades pessoais.

[38] Veja o item 5.2.4 MINDSTORMS LEGO, deste trabalho

Assim que os grupos se formaram, foram distribuídos os kits básicos de robótica (vermelho) e os materiais didáticos de apoio usados durante a aula.

O professor então demonstrou que duas engrenagens, uma de 40 e outra de 8 deveriam ser fixadas em um eixo, montado sobre uma plataforma, de acordo com as instruções do livreto "D", conforme figura 35.

Neste ponto o professor passou a discutir a relação de transmissão, e demonstra que a fórmula, "saída dividida por acionador (motor ou manivela)", deveria ser aplicada às duas engrenagens, gerando o cálculo 8 (oito) dividido por 40 (quarenta), que simplificado resulta em 1 (um) sobre 5 (cinco).

Esta relação significa que a cada 5 (cinco) voltas do acionador (engrenagem de 40) a saída (engrenagem de 8) irá completar uma volta. Este modelo resulta em uma transmissão lenta, porém forte, porque o torque na saída é 5x superior ao acionador.

Em relação ao conceito de torque, a primeira lei de Newton para o movimento de rotação baseia-se na seguinte idéia: se um corpo esta girando ao redor de um eixo com velocidade de rotação constante, ele tende a manter seu estado de rotação ao redor desse eixo, ou seja, para que o corpo aumente sua velocidade de rotação, ou mude a direção do eixo de rotação, é preciso aplicar um torque sobre o corpo. Para deslocarmos um corpo sobre uma superfície aplicamos uma força sobre ele. Agora, se quisermos girar um corpo ao redor de um ponto ou de um eixo devemos aplicar-lhe um torque. O torque tende a girar ou mudar o estado de rotação dos corpos, representando o efeito girante de uma força. CONVITEAFISICA (2006).

Os grupos passaram a dedicar-se à montagem do modelo solicitado de acordo com as instruções recebidas e o material didático distribuído. Os alunos interagiram entre si, recorrem ao professor e em alguns casos, observaram as soluções adotadas pelos outros grupos.

O professor passou então a discutir o conceito de engrenagem auxiliar, pois para que a engrenagem de saída gire no mesmo sentido da engrenagem acionadora, deve-se utilizar um número ímpar de engrenagens. Conforme apresentado na figura 37 em uma relação 24, 24, 40.

Com base nos conhecimentos adquiridos, foi apresentado então o desafio de se construir uma esteira transportadora rolante, usando o material didático como instrumento de apoio.

Ao olhar do pesquisador, merece destaque o fato de que enquanto o professor estava atendendo ao grupo I, os outros grupos (II, III e IV) permaneceram imersos em suas atividades, demonstrando grande interesse por obter uma solução para o desafio proposto. Veja figura 38.

Ao cabo do período determinado, todos os quatro grupos haviam concluído a montagem da esteira rolante, contando com eventuais intervenções do professor. Durante o processo ficou evidente o ambiente agradável com freqüentes interações entre os alunos do mesmo grupo, mesmo quando ocorriam erros. Alguns grupos até mesmo faziam registros fotográficos durante o processo. Compare as figuras 39 e 40.

O grupo I, em virtude da presença de um aluno extremamente motivado e interessado no assunto e com maior destreza nos conceitos de mecânica, consegue montar espontaneamente uma estrutura mais complexa que extrapola a solicitação do professor. Veja a figura 41.

Ao ser questionado sobre os possíveis desdobramentos deste projeto, a esteira transportadora rolante, o professor argumentou que a introdução de sensores de luminosidade que permitiriam ao programa determinar velocidades diferentes de acionamento do motor em virtude da cor da peça a ser transportada seria um desafio interessante para as aulas posteriores, após a introdução da programação.

Pela observação, foi possível perceber grandes semelhanças entre a aula analisada e a metodologia LEGO descrita no item 5.2.4 deste trabalho. Esta observação remete a COSTA e OLIVEIRA (2004, p. 116), que afirmam que dentro de uma perspectiva construtivista (interacionista) do processo educacional, a aprendizagem do aluno resulta de sua interação com o objeto do conhecimento, intermediada por educadores. A utilização no processo didático pedagógico de ferramentas capazes de promover interações mais efetivas e reais como em atividades práticas de laboratórios em atividades com a presença de novas tecnologias da informação e comunicação (NTIC), deve ocorrer fundamentada na possibilidade de os alunos construírem um maior conhecimento e uma maior autonomia de aprendizagem.

O aluno típico do curso de SI do centro universitário UNA não detêm informações expressivas sobre conceitos de mecânica, fundamentais para a montagem dos robôs. Contudo, após esta primeira etapa, exemplificada pela aula comentada, a turma demonstrou ter desenvolvido habilidades suficientes para avançar para a automação dos robôs montados, usando a programação de computadores. Esta demonstração deu-se pela avaliação, obrigatória em virtude de uma norma regimental do centro universitário, referente à primeira etapa, ocorrida no dia 06 de abril de 2006, conforme o planejamento do professor elaborado no início do semestre letivo e formalizado no plano de ensino oficial da disciplina. A avaliação ocorreu na sala onde haviam ocorrido as aulas até aquela data.

Seguindo o esquema das aulas, foi proposto como desafio, ilustrado na figura 42, a construção de um maquinário que deveria simular um guindaste. O tempo de duração da atividade foi de 2 (duas) horas aula (100 minutos). A turma se organizou em 4 (quatro) grupos, sendo 3 (três) com 4 (quatro) integrantes e 1 (um) com 5 (cinco) , totalizando os 17 (dezessete) alunos

matriculados na disciplina. Cada grupo poderia utilizar um KIT e o material didático de apoio.

Os alunos somente poderiam interagir dentro do grupo e o professor orientou todo o processo. Não foi solicitado que os grupos desenvolvessem um programa de computador para controlar o artefato, pois este tema só viria a ser trabalhado na segunda etapa do semestre.

Como não havia um roteiro pré-determinado para a montagem do guindaste, que representava um sumário de todas as atividades anteriores, a criatividade foi fator determinante para a conclusão do desafio. Os critérios de avaliação, explicitados pelo professor, no início do processo estão descritos na Tabela 4.

Tabela 4 - Os critérios de avaliação da 1ª etapa

TAREFA	MONTAGEM DO(A)	PONTUAÇÃO
1	Guindaste	8
2	Base Giratória	6
3	Base Móvel	6
4	Motorização do gancho	4
5	Trava do eixo	2
6	Trava das rodas O guindaste só pode se mover para a frente.	2
7	Trava da base giratória	2
	TOTAL	**30**

Fonte: Autoria Própria

Convém esclarecer que os outros 20 (vinte) pontos foram distribuídos pelo professor, durante as atividades realizadas em sala. Desta maneira, foram totalizados os 50 (cinqüenta) pontos referentes à primeira etapa do semestre.

No início do processo, o professor orientou que os grupos não deveriam simplesmente copiar o material didático de apoio, veja Figura 35, e que deveriam trabalhar em grupos, caso contrário, não haveria tempo hábil para concluírem a montagem. Merece registro o fato de que 3 (três) dos grupos adotaram a estratégia de dividir a montagem entre os integrantes, e um grupo adotou a postura de uma montagem coletiva, sendo o mais bem sucedido, pois os outros 3 (três) grupos apresentaram dificuldades na montagem final das partes.

As dificuldades encontradas durante o processo foram superadas, em alguns casos, com a intervenção do professor, que transitava entre os grupos, interagindo com os alunos durante o

processo, questionando as soluções adotadas e oferecendo alternativas sem, contudo, determinar a adoção de algum procedimento, conforme ilustrado na Figura 43.

Apesar das diferentes estratégias adotadas, e das dificuldades encontradas, os 4 (quatro) grupos conseguiram concluir a tarefa no tempo estipulado. Quando considerava concluída a montagem o grupo submetia o guindaste ao professor, como na Figura 44, que então aferia o grau de atendimento a cada um das tarefas demandadas expressas na Tabela 4 e conferia a pontuação que julgava adequada.

Foi possível observar, durante a avaliação, serenidade nos alunos, ao contrário das avaliações "tradicionais", em que a média dos alunos tende a ficar apreensiva. Tamanha era a descontração, que alguns alunos chegaram a interromper o trabalho para fotografar o grupo e o guindaste, como se tivessem orgulho de sua produção.

Ao olhar do pesquisador, foi possível estabelecer uma convergência entre os eventos observados durante a avaliação e o referencial teórico utilizado, o construcionismo, descrito no item 4.2.2 deste trabalho, demonstrando uma clara divergência ao que FREIRE (1996) chama de "educação bancária".

Após a conclusão da primeira etapa do semestre, esperava-se que os alunos tivessem construído conhecimentos suficientes sobre os aspectos mecânicos dos robôs, pela construção de diversos modelos, para que pudessem então, ser apresentados ao mundo dos robôs autônomos, pela interação com o ambiente através dos sensores e controlados por programas a ser desenvolvidos na linguagem ROBOLAB, retratada no momento II, e da linguagem NQC, retratada no momento III.

Na segunda etapa do semestre, as aulas passaram a ser ministradas em um laboratório de informática, dotado de 20 (vinte) computadores[39] com processador *Celeron* de 2.0 *Ghz*, disco rígido de 80 *Gigabytes*, 512 *Megabytes* de memória *RAM*, monitor de 15 polegadas, teclado e mouse ótico. O sistema operacional instalado nas máquinas era o Windows XP. Neste laboratório também havia disponível um *datashow*[40], que permitia a projeção da imagem do computador reservado para uso do professor, além de 2 (dois) quadros brancos dispostos em "V", para que o professor pudesse

[39] Para maiores informações, veja o capítulo 2 deste trabalho, "O computador e sua arquitetura".

[40] Equipamento que permite projetar a imagem de um determinado computador em uma tela, com tamanho expandido.

apresentar suas notas de aula. Antes do início da discussão da programação dos robôs propriamente dita, os alunos tiveram uma explanação sobre as características do RCX e dos sensores que compõem o kit MINDSTORMS LEGO[41].

O professor demonstrou a execução de 4 (quatro) programas, previamente armazenados no RCX de um robô, tipo carro bate e volta, montado por ele, com 2 (dois) sensores de toque e 1 (um) sensor de luminosidade, além de um motor traseiro. O 1º programa só movimentava o carro para frente, o 2º parava o motor quando um dos sensores de toque era acionado, o 3º executava movimentos aleatórios e finalmente o quarto simulava o comportamento do carro bate e volta. Após a demonstração, o professor permitiu aos alunos manipular o kit avançado (verde), especialmente o RCX e os sensores.Neste ponto do curso, os alunos montaram em grupos os robôs que foram usados durante as aulas de programação utilizando o ROBOLAB e o NQC.

7.2.2 Discussão do momento II – ROBOLAB

> O programa ROBOLAB é uma linguagem de programação, desenvolvida especificamente para ser utilizada na programação do RCX e, que utiliza uma *graphical user interface* (GUI), baseada em ícones. Foi desenvolvido em parceria entre a *National Instruments* e a *Tufts University (College of Engineering)*.

O Robolab é a linguagem de programação oficial dos kits da LEGO, sendo uma marca registrada da divisão educacional da LEGO, LEGO Dacta, que detém exclusividade na sua especificação e comercialização.

Inicialmente, o professor demonstrou, através do *datashow,* os recursos básicos e os procedimentos elementares de interação com o ambiente do ROBOLAB. Por se tratar de um ambiente amigável, um exemplo é exibido na figura 48, a assimilação por parte dos alunos foi muito rápida, e a *interface* intuitiva facilitava a interação dos alunos com o ambiente de programação. Os alunos ficaram tão entusiasmados que em alguns momentos, o professor teve que solicitar a atenção dos alunos, para concluir sua breve demonstração.

---

[41] Para maiores informações, veja o item 5.2.6 MINDSTORMS LEGO, deste trabalho.

Durante a introdução, ao serem questionados pelo professor sobre quais as linguagens de programação já estudaram, eles respondem “Pascal” e “C”, em concordância ao exposto no item 7.1, descrição e analise do caso, deste trabalho. Por outro lado, os alunos quando questionados, pelo professor, sobre seus conhecimentos prévios em fluxogramas ou diagrama de blocos, que são definidos por FARRER et al (1989, p.24) como um conjunto de figuras geométricas, ligadas por setas, para indicar a seqüência de execução de um determinado algoritmo, responderam que desconheciam este recurso de programação. Há um exemplo de fluxograma na Figura 8, deste trabalho.

Pela interface do ROBOLAB é possível executar algumas tarefas administrativas[42], como, por exemplo, selecionar a porta serial através da qual será feita a comunicação entre o computador e o RCX; Atualizar o *firmware* do RCX, para transferir os programas elaborados no computador para o RCX, é necessário transferir primeiro o *firmware*. Uma vez feita a transferência do *firmware*, o RCX fica pronto para receber, interpretar e executar os programas; Verificar se o RCX responde aos sinais enviados pelo computador por intermédio da torre infravermelho. Para realizar o teste é preciso que a torre de infravermelho esteja conectada ao computador e que o RCX esteja ligado e próximo a torre (10 a 20 cm). No que tange a programação, o conceito assemelha-se a um fluxograma, onde os sensores e motores conectados ao RCX são manipulados de forma seqüencial e linear, de acordo com o paradigma imperativo[43], ou seja, programar o computador significa "dar-lhe ordens" que são executadas seqüencialmente. Do ponto de vista estrutural, pode-se afirmar, que os programas em Robolab, são listas duplamente encadeadas de componentes, onde cada nó aponta para seu antecessor e seu sucessor e nos extremos estão dispostos dois semáforos, um verde que sinaliza o inicio do programa e outro vermelho, que representa seu termino. Veja a figura 46.
O ROBOLAB apresenta 2 (duas) opções chamadas de *Pilot* e *Inventor* cada uma com 4 (quatro) níveis. O Pilot oferece *templates* (programas modelo com um formato fixo associado a eles), sendo indicado para iniciantes em programação. Tipicamente usado por crianças.

O programa Robolab Pilot nível 1, na figura 46, liga o motor na porta A, em marcha ré, com potência máxima durante 4 segundos e depois finaliza a execução. Para visualizar o esquema de conexão de motores, lâmpadas e sensores ao RCX, veja a figura 22.

---

[42] Veja o item 5.2.6 Programação do RCX, deste trabalho.

[43] Veja o item 3.1.1 O Paradigma Imperativo, deste trabalho.

O programa Pilot nível 3, na figura 47, usa as portas de saída A, B e C. Ele liga o motor na porta A, em marcha ré com potência máxima (5). A porta B recebe a instrução de ligar a lâmpada a ela conectada, enquanto a porta C, liga o motor a frente com potência 3. Após 6 (seis) segundos de espera, a lâmpada B permanece ligada e os motores A e C invertem o sentido de rotação, com as potências inalteradas e funcionam até que o sensor de toque conectado na Porta 1 seja pressionado.

O Robolab Inventor usa os mesmos ícones de comandos do Robolab Pilot. Porém, são acrescentadas várias opções de comando conforme o usuário avança pelos níveis. Ele é indicado para quem já é familiarizado com os conceitos de programação. Tipicamente usado por jovens e adultos.

A dinâmica das aulas na segunda etapa do semestre letivo seguiu o seguinte esquema:

1. Uma breve exposição inicial do professor, sobre o tema da aula, fazendo referência à(s) aula(s) anterior(es).
2. A proposição pelo professor do desafio, ou seja, o comportamento desejado para o robô.
3. O ajuste dos robôs, pelos grupos, seguindo as especificações de motores e sensores, semelhante a exposta através da figura 52 na página 92.
4. A implementação do programa controlador através do ROBOLAB ou do NQC, pelo mesmo grupo que montou o robô.
5. A transferência do programa para o robô.
6. A execução do programa pelo robô.
7. A análise e reflexão em grupo sobre o comportamento do robô observado.
8. Quando da ocorrência de comportamentos inesperados, ou diferentes do desejado, o grupo discute e retorna à montagem do robô (passo 3) ou ao programa controlador (passo 4), eventualmente com o apoio do professor.
9. Uma breve análise final, pelo professor, sobre os eventos ocorridos na aula do dia.

Como exemplo desta dinâmica, o professor propôs que fosse desenvolvido um programa através do *Robolab Inventor*, que tendo como referência a especificação de motorização representada na figura 49, fizesse com que o robô avançasse pelo acionamento dos motores A e C, com potência intermediária (3), do ponto onde estivesse durante 2 (dois) segundos e interrompesse seu movimento.

Durante as aulas observadas foi utilizado o ROBOLAB Inventor nível 4 (quatro), em que as estruturas básicas de programação estruturada (seqüência, seleção e iteração), descritas no item 3.1.1.1 “A Programação Estruturada”, deste trabalho, estão presentes através fluxos de execução, dos sensores e das estruturas *Jump* (pulo) e *Land* (pouso) que permitem criar os laços de repetição ou *loops*, no jargão da programação.

Como exemplo de comportamento esperado de um robô controlado pelo programa ilustrado através da Figura 51, ele deverá acionar o motor A em frente, enquanto o sensor de luminosidade captar do ambiente um valor inferior a 50 (cinqüenta), que significa média claridade, o laço de repetição permanece através de um *jump* (↥) e um *land* (↧). Quando a luminosidade atingir 50, o robô deverá emitir um sinal sonoro, desligando o motor A e encerrando a execução do programa.

Merece destaque ainda, o incremento gradual do nível de complexidade dos comportamentos solicitados, e conseqüentemente dos programas necessários para atingir as metas propostas. Rotineiramente, os grupos logravam êxito, superando os desafios propostos e vivendo um ambiente de saudável competição, mas sem deixar de cooperar entre eles.

Para ilustrar este incremento do nível de complexidade demandada, tipicamente, através da sofisticação do grau de autonomia do robô em sua interação com o ambiente, por meio dos sensores, foi proposto um modelo onde seriam utilizados 2 (dois) sensores de toque nas portas 1 (esquerda) e 3 (direita). Também deveriam haver 2 (dois) motores na traseira do veículo, para fornecer a tração, um na esquerda (A) e outro na direita (C). O comportamento proposto foi, que ao colidir com algum obstáculo a sua frente, o robô, retroceda e vire para a direção contrária a da colisão. Veja a figura 52.

Usando o Robolab Inventor, os alunos puderam implementar, testar, e remodelar sua solução, dentro da dinâmica de aula descrita acima. Quando ocorriam dúvidas ou impasses nos grupos, o professor intervinha, questionando as soluções adotadas e oferecendo alternativas sem, contudo, determinar a adoção de algum procedimento especifico.

Figura 53 – Professor e aluno discutindo informalmente

Como na primeira etapa, uma característica marcante nas aulas foi o ambiente agradável e

descontraído, com alunos e o professor, agachados no chão, demonstrando uma atitude incomum, nas salas de aula típicas de programação de computadores, para observar o robô e discutir eventuais modificações no projeto de robô ou do programa controlador. Figura 53.

Esta aula ocorreu no dia 25 de abril de 2006, conforme o planejamento do professor e novamente, ante ao olhar do pesquisador, foi possível perceber aprendizagem sintônica, como sugere o construcionismo, descrito no item 4.2.2 deste trabalho.

7.2.3 Discussão do momento III - NQC

Em contraste como o Robolab, o NQC é uma linguagem de programação distribuída gratuitamente na Internet[44], através da licença Mozilla Public License (MPL)[45]. O nome vem de "*Not Quite C*", algo como "não exatamente C", pois ao contrário da linguagem C, este dialeto não tem propósito geral, tendo sido especialmente projetada para a programação do RCX por David Baum. BAUM & HANSEN (2006).

Em virtude desta especificidade a linguagem NQC apresenta limitações, tais como o número máximo das sub-rotinas e as variáveis permitidas, que diferem dependendo da versão do *firmware* do RCX. Um exemplo marcante destas limitações é a falta de implementação de variáveis de ponto flutuando, o que limita o NQC a 32 (trinta e duas) variáveis do tipo inteiro. Tipicamente a migração da linguagem Robolab para a linguagem NQC ocorre em virtude de o NQC ser uma linguagem textual e mais flexível.

Para iniciantes em programação de robôs, o Robolab é indicado em virtude de sua interface baseada em ícones. As linguagens gráficas, como o Robolab, são freqüentemente mais fáceis de aprender, sem erros da sintaxe, mas geralmente são mais lentas durante a execução. Os gráficos do Robolab também limitam significativamente os tipos de programas que podem ser escritos. O processo de

---

44 http://bricxcc.sourceforge.net/nqc/

45 É uma licença para software livre de código aberto

compilação do Robolab tende a gerar programas que consomem mais memória, que é um recurso escasso no RCX[46], e apresentam uma performance inferior aos programas compilados no NQC. Por se tratar de robôs que devem interagir em tempo real com o ambiente, esta diferença de performance pode ser comprometedora. Para programadores C que não se intimidam em digitar algumas linhas de código o NQC é uma opção atraente. O NQC não possui uma interface gráfica de programação, sendo um compilador de linha de comandos. Porém, quando desejado, ou necessário, podem ser encontrados na Internet programas que fornecem ambientes integrados para construir, copilar e baixar programas, além de receber dados do RCX. Uma das IDEs[47] mais utilizadas é o *Bricx Command Center* (BricxCC). O BricxCC é um aplicativo para a plataforma Windows, originalmente desenvolvido por Mark Overmars, que pode ser obtido gratuitamente na Internet[48], atualmente na versão 3.3, através da licença MPL.

Da mesma forma que na abordagem inicial do Robolab, o professor, na aula do dia 4 de maio de 2006, iniciou as discussões sobre o NQC, demonstrando, através do *datashow,* os recursos básicos e os procedimentos elementares de interação com o ambiente do NQC através do BricxCC. Ele passou então as discutir as semelhanças do NQC em relação a linguagem C, já estudada pelos alunos em semestres anteriores e ressaltou a compatibilidade com a sintaxe da linguagem C, inclusive o fato de as duas linguagens fazerem diferenciação de maiúsculos e minúsculos, no jargão da computação linguagens *case sensitive*.

Um recurso de programação muito importante, disponível no NQC, é o suporte a execução de múltiplas tarefas simultaneamente *(multi-tasking*). O número máximo de tarefas suportadas pelo RCX é 10 (dez), sendo obrigatória a presença de pelo menos uma, chamada *main*. BAUM & HANSEN (2006, p.4). Tal qual a linguagem C, o NQC está inserido no paradigma imperativo[49] de programação e trás de forma nativa as estruturas básicas de programação estruturada.

Como exemplo, o professor propôs o desenvolvido um programa na linguagem NQC, através do BricxCC, que tendo como referência a especificação de motorização representada na figura 49, que fizesse o robô avançar através do acionamento dos motores A e C, com potência intermediária (3), do ponto onde estivesse durante 2 (dois) segundos e interrompesse seu movimento. Compare com a

---

46 Item 5.2.5 deste trabalho

47 *Integrated Development Environment* (Ambiente integrado de desenvolvimento)

48 http://bricxcc.sourceforge.net

49 Veja o item 3.1.1 O Paradigma Imperativo, deste trabalho.

figura 50, o mesmo programa no Robolab e o programa desenvolvido em NQC, apresentado na figura 54 e descrito na tabela 5.

Tabela 5 – Descrição do programa Exemplo1.nqc

CÓDIGO NQC	SIGNIFICADO
task main()	Define a principal tarefa (*task*) do programa (*main*).
{	Delimita o início da tarefa
SetPower(OUT_A+OUT_C, 3);	Define a potência intermediária (3) para os motores A e C.
OnFwd(OUT_A+OUT_C);	Avança através do acionamento dos motores A e C
Wait(200);	Espera 200 centésimos de segundo , ou 2, de segundos
Off(OUT_A+OUT_C);	Interrompe o movimento dos motores A e C
}	Delimita o fim da tarefa

Fonte: Autoria Própria

Com o emprego do software BricxCC é possível editar o código fonte, consultar o *help on-line*, compilar e transferir o programa compilado no computador para o RCX, através da torre de infravermelho. Uma característica relevante, é que enquanto o Robolab liga um motor e depois outro, o NQC, permite acioná-los simultaneamente, através da instrução OnFwd(OUT_A+OUT_C).

Para desafiar a turma, o professor propôs o desenvolvimento de um programa na linguagem NQC, utilizando um sensor de luminosidade e dois motores. O robô deveria avançar sobre uma mesa branca até atingir um limite demarcado com fita preta.

A dinâmica da aula seguiu o mesmo modelo em 9 (nove) passos, descrito previamente, e após diversas tentativas e de algumas intervenções do professor, os grupos atingiram o objetivo desejado através do programa listado na figura 55. Novamente foi notada, pelo observador, a estratégia de incremento gradual do nível de complexidade dos comportamentos solicitados, e conseqüentemente dos programas codificados pelos alunos. Um fato interessante foi o desenvolvimento, em NQC, de alguns programas que já haviam sido desenvolvidos em Robolab.

Figura 55 – 2º exemplo em NQC

```
task main()
{
  SetSensorType(SENSOR_1,  SENSOR_TYPE_LIGHT);
  SetPower(OUT_A+OUT_C,3);
  OnFwd(OUT_A+OUT_C);
```

```
  while (SENSOR_1 > 40);
  Off(OUT_A+OUT_C);
}
```

Fonte: Autoria Própria

A instrução, que compõe o programa apresentado na figura 55, "while (SENSOR_1 > 40)" permite obter em tempo real o resultado da leitura do sensor de luminosidade. Enquanto for superior a 40 (quarenta), o leitor está sobre um piso claro. Nesta segunda etapa o critério de distribuição de pontos adotado divergiu da proposta inicial que consta no plano de ensino da disciplina. Foram distribuídos 15 (quinze) pontos para as atividades desenvolvidas no Robolab, outros 15 (quinze) pontos para o projeto de robô "segue linha" e 20 (vinte) pontos em seminários referentes a tópicos previstos no ementário da disciplina: Realidade Virtual, Processamento Digital e *Personal Software Process* (PSP) e Multimídia e Linguagens Multimídia para Internet.

O projeto "Robô Futebol", previsto no planejamento, não foi implementado em virtude do ritmo das aulas ter sido mais lento do que o previsto pelo professor.

O projeto "segue linha" tomou proporções de uma competição para a turma, tendo inclusive sido realizado em um sábado pela manhã, entre 8:00 e 12:00, para que houvesse mais tempo disponível para os grupos, pois no horário habitual de aula, no turno noturno, eles teriam somente 2 (duas) horas aula, 100 (cem) minutos, o que poderia ser insuficiente. Cada grupo deveria montar um robô dotado de 1 (um) sensor de luminosidade e 2 (dois) motores, além de desenvolver o programa em NQC que o faria seguir o traçado de uma pista, nos moldes de um autódromo. O chão era coberto por um piso branco regular e o traçado demarcado por uma fita preta.

Os robôs foram submetidos a 3 (três) configurações diferentes de traçado, uma linha reta, um retângulo e um traçado irregular, dotado de 7 (sete) curvas, com graus de abertura diferentes, ligadas por trechos de retas. Durante o percurso, não poderia haver nenhuma interação dos alunos com o robô, que deveria agir de forma autônoma em relação ao ambiente. Cada um dos robôs deveria percorrer o trajeto final nos dois sentidos, ou seja, ele deveria retornar ao ponto inicial do percurso. Figura 56.

Foi declarado vencedor da competição “segue linha” o robô que percorreu o trajeto no menor tempo.

Como nas outras aulas, após experimentar, observar (inclusive o desempenho dos outros grupos), refletir individual e coletivamente, efetuar ajustes no robô ou no programa, interagir com o professor, percorrendo este ciclo diversas vezes, cada um dos grupos conseguiu completar as tarefas propostas, tornando perceptível, ao olhar do pesquisador, a construção de robôs e programas de computadores. Mas, além disto, foram observadas manifestações da construção de auto-estima, amizades, trabalho em equipe, criatividade e finalmente o almejado conhecimento técnico multidisciplinar e significativo.

Estas impressões manifestaram-se também pela análise das entrevistas com alguns alunos selecionados aleatoriamente e o professor, além da reflexão do pesquisador após a observação do processo.

7.2 Análise dos dados coletados

Neste tópico serão discutidos os dados coletados através dos instrumentos previstos nas etapas de pesquisa, descritas no item 6.6 deste trabalho.

7.2.1 Entrevista inicial com o professor da disciplina

Esta entrevista foi realizada nas instalações da UNA em 14 de fevereiro de 2006 e teve duração aproximada de 30 (trinta) minutos e será apresentada de forma resumida, buscando relatar os pontos fundamentais. Em resposta ao questionamento inicial sobre sua trajetória acadêmica, o professor relatou que esta compreendia 2 (dois) cursos de nível médio técnico, um em informática e outro em eletrônica, além da graduação em sistemas de informação pela faculdade Cotemig. O professor manifestou a intenção de ingressar em um programa de mestrado com o objetivo de pesquisar questões referentes a arquitetura e organização de computadores e robótica.

Sua experiência como professor estava limitada a cursos de nível médio em outra instituição. Este fato fomentava suas expectativas de poder trabalhar questões ligadas à construção de robôs autônomos, via programação, pois em suas experiências anteriores, o foco era em questões referentes a física (mecânica), em virtude da preocupação dos alunos em relação ao vestibular.

Apesar disto, as equipes orientadas por ele haviam conseguido obter o título de campeões, em duas oportunidades, em competições internas organizadas pela instituição na qual lecionava (Cotemig).

Suas expectativas em relação à turma eram otimistas, pois apesar de não conhecer os alunos, uma análise da proposta do curso, sugeria a possibilidade de explorar em profundidade a programação dos robôs, desejo expresso na resposta anterior. Sua proposta de atuação era bastante democrática, com aulas na forma de oficinas, onde os momentos de integração entre os alunos seriam privilegiados, em detrimento a postura “clássica” das aulas expositivas, sendo que seu papel seria mais de orientador, permitindo aos alunos fazerem escolhas próprias.

7.2.2 Descrição da Análise documental

Com o propósito de avaliar o desempenho comparativo dos alunos que compõem a população do estudo de caso, nas disciplinas de programação do curso, foi feita uma pesquisa documental nos históricos e um recorte é apresentado na tabela 6.

Merece destaque o fato de que em virtude do regime seriado adotado pelo centro universitário UNA, não é utilizado o conceito de pré-requisitos, política que possibilita, por exemplo, a um aluno que não cursou, ou curso e foi reprovado em TP2, cursar TP3.

No caso específico da disciplina "tópicos especiais em informática II (RP)", é importante notar que apesar das atividades terem sido feitas em grupos mais ou menos fixos, as notas variaram, mesmo entre alunos que rotineiramente ficavam no mesmo grupo, como no caso dos alunos 2 e 6.

Tabela 6 – Notas nas disciplinas de programação

Período	1o	1o	2o	3o	4o	6o	6o	7o	8o
Aluno* Disciplina**	IL	TC	TP1	TP2	TP3	CA	IA	ES	RP
1	73	70	83	0***	0	89	74	85	81
2	70	77	99	89	73	83	0	82	90
3	70	0	0	0	0	70	0	79	70
4	98	98	99	97	86	90	96	97	90
5	82	71	71	0	0	88	0	83	90
6	0	0	93	80	78	89	85	95	84
7	79	0	100	95	95	83	83	82	89
8	82	83	89	70	0	0	0	76	73
9	74	84	93	74	76	83	73	93	75
10	84	88	79	91	70	91	84	93	84
11	74	84	75	75	0	0	0	83	71
12	73	70	79	0	0	77	77	87	79
13	89	81	90	71	75	90	83	100	85
14	0	0	0	0	0	0	0	0	82
15	73	83	92	87	70	81	83	91	81
16	95	96	95	96	83	85	84	94	84
17	77	98	96	81	0	93	80	99	85

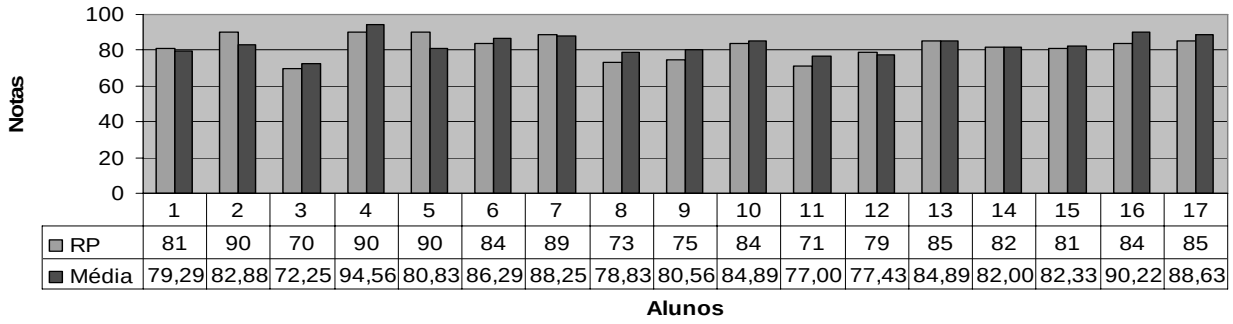
Fonte: Autoria Própria (compilado a partir dos históricos dos alunos)

*Para preservar os nomes dos alunos, eles serão identificados pelo seu número de chamada.

** Para descrição das siglas adotadas para as disciplinas, consultar a tabela 3, na página 76 deste trabalho.

*** 0 (zero) significa que o aluno não cursou, foi dispensado ou reprovado na disciplina.

Figura 57 – Gráfico comparativo entre a nota de RP e a média das notas nas disciplinas de programação



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
RP	81	90	70	90	90	84	89	73	75	84	71	79	85	82	81	84	85
Média	79,29	82,88	72,25	94,56	80,83	86,29	88,25	78,83	80,56	84,89	77,00	77,43	84,89	82,00	82,33	90,22	88,63

Fonte: Autoria Própria

A análise do gráfico contido na figura 57 fornece indícios de que a disciplina RP funcionou como uma síntese das demais disciplinas do núcleo de programação, pois tipicamente a nota final dos alunos, não apresentou variações significativas, no máximo 10%, em relação à média das notas nestas disciplinas.

Em vista deste dado isoladamente, não se pode afirmar que a introdução da RP, da forma como ocorreu, muda drasticamente o desempenho acadêmico dos alunos no que tange a programação de computadores. Entretanto, outros dados, de natureza qualitativa, demonstram uma tendência diferente.

7.2.3 Entrevista inicial com alunos da disciplina

Entrevista semi-estruturada com uma amostra de 5 (cinco) alunos que representam aproximadamente 30% da população, para obter suas expectativas e motivação em relação a disciplina. Os cinco alunos foram escolhidos aleatoriamente, mediante sorteio, através de um programa de computador que usava como referência o número de matrícula do aluno na disciplina. Desta forma foram selecionados os alunos de número 6 (seis), 7 (sete), 10 (dez), 11 (onze) e 12 (doze). As entrevistas foram realizadas nas instalações da UNA em 14 de fevereiro de 2006 e cada uma teve duração aproximada de 10 (dez) minutos e da mesma forma que na descrição da entrevista com o professor, buscar-se-á relatar os pontos fundamentais.

Aluno 6:

Apresentou expectativa moderada em relação à disciplina, em virtude de ter mantido uma média de aproximadamente 86 (oitenta e seis pontos) nas disciplinas do núcleo de programação, até aquele momento e por não trabalhar com desenvolvimento de sistemas, já não tinha muito interesse em programação, por acha-la muito abstrata. Não tinha conhecimentos prévios de robótica. Havia sido dispensado em IL e TC.

Aluno 7:

Também não apresentou grande expectativa. Já atuava profissionalmente como analista de sistemas e sua nota média era de 88 (oitenta e oito) nas disciplinas do núcleo de programação. Não tinha conhecimentos prévios de robótica. Ainda não cursou TC.

Aluno 10:

Nota média nas disciplinas do núcleo de programação até aquele momento 85 (oitenta e cinco). Já atuava profissionalmente como programador. Curioso em relação a robótica.

Aluno 11:

Desinteressado em relação a programação, que via como "uma coisa muito parada e chata de se fazer". Nota média nas disciplinas do núcleo de programação até aquele momento 77 (setenta e sete pontos). Quase nenhum conhecimento de robótica. Reprovado em TP3, CA e IA.

Aluno 12

Pouco motivado. Já havia visto muitas disciplinas "só para software". Reprovado em TP2 e TP3. Sem conhecimentos prévios significativos em robótica.

7.2.4 Observação da utilização da Robótica Pedagógica

Foram observadas aproximadamente 80 (oitenta) por cento das aulas e gravadas em formato VHS, aproximadamente 40 (quarenta) horas durante o experimento. A descrição pormenorizada desta observação está no item 7.1 deste trabalho.

7.2.5 Entrevista final com o professor da disciplina

Esta entrevista foi realizada nas instalações da UNA em 06 de julho de 2006 e deve duração aproximada de 25 (vinte e cinco) minutos, e será apresentada uma síntese, buscando relatar os pontos fundamentais. Suas expectativas em relação à turma foram parcialmente frustradas, apesar do interesse demonstrado nas aulas, provavelmente por estarem na eminência de se formarem a pontualidade e a assiduidade ficaram abaixo das expectativas. Provavelmente, este foi um dos fatores que provocou o fato de o projeto "Robô Futebol", não ter sido implementado, explicando porque o ritmo das aulas ter sido mais lento do que o previsto pelo professor.

Quanto a sua proposta de atuação o professor demonstrou satisfação, tendo inclusive explicitado o interesse em colaborar com a produção de uma publicação sobre o trabalho em curso, fato que enriqueceria sua primeira experiência com o magistério em nível superior.

7.2.6 Entrevista final com alguns alunos da disciplina

As entrevistas foram realizadas, com os mesmos alunos, nas instalações da UNA em 17 de julho de 2006, após o termino do semestre letivo, e cada uma teve duração aproximada de 20 (vinte) minutos. Será apresentado um resumo, buscando relatar os pontos mais relevantes. Estas entrevistas foram gravadas em VHS.

Aluno 6:
Gostou da disciplina RP, mas não gostou do fato dela ter sido ofertada somente ao final do curso. Em suas palavras "Onde aprende a programar (TP1, TP2 e TP3, é onde não vê a capacidade prática da programação. A RP foi a única disciplina onde nós vimos a prática aplicada à programação computadores". Depois da RP, desenvolveu mais interesse pela programação, mas achava que já era tarde, pois estava formado. Em relação ao projeto segue linha, disse : "Era um evento no qual nunca me vi, por ser totalmente incapaz. Depois da disciplina dá para perceber que é uma coisa totalmente tranqüila, para quem fez o curso a sério".

Entende que a RP deveria ser oferecida aos alunos desde o começo do curso, pois dá uma "idéia global", ao passo que fez uma calculadora em TP2 e um programa de tratamento de imagens em CA, e não conseguiu estabelecer relação entre as duas. Achou a dinâmica das aulas muito diferente, "onde os alunos não se sentiam subordinados ao professores, mas era como se estivessem em uma reunião". As aulas não eram cansativas, a ponto de vir em um sábado, porque a idéia da competição era muito interessante e não reprimia o "clima de ajuda" entre os alunos.

Aluno 7:

Entendeu que a matéria desenvolvia bem o raciocínio, sendo um “diferencial do curso”, por envolver a parte de programação com a “máquina mesmo”. Descreveu as aulas como “descontraídas”. O professor deve ser muito “bem preparado, com conhecimentos em física e lógica de programação”. A competição estimula os alunos a buscar “um melhor projeto, robô, programa e performance”. Entendeu como negativo o fato de ser só um semestre, em virtude da “amplitude” do conteúdo. “Aprende o básico, se envolve e a matéria termina”.

Aluno 10:

Percebeu que a robótica “não é bicho de sete cabeças”. A disciplina “interagiu muito a turma”, fomentando o trabalho em equipe que “o mercado exige”, pois “pois cada membro tinha uma função dentro do grupo”. Entende que a RP no início do curso “abre a cabeça dos alunos”. Quanto ao papel do professor foi fundamental, já que “não dava nada feito para os alunos”. Acredita que a postura própria, bem como a da turma foi muito positiva, uma vez que no 8º período, já estão “cansados, doidos para formar”. A RP era a única disciplina que tinha vontade de assistir aula, pois era “legal”.

Aluno 11:

Sua opinião sobre a disciplina foi positiva, pois em seu entendimento, “a maioria das aulas foi prática”. O professor foi diferente de todos os outros, orientando a aplicação dos outros conteúdos. Opinou que “não deveria ser só uma disciplina, mas pedaços em todo o curso, já que ver o resultado do programa em um braço de robô que se move é muito interessante”. Relata que teve uma postura diferente em relação as outras disciplinas, nas quais abandonava a sala, algumas vezes, pois ficava cansado de ficar parado na frente de um computador, só pensando. Entretanto, em RP “me peguei depois do horário, tentando fazer o robô funcionar”. Concluiu dizendo “Quando começa a empolgar, embalar, acaba!”.

Aluno 12:

Gostou da RP, pois “é voltada não apenas a programação, mas envolve a montagem”. Entende que o curso tem “muitas disciplinas só pra *software*, sem interação com o meio”. A melhor hora, para a RP, em sua opinião, seria “logo após a programação básica”. Entende que seu conhecimento em programação evoluiu, pois está “pensando bem mais na hora da programação”. Reconheceu que a disciplina fomenta a pesquisa para quem tem interesse.

7.2.7 Competições Externas

Como as competições de robótica são um parâmetro importante, convém notar que esta metodologia propiciou que na 5ª edição da copa de robótica organizadas pela SUCESU-MG durante o INFORUSO na edição de 2005, as equipes do centro universitário UNA, fossem classificadas em 1º e 3º lugar na modalidade "Robo-Pong", competindo com instituições como CEFET-MG, PUC-MG, FUMEC, COTEMIG, entre outras.

Diante do exposto, sob a ótica do pesquisador, alguns ajustes na aplicação da robótica pedagógica deveriam ser adotados, visando atingir resultados mais expressivos. Estas propostas serão apresentadas nas considerações finais.

8 CONSIDERAÇÕES FINAIS

O presente trabalho foi concebido na perspectiva de que a presença das novas tecnologias no cotidiano da sociedade contemporânea vem se tornando lugar comum, e como conseqüência, suas possíveis aplicações devem ser avaliadas de forma sistemática sem encantamento ou preconceitos. Entretanto, é pertinente tomar o cuidado de não sucumbir ao chamado "mito da tecnologia", que afirma que o uso das novas tecnologias no ensino, particularmente o microcomputador, garante melhorias na aprendizagem e no desenvolvimento do aluno OLIVEIRA (1999, p.153). O contexto atual em que as novas modalidades de uso do computador, e por extensão, da robótica, na educação apontam para uma nova direção: o uso desta tecnologia não como 'máquina de ensinar' mas, como uma nova mídia educacional, uma ferramenta de complementação, de mediação, de aperfeiçoamento e de possível mudança na qualidade de ensino. Tais acontecimentos têm sido fomentados pela própria mudança na nossa condição de vida e pelo fato de a natureza do conhecimento ter mudado. VALENTE (1998, p.2).

Tomando como referência a definição de que um campo conceitual é um conjunto de problemas e situações cujo tratamento requer conceitos, procedimentos e representações de tipos diferentes, mas intimamente relacionados VERGNAUD *apud* MOREIRA (2002), a robótica pedagógica (RP) é eminentemente interdisciplinar na medida em que envolve uma diversidade de conhecimentos oriundos de diversas disciplinas.

A combinação de renomados pesquisadores das chamadas ciências "moles" como, por exemplo, Piaget, Freire e Castells, com destacados autores das chamadas ciências "duras" como, por exemplo, Sebesta, Deitel e Groover, além de clássicos do ensino de programação de computadores como Guimarães e Lages, Kernighan e Ritchie e Mizrahi, proporcionou, metaforicamente, um caldo primordial de onde emergiu este trabalho, temperado com autores nacionais e contemporâneos, como Baranauskas, D'abreu, Oliveira e finalmente Costa. O fio condutor que permitiu o enlace destes diversos teóricos foi o construcionismo de Papert.

Outro importante aporte teórico foram os artigos recentes publicados em conceituados órgãos nacionais como a SBC e o NIED (UNICAMP) e internacionais, como por exemplo, IEEE, ACM e MIT, entre outras, sobre a robótica pedagógica, o que destaca a relevância do estudo. Diversos artigos que investigam sua aplicação em escolas de nível fundamental, médio e superior, atestam suas múltiplas possibilidades de aplicação.

A hipótese investigada de que a introdução, no ambiente da aula, de um robô, que permita ao aluno observar o comportamento dos programas de computador desenvolvidos por ele e executados pelo robô e as interações deste com o ambiente em tempo real, pudesse favorecer outras formas de percepção e de estímulo, contribuindo efetivamente para a aprendizagem de técnicas de programação de computadores demonstrou-se como potencialmente verdadeira, desde que inserida em uma metodologia mais abrangente e acoplada ao curso em geral e ao núcleo de programação em particular.

As questões básicas de pesquisa, a saber, "Quais os efeitos da introdução da robótica pedagógica em disciplinas de programação de computadores?", "Quais as abordagens metodológicas desejáveis na aplicação da robótica pedagógica?" e "Qual o papel do professor neste processo?", foram investigadas através de um extenso processo de pesquisa seguida por uma profunda reflexão, alicerçada em um multidisciplinar referencial teórico.

O objetivo deste trabalho, que visava contribuir para a melhoria na qualidade do processo ensino-aprendizagem de conceitos relacionados à programação de computadores, foi contemplado através das propostas apresentadas pelo pesquisador. Importante salientar que em virtude do campo de pesquisa delimitado, este conjunto de propostas visa primariamente o curso de Sistemas de Informações do Centro Universitário UNA. Entretanto, não existem impedimentos teóricos que inviabilizem sua eventual adaptação e aplicação em outros cursos de outras instituições.

8.1 Propostas do pesquisador

Em uma combinação da teoria dos campos conceituais em que VERGNAUD *apud* MOREIRA (2002) afirma que seu domínio, por parte do sujeito, ocorre ao longo de um largo período de tempo, através de experiência, maturidade e aprendizagem decorrente de novos problemas e novas propriedades que devem ser estudados ao longo de vários anos se a meta for que os alunos progressivamente os dominem e o construcionismo que se preocupa em criar ambientes de aprendizagem que possam produzir mudanças no intelecto, a proposta deste trabalho sugere que a robótica pedagógica seja adotada desde o primeiro semestre do curso, não como uma disciplina isolada, mas através de oficinas aplicadas nas disciplinas do núcleo de programação, sem perder de vista a afirmação de PAPERT (1986) de que é preciso desenvolver a idéia de que o trabalho, e neste caso, o estudo deve ser o brinquedo dos adultos.

A montagem de robôs e sua programação através do *Robolab* deveriam ser usadas na disciplina Teoria da Computação (TC), servindo inclusive como apoio para a discussão de questões básicas de *hardware, software* e comunicação de dados, que compõem a ementa da disciplina. Para alunos de primeiro período, este recurso poderia servir como uma forma de estímulo, tornando o contato inicial com a informática e com a programação mais atrativa. Atividades que envolvam a disciplina Introdução a Lógica (IL) poderiam ser elaboradas e aplicadas, permitindo aos alunos apropriarem-se das vantagens da metodologia proposta desde o início do curso.

Poderia ser estabelecida uma parceria com o departamento de matemática, de forma que fossem elaboradas atividades que privilegiassem, ainda mais, a interdisciplinaridade, buscando minimizar dificuldades históricas nesta área de conhecimento. Pode-se inclusive pensar em diminuição dos índices de evasão, em virtude de uma inserção mais prazerosa dos alunos calouros no ambiente universitário.

Na disciplina Técnicas de Programação I (TP1) poderia ser adotada a linguagem NQC, não em substituição ao Pascal e ao C, mas como um complemento através de dinâmicas que ajudassem aos alunos a construir conhecimentos referentes a variáveis, comandos condicionais (via sensores) e laços de repetição, bem como a questão da modularização dos programas através de funções.

Algoritmos de ordenação de vetores, como o tradicional método da bolha, poderiam ser demonstrados de forma lúdica e os tipos abstratos de dados (*ADT*) como listas encadeadas, poderiam ser simuladas, através da comunicação entre vários RCX[50], trabalhando de forma ordenada, na disciplina Técnicas de Programação 2 (TP2).

Em Técnicas de Programação 3 (TP3), a programação orientada por objetos, com o emprego do Java, poderia ser contemplada através do leJOS, enquanto a disciplina de Inteligência Artificial (IA) poderia utilizar as linguagens Legolog (Prolog) e XS (Lisp) para trabalhar a pesquisa da inteligência artificial em robôs construídos com o kit LEGO MINDSTORMS.

Já na disciplina de Computação Avançada (CA) o recurso de multitarefa do RCX, poderia ser explorado, para criar situações em que os alunos vivenciariam simulações estimulantes. Finalmente a Engenharia de Software poderia promover projetos de automação de robôs com comportamentos mais sofisticados, que demandariam projetos de software mais complexos.

---

[50] Veja o item 5.2.6 PROGRAMAÇÃO DO RCX, deste trabalho.

Quanto à disciplina Tópicos Especiais em Informática II (RP), deveriam ser retirados da ementa questões não relacionadas diretamente a robótica, como por exemplo, realidade virtual, processamento digital, *personal software process* (PSP) e linguagens multimídia para Internet, dando prioridade a preparação para a participação em competições internas e externas de RP.

De forma complementar a adoção desta estratégia de oficinas, sugere-se a criação de um núcleo de robótica pedagógica (NRP) dotado de um laboratório com a infra-estrutura apropriada e professores pesquisadores com horas de dedicação visando a elaboração de atividades que possam ser conduzidas pelos professores específicos das disciplinas de programação. Estes professores pesquisadores atuariam como difusores da robótica, apoiando e colaborando com os demais professores. Para a montagem do laboratório, sugere-se a aquisição de novos kits, com a tecnologia NXT, ilustrado na Figura 29, visando explorar recursos mais avançados. Também deveriam ser adquiridos kits de robótica de outros fabricantes, para expandir as possibilidades de comparações. Outro importante aporte seria a aquisição de títulos específicos e atuais sobre o assunto. O NRP deveria ser integrado aos departamentos de matemática e engenharia.

Uma das principais atribuições do NRP deveria ser a organização de competições internas e o incentivo à participação de alunos e professores, em eventos externos ligados a RP (simpósios, *workshops* e competições), através da integração com o programa de iniciação cientifica e tecnológica, fomentando, através da concessão de bolsas, a produção acadêmica com a publicação de artigos e a condução de projetos de pesquisa em trabalhos de conclusão de curso em graduação e pós-graduação *lato* e *stricto sensu.*

Diante do exposto, pode-se imaginar um processo de aprendizagem mais agradável e eficaz, onde ao contrário do cenário atual, em que a programação de computadores é considerada por muitos alunos como penosa e enfadonha, poderia tornar-se mais estimulante, produzindo resultados mais ricos e duradouros.

A proposta sugerida torna a robótica um fio condutor que acopla as disciplinas do núcleo de programação, proporcionando ao aluno uma visão sistêmica do processo de programação de computadores.

REFERÊNCIAS

ALVES-MAZZOTI, A J. **O método nas ciências naturais e sociais**: pesquisa quantitativa e qualitativa. São Paulo: Pioneira, 1999.

ASCENCIO A. F. G; CAMPOS E. A. V. **Fundamentos da programação de computadores**. São Paulo: Prentice Hall, 2002.

BARANAUSKAS M. C. C. **Procedimento, função, objeto ou lógica?** Linguagens de programação vistas pelos seus paradigmas. In VALENTE, José Armando. (ORG.) Computadores e conhecimento: repensando a educação. Campinas: UNICAMP/NIED, 1998. p.45-63.

BASTOS, J. O. A. F. **Inserção do software livre no curso de informática industrial – CET-Itabirito/CEFET-MG, através da disciplina Análise de sistemas**. Dissertação de mestrado – CEFET-MG, 2006

BAUM, D.; HANSEN, J. **NQC Programmer's Guide**. Versão 3.1 r4. Disponível em http://bricxcc.sourceforge.net/nqc. Acessado em 15/06/2006.

BEER, R. ; CHIEL, H. **Using autonomous robotics to teach science and engineering**. Commun. ACM 42, 6 (June 1999), 85-92.

BIGGE, M. L. **Teorias da Aprendizagem para Professores**. E.P.U., 1977.

BOGDAN, R. C. e BIKLEN, S. K. **Investigação Qualitativa em Educação**: uma introdução à teoria e aos métodos.Trad. Maria João Alvarez e Sara Bahia dos Santos. Portugal: Porto Editora. 1994.

BORGES, M. A. F. **Avaliação de uma Metodologia Alternativa para a Aprendizagem de Programação**. VIII Workshop de Educação em Computação – WEI 2000. Curitiba, PR, Brasil.

CAPELLI, Alexandre. **Mecatrônica Industrial**. São Paulo: Editora Saber Ltda, 2002.

CAPRON H.L.; JOHNSON J.A. **Introdução à informática**. Trad. Santos, J. C. B. 8 ed. São Paulo: Peason Prentice Hall, 2004.

CASTELLS, Manuel. **A sociedade em rede**; tradução: Roneide Venâncio Majer.
(A era da informação: economia, sociedade e cultura; V.1) 3 ed. São Paulo: Paz e Terra, 1999.

CERUZZI P.E. **A History of Modern Computing**. 2nd ed. MIT PRESS: LONDON, 2003.

CERVO, A.L. e BERVIAN, P.A. **Metodologia Científica**: para uso dos estudantes universitários. 11ª Edição. São Paulo: McGraw Hill do Brasil, 2000

CHAVES DE CASTRO, T., CASTRO JÚNIOR, A., MENEZES, C., BOERES, M. e RAUBER, M. **Utilizando Programação Funcional em Disciplinas Introdutórias de Computação**. XI Workshop de Educação em Computação – WEI 2003. Campinas, SP, Brasil, 2003.

COLL C., MARCHESI A. & PALACIOS J. **Desenvolvimento psicológico e educação**: psicologia da educação escolar. Trad. Fátima Murad. 2 ed. Porto Alegre: Artmed, 2004.

COMPANHIA DOS NUMEROS. **Curiosidades Matemáticas**. Disponível em http://www.ciadosnumeros.com.br/curiosidades/cur006.asp. Acessado em 10/04/2006.

CONVITEAFISICA. **Física dos brinquedos**. Disponível em http://www.conviteafisica.com.br/home_fisica/fisica_dos_brinquedos/piao.htm#torque. Acessado em 16 de junho de 2006.

CORNACHIONE Jr. E. B. **Informática aplicada às áreas de contabilidade, administração e economia**. 3 ed. São Paulo: Atlas, 2001.

COSTA, José Wilson. **Informação, aprendizagem, conhecimento**; A tríplice aliança no contexto da educação a distância. Belo Horizonte: UFMG/ECI, 2002. 185 p.
Tese (Doutorado).

COSTA, J. W.; OLIVEIRA, Maria A. Monteiro (orgs.). **Novas linguagens e novas tecnologias**: Educação e sociabilidade. Petrópolis,RJ: Vozes, 2004.

COUSINEAU, Guy and MAUNY, M. **The Functional Approach to Programming**. Cambridge, UK: Cambridge University Press, 1998

COUTINHO, M.T.C.; MOREIRA M. **Psicologia da Educação**: um estudo dos processos psicológicos de desenvolvimento e aprendizagem humanos, voltados à educação: ênfase na abordagem construtivista. Belo Horizonte, Editora Lê, 1992.

D'ABREU, V. V. J. **Integração de dispositivos mecatrônicos para ensino/aprendizagem de conceitos na área de automação**. 2002. Tese (Doutorado) – Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas.

D´ABREU, J. V. V. **Disseminação da robótica pedagógica em diferentes níveis de ensino**. Revista Educativa, Nova Odessa, v.1, n.1, p-11-16, dez. 2004 – issn: 1808-5954. Disponível em http://201.28.104.78:8080/ojsped/viewarticle.php?id=2&layout=abstract. Acessado em 13/04/2006.

DEEPAK K. and MEEDEN L.: **Robots in the Undergraduate Curriculum**. Proceedings of the Third Annual CCSC (Consortium for Computing in Small Colleges) Northeastern Conference, The Journal of Computing in Small Colleges, John G. Meinke (editor), Volume 13, Number 5, May 1998.

DEITEL, H. M.; DEITEL, P. J. **JAVA, como programar**. Trad. Lisboa, C. A. L. 4 ed. Porto Alegre: Bookman, 2003.

DEITEL, H. M.; DEITEL, P. J. **JAVA how to program.** 6 ed. NY: Peason Prentice Hall, 2005.

DELGADO, C., XEXEO, J. A. M., SOUZA, I. F., CAMPOS, M., RAPKIEWICZ, C. E. **Uma Abordagem Pedagógica para a Iniciação ao Estudo de Algoritmos**. XII Workshop de Educação em Computação (WEI'2004). Salvador, BA, Brasil, 2004.

EDACOM. **Modelo de Educação Tecnológica LEGO ZOOM**. Disponível em http://www.edacom.com.br/site/Carregar.aspx?Secao=Tecnologia. Acessado em 8/6/2006.

FAGIN, B. **Ada/Mindstorms 3.0 a computational environment for introductory robotics and programming**, IEEE Robotics and Automation. Special issue on robotics and education, 2003.

FAGIN, B. and MERKLE, L. **Measuring the Effectiveness of Robots in Teaching Computer Science**, ACM Journal of Educational Resources in Computing, June 2002.

FARRER H. et al. **Algoritmos Estruturados**. 2 ed. Rio de Janeiro: LTC, 1989.

FERRETTI et al. **Novas Tecnologias, Trabalho e Educação**: Um debate Multidisciplinar. Petrópolis, RJ: Vozes, 1994.

FLOWERS, T. AND GOSSETT, K. **Teaching problem solving, computing, and information technology with robots**. Disponível em http://www.dean.usma.edu/dean/ComputingAtWestPoint/Robots.htm. Acessado em 10 de julho de 2006.

FREIRE, Paulo. **Pedagogia da autonomia**: saberes necessários à prática educativa. São Paulo: Paz e Terra, 1996.

GIL, A. C. **Métodos e técnicas de pesquisa social**. 4. ed. São Paulo: Atlas, 2002.

GOODE, W. J.; HATT, P. K. - **Métodos em Pesquisa Social**. 3 ed., São Paulo: Cia Editora Nacional, 1969.

GROOVER, Mikell P.; WEISS, Mitchell; NAGEL, Roger N.; ODREY, Nichola G.
Robótica Tecnologia e Programação. São Paulo: McGraw-Hill, 1988.

GUBA, E. G. & LINCOLN, Y. S. **Competing paradigms in qualitative research**. In N. K. Denzin & Y. S. Lincoln (Orgs.), Handbook of qualitative research (pp. 105-117). Califórnia: Sage. 1994.

GUIMARAES, A. M.; LAGES, N. A. C. **Algoritmos e estruturas de dados**. Rio de Janeiro: LTC, 1985.

HENDERSON, P. **Modern Introductory Computer Science**. In Proceedings of the eighteenth SIGCSE technical symposium on Computer science education, ACM Press, pp. 183-190. 1987.

IEEE-CS and ACM. **On computing curricula IEEE computer society and association for computting machinery (ACM)**, T. J. T. F. Computing Curricula Computer Science. 2001.

INOVACAOTECNOLOGICA. **O Ano Dos Robôs**. Disponível em: http://www.inovacaotecnologica.com.br/noticias/010180020322.html. Acessado em 05 jun. 2006.

JAPIASSU, H.F. **Epistemologia**: O mito da neutralidade científica. Rio: Imago, 1975.

JOHANSON. R. P. **Computers, Cognition and Curriculum**: Retrospect and Prospect. J. Educational Computing Research, 4(1),1-30. 1988

JOHNSON-LAIRD. P.N. **Mental models**. Cambridge: Harvard University Press. 1983.

JOLIVET, R. **Curso de Filosofia.** 13ª Edição. Rio de Janeiro: Agir. 1979.

KERNIGHAN B.W. ; RITCHIE D. M. **C a linguagem de programação**. 7 ed. Rio de Janeiro: Campus, 1986.

KLASSNER, F. **A case study of LEGO Mindstorms**. suitability for artificial intelligence and robotics courses at the college level. In Proceedings of the 33rd SIGCSE Technical Symposium on Computer. Science Education (Feb. 2002), 8-12. 2002.

KOLIVER, C., DORNELES, R. V., CASA, M. E. **Das (muitas) dúvidas e (poucas) certezas do ensino de algoritmos**. XII Workshop de Educação em Computação (WEI'2004). Salvador, BA, Brasil. 2004.

KNUDSEN B.J. **The Unofficial Guide to LEGO MINDSTORMS Robots**. 1 ed. O'REILLY, 1999.

KUMAR D. ; MEEDEN Lisa. **Robots in the Undergraduate Curriculum**. Proceedings of the Third Annual CCSC (Consortium for Computing in Small Colleges) Northeastern Conference, The Journal of Computing in Small Colleges, John G. Meinke (editor), Volume 13, Number 5, May 1998.

LAVERDE D. **Programming Lego Mindstorms with Java**. Paperback 2006.

LAKATOS, E.M. & MARCONI, M. A. **Fundamentos de metodologia cientifica**. 3 ed., São Paulo: Atlas, 1991.

LEGO. **LEGO MINDSTORMS for Schools**. Disponível em http://www.lego.com/eng/education/mindstorms/default.asp. Acessado em 20 de julho de 2006.

LUCKESI, C. C. e PASSOS, E.S. **Introdução à filosofia**: aprendendo a pensar. São Paulo: Cortez, 1996.

LUND, H. H. AND PAGLIARINI, L**. Robot soccer with Lego Mindstorms**. In RoboCup-98: Robot Soccer World Cup II. Lecture Notes in Artificial Intelligence (LNAI), vol. 1604. Springer Verlag, New York. 1998

MARTINS, A. **O que é Robótica**. São Paulo: Brasiliense, 1993.

METOYER, Ronald A., XU, Lanyue, SRINIVASAN, Madhusudhanan. **A Tangible Interface for High-Level Direction of Multiple Animated Characters**. Graphics Interface, School of Electrical Engineering and Computer Science. Oregon State University. 2003.

MIZRAHI V. V. **Treinamento em linguagem C**. São Paulo: Makron Books, 1990.

MIZRAHI V. V. **Treinamento em linguagem C ++**. 2 ed. São Paulo: Makron Books, 2006.

MINDELL, David et al. **LEGO Mindstorms, The Structure of an Engineering (R)evolution**. 2000.

MOREIRA, M.A. **A teoria dos campos conceituais de Vergnaud, o ensino de ciências e a pesquisa nesta área**. Investigações em Ensino de Ciências, v.7, n.1, 2002. Acessado em 14/06/2006. Disponível em http://www.if.ufrgs.br/public/ensino/vol7/n1/v7_n1_a1.html

NÉRICI, I.G. **Introdução à lógica**. 5ª Edição. São Paulo: Nobel, 1978.

NOBRE, I. A. M. N., MENEZES, C. S. **Suporte à Cooperação em um Ambiente de aprendizagem para Programação (SAmbA)**. XIII Simpósio Brasileiro de Informática na Educação – SBIE 2002. São Leopoldo, RS, Brasil. 2002.

O'BRIEN J.A. **Sistemas de informação e as decisões gerenciais na era da Internet**. Trad. Moreira C. K. e Moreira C. K. 2 ed.São Paulo: Saraiva, 2004.

OLIVEIRA, M.R.N.S. **Tecnologias interativas e educação**. Educação em debate, Fortaleza, n.37, p.150-156. 1999.

OLIVEIRA, J. A. C. **Robótica e educação**: aproximações piagetianas numa tese de doutorado. XI Seminário Internacional de Educação Tecnológica. Novo Hamburgo-RS. 2004.

OLIVEIRA, C. C; COSTA J. W.; MOREIRA M. **Ambientes informatizados de aprendizagem**: produção e avaliação de software educativo. Campinas, Papirus, 2001.

PAPERT, S. **Mindstorms Children, Computers, and Powerful Ideas**. Basic Books, New York. 1980. Traduzido para o Português como LOGO: Computadores e Educação, 3 ed. São Paulo: Brasiliense, 1986. Trad. VALENTE, J. A. et al.

PAPERT, S. **New images of programming**: in search of an educationally powerful concept of technological fluency. (A proposal to the National Science Fundation NSF), Cambridge: MIT Technology laboratory. 1991.

PAPERT, S. **A máquina das crianças**: Repensando a Escola na Era da Informática. Porto Alegre, Artes Médicas, 1994.

RAPKIEWICZ, C. E.; PEREIRA J. C. R. Júnior. **O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil.** WEI RJ/ES I : 2004 nov. 19-21 : Vitória - ES, Rio das Ostras - RJ

ROCHA, R. **Programação orientada a objetos**: uma abordagem lúdica. Revista Educação & Tecnologia, CEFET-MG, Belo Horizonte, v. 8, n. 1, p. 45-49, 2003.

ROJAS, R; HASHAGEN U. **The First Computers**: History and Architectures.
MIT PRESS: LONDON, 2000.

SALANT, Michael A. **Introdução à Robótica**. São Paulo: McGraw-Hill, 1990.

SANTOS, Rodrigo P. e COSTA, Heitor A. X.. **Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática**. p.41-50. INFOCOMP Journal of Computer Science – v.5, n.1 (2006) – Lavras: Universidade Federal de Lavras.

SEBESTA, R. W. **Conceitos de linguagens de programação**. Trad. José Carlos Barbosa dos Santos. 5 ed. Porto Alegre: Bookman, 2003.

STALLINGS, W. **Arquitetura e organização de computadores**: projeto para o desempenho. Trad. Figueiredo C.C. e Figueiredo L.C. 5 ed. São Paulo: Prentice Hall, 2002.

STAHL, Marimar M. **Ambientes de ensino-aprendizagem computadorizados**: da sala de aula convencional ao mundo da fantasia. Rio de Janeiro: COPPE-UFRJ, 1991.

UNECE, United Nations Economic Commission for Europe. **World Robotics Survey**, 2004. Disponível em http://www.unece.org/press/pr2004/04robots_index.htm. Acessado em 05 jun. 2006.

UNIVERSITY OF ST ANDREWS. **The MacTutor History of Mathematics archive.** Disponível em http://www-history.mcs.st-andrews.ac.uk. Acessado em 10/04/2006.

VALENTE, José Armando. **Computadores e conhecimento**: repensando a educação. 2 ed. Campinas: UNICAMP/NIED, 1998.

VIEIRA, Darli Rodrigues. **Funções da Robótica no processo de acumulação Brasileiro**. Petrópolis: Vozes, 1995.

WATT, David, **Programming Language Concepts and Paradigm**s, C.A.R. Hoare series editor, Prentice Hall International Series in Computer Science, 1990.

WIKIPÉDIA. **Runaround**. Disponível em http://en.wikipedia.org/wiki/Runaround. Acessado em 13 de julho de 2006.

WOLZ, U. **Teaching design and project management with lego RCX robots**. In Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education (Charlotte, NC, Feb. 2001), 95-99. 2001.

YIN, Robert K. - **Case Study Research - Design and Methods**. Sage Publications Inc., USA, 1989.

ANEXO

Grade curricular do curso de Sistemas de Informação do Centro Universitário UNA

Período	Disciplina	Horas
1	Comunicação e Expressão	72
1	**Introdução à Lógica**	**72**
1	Matemática Básica	72
1	**Teoria da Computação**	**72**
1	Teoria Econômica	72
2	Álgebra	72
2	Estatística	72
2	Inglês	72
2	**Técnicas de Programação I**	**72**
2	Teorias da Administração	72
3	Arquitetura de Computadores	72
3	Cálculo Diferencial e Integral	72
3	**Técnicas de Programação II**	**72**
3	Contabilidade	72
3	Organização, Sistemas e Métodos	72
4	Administração da Produção e Operações	72
4	Administração de Recursos Materiais e Patrimoniais	72
4	Matemática Financeira	72
4	Redes e Telecomunicações I	72
4	**Técnicas de Programação III**	**72**
5	Administração Financeira e Orçamentária	72
5	Análise e Desenvolvimento de Sistemas I	72
5	Arquitetura de Sistemas de Informação I	72
5	Computador e Sociedade	72
5	Redes e Telecomunicações II	72
6	Análise e Desenvolvimento de Sistemas II	72
6	Arquitetura de Sistemas de Informação II	72
6	**Computação Avançada**	**72**
6	**Inteligência Artificial**	**72**
6	Pesquisa Operacional	72
7	Automação de Negócios	72
7	**Engenharia de Software**	**72**
7	Gerência de Recursos de Informática	72
7	Instituições de Direito Público e Privado	72
7	Tópicos Especiais em Informática I	72
8	Auditoria de Sistemas	72
8	Controle e Avaliação de Sistemas	72
8	Estágio Supervisionado	36
8	Planejamento Estratégico	72
8	Sistemas Cooperativos	36
8	**Tópicos Especiais em Informática II**	72