

FACULDADE DE TECNOLOGIA SENAC PELOTAS
CURSO TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUTHIANO RODRIGUES VENECIAN

SISTEMA PARA GERENCIAMENTO DE COLABORADORES UTILIZANDO
DISPOSITIVOS MÓVEIS

PELOTAS

2007

FACULDADE DE TECNOLOGIA SENAC PELOTAS
CURSO TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LUTHIANO RODRIGUES VENECIAN

SISTEMA PARA GERENCIAMENTO DE COLABORADORES UTILIZANDO
DISPOSITIVOS MÓVEIS

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Faculdade de Tecnologia Senac Pelotas.

Orientador: Prof. Msc. Guilherme Tomansky Netto

PELOTAS

2007

LUTHIANO RODRIGUES VENECIAN

**SISTEMA PARA GERENCIAMENTO DE COLABORADORES UTILIZANDO
DISPOSITIVOS MÓVEIS**

Trabalho de Conclusão de Curso apresentado ao como requisito parcial à obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, pela Faculdade de Tecnologia Senac Pelotas.

Aprovado pela banca examinadora em 06 de dezembro de 2007.

BANCA EXAMINADORA:

Prof. Msc. Guilherme Tomansky Netto - Faculdade de Tecnologia Senac - RS

Prof. André Moraes - Faculdade de Tecnologia Senac - RS

Prof. Edécio Fernando Iepsen - Faculdade de Tecnologia Senac - RS

Dedico este trabalho em especial aos meus pais José Alberto Souza Venecian e Heloisa Rodrigues Venecian por serem os meus formadores e exemplos de vida, ao meu irmão Jean Rodrigues Venecian e aos demais familiares pelo carinho e compreensão.

AGRADECIMENTOS

Agradeço a Deus, autor da vida, princípio e fim. Obrigado por mais um passo segurado por Ti.

Ao professor Guilherme Tomansky Netto, meu orientador, por me orientar nesse trabalho.

Aos meus pais, por acreditarem na minha capacidade e na realização de seus sonhos em mim.

Ao meu irmão, que enxerguei seu crescimento e assumiu com bravura a minha ausência. E também a minha cunhada.

À minha família, quanta luta... Só Deus e vocês sabem do que passamos....

Ao seu Silvio e dona Soila, muito obrigado por tudo, principalmente pela confiança.

A todos os funcionários da Justiça Eleitoral de Pelotas, em especial a 164ª Zona Eleitoral e ao André Krolow, foi estagiando nesse cartório eleitoral que comecei a dar os primeiros passos na informática.

Ao Centro de Informática da UFPel, em especial ao diretor João Ladislau, que confiou e acreditou em mim, dando oportunidade de trabalho, onde dei os primeiros passos em programação.

A Comunidade Eclesial de Base São Lucas, da Paróquia São José do Fragata, onde cresci.

Aos meus amigos e amigas, infelizmente não posso nomeá-los, pode ficar gente de fora.

Ao Alfa Ômega, aprendi muito com todos, continuem a luta, não é fácil.

A Renovação Carismática Católica, da Diocese de Pelotas, à tia Isabel e a todos.

Ao Escritório Administrativo da RCC do Brasil, ao diretor Márcio Zolin e a todos, nessa construção dos projetos da obra de Deus.

Aos meus colegas da Faculdade Senac, em especial ao José Fernando e Josué, por toda ajuda e também aos meus ex-colegas do curso de Ciência da Computação da Universidade Católica de Pelotas.

E as pessoas que passaram na minha vida, que contribuíram no meu crescimento tanto pessoal quanto profissional.

RESUMO

O trabalho tem como finalidade o desenvolvimento de uma aplicação móvel para dispositivos celulares, utilizando a plataforma J2ME (Java 2 Micro Edition), que possibilitará aos colaboradores, do Centro Nacional de Formação da Renovação Carismática Católica, visualizarem o histórico das suas contribuições, dados cadastrais e informativos. Esse trabalho também aborda uma aplicação Web, que utiliza tecnologia PHP com banco de dados MySQL, cuja finalidade é o cadastramento e gerenciamento dos colaboradores e suas contribuições. A comunicação entre o dispositivo móvel e a aplicação Web é feita por uma conexão HTTP, acionada pela aplicação e suportada pela rede de dados do celular. Desse modo, é possível o emprego de aparelhos celulares com aplicações Java acessando base de dados no servidor Web, o que demonstra a versatilidade da arquitetura, que pode ser aplicada a outros sistemas.

Palavras-Chave: Dispositivos móveis, celulares, J2ME, PHP

ABSTRACT

The work aims to the development of a mobile application for wireless devices, using the platform J2ME (Java 2 Micro Edition), which will enable the staff of the, Centro Nacional de Formação from Renovação Carismática Católica, for Training view the history of their contributions, cadastral data and information. This work also raises a Web application, which uses technology PHP with the database Mysql, whose purpose is the registration and management of employees and their contributions. The communication between the mobile device and Web application is made by an HTTP connection, turned the application and supported by the network of data from the phone. Thus, it is possible the use of cell phones with Java applications accessing the database on the Web server, which shows the versatility of the architecture, which can be applied to other systems.

Key-words: Mobile Devices, cellular, J2ME, PHP.

LISTA DE ILUSTRAÇÕES

Figura 1 - Plataforma JAVA	22
Figura 2 – Plataforma J2ME	22
Figura 3 - Hierarquia de classes GCF (fonte: (MUCHOW, 2004))	27
Figura 4 - Classe de Interface do MIPD	29
Figura 5 - Cadastro de colaboradores	39
Figura 6 - Gerenciamento de colaboradores.....	39
Figura 7 - Gerenciamento de Estados	40
Figura 8 - Gerenciamento de lotes de contribuição	42
Figura 9 - Geração de listagens	43
Figura 10 - Boleto bancário gerado através do código do campo nosso número.....	44
Figura 11 - Modelo relacional	45
Figura 12 - Visão geral.....	48
Figura 13 - Visão do aplicativo móvel.....	50
Figura 14 - Identificação do colaborador	55
Figura 15 - Visualização dos dados cadastrais	55

LISTA DE TABELAS

Tabela 1 - Função das tabelas do banco de dados	46
---	----

LISTA DE CÓDIGO FONTES

Código Fonte 1 - Lógica em PHP para consulta de contribuições.....	48
Código Fonte 2 - Classe principal da apresentação – CNF.java.....	51

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface
CDC - Connected Device Configuration
CLDC - Connected Limited Device Configuration
CNF – Centro Nacional de Formação
GCF - Generic Connection Framework
HTML – Hypertext Markup Language
HTTP - HyperText Transfer Protocol
HTTPS - HyperText Transfer Protocol Security
JAR – Java Archive
JAD – Java Application Descriptor
J2EE - Java 2 Enterprise Edition
J2ME - Java 2 Micro Edition
J2SE - Java 2 Standard Edition
JVM – Java Virtual Machine
KVM – Kilo Virtual Machine
MID- Mobile Information Device
MIDP - Mobile Information Device Profile
PDA - Personal Digital Assistants
PHP - Hypertext Pre Processor
RCC – Renovação Carismática Católica
RMS - Record Management System
SGDB – Sistema Gerenciador de Banco de Dados
UDP - User Datagram Protocol
URL - Universal Resource Locator
WEB - World Wide Web
XML - Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO	14
1.1. OBJETIVOS	14
1.1.1. Objetivo Geral	15
1.1.2. Objetivos Específicos	15
1.2. JUSTIFICATIVA.....	15
1.3. VISÃO GERAL DOS CAPÍTULOS	16
2. REVISÃO BIBLIOGRÁFICA	17
2.1. VANTAGENS	17
2.2. PROBLEMAS.....	17
2.3. TECNOLOGIAS DISPONÍVEIS	19
2.4. METODOLOGIA	20
3. AMBIENTE.....	21
3.1. J2ME	21
3.1.1. Configuração	23
3.1.2. Perfil	24
3.1.3. Pacotes Opcionais.....	25
3.1.4. Máquina Virtual	25
3.2. CLDC	26
3.2.1. GCF	26
3.3. MIDP	27
3.3.1. Interface do usuário	28
3.3.2. Meio de armazenamento	29
3.3.3. Formato de Transmissão	30
3.4. COMUNICAÇÃO ENTRE DISPOSITIVO E O COMPUTADOR.....	32
3.4.1. HTTP	32
3.4.2. Sockets	33
3.4.3. Datagramas.....	33
4. INSTITUCIONAL E O DOMÍNIO DA APLICAÇÃO	34
4.1. CENTRO NACIONAL DE FORMAÇÃO	34
4.1.1. Objetivos	34
4.1.2. Metas.....	34
4.1.3. Ações	34
4.1.4. Viabilidade	35
4.1.5. Infra-Estrutura	35
4.1.6. Construção da sede do Centro Nacional de Formação	36
4.2. PROCESSO ATUAL DE CADASTRAMENTO	36
4.2.1. SAS	36
4.2.2. Tecnologias utilizadas	37
5. APRESENTAÇÃO DO SISTEMA	38
5.1. MÓDULO SERVIDOR.....	38
5.1.1. Colaboradores.....	38
5.1.2. Cadastro gerais	40
5.1.3. Contribuições	41
5.1.4. Usuários	42
5.1.5. Configurações.....	42
5.1.6. Relatórios	43

5.2. ESTRUTURA DO BANCO DE DADOS	45
5.3. MÓDULO MÓVEL	47
5.3.1. Arquitetura	47
5.3.2. Interface	50
5.3.3. Transmissão de Dados	56
5.4. TESTES	57
5.4.1. Testes de Unidade.....	57
5.4.2. Testes de Sistema	57
6. CONCLUSÃO E TRABALHOS FUTUROS	58
REFERÊNCIAS BIBLIOGRÁFICAS.....	60

1. INTRODUÇÃO

Devido ao crescimento na utilização de dispositivos móveis, em especial aos telefones celulares, cresce também a motivação no sentido de desenvolver novas aplicações para esses dispositivos. A presença da máquina virtual Java nesses equipamentos torna possível o desenvolvimento de aplicações utilizando a tecnologia *Java 2 Micro Edition* (J2ME), uma versão reduzida da linguagem Java utilizada para dispositivos móveis.

O desenvolvimento de aplicativos para celulares vem tornando-se cada vez mais freqüente nas empresas especializadas em produção de software. Contudo, esses sistemas não devem funcionar de maneira isolada. Entretanto, ignorar o fato de tais aparelhos serem inerentemente objetos de comunicação é subestimar o potencial deste artefato. Para tanto é necessário definir um protocolo de comunicação com sistemas corporativos, de forma que, possa ocorrer envio de informações entre as partes. Os celulares começaram a adquirir novas características, deixando de servir simplesmente para telefonia, ou seja, a mera transmissão de voz está perdendo o espaço para transmissão de dados.

Deste modo, este trabalho trata do desenvolvimento de um aplicativo móvel para telefones celulares que se comunica com uma base de dados remota. O trabalho também faz referência a um aplicativo servidor, onde são implementadas outras funcionalidades.

1.1. OBJETIVOS

O objetivo desse trabalho é a presente monografia, o desenvolvimento do projeto e a implementação do software, correspondente ao Trabalho de Conclusão de Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Faculdade de Tecnologia Senac Pelotas.

1.1.1. Objetivo Geral

O objetivo geral desse trabalho é o estudo da tecnologia Java para dispositivos móveis em telefones celulares, aplicado a uma solução para os colaboradores do Centro Nacional de Formação da Renovação Carismática Católica – RCC.

Através do celular, será possível aos colaboradores, obterem histórico de contribuições, consultas de seus respectivos dados cadastrais e visualização de informativos referentes ao Centro Nacional de Formação.

1.1.2. Objetivos Específicos

- Estudar a arquitetura da plataforma J2ME;
- Construir aplicações móveis usando a plataforma J2ME;
- Implementar operações de dados através de conexões entre aparelhos de celulares e servidor;
- Proporcionar portabilidade e boa performance a custos reduzidos utilizando ferramentas livres e linguagem Java.

1.2. JUSTIFICATIVA

A explosão da tecnologia móvel nos últimos anos abriu um universo de possibilidades para os desenvolvedores de aplicativos. Entretanto, apesar dos recursos estarem disponíveis, o mercado para aplicações móveis ainda não foi devidamente explorado. Além disso, a diversidade de plataformas a serem integradas, a preocupação com a performance e os custos de implementação e operação são desafios a serem considerados.

A tecnologia móvel usada no ambiente organizacional fornece agilidade, economia e inovação na integração entre a empresa e o seu cliente. Pode ser incorporada em diversos setores da empresa desde vendas e logística até relacionamento e marketing. Aplicações que

disponibilizam a vendedores acesso, a qualquer momento, à tabela de preços atualizada, disponibilidade e outras informações referentes a um produto, por meio de um aparelho celular, com possibilidades inclusive de efetuar o pedido do cliente em tempo real e *softwares* que possibilitam o acesso móvel à informações corporativas de uma empresa, são alguns exemplos de aplicações.

1.3. VISÃO GERAL DOS CAPÍTULOS

No capítulo 2, são apresentados a revisão bibliográfica sobre dispositivos móveis, suas vantagens, problemas e tecnologias disponíveis. No capítulo 3, são abordados conceitos fundamentais da plataforma J2ME, suas características e subdivisões.

Seguindo, no Capítulo 4, descreve-se a apresentação do local onde a solução será utilizada.

No capítulo 5, é apresentado o Módulo Servidor e o Módulo Móvel. O Módulo Móvel é responsável pela integração entre os dispositivos móveis e o banco de dados. No Módulo Servidor são abordados funcionalidades de cadastramento, gerenciamento de contribuintes e suas contribuições.

2. REVISÃO BIBLIOGRÁFICA

Para o desenvolvimento deste trabalho foram realizadas pesquisas e estudos sobre desenvolvimento de aplicações para dispositivos móveis. Esses estudos estão disponibilizados nas seções seguintes.

2.1. VANTAGENS

A maior vantagem do uso de aplicações móveis vem da própria definição de mobilidade. Num cenário onde os colaboradores das empresas precisam enfrentar viagens, engarrafamentos e deslocamentos constantes, a possibilidade de acessar relatórios de vendas, agenda de compromissos, entre outras informações relevantes ao desempenho de suas funções, a qualquer momento, é vital para o crescimento das organizações.

Segundo Leal (2004), o impacto produzido pelo uso de aplicações móveis está criando o conceito de Economia Livre, onde o acesso à informação passa a ser livre e independente da localização do usuário. O autor também destaca uma outra vantagem muito importante: não é preciso necessariamente criar aplicações novas, pode-se simplesmente adicionar módulos de mobilidade às soluções convencionais existentes.

Desta forma, o desenvolvedor pode projetar soluções móveis completamente integradas aos recursos já em operação na organização do cliente. Esta prática permite agregar valor às ferramentas pré-existentes evitando o descarte de tecnologia e os custos desta operação. O desenvolvedor ganha flexibilidade ao negociar novos projetos contornando a resistência natural dos clientes a mudanças drásticas.

2.2. PROBLEMAS

O maior problema enfrentado no desenvolvimento de aplicações móveis é a escassez de recursos dos equipamentos sem fio disponíveis atualmente no mercado.

Considerando os aparelhos celulares, pode-se, destacar as seguintes restrições e maneiras de contorná-las:

- Disponibilidade da rede: se o usuário de uma aplicação em tempo real estiver numa área sem cobertura este pode ser um problema. Entretanto nem todas as aplicações móveis necessitam estar on-line o tempo todo;

- Tamanho da tela: como a tela dos telefones celulares tem a resolução de 96 x 128 pixels, a interação do usuário com a aplicação móvel deve ser feita através de interfaces simples, com poucos elementos;

- Memória: as aplicações devem conter apenas as funcionalidades realmente indispensáveis uma vez que a quantidade de memória, tanto para execução como para armazenamento, é bastante limitada;

- Poder de processamento: diferentemente dos microcomputadores, os dispositivos móveis possuem poder de processamento reduzido e são normalmente alimentados por baterias. O processamento deverá acontecer, preferencialmente, no servidor e o aparelho celular receberá somente os resultados. Entretanto, esta técnica poderá gerar altos custos com transferência de dados.

Existe também o problema da diversidade de arquiteturas dos celulares, o que dificulta a portabilidade das aplicações móveis. Apesar dos fabricantes disponibilizarem a integração com algumas funcionalidades do dispositivo, o uso destes recursos será restrito a determinada arquitetura. Quanto mais específica for a aplicação menor será a gama de aparelhos que poderão suportá-la.

Além disso, a dificuldade de integrar diferentes plataformas a fim de obter o melhor proveito das aplicações móveis também deve ser apontada. A conectividade entre o ambiente de produção e os equipamentos sem fio torna-se um grande desafio uma vez que tanto a estrutura de tecnologia de informação como os dispositivos móveis terão arquiteturas diversas em organizações diferentes.

2.3. TECNOLOGIAS DISPONÍVEIS

Nesta seção, serão apresentadas algumas tecnologias disponíveis para a implementação de soluções móveis. É importante salientar que estas tecnologias não são excludentes podendo uma aplicação móvel utilizar-se de mais de uma delas.

- SMS: transmissão de pequenas mensagens de texto entre dispositivos tais como telefones celulares, aparelhos de fax e endereços IP.

- MMS: transmissão de mensagens cujo conteúdo poderá ser uma combinação de texto, som, imagem e vídeo, entre dispositivos que suportam tal tecnologia;

- Java 2 Micro Edition (J2ME): edição da linguagem Java para pequenos dispositivos como pagers, celulares, set-top boxes para TV a cabo, entre outros. Possui as funcionalidades básicas da linguagem Java adaptadas à escassez de recursos dos dispositivos.

Segundo Leal (2004) e Soares (2004) a tecnologia J2ME proporciona o melhor custo/benefício na medida que:

- Profissionais da linguagem Java podem ser aproveitados;
- Amplamente adotada pelos fabricantes e operadoras de telefonia móvel;
- Uma vez que a aplicação estiver instalada poderá ser utilizada fora da área de cobertura ou sem sinal, fazendo sincronização dos dados no retorno;
- Usuários podem personalizar seu celular escolhendo pacotes de aplicações oferecidos pelas operadoras de telefonia móvel ou por terceiros;
- Compatibilidade de plataformas: aplicativos Java são escritos uma vez e rodam em diferentes dispositivos com sistemas operacionais diferentes;
- Segurança: permite implementar criptografia extra para aplicativos de acesso a bancos, comércio, entre outros.

A utilização da tecnologia J2ME oferece liberdade e personalização do celular para o usuário e aumento da receita média por assinante nos serviços de dados para as operadoras. Para o desenvolvedor possibilita um desenvolvimento rápido para uma variedade de dispositivos e um mercado potencial enorme.

2.4. METODOLOGIA

A metodologia utilizada para o processo de desenvolvimento do sistema iniciou-se com estudos teóricos sobre programação para dispositivos móveis, plataforma J2ME e a suas conexões.

Após estes estudos foi realizado uma reunião, onde foi proposto a utilização de uma aplicação para dispositivos celulares aos colaboradores do Centro Nacional de Formação, onde também ficou definido suas funcionalidades. Essa aplicação deveria comunica-se com o sistema já utilizado pelo Centro Nacional de Formação.

Início-se então a etapa de codificação do sistema, inicialmente a aplicação Móvel se comunicava com uma aplicação Web bastante simplificada. Essa aplicação tinha como finalidade única, o acesso ao banco de dados. Esse processo serviu de testes para a aplicação Móvel como também de comunicação entre o dispositivo móvel e o servidor Web.

No decorrer do processo de desenvolvimento, percebeu-se dificuldades de comunicação da aplicação móvel com o sistema já existente. Esse sistema e seus dados são executados na rede interna do Centro Nacional de Formação, logo necessitaria de uma séria de medidas de segurança para o acesso externo, no qual, torno-se inviável.

Decidiu-se então o desenvolvimento do Módulo Servidor, onde foi acrescentado na aplicação Web, já existente para testes, as demais funcionalidades que compõem o módulo.

3. AMBIENTE

Até o surgimento da plataforma J2ME as aplicações tinham que ser escritas na linguagem nativa de cada dispositivo usando bibliotecas proprietárias, o que tornava estas aplicações incompatíveis com dispositivos diferentes.

Segundo (MUCHOW, 2004), a tecnologia J2ME trouxe flexibilidade e liberdade aos desenvolvedores quanto aos modelos de aparelhos e operadoras de telefonia móvel, e a tendência é que em breve torne-se padrão na indústria de dispositivos móveis. Nas seções seguintes, serão abordados a arquitetura do J2ME, bem como o processo de desenvolvimento de aplicações usando essa tecnologia.

3.1. J2ME

O J2ME consiste em uma coleção de APIs (*Application Programming Interface*) ou Interfaces de Programação de Aplicativos da linguagem Java, e a especificação de uma máquina virtual, que visa possibilitar a programação para dispositivos móveis com maior flexibilidade e portabilidade.

A intenção é que um programa seja feito uma única vez, em uma única linguagem, mas que possa ser executado em diversos dispositivos (MUCHOW, 2004). No entanto, levando-se em conta a enorme quantidade de aparelhos móveis existentes, onde cada um possui suas próprias características de processamento, quantidade de memória disponível, conectividade, consumo de energia, entre outros aspectos, o J2ME adota o mesmo conceito de máquina virtual visto nas plataformas Java *Standard* (J2SE) e *Enterprise* (J2EE), aumentando assim a portabilidade das aplicações.

A arquitetura da plataforma J2ME dentro do ambiente Java é mostrada na figura 1.

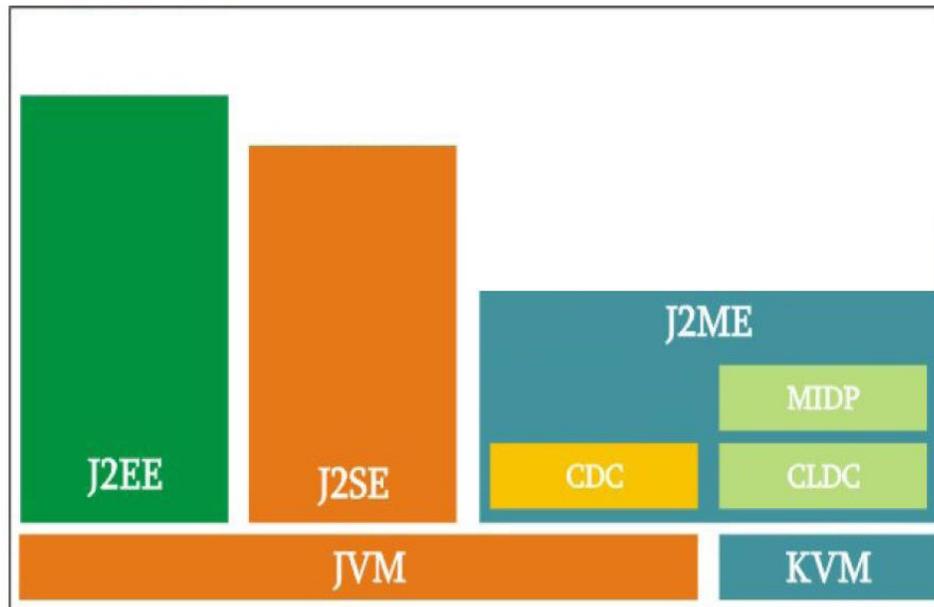


Figura 1 - Plataforma JAVA

Na arquitetura do J2ME a camada entre o sistema operacional e o hardware do dispositivo é a máquina virtual, que define quais limitações e recursos que o dispositivo atende. Assim para executar um aplicativo J2ME, um dispositivo deve conter uma implementação da máquina virtual definida à sua categoria de dispositivos.

A figura 2 expõe a arquitetura geral da plataforma J2ME.



Figura 2 – Plataforma J2ME

Para aumentar a eficiência da plataforma, acima da máquina virtual existem ainda os níveis de configuração e perfil e cada aparelho é encaixado em uma configuração e perfil, de acordo com o escopo de suas características. Porém, por não ser uma definição exata, alguns dispositivos podem ser incluídos em mais de uma categoria.

3.1.1. Configuração

Uma configuração está vinculada diretamente à máquina virtual Java possível de ser utilizada no dispositivo e define as funcionalidades e serviços que devem ser oferecidos por ela ao programador. As configurações definem as APIs básicas, baseadas nas características essenciais do dispositivo. Atualmente são definidas duas configurações:

Connected Device Configuration (CDC)

Configuração destinada a dispositivos que possuem maior poder computacional, ou seja, maior processamento, maior quantidade de memória, etc. Exemplos de dispositivos desta categoria são: set-top boxes para televisores, alguns PDAs, sistemas de navegação para automóveis, etc. A seguir algumas características típicas dos dispositivos dentro dessa configuração (MUCHOW, 2004):

- 512 kilobytes no mínimo de memória para execução do Java;
- 256 kilobytes pelo menos, para alocação de memória em tempo de execução.
- Conectividade de rede, largura de banda possivelmente persistente e alta.

Connected Limited Device Configuration (CLDC)

Designada para dispositivos menores, mais lentos e com maiores limitações, com capacidade de conexão em rede. Nessa categoria se encaixam aparelhos celulares, pagers e PDAs. Os requisitos de hardware para dispositivos dessa categoria (MUCHOW, 2004):

- 128 kilobytes de memória não-volátil (que preserva o conteúdo mesmo com o dispositivo desligado), para a máquina virtual e as bibliotecas CLDC;
- 32 kilobytes de memória volátil, disponível durante o tempo de execução da aplicação.

Mesmo sendo uma divisão clara, conforme a evolução tecnológica contínua venha oferecer maior possibilidades, com aumento de processamento, memória e outros recursos, a sobreposição entre as categorias irá se tornar maior, aumentando assim o significado e necessidade de existência dos perfis (MUCHOW, 2004).

3.1.2. Perfil

Um perfil pode ser tratado como uma extensão da configuração. Uma configuração pode conter vários perfis que por sua vez são especificados por apenas uma configuração. Assim, para aumentar a especificação dos dispositivos com relação às suas características, o J2ME adota esse conceito, onde um perfil é um conjunto de bibliotecas de mais alto nível, tratando de características mais específicas dos dispositivos (MUCHOW, 2004). Os perfis atualmente definidos são (CARDOSO, 2006):

- Perfis definidos para CLDC:

Mobile Information Device Profile (MIDP): que já conta com duas versões, e será detalhado num item 3.3.

Information Module Profile (IMP): é baseado no MIDP, é destinado a dispositivos com interface mais limitada.

- Perfis definidos para CDC:

Foundation Profile (FP): nível mais baixo de perfil para configuração CDC. Fornece implementação de rede e é utilizado principalmente para construção de aplicações sem uso de interface.

Personal Profile (PP): perfil CDC usado em dispositivos que precisam de suporte completo de interface. Usado em PDAs e consoles de jogos Suporte para compatibilidade total com o AWT e o modelo de aplicação applet.

Personal Basis Profile (PBP): é uma divisão do PP, fornece um ambiente para dispositivos conectados e com suporte básico de apresentação gráfica, suporte a modelo de aplicação xlet8 (é semelhante a applets, as aplicações são descarregadas dinamicamente e executadas).

A utilização de um perfil é opcional, porém é de extenso valor, pois fornecerá recursos para que sejam desenvolvidas aplicações para tipos de dispositivos específicos de forma customizada. Tratando de características relativas à espécie de dispositivo alvo, tornando o resultado da aplicação mais eficaz.

3.1.3. Pacotes Opcionais

Os pacotes opcionais são bibliotecas específicas para uma determinada tarefa que não esteja definida no perfil e na configuração para um determinado dispositivo, aumentando o poder da plataforma e facilitando a programação. Na plataforma J2ME pode-se existir a combinação de vários pacotes opcionais com as configurações CLDC, CDC e seus perfis, e assim atingir requisitos específicos de mercado. Esses pacotes opcionais fornecem APIs padrões para manipulação de tecnologias como *Bluetooth*, *Web Services*, *multimedia* e conexão com banco de dados (CARDOSO, 2006).

3.1.4. Máquina Virtual

Na plataforma J2ME as configurações definem os requisitos que a máquina virtual do dispositivo deve atender. Tendo em vista que todo aplicativo Java precisa ser executado em uma máquina virtual. Para os aplicativos pertencentes à configuração CDC a especificação da máquina virtual é a mesma do J2SE. Já na configuração CLDC, foi desenvolvida uma especificação de máquina virtual que considera as restrições de recursos dos dispositivos que se encaixam nesta categoria.

A implementação referêcia da maquina virtual criada pela *Sun Microsystems* chama-se KVM (maquina virtual K), mas não é necessariamente a única que pode estar executando em um dispositivo CLDC (MUCHOW, 2004).

3.2. CLDC

As configurações são definidas por meio de especificações que definem as funcionalidades da máquina virtual, as principais características dos dispositivos alvo, bibliotecas e modelos de segurança da plataforma. A CLDC é destinada para dispositivos pequenos e limitados ao nível de processamento, memória e interface com o utilizador. Neste grupo de dispositivos se encaixam os telefones celulares, *paggers*, PDAs e alguns eletrodomésticos.

O modelo de segurança CLDC serve para garantir que não ocorram deteriorações nos dispositivos ou corrupção de dados por meio de aplicativos Java, propositais ou acidentais, a CLDC define dois níveis de segurança, segurança de baixo nível, ou nível da máquina virtual, que trata a execução dos aplicativos Java, caucionando-os para que não possam danificar o dispositivo, e a segurança ao nível da aplicação (CARDOSO, 2006).

Além das classes derivadas do J2SE, a configuração CLDC define classes específicas, que constituem um *framework* genérico de conexão ou GCF (*Generic Connection Framework*).

3.2.1. GCF

A GCF ou estrutura de conexão genérica, é um subconjunto das APIs encontradas na plataforma J2SE, adequado às restrições encontradas em dispositivos móveis, com a finalidade de fornecer meios para manipular os tipos de conexões suportadas por estes dispositivos (MUCHOW, 2004). A idéia é que exista apenas uma classe que crie qualquer tipo de conexão (*http*, *socket*, arquivo e entrada por portas seriais).

A figura 3 mostra a hierarquia de classes do framework de conexão genérica.

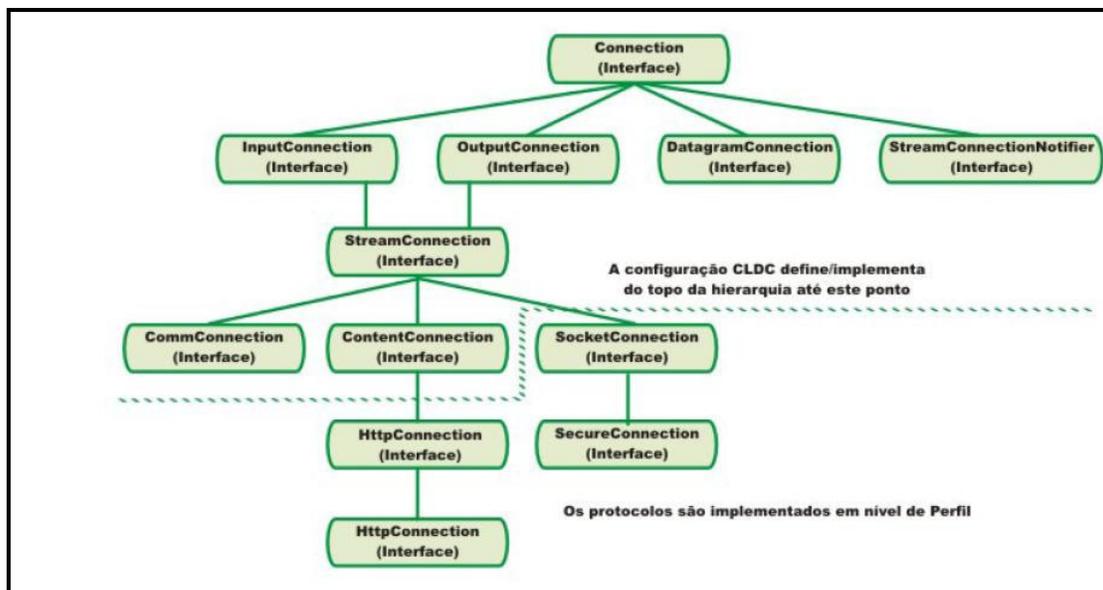


Figura 3 - Hierarquia de classes GCF (fonte: (MUCHOW, 2004))

Em tempo de execução a classe Connector instancia a classe específica do tipo de conexão requisitada. A implementação real do protocolo se dá em nível de perfil. A única implementação exigida pela configuração é o protocolo HTTP (*HyperText Transfer Protocol*), mas nada impede que a própria máquina virtual implemente outros protocolos.

3.3. MIDP

Alguns campos ficam fora das especificações das configurações como a interface com o usuário, tratamentos de eventos nas aplicações, entre outros. Estes artefatos são tratados em nível de perfil. O perfil MIDP foi especificado para dispositivos do tipo MID (*Mobile Information Device*). Ele também define quais os requisitos de *hardware* e *software* necessários para esses dispositivos (CARDOSO, 2006). O perfil fornece também algumas bibliotecas básicas.

Dentro das áreas de abrangência dessas bibliotecas são fornecidos recursos para funcionalidades básicas da configuração CLDC, definição e controle do ciclo de vida da aplicação, armazenamento persistente, interface gráfica, multimídia, comunicação e segurança.

Aplicativos feitos sobre o perfil MIDP são chamados MIDlets, um conjunto de MIDlets podem ser chamadas de MIDlet Suíte, quando estão armazenadas dentro de um mesmo arquivo JAR (Java *archive*) e assim também compartilhando os mesmos recursos. Os arquivos JAD (Java *Application Descriptor*) contém informações sobre as MIDlets contidas no arquivo JAR, são utilizados pelos dispositivos, e possibilitam que elas sejam instaladas nos aparelhos móveis (CARDOSO, 2006).

3.3.1. Interface do usuário

A interface do usuário do MIDP, Displayable, é formada por uma API de alto-nível e outra de baixo-nível. A API de baixo-nível é baseada no uso da classe abstrata Canvas. Já a API de alto-nível é formada pelas classes Alert, Form, List e TextBox, que são extensões da classe abstrata Screen (CARDOSO, 2006).

As classes da API de alto-nível fornecem abstrações e componentes que são altamente portáteis em diferentes MIDs, tais como: tipo de fonte, navegação, movimento de tela e alerta de erro. A implementação das classes de alto-nível em um certo dispositivo é “auto-ajustável” ao hardware e à interface de usuário nativa, não necessitando de configuração específica.

A classe Canvas da API de baixo-nível permite aplicações com um controle mais direto da interface de usuário. Ela permite um maior controle do que é desenhado no display e recebido do teclado. Diferentemente da classe Screen, é de responsabilidade do programador da aplicação certificar-se da portabilidade destas funções através de MIDs de diferentes características, tais como: tamanho do display, se o display é colorido ou preto-e-branco e tipo de teclado. A figura 4 mostra a relação entre as classes de interface de usuário do MIDP.

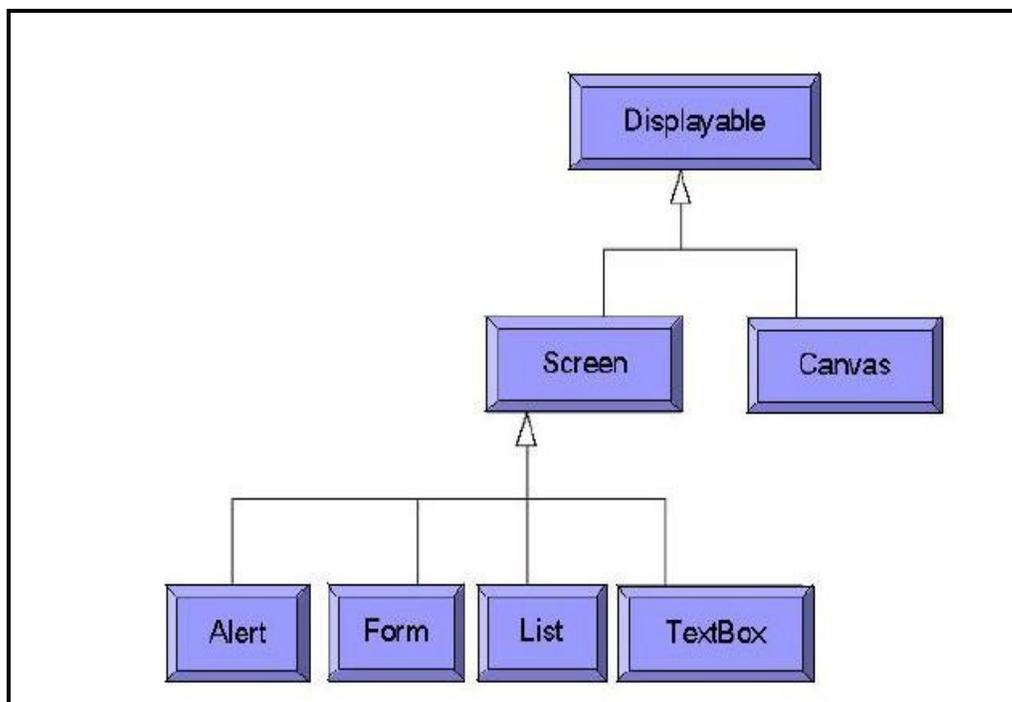


Figura 4 - Classe de Interface do MIPD

3.3.2. Meio de armazenamento

Um dos recursos mais significativos na computação é a possibilidade de gerenciamento, armazenamento e recuperação de dados. Em aplicativos que são executados em aparelhos móveis existem preocupações que devem ser consideradas: com relação ao tamanho, desempenho, suporte para sistemas de arquivos nos dispositivos, etc.

Uma alternativa para o armazenamento persistente dentro do perfil MIDP, é o RMS (*Record Management System*), um sistema de gerenciamento de registros, onde por meio de uma API de armazém de registros é possível se realizar a leitura, escrita, ordenação e pesquisa dos dados.

O RMS é um banco de dados orientado a registros, ele utiliza memória não volátil para armazenar as informações e pode ser abstraído como sendo um arquivo puro, onde uma série de fileiras são organizadas em forma de tabela, tendo um identificador para cada fileira, e um vetor de bytes.

Este identificador, é um número inteiro que atua como uma chave primária em um banco de dados e um vetor de bytes armazena os dados do registro. Um arquivo contendo vários registros é chamado de *record store* e é identificado por um nome (MUCHOW, 2004) (CARDOSO, 2006).

Os *record stores* devem ser únicos dentro de uma MIDlet Suíte. As *record stores* estão associadas às MIDlet Suítes e não às MIDlets. Em MIDP 2.0 existe também a possibilidade de partilhar *record stores* entre MIDlets pertencentes a MIDlet Suítes diferentes (CARDOSO, 2006).

3.3.3. Formato de Transmissão

A utilização conjunta das tecnologias Java e XML (*eXtensible Markup Language*) cria uma poderosa combinação entre portabilidade de código e portabilidade de dados. E apesar de algumas limitações, os dispositivos móveis que tem suporte para a plataforma J2ME podem fazer uso da linguagem de marcação XML em seus aplicativos.

Os *parsers XML* são programas que auxiliam a conversão de um arquivo, para que seja possível a manipulação dos dados que estão nele contidos. Descrevem as etapas para um XML ser processado.

Um parsing XML opera em três fases:

- processo de entrada XML: valida o documento de origem, reconhece e procura as informações relevantes baseada na localização ou em suas tags;
- lógica de negócio da informação: este é o estágio em que o processo real da informação de entrada ocorre. Pode resultar na geração da informação de saída;
- processo de saída XML: neste estágio a aplicação constrói o modelo do documento para ser gerado com o DOM (*Document Object Model*);

Existem três tipos fundamentais de *parser XML*, *model parser*, *push parser* e *pull parser*. O *model parser* faz a leitura das entradas do documento e cria representações desse documento na memória. É o modelo que usa maior quantidade de memória. No *push model* ocorre a leitura do documento inteiro. Enquanto encontra as partes do documento, notifica um objeto do ouvinte.

No modelo *pull parser* é feita leitura de uma parte do documento de cada vez. A aplicação dirige o parser através do documento repetidamente pedindo pela parte seguinte.

Em aplicativos J2ME (CARDOSO, 2006) é recomendável a utilização *parsers* do tipo *pull parser*. Devido às limitações de hardware dos dispositivos móveis, o grande consumo de recursos dos dois primeiros modelos, torna menos viáveis suas utilizações.

O uso de *parsers XML* em aplicativos MIDP tem reflexos nas suas performances. Como principais efeitos encontrados, tem-se o aumento considerável no tamanho do arquivo, uso intenso de *string parsing* para realizar o trabalho, que pode adicionar *overhead*, ocasionando assim um custo adicional no processamento ou armazenamento da aplicação que disponibilizar pouca memória, causando diminuição do tempo de resposta da aplicação.

3.4. COMUNICAÇÃO ENTRE DISPOSITIVO E O COMPUTADOR

A configuração CLDC define o GCF para que seja possível a comunicação em rede dos dispositivos móveis. A implementação dos protocolos ocorre em nível de perfil, e a classe Connector é uma fábrica de objetos do tipo Connection, uma interface genérica que simula uma conexão (MUCHOW, 2004).

Abaixo são expostos alguns tipos de conexões empregadas em dispositivos móveis com aplicativos da plataforma J2ME.

3.4.1. HTTP

Protocolo o HTTP e sua versão segura HTTPS são implementações obrigatórias em dispositivos pertencentes a categoria CLDC/MIDP. O HTTP é um protocolo baseado em pedidos e respostas entre o cliente e o servidor. Existem vários tipos de pedidos HTTP, mas o MIDP apenas obriga a que sejam implementados os métodos: GET, POST e HEAD (MUCHOW, 2004).

Um método GET faz um pedido ao servidor, onde o cliente solicita um recurso que está explícito na URL, e os parâmetros são passados pela própria URL. Para que o cliente envie informações para o servidor por meio do corpo do pedido, utiliza-se o método POST.

O método HEAD é semelhante ao método GET, difere apenas na resposta do servidor, onde a resposta não inclui o corpo da mensagem, e sim apenas os cabeçalhos, é usado geralmente para pedidos que não aguardam resposta.

A versão segura do protocolo, o HTTPS, possibilita que os dados sejam enviados encriptados, a diferença entre os dois é conexão utilizada e o protocolo do URL. O HTTPS também possibilita de obtenção de certificados de segurança das *Webginas* Web (CARDOSO, 2006).

3.4.2. Sockets

O socket é uma conexão de mais baixo nível que o protocolo HTTP, mas não é exigido pelo CLDC/MIDP, assim o dispositivo podem ou não conter a implementação de suas interfaces. Para obtermos um socket além de especificar um URL do gênero `socket://host:port`, no parâmetro do método `Connector.open()`, deve-se implementar a classe `SocketConnection`.

Além de conexões onde o dispositivo móvel é um cliente, existe também a possibilidade que este aceite conexões externas, tornando-o assim um servidor socket. Também, como no protocolo de comunicação HTTP, as conexões socket seguras são suportadas, são feitas pela definição do protocolo SSL. Essas conexões se assemelham com HTTPS e também permitem que se obtenha informações de segurança da conexão. De maneira geral funcionam como sockets normais (CARDOSO, 2006).

3.4.3. Datagramas

Conexões datagramas funcionam sobre o protocolo UDP (*User Datagram Protocol*), sem o estabelecimento de uma conexão lógica entre os pontos de ligação antes do envio dos dados (CARDOSO, 2006).

A informação enviada está inserida num datagrama que contém o endereço do destino. Como não existe o estabelecimento de uma conexão, a comunicação por UDP é normalmente mais rápida. Porém, o protocolo não garante a entrega dos datagramas, nem a ordem em que chegarão ao destino.

O protocolo UDP é utilizado normalmente quando não é muito importante garantir que toda a informação chega ao destino, mas é importante enviar a informação o mais rápido possível.

4. INSTITUCIONAL E O DOMÍNIO DA APLICAÇÃO

As sessões a seguir, abordam a estrutura do Centro Nacional de Formação.

4.1. CENTRO NACIONAL DE FORMAÇÃO

O Centro Nacional de Formação da Renovação Carismática Católica nasceu das novas necessidades para estes tempos e do desejo de levar a RCC a um novo patamar na evangelização. Quer ser um centro de excelência de formação, um espaço privilegiado para a formação de líderes, missionários, formadores, um centro de pesquisa e estudos sobre o fenômeno Carismático.

4.1.1. Objetivos

O objetivo é formar Líderes e Formadores em vista do ministério de edificação de comunidade eclesial em comunhão com a Igreja e sua doutrina, a partir da experiência da Renovação Carismática Católica em seus conteúdos próprios.

4.1.2. Metas

- Atrair e cadastrar sócios contribuintes para o Centro Nacional de Formação;
- Difundir o CNF;
- Organizar e cadastrar os Centros Regionais de Formação no Brasil.

4.1.3. Ações

- Criação do selo do Centro Nacional de Formação;

- Usar, para promover o CNF, todos meios e veículos de divulgação, como: Revista, Internet, programas de Rádio, TV e o contato com o povo nos eventos promovidos pela RCC em todo Brasil;
- Fazer que o programa de cadastro dos sócios contribuintes funcione com eficácia;
- Criar um meio de comunicação eficiente e personalizado entre a RCC e os sócios contribuintes.

4.1.4. Viabilidade

Uma vez cadastrado, o contribuinte passa a receber a Revista Renovação, com veiculação bimestral, um informativo ou carta tratando sobre o andamento do projeto e o boleto para contribuições.

O boleto e a revista não podem ser as únicas ferramentas motivadores para que os contribuintes mantenham-se ativos. A grande maioria dos carismáticos já contribuem com alguma obra evangelizadora.

O CNF pode, então, desde já começar o trabalho proposto a partir de formações por carta e on-line, através do envio de email. Posteriormente deverá estender suas ações através de material audiovisual e pela organização de vídeo conferências.

4.1.5. Infra-Estrutura

O Centro Nacional de Formação será criado com a missão de oferecer para a comunidade missionária, um Centro de Pesquisas Especializado e contará com uma moderna infra-estrutura para abrigar:

- Biblioteca carismática;
- Sala de informática;
- Auditórios;
- Espaços e oração;
- Espaços de convivência e realização de encontros;
- Centro de pesquisas e documentação carismática

4.1.6. Construção da sede do Centro Nacional de Formação

Objetivos:

- Implantar formações e capacitações periódicas no Centro Nacional de Formação;
- Tornar o CNF um referência de Vivência e Batismo no Espírito Santo;
- Transformar o Centro Nacional de Formação em um Centro de Educação para todos os níveis de aprendizagem, desde educação infantil ao Ensino Superior, Pós – Graduação, e afins, sem perder a identidade carismática.

4.2. PROCESSO ATUAL DE CADASTRAMENTO

Atualmente os colaboradores se utilizam de dois meios para se cadastrarem:

- Internet - <http://www.rccbrasil.org.br/cnf>
- Formulário cadastral manual

No caso dos formulários manuais, eles são cadastrados em um sistema por digitadores. Como esses dados são escritos manualmente, muitas vezes ocorrem inconsistência nos dados, letra ilegível, o que acaba trazendo gastos, retorno da correspondência, além disso, muitas vezes é necessário a ligação telefônica para a confirmação dos dados.

4.2.1. SAS

Atualmente o Centro Nacional de Formação gerencia seus sócios contribuintes através de um sistema chamado SAS (Sistema de Associados).

O sistema visa controlar toda a parte de cadastramento de colaboradores, assim como a geração de boletos bancários e recebimento de contribuições via débito bancário. Essas são atualmente as duas formas de contribuição dos colaboradores do Centro Nacional de Formação.

4.2.2. Tecnologias utilizadas

O desenvolvimento do SAS foi realizado utilizando-se da linguagem Delphi. O que permite que o aplicativo utilize apenas plataforma Windows, no qual, isso torna-se desvantajoso devido a licença paga a esse sistema operacional. Devido a esse motivo, o servidor de aplicação utiliza a versão Windows 2003 Server.

Para o armazenamento dos dados é utilizado o sistema gerenciador de banco de dados Sql Server 2005, cujo também é licenciado.

5. APRESENTAÇÃO DO SISTEMA

O sistema proposto refere-se a uma aplicação móvel para telefones celulares, que visa oferecer aos colaboradores do Centro Nacional de Formação uma opção de acesso aos seus dados através de dispositivos móveis.

O sistema está dividido no Módulo Servidor e no Módulo Móvel que serão abordados nas próximas seções.

5.1. MÓDULO SERVIDOR

O Módulo Servidor corresponde ao processo de cadastramento e gerenciamento de colaboradores e suas contribuições. Para tanto, foi desenvolvido um sistema Web, utilizando a linguagem PHP¹ em conjunto do bando de dados MySQL².

A aplicação apresenta as seguintes funcionalidades:

5.1.1. Colaboradores

Essa funcionalidade é responsável pelo cadastramento de colaboradores - Figura 5, gerenciamento dos dados cadastrais e o controle de suas contribuições.

Para o gerenciamento - Figura 6 encontra-se no topo da tela, uma opção de filtros de consulta, o que possibilita um melhor controle dos colaboradores. Pelo gerenciamento é possível também, realizar alteração dos dados cadastrais, exclusão do colaborador, consulta de contribuições e o controle da situação cadastral, utilizado para inativar cadastros.

¹ PHP (*Hypertext Preprocessor*). Linguagem voltada para o desenvolvimento *web* desenvolvida para trabalhar em conformidade com a estrutura cliente-servidor.

² Sistema gerenciador de banco de dados, é o mais usado em função do seu licenciamento GPL.

Centro Nacional de Formação - RCC Brasil - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost/cnf/areas/cadastro/index.php

Colaboradores | Cadastro gerais | Contribuições | Usuários | Configurações | Relatórios SAIR

Colaboradores
Cadastrar
Gerenciar

DADOS PESSOAIS

Nome:

CPF:

Data de nascimento: XXXXX/XXXX

Endereço:

Complemento:

Bairro:

País: Brasil Outros países Estado: Seleccione o estado Cidade: Primeiro seleccione o estado

Diocese: Primeiro seleccione a cidade

Telefone Residencial:

Telefone Celular:

Email:

Função na RCC:

RG:

Sexo: Masculino Feminino

Número:

CEP: XXXXX-XXX

Telefone Comercial:

CONTRIBUIÇÃO

Valor:

Concluído Intranet local

Figura 5 - Cadastro de colaboradores

Centro Nacional de Formação - RCC Brasil - Microsoft Internet Explorer

Arquivo Editar Exibir Favoritos Ferramentas Ajuda

Endereço http://localhost/cnf/areas/cadastro/gerenciar.php

Colaboradores | Cadastro gerais | Contribuições | Usuários | Configurações | Relatórios SAIR

Colaboradores
Cadastrar
Gerenciar

Contribuições | Gerenciar

Busca País/Estado: Seleccione o estado Situação: Seleccione o país Código:

Palavra-chave:

Listar Descrição crescente

Código	Nome	Estado/Cidade	
000002	Luthiano Rodrigues Venecian	RS Pelotas	Contribuição Alterar Excluir

1

Intranet local

Figura 6 - Gerenciamento de colaboradores

5.1.2. Cadastro gerais

Nessa funcionalidade são realizadas as inclusões e a manutenções dos dados que compõe os sub-campos cadastrais. Esses sub-campos atuam diretamente no processo de cadastramento e atualização dos dados dos colaboradores.

- **Função:** Identifica a função do colaborador dentro da organização;
- **País:** Destinado para a inclusão e gerenciamento de países;
- **Estados:** Utilizado para o cadastramento e gerenciamento de estados de um determinado país. Na figura 7, é demonstrado a tela para gerenciamento de estados, processo semelhante utilizado nos demais sub-campos;
- **Cidades:** Destinado para a inclusão e gerenciamento de cidades de um determinado estado;
- **Dioceses:** Assim como nas cidades, uma diocese é relacionada com um estado.

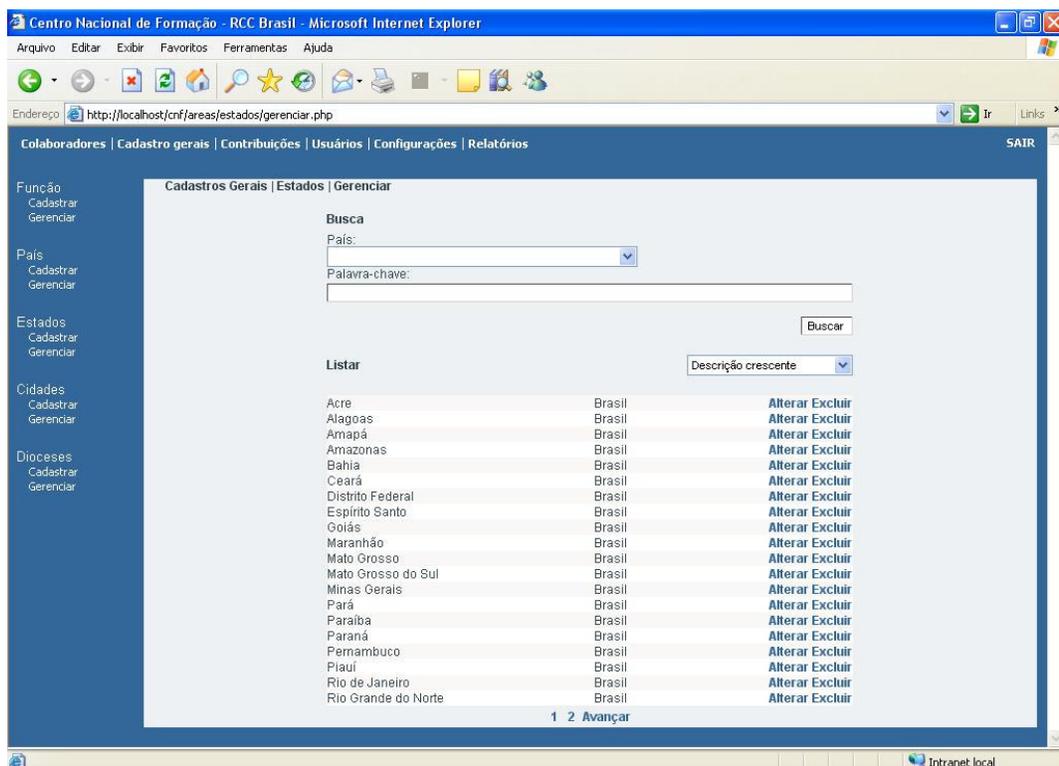


Figura 7 - Gerenciamento de Estados

5.1.3. Contribuições

Nessa seção são apresentadas as funcionalidades, destinadas para a geração de contribuição, importação do arquivo retorno e consulta do recebimento de contribuições dos colaboradores.

Arquivo retorno:

Destinado à importação do arquivo de retorno do banco conveniado. É por esse processo que é obtido os colaboradores que contribuíram, assim como, a data que foi realizado a contribuição e o valor respectivo da sua contribuição.

Gerar Contribuição:

O processo de geração de contribuição é bastante simplificado e de fundamental necessidade para o gerenciamento das contribuições. Por meio dele é gerado lotes de contribuição. Esses lotes são destinados por período e data de vencimento. Como cada colaborador estipula o valor de sua contribuição, o mesmo é obtido pelo relacionamento entre o lote e os colaboradores “ativos”. Concluindo o processo, é informado na tela o número de registros gerados no lote, o intervalo do código nosso numero do boleto bancário, data de vencimento e período. Para a impressão dos boletos bancários, basta somente informar o intervalo do nosso número inicial e final do boleto bancário gerado no lote.

Gerenciar e consultar:

Na figura 8, representa a forma que de gerenciamento dos lotes de contribuições. Onde é possível obter pelo lote, o período que foi gerado, a data de vencimento, quantidade registros (colaboradores) e o valor parcial recebido. Por fim, na opção consultar possibilita um filtro de consultas, semelhante ao gerenciamento de coladores na funcionalidade “Colaboradores”, para que possam ser realizadas buscas da situação das contribuições dos colaboradores.

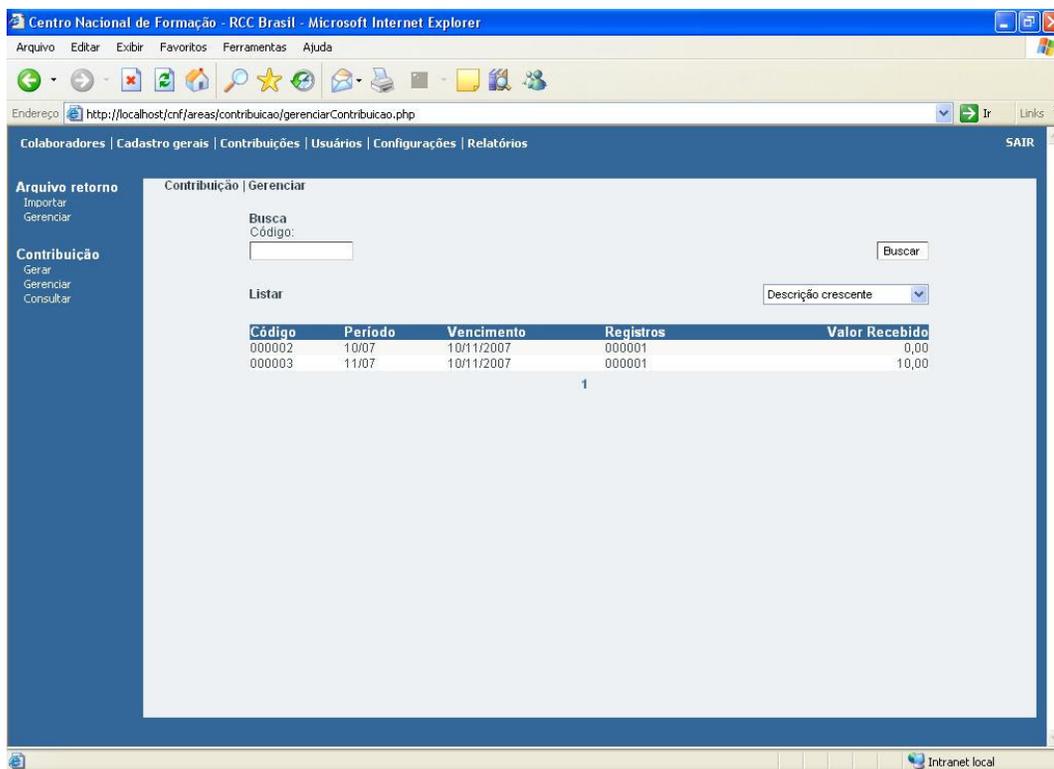


Figura 8 - Gerenciamento de lotes de contribuição

5.1.4. Usuários

Essa funcionalidade possibilita a inclusão, alteração e exclusão de usuários que terão acesso ao Módulo Servidor. Existem dois níveis de acesso:

- Administrador: Nesse nível o usuário possui acesso total;
- Usuário: Possui apenas acesso ao cadastramento de novos colaboradores e alteração dos dados dos colaboradores, por ele cadastrado.

5.1.5. Configurações

- Backup: Possibilita ao administrador do Módulo a geração de backup do banco de dados. Essa ferramenta é muito importante, tendo em vista que é uma maneira para prevenção de eventual ataque de hackers ao banco de dados.

5.1.6. Relatórios

Funcionalidade responsável pelo gerenciamento dos colaboradores, seja para geração de etiquetas, impressão de boletos bancários, bem como, listagens de colaboradores. Abaixo, segue as descrições dessas funcionalidades:

- **Listagens:** Por meio de um filtro, é possível a geração de listagens - Figura 9, de colaboradores por: período de cadastro, estado, cidade, diocese, situação, usuários que cadastraram;
- **Etiquetas de Endereço:** Utiliza os mesmos filtros da listagem e gera etiquetas para o envio de correspondências aos colaboradores;
- **Geração de Boleto Bancário:** Figura 10 - por essa opção são gerados os boletos bancários. Na tela de geração, é requisitado o código do campo nosso número do boleto bancário. Esse que está relacionando com o código do colaborador e com o respectivo lote de contribuição.

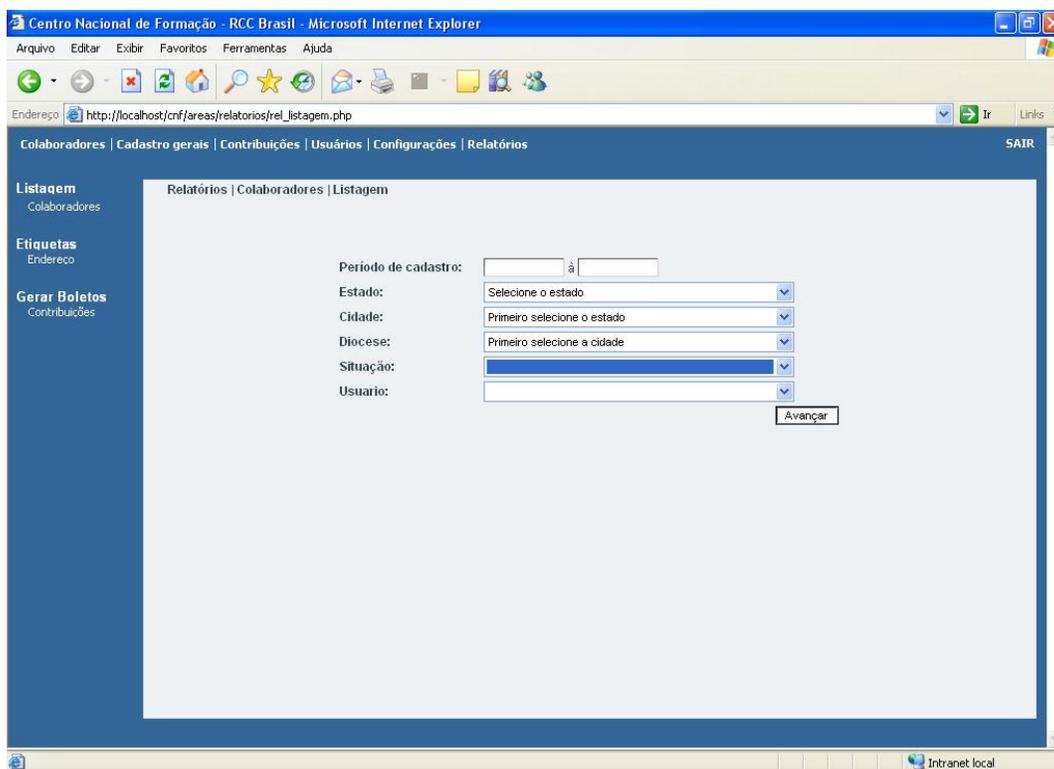


Figura 9 - Geração de listagens

					
<p>Estimado(a) Luthiano Rodrigues Venecian</p> <p>Fundamentados na esperança de salvação em Nosso Senhor Jesus Cristo e motivados pela palavra profética: "assumi os vossos postos e ampliai a vossa visão", estamos empenhados no trabalho de melhor servir ao Reino de Deus. Para tanto, contamos com sua fidelidade em contribuir com o Escritório Nacional da RCC.</p> <p>Desde já, agradecemos sua generosidade e rogamos ao Pai em Jesus Cristo, copiosas bênçãos sobre você e toda sua diocese.</p> <p>Fraternalmente,</p> <p>Escritório Nacional da RCC</p>					
<p>CAIXA 104-0 10491.04571 20990.000000 00000.000018 7 36860000001000</p>					
Cedente		Agência/Código do Cedente		Espécie	Quantidade
Esc. Adm. Ren. Caris. Católica do Brasil		0495 / 045720		R\$	
Número do documento		CPF/CNPJ	Vencimento		Nosso número
000001		000665299/0001-01	10/11/2007		990000000000000001-8
Valor documento		10,00			
(-) Desconto / Abatimentos	(-) Outras deduções	(-) Mora / Multa	(+) Outros acréscimos		(=) Valor cobrado
Sacado					
Luthiano Rodrigues Venecian					
Demonstrativo					
Contribuição					
RCC - Brasil					
Diocese de Pelotas / RS					
Corte na linha pontilhada					
<p>CAIXA 104-0 10491.04571 20990.000000 00000.000018 7 36860000001000</p>					
Local de pagamento				Vencimento	
CASAS LOTÉRICAS, AGÊNCIAS CAIXA E REDE BANCÁRIA				10/11/2007	
Cedente				Agência/Código cedente	
Esc. Adm. Ren. Caris. Católica do Brasil				0495 / 045720	
Data do documento	Nº documento	Espécie doc.	Acerte	Data processamento	Nosso número
13/11/2007	000001	OU	N	13/11/2007	990000000000000001-8

Figura 10 - Boleto bancário gerado através do código do campo nosso número

5.2. ESTRUTURA DO BANCO DE DADOS

Para o desenvolvimento do sistema foi necessário criar um banco de dados com o intuito de armazenar as informações dos colaboradores e suas contribuições. A figura 11 apresenta o modelo relacional do BD, apresentando todas as suas tabelas e relacionamentos.

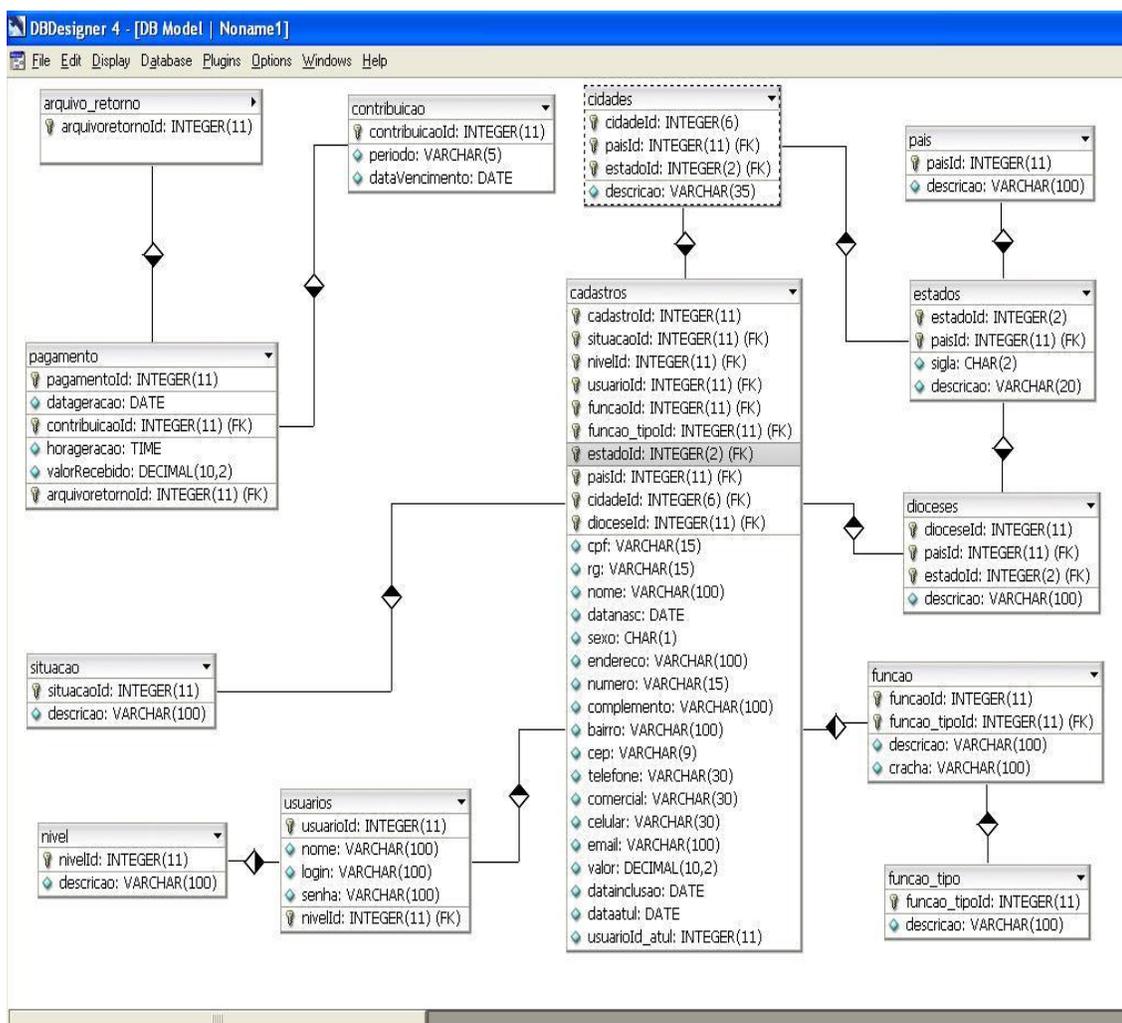


Figura 11 - Modelo relacional

O banco de dados do sistema possui 13 tabelas. A Tabela 1 abaixo, especifica a função de cada tabela deste banco de dados para o sistema.

Tabela 1 - Função das tabelas do banco de dados

Tabela	Função
cadastros	Contém as informações dos colaboradores, dados cadastrais, valor da contribuição e situação cadastral.
pais	Possui as informações dos países
estado	Possui as informações dos estados
cidade	Possui as informações das cidades
diocese	Possui as informações das dioceses
funcao	Contém as funções dos membros da organização
funcao_tipo	Responsável por caracterizar os níveis de função
usuarios	Contém as informações dos usuários e o seu nível de operação.
nivel	Responsável por caracterizar os níveis de usuários
situacao	Contém as opções de situação cadastral dos colaboradores
contribuicao	Possui as informações do lote de contribuição
pagamento	Responsável por guardar as informações do código do colaborador, lote e os registros de data de recebimento e valor recebido do arquivo de retorno
arquivo_retorno	Armazena as informações do arquivo gerado pelo banco

5.3. MÓDULO MÓVEL

O sistema para dispositivos móveis desenvolvido, propõe a atuar diretamente na consulta de contribuições, dados cadastrais e informativos para os colaboradores do Centro Nacional de Formação da RCC. A visualização dos dados será disponibilizada por meio de uma interface amigável e adequada às restrições do dispositivo.

Para construção do software foi adotado a plataforma de desenvolvimento J2ME, por ser uma tecnologia voltada especificamente para a criação de aplicações embarcadas, que, entre outras vantagens, possibilita a portabilidade desejável a esta categoria de equipamentos, sendo possível de ser executado nos mais diversos tipos de aparelhos compactos. Apresenta ainda programação orientada à objetos, o que aumenta o nível de abstração, modularidade e usabilidade do programador.

Dentro das configurações disponíveis pela plataforma J2ME, foi escolhido a utilização da configuração CLDC, que define um ambiente de execução, ou seja, uma implementação da máquina virtual necessária para execução do aplicativo.

5.3.1. Arquitetura

A arquitetura definida para concretização do projeto, compreendendo todo o ciclo de coleta de dados cadastrais dos colaboradores e suas contribuições na mesma base de dados, é demonstrada na figura 12, onde, de modo geral, os dispositivos móveis, por meio de seus respectivos meios de acesso à Internet, fazem pedidos HTTP, os quais são tratados pelos módulos PHP contidos no servidor Web, que fazem a manipulação necessária na base de dados para atendê-lo.

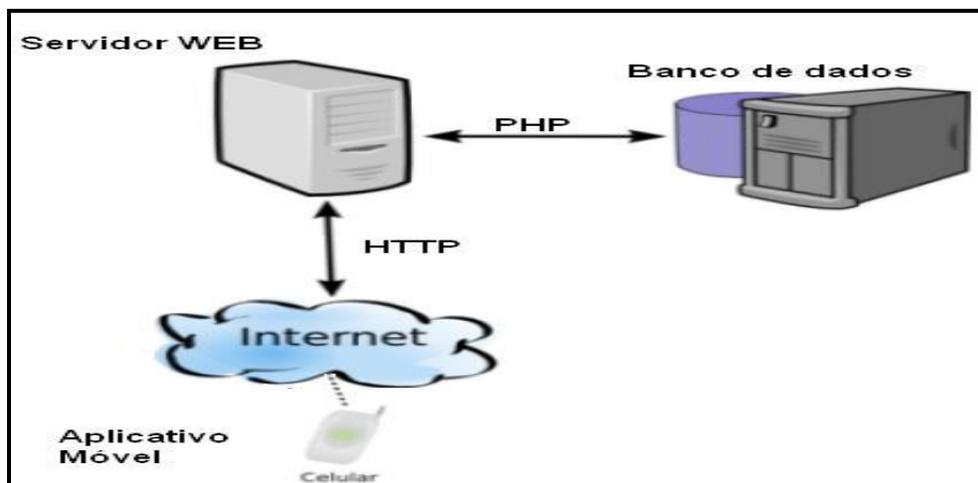


Figura 12 - Visão geral

Para possibilitar a integração entre o Módulo Servidor e o Módulo Celular, no computador servidor, ao invés de fazer-se o uso de módulos PHP já criados para todo o sistema, adotou-se a proposta de criação de módulos específicos para tratarem solicitações do Módulo Móvel. Esta solução foi escolhida devido ao fato de que módulos previamente criados tratavam especificamente de pedidos de página HTML, ou seja, para construção e visualização em Web Browsers, o que tornaria a construção dos pedidos de transmissão e interpretação das respostas do aplicativo móvel muito complexa e aumentando o custo computacional.

Deste modo, foram implementados arquivos em linguagem PHP, que teriam a finalidade de receber as informações vindas no corpo do pedido executado pelo móvel, e fazer a manipulação necessária no banco de dados, para atendê-lo. A seguir no Código Fonte 1 é mostrado um trecho do código implementado em PHP.

Código Fonte 1: Lógica em PHP para consulta de contribuições

```
<? php
    include (“../classes/conecta.php”);
// Recebimento das variáveis
    $status = $_POST['status'];
    $codigo = $_POST['codigo'];
    $cpf    = $_POST['cpf'];
```

```
$buscaContribuinte = mysql_query("select * from cadastros where cadastroId=$codigo and
cpf='$cpf'");
```

```
//Verifica a existência do colaboradores
```

```
if (mysql_fetch_row($buscaContribuinte) and $status=='contribuicao')
```

```
{
```

```
    //Busca por contribuição para o colaborador
```

```
    $buscaContribuicao = mysql_query("select A.valorRecebido, B.datainclusao from
pagamento A, arquivo_retorno B where A.arquivoretornoId=B.arquivoretornoId and
A.cadastroId=$codigo and A.valorRecebido<>'0.00'");
```

```
    //Verifica a existência de contribuições
```

```
    if (mysql_fetch_row($buscaContribuicao))
```

```
    {
```

```
        while($row=mysql_fetch_array($buscaContribuicao))
```

```
        {
```

```
            echo $row[datainclusao] . " - " . $row[valorRecebido] . "<br>";
```

```
        }
```

```
    }
```

```
    else
```

```
    {
```

```
        echo "Nenhuma contribuição encontrada!";
```

```
    }
```

```
}
```

```
?>
```

A lógica contida nos arquivos PHP criados que tratam de pedidos advindos de dispositivos móvel, fazem controle de sessão pelo envio do código e CPF enviados no corpo do pedido HTTP. A autenticação do usuário deve ser validada no banco, assim o servidor pode fazer a verificação de validade da ação solicitada.

5.3.2. Interface

Na figura 13, são mostradas duas telas do Sistema Móvel, mostrando a interface do sistema visto no software emulador empregado no desenvolvimento da aplicação.



Figura 13 - Visão do aplicativo móvel

Para a construção da camada de apresentação foi criada uma classe principal, denominada CNF, derivada da classe MIDlet, fornecida pela implementação da API MIDP 2.0, que faz a referência necessária para o *display* do equipamento, trechos da codificação da classe são mostrados no Código Fonte 2.

Código Fonte 2: Classe principal da apresentação – CNF.java

```

public class CNF extends MIDlet implements CommandListener {

//referência ao display do dispositivo
    private Display display = null;
// Formulário principal, comandos...
    private Form formMaim = null;
    private Image image = null;
    private Alert alerta = null;
    private Command commExit, commContribuicao,
        commDadosCadastrais, commInformativos, commHelp;
//Criação do objeto da classe ModelFormulario , derivada da classe Stack, que faz herança da
//classe Form
    private ModelFormulario modelFormulario = null;

//método construtor
public CNF(){
    display = Display.getDisplay (this);
    formMaim = new Form( "Centro Nacional de Formação da RCC-BR" );
    formMaim.addCommand (commExit = new Command ("Sair", Command.EXIT, 0));
    formMaim.addCommand (commContribuicao = new Command ("Contribuições",
Command.OK, 1) );
    formMaim.addCommand (commHelp = new Command ( "Help",Command.HELP, 1) );
    try{
        //cria a imagem
        image = Image.createImage ( "/imagem/Inicial.png" );
    }catch (Exception e) { }
    formMaim.append (new ImageItem (null, image, ImageItem.LAYOUT_CENTER, null));
    formMaim.setCommandListener (this);
    modelFormulario = ModelFormulario.instancia (display, formMaim);
}

```

//Chamando do gerenciador de aplicação para o MIDlet

```
public void startApp() {
    display.setCurrent(formMaim);
}
```

//método exigido

```
public void pauseApp() { }
```

//método exigido

```
public void destroyApp(boolean unconditional) { }
```

//implementação da interface CommandListener

```
public void commandAction(Command c, Displayable s){
```

```
    //Caso seja o comando SAIR
```

```
    if (c == commExit){
        destroyApp(false);
        notifyDestroyed();
    }
```

```
    //Caso seja o comando CONTRIBUIÇÃO
```

```
    else if (c == commContribuicao){
        modelFormulario.pushForm (new ComunicacaoServidorWeb("CNF - Controle de
Contribuição", this,1));
    }
```

```
    //Caso seja o comando HELP
```

```
    else if (c == commHelp) alerta (lerArquivoHelp(), formMaim);
    }
```

//método para leitura de arquivo texto do help

```
private String lerArquivoHelp (){
    try{
        int chr = 0;
        InputStream arquivoHelp = getClass().getResourceAsStream("/ajuda/help.txt");
        StringBuffer stb = new StringBuffer();
        while ((chr = arquivoHelp.read())!= -1)
            stb.append((char)chr);
        return stb.toString();
    }catch (Exception e){
        alerta("Erro ao abrir o arquivo de ajuda!", formMaim);
    }    return null;
}
```

//Para a exibição de alertas

```
public void alerta (String mensagem, Form frm){
    try{
        Image img = Image.createImage("/imagem/Duke.png");
        //cria o alerta
        alerta = new Alert("Centro Nacional de Formação da RCC-BR", mensagem, img,null);
        //configura o alerta com o tipo Modal
        alerta.setTimeout(Alert.FOREVER);
        //exibe o alerta. Quando dispensado exibe o formulário
        display.setCurrent(alerta, frm);
    }catch (Exception e){ }
}

}
```

O usuário do sistema móvel pode navegar pelo aplicativo, tendo como base uma tela principal de opções, nessa Interface será possível obter:

Contribuições: Através dessa opção será possível ao usuário obter o histórico de suas contribuições. Para isso é necessário sua identificação - Figura 14, que ocorrerá pelo seu Código e seu CPF, em seguida será listado na tela do dispositivo a relação de contribuições realizadas.

Dados cadastrais: é possível a visualização dos dados cadastrais do respectivo colaborador Figura 15, se faz necessário sua identificação. O que possibilitará, manter-se informado sobre seus dados cadastrais ou eventuais erros de digitação ocorrido no processo de cadastramento de colaboradores realizado no Módulo Servidor.

Informativos: Não é necessário identificação. Por meio dessa opção será possível o colaborador obter notícias do Centro Nacional de Formação.

Help: Exibe como navegar no aplicativo móvel, através de um detalhamento das funcionalidades da aplicação.

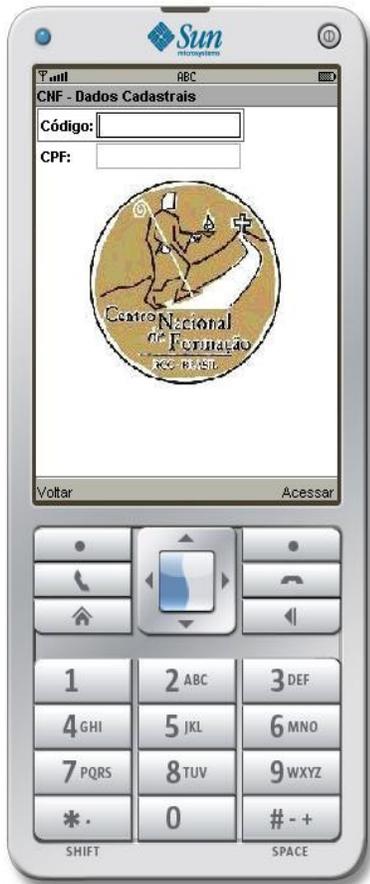


Figura 14 - Identificação do colaborador



Figura 15 - Visualização dos dados cadastrais

5.3.3. Transmissão de Dados

Outro aspecto fundamental para o desenvolvimento do projeto é o envio e recebimento de informações via a Internet. A implementação do módulo referente à transmissão dos dados será realizada por meio da API GCF, definida pelo CLDC/MIDP.

Graças à flexibilidade da plataforma, para conectar o dispositivo à Internet, no endereço de um servidor Web responsável por responder as requisições do Módulo Móvel, pode ser utilizado qualquer meio de conexão disponível. Desta forma, todas as possibilidades de conexão sem fio e com fio podem ser utilizadas. Tornando ainda mais robusta a solução utilizada.

Dentre os protocolos de conexão possíveis de serem realizadas em um dispositivo portátil, utilizando a GCF, optou-se pela comunicação por meio do protocolo HTTP, por ser o protocolo vigente para comunicação entre navegador Web e Servidor Web, além de ser o único exigido pela especificação do MIDP.

Inicialmente, a transmissão dos dados seria realizada utilizando o formato de arquivos XML. Porém, o uso de parsers XML em aplicativos MIDP, tem reflexos nas suas performances, efeitos como aumento no tamanho do arquivo e overheading, podem ser encontrados ocasionando assim um custo adicional no processamento ou armazenamento da aplicação.

Contudo, durante os estudos, observou-se a não necessidade de transmissão de um arquivo, e optou-se por outra solução, uma vez que o protocolo HTTP disponibiliza métodos que possibilitam o envio direto de informações ao servidor. Assim, os dados armazenados no dispositivo, são enviados, realizando-se o estabelecimento de uma conexão HTTP e a informação é enviada no corpo do pedido pelo método POST.

5.4. TESTES

Durante a fase de testes do projeto foram realizados testes de unidade e sistema. Os testes de unidade, realizados na aplicação móvel, verificam a funcionalidade interna de cada método das classes e o relacionamento entre eles. Os testes de sistema verificam o desempenho e a aplicação como um todo, incluindo a aplicação Móvel e o acesso aos demais recursos utilizados no Módulo Servidor.

5.4.1. Testes de Unidade

Os testes de unidade realizados no simulador da aplicação Móvel demonstraram que as classes apresentavam a funcionalidade esperada. As funcionalidades sobre contribuições, dados cadastrais, informativos retornaram os dados corretamente. As operações no Módulo Servidor também corresponderam aos resultados esperados.

5.4.2. Testes de Sistema

Os testes de sistema realizados no simulador demonstraram que a aplicação consegue atingir os objetivos a que se propõe. O tempo de resposta da conexão com o Módulo Servidor no ambiente simulado de testes é aceitável, variando entre 1 e 3 segundos.

O desempenho da aplicação no ambiente de testes, no Módulo Servidor como no Módulo Móvel, foi satisfatório. A aplicação Móvel conseguiu comunicar-se com o Módulo Servidor e efetuar operações no banco de dados do Servidor.

6. CONCLUSÃO E TRABALHOS FUTUROS

Com o desenvolvimento deste trabalho foi possível obter uma conclusão satisfatória nos objetivos propostos. Foi analisado no simulador o correto funcionamento de todos os métodos empregados na interface do menu de navegação do aplicativo, como também o envio e recebimento de dados do servidor destino.

Uma das vantagens do aplicativo é a sua independência do meio de comunicação que o dispositivo tem disponível para o envio dos dados. Este fato aumenta ainda mais a gama de aparelhos possíveis de serem utilizados.

Outra característica importante é a integração da plataforma J2ME com o sistema Web, que utiliza tecnologia PHP, o que demonstra a versatilidade da arquitetura, que pode ser utilizada a outros sistemas.

Dentre os fatores negativos encontrados no projeto, podem ser citados, a menor segurança em termos de autenticação do usuário e a forma de envio de dados realizado de maneira simples pelo método POST. Devido à criação de módulos adicionais no sistema Web, para atender especificamente a pedidos vindos do aplicativo móvel, a política de segurança previamente criada para o sistema Web precisa ser re-elaborada.

Apesar das dificuldades encontradas como pouca informação prática sobre implementação para dispositivos móveis, a área de desenvolvimento de aplicações Java para aparelho celular tende a crescer bastante nos próximos anos.

Sobretudo, o aplicativo móvel possibilitará novos benefícios aos colaboradores do Centro Nacional de Formação, permitindo uma velocidade de informação muito mais rápida que o acesso a um site ou até mesmo o contato telefônico.

Trabalhos futuros

Como trabalho futuro poderia ser adicionado novas funcionalidades ao dispositivo móvel. Atualmente o usuário do aplicativo móvel realiza consultas de seus dados cadastrais. Contudo, podem ser acrescentados novos requisitos, como a alteração de dados cadastrais pelo próprio usuário no dispositivo celular.

Uma outra forma de melhoria é tornar a interface gráfica mais rica de detalhes, com a utilização de componentes otimizados que herdam da classe Canvas. Tornando assim, as telas

do aplicativo móvel, personalizadas e possuindo um design muito mais bonito, com detalhes atrativos e possibilidade de animações gráficas.

REFERÊNCIAS BIBLIOGRÁFICAS

CARDOSO, J. *Java para Tele-móveis MIDP 2.0*. 2006. Disponível em:
<<http://livromidp.jorgecardoso.eu/html>>. Acesso em: novembro de 2007.

GHOSH, S. *Add XML parsing to your J2ME applications*. 2003. Disponível em:
<<http://www-128.ibm.com/developerworks/library/wi-parsexml/>>. Acesso em:
novembro de 2007

LEAL, Maurício. *Começando com Java Wirelles*. 18 ed. Java Magazine, Rio de Janeiro, 2004. Primeiros Passos, p.16-19.

MICROSYSTEMS, S. *Java Platform, Micro Edition (Java ME)*. 2006. Disponível em:
<<http://java.sun.com/javame/>>. Acesso em: junho 2007.

MUCHOW, J. W. *Core J2ME Tecnologia e MIDP*. 1. ed. São Paulo, SP: Sun Microsystems, Inc, 2004. (Sun Microsystems Press Java series).

PINHEIRO, C. *J2ME - Java para os portáteis*. 2003. Disponível em:
<<http://www.imasters.com.br/artigo/1539>>. Acesso em: junho 2007.

SOARES, Marilson. *A Tecnologia Java como Diferencial no Mundo dos Celulares*. 5 ed. Mundo Java. Curitiba, maio 2004. Foco, p.53-54.

SOUZA, Bruno. *Fazendo Wireless Acontecer: J2ME e o Mercado Brasileiro*. 3 ed. Java Magazine, Rio de Janeiro, 2003. A vida, o Universo e Tudo Mais, p. 12-15.