



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ  
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS  
LICENCIATURA EM COMPUTAÇÃO**

**ROMERITO CAMPOS DE ANDRADE**

**MOBILOGO: implementação da linguagem Logo para  
dispositivos móveis**

Patos-PB,  
2010.

**ROMERITO CAMPOS DE ANDRADE**

**MOBILOGO: implementação da linguagem Logo para dispositivos móveis**

Monografia apresentada ao Curso de Licenciatura Plena em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do título de graduado.

**Orientador: Prof. Msc. Wellington Candeia de Araújo.**  
**Co-Orientador: Prof. Msc. Frederico Moreira Bublitz.**

Patos-PB,  
2010

R553m Andrade, Romerito Campos

Mobilogo: Implementação da Linguagem Logo para Dispositivos Móveis. Patos: UEPB, 2010. 58f.

Monografia (TRABALHO Acadêmico Orientado – (TAO) - Universidade Estadual da Paraíba.  
Orientador: prof.Msc. Wellington Candeia de Araújo.

1. Computação 2. Computação Móvel I. Título  
II. Araujo, Wellington Candeia

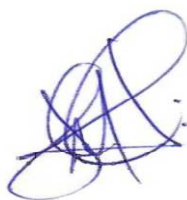
CDD 629.895

**ROMERITO CAMPOS DE ANDRADE**

## **MOBILOGO: implementação da linguagem Logo para dispositivos móveis**

Monografia apresentada ao Curso de Licenciatura Plena em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do título de graduado.

Aprovado em 10/12/2010.



---

Prof. Msc. Frederico Moreira Bublitz/UEPB

Co-Orientador



---

Prof. Msc. Wendell Campos Veras/UEPB

Examinador



---

Prof. Msc. Vítor A. S. D. Afonso/UEPB

Examinador

Em Memória de Jovelina Mendes de Araújo,  
Lembranças eternas.

## **AGRADECIMENTOS**

O mais importante de tudo é saber que muitas pessoas contribuíram diretamente para este trabalho. Em todos os momentos durante este percurso pude contar com o apoio, orientação, críticas, “puxões de orelha”, força, incentivo e muito mais. Por isto, deixo meus agradecimentos à:

Deus, pela saúde e vida;

Aos meus pais, José Fernando do N. Andrade e Cleonice Ferreira C. de Andrade pelo apoio incondicional em toda minha vida;

Aos orientadores Fred M. Bublitz e Wellington C. de Araújo que são exemplos que segui durante o curso;

Márcia R. de Lima pela paciência, e pelo apoio em todas as batalhas;

Aos grandes amigos Makarrão, Renata, Olegário e Peu pelo apoio, amizade e leadade.

“A riqueza é algo comum, mas o saber é raro” (Dan Bronw, 2009).

## RESUMO

Este trabalho apresenta a versão 1.0 do Mobilogo, artefato de *software* para dispositivos móveis, cujo objetivo é implementar a linguagem de programação Logo, sendo o mesmo desenvolvido durante o projeto de extensão da universidade estadual da Paraíba intitulado *Mobilogo: uma proposta de software educativo para telefones celulares*. Além disso, apresenta-se também a Computação Móvel que tem se constituído como um importante paradigma computacional tendo como principal característica a mobilidade dos dispositivos o que permite aos seus usuários maior comodidade, praticidade e flexibilidade na realização das tarefas. Como consequência do potencial dos dispositivos computacionais móveis surgiu o *Mobile Learning* que é uma nova área da ciência preocupada em estudar todos os aspectos relacionados a aprendizagem com dispositivos móveis, neste trabalho dois pontos importantes do *Mobile Learning* são discutidos: uma definição da natureza exata do *Mobile Learning*, tendo em vista que seu sucesso fez com que várias definições surgissem atenuando sua definição exata; avaliação em *Mobile Learning* que trata de questões relacionadas às forma de avaliação em aprendizagem apoiada por dispositivos móveis. Juntos, computação móvel e *Mobile Learning* formam as bases teóricas para o Mobilogo, tendo em vista que a aplicação visa aprendizagem através de dispositivos móveis com a utilização da linguagem Logo. O *software* está em sua primeira versão com implementação dos comandos primitivos de movimento: para frente, para trás, para direita e para esquerda.

**Palavras-Chave:** Computação Móvel, Mobile Learning, Logo, software educativo.



## ABSTRACT

This paper presents version 1.0 of Mobilogo artifact of software for mobile devices, whose goal is to implement the Logo Programming Language, being further developed during the project to extend the State University of Paraíba named Mobilogo: a proposal of educational software for cell phones. Moreover, it is also present in Mobile Computing that has been constituted as an important computing paradigm, whose main characteristic the mobility of devices allowing its users more convenience and flexibility in performing the tasks. As a result of the potencial of mobile computing devices has emerged that Mobile Learning is a new of science concerned with studying all aspects of learning with mobile devices, in this work two important points are discussed in Mobile Learning: a definition of the exact nature of Mobile Learning, considering that its success caused many definitions reducing its exact definition; Mobile Learning assessment that address issues related to the rating form on learning supported by mobile devices. Together, Mobile Learning and Mobile Computing form the theoretical basis of Mobilogo, considering that the application seeks to learning through mobile devices with th use of Logo. The software is in its first release with implementation of the primitive motion commands: forward, backward, right and left.

**KeyWords:** Mobile Computing, Mobile Learning, Logo, Educational Software.

## Índice de ilustrações

Figura 1: Plataformas Java SE, EE e ME (2010).....	23
Figura 2: Interpretador SuperLogo 3.0.....	36
Figura 3: Procedimento "Quadrado" .....	37
Figura 4: Mobilogo - Casos de Uso.....	41
Figura 5: Diagrama de Classes da Aplicação.....	46
Figura 6: Transformação da Tartaruga.....	48
Figura 7: Menu da aplicação.....	50
Figura 8: Ambiente Logo.....	51
Figura 9: Ajuda - Sintaxe.....	51
Figura 10: Ajuda - Descrição.....	51

## **Lista de Abreviaturas e Siglas**

ADT – Android Development Tools  
API – Application Programming Interface  
CDC – Connected Device Configuration  
CDMA – Code Division Multiplexing Access  
CLDC – Connected Limited Device Configuration  
CPU – Central Processing Unit  
GCF – Generic Connection Framework  
IDE – Integrated Development Environment  
JEE – Java Enterprise Edition  
JME – Java Mobile Edition  
JSE – Java Standart Edition  
JSR – Java Specificatio Request  
KVM – Kilobyte Virtual Machine  
MIDP – Mobile Information Device Perfil  
PAD – Personal Digital Assistent  
PROEAC – Pró-Reitoria de extensão e assuntos comunitários  
RAM – Random Access Memory  
ROM – Read Only Memory  
SDK – Software Development Kit  
TCP/IP – Trasmision Control Protocol/Internet Protocol  
UDP - User Datagrama Protocol  
UEPB – Universidade Estadual da Paraíba  
VHE – Virtual Home Enironment

# Sumário

1	INTRODUÇÃO.....	13
2	COMPUTAÇÃO MÓVEL.....	15
2.1	Característica da Computação Móvel.....	16
2.2	Funções da Computação Móvel.....	18
2.3	Desenvolvimento de aplicações móveis.....	19
2.3.1	Sistemas operacionais móveis.....	20
	Android.....	20
	Symbian.....	21
	Bada.....	21
	Iphone.....	22
	Windows Phone 7.....	22
2.3.2	Plataforma JME.....	22
	Configurações e Perfis.....	24
3	MOBILE LEARNING.....	26
3.1	O que é Mobile Learning.....	27
3.1.1	Perspectivas em Torno do Mobile Learning.....	28
3.1.2	Características do Mobile Learning.....	29
3.2	Avaliação em Mobile Learning.....	30
4	LOGO.....	33
4.1	O que é Logo?.....	34
4.1.1	Características do Logo.....	34
4.1.2	Filosofia do Logo.....	35
4.2	Ambiente Logo.....	36
4.2.1	Característica computacional.....	37
4.2.2	Geometria da Tartaruga.....	38
4.3	Mal entendidos sobre Logo.....	38

5	MOBILOGO.....	40
5.1	Requisitos.....	40
5.1.1	Requisitos Funcionais.....	41
5.1.2	Requisitos não-funcionais.....	42
5.2	Usabilidade.....	42
5.3	Ferramentas Utilizadas.....	43
5.3.1	NetBeans IDE .....	43
5.3.2	SDK Sony Ericsson.....	44
5.3.3	Project Hosting.....	44
5.4	Implementação.....	45
5.4.1	Dispositivo de Referência.....	45
5.4.2	Classes da Aplicação.....	46
	MidletApp.....	46
	Menu.....	47
	ViewWindow.....	47
	Turtle.....	48
	Help.....	49
	Screen.....	49
5.4.3	Telas da aplicação.....	49
	Tela do Menu.....	50
	Tela do Ambiente Logo.....	50
	Telas de Ajuda.....	51
6	Conclusões.....	52
	Referências.....	54

# 1 INTRODUÇÃO

O computador já está inserido em praticamente todas as atividades humanas, desde as mais simples às mais complexas. Quase sempre é possível inserir uma máquina para auxiliar em alguma atividade humana. Estas atividades variam em complexidade, tempo e importância, no entanto o computador geralmente é adaptado e estruturado para servir não só como ferramenta de apoio, mas também ferramenta principal da tarefa.

Neste contexto, o computador também vem sendo explorado, mesmo que timidamente, no âmbito educacional. Estas iniciativas vêm aumentando diante das inúmeras possibilidades que o mesmo pode oferecer para o processo de ensino-aprendizagem. Por exemplo, simulação de eventos que possam comprometer o bem estar dos envolvidos.

No entanto, um novo paradigma computacional vem ganhando força, a computação móvel. Este paradigma tem fomentado as mais variadas experiências, inclusive no contexto educacional tanto do ponto de vista informal quando os usuários podem fazer uso de seus dispositivos para alguma atividade educativa (ESTADÃO, 2007), (PEREIRA, 2006) quanto do ponto de vista formal – *Mobile Learning*.

O *mobile learning* é uma nova área de conhecimento que visa a criação de ambiente de aprendizagem com base na utilização de dispositivos computacionais móveis e os recursos oferecidos pelos mesmos. No entanto, o uso de dispositivos computacionais móveis em si não se configura *Mobile Learning*, pois é necessário que haja planejamento e organização tal como ocorre no ensino convencional.

Diante deste contexto, surgiu o projeto de extensão da Universidade Estadual da Paraíba com o intuito de criar aplicações educativas para dispositivos móveis. O primeiro projeto do programa tem por objetivo implementar a linguagem Logo para dispositivos móveis, visando promover iniciativas com relação à implementação de *Mobile Learning* a exemplo de (VAVOULA et. al. 2009).

Tendo em vista que no cenário nacional as iniciativas com relação ao *Mobile Learning* ainda não são tão evidentes, este projeto pretende explorar esta área de conhecimento no intuito de ampliar a gama de ferramentas utilizadas no processo de ensino-aprendizagem. Para tal, surge o Mobilogo que é uma implementação do Logo para os dispositivos móveis. Neste

trabalho é apresentado a versão 1.0.

O objetivo desta monografia é apresentar a primeira versão do Mobilogo e os temas relacionados ao mesmo. Para isto, fez-se extensa pesquisa bibliográfica acerca dos temas, como também acerca das ferramentas necessárias para concepção da aplicação.

O resultado deste trabalho é apresentado em 6 capítulos, sendo este o primeiro. No segundo capítulo é feita uma menção a computação móvel, nele apresentam-se definições encontradas na literatura, características entre outros detalhes. O terceiro capítulo trata do tema *Mobile Learning* apresentando definições sobre o mesmo, destacando as principais características e linhas de pesquisas. O quarto capítulo traz os principais conceitos acerca da linguagem Logo como a definição do que é Logo, sobre o ambiente e a geometria do logo conceituada por Papert. O quinto capítulo apresenta o projeto Mobilogo, nele são apresentados todos os detalhes do processo de desenvolvimento, desde o levantamento de requisitos até as telas do Mobilogo em execução. Por fim, são apresentadas as conclusões sobre o trabalho, como também sugestões para trabalhos futuros, haja vista que o Mobilogo está na versão 1.0 sendo necessário incremento de várias funcionalidades ainda não disponíveis.

## 2 COMPUTAÇÃO MÓVEL

A evolução dos computadores vem transformando vertiginosamente as relações humanas em todos os aspectos, seja profissionalmente ou pessoalmente. Devido à drástica miniaturização no tamanho dos *microchips* ocorrido nas últimas duas décadas, tornou-se possível fazer uso de dispositivos computacionais portáteis. Estes dispositivos computacionais diferem dos computadores convencionais por terem como característica a mobilidade, sendo esta considerada como um novo paradigma computacional (LOUREIRO, 1998, p.2).

Neste contexto surge a computação móvel que leva em consideração a mobilidade do usuário que carrega o dispositivo. Computação Móvel pode ser “definido como um ambiente computacional sobre a mobilidade física” (TALUKDER; YAVAGAL, 2007, p.7), (AROKIAMARY, 2009, p.12). B'Far enfatiza ainda mais a definição mencionada acrescentado que além desta característica as “capacidades computacionais podem ser usadas enquanto o dispositivo se move” (B'FAR 2005, p.7).

Segundo Talukder e Yavagal (2007, p.7-8) dependendo do contexto diferentes nomes são atribuídos a computação Móvel:

- Computação Móvel: O ambiente de computação move-se junto com seu usuário. Deste modo, o usuário estará apto a acessar informações remotamente mediante conectividade com alguma rede;
- Informação em qualquer hora e lugar: definição genérica de ubíquo, onde a informação está disponível em qualquer hora e lugar;
- Ambiente Virtual de Casa – VHE: operar dispositivos domésticos mesmos através de rede externa em outro ambiente;
- Computação Nômade: ambiente totalmente móvel, tanto usuário quanto dispositivo estão se movendo;
- Computação Pervasiva: o ambiente computacional é pervasivo (entenda-se penetrante) em natureza sendo concebida por qualquer dispositivo em qualquer lugar;
- Computação Ubíqua: algo invisível, tão comum que a presença do dispositivo não será percebida;



- Portabilidade de serviço global: serviço portátil e disponível em qualquer ambiente. Ou seja, qualquer serviço disponível globalmente;
- Computador Vestível: utilizados por seres humanos tal como sapatos, chapéu etc. Precisam ter algumas características adicionais se comparado aos dispositivos móveis padrão. Equipados com atenção e notificação pró-ativa.

Em relação aos termos definidos acima Arokiamary menciona apenas três deles como denominações empregadas para definir a computação móvel de acordo com seu papel: VHE; computação nômade e computação vestível (AROKIAMARY, 2009, p.13).

Ainda sobre os termos utilizados para referenciar a computação móvel Zheng e Li definem Computação nômade como “caso especial – utilizando um dispositivo móvel para conectar a uma rede com fio ou sem intermitente de lugar em lugar, com suporte para mobilidade de alto nível” (ZHENG P.; NI L, 2006, p.3).

Por fim, Satyanarayanan mostra que a computação móvel surgiu do esforço de pesquisadores em construir um sistema distribuído que possuísse clientes móveis. Já a computação pervasiva é uma evolução da computação móvel levando-se em consideração espaços inteligentes, invisibilidade, escalabilidade localizada e condições desiguais dos dispositivos no ambiente (SATYANARAYANAN, 2002).

## ***2.1 Característica da Computação Móvel***

A principal e mais importante característica deste paradigma de computação é a mobilidade, mesmo assim os sistemas de computação móvel podem ser convencionais caso não haja possibilidade de movimento.

Sendo a mobilidade uma característica fundamental para computação móvel tem-se como consequência alguns fatores importantes a serem considerados com relação aos dispositivos móveis. Estes fatores surgem como restrições impostas pela mobilidade à capacidade de armazenamento e CPU dos dispositivos móveis favorecendo a portabilidade dos mesmos, como também qualidade de serviço, alimentação de energia limitada e interface

de usuário.

A primeira característica, localização, esta intimamente ligada ao fato dos dispositivos estarem quase sempre em posições físicas diferentes, ou seja, a mobilidade esta ocorrendo de fato. Neste sentido, é importante utilizar informações relacionadas à localização do dispositivo para fornecer recursos às aplicações. Vale ressaltar que os dispositivos fixos também possuem localização, embora isso não venha a ser considerado uma característica tão fundamental para eles, além disso a localização não muda com o passar do tempo.

Há duas formas de localização: localização; localização sensível. Localização “é a mera habilidade da arquitetura da aplicação móvel acomodar lógica que permite a seleção de diferentes lógicas, nível de fluxo de trabalho e interface baseados na localização” (B'FAR, 2005, p.9), é evidente que esta característica não é exclusiva à aplicações móveis, porém são mais importantes nelas. Por outro lado, localização sensível “é habilidade do dispositivo obter primeiramente a informação de localização enquanto está sendo usado e então levar vantagem da mesma oferecendo recursos e funcionalidades” (B'FAR, 2005, p.9).

A conexão com algum tipo de rede é fundamental para o conceito de computação móvel, embora estar conectado não seja condição exclusiva. Partindo deste pressuposto é importante considerar a qualidade de serviço oferecida. Uma vez que “se a conexão é com ou sem fio, mobilidade significa perda de segurança na conectividade[.]”(B'FAR, 2005, p.10). Além disso, para a conexão sem fio alguns fatores adicionais devem ser mencionados como condições climáticas. Por fim, “o efeito destes fatores é muito profundo em aplicações móveis, pois é importante saber continuar em uma tarefa após se desconectar da rede” (B'FAR, 2005, p.11).

Cada vez mais os recursos de armazenamento e de processamento são exigidos, mesmo em dispositivos de pequeno porte, tais como celulares e PDA's. Os avanços tecnológicos vêm correspondendo as exigências por dispositivos com mais processamento, mais memória, no entanto estes recursos ainda se impõem como limitações a serem contornadas como enfatiza B'Far:

“o tamanho físico dos dispositivos móveis impõe restrições no armazenamento volátil, não-volátil e CPU[...], impõe restrições na forma como desenvolver aplicações móveis[...]. As aplicações devem ser otimizadas para uso do armazenamento de dados e poder de processamento em termos do uso de aplicações

pelo usuário” (B'FAR, 2005, p.11).

A mobilidade tem um alto custo para a autonomia de energia dos dispositivos móveis, já que não seria conveniente caminhar arrastando fios por aí. Por esta razão, os dispositivos móveis usam as baterias como fonte de energia. Este tipo de fonte de energia cria uma nova característica como afirma B'Far:

“A insegurança do uso de baterias ao invés de fontes de energia AC combinada com as restrições de tamanho cria, ainda, outra restrição nomeada alimentação de energia limitada. Esta restrição deve ser balanceada com restrições de processamento, armazenamento e tamanho” (B'FAR, 2005, p.14).

Por fim, tem-se uma característica fundamental para o conceito de computação móvel, interface de usuário. Esta característica define claramente quão interessante pode se tornar uma aplicação móvel, além disso ele permite ao usuário melhor aproveitamento das funcionalidades da aplicação – caso seja bem projetada e implementada. Para aplicações não-móveis o teclado, monitor e o mouse são comprovadamente eficientes, no entanto as limitações encontradas em pequenos dispositivos limitam circunstancialmente o uso dessas interfaces. Portanto, “[...] a maior mudança no paradigma que os projetistas e implementadores devem sofrer é entender a necessidade de encontrar melhores interfaces de usuário para a aplicação[...]” (B'FAR, 2005, p.16).

## ***2.2 Funções da Computação Móvel***

A computação móvel possui várias funções, embora todas elas tenham um ponto em comum, a mobilidade. Neste contexto, alguns autores associam várias funcionalidades da computação móvel com base no aspecto de mobilidade dos dispositivos e o que isto pode proporcionar aos usuários.

Segundo Arokiamary (2009, p.16-17) um ambiente de computação é dito móvel se ele provê suporte as seguintes características:

- Mobilidade do usuário: relacionado à mudança de posição física do usuário enquanto utiliza o dispositivo;

- Mobilidade de portadora: relacionado à mudança de portadora sem interferir no serviço utilizado;
- Mobilidade de *host*: relacionada ao estado do dispositivo que pode ser um host, ou um dispositivo servidor. Embora este último lhe adicione complexidades;
- Mobilidade de serviço: relacionada a mudança de um serviço para outro, sem que o anterior seja finalizado. Deste modo permite ao usuário voltar a tarefa anterior.

Uma outra definição mais abrangente, porém não muito diferente da anterior é encontrada em (TALUKDER, 2007, p.9) onde são definidos mais algumas funcionalidades além das definidas anteriormente, veja-se:

- Mobilidade de rede: relacionada a mudança entre vários tipos de rede utilizando o mesmo serviço;
- Mobilidade de serviço: relacionada a mudança ente vários tipos de dispositivos;
- Mobilidade de sessão: relacionada a mudança de ambiente do usuário sem prejudicar a utilização do serviço. Finalizar uma aplicação em uma conexão CDMA e posteriormente retomar a sessão em seu *Desktop*.

### **2.3 Desenvolvimento de aplicações móveis**

O desenvolvimento de aplicações móveis incorpora várias características que não se encontram em aplicações para *Desktop* ou *Web*. As características encontradas, na verdade, estão mais relacionadas às limitações impostas pelos dispositivos móveis, veja a Seção 2.1.

Estas limitações conduziram ao desenvolvimento e adaptação de linguagens de programação e ferramentas para uso na criação de aplicações móveis. Além disso, vários sistemas operacionais para dispositivos móveis foram criados, tais como BlackBerry, Android e Symbian.

### 2.3.1 Sistemas operacionais móveis

O crescimento do mercado de dispositivos móveis aliado a necessidade de aplicações mais dinâmicas fez surgir um mercado muito lucrativo para a indústria de dispositivos móveis. Este crescimento trouxe uma variedade de sistemas operacionais móveis, cada um com suas características como é mostrado na tabela abaixo:

<i>Sistemas Operacionais</i>	<i>Linguagens</i>	<i>Comentários</i>
Android	Java, C	Código aberto (baseado em linux)
Bada	C, C++	Plataforma da Samsung, roda sobre Linux
Blackberry	Java	Compatível com JME, API nativa fornece melhor integração
MeeGo	Qt, Web Apps, C++	Baseado em Linux
OS X / iPhone	Objective-C, C	Necessário conta <i>Apple Developer</i>
Symbian	C, C++, outras	Código aberto. Alguns suportam JME.
Web OS	HTML, CSS, JavaScript	Baseado em Linux
Windows Mobile	C#, C, Java	Alguns dispositivos já possuem a JVM pré instalada.
Windows Phone	Silverlight, XNA	Com foco nos consumidores

Tabela 1: Sistemas Operacionais

#### **Android**

Criado em 2007 pela Google em parceria com *Open Handset Alliance*. É um sistema operacional para dispositivos móveis baseado em Linux. O desenvolvimento de aplicações tem Java 5 como principal linguagem de programação (VIRKUS et. all, 2010, p.15). Atualmente está na versão 2.2. Para iniciar no desenvolvimento de aplicações Android é necessário fazer a instalação do SDK<sup>1</sup>. O SDK Android é compatível com Windows, Linux e Mac OS X. Existe também um excelente plugin para o Eclipse chamado ADT que permite o desenvolvimento através desta IDE.

<sup>1</sup> O seguinte link mostra como instalar a plataforma de desenvolvimento para aplicações Android <http://developer.android.com/sdk/installing.html>

## **Symbian**

A plataforma Symbian é amplamente utilizada no mundo dos dispositivos móveis. Além disso, a plataforma tem código aberto permitindo maior acesso por parte dos desenvolvedores. A partir de 2009 a plataforma vem sendo mantida pela Fundação Symbian<sup>2</sup>. A versão atual do Symbian é chamada Symbian^3. A linguagem de programação utilizada para criação de aplicações é o Symbian C++ uma adaptação do C++, além disso é possível utilizar o framework multiplataforma Qt. Oficialmente só há plataforma de desenvolvimento para Microsoft Windows (VIRKUS et. all, 2010, p.27-28).

## **Bada**

A plataforma Bada é muito recente no mundo dos sistemas operacionais para dispositivos móveis, foi lançada em 2009 pela Samsung. Para desenvolver aplicações para Bada é necessário se registrar em seu site oficial<sup>3</sup>, o SDK é disponibilizado apenas para Windows e inclui uma IDE baseado no eclipse, um emulador e ferramentas para design da interface com usuário. Por fim, a linguagem de programação utilizada é o C++ com alguns restrições como: não utiliza exceções; sem cast dinâmico (VIRKUS et. all, 2010, p.69-70).

---

2 <http://www.symbian.org>

3 <http://developer.bada.com>

## ***Iphone***

O Iphone utiliza o iOS 4 que é um sistema operacional para ambientes móveis. O iOS 4 é baseado no sistema operacional para *desktop* Mac OS X. Apresenta vários recursos: interface elegante; multitarefa; ampla plataforma de aplicativos entre outros (APPLE, 2010). A principal linguagem utilizada é Objective-C que consiste em C baseado em orientação a objetos, também pode ser utilizado o construtor de interface que usa arquivo no formato proprietário XIB (VIRKUS et. all, 2010, p.22).

## ***Windows Phone 7***

O Windows Phone 7 é o mais recente lançamento da Microsoft para dispositivos móveis. Esta plataforma está focada nos consumidores com recursos atrativos e intuitivos. Para desenvolvimento os programadores podem usar Silverlight, XNA Game Studio ou Visual Studio. A linguagem de programação necessária é o C#. A versão inicial não suporta multitarefa, nem copiar/colar. A API do Windows não será acessada diretamente por questões de segurança (VIRKUS et.all, 2010, p.64-67).

### **2.3.2 Plataforma JME**

Em 1991 a Sun Microsystems fundou um projeto de pesquisa nomeado *Green*. O objetivo deste projeto era inovar nos dispositivos eletrônicos de consumo. O resultado deste trabalho foi uma linguagem de programação baseada em C++, esta linguagem foi chamada, inicialmente, de *Oak*. No entanto, devido uma outra linguagem com este nome já existir ela foi renomeada para Java (DEITEL, 2009, p.9).

As vantagens desta nova linguagem com relação ao C++ é o fato de o programador não manipular o gerenciamento de memória, ponteiros e heranças múltiplas (WELLS, 2007, p.4).

A linguagem Java começou a ganhar destaque com o crescimento da *World Wide Web*, uma vez que a mesma permitia adicionar conteúdo dinâmico as páginas até então estáticas. O java foi oficialmente lançado em 1995.

Atualmente a linguagem dá suporte ao desenvolvimento dos mais variados tipos de aplicações, devido esta flexibilidade ela é distribuída em três versões diferentes:

- JSE – “permite-lhe desenvolver e implantar aplicativos Java em desktops e servidores, bem como as exigências atuais dos ambientes integrados e em tempo real” (ORACLE, 2010a);
- JEE – é o Java utilizado para desenvolvimento de aplicações corporativas (ORACLE, 2010b).

A plataforma JME foi criada no intuito de permitir o desenvolvimento aplicações para dispositivos de pequeno porte, tais como: dispositivos eletrônicos de consumo, telefones celulares, PDA's entre outros aparelhos desta natureza. A figura abaixo ilustra a relação da plataforma JME:

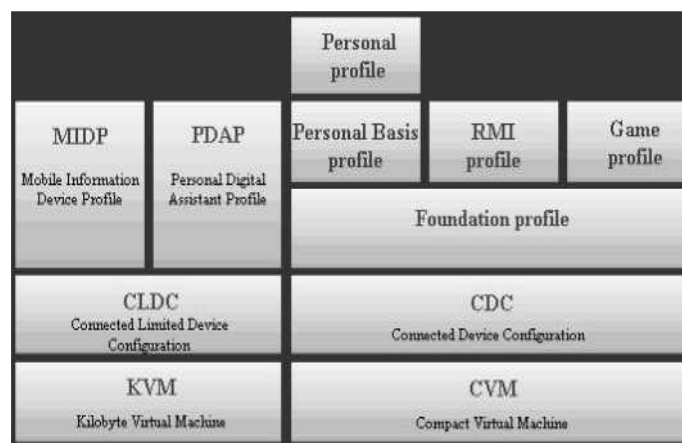


Figura 1: Plataformas Java SE, EE e ME (2010).

Esta variedade de dispositivos aliada as diferenças na capacidade de memória, processamento e tamanhos de tela fez como que algumas modificações fossem necessárias. Tais modificações visam adaptar a linguagem Java ao contexto dos dispositivos de pequeno porte. Dessas modificações surgiram dois novos conceitos importantes em JME:



Configurações e Perfis.

### **Configurações e Perfis**

Segundo Richpater (2008, p.8) uma configuração “define o conjunto básico de bibliotecas e máquina virtual Java (VM) para uma ampla gama de dispositivos”. Na plataforma Java ME temos duas configurações definidas CLDC e CDC como ilustra a figura 1.

A CLDC engloba os dispositivos com menos capacidade de memória volátil e não-volátil, os celulares – por exemplo. Inclui uma configuração mínima com 160KB de memória ROM e 192KB de memória RAM (Oracle 2010c). Esta configuração está, atualmente, na versão 1.1 tendo como principal incremento o suporte a operações matemáticas com ponto flutuante.

Em relação a segurança, as aplicações CLDC executam em ambiente fechado, deste modo somente as classes pré-definidas na configuração, perfis e pacotes adicionais podem ser acessadas. Por fim, o CLDC inclui o GCF que consiste em uma hierarquia de classe que generalizam as conexões de porta; entre os tipo de protocolos temos: TCP/IP , UDP e portas seriais (RICHPATER, 2008, p.11).

A CDC contempla dispositivos com mais capacidade que os dispositivos CLDC. Ela é definida pela JSR 36 e 128. Os dispositivos alvo da CDC devem possuir as seguintes características: incluir processador/controlador de 32-bit; memória RAM 2MB; memória ROM 2,5 MB (ORACLE, 2010d).

As configurações em si não são completas o suficiente para que se tenha uma plataforma de desenvolvimento de aplicações robusta. É neste contexto que surge a definição de perfil. Um perfil é “uma extensão de uma configuração. Ele fornece bibliotecas para um desenvolvedor escrever aplicações para um tipo particular de dispositivo” (MUSHOW, 2001, p.4).

Um perfil bem conhecido e amplamente utilizado é MIDP que segunda definição supramencionada seria uma extensão da CLDC, ele fornece API para o desenvolvedor

manipular conexões de rede, entrada e saída, armazenamento persistente entre outros. Além deste perfil, temos ainda o *Foundation Profile* (JSR 219), *Personal Basic Profile* (JSR 217) e *Personal Profile* (JSR 216) todos utilizados em conjunto com a CDC, veja a estruturação das configurações e perfis na figura 1.

Por fim, e não menos importante temos a KVM. Ela é a implementação de referência para a configuração CLDC. É assim chamada por ser constituída de uma pequena porção de classes em relação a JVM padrão, o que reduz seu tamanho à margem de *Kilobytes*. Além disso, duas restrições importantes estão presentes na KVM: métodos nativos não podem ser adicionados em tempo de execução; inclui apenas um subconjunto padrão de verificador de bytecodes, isto implica que a tarefa de pré-verificação é dividida entre o dispositivo e alguma ferramenta externa (KNUDSEN; LI, 2005, p.4).

### 3 MOBILE LEARNING

Os dispositivos computacionais móveis estão cada vez mais sofisticados, isto vem proporcionando grandes inovações na forma como eles são utilizados. Os recursos presentes em tais artefatos estão permitindo aos seus proprietários comodidade, praticidade, agilidade nas mais variadas tarefas diárias.

Desde a concepção do primeiro telefone celular (MOTOROLA, 2010) até o presente momento o mundo viu uma transformação imensa nos dispositivos computacionais móveis. Eles estão, cada vez mais, menores, leves, repletos de recursos, conectados etc. Esta brusca mudança possibilitou o uso dos mesmos em várias setores da sociedade como: negócios; comunicação; entretenimento; educação.

Com relação à utilização dos dispositivos móveis na educação esforços vêm sendo realizados à aplicabilidade deste paradigma computacional no auxílio do ensino-aprendizagem. Uma área do conhecimento bem recente trata dos problemas oriundos da utilização de dispositivos móveis na aprendizagem, tal área é denominada *Mobile Learning*.

Esta área vem crescendo muito, um bom exemplo disso é a Conferência Internacional de *Mobile Learning*<sup>4</sup> que ocorre anualmente desde 2005, cujo objetivo é divulgar trabalhos realizados, discutir questões importantes sobre novas formas de aplicação do *Mobile Learning* etc. Duas questões bastante discutidas sobre *Mobile Learning* são: reconceituação da natureza exata do *Mobile Learning*; avaliação em *Mobile Learning*.

Este capítulo discute os aspectos mais importantes sobre esta nova área do conhecimento, contemplando, principalmente, as questões supramencionadas. Além disso, apresenta algumas experiências de aplicação do *Mobile Learning* em situações reais. O objetivo aqui não é discutir a validade do *Mobile Learning* como forma de educação, pois isto em si renderia bons trabalhos de conclusão de curso. Para informações mais detalhadas conferir bibliografia dos autores mencionados.

---

<sup>4</sup><http://www.mlearning-conf.org/>

### 3.1 O que é *Mobile Learning*

O sucesso do *Mobile Learning* tornou-se seu próprio vilão. Isto se justifica pela grande quantidade de definições que surgiram a seu respeito. Várias são as tendências que buscam definir *Mobile Learning* e muitas vezes estas definições são concebidas com base em uma experiência particular. Neste sentido tornou-se difícil explicitar a natureza essencial do *Mobile Learning*, ou seja, não há um consenso acerca do que realmente é *Mobile Learning*.

Algumas definições focam simplesmente no uso de tecnologias móveis, considerando que *Mobile Learning* se caracteriza pelo uso de telefones celulares, *smartphones*, *PDA's* etc. Estas definições são as mais comuns na literatura.

Neste sentido, GEDDES (2004 apud Savil-Smith, Attewell e Tribal 2006, p. 2) define *Mobile Learning* como “aquisição de algum conhecimento e habilidade através do uso de tecnologia móvel, em qualquer lugar e qualquer momento, que resulta em alteração no comportamento”. Do mesmo modo, Trexler (2005 apud Savil-Smith, Attewell e Tribal 2006, p. 2) afirma que *Mobile Learning* é “qualquer oferta de educação onde a única tecnologia dominante são os *handhelds* e *palmtops*”.

Estas definições são, de certa forma, genéricas, deixando de levar em consideração alguns aspectos importantíssimos da educação como: conteúdo; avaliação; relação professor-aluno, aluno-aluno. Muito embora o uso de tecnologia móvel possa, devido suas características, potencializar aprendizagem dos usuários.

Outros autores colocam *Mobile Learning* diretamente relacionada ao *Eletronic Learning* e *Distance Learning* como é o caso de Quinn que define da seguinte forma:

*Mobile Learning* é a interseção da computação móvel com *eletronic learning*: recursos acessíveis onde quer que esteja, forte capacidade de pesquisa, integração rica, suporte poderoso para aprendizagem eficaz e avaliação baseada no desempenho. *Eletronic Learning* é independente da localização no tempo e no espaço (QUINN, 2000 apud SAVIL-SMITH; ATTEWELL; TRIBAL, 2006, p. 2).

Esta definição é bem consistente, pois leva em consideração não somente a tecnologia, mas também fatores importantes no processo de ensino-aprendizagem como: pesquisa; avaliação; integração.

Outra definição que segue o mesmo sentido, porém menos elaborada, diz que “*Mobile Learning* é uma forma de *Distance Learning* e *Eletronic Learning*” (GEORGIEV et all,

2004). Essas visões colocam o *Mobile Learning* com apenas uma diferença em relação as outras formas de educação, ou seja, a mobilidade. Embora esta seja a principal característica do *Mobile Learning* ela não é a única.

Uma outra visão bem diferente das mencionadas anteriormente é apresentada por Nyiri (2002) que diz que “*Mobile Learning* é aprendizagem que ocorre no curso da comunicação móvel pessoa-pessoa”. Nota-se que esta definição tem um foco diferente com relação ao *Mobile Learning*, pois o autor destaca a comunicação pessoa-pessoa como característica principal.

A próxima seção apresenta as principais tendências com relação as definições do *Mobile Learning* que são encontradas na literatura.

### 3.1.1 Perspectivas em Torno do Mobile Learning

Nota-se que até o presente momento não há um consenso acerca da definição exata do *Mobile Learning*. Um excelente trabalho realizado por (WINTERS, 2005, p.7) apresenta as perspectivas em torno do *Mobile Learning*:

<i>Perspectiva</i>	<i>Característica</i>
Tecnocêntrica	Nesta perspectiva <i>Mobile Learning</i> é vista como aprendizagem usando dispositivos móveis (WINTERS, 2005, p.7).
Relacionada ao Eletronic Learning	Esta perspectiva vê o <i>Mobile Learning</i> como uma extensão do <i>Eletrocic Learning</i> , o que não ajuda muito em caracterizar uma natureza única de <i>Mobile Learning</i> (WINTERS, 2005, p.7).
Aumentando Educação Formal	Esta perspectiva está relacionada à posição do <i>Mobile Learning</i> com relação a educação formal (WINTERS, 2005, p.7).
Centrada no Aluno	Esta perspectiva considera o <i>Mobile Learning</i> da perspectiva do aluno, deste modo a tecnologia não está mais no centro da questão. Além disso, o foco passa a ser na mobilidade dos alunos (WINTERS, 2005, p.7).

Tabela 2: Perspectivas do Mobile Learning

Como pode ser visto, houve um melhora significativa em relação a caracterização da natureza única do *Mobile Learning*. Ainda é pouco, mas a divisão apresentada mostra que as pesquisas na área estão em pleno desenvolvimento.

No entanto, o contribuição apresentada acima é de grande valia para os avanços necessários ao *Mobile Learning*, tendo em vista que seu potencial está longe de ser esgotado.

### **3.1.2 Características do Mobile Learning**

Mesmo com o impasse apresentado acerca da melhor definição de *Mobile Learning* é possível definir algumas características chaves. Winters apresenta, no mesmo trabalho mencionado acima, algumas características do *Mobile Learning*.

A primeira característica está relacionada à “habilidade de construção de conhecimento em contextos diferentes” (WINTERS, 2005, p.8). Neste caso, vale ressaltar a importância da mobilidade que permite ao aluno ir além do contexto da sala de aula. Além disso, os dispositivos móveis estão com recursos suficientes para proporcionar várias formas de aprendizagem, como também de coleta de informações.

A segunda característica tem como ênfase a “construção do entendimento pelos próprios alunos” (WINTERS, 2005, p.8). Esta possibilidade é ampliada pelos dispositivos móveis, uma vez que o contato dos alunos com tais dispositivos é bem natural. Vale ressaltar que outros dispositivos computacionais podem possuir esta característica, mas o uso de dispositivos móveis apresenta-se mais natural mediante a mobilidade dos mesmos e dos alunos que os portam.

A terceira característica se refere as mudanças que as tecnologias móveis provocam nas atividades de trabalho e aprendizagem (WINTERS, 2005, p.8). Isto se verifica nas formas de comunicação, potencialização de trabalho colaborativo, acesso a informação em qualquer lugar e qualquer hora. As tecnologias móveis permitem a realização de atividades que outros tipos de tecnologia não o fazem, isto muda os padrões na forma de conduzir atividades de estudo e trabalho.

A última característica está relacionada ao tempo/espaço do contexto de aprendizagem móvel. A autora coloca que “o contexto do *Mobile Learning* está além do tempo e espaço” (WINTERS, 2005, p.8). Estas características são bem fáceis de verificar, com relação ao tempo o *Mobile Learning* se mostra bem flexível, haja vista que o aluno pode usar seu dispositivo em qualquer momento não ficando preso a horários pré-definidos. Já em relação ao espaço, a própria mobilidade torna o contexto do *Mobile Learning*, do ponto de vista espacial, bem mais amplo, por exemplo, em comparação ao espaço da sala de aula convencional.

### **3.2 Avaliação em Mobile Learning**

Um dos pontos-chaves discutidos em relação ao *Mobile Learning* está relacionada as formas de avaliação. Independente da perspectiva teórica escolhida muitos questionamentos acerca da avaliação surgiram de imediato. Por exemplo, como recolher as variáveis para avaliação? Como garantir que as atividades estão sendo realizadas?

Além dos detalhes técnicos e pedagógicos, existem vários fatores de ordem ética que ampliam as dificuldades com relação a avaliação em *Mobile Learning*. Traxler e Bridges (2004, p.203-207) discutem as dificuldades éticas e legais no processo de avaliação em *Mobile Learning*, definindo alguns itens importantes:

- Consentimento informado: se refere “aos participantes entenderem a natureza, dimensão, duração e significado de seu envolvimento com a pesquisa”(TRAXLER; BRIDGES, 2004, p. 204);
- Risco de Participante: Podem surgir vários fatores de risco aos participantes como: vazamento de informações; *spam* em *e-mail*; consentimento dos pais (menores de idade) etc;
- Participante se retira: Este item diz respeito a saída do participante em qualquer ponto. Isto entra em conflito com os interesses dos avaliadores que desejam manter o participantes no processo;

- Pagamento ou compensação: Este item diz respeito a alguma compensação aos participantes pelo tempo dedicado, problemas podem surgir caso a compensação não seja considerada justa;
- Confidencialidade e anonimato: É importante evitar falsos participantes, além disso a confidencialidade dos participantes deve ser garantida;
- Distinção entre público e privado: O usuário necessita de distinção clara sobre se uma ação é privada ou pública;
- Regras para diferentes papéis: Cada agente do processo tem regras diferentes: professor, organizador, aluno etc;
- Status e poder: As relações aluno/professor, aluno/pesquisar caracterizam relações de status e poder, podendo impor questões éticas;
- Diferenças Culturais: “Muitas avaliações ocorrem através das fronteiras culturais onde várias partes podem ser diferentes em termos de linguagem, expectativas e valores” (TREXLER; BRIDGES, 2004, p.206);
- Desenvolver esclarecimento eficaz: é obrigatório assegurar que os participantes estejam conscientes de sua participação até o final do processo. Para tal, entrevistas devem ser realizadas no decorrer do processo.

Outro trabalho importante que expõe questões importantes que devem ser levadas em consideração durante a avaliação em *Mobile Learning* é apresentado por Taylor (2006, p.26-28) que enfatiza alguns itens não relacionados diretamente com questões éticas, porém não menos importante:

- Acesso aos dados: segundo o autor as tecnologias móveis estão mudando as formas de acesso aos dados impactando diretamente nas atividades;
- Perda de controle: o professor já não define tarefas com tempo pré-determinado para conclusão. O aluno agora pode utilizar seu dispositivo móvel e realizar a tarefa assincronamente;
- Cada vez mais pessoal: os dispositivos são cada vez mais pessoais, tornando difícil a coleta de informações do mesmo por terceiros;



- Avaliação e *design*: “a relação entre avaliação e design é multifacetada, mas o conceito chave para avaliação no *Mobile Learning* é a urgente necessidade por envolvimento no processo de design” (TAYLOR, 2006, p.27).

Em seu trabalho Taylor expõe problemas que são típicos do processo de ensino-aprendizagem convencional, porém deve ser levado em consideração que “acessar” o caderno de um aluno para verificar uma atividade é diferente de fazer o mesmo com seu celular. Outro ponto importante mencionado é a perda de controle que pode ocorrer em atividades de *Mobile Learning*, já que haverá momentos em que os alunos estarão dispersos o que amplia a dificuldade de acompanhamento.

Shaples e Vavoula (2008) apontam seis dificuldades no processo de avaliação em *Mobile Learning*:

- Captura do contexto de aprendizagem e aprendizagem através do contexto: a aprendizagem ocorre em uma variedade de sentidos, além disso novos contextos são criados através das interações;
- Alguém não aprendeu?: a aprendizagem móvel informal pode ser muito pessoal e indescritível. Portanto, torna-se difícil determinar antecipadamente onde ela pode ocorrer e que resultados produz;
- Questão ética: está relacionada a privacidade que os usuários querem em relação aos seus dispositivos, deste modo o acesso aos mesmos pode se configurar como invasão de privacidade;
- Tecnologia: as limitações dos dispositivos podem comprometer a avaliação.
- Contexto mais amplo: segundo o autor é importante considerar também o contexto institucional do *Mobile Learning* o que amplia a dificuldade na avaliação.
- Formal ou informal?: segundo o autor caracterizar como formal ou informal mediante a diferença existente entre *Mobile Learning* e a educação tradicional é problemático.

Como pode ser visto nos itens mencionados acima vários são os fatores que devem ser levados em consideração ao avaliar em *Mobile Learning*. Desde aspectos éticos a tecnológicos são citados. Esta problemática, embora exista em ambientes educacionais, ganha força graças a mobilidade presente em *Mobile Learning*.

## 4 LOGO

Atualmente existem várias iniciativas com relação ao uso de *softwares* com propósitos educacionais ou utilizados para esta finalidade, muito embora ainda haja bastante resistência quanto a utilização de tais artefatos na educação formal. Os exemplos de uso são isolados em sua maioria, porém esta prática vem ganhando força.

Algumas iniciativas interessantes em relação à disponibilidade de *softwares* educacionais são (EDUBUNTU, 2010) que pode ser adicionado ao (UBUNTU, 2010) disponibilizando várias aplicações educacionais. Além deste, temos ainda o (LINUX EDUCACIONAL, 2010) totalmente voltado para uso em ambiente educacional. Estas duas iniciativas se destacam pelo fato de agruparem várias opções de *software* em um único pacote, diferentemente de outras opções de *softwares* que são encontrados de forma dispersa.

Com relação aos tipos de *software* existem várias definições de acordo com o propósito como aponta Valente (1999, p. 90-97): tutoriais; programação; processador de texto; multimídia e internet; desenvolvimento de multimídia e internet; simulação e modelagem; jogos. Cada um destes tipos de *softwares* possui características que definem suas formas de uso, sendo possível a adaptação dos mesmos para o contexto ideal – tarefa do professor.

Embora os tipos de *software* supramencionados possam contribuir para a construção do conhecimento por parte dos envolvidos, um deles possui características interessantes – a programação. Durante o processo de programação o aluno deve realizar algumas tarefas essenciais para resolução de um problema – pensar o problema, descrevê-lo em termos de uma linguagem formal e sem ambiguidades, verificar a validade de sua idéia, entre outras atividades. Estas ações possibilitam o desenvolvimento intelectual do aluno à medida que o mesmo pode verificar se a solução apresentada é válida ou não.

Neste contexto, apresenta-se o Logo. Seu sucesso se justifica pela simplicidade e potencial para auxiliar no processo de construção de conhecimento dos alunos.

## 4.1 O que é Logo?

O Logo é uma linguagem de programação voltada para o âmbito educacional no intuito de promover a construção de conhecimento pelo próprio usuário. Este objetivo é obtido através de uma interface simples e intuitiva que permite ao usuário foco na atividade, além disso, apresenta um *feedback* claro do desenvolvimento das idéias do usuário. Ela foi criada em 1967 por Seymour Papert e Wallace Feurzeig no laboratório de Inteligência artificial do Instituto de Tecnologia de Massachusetts.

Os elementos básicos do ambiente de programação Logo são um *prompt* de comandos no qual o usuário adiciona os comandos que ele deseja e, geralmente, uma tartaruga na tela que responde imediatamente aos comandos fornecidos. A ação realizada pela tartaruga é verificada pelos rastros (geralmente riscos na tela) que a mesma deixa ao se movimentar. Comandos podem ser agrupados em procedimentos para criar um novo comando, no entanto eles produzem os mesmos efeitos se fossem executados individualmente.

### 4.1.1 Características do Logo

O Logo apresenta várias características que o tornam realmente válido para uso em ambientes educacionais, tais características são as seguintes (LOGO FOUNDATION, 2010):

- Interatividade: possui *feedback* imediato para o usuário, além de mensagem sobre erros e comandos.
- Modular: é possível criar procedimentos de forma a tornar a descrição de uma idéia mais modularizada.
- Extensível: através da criação de procedimentos a linguagem vai sendo estendida.
- Flexível: números são tratados como palavras, mas se pode realizar operações aritméticas normalmente. Não se prende a definições de tipo de dados.

### 4.1.2 Filosofia do Logo

A filosofia do Logo tem como proposta fundamental a construção do conhecimento pelo próprio usuário. Isto é fomentado através da interação com o ambiente de programação permitindo que o usuário teste suas idéias acerca da solução de um problema que o mesmo deseja resolver.

Esta ação desencadeia várias atividades importantes para o desenvolvimento do intelecto. Pois, para aplicar uma solução à um problema em Logo é preciso descrevê-lo de modo formal. Em seguida o usuário verifica se sua idéia é válida, neste momento o usuário terá a resposta sobre a validade de sua idéia. Todo esse processo é acompanhado pela reflexão do próprio usuário da ferramenta. Portanto, o ponto fundamental é a elaboração e refinamento da solução para um determinado problema. Sendo esta uma característica exclusiva do Logo como aponta Valente (1993B, p.14) dizendo que “ O fato de a atividade de programação em Logo propiciar a descrição das idéias como subproduto do processo de resolver um problema, não é encontrada em nenhuma outra atividade que realizamos”.

Além disso, a forma como as atividades em Logo são conduzidas permitem mais autonomia ao aluno já que “[...] o controle do processo de aprendizagem está na mãos do aprendiz e não nas mãos do professor[...]” (VALENTE, 1993A, p.23) e isto permite, também pelas características da linguagem, que o aluno aprenda pondo a “mão na massa”, ou seja, construindo seu conhecimento.

Uma consequência desta autonomia que é atribuída aos alunos é o erro em uma das fases de elaboração de uma solução para um problema em questão. No Logo, diferentemente dos ambientes educacionais tradicionais que condenam o erro sob todos aspectos, o erro é uma etapa importante para a construção do conhecimento, pois “[...] ele ocorre da formalização da idéia. Portanto, o aluno ao implementar sua idéia em termos de linguagem de programação pode verificar sua validade, ou perceber erros conceituais..” (VALENTE, 1993A, p.24).

## 4.2 Ambiente Logo

Quando o Logo foi criado seu objetivo era comandar um robô, de alguma forma este robô se assemelhava com uma tartaruga. Talvez este fato seja o motivo de a maioria dos interpretadores virtuais do Logo usarem uma tartaruga na tela para simular os movimentos.

A figura 2 abaixo ilustra o ambiente virtual Logo denominado SuperLogo 3.0 (NIED, 2010). Como se pode ver o ambiente apresenta um janela de comandos e a tela na qual o tartaruga fará os movimentos e deixará seu rastro.

Este ambiente permite também a criação de procedimentos, cuja é idéia principal é utilizar os comando primitivos para criar comandos mais complexos. Além disso, ele informa ao usuário quando um comando não faz parte da base de conhecimento. Existem outras implementações de interpretadores Logo, no entanto a essência dos mesmos não muda muito com relação ao SuperLogo, afinal todos devem incorporar as principais características da linguagem.

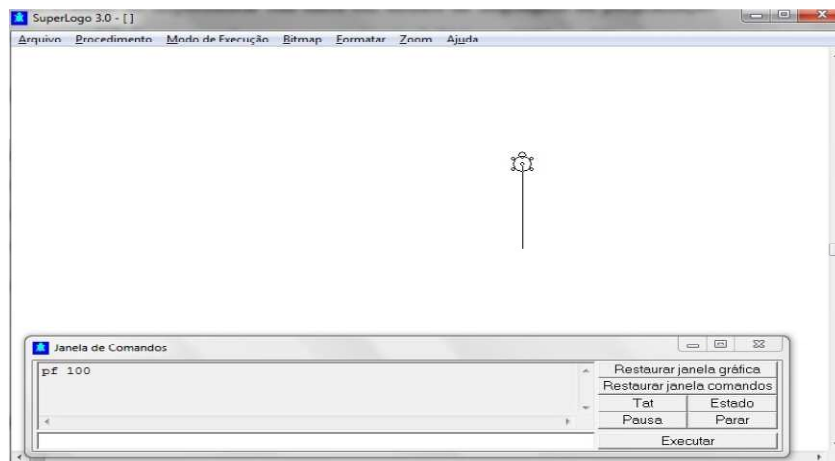


Figura 2: Interpretador SuperLogo 3.0

### 4.2.1 Característica computacional

As principais atividades desenvolvidas em Logo estão, geralmente, relacionadas à exploração de conceitos espaciais. Isto permite um contato imediato do usuário com o computador fazendo com que o mesmo explore conceitos espaciais ao comandar a tartaruga para se mover de um lugar para outro na tela, deste modo desenvolvendo conceitos sobre espaço, geometria, matemática entre outros (VALENTE, 1993A, p.20).

Para fazer com que a tartaruga realize tais movimentos o usuário necessita aprender comandos da linguagem. Estes comandos são semelhantes a linguagem natural. Por exemplo, para fazer a tartaruga andar para frente o usuário basta utilizar **pf 100** ou **parafrente 100**.

Além disso, permite-se que o usuário crie novos procedimentos. Isto é útil quando ele quiser, por exemplo, criar um quadrado. Ao invés de fornecer uma sequência de comandos individualmente ele pode criar um procedimento chamado **quadrado** com os comandos necessários e então executá-lo, como é ilustrado figura 3.

```
aprenda quadrado
repita 4
[
parafrente 100
paradireita 90
]
fim
```

*Figura 3: Procedimento "Quadrado"*

Por fim, vale ressaltar que “o objetivo não deve ser centrado no produto que o aluno desenvolve, mas na filosofia de usar o computador e como ele está facilitando a assimilação de conceitos que permeiam as diversas atividades” (VALENTE, 1993A, p.22).

## 4.2.2 Geometria da Tartaruga

Geometria da Tartaruga é um termo criado por Seymour Papert em seu livro *Mindstorms: children, computers, and powerful ideas*. Segundo o próprio autor, existe uma nova forma de ensinar geometria que é através da geometria da tartaruga.

Esta nova forma é diferente, segundo Papert, do estilo euclidiano – com base na lógica e do estilo de Descartes – com base em álgebra. Semelhante ao estilo euclidiano a geometria da tartaruga também possui uma entidade fundamental, neste caso a tartaruga seria esta entidade assim como o ponto para o estilo euclidiano (PAPERT, 1993, p.55).

A vantagem que a tartaruga tem em relação ao ponto é que ela pode ser relacionada a coisas que as pessoas conhecem diferentemente de um ponto, pois este precisa de uma abstração mais apurada. Além disso, a tartaruga apresenta outras propriedades, tais como a cabeça que indica uma orientação, o ângulo de inclinação.

Esta nova concepção de fazer geometria é chamada por Papert de “estilo computacional de geometria” devido o fato de se utilizar o computador como ferramenta mediadora.

## 4.3 Mal entendidos sobre Logo

Embora a linguagem Logo já esteja consolidada como ferramenta educacional, alguns mal entendidos acompanham a mesma há um bom tempo. A primeira questão diz respeito ao Logo ser uma linguagem boa para crianças, em resposta a isto Papert (1983) afirma que “uma linguagem que for boa para crianças no sentido estrito não seria boa para crianças, Logo é bom para crianças à medida que é para adultos”.

Em segundo lugar entra questão relacionada ao Logo ser excelente para gráficos, quanto a isto autor diz que uma linguagem para ser boa para gráficos precisa ser uma linguagem poderosa e de propósito geral, tal como o Logo (PAPERT, 1983).

Por fim, entra a questão acerca do nível de complexidade da linguagem, o qual muitos consideram como sendo fácil. Em contrapartida o autor afirma que Logo é simples no início,

porém vai se tornando complexa a medida que o usuário vai aumentando seu nível de conhecimento. Deste modo, a complexidade vai aumentando a medida que as atividades vão sendo realizadas, configurando um processo gradativo de complexidade (PAPER, 1983).



## 5 MOBILOGO

O Mobilogo é fruto do projeto de extensão da UEPB denominado *Mobilogo: uma proposta de desenvolvimento de software educativo para telefones celulares*, este projeto está cadastrado na PROEAC sob o número 07.49.012.09. Neste projeto se iniciou o desenvolvimento do Logo para telefones celulares com o objetivo de permitir aos usuários do aplicativo uma forma de entretenimento com aprendizado. O projeto Mobilogo conta com a participação de dois alunos de graduação e dois professores da instituição mencionada.

Atualmente o Mobilogo está em sua versão 1.0, cujas funcionalidades da Linguagem Logo implementadas são os comandos primitivos: para frente, para trás, para direita e para esquerda. Algumas funcionalidades estão em desenvolvimento e testes, por conta disto considerou-se mais cabível não colocá-las na versão 1.0.

Este capítulo apresenta os principais detalhes do projeto, tais como: requisitos, ferramentas, detalhes da implementação etc.

### 5.1 Requisitos

Os requisitos funcionais e não-funcionais definem as principais características da aplicação, desde os componentes do *software* até os detalhes mais técnicos relacionados a implementação. Neste sentido, buscou-se identificar requisitos tanto funcionais quanto não-funcionais que possibilitassem o desenvolvimento de um versão inicial utilizável.

Deste modo, a versão 1.0 do Mobilogo apresenta ao usuário recursos básicos que possibilitam a utilização do *software*, muito embora seja necessário a implementação de recursos adicionais para a aplicação em versões posteriores.

### 5.1.1 Requisitos Funcionais

Os requisitos definidos para versão 1.0 contemplam os comandos primitivos da linguagem Logo. O objetivo desta escolha está relacionada com a tentativa de definir um núcleo base para o restante dos comandas, dado que a partir destes comandos pode-se criar procedimentos mais complexos.

Deste modo, os seguintes requisitos funcionais foram definidos:

- Componente 1 – Menu da aplicação: o objetivo deste componente é permitir ao usuário navegar entre as opções fornecidas na versão 1.0;
- Componente 2 – Tela do Logo: representa o ambiente Logo, aqui o usuário fornece comandos para a tartaruga;
- Componente 3 – Tela Ajuda: aqui o usuário obterá ajuda sobre como utilizar a aplicação e como fornecer comandos para a tartaruga;

Além disso, o diagrama de casos de uso abaixo ilustra algumas tarefas que o usuário poderá realizar.

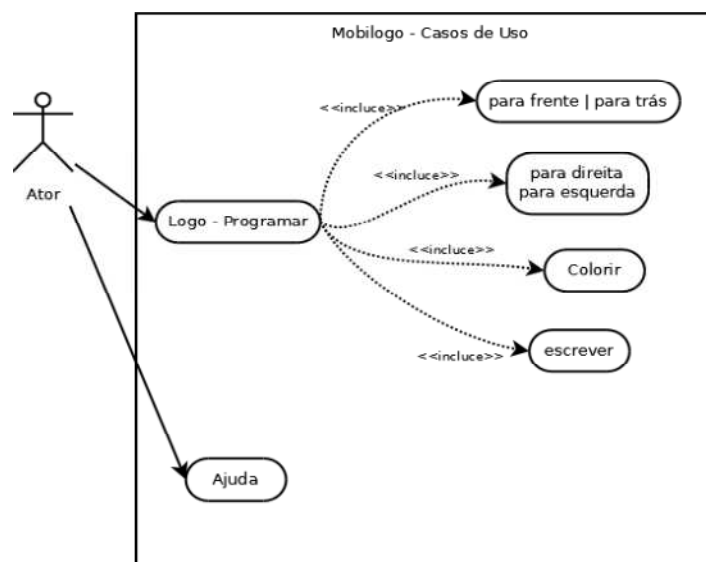


Figura 4: Mobilogo - Casos de Uso

Deste modo, os comandos da linguagem disponíveis nesta versão da aplicação são os seguintes:

- Para frente – representado pelas iniciais pf <distância>;
- Para trás – representado na aplicação por pt <distância>;
- Para direita – representado na aplicação por pd <grau de rotação>;
- Para esquerda – representado na aplicação por pe <grau de rotação>.

### 5.1.2 Requisitos não-funcionais

Os requisitos não funcionais definidos para o Mobilogo contemplam, principalmente, a plataforma de desenvolvimento. Além disso, tem-se um manual de ajuda para facilitar o uso do *software*.

Com relação a plataforma optou-se por desenvolver o aplicativo utilizando a Plataforma JME. Primeiro, porque é uma plataforma amplamente utilizada para desenvolvimento de *software* para dispositivos móveis, que tem como consequência uma grande quantidade de material de apoio disponível. Segundo, porque a maioria dos dispositivos móveis suportam aplicações desenvolvidas em JME.

As configurações da plataforma foram assim definidas:

- CLDC versão 1.1;
- MIDP versão 2.1.

## 5.2 Usabilidade

No intuito de fornecer mais usabilidade para aplicação algumas mudanças com relação ao ambiente Logo disponível para computadores foram efetuadas. A forma como as instruções

são passadas para tartaruga é a principal mudança que há no Mobilogo. Para fornecer comandos o usuário utilizará as teclas de game dos telefones celulares ou contrário de fornecer comandos de modo textual. Portanto, a seta para cima indica um comando, a seta para baixo indica outro comando e assim segue.

### **5.3 Ferramentas Utilizadas**

O bom desenvolvimento de uma aplicação está diretamente relacionada as ferramentas escolhidas. Pensando nisto foram escolhidas várias ferramentas para suporte ao Mobilogo que permitissem desenvolvimento ágil e com qualidade. A ferramentas escolhidas incluem uma IDE robusta, um SDK, além de outras ferramentas auxiliares. Esta seção descreve de forma resumida cada uma das ferramentas utilizadas.

#### **5.3.1 NetBeans IDE**

O netbeans é um ambiente de desenvolvimento de *software* integrado, é gratuito e possui código aberto. Além disso, é possível desenvolver *software* para *Web*, *desktop* e dispositivos móveis. O ambiente permite desenvolver aplicações em várias linguagens de programação tais como: c/c++, ruby, java entre outras (NETBEANS, 2010).

Com relação ao desenvolvimento de aplicações móveis o NetBeans IDE apresenta vários recursos interessantes. Com ele é possível desenvolver aplicações para o versões 1.0, 2.0 e 2.1 do MIDP e para as versões 1.0 e 1.1 do CLDC. Por padrão, o NetBeans vem com o JME SDK 3.0, no entanto é possível configurar SDK's de terceiros.

Outros recursos interessantes estão presentes no IDE como:

- *Visual Mobile Designer*: criação rápida de componentes como telas de espera, de *login* etc.
- Desenvolvedor de jogos para dispositivos móveis: é possível desenvolver jogos com

um assistente que permite a criação de cenas com *sprites*.

Este ambiente de desenvolvimento é muito robusto fornecendo os mais variados tipos de ferramentas. Aqui foram mencionadas algumas ferramentas e características importantes sobre o desenvolvimento de aplicações móveis. Para mais detalhes consulte <http://netbeans.org/index.html>.

### 5.3.2 SDK Sony Ericsson

O Sony Ericsson SDK é o *kit* de ferramentas específico para desenvolvimento de aplicações Java para dispositivos Sony Ericsson. Ele permite o desenvolvimento de aplicações MIDP nas versões 1.0, 2.0 e 2.1. Várias ferramentas interessantes para o desenvolvimento de aplicações para dispositivos móveis são fornecidas no kit. Estas ferramentas em conjunto formam um plataforma de desenvolvimento robusta, agilizando o desenvolvimento de aplicações (SONY ERICSSON, 2010).

As seguintes ferramentas são fornecidas com o Kit:

- *Device Explorer*: esta ferramenta permite testar as aplicações em telefones celulares reais, permitindo ao desenvolvedor verificar possíveis problemas;
- *Run MIDP Application*: permite a execução de arquivos \*.jar, pode ser utilizado para testar aplicações;
- Ktoobar: emulador para aplicações java, traz consigo vários modelos de telefones celulares Sony Ericsson.

### 5.3.3 Project Hosting

O *project hosting* é um serviço de hospedagem de código aberto oferecido pela Google. Com este serviço é possível realizar o desenvolvimento colaborativo. O serviço oferece as

seguintes possibilidades (GOOGLE, 2010):

- Hospedagem no Subversion. É possível integrar ao Netbeans;
- Ferramentas de análise de código;
- Rastreador de problemas e *wiki* simples;
- Destaque e fluxo de atualizações que facilitam o controle de projetos.

Este serviço é gratuito, para utilizá-lo basta ter apenas uma conta no google. Esta ferramenta foi útil para o gerenciamento do projeto. Além disso, permitiu compartilhamento de código entre os participantes do projeto.

## **5.4 Implementação**

Esta seção apresenta os principais detalhes técnicos relacionados ao processo de desenvolvimento do Mobilogo. Os códigos fontes do Mobilogo estão disponíveis no seguinte endereço <http://code.google.com/p/mobilogo-rca>. O projeto está sob a *GNU General Public Licence* v3 (GNU, 2007) podendo o mesmo ser totalmente modificado.

### **5.4.1 Dispositivo de Referência**

O dispositivo de referência é peça importante do processo de desenvolvimento de aplicações para dispositivos móveis. Ele define os recursos básicos para executar a aplicação como: tamanho de tela, versão do MIDP e CLDC entre outros fatores. Deste modo, a escolha do dispositivo de referência está diretamente relacionada aos requisitos mínimos da aplicação, como também o público-alvo (WELLS, 2004, p.254).

Neste sentido, o dispositivo escolhido como sendo referência para o Mobilogo foi o Sony Ericsson W200. A escolha deste está relacionada a disponibilidade de um dispositivo para realização de testes. Isto possibilitará melhor visualização do aplicativo em um

dispositivo real, afinal não se cria aplicações para dispositivos móveis e as usa em computadores de mesa ou *notebooks*

## 5.4.2 Classes da Aplicação

A aplicação está dividida em algumas classes, cujo objetivo é representar objetos distintos para melhor modularização do projeto. Esta organização permite escrever um código com mais clareza e conseqüentemente mais fácil de ser modificado. A figura 5 abaixo ilustra o diagrama de classes da aplicação:

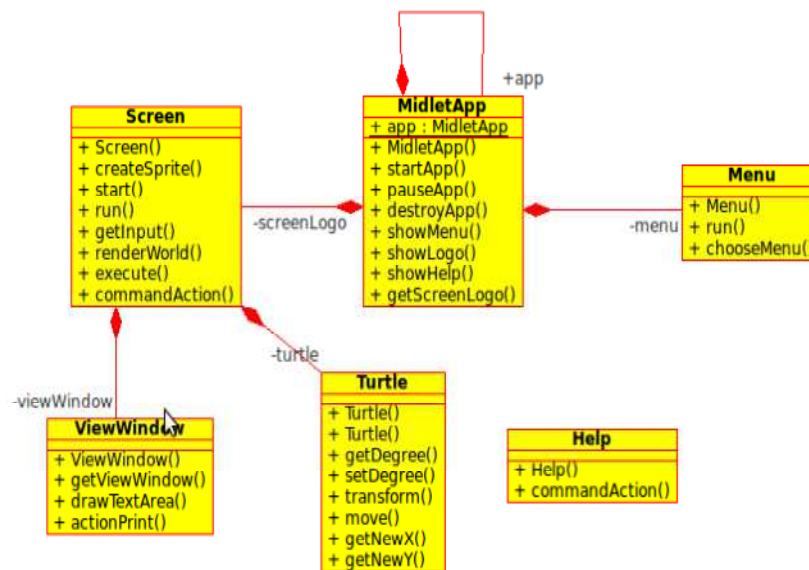


Figura 5: Diagrama de Classes da Aplicação

### **MidletApp**

Toda aplicação JME necessita de uma *MIDlet* para ser executada em um dispositivo que suporte JME MIDP. A classe *MidletApp* estende a classe `javax.microedition.midlet.MIDlet`, além dos métodos padrões de *MIDlet* a classe *MidletApp* implementa os seguintes métodos como:

- `public void showMenu()` - Este método exibe na tela o menu da aplicação. Que permite

navegar entre as opções do aplicativo;

- *public void showLogo()* - Este método exibe o ambiente Logo. Momento no qual o usuário pode fornecer comandos para a tartaruga;
- *public void showHelp()* - Mostra a tela de ajuda da aplicação.

## **Menu**

Esta classe representa o menu de opções da aplicação, ele estende a classe `javax.microedition.lcdui.Canvas`. A partir dela é possível navegar pelas opções do aplicativo. O menu de opções apresenta uma estrutura simples que permite boa usabilidade por parte dos usuários. Na versão 1.0 apresenta três opções. Além disso, a classe `Menu` implementa *Runnable* do pacote `java.lang`, deste modo foi possível fazer os efeitos de navegação para o menu da aplicação.

Além das opções mencionadas, a classe `Menu` também carrega uma imagem que representa o Logo da aplicação. Tanto os itens, quanto a imagem são desenhados em posições relativas ao tamanho da tela do celular. Deste modo, é possível ter boa aparência em celulares diferentes do Sony Ericsson W200.

## **ViewWindow**

Esta classe é utilizada para representar a área da tela do usuário. Ela possui um objeto do tipo `javax.microedition.lcdui.Image` no qual os movimentos da tartaruga são desenhados, além disso é neste objeto que é desenhado a área de texto referente a entrada de dados do usuário.

A idéia de utilizar uma classe para representar a área de desenho tem como base a técnica *Double Buffering*. Esta técnica cria dois *buffers* um na tela e outro fora dela.



O primeiro é realizado através do método *paint()* da classe *Canvas* onde os códigos para desenho são colocados. Já o segundo é realizado por intermédio de uma imagem onde todos os gráficos são desenhados na imagem e em seguida na tela. Desenhar na imagem garante que o processo só pare quando todos os desenhos forem realizados, deste modo evitando gráficos mal formados(WELLS, 2004, p.222).

Neste sentido, a classe *ViewWindow* foi criada para implementação da técnica mencionada. Na mesma classe temos um método importante, *actionPrint()*. Este método é responsável por desenha na tela a caixa de texto e a entrada de dados do usuário.

## **Turtle**

A classe *Turtle* representa o principal componente do aplicativo, a tartaruga. Ela estende a classe *javax.microedition.lcdui.game.Sprite*. Esta classe possui dois métodos importantes *transform()* responsável por fazer a rotação da tartaruga para direita ou para esquerda, o outro método é *move(int distance, String cmd, ViewWindow viewWindow)* responsável por fazer a tartaruga se mover de uma posição para outra.

A classe possui uma imagem com as tartarugas de 0 a 90°. Através do método *transforme()* é possível obter o restante dos graus na circunferência trigonométrica como ilustra a figura 6 abaixo, onde os ângulos das áreas cinzas são obtidas através de transformações presentes na classe *javax.microedition.lcdui.game.Sprite*.

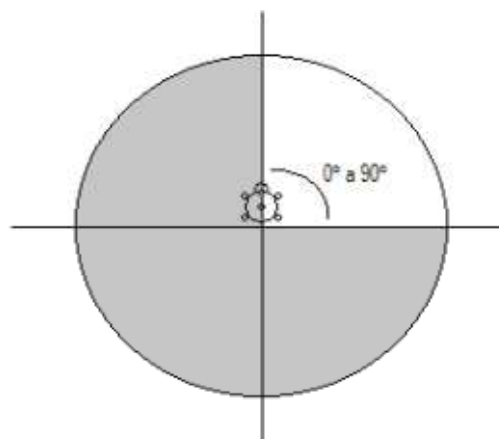


Figura 6: Transformação da Tartaruga

O método *move(int distance, String cmd, ViewWindow viewWindow)* utiliza o *ViewWindow* para desenhar os traçados da tartaruga.

## **Help**

Esta classe é responsável por implementar uma seção de ajuda sobre os comandos utilizados para dar instruções a tartaruga. Ela estende a classe *Canvas*. Ela não possui nenhum método com funções especiais para aplicação.

## **Screen**

A classe *Screen* representa a tela da aplicação, é nela que o *ViewWindow* é desenhando. Nela também se captura a entrada de dados do usuário. Ela estende a classe *javax.microedition.lcdui.Canvas*.

Os principais métodos desta classe são:

- *execute()* - Este método é responsável por pegar a entrada do usuário e passar para o objeto *Turtle* para execução dos comandos fornecidos;
- *renderWorld()* - O objetivo deste método é renderizar todos os gráficos na tela do usuário. Inclusive os traçados e a tartaruga.

### **5.4.3 Telas da aplicação**

A versão 1.0 do *Mobilogo* possui poucas telas devido as funcionalidades implementadas para esta versão. É possível que este o número de telas aumente de acordo com o aumento de opções disponíveis na aplicação para o usuário.

## Tela do Menu

A primeira tela que o usuário tem contato é com o menu da aplicação. Esta tela é bem intuitiva, possuindo uma imagem e o título da aplicação. Além disso, abaixo da imagem e do título do aplicativo tem-se as opções disponíveis ao usuário. A figura 7 ilustra melhor o menu da aplicação.



*Figura 7: Menu da aplicação*

## Tela do Ambiente Logo

A próxima tela representa o ambiente Logo. Também é um interface simples e sugestiva, deste modo o usuário tende a manter o foco na aplicação. Esta tela possui um caixa de texto no canto inferior da tela de dispositivo, nesta caixa de texto é apresentado os comandos fornecidos pelo usuário como ilustra a figura 8.



Figura 8: Ambiente Logo

## Telas de Ajuda

As telas de ajuda tem por objetivo orientar o usuário em como fornecer comandos para tartaruga. Inicialmente é fornecido a sintaxe dos comandos como ilustra a figura 9 e 10.



Figura 9: Ajuda - Sintaxe



Figura 10: Ajuda - Descrição

A outra tela complementa estas informações, através de uma descrição dos comandos mostrados na figura acima. Esta descrição, embora resumida, tende ser objetiva.

## 6 Conclusões

Neste trabalho realizou-se extensa pesquisa bibliográfica acerca de computação móvel, *mobile learning* e a linguagem Logo, que em conjunto formam as bases para o projeto Mobilogo. Os resultados obtidos neste estudo mostraram que tanto computação móvel quanto *mobile learning* estão em crescente ascensão.

Além disso, estas duas áreas do conhecimento possuem uma característica em comum, a mobilidade. Esta característica é a mola propulsora do sucesso de tais áreas devido a necessidade quase que natural que as pessoas sentem por dispositivos que ofereçam condições para serem usados nos ambientes mais distintos.

Especificamente sobre computação móvel mostra-se consolidada, haja vista a grande variedade de dispositivos móveis oferecidos e serviços. No entanto, a computação móvel ainda tem muito potencial para ser explorado através de avanços em *hardware*, conectividade e técnicas de desenvolvimento de *software*.

Em relação ao *Mobile Learning* pode-se dizer que esta ainda é uma área muito recente, porém vem ganhando força através de várias experiências com a mesma. Além do mais, a evolução do *mobile learning* tem forte influência dos dispositivos computacionais móveis aliado à organização e planejamento.

Neste contexto desenvolveu-se o Mobilogo que é uma implementação do Logo para dispositivos móveis que tem por objetivo explorar as principais características da computação móvel e *Mobile Learning*.

A versão atual do Mobilogo permite ao usuário aplicar comandos para desenho de várias figuras geométricas, possibilitando o trabalho com ângulos e distância. Embora alguns aspectos importantes do Logo ainda não estejam incorporados na implementação é possível desenvolver várias conceitos importantes relacionados ao espaço e formas geométricas, como também noções relacionadas a ângulos.

Como sugestão para trabalhos tem-se:

- Construir módulo que permita criação de procedimentos;
- Criar comando repita;

- Criar módulo para persistência de desenhos do usuário;
- Criar ambiente de *Mobile Learning* com uso da aplicação;
- Adicionar outros comandas da linguagem Logo;

## Referências

APPLE. **IOS 4: O Sistema Operacional mais Avançado do Mundo**. Disponível em: <<http://www.apple.com/br/iphone/ios4/>>. Acesso em 03 nov. 2010, 20:15:38.

AROKIAMARY, V.J. **Mobile Computing**. 3. ed. India: Technical Publications, 2009, p.13,16-17.

B'FAR, R. **Mobile Computing principles: designing and developing mobile applications with UML and XML**. New York: Cambridge University press, 2005, pg.7,9-11,14,16.

Brown, D. **Símbolo Perdido**. Rio de Janeiro: Editora Sextante Ltda, 2009.

Deitel. H. M. **Java Como Programar**. 6. ed. São Paulo: Pearson Education, 2007, pg. 9.

EDUBUNTU. **What is Edubuntu?** Disponível em <<http://edubuntu.org/about>>. Acesso em 18 nov. 2010.

ESTADÃO. **Uso de Celular na Educação Ganha Prêmio Internacional**. Disponível em: <[http://www.estadao.com.br/estadaodehoje/20071102/not\\_imp74434,0.php](http://www.estadao.com.br/estadaodehoje/20071102/not_imp74434,0.php)>. Acesso em: 25 nov. 2010.

GEORGIEV, T., GEORGIEVA, E. e SMRIKAROV, A. **M-Learning – A new stage of e-learning**. In. INTERNATIONAL CONFERENCE ON COMPUTER SYSTEMS AND TECHNOLOGIES, 2004, Rouse. Artigo... Rouse: Bulgaria, 2004. p. V.28.1-V.28.5.

GNU. **Gnu General Public Licence**. Disponível em: <<http://www.gnu.org/licenses/gpl.html>>. Acesso em: 25 nov. 2010.

GOOGLE. **Hospedagem de projetos no Google Code**. Disponível em <<http://code.google.com/intl/pt-BR/projecthosting/>>. Acesso em: 25 nov. 2010.

LI, S; KNUDSSSEN, J. **Beginning J2ME: from novice to professional**. New York: Apress, 2005, pg. 4.

LINUX EDUCACIONAL. **Linux Educacional**. Disponível em <<http://linuxeducacional.com/mod/wiki/view.php?id=53>>. Acesso em 18 nov. 2010.

LOGO FOUNDATION. **The Logo Programming Language**. Disponível em <<http://el.media.mit.edu/logo-foundation/logo/programming.html>>. Acesso em 18 nov. 2010.

LOUREIRO, A.A.F; MATEUS, G. R. **Introdução a Computação Móvel**. Rio de Janeiro: Dcc/imp, Coppe/UFRJ, 1998, pg.2.

MOTOROLA. **Making History: development the portable cellular system**. Disponível em: <<http://www.motorola.com/staticfiles/Business/Corporate/US-EN/history/feature-cell-phone-development.html?globalObjectId=7662-10813>>. Acesso em 20 ago. 2010, 15:18:56.

MUSHOW, J. W. **Core J2ME Technology & MIDP**. Michigan: Prentice Hall, 2001, pg. 4.

NETBEANS. **Netbeans IDE – a forma inteligente de codificar**. Disponível em: <[http://netbeans.org/features/index\\_pt\\_BR.html](http://netbeans.org/features/index_pt_BR.html)>. Acesso em 25 de nov. 2010.

NIED. **SuperLogo 3.0**. Disponível em: <[http://www.nied.unicamp.br/software/software\\_detalhes.php?id=33](http://www.nied.unicamp.br/software/software_detalhes.php?id=33)>. Acesso em 30 de nov. 2010.

NYIRI, K. **Towards a philosophy of m-learning**. In. IEEE INTERNATIONAL WORKSHOP ON WIRELESS AND MOBILE TECHNOLOGIES IN EDUCATION, 2002, Växjö. Artigo... Växjö: TELEBORG CAMPUS, 2002.

ORACLE. **Connected Limited Device Configuration (CLDC); JSR 30, JSR 139 Overview**. Disponível em: <<http://www.oracle.com/technetwork/java/overview-142076.html>>. Acesso em 01 nov. 2010C, 11:00:20.

ORACLE. **Java EE at a Glance**. Disponível em: <<http://www.oracle.com/technetwork/java/javaaee/overview/index.html>>. Acesso em 26 out. 2010B, 15:30:45.

ORACLE. **Java ME Technology – CDC**. Disponível em <<http://www.oracle.com/technetwork/java/javame/tech/overview-jsp-141092.html>>. Acesso em 01 nov. 2010D, 11:29:25.



ORACLE. **Java SE at a Glance**. Disponível em:  
<<http://www.oracle.com/technetwork/java/javase/overview/index.html>>. Acesso em: 26 out. 2010A, 16:52:15.

PAPERT, S. **Turtle Geometry: a mathematics made for learning**. In. \_\_\_\_\_.  
Mindstorms: children, computers, and powerful ideas. New York: BASIC BOOKS, 1993,  
pg.55.

PAPERT, S. **Misconceptions about Logo**. Cretive Computing Vol.10, No.11, Novembro  
1984, PP.229. Disponível em:  
<[http://www.atarimagazines.com/creative/v10n11/229\\_Misconceptions\\_about.php](http://www.atarimagazines.com/creative/v10n11/229_Misconceptions_about.php)>. Acesso:  
15 ago 2010.

PEREIRA, R. **Resultado de Exame Via Celular**. Disponível em: <<http://www.parana-online.com.br/canal/tecnologia/news/201384/>>. Acesso em: 26 nov. 2010.

RISCHPATER, R. **Beggining Java™ Me Platform**. New York: Apress, 2008, pg.8, 11.

SATYANARAYANAN, M. **Pervasive Computing: Vision and Challanges**. Personal  
Communications IEEE, Vol. 8, Issue 4, p. 10 – 17, ago. 2002. ISSN 1070-9916.

SAVIL-SMITH C.; ATTEWELL, J.; TRIBAL, G.S. **Mobile In Practice**. Learning and Skill  
Network, London. 2006, pg.2.

SONY ERICSSON. **Sony Ericsson SDK for the Java ME Plataform**. Disponível em:  
<<http://developer.sonyericsson.com/wportal/devworld/technology/java/sdk?cc=gb&lc=en>>.  
Acesso em: 25 nov. 2010.

TALUKDER, A.K; YAVAGAL, R.R. **Mobile Computing**. New York: Tata McGraw-Hill,  
2007,pg.7-8.

TAYLOR, J. **Evaluating Mobile Learning: what are appropriate methods for evaluating  
in mobile environments?** In. SHARPLES, M. *Big Issues in Mobile Learning*. Nottingham:  
The University of Nottingham, 2006. p. 26-29.

TRAXLER, J.; BRIDGES, N. **Mobile Learning – the ethical and legal challenges**. In: SAVIL-SMITH, C.; ATTEWELL, J. *Mobile Learning Anytime Everywhere*. London: Learning and Skill Development Agency, 2005. p. 203-207.

UBUNTU. **About Ubuntu**. Disponível em <<http://www.ubuntu.com/project/about-ubuntu>>. Acesso em 18 nov. 2010.

VALENTE, J. A. (Org). **Análise dos diferentes tipos de softwares usados na Educação**. In. \_\_\_\_\_. O computador na sociedade do conhecimento. Campinas: UNICAMP/NIED, 1999. pg. 90-97.

VALENTE, J.A (Org). **Diferentes usos do computador na educação**. In. \_\_\_\_\_. Computadores e Conhecimento: Repensando a educação. Campinas: UNICAMP/NIED, 1993A. pg.20,22-23.

VALENTE, J.A (Org). **Por quê computador na educação**. In. \_\_\_\_\_. Computadores e Conhecimento: Repensando a educação. Campinas UNICAMP/NIED, 1993B. pg.14.

VAVOULA, G. & SHARPLES, M. (2008). **Challenges in Evaluating Mobile Learning**. In TRAXLER, J., RIORDAN, B. and DENNETT, C., *Proceedings of MLearn 2008: The bridge from text to context*, Wolverhampton, UK: University of Wolverhampton, pp. 296-303.

VAVOULA, G. et.all. **Myartspace: Design and evaluation of support for learning with multimedia phones between classrooms and museums**. Computers and Education, 53, 2, 286-299. Disponível em <<http://dx.doi.org/10.1016/j.compedu.2009.02.007>>. Acesso em: 18 nov. 2010.

VIRKUS R. et. all. **Don't panic: Mobile Developer's Guide**. 6. ed. Bremen: Enough Software, 2010, pg.22,27-28,64-67,69-70.

WELLS, M. **J2ME Game Programming**. 2. ed. Boston: Thomson Couse Technology, 2004. pg.4, 252, 254.

WINTERS, N. et.all. **What is a mobile learning**, In. Shaples M. (ed.) *Big Issues in Mobile Learning*, The University of Nottingham, Nottingham. 2006, pg.5, 7-8.

ZHENG, L. NI, L. **Smart Phone and Next Generation Mobile Computing**. San Francisco:

Elsevier, 2006, pg.3.

\_\_\_\_\_. **In J2ME/MIDP Achieve image rotation.** Disponível em:  
<<http://www.anyang-window.com.cn/in-j2memidp-achieve-image-rotation/>>. Acesso em 20  
nov. 2010.