

Um Ambiente de Ensino para Redes de Comunicação Sem Fio Integrando Componentes de Software e J2ME

Ivan J. Foschini

Universidade Federal de São Carlos - Departamento de Computação
São Carlos – Brasil - Zipcode: 13565-905
ivan@dc.ufscar.br

e

Sérgio D. Zorzo

Universidade Federal de São Carlos - Departamento de Computação
São Carlos - Brasil - Zipcode: 13565-905
zorzo@dc.ufscar.br

Abstract

This article describes an application that allows the creation of educational content for Mobile Computing environments. Although this application had been developed to be used in desktops, the educational content generated must be visualized in devices that uses Wireless Communication to access the World Wide Web, specifically Palmtops. Therefore, it is expected that Palmtop users may obtain educational content wherever they are, profiting the inherent advantages of the Wireless Networks. The application described was implemented using the Components-Based Software Development technique, exploiting this development technique together to Java 2 Micro Edition, version of the Java technology created for the development of mobile devices applications.

Keywords: Data Communication and Computer Networks, Software Components, Java 2 Micro Edition

Resumo

Este artigo descreve uma aplicação que possibilita a criação de conteúdo didático para ambientes de Computação Móvel. Embora esta aplicação tenha sido desenvolvida para ser utilizada em computadores de mesa convencionais, o conteúdo didático por ele gerado é direcionado para ser visualizado por dispositivos que utilizam a Comunicação Sem Fio para acessar a Rede Mundial de Computadores, notadamente os Palmtops. Com isso, deseja-se que os usuários de Palmtops possam obter material didático em qualquer lugar onde estejam, usufruindo das vantagens inerentes às Redes de Comunicação Sem Fio. O ambiente descrito foi implementado utilizando-se as técnicas do Desenvolvimento de Software Baseado em Componentes, explorando este método de desenvolvimento em conjunto com a Java 2 Micro Edition, versão da tecnologia Java desenvolvida especificamente para a criação de aplicações para dispositivos móveis.

Palavras Chave: Transmissão de Dados e Redes de Computadores, Componentes de Software, Java 2 Micro Edition

1 Introdução

A Educação a Distância (EAD) é uma modalidade de ensino consagrada e consolidada em diversas partes do mundo [5]. Há tempos, educadores e outros profissionais perceberam as vantagens de se oferecer cursos a distância, e sentiram a necessidade de utilizar novas tecnologias no processo educacional. Quanto às vantagens dos cursos a distância, podem-se destacar: os aspectos econômicos, já que os estudantes não precisam ter custos com viagens e hospedagem, a

possibilidade de o aluno poder estudar nos momentos que melhor lhe convierem e o fato de cada um poder seguir seu próprio ritmo de aprendizado, dentre outros fatores.

As novas tecnologias, principalmente os computadores e as redes de computadores, ampliaram ainda mais as possibilidades e vantagens da Educação a Distância. As redes de computadores, notadamente a *World Wide Web*, tornam possível que qualquer pessoa, em qualquer parte do mundo onde seja possível acessar a rede mundial de computadores, tenha fácil acesso a uma grande quantidade de conteúdo didático.

A utilização das tecnologias de comunicação sem fio e de computação móvel ampliam ainda mais os horizontes das aplicações de Educação a Distância, uma vez que, ao se utilizar meios de comunicação sem fio, o único limitante para a distribuição de conteúdo didático passa a ser a área de cobertura de uma Rede Sem Fio, e não mais a existência de uma conexão cabeada convencional com uma rede de computadores.

O objetivo principal deste artigo é apresentar um ambiente, baseado em componentes de software, que possibilita a criação de conteúdo educacional de forma rápida, eficiente e intuitiva. Este material didático, no entanto, não é direcionado para ser utilizado em microcomputadores de mesa ou quaisquer outros dispositivos semelhantes. A preocupação é criar-se conteúdo educacional que possa ser visualizado em pequenos dispositivos dedicados, com tela pequena e características arquiteturais próprias.

O dispositivo escolhido para a visualização do material didático a ser desenvolvido é o *Palmtop*, pequeno computador de mão. Dispositivos desse tipo estão capacitados, atualmente, a receber dados da *World Wide Web* através de um acesso sem fio, o que possibilita a seus usuários receber o material didático onde quer que se encontrem, independentemente de possuírem a sua disposição uma rede de computadores convencional ou não.

Com isso, une-se as vantagens de três frentes de pesquisa atuais: a Educação a Distância, a Computação Móvel e o Desenvolvimento de Software Baseado em Componentes, principal técnica de desenvolvimento de software utilizada nos dias atuais.

Este artigo está dividido em seções que abordam os seguintes temas: a seção 2 trata da Java 2 Micro Edition (J2ME), tecnologia desenvolvida pela Sun Microsystems para a criação de aplicações para dispositivos dedicados; a seção 3 aborda o processo de Desenvolvimento de Software Baseado em Componentes; o objetivo da seção 4 é fazer uma descrição informal do ambiente descrito; a seção 5, por sua vez, apresenta a funcionalidade deste ambiente; na seção 6 é apresentada a arquitetura do ambiente; a sua interface é exposta na seção 7; finalmente, a seção 8 apresenta as conclusões a respeito do uso da Educação a Distância em ambientes de Computação Móvel, bem como do emprego da técnica de Desenvolvimento de Software Baseado em Componentes agregado à tecnologia J2ME, e como tudo isso contribui para a distribuição de conteúdo didático.

2 Java 2 Micro Edition (J2ME)

O ambiente descrito conta com dois módulos: o Módulo do Professor e o Módulo do Aluno. O Módulo do Professor permite ao educador criar listas de exercícios para avaliação dos alunos e, posteriormente, levantar dados estatísticos a respeito do desempenho dos estudantes. Este módulo é executado em computadores de mesa (*desktops*). Já o Módulo do Aluno, desenvolvido com o auxílio da tecnologia Java 2 Micro Edition (J2ME), é executado em *Palmtops*. Esta seção apresenta esta tecnologia, mostrando a sua origem e as suas principais características.

2.1 Origem da J2ME

A tecnologia Java percorreu, desde a sua criação, um longo caminho. Graças, principalmente, à portabilidade inerente à tecnologia, foram desenvolvidas aplicações Java para computadores de mesa, para servidores e para a *World Wide Web*. Atualmente também estão sendo criadas aplicações Java para serem executadas em pequenos dispositivos dedicados, tais como telefones celulares e *handhelds* [2].

Com a popularização da tecnologia e com a certeza de que uma única versão da Java poderia não ser tão eficiente para os diversos dispositivos que poderiam utilizá-la, a Sun Microsystems passou a desenvolver uma nova Plataforma Java, que atendesse às necessidades individuais de cada um desses diferentes dispositivos. Surgiu então a Java 2 Platform, que reúne três novas edições Java: a Java 2 Enterprise Edition (J2EE), a Java 2 Standard Edition (J2SE) e a Java 2 Micro Edition (J2ME).

2.2 A Java 2 Platform

Cada uma das novas edições da Java 2 Platform foi projetada, portanto, para um tipo de dispositivo específico, respeitando as características individuais de cada um deles, tais como memória, capacidade de armazenamento e tamanho da tela, por exemplo.

Enquanto a J2EE é voltada para o desenvolvimento de aplicações que serão executadas em servidores e a J2SE é uma ferramenta para a criação de sistemas a serem implantados em *desktops*, a J2ME foi criada para o desenvolvimento de aplicações a serem exibidas em dispositivos pequenos, tais como Palmtops e telefones celulares, que possuem pequenos computadores dedicados e que frequentemente utilizam baterias como fonte de energia. Tais dispositivos são chamados de *embedded devices* [8].

Embora cada uma das edições tenha sido desenvolvida pensando em dispositivos específicos, as três edições da Java 2 Platform compartilham algumas características comuns [12]. A primeira delas é a mais famosa e notável característica da tecnologia Java: uma vez criada uma aplicação, esta pode ser executada em qualquer outro dispositivo com suporte à tecnologia. Pensando mais especificamente em dispositivos como Palmtops, que possuem dezenas de Sistemas Operacionais disponíveis e uma grande diversidade de fabricantes com arquiteturas totalmente distintas umas das outras, uma linguagem de programação que permita ao desenvolvedor criar aplicações que possam ser executadas independentemente do Sistema Operacional ou do fabricante do produto possui uma enorme vantagem em relação às suas concorrentes.

Todas as edições têm suas aplicações interpretadas por uma máquina virtual. O conceito de máquina virtual é amplamente utilizado pela tecnologia Java, e é o principal responsável pela portabilidade dos aplicativos. Para que uma aplicação Java possa ser executada em uma máquina real qualquer, esta máquina real deve possuir uma máquina virtual Java instalada. É essa máquina virtual que vai fazer com que o programa seja executado, independentemente da arquitetura ou das características reais da máquina. Cada uma das edições da Java 2 Platform possui uma máquina virtual otimizada para o dispositivo específico do seu escopo de aplicações.

Algo também importante e comum às três edições é que toda a integridade do código Java é verificada pela máquina virtual antes do programa ser executado, o que dificulta a disseminação de vírus. Antes de executar o programa, a máquina virtual vasculha o código em busca de trechos que possam vir a causar algum dano ao dispositivo do usuário, visando com isso protegê-lo de programas maliciosos, tais como vírus e vermes.

2.3 Características da Java 2 Micro Edition

Embora compartilhe com as demais edições da Java 2 Platform diversas características comuns, a J2ME possui também uma série de características próprias e outras que, embora também apareçam tanto na J2EE como na J2SE, merecem um destaque especial quando se fala na J2ME, dadas as particularidades dos dispositivos aos quais esta edição se destina.

A primeira característica que se destaca na J2ME é que as suas aplicações provêm interfaces gráficas de qualidade, com gráficos coloridos e diversos componentes de interface, o que facilita a interação dos usuários com as aplicações [6]. O fato de a J2ME oferecer interfaces gráficas de qualidade pode vir a resolver um dos maiores problemas práticos encontrados por desenvolvedores de aplicações para dispositivos de tela pequena: criar projetos que realmente atraiam a atenção dos usuários, acostumados a aplicativos multimídia e altamente atrativos em *desktops*. Um dos maiores desafios dos desenvolvedores desse tipo de aplicativo é fazer com que os usuários encontrem vantagens reais em utilizar algum tipo de aplicação em dispositivos de tela pequena, deixando de lado aplicativos similares em seus *desktops*.

Além disso, componentes de interface tais como botões e caixas de texto podem ser facilmente implementados quando se utiliza a J2ME, sem o grande esforço de programação encontrado quando se trabalha com outras linguagens para o desenvolvimento de aplicações para dispositivos de tela pequena. A linguagem C, por exemplo, amplamente utilizada para o desenvolvimento de aplicativos para Palmtops, exige um número excessivo de linhas de código para a criação de interfaces, e a qualidade de tais interfaces é, frequentemente, duvidosa.

Adicionalmente, aplicações J2ME podem fazer uso de qualquer infra-estrutura de redes sem fio, tirando vantagens de protocolos voltados para este tipo de rede. Aplicações J2ME podem ser distribuídas utilizando-se tanto redes comutadas por circuito quanto redes comutadas por pacotes. Redes comutadas por pacotes significarão um aumento de dez vezes na velocidade da comunicação, além de permitir que a distribuição dos aplicativos utilize protocolos padrões da Internet convencional, tais como HyperText Transfer Protocol Secure (HTTPS) sobre Secure Socket Layer (SSL), para encriptação de dados, por exemplo [14]. Outro aspecto relevante da tecnologia J2ME é a sua preocupação em oferecer segurança às aplicações. Segurança é uma das questões mais importantes e discutidas quando se fala em comunicação sem fio, pois, ao contrário de uma rede cabeada, onde os dados trafegam por vias bem definidas, informações trocadas através de uma rede sem fio trafegam livremente pelo ar, e qualquer pessoa com um equipamento apropriado pode interceptar o fluxo de dados. Conforme dito anteriormente, o fato de aplicações J2ME poderem fazer uso de protocolos como HTTPS e SSL oferece maior segurança às aplicações, o que pode atrair a atenção de empresas que necessitem de aplicativos nos quais a segurança é fundamental, tais como instituições bancárias e de comércio eletrônico.

Outra característica importante da J2ME é que suas aplicações podem comunicar-se com *Servlets*, o que permite que estes aplicativos acessem Servidores Web e bancos de dados remotos através da *World Wide Web* [3]. *Servlets* são classes baseadas em tecnologia Java que são executadas e instanciadas em associação com Servidores Web, atendendo

requisições realizadas por meio do protocolo HyperText Transfer Protocol (HTTP) [7]. Com isso, as aplicações móveis passam a ter muito mais alcance e tornam-se tão poderosas quanto aquelas desenvolvidas para redes convencionais. Conforme mencionado previamente, cada uma das edições da Java 2 Platform possui uma máquina virtual otimizada para as características particulares dos dispositivos aos quais as edições se destinam. A máquina virtual da J2ME é chamada de K Virtual Machine (KVM) [9]. A KVM é uma máquina virtual Java projetada especialmente para dispositivos com telas, memórias e capacidade de processamento limitados, tais como Palmtops e telefones celulares, visando tirar o máximo proveito desses dispositivos, sem, contudo, exaurir os seus recursos.

3 Desenvolvimento de Software Baseado em Componentes

O desenvolvimento de componentes de software, ou, simplesmente, componentes vem ganhando notoriedade nos estudos de Engenharia de Software, por oferecer, principalmente, grande possibilidade de reuso de código e redução nos custos de implementação. Os componentes são, basicamente, trechos de código previamente testados, que podem ser instalados e executados em diversas aplicações distintas [10]. Essa propriedade, aliada ao fato de que um componente deve ser tão genérico quanto possível, faz com que o desenvolvedor poupe tempo e recursos de implementação. Atualmente, existem desenvolvedores que se dedicam exclusivamente à criação de componentes, que posteriormente podem ser comprados e utilizados pelos mais diversos usuários, de acordo com as necessidades individuais de cada um. Um componente oferece serviços aos usuários através da sua interface. A interface de um componente é composta por um conjunto de operações que determinam quais são os serviços que um determinado componente pode oferecer. Tais serviços podem ser, eventualmente, personalizados pelo desenvolvedor, através da alteração das propriedades do componente. As propriedades de um componente são os atributos relacionados a este componente. Um componente Cliente pode ter associado a ele algumas propriedades como Nome do Cliente e Número da Conta, por exemplo. Além das propriedades, os componentes também podem possuir alguns eventos associados a eles. Um evento ocorre por uma ação externa ao sistema, como uma ação do usuário, ou por uma ação interna do sistema, como o vencimento de um temporizador.

Pode-se dizer que os componentes são uma evolução no que diz respeito às técnicas de programação, que vêm sendo aprimoradas com o passar do tempo. Inicialmente, os softwares eram criados na linguagem de programação Assembler, muito mais próxima de uma linguagem de máquina do que de uma linguagem agradável ao ser humano. Posteriormente, surgiram as chamadas linguagens de alto nível, que permitiram expressar as idéias de forma simples e compacta. No entanto, logo os desenvolvedores de software perceberam que perdiam um longo tempo reprogramando estruturas de dados pré-existentes, o que deu origem às bibliotecas de funções. Tais bibliotecas aumentaram o nível de abstração e de reuso no desenvolvimento de aplicativos.

O próximo passo foi a Orientação a Objetos, que trouxe consigo, entre outros, os conceitos de encapsulamento e polimorfismo, que até hoje atraem muitos desenvolvedores. Contudo, apenas programar com objetos não garante o reuso prometido pela Orientação a Objetos. Somente depois que a idéia de objetos foi estendida às bibliotecas de classes o reuso começou a tomar forma, com o surgimento de várias bibliotecas comerciais de classes. Apesar do sucesso dessas bibliotecas, surgiu uma nova limitação: bibliotecas de classes são genéricas, e não trabalham, portanto, com modelos que poderiam fornecer soluções para problemas específicos.

Os *Frameworks* foram descritos como uma solução para esta limitação. Estes ambientes fornecem uma solução para um problema através de um modelo abstrato, que pode ser personalizado e ajustado às necessidades específicas de cada um. A dificuldade em se usar *Frameworks* é que eles são modelos reutilizáveis abstratos, que precisam de programadores experientes para ser especializados e concretizados. Surgiram então os componentes de software, que buscam combinar as vantagens dos modelos reusáveis com implementações igualmente reutilizáveis.

Com a popularização da técnica de desenvolvimento de componentes, algumas metodologias surgiram para orientar o processo de criação de componentes de software, tais como o Catalysis [1] e o Unified Process [4].

Independentemente da metodologia adotada para o desenvolvimento de componentes de software, os desenvolvedores devem ter em mente alguns princípios gerais que tornam o uso de componentes realmente atrativos:

- um componente deve ser simples de ser compreendido, para que possa ser facilmente implementado;
- um componente deve ser genérico, para que possa ser reutilizado em um grande número de aplicações com propósitos distintos;
- quando se desenvolve um conjunto de componentes independentes, estes devem ser facilmente combináveis para facilitar a montagem da aplicação;
- embora devam ser facilmente combináveis, diferentes componentes devem ter suas dependências minimizadas ou, se possível, evitadas;
- deve haver uma preocupação em se criar componentes que possam ser facilmente estendidos, para que novos serviços possam ser adicionados a eles, em caso de necessidades futuras;

- os componentes devem ser portáveis para diversas plataformas distintas, o que aumenta ainda mais as possibilidades de reuso.

O ambiente descrito neste artigo é implementado utilizando componentes de software tanto no desenvolvimento do Módulo do Professor, destinado para ser utilizado em *desktops*, quanto no Módulo do Aluno, onde se utilizam componentes desenvolvidos para a tecnologia Java 2 Micro Edition. Com isso, explora-se as vantagens deste método de desenvolvimento de aplicações, expostas nesta seção.

4 Descrição Informal do Ambiente

Os dois atores que interagem com o ambiente descrito neste artigo são o Professor e o Aluno. O Professor é identificado pelo seu nome, pela disciplina que ministra, pelo código desta disciplina e pela turma pela qual é responsável. O ambiente proporciona ao Professor a criação de uma ou mais Listas de Exercícios, que possuem os seguintes atributos: um código para identificação da Lista, o seu título, a sua data de criação e a sua data de entrega. As Listas são compostas por uma ou mais Questões.

Cada Questão possui um número único para identificá-la e é de um determinado tipo (Questão de Verdadeiro ou Falso ou Questão de Múltipla Escolha). As Questões de Verdadeiro ou Falso têm como atributos o texto da questão, uma justificativa que explica qual é a razão do texto da questão ser verdadeiro ou falso, e qual é a resposta correta à questão (verdadeiro ou falso). Já as Questões de Múltipla Escolha, por sua vez, têm como atributos o texto da questão, uma justificativa que explica porque determinada alternativa é considerada correta, os textos das cinco alternativas a serem apresentadas ao Aluno e a resposta correta da Questão (alternativa A, B, C, D ou E).

Os Alunos são identificados pelo seu código de aluno e pelo seu nome. O Aluno resolve uma Lista de Exercícios, gerando, com isso, uma Resolução, que contém o seu índice de acertos e um relatório a respeito do seu desempenho, indicando, por exemplo, quais foram as questões respondidas corretamente, quais foram respondidas incorretamente e uma justificativa para o fato de a resposta dada à questão ter sido considerada incorreta.

5 A Funcionalidade do Ambiente

Conforme dito anteriormente, o ambiente conta com dois módulos: o Módulo do Professor e o Módulo do Aluno. O Módulo do Professor é executado em *desktops*. Sua principal funcionalidade é permitir a criação simples e intuitiva de listas de exercícios com questões de múltipla escolha e de verdadeiro ou falso. Estes exercícios serão posteriormente acessados pelos *Palmtops* dos alunos, que deverão resolvê-los nestes dispositivos. Os dados relativos às resoluções dos alunos ficarão residindo no Servidor Web no qual também estão armazenadas as listas de exercícios.

Nesse momento entra em cena outra funcionalidade do Módulo do Professor, que oferece ao educador estatísticas relativas ao desempenho dos alunos. O professor pode saber, por exemplo, qual foi o percentual de acerto dos alunos em uma determinada lista de exercícios, bem como visualizar um relatório a respeito do desempenho dos alunos na resolução desta lista. Dessa forma, além de poder criar exercícios facilmente, o professor pode saber em quais partes da disciplina os alunos encontraram maiores dificuldades e quais os assuntos que foram dominados com maior facilidade, ganhando subsídios para aprimorar o processo de ensino.

O Módulo do Aluno, conforme foi dito, é executado em *Palmtops*, e foi criado com o auxílio de componentes baseados na tecnologia J2ME. Com isso, podem ser utilizadas nos exercícios todas as qualidades inerentes à J2ME, buscando-se, com isso, tornar os processos de aprendizado e de resolução de listas de exercícios mais agradáveis ao aluno atual e às suas aspirações. Acostumados com a tecnologia multimídia dos computadores de mesa atuais, dificilmente os alunos de hoje interessam-se por um conteúdo educacional visualmente pobre e desinteressante.

Com o uso desta tecnologia, os alunos encontram vantagens em resolver seus exercícios em dispositivos de tela pequena, o que os oferece, sobretudo, mobilidade. Uma vez que aplicações J2ME podem comunicar-se com *Servlets* e, com isso, fazer acesso a Servidores Web remotos, os alunos estão livres para resolver os exercícios em qualquer lugar que prefiram, em suas casas ou mesmo dentro de veículos, por exemplo. Dessa forma, não se faz mais necessário que os alunos tenham acesso às listas de exercícios apenas em um lugar que possua uma conexão cabeada tradicional, podendo resolver estas listas em qualquer ponto da área de cobertura de uma Rede de Comunicação Sem Fio.

6 A Arquitetura do Ambiente

Baseado na Descrição Informal e na Funcionalidade da aplicação, definiu-se, para o ambiente descrito neste artigo, a arquitetura exibida na Figura 1.

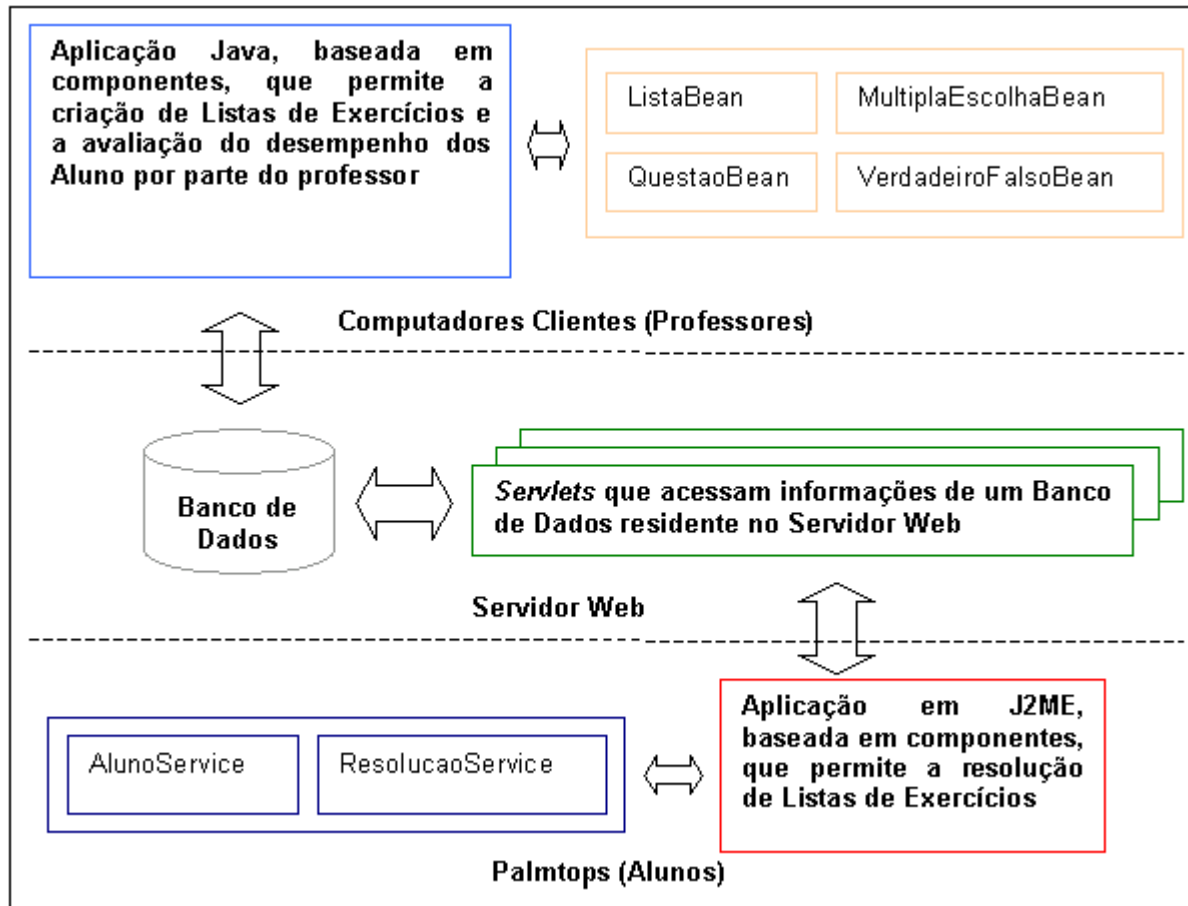


Figura 1 – Arquitetura de uma Aplicação de EAD que utiliza os componentes descritos

O Módulo do Professor previamente mencionado baseia-se em *JavaBeans*, e permite que professores criem listas de exercícios, contendo questões de verdadeiro ou falso e de múltipla escolha. A tecnologia *JavaBeans* foi responsável pela introdução do conceito de componentes em Java. Utilizando-se *JavaBeans*, pode-se criar componentes de software reutilizáveis e independentes de plataforma [11]. Esta aplicação conta com os seguintes *JavaBeans*: *ListaBean*, *QuestaoBean*, *MultiplaEscolhaBean* e *VerdadeiroFalsoBean*. Cada um destes *JavaBeans* possui alguns atributos específicos e métodos para inserir, alterar, recuperar e excluir informações de um banco de dados que armazena as listas de exercícios criadas pelos professores. O *JavaBean* *VerdadeiroFalsoBean*, por exemplo, possui como atributos o texto da questão, a resposta correta da questão (verdadeiro ou falso) e uma justificativa para a resposta correta, que é apresentada ao aluno caso este não acerte a questão que lhe foi proposta. Com isso, deseja-se que o aluno possa testar seus conhecimentos e aprender qual é a resposta correta em caso de algum erro de sua parte. As listas de exercícios geradas pelos professores ficam disponíveis em um banco de dados localizado em um Servidor Web remoto, no qual também residem *Servlets* que podem recuperar e armazenar dados neste banco de dados.

A arquitetura da Figura 1 exibe ainda o Módulo do Aluno, desenvolvido em J2ME, que faz uso de componentes de software especialmente desenvolvidos para esta tecnologia, o *AlunoService* e o *ResolucaoService*. Nesta tecnologia, componentes de software são chamados de *Services* [13]. Esta aplicação permite que os alunos resolvam as listas de exercícios previamente criadas pelos professores. O componente *AlunoService* possui como atributos informações relativas aos alunos, tais como código do aluno e nome do aluno. Já o componente *ResolucaoService* armazena dados relativos à resolução de uma lista de exercícios por parte de um aluno. A aplicação J2ME envia os dados da resolução aos *Servlets* residentes no Servidor Web. Os *Servlets*, por sua vez, recebem estas informações e as armazenam no banco de dados do Servidor Web, deixando-as à disposição ao professor. Com isso, o professor pode fazer uma avaliação do desempenho dos alunos na resolução de uma determinada lista de exercícios, utilizando outra funcionalidade do Módulo do Professor.

Cabe ressaltar que esta aplicação para EAD é apenas uma dentre várias possíveis, e a utilização de componentes de software permite que estes componentes sejam reutilizados por outros aplicativos, inclusive em plataformas distintas.

Além disso, os componentes descritos neste artigo podem ser facilmente estendidos ou modificados de acordo com as necessidades de outras aplicações de EAD.

7 A Interface do Ambiente

Esta seção apresenta a interface do ambiente descrito neste artigo, exibindo segmentos tanto do Módulo do Professor quanto do Módulo do Aluno. Quando o professor acessa o ambiente, a tela exposta na Figura 2 é apresentada:

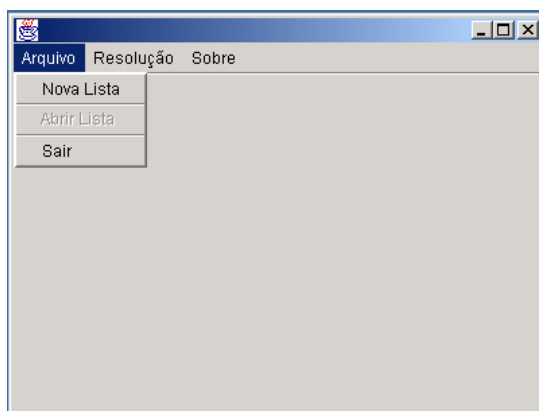


Figura 2 – Tela Principal do Ambiente

Conforme pode ser visto na Figura 2, uma série de opções são apresentadas ao professor, como, por exemplo, criar uma nova lista (opção “Nova Lista” no Menu “Arquivo” da Figura 2) ou visualizar uma lista previamente criada (opção “Abrir Lista”). Quando optar por criar uma nova lista de exercícios, o professor é conduzido à tela apresentada na Figura 3:

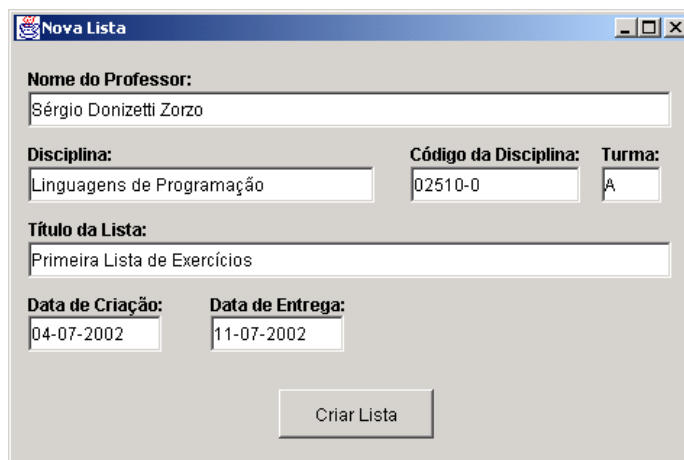
A imagem mostra uma janela de formulário com o título 'Nova Lista'. Os campos são: 'Nome do Professor:' com o valor 'Sérgio Donizetti Zorzo'; 'Disciplina:' com o valor 'Linguagens de Programação'; 'Código da Disciplina:' com o valor '02510-0'; 'Turma:' com o valor 'A'; 'Título da Lista:' com o valor 'Primeira Lista de Exercícios'; 'Data de Criação:' com o valor '04-07-2002'; e 'Data de Entrega:' com o valor '11-07-2002'. Um botão 'Criar Lista' está localizado na parte inferior.

Figura 3 – Tela para a criação de uma nova Lista de Exercícios

Nesta tela o professor fornece os dados relevantes para a criação de uma nova lista de exercícios, tais como o seu nome, o código e o nome da disciplina, a turma à qual a lista se destina, o título da lista de exercícios e a sua data de entrega e criação. Clicando no botão “Criar Lista” uma nova Lista de Exercícios é criada.

A partir daí, o professor pode começar a trabalhar na elaboração das questões que irão compor a lista recém-criada. Se optar por incluir na lista uma questão de verdadeiro ou falso, a tela da Figura 4 será exibida.

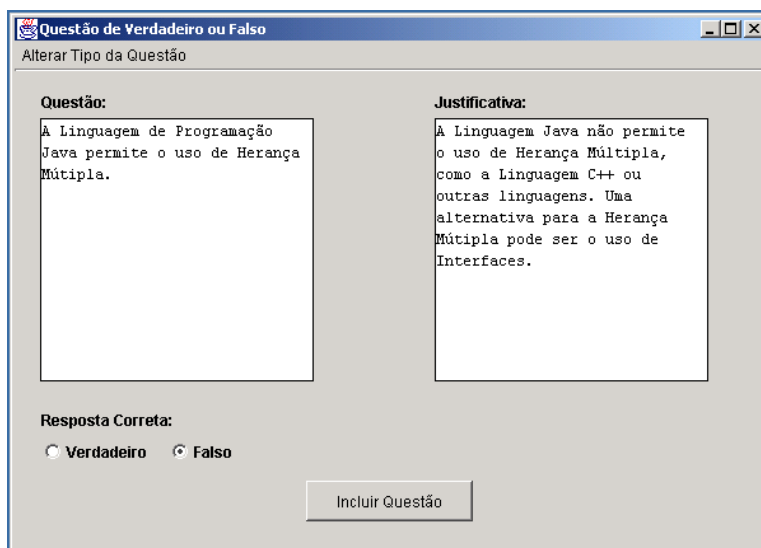


Figura 4 – Tela para a criação de questões de Verdadeiro ou Falso

O professor deverá escrever o texto da questão e uma justificativa sobre a razão do texto da questão ser verdadeiro ou falso. Esta justificativa será apresentada ao aluno quando este responder de forma incorreta a uma questão. O professor deverá assinalar ainda qual é a resposta correta (verdadeiro ou falso). Clicando no botão “Incluir Questão”, esta questão será imediatamente incluída na Lista de Exercícios.

Caso o professor deseje criar uma questão de múltipla escolha, a tela a ser exibida será a da Figura 5:

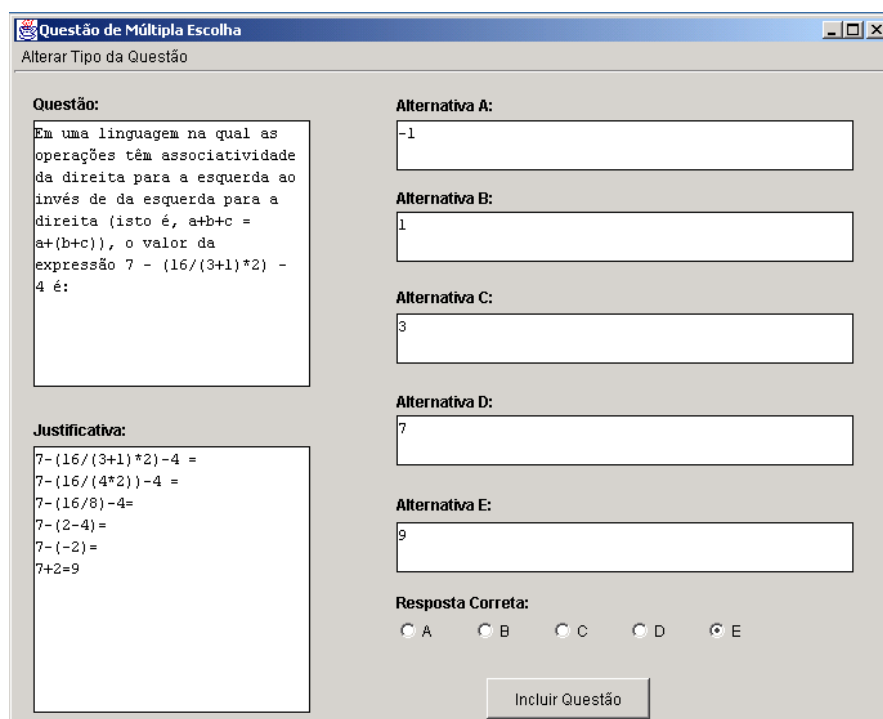


Figura 5 – Tela para a criação de questões de Múltipla Escolha

Analogamente, o professor escreverá o texto da questão e uma justificativa da resposta, a ser apresentada ao aluno em caso de erro na resposta. Será necessário, ainda, que o professor escreva o texto das alternativas e assinale a resposta correta. Clicando no botão “Incluir Questão”, a questão será incluída na Lista de Exercícios. As listas de exercícios criadas com o auxílio deste ambiente ficam armazenadas em um banco de dados residente em um Servidor Web remoto,

acessível para os professores que utilizam o ambiente. Todo o processo de armazenamento, consulta, remoção e edição de listas de exercícios é feita com o auxílio do ambiente, que, para isso, utiliza-se de componentes *JavaBeans*. No Servidor Web onde ficam armazenadas as listas de exercícios também residem *Servlets*.

Conforme foi dito anteriormente, aplicações J2ME podem se comunicar com *Servlets*, o que torna possível que estas aplicações acessem bancos de dados remotos na *World Wide Web*. No caso do ambiente descrito neste artigo, a aplicação J2ME correspondente ao Módulo do Aluno se comunica com *Servlets* que fazem acesso às listas de exercícios, o que permite que a aplicação J2ME desenvolvida tenha acesso aos dados destas listas. Estes *Servlets* também são responsáveis por corrigir e por armazenar os dados relativos à resolução dos alunos no Servidor Web, de tal maneira que os professores possam acessar essas resoluções em seus *desktops*, o que lhes permite avaliar o desempenho dos alunos. Dessa forma, o ambiente também permite que os professores possam fazer um acompanhamento e uma avaliação do desempenho dos alunos no processo de resolução dos exercícios. A Figura 6 exibe exemplos de telas dos Palmtops dos alunos acessando uma lista de exercícios, enquanto a Figura 7 mostra a tela que permite ao professor analisar o desempenho de um aluno na resolução de uma lista de exercícios.



Figura 6 – Exemplos de Telas dos alunos acessando os exercícios

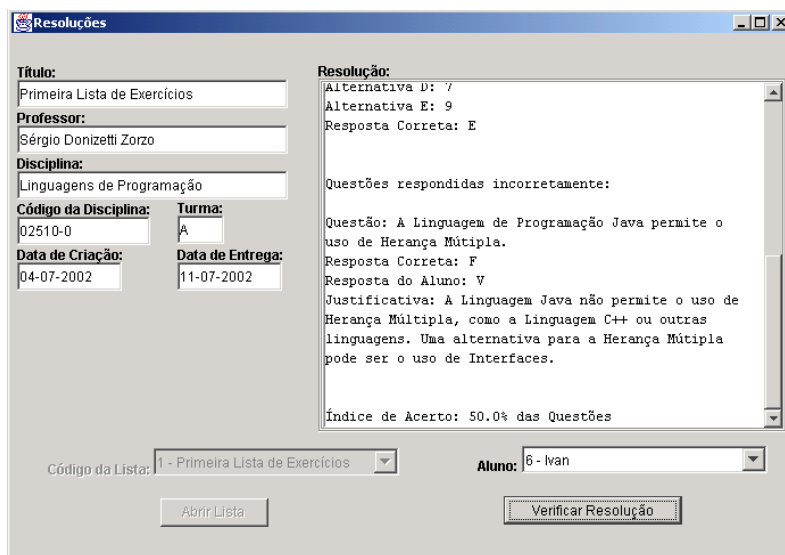


Figura 7 – Relatório do Desempenho de um Aluno exibido ao Professor

O Módulo do Aluno foi desenvolvido utilizando componentes de software para a tecnologia J2ME. Conforme foi dito, este módulo permite que os alunos acessem as listas de exercícios diretamente dos seus Palmtops, para resolvê-las. Este módulo comunica-se com o Servidor Web onde residem as listas de exercícios através de *Servlets*, o que também

permite que os dados das resoluções das listas fiquem armazenados neste Servidor. Conforme mencionado anteriormente, os *Servlets* também se encarregam de fazer a correção das listas de exercícios e de gerar um relatório de desempenho que será visualizado pelos alunos em seus Palmtops.

Este relatório traz o índice de acerto de um aluno na resolução de uma determinada lista de exercícios e também todas as questões respondidas correta e incorretamente pelo aluno, bem como as respostas corretas de cada questão e uma justificativa que explica porque uma determinada resposta é considerada correta. A Figura 8 exibe um exemplo de relatório visualizado por um aluno em seu Palmtop:



Figura 8 - Relatório do Desempenho do Aluno exibido em um Palmtop

Com isso, o ambiente auxilia o aluno no processo de aprendizado, uma vez que os estudantes não apenas respondem às questões, mas também podem verificar quais questões erraram e quais são as razões pelas quais a sua resposta foi considerada incorreta.

8 Conclusão

O ambiente descrito é composto por dois módulos distintos: um Módulo do Professor e um Módulo do Aluno. Enquanto o Módulo do Professor é executado em computadores de mesa convencionais, o Módulo do Aluno é direcionado para ser utilizado em Palmtops, e foi desenvolvido com a tecnologia Java 2 Micro Edition (J2ME) também apresentada neste artigo. Com isso, expande-se ainda mais as possibilidades oferecidas pela Educação a Distância (EAD), que tem como uma das suas principais características levar conhecimento para qualquer pessoa, independentemente do lugar onde esta se encontra.

Utilizando-se Palmtops, dispositivos pequenos e capazes de acessar dados da *World Wide Web* em ambientes sem fio, pessoas interessadas em acessar conteúdo didático poderão fazê-lo facilmente, sem a necessidade de estarem conectadas a redes de computadores convencionais.

Com a utilização da tecnologia J2ME, aplicações desenvolvidas para dispositivos dedicados adquirem qualidade comparável a aplicativos para *desktops*. Dessa forma, esta tecnologia resolve um dos maiores problemas encontrados por desenvolvedores de aplicações para dispositivos de tela pequena: tornar atrativas as suas aplicações, de tal maneira que seus usuários possam encontrar vantagens em utilizar aplicativos para dispositivos dedicados em detrimento de aplicações similares para *desktops*.

O processo de criação do ambiente é apoiado pelas técnicas do Desenvolvimento de Software Baseado em Componentes, tanto na aplicação desenvolvida para *desktops* quanto na aplicação para Palmtops. Detalhes relativos a esta técnica também foram expostos neste artigo. Os componentes desenvolvidos para *desktops*, baseados na tecnologia *JavaBeans*, são facilmente extensíveis e podem ser reutilizados por outras aplicações de EAD. Os componentes desenvolvidos para Palmtops baseiam-se na tecnologia J2ME, na qual recebem o nome de *Services*, e também são extensíveis e reutilizáveis.

Foi apresentada, ainda, uma descrição informal do ambiente descrito, visando especificá-lo claramente antes de elaborar-se uma arquitetura formal para este ambiente. A funcionalidade proporcionada pelo ambiente também foi descrita neste artigo. Baseado na descrição informal e na funcionalidade do ambiente, apresentou-se também a sua arquitetura. Esta arquitetura destacou os relacionamentos entre os diferentes módulos, os componentes de software utilizados por cada um destes módulos e o dispositivo no qual cada uma das partes da arquitetura é executada.

Por fim, expôs-se a interface dos Módulos do Aluno e do Professor, mostrando-se um exemplo do processo de criação de uma lista de exercícios com o auxílio do Módulo do Professor e a resolução desta lista por parte do aluno, através do Módulo do Aluno. Mostrou-se também a interface do ambiente que permite ao professor avaliar o desempenho dos alunos na resolução das listas de exercícios criadas.

Referências Bibliográficas

- [1] D'SOUZA, D. F., WILLS, A. C. - *Objects, Components and Frameworks with UML: the Catalysis Approach*, Addison-Wesley, 1998
- [2] DAY, B. - *Developing Applications Using de Java 2 Platform, Micro Edition*, 2001, disponível em <http://wireless.java.sun.com/getstart/articles>
- [3] GUIGERE, E. - *J2ME Tech Tips – Client-Server Communication Over HTTP Using MIDP and Servlets*, 2001, disponível em www.microjava.com/developer/resource_articles
- [4] JACOBSON, L., BOOCH, G., RUMBAUGH, J. - *The Unified Software Development Process*, Addison-Wesley, 1999
- [5] KEEGAN, D. - *Foundations of Distance Education*, 2ª ed. Londres: Routledge, 1991
- [6] MUCHOW, J. W. - *Core J2ME Technology and MIDP*, Prentice-Hall, 2002
- [7] OLIVEIRA, A. P. - *Apostila Servlet/JSP*, 2001. Disponível em: <http://www.dpi.ufv.br/~alcione/proginter2001/ApostilaServletJSP.pdf>
- [8] PAWLAN, J. - *World Of Wireless Communications*, 2000
Disponível em <http://wireless.java.sun.com/getstart/articles/hardware/>
- [9] PAWLAN, M. - *Introduction to Wireless Technologies*, 2000
Disponível em <http://wireless.java.sun.com/getstart/articles/intro/>
- [10] PRADO, A. F. - *Desenvolvimento de Software Baseado em Componentes (DBC)*, 2001. Disponível em: <http://www.recope.dc.ufscar.br/engenhariadesoftware/principal/>
- [11] QUINN, A. - *Trail: JavaBeans*, 2002. Disponível em: <http://java.sun.com/docs/books/tutorial/javabeans/>
- [12] SASAKI, C - *Building Compelling Services for the Wireless Market Using Java[tm] Technology*, 2001
Disponível em <http://wireless.java.sun.com/getstart/articles/whyjava/>
- [13] SUN MICROSYSTEMS - *How To Write Your First JES Service*, 2000
Disponível em <http://java.sun.com/j2me/docs/pdf/tutorialsol.pdf>
- [14] SUN MICROSYSTEMS - *Building Compelling Services for the Wireless Marketing Using Java Technology*, 2002
Disponível em <http://wireless.java.sun.com/getstart/articles/whyjava/>