

Cleimar José Salvador
Diego da Silva de Medeiros
Fabício Emydio da Silva
Patrick Hedlund Rosa

Transmissão de Seqüências de Imagens
Através da Rede Celular GSM/EDGE

São José – SC
setembro / 2008

Cleimar José Salvador
Diego da Silva de Medeiros
Fabrcio Emydio da Silva
Patrick Hedlund Rosa

Transmissão de Seqüências de Imagens ***Através da Rede Celular GSM/EDGE***

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Centro Federal de Educação Tecnológica de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador:
Prof. Marcos Moecke

Orientador:
Prof. Mario de Noronha Neto

**CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE
TELECOMUNICAÇÕES
CENTRO FEDERAL DE EDUCACAO TECNOLOGICA DE SANTA CATARINA**

São José – SC
setembro / 2008

Monografia sob o título “Transmissão de Seqüências de Imagens Através da Rede Celular GSM/EDGE”, defendida por Cleimar José Salvador; Diego da Silva de Medeiros; Fabrício Emydio da Silva e Patrick Hedlund Rosa e aprovada no dia 11 de setembro de 2008, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Marcos Moecke
Orientador

Prof. Mario de Noronha Neto
Orientador

Prof. Emerson Ribeiro de Mello
CEFET / SC

Prof. Rubem Toledo Bérnago
CEFET / SC

*Sempre que te perguntarem se podes fazer um trabalho, responde
que sim e te ponhas em seguida a aprender como se faz.*

F. Roosevelt

Agradecimentos

Agradecemos, primeiramente, aos que conviveram conosco no Laboratório de Projeto Final durante a execução deste projeto. Com certeza, essas pessoas tornaram a realização deste trabalho mais fácil.

Também agradecemos aos nossos familiares. Eles estiveram sempre presentes na nossa vida, confiando, nos apoiando e incentivando a estudar e vencer desafios.

Nossos sinceros agradecimentos a todos os colegas que estudaram conosco, e em especial à Nathalie Cardone, que desde o início do Curso nos acompanha dando força para seguir sempre em frente.

Por último aos nossos Professores, que com maestria nos forneceram a luz do conhecimento, bem como motivação para traçar e alcançar objetivos. Em especial, aos orientadores, pelo empenho e tempo extra dedicado.

Resumo

Este trabalho apresenta um procedimento para transmitir imagens de um computador para um aparelho celular utilizando a rede celular GSM/EDGE. Para o envio das imagens, é utilizado um módulo com suporte a tecnologia EDGE e um computador com câmera de vídeo. No lado do receptor é usado um telefone celular com EDGE e Java para a exibição das imagens no formato JPEG. A comunicação entre o telefone celular e o módulo é estabelecida por uma conexão TCP. Devido à limitação na taxa de transmissão da tecnologia EDGE, a transmissão das imagens foi realizada em baixa resolução e baixa taxa de quadros, sendo porém suficientes para a visualização de imagens em tempo real

Palavras chave: GSM, EDGE, SMS, Java Micro Edition, transmissão de imagem, JPEG.

Abstract

This work presents a system which transmits images in real time from a camera to a mobile phone based on GSM/EDGE cellular network. To accomplish the transmission an EDGE wireless module and a computer connected to camera are used as transmitter. A mobile phone with EDGE and Java technology is used as receiver. In order to simplify the system JPEG format is used to transmit the images. A TCP connection between the mobile phone and EDGE module is used to transmit the data. Due to EDGE technology transmission rate restrictions images are send in low resolution and frame rates. To conclude a cost analysis and some possible improvements are presented.

Keywords: GSM, EDGE, SMS, Java Micro Edition, image transmission, JPEG.

Sumário

| | |
|---|-----------|
| Agradecimentos | 5 |
| Lista de Figuras | 10 |
| Lista de Tabelas | 11 |
| 1 Introdução | 12 |
| 2 Fundamentação Teórica..... | 14 |
| 2.1 Compressão de Imagem em Formato JPEG..... | 15 |
| 2.2 Protocolo TCP/IP | 19 |
| 2.3 Rede GSM com GPRS e EDGE..... | 23 |
| 2.4 Mensagens SMS e Protocolo PDU..... | 24 |
| 2.5 Módulo MC75 e Terminal T75 EDGE-BR..... | 28 |
| 2.6 Comandos AT | 30 |
| 2.7 A Tecnologia Java..... | 31 |
| 3 Desenvolvimento e Implementação do Sistema | 38 |
| 3.1 Materiais Utilizados no Projeto..... | 39 |
| 3.2 Sistema CaTICIS - Captura e Transmissão de Imagem entre dois Computadores via Interface Serial | 40 |
| 3.3 Sistema RICS (Recepção de Imagem no Celular de um Servidor HTTP)..... | 46 |
| 3.4 Sistema EDGE2JAVA – Captura e Transmissão de Imagem entre um Computador e Unidade Móvel via Rede GSM/EDGE..... | 49 |
| 4 Conclusões e Comentários Finais..... | 56 |
| 4.1 Custo do uso do sistema..... | 57 |

| | | |
|-----|--|-----------|
| 4.2 | Taxa de Transmissão | 58 |
| 4.3 | Implementações Alternativas | 59 |
| 4.4 | Aplicações do Sistema Desenvolvido | 59 |
| 4.5 | Trabalhos Futuros..... | 60 |
| | Referências Bibliográficas | 61 |
| | Lista de Abreviaturas e Siglas..... | 64 |

Lista de Figuras

| | |
|--|----|
| Figura 1 – Diferença do componente contínuo (DIFF) e seqüência de codificação em zigzag no JPEG..... | 17 |
| Figura 2 – Predição do valor do <i>pixel</i> no JPEG sem perdas. | 17 |
| Figura 3 – Quadro PDU. | 25 |
| Figura 4 – Terminal T75 EDGE-BR e Módulo MC75. | 29 |
| Figura 5 – Subdivisões da Plataforma Java..... | 32 |
| Figura 6 – Camadas da JME. | 33 |
| Figura 7 – Ciclo de vida de um aplicativo MIDlet. | 35 |
| Figura 8 – Ambiente de desenvolvimento NetBeans..... | 37 |
| Figura 9 – Emulador Sun Java Wireless Tool Kit. | 37 |
| Figura 10 – Interligação dos equipamentos no sistema CaTICIS..... | 40 |
| Figura 11 – Fluxograma de funcionamento do sistema CaTICIS..... | 41 |
| Figura 12 – Processo de codificação e decodificação JPEG utilizado..... | 42 |
| Figura 13 – Quadro de transmissão de imagens no sistema CaTICIS..... | 44 |
| Figura 14 – Interligação dos equipamentos no Sistema RICS..... | 46 |
| Figura 15 – Fluxograma de funcionamento do Sistema RICS..... | 47 |
| Figura 16 – Interligação dos equipamentos no Sistema EDGE2JAVA..... | 49 |
| Figura 17 – Fluxograma de funcionamento do Sistema EDGE2JAVA..... | 52 |
| Figura 18 – Divisão em blocos da imagem codificada em JPEG para transmissão. | 53 |
| Figura 19 – Codificação HCA..... | 54 |

Lista de Tabelas

| | |
|---|----|
| Tabela 1 – Camadas do modelo OSI..... | 20 |
| Tabela 2 – Comparação entre o modelo OSI e as 4 camadas do TCP/IP | 20 |
| Tabela 3 – Tipos de serviços e protocolos das camadas TCP/IP | 21 |
| Tabela 4 – Função e sintaxe dos comandos AT | 30 |
| Tabela 5 – Custo do envio de mensagens SMS | 57 |
| Tabela 6 – Custo da transmissão de dados..... | 58 |

1 Introdução

“Mr. Watson – come here – I want to see you.” (Bruce, 1990). Com esta frase, dita a seu ajudante em 10 de março de 1876 por Alexander Graham Bell através da sua última invenção, o telefone, inaugurava-se o mundo das telecomunicações. De lá para cá, o universo das telecomunicações passou por diversas transformações, das quais se pode destacar a utilização da transmissão sem fio por pioneiros como o padre Landell de Moura. Tudo isso contribuiu para o surgimento da telefonia móvel celular, que nas últimas décadas teve uma verdadeira explosão de uso. Esta explosão sofrida deve-se principalmente ao conforto e facilidade de uso, bem como outras características e vantagens que apresenta em relação à telefonia fixa. Estima-se que em agosto de 2008 existam no Brasil 133,15 milhões celulares ativados e no mundo 1,28 bilhões (JORNAL CIDADE, 2008). Os sistemas celulares são classificados, de acordo com o padrão tecnológico, em geração 2.5G, tendo como exemplos as tecnologias GPRS, EDGE e CDMA2000 1×RTT; e geração 3G, usando tecnologias UMTS e CDMA2000 1×EVDO (SVERZUT, 2005).

Devido à taxa de transmissão de dados que estes sistemas proporcionam, principalmente os de terceira geração, a transmissão de vídeo e imagem em tempo-real através da rede celular está surgindo como uma possibilidade, resultando em uma crescente demanda por aplicações envolvendo a transmissão destes dados, como por exemplo, o monitoramento por imagens e videoconferência.

Visando estes tipos de aplicações, este trabalho apresenta uma implementação de um sistema capaz de enviar imagens de uma câmera conectada a um computador para uma unidade móvel usando a rede GSM/EDGE. O sistema utiliza um módulo de transmissão sem fio conectado a um computador como fonte da informação e aparelhos celulares como destino, ambos com suporte a tecnologia GPRS/EDGE. Para o estabelecimento da comunicação entre a unidade móvel e o módulo terminal foi utilizada uma conexão TCP. Em função da simplicidade do sistema e do baixo custo de transmissão, espera-se que o sistema desenvolvido possa ser empregado em diversas aplicações.

A apresentação deste documento está organizada de modo que o leitor possa entender o

trabalho, sem a necessidade de recorrer a fontes secundárias de informação. No Capítulo 2 são introduzidos os conceitos básicos necessários dos sistemas e da tecnologia de programação utilizados neste trabalho. No Capítulo 3 é apresentada a proposta do trabalho através de três sistemas, sendo os dois primeiros sistemas construídos apenas com o objetivo de testar a viabilidade da proposta, bem como permitir a validação de cada bloco constituinte do sistema final. O primeiro sistema realiza a transmissão de imagens capturadas por uma webcam entre dois computadores. O segundo sistema permite a visualização em um aparelho celular de imagens depositadas previamente em um repositório HTTP. O Sistema EDGE2JAVA, que é o resultado final deste projeto, utiliza componentes dos dois primeiros sistemas, transmitindo as imagens capturadas por uma webcam para um aparelho celular, que as recebe e apresenta em tempo real. O Capítulo 4 apresenta algumas comparações entre os resultados obtidos e os esperados, soluções alternativas, possíveis aplicações para o sistema desenvolvido, bem como temas para trabalhos futuros. Finalizando o trabalho, no Capítulo 5 são apresentadas as principais conclusões. No anexo são apresentados os códigos fontes dos programas dos aplicativos implementados para a realização dos sistemas

2 Fundamentação Teórica

Este capítulo descreve de forma sucinta as tecnologias aplicadas a este trabalho. A Seção 2.1 explicará o formato de codificação de imagem JPEG. Na Seção 2.2 será apresentado o modelo de transmissão de dados tipo cliente-servidor via protocolo TCP/IP. A Seção 2.3 traz informações sobre os sistemas de redes celulares atuais e tecnologias de transmissão de dados. Na Seção 2.4 é descrito o processo de elaboração de mensagens SMS em formato PDU. A Seção 2.5 mostra as características do módulo MC75 e do terminal T75 EDGE-BR. Na Seção 2.6 é apresentada a estrutura básica dos comandos AT usados. Finalmente, na Seção 2.7 são apresentadas as principais características da tecnologia Java.

2.1 Compressão de Imagem em Formato JPEG

Para a transmissão da imagem, foi escolhido um formato padrão de codificação de imagem que permitisse uma decodificação simples no aparelho celular. Como o objetivo do trabalho é a transmissão em baixas taxas de quadros por segundo e o principal foco não é o desenvolvimento de algoritmos de compressão, foi adotado o JPEG, que é um padrão consolidado com uma boa taxa de compressão para imagens estáticas.

Para o melhor entendimento do processo de transmissão, algumas noções sobre a imagem digital são apresentadas nas seções que seguem, começando com uma explicação sobre o mapa de bits que representa uma imagem sem compressão. Este formato será usado como padrão de comparação entre imagens sem compressão e imagens comprimidas.

2.1.1 Mapa de Bits (*Bitmap*)

O mapa de bits (*bitmap* – BMP) é um formato bastante utilizado quando se deseja obter a imagem sem os ruídos causados por compressão. Nele, cada *pixel* é formado pela combinação de três componentes de cor, vermelho (*red* – R), verde (*green* – G) e azul (*blue* – B), e a imagem é constituída em um conjunto de *pixels* correspondentes às linhas e colunas da resolução espacial total. Cada componente de cor é codificado em 8 bits, podendo variar de zero (para ausência total da cor) até 255 (para a intensidade máxima). A cor preta é obtida quando todos os componentes RGB possuem valor zero, e a cor branca quando todos os componentes possuem valor 255.

Por não possuir nenhum método de compressão o tamanho final de uma imagem *bitmap* é extenso. Por exemplo, uma imagem *bitmap* padrão colorida codificada com 8 bits para cada cor, com resolução espacial de 1024 por 768 *pixels* e 54 bytes de cabeçalho, resulta em um arquivo com 2,25 MB, pois $(1024 \times 768 \times 3 \times 8) + (54 \times 8) = 18.874.800$ bits.

2.1.2 JPEG

Com a popularização das redes internet, a crescente necessidade do compartilhamento de arquivos de imagens e fotos exigiu um padrão para a compressão de imagens que trouxesse uma alta taxa de compressão com uma grande fidelidade à imagem *bitmap* original. Em 1986

o grupo de especialistas em fotografia (*Joint Photographic Experts Group* – JPEG) elaborou o padrão JPEG, que foi aprovado em 1992 pela Organização Internacional para Padronização (*International Organization for Standardization* - ISO) (ISO; ITU-T, 1992).

O padrão JPEG se adapta bem a imagens com muitas variações de cores e ricas em detalhes, tais como as que ocorrem em fotografias reais. O nível de compressão é ajustável permitindo estabelecer um compromisso entre o tamanho do arquivo e a qualidade da imagem. Uma redução de até dez vezes do tamanho original de um arquivo pode ser obtida sem grandes perdas na qualidade da imagem (ISO; ITU-T, 1992).

No JPEG o processo de codificação e decodificação pode ser realizado sem perdas (*lossless*) ou com perdas (*lossy*). Para obter a compressão sem perdas, é utilizada uma predição espacial, que permite a recuperação integral da imagem na exibição; e uma codificação de entropia, que é um algoritmo de compressão também sem perdas. No caso da codificação com perdas, a alta taxa de compressão é obtida a partir do uso da transformada discreta de cosseno (*discrete cosine transform* – DCT), em conjunto da codificação de entropia.

O processo de codificação baseado na DCT é iniciado com a conversão da imagem do formato RGB para o YCbCr. O formato YCbCr separa a imagem em um componente de luminância Y, que representa a intensidade de luz da imagem; e dois componentes de crominância Cb e Cr, que indicam respectivamente o desvio de cor para o azul e para o vermelho. Como o olho humano é menos sensível a variações de cor que de luz, é possível aplicar uma subamostragem nos componentes Cb e Cr, chamada de *chroma sub sampling*, sem que haja muita perda na qualidade da imagem.

Cada componente da imagem YCbCr é dividido em blocos de 8 X 8 *pixels* e em seguida transformado em coeficientes freqüenciais pela DCT. Os coeficientes obtidos são quantizados com uma tabela de quantização pré-definida, que leva em conta as diferenças de sensibilidade a freqüências do olho humano. O padrão JPEG não define a tabela de quantização a ser utilizada, podendo ser diferente em aplicações diversas.

Após esse processo a diferença entre o primeiro termo (componente contínuo) do bloco atual e do bloco anterior é computada, sendo denotada por DIFF. Os demais coeficientes são codificados em seqüência ziguezague conforme mostra a Figura 1. Desta forma, a matriz de 8 linhas por 8 colunas se transforma em um vetor com 64 posições. Os 64 símbolos obtidos são codificados através de um codificador de entropia, que utiliza a probabilidade de ocorrência de cada símbolo para determinar um novo código para a amostra. Símbolos mais freqüentes

recebem códigos menores, permitindo uma compressão dos dados sem perdas. No JPEG, dois tipos de codificação podem ser utilizados: a codificação Huffman e a codificação aritmética (ISO; ITU-T, 1992).

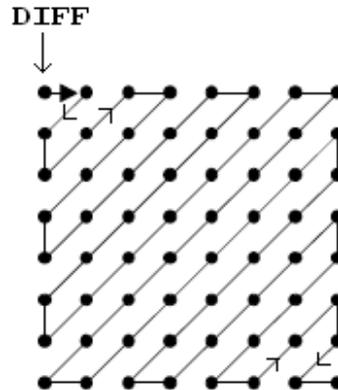


Figura 1 – Diferença do componente contínuo (DIFF) e seqüência de codificação em ziguezague no JPEG.

No caso da codificação sem perdas, valores de três *pixels* vizinhos *a*, *b* e *c* são combinados para formar a predição do *pixel* da posição *x*. Essa predição é subtraída do valor atual do *pixel* em *x*, e a diferença passa por um codificador de entropia (ISO; ITU-T, 1992). Como a codificação sem perdas não envolve etapas de quantização e transformação DCT, a taxa de compressão é menor que na codificação com perdas, porém, a imagem resultante na decodificação é idêntica à original.

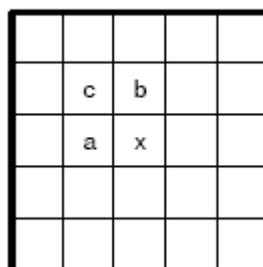


Figura 2 – Predição do valor do *pixel* no JPEG sem perdas.

2.1.3 Modos de Operação do JPEG

Quatro modos de operação são definidos: seqüencial, progressivo, sem perdas e hierárquico.

Para ambos os modos, seqüencial e progressivo, a codificação se baseia no método de

codificação com perdas apresentado na seção anterior, diferindo porém, na forma com que os blocos de 8 X 8 *pixels* são agrupados. Como resultado, no modo seqüencial a imagem é decodificada seqüencialmente, enquanto que no modo progressivo algumas partes da imagem são decodificadas antes de outras.

Da mesma forma como os dois acima, o modo sem perdas segue o método de codificação homônimo apresentado na seção anterior.

No último modo definido, o modo hierárquico, uma imagem estática é codificada como uma seqüência de quadros (ISO; ITU-T, 1992). Com a decodificação de um quadro, têm-se uma predição para o quadro subsequente. A codificação para a diferença entre os quadros pode seguir a forma com perdas, sem perdas ou ambas. Filtros de subamostragem são utilizados para simplificar a decodificação dos primeiros quadros, fazendo com que as primeiras decodificações tratem de imagens com baixa resolução, que melhora à medida que novos quadros são decodificados.

2.1.4 Escolha da codificação de imagem utilizada na transmissão

Além do JPEG, outros formatos de arquivos de imagens, como os “gráficos portáteis para rede” (*portable network graphics* – PNG), o “formato de intercâmbio de gráficos” (*graphics interchange format* – GIF) e o próprio *bitmap* são utilizados em diferentes aplicações. No sistema proposto neste trabalho, o JPEG foi escolhido devido ao seu bom desempenho com imagens de ambientes reais, devido a sua portabilidade, e baixa complexidade, uma vez que a decodificação da imagem é realizada em um aparelho celular com capacidade computacional limitada.

O uso de formatos de vídeo, como o MPEG-1, MPEG-2 ou MPEG-4 (*Moving Picture Experts Group*), *motion JPEG* ou WMV (*windows media video*) também poderia ser utilizado. Porém, devido aos requisitos da aplicação proposta, optamos pela utilização de seqüência de imagens codificadas em JPEG utilizando o algoritmo do tipo sem perdas (*lossy*).

2.2 Protocolo TCP/IP

O protocolo de controle de transmissão (*transmission control protocol* – TCP) e o protocolo de internet (*internet protocol* – IP) foram criados com o intuito de realizar a comunicação entre computadores na rede mundial de computadores (*world wide web* – WWW ou WEB). Esses protocolos têm a função de controlar como a informação é transmitida de uma rede para outra, e como lidar com o endereçamento dos pacotes, a fragmentação dos dados e a checagem de erros.

2.2.1 Modelo Cliente-Servidor

Pode ser definido como o processamento cooperativo de requisições submetidas por um cliente a um servidor que as processa e retorna um resultado. Os clientes são processos que requisitam serviços e, os servidores são processos que recebem requisições de clientes, realizam uma operação e retornam resultados esperados. Uma máquina pode executar um simples processo (cliente ou servidor), múltiplos clientes, múltiplos servidores ou múltiplos clientes e servidores. O exemplo mais comum de um modelo cliente-servidor é a *web*, no qual o cliente é o navegador (*browser*) e o servidor é a máquina onde a página solicitada está armazenada.

2.2.2 Modelo de Referência ISO/OSI

Para facilitar o desenvolvimento de sistemas de comunicação entre computadores, a ISO definiu o modelo de referência de interconexões de sistemas abertos OSI que divide todo o processo em camadas. Com isso facilita-se o tratamento de problemas relativos a cada camada e a sua otimização. O modelo ISO/OSI é constituído por sete camadas, as quais são descritas sucintamente na Tabela 1 (KUROSE, 2003).

Tabela 1 – Camadas do modelo OSI

| ID | CAMADA | ESPECIFICAÇÃO |
|----|--------------|--|
| 7 | Aplicação | Esta camada funciona como uma interface de ligação entre os processos de comunicação de rede e as aplicações utilizadas pelo usuário. |
| 6 | Apresentação | Aqui os dados são convertidos e garantidos em um formato universal. |
| 5 | Sessão | Estabelece e encerra os enlaces de comunicação. |
| 4 | Transporte | Efetua os processos de envio dos pacotes na seqüência certa e confirmação de recebimento dos pacotes de dados. |
| 3 | Rede | Implementa o roteamento dos dados através da rede. |
| 2 | Enlace | Formatada a informação em quadros, que representam a exata estrutura dos dados fisicamente transmitidos através do meio físico. |
| 1 | Física | Define a conexão física entre o sistema computacional e a rede. Especifica o conector, níveis de tensão, dimensões físicas, características mecânicas e elétricas. |

2.2.3 Modelo TCP/IP

O TCP/IP baseia-se em um modelo de referência de quatro camadas, no qual o conjunto de protocolos TCP/IP está localizado nas três camadas superiores desse modelo. Conforme ilustra a Tabela 2, cada camada do modelo TCP/IP corresponde a uma ou mais camadas do modelo de referência OSI (KUROSE, 2003).

Tabela 2 – Comparação entre o modelo OSI e as 4 camadas do TCP/IP

| Modelo OSI | | Modelo TCP/IP | |
|------------|--------------|---------------|----------------------|
| ID | CAMADA | ID | CAMADA |
| 7 | Aplicação | 4 | Aplicação |
| 6 | Apresentação | | |
| 5 | Sessão | | |
| 4 | Transporte | 3 | Transporte |
| 3 | Rede | 2 | Internet |
| 2 | Enlace | 1 | Interface com a rede |
| 1 | Física | | |

Os tipos de serviços executados e os protocolos usados em cada camada do modelo TCP/IP são descritos mais detalhadamente na tabela a seguir.

Tabela 3 – Tipos de serviços e protocolos das camadas TCP/IP

| ID | CAMADA | ESPECIFICAÇÃO | PROTOCOLOS |
|----|-------------------|---|--|
| 4 | Aplicação | Define os protocolos de aplicativos TCP/IP e como os programas hospedeiros estabelecem uma interface com os serviços de camada de transporte para usar a rede. | HTTP, Telnet, FTP, TFTP, SNMP, DNS, SMTP, X Windows, outros protocolos de aplicativos. |
| 3 | Transporte | Fornecer gerenciamento de sessão de comunicação entre computadores hospedeiros. Define o nível de serviço e o status da conexão usada durante o transporte de dados. | TCP, UDP, RTP |
| 2 | Internet | Empacota dados em datagramas IP, que contêm informações de endereço de origem e destino usados para encaminhar datagramas entre hospedeiros e redes. Executa o roteamento de datagramas IP. | IP, ICMP, ARP, RARP |
| 1 | Interface de Rede | Especifica os detalhes de como os dados são enviados fisicamente pela rede. Especifica também o <i>hardware</i> que estabelece interface com o meio da rede, como cabo coaxial, fibra óptica ou par trançado. | Ethernet, <i>Token Ring</i> , FDDI, X.25, retransmissão de quadros, RS-232, V.35. |

(TELECO, 2008)

2.2.4 Protocolo TCP

O TCP é um protocolo da camada de transporte do modelo TCP/IP, responsável pela entrega segura e ordenada de um fluxo de bits de um programa em um computador através da rede a outro programa em outro computador. Os erros ocorridos na transferência, como pacotes perdidos, duplicados ou entregues fora de ordem, são tratados pelo TCP. Através do uso de portas, que consiste em um número inteiro que endereça a aplicação dentro de uma máquina, o TCP permite a execução de múltiplas aplicações em um dispositivo computacional.

2.2.5 Protocolo UDP

O protocolo de datagrama de usuário (*user datagram protocol* - UDP) é um protocolo da camada de transporte que permite a comunicação simplificada entre aplicações. O pacote UDP é enviado ao destinatário sem garantia e sem confirmação de entrega. É utilizado em aplicações nas quais, a entrega dos dados deve ser realizada o mais rápido possível, admitindo-se que os pacotes sejam entregues com erros, fora de ordem e até perdidos. A responsabilidade pelo tratamento ou não dos prováveis erros de transmissão neste caso deve ficar a cargo da aplicação.

2.2.6 Conceito de Soquetes

Especificamente em computação, um soquete é uma interface entre um processo e o protocolo TCP/IP provido pelo sistema operacional.

O soquete é também uma abstração computacional que mapeia um endereço de rede diretamente a uma porta de transporte (TCP ou UDP). Com esse conceito é possível identificar unicamente um aplicativo cliente ou servidor na rede de comunicação IP. É usado em ligações de redes de computadores com a finalidade de estabelecer um elo bidirecional de comunicação entre dois programas.

2.3 Rede GSM com GPRS e EDGE

O sistema global para comunicações móveis (*global system for mobile communications - GSM*) foi desenvolvido na Europa, nos anos 80, sendo considerado uma tecnologia de segunda geração. A rede GSM foi otimizada para serviços de voz, utilizando comutação de circuitos. Devido a sua flexibilidade é possível transmitir dados via comutação de circuitos (*circuit switched data - CSD*), no qual o aparelho celular funciona como um modem conectado à rede de telefonia pública comutada (RTPC), podendo atingir taxas de transmissão de até 9,6 kbps (SVERZUT, 2005).

Devido a crescente necessidade maiores taxas de transmissão de dados foi desenvolvido o serviço de rádio de pacote geral (*general packet radio service - GPRS*), que é serviço de transmissão de dados com taxas teóricas de até 171,2 kbps (SVERZUT, 2005). Este serviço é considerado de geração 2,5 é implantado nas redes GSM com atualizações de *software*. O GPRS utiliza a comutação por pacotes através do protocolo IP para efetuar a transmissão de dados, ao contrário do GSM que usa comutação de circuitos para fazer transmissões de dados.

Posteriormente foi desenvolvido o padrão o EDGE que é uma evolução da tecnologia GPRS, “taxa de dados melhorada para evolução do GSM” (*enhanced data rates for GSM evolution - EDGE*), visando aumentar mais a taxa de dados nas redes GSM. Ele também é conhecido como GPRS melhorado (*enhanced GPRS - EGPRS*). Esta tecnologia pode atingir taxas teóricas de até 473,6 kbps usando modulação 8-PSK e de novos esquemas de codificação de canal (SVERZUT, 2005).

A taxa de transmissão que um dispositivo pode usar é indicada pelas classes tanto no GPRS como no EDGE. Os valores das classes variam de 2 a 12, conforme a velocidade alcançada (GSM, 2008).

2.4 Mensagens SMS e Protocolo PDU

O serviço de mensagens curtas (*short message service* – SMS) é um serviço de comunicação da rede celular que permite a troca de mensagens curtas de texto entre aparelhos celulares. Projetado inicialmente para o GSM e posteriormente expandido para outros sistemas, este protocolo é especificado pelo Instituto Europeu de Padrões de Telecomunicações (*European Telecommunications Standards Institute* – ETSI), nos documentos GSM 03.40 e GSM 03.38 (ETSI, 2001). Nestes documentos, dois serviços diferentes são definidos: 1) os originados pelo aparelho celular (*short message mobile originated* - SMS-SUBMIT), e os terminados no aparelho celular (*short message mobile terminated* - SMS-DELIVER).

O tipo SMS-SUBMIT é usado para encaminhar mensagens SMS do aparelho celular ao centro de serviço (*service centre* – SC), para depois serem entregues ao aparelho celular destinatário. O SC pode prover relatórios de sucesso ou de erro na entrega da mensagem. No SMS-DELIVER a mensagem é encaminhada pela rede GSM, do SC ao aparelho celular.

2.4.1 O Formato PDU

Para gerar as mensagens, o SMS utiliza um protocolo de baixo nível chamado de protocolo de unidade de dados (*protocol data unit* – PDU). Nele são configuradas as informações como o período de validade da mensagem, prioridade e caminho de retorno.

As informações de aplicação¹ são enviadas no corpo da mensagem usando o conceito de cabeçalho de dados do usuário (*user data header* – UDH), através do qual é possível também indicar a concatenação de mensagens e fornecer alguns parâmetros de controle ao SC. Uma das possibilidades de utilização do UDH na mensagem no formato PDU é o envio do SMS contendo a porta de destino aparelho celular. Com esta informação e o uso da classe PushRegistry do java (SUN, J, 2008), é possível ativar de um aplicativo java no aparelho celular com a recepção de uma mensagem SMS, ficando ainda o corpo da mensagem

¹ Informações de aplicação podem ser usadas para indicar que a mensagem é concatenada (tem várias partes, o que ocorre quando o SMS tem mais de 160 caracteres) ou para escolher a porta de destino da mensagem, entre outros.

acessível para a aplicação ativada.

O formato PDU para uma mensagem do tipo SMS-SUBMIT é escrito em bytes e os parâmetros mostrados na Figura 3. Cada parâmetro será detalhado nas subseções que seguem.

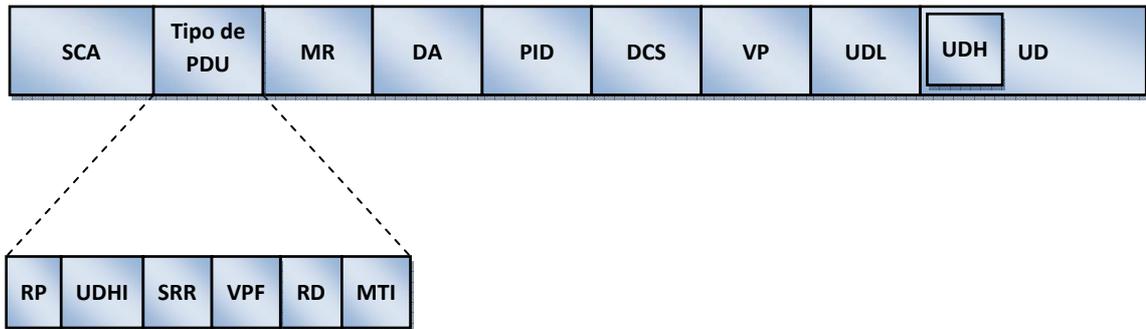


Figura 3 – Quadro PDU.

a) Endereço do Centro de Serviço (*Service Center Address – SCA*)

Este campo do quadro é utilizado indicar o centro de serviço (SC) usado para a transferência da mensagem quando este não é o padrão do cartão SIM.

b) Tipo de PDU

Vários parâmetros são especificados neste byte do quadro: 1) caminho de retorno (*reply path – RP*), que habilita o caminho de retorno; 2) indicador cabeçalho de dados do usuário (*user data header indicator – UDHI*), que indica se no campo de dados do usuário existirá um cabeçalho de aplicação; 3) requisição de relatório de estado (*status report request – SRR*), que solicita ao SC um relatório de estado; 4) formato do período de validade (*validity period format – VPF*), que indica se o campo Período de Validade estará presente e seu formato em caso positivo; 5) rejeição de duplicatas (*reject duplicate - RD*), que indica ao SC que a mensagem deve ser rejeitada, caso já exista registro de outros SMS com a mesma referência da mensagem e endereço de destino; 6) indicador de tipo da mensagem (*message type indicator - MTI*), que indica o tipo da mensagem, sendo normalmente configurado com SMS-DELIVER ou SMS-SUBMIT.

c) Referência da Mensagem (*Message Reference – MR*)

O campo referência da mensagem contém um número entre zero e 255 usado para

endereçar as mensagens SMS-SUBMIT e SMS-DELIVER.

d) Endereço de Destino (*Destination Address – DA*)

O endereço de destino da mensagem SMS pode ser especificado de duas maneiras. No modo nacional, este campo contém o código nacional e número do assinante. No modo internacional, essas informações são precedidas do código internacional.

e) Identificação do Protocolo (*Protocol Identifier – PID*)

Permite informar ao SC qual dispositivo receberá a mensagem, indicando os protocolos utilizados nas camadas superiores. Isto é útil quando a mensagem é enviada para um dispositivo que não é um aparelho celular padrão. Para mais informações sobre os diversos dispositivos suportados, ver a página 52 da norma do SMS (ETSI, 2001).

f) Esquema de Codificação dos Dados (*Data Coding Scheme – DCS*)

O padrão GSM define dois esquemas de codificação para o texto da mensagem. No esquema mais utilizado, chamado de alfabeto padrão (*default alphabet*), a mensagem é codificada usando um alfabeto de 7 bits (ETSI, 1999), de modo que cada oito caracteres são empacotados em sete octetos. Segundo o padrão, todo aparelho celular GSM com suporte à SMS deve entender a codificação de 7 bits do alfabeto padrão. Uma outra opção é a codificação do texto em 8 bits de acordo com a tabela ASCII, no entanto essa codificação não é suportada por todos os aparelhos celulares.

Neste mesmo parâmetro DCS é possível indicar a localização do armazenamento da mensagem, que pode ser no cartão SIM ou na memória do aparelho celular.

g) Período de Validade da Mensagem (*Validity Period – VP*)

Caso seja especificado, o período de validade indica o período pelo qual o SC deve manter a mensagem em memória antes que a entrega seja descartada. Isso é usado em caso de impossibilidade de entrega da mensagem ao destinatário. A definição da validade pode ser relativa ou absoluta. Na forma relativa o tempo de validade é relacionado ao momento de envio da mensagem. Na forma absoluta a data e hora exata de descarte são especificadas.

h) Comprimento dos Dados do Usuário (*User Data Length – UDL*)

O campo UDL define o tamanho em octetos (bytes) ou septetos do campo UD,

dependendo do esquema de codificação utilizado.

i) Dados do Usuário (*User Data – UD*)

O campo UD contém a mensagem de texto, codificada de acordo com o esquema de codificação escolhido. O número de caracteres é limitado a 160 ou 140 dependendo se a codificação é de 7 ou 8 bits por caractere.

j) Cabeçalho dos Dados do Usuário (*User Data Header – UDH*)

O cabeçalho de dados do usuário (*user data header – UDH*) é de tamanho variável localizado dentro do UD. O UDH consiste de: 1) um byte com o número de octetos utilizados na declaração do UDH; 2) um byte indicador do tipo de serviço especial que será declarado nos bytes seguintes; 3) um byte com o total de bytes utilizados para a declaração do serviço e; 4) a declaração dos parâmetros específicos do serviço (portas utilizadas, identificação da mensagem concatenada, entre outros). (ETSI, 2001).

2.5 Módulo MC75 e Terminal T75 EDGE-BR

O MC75 é o primeiro módulo GSM/GPRS lançado comercialmente pela Siemens com suporte a tecnologia EDGE (SIEMENS, 2008). É um dispositivo de suporte ao acesso a rede celular tendo todas as funcionalidades de um celular convencional. Para simplificar o desenvolvimento de aplicações com acesso à rede GSM/EDGE a empresa nacional DuoDigit desenvolveu o terminal T75 EDGE-BR que possui interfaces de entrada e saída padronizadas, que facilitam a conexão do módulo MC75 a outros elementos de *hardware*, como computadores e dispositivos embarcados.

Através do T75 EDGE-BR, podem-se acessar serviços de operadoras de telefonia celular. Com esta finalidade, é instalado no terminal um cartão SIM, que lhe fornece o número telefônico e o associa a uma conta telefônica junto à operadora celular.

2.5.1 Características do Módulo MC75

O MC75 possui as seguintes características: 1) Suporte as quatro bandas do espectro de radiofrequência da rede GSM 850/900/1800/1900 MHz; 2) GPRS *multi-slot* classe 12 e EDGE *multi-slot* classe 10; 3) controle e operação através de comandos AT (3GPP, 2003); 4) suporte ao envio de mensagens SMS ponto a ponto e ponto a multiponto (*broadcast*); 5) suporte a especificações de fax do grupo 3 e Classe 1; 6) suporte a chamadas de voz, estacionamento de chamadas e chamadas em espera;

O MC75 suporta a transferência de dados nos modos de CSD e no GPRS/EDGE. No modo GPRS/EDGE o módulo usa a rede de pacotes para realizar a transferência dos dados.

Para facilitar o acesso à internet pelas aplicações o MC75 implementa também o protocolo TCP/IP, sendo fornecidos os seguintes serviços de internet: 1) soquete TCP cliente e servidor; 2) soquete cliente UDP; 3) cliente para o protocolo de transferência de arquivos (*file transfer protocol* – FTP); 4) cliente para o protocolo de transferência de hipertexto (*hypertext transfer protocol* – HTTP); 5) cliente para protocolo simples de transferência de *email* (*simple mail transfer protocol* – SMTP) e 6) cliente para o protocolo de correio (*post office protocol version 3* – POP3).

Para o envio de SMS, o MC75 define dois modos de escrita das mensagens, o modo texto e o modo PDU. O modo texto é uma abstração utilizada para facilitar o envio das mensagens.

Neste modo, a mensagem é redigida e montada automaticamente no formato do protocolo PDU pelo MC75, não sendo possível alterar o cabeçalho. No modo PDU, a mensagem a ser enviada deve ser montada manualmente, devendo ser formatada byte-a-byte desde o cabeçalho até o corpo da mensagem. Desta forma, embora mais complexa, todas as opções do protocolo estão acessíveis ao desenvolvedor.

2.5.2 Características do Terminal T75 EDGE-BR

O terminal T75 EDGE-BR possui duas interfaces serial RS-232, denominadas ASC0 e ASC1, para comunicação com outros dispositivos do tipo equipamento terminal de dados (*data terminal equipment* – DTE). Essas interfaces podem operar de 300 bps a 230.400 bps e suportam controle de fluxo por *hardware* RTS (*request to send*) e CTS (*clear to send*). Também suportam o protocolo de comunicações assíncrono XON/XOFF, no qual o dispositivo receptor usa caracteres especiais para controlar o fluxo de dados do dispositivo transmissor (EIA, 1969). O terminal T75 EDGE-BR possui também uma conexão USB 2.0, que pode ser usada para comunicação com outros dispositivos como microcomputadores, situação na qual o fluxo de dados do T75 EDGE-BR é controlado por *software*. O terminal ainda dispõe de uma interface de leitura para cartões SIM de 1,8 V e 3 V.

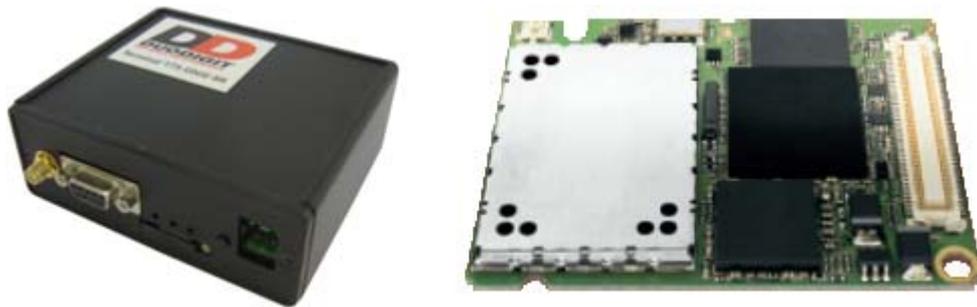


Figura 4 – Terminal T75 EDGE-BR e Módulo MC75.

2.6 Comandos AT

Os comandos AT são comandos específicos para modems constituídos de palavras curtas que combinadas produzem um conjunto completo de comandos. Estes comandos são usados para todos os tipos de operações tais como discagem, desligamento, e alteração dos parâmetros da conexão, transmissão e recepção de dados. Um equipamento que utiliza o conjunto de comandos AT, possui dois modos de operação: modo de comandos e modo de dados. No modo de comandos as palavras recebidas são interpretadas como comandos a serem executados pelo modem local. A configuração dos parâmetros de conexão é realizada no modo de comandos. No modo de dados todos dados recebidos são tratados como mensagem que será enviada através da conexão estabelecida.

A Tabela 4 apresenta a sintaxe e função de cada um dos comandos. Os comandos AT começam sempre com as letras AT seguidas das letras do comando. Nesta tabela, o comando é representado por um X. Os comandos AT podem ser de quatro tipos: comandos básicos, estendidos, proprietários e de registradores.

Tabela 4 – Função e sintaxe dos comandos AT

| TIPOS DE COMANDOS AT | SINTAXE | FUNÇÃO |
|----------------------|---------|--|
| Comando de teste | ATX=? | Retorna a lista de valores que o parâmetro pode receber |
| Comando de leitura | ATX? | Retorna o valor que está atualmente configurado no parâmetro |
| Comando de escrita | ATX=< > | Configura um parâmetro com o valor passado |
| Comando de execução | ATX | Executa um comando |

Os comandos básicos são constituídos de uma letra seguida de um dígito. Os comandos estendidos contêm um caractere “&” e uma letra seguida de um dígito. Os comandos proprietários geralmente iniciam com uma barra invertida (“\”) ou com percentual (“%”), sendo que os comandos proprietários GSM iniciam com o símbolo “+”, enquanto que os comandos proprietários da Siemens iniciam com um acento circunflexo (“^”). Os comandos de registradores são usados para alterar o valor dos registradores, que são posições específicas da memória do equipamento. Esses comandos têm a forma “S_x=y”, onde “x” é o número do registrador e “y” o valor que lhe será atribuído.

2.7 A Tecnologia Java

Em 1994 a empresa Sun Microsystems desenvolveu e disponibilizou a tecnologia java. Desde então ela tornou-se muito popular, obtendo sucesso tanto no meio empresarial como no meio acadêmico (JOHNSON, 2007). Este sucesso pode ser atribuído principalmente à portabilidade do código, que permite que um programa escrito em linguagem java funcione em qualquer plataforma, seguindo a filosofia “Escreva uma vez, execute em qualquer lugar”, (*Write once, run anywhere*) (FLANAGAN, 2005).

Para que a linguagem java seja uma linguagem de programação universal que possa ser executada em qualquer sistema computacional e *hardware*, o programa fonte em java é compilado gerando um código de bytes (*byte code*), que é lido por um interpretador conhecido como Máquina Virtual Java (*Java Virtual Machine – JVM*) (JOHNSON, 2007).

Durante a execução, a JVM traduz as instruções para uma linguagem de máquina compatível com o tipo de *hardware* utilizado. Dessa forma, pode-se utilizar um mesmo código fonte e executá-lo em qualquer sistema que tenha uma JVM instalada (JOHNSON, 2007).

2.7.1 Tipos de Plataforma Java

Teoricamente, seria possível também a utilização de um código de bytes em dispositivos móveis com a JVM instalada. No entanto, existe uma grande diferença de capacidade de processamento e memória entre os vários dispositivos computacionais. Por isso, a plataforma java foi dividida em quatro subdivisões (JOHNSON, 2007), sendo cada uma adequada a um tipo de dispositivo conforme ilustra a Figura 5.

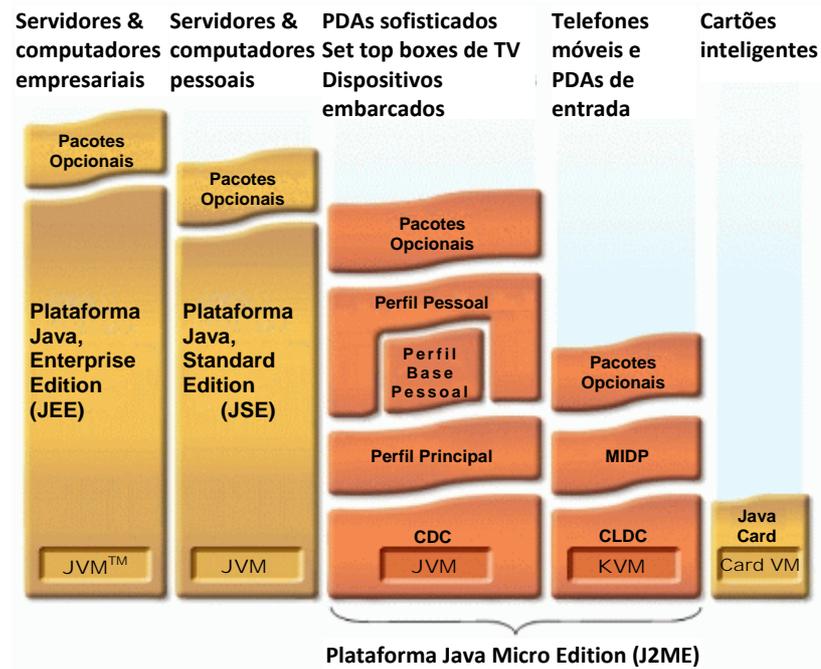


Figura 5 – Subdivisões da Plataforma Java.

A plataforma *java enterprise edition* (JEE) é a plataforma java mais usada principalmente para aplicações corporativas. Os programas desenvolvidos são executados em servidores de aplicações.

A plataforma *java standard edition* (JSE) é a plataforma instalada nos computadores pessoais de mesa, sendo utilizada em diversos tipos de aplicações. Esta plataforma é mais adequada a estes computadores, pois têm grande capacidade de processamento, alta resolução de tela e grande quantidade de memória.

A plataforma *java micro edition* (JME) foi criada para tornar possível a execução de programas em linguagem java em dispositivos móveis, os quais têm grandes restrições quanto à resolução de tela, memória e capacidade de processamento. Nessa plataforma a JVM é bastante reduzida tendo apenas cerca de 80 kB, sendo por isso conhecida como máquina virtual de kilobyte (*kilobyte virtual machine* - KVM). Essa plataforma será usada neste projeto o por isso será detalhada melhor na próxima seção.

A plataforma *java card* é a menor plataforma java. Tem como objetivo a construção de programas para serem usados em cartões inteligentes (*Smart Cards*).

2.7.2 Plataforma JME

A plataforma JME provê portabilidade de *software* para os diferentes modelos de

aparelhos celulares e PDAs, desde que esses dispositivos disponibilizem os mesmos perfis de configuração. Para permitir a utilização da mesma plataforma em diferentes dispositivos, o JME foi dividido nas seguintes camadas: perfil, configuração, máquina virtual e sistema operacional, conforme ilustra a Figura 6 (JOHNSON, 2007).

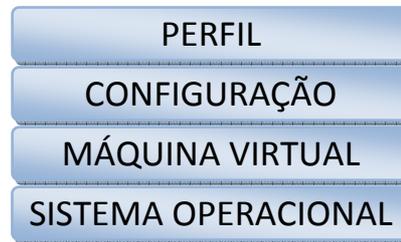


Figura 6 – Camadas da JME.

Nessa plataforma a camada máquina virtual foi projetada especialmente para dispositivos pequenos (de máximo de 512 kB de memória total), tais como aparelhos celulares.

A camada de configuração provê o conjunto básico de bibliotecas e capacidades da máquina virtual para uma determinada categoria de dispositivos. Cada categoria é uma classificação dos diversos dispositivos existentes em termos de suas capacidades computacionais, desde computadores de mesa até unidades móveis.

As configurações atuais da JME são: configuração de dispositivos de conexão limitada (*connected limited device configuration* - CLDC) destinada a aparelhos celulares e a configuração de dispositivo conectado (*connected device configuration* – CDC) para dispositivos com maior capacidade tais como PDAs.

O perfil é um conjunto de interfaces de programação de aplicativos (*application programming interface* – API) mais específicas a uma categoria de dispositivos do que as bibliotecas disponíveis pela configuração. Perfis são implementados ao topo da configuração. Os perfis mais frequentemente utilizados são: o perfil de dispositivo móvel (*mobile information device profile* – MIDP) usado em conjunto com a configuração CLDC, e o perfil pessoal (*personal profile* – PP) em conjunto com a configuração CDC.

Além desses perfis existem ainda pacotes opcionais que disponibilizam recursos especiais para aplicações mais específicas, tais como multimídia e jogos.

a) Configurações

O objetivo de se ter uma configuração é definir um padrão de ambiente de execução de programas para uma mesma categoria de dispositivos, que é determinada por suas

características de *hardware*. Essa camada possui as bibliotecas básicas da linguagem.

Os dispositivos compatíveis com a configuração CLDC têm pouca capacidade de processamento, memória e resolução gráfica. Para suportar a CLDC, o dispositivo precisa ter processadores de no mínimo 16 bits com velocidade de 16 MHz, 160 kB de memória não volátil e 192 kB de memória volátil. A CLDC é implementada em dispositivos que têm capacidade de transmissão limitada para se conectar em redes sem fio (JOHNSON, 2007). A configuração CDC é utilizada em dispositivos móveis com mais recursos, os quais têm processadores de no mínimo 32 bits, 2,5 MB de memória não volátil e 2 MB de memória volátil. Essa configuração permite o uso de todos os recursos da linguagem java, usando uma máquina virtual chamada *CDC hotspot implementation* (JOHNSON, 2007), além de ter pacotes opcionais próprios para aplicações especiais.

b) Perfis

Associado a cada configuração existe um perfil que complementa a API do JME, contendo as bibliotecas relacionadas ao dispositivo. Esses perfis implementam as funcionalidades de alto nível, como o ciclo de vida da aplicação, interface gráfica e persistência de dados (armazenamento dos dados na memória não volátil). Desta forma um aplicativo desenvolvido para funcionar em um determinado perfil também funcionará em outros dispositivos que implementem esse mesmo perfil, garantindo a portabilidade entre dispositivos desde que o aplicativo use apenas os recursos padrões da plataforma. Alguns fabricantes desenvolvem bibliotecas proprietárias para os seus produtos, de modo que um código escrito utilizando essas bibliotecas só funcionará no aparelho deste fabricante. Caso o código de bytes seja portado e executado em um dispositivo de outro fabricante ocorrerá um erro de execução.

No projeto descrito neste documento, o perfil MIDP foi utilizado por ser o único disponível para dispositivos móveis. A especificação completa do MIDP é definida no padrão JSR 118 (SUN, T, 2008). Entre os recursos que o perfil MIDP implementa, há suporte para protocolos de rede como HTTP, soquetes (*sockets*), reprodução de multimídia, APIs para jogos e suporte ao sistema de cores RGB.

c) MIDlets

Os aplicativos baseados no perfil MIDP são implementações de uma classe abstrata chamada MIDlet, a qual gerencia o ciclo de vida de uma aplicação. Este ciclo pode ser visto

na Figura 7. Quando uma aplicação é ativada, a Máquina Virtual a inicia através da chamada do método `startApp`. A classe `MIDlet` então entra no estado ativo, permanecendo neste estado até ser pausada ou fechada pelos métodos `pauseApp`, `destroyApp` respectivamente. Estes métodos devem estar presentes, obrigatoriamente, em todos os `MIDlets`

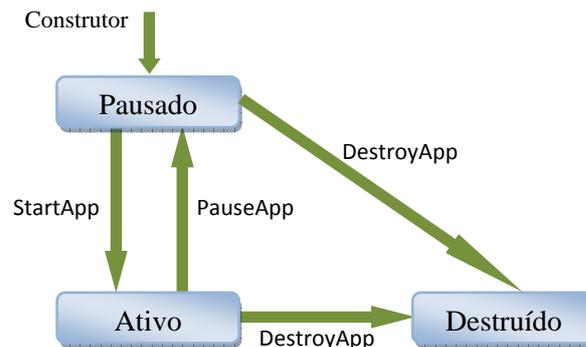


Figura 7 – Ciclo de vida de um aplicativo MIDlet.

A coleção de arquivos e classes da linguagem java empacotadas que forma um aplicativo do perfil MIDP é denominada de suíte MIDlet, sendo geralmente inserida dentro de um arquivo java (*java archive* – JAR). No JAR, informações do MIDlet são empacotadas junto com o descritor de aplicação java (*java application descriptor* – JAD), onde estão contidas informações de execução do aplicativo.

d) A Classe `PushRegistry`

A plataforma JME contém uma classe chamada `PushRegistry` capaz de manter um registro de possíveis conexões entrantes (SUN, J, 2008). Estas conexões podem ser registradas pelo aplicativo de duas formas: no momento de sua instalação no dispositivo, mediante a adição de comandos ao descritor do aplicativo (forma estática de registro); ou durante sua execução, através de métodos no próprio código (forma dinâmica de registro). Estas conexões são registradas e identificadas com parâmetros como o endereço IP e a porta de aplicação pela qual pode vir a ser feito um pedido de conexão com o dispositivo.

Uma vez registrada certa conexão, a KVM pede para o *software* de controle de aplicação (*application management software* – AMS) para ficar monitorando pedidos de conexão. O AMS é o *software* nativo do dispositivo que controla a ativação de aplicativos java, e monitora enquanto a aplicação não está ativa. Quando chega ao dispositivo um pedido de

conexão que combina com os parâmetros fornecidos pelo aplicativo, a AMS ativa o aplicativo através do método `startApp`. Deste modo, através do envio de uma mensagem SMS específica para o aparelho celular é possível ativar automaticamente um aplicativo java sem a intervenção do usuário.

2.7.3 Ferramentas de Desenvolvimento

Para o desenvolvimento de aplicativos java são usadas as ferramentas conhecidas como ambiente de desenvolvimento integrado (*integrated development environment* – IDE), as quais normalmente possuem diversos recursos, tais como: 1) O editor para edição do código fonte do programa; 2) O compilador que gera a linguagem de máquina a partir do código fonte; 3) O depurador que auxilia no entendimento do programa e facilitando a correção de erros; e, 4) A Geração de código, que consiste basicamente em amostras de código que auxiliam a resolver problemas rotineiros.

Na internet estão disponíveis livremente para o desenvolvimento de aplicativos em JME duas ferramentas IDE: o *Eclipse* (ECLIPSE, 2008) e o *NetBeans* (NETBEANS, 2008). Neste projeto foi usada *NetBeans* (Figura 8) por ser uma IDE mais completa e não necessitar de complementos para suportar JME. Foi usado o *Sun Java Wireless Tool Kit 2.5.2 for CLDC* (Figura 9) para emular o funcionamento dos programas em uma unidade móvel. Este emulador é uma ferramenta para desenvolvimento de aplicações sem fio com configuração CLDC e perfil MIDP que já vem embutido no ambiente *NetBeans*.

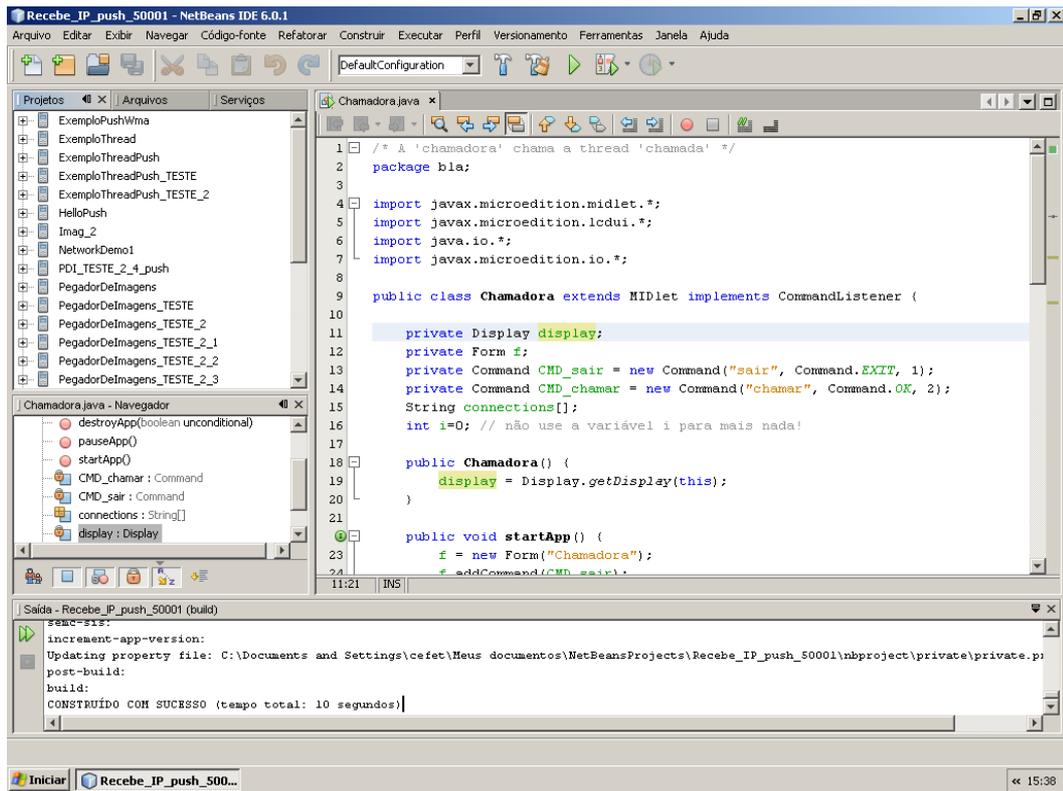


Figura 8 – Ambiente de desenvolvimento NetBeans.



Figura 9 – Emulador Sun Java Wireless Tool Kit.

3 Desenvolvimento e Implementação do Sistema

Para o desenvolvimento do sistema proposto, foi elaborado um projeto organizado por metas, no qual as diferentes atividades foram atribuídas aos integrantes do grupo de trabalho. Quatro metas foram definidas para a conclusão do projeto:

Meta 1 – Desenvolvimento de um sistema para captura e compressão da imagem;

Meta 2 – Transmissão de dados utilizando o terminal T75 EDGE-BR;

Meta 3 – Desenvolvimento de um aplicativo java ME para exibição das imagens recebidas na unidade móvel;

Meta 4 – Documentação e publicação dos resultados.

Como são quatro pessoas envolvidas no projeto, foram formados dois subgrupos de dois graduandos que seriam responsáveis pela concretização de algumas metas. Esta definição foi flexibilizada sempre que foram encontradas dificuldades maiores, momentos em que todos se concentraram em um só objetivo. Nos demais momentos os dois subgrupos trabalharam em paralelo, visando agilizar a conclusão do sistema proposto.

O desenvolvimento do sistema proposto passou por várias etapas. Dois sistemas (Sistema CaTICIS e Sistema RICS) foram construídos com a finalidade de testar o funcionamento em separado de partes do sistema final (Sistema EDGE2JAVA), o qual é a proposta deste trabalho. Neste capítulo os sistemas serão descritos na ordem de construção, sendo que os blocos já descritos nos sistemas anteriores não são repetidos na descrição do sistema EDGE2JAVA.

3.1 Materiais Utilizados no Projeto

Para a realização do projeto, foram utilizados os seguintes equipamentos de *hardware* e *softwares*: 1) Computador *desktop* com: processador AMD Sempron 3000+ 1,81 GHz, 512 MB de memória RAM, sistema operacional Windows XP Professional SP2; 2) Cabo RS-232 DB9 *crossover* para conexão de interfaces seriais; 3) terminal Duodigit T75 EDGE-BR com módulo Siemens MC75 versão 2.001b; 4) aparelhos celulares GSM Nokia 5310, 5200, 6265; SonyEricsson w300i, w200i, w900; Samsung d820; LG MG160b; Siemens C75; Intelbras A6; Motorola c650; 5) Cartões SIM das operadoras TIM, Brasil Telecom e Claro; 6) *webcam* Leadership Gotec Drop Cam 100; 7) *software* NetBeans 6.0.1; 8) *software* Matlab 7.2.0232 (R 2006a); 9) *software* Hiperterminal;

3.2 Sistema CaTICIS - Captura e Transmissão de Imagem entre dois Computadores via Interface Serial

O Sistema CaTICIS foi desenvolvido no Matlab para a realização de testes. Foram testadas a captura e codificação da imagem, o acesso à porta serial e a estrutura geral do aplicativo do servidor de imagens.

Neste sistema, uma seqüência de quadros de imagens é capturada por uma câmera do tipo *webcam* e cada quadro de imagem é codificado no formato JPEG. As imagens codificadas são enviadas de um computador transmissor para o receptor através de um cabo usando suas interfaces seriais do tipo RS-232 (conforme mostra a Figura 10). No computador receptor cada quadro é decodificado e exibido seqüencialmente na tela, simulando um vídeo. Tanto para realização da codificação em JPEG, como para a programação das interfaces seriais foram implementados programas em Matlab utilizando funções fornecidas pela caixa de ferramentas de aquisição de imagens (*image acquisition toolbox*).

Dois aplicativos foram desenvolvidos. Para o computador transmissor, foi desenvolvido o aplicativo CaTICIS-TX, para o receptor, o CaTICIS-RX.

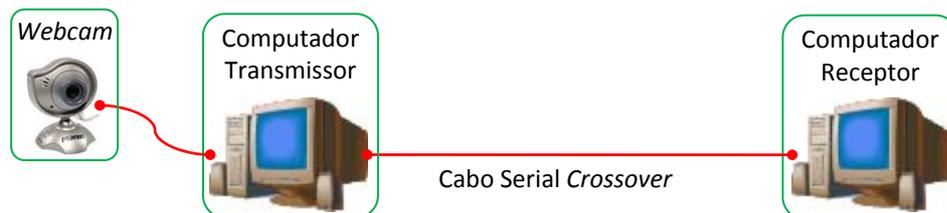


Figura 10 – Interligação dos equipamentos no sistema CaTICIS.

O fluxograma da Figura 11 apresenta em nível macro o funcionamento do sistema. O código fonte deste sistema está disponível na internet, no endereço <http://www.sj.cefetsc.edu.br/wiki/index.php/Edge2java>. A seguir serão descritas as principais etapas deste sistema.

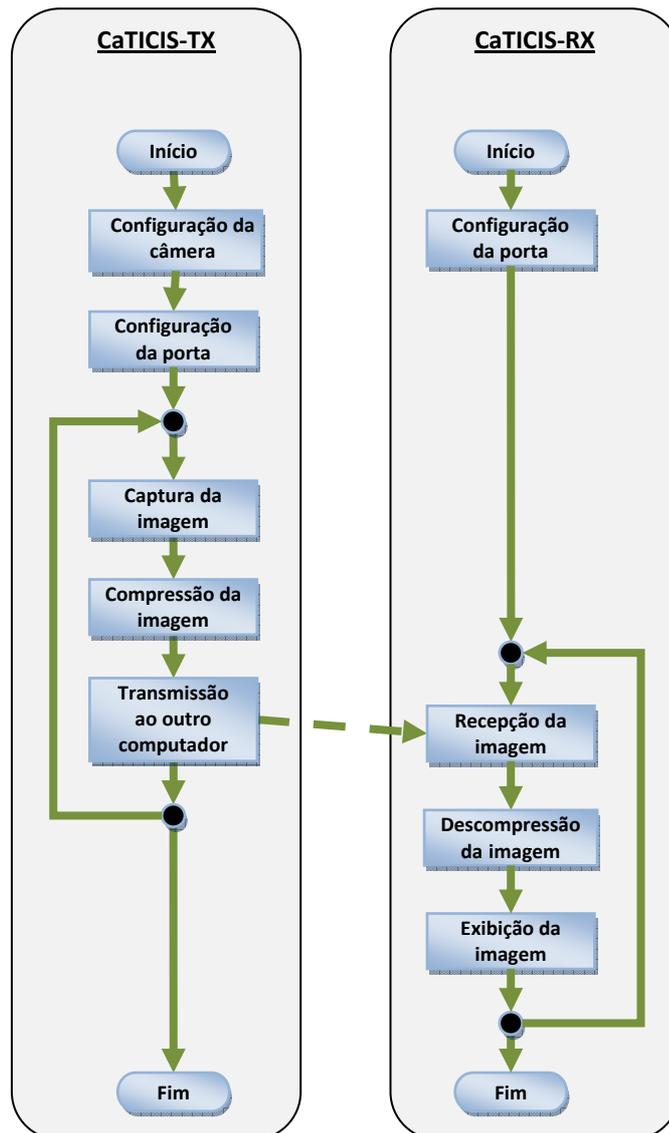


Figura 11 – Fluxograma de funcionamento do sistema CaTICIS

3.2.1 Captura da Imagem

A captura da imagem é iniciada com a configuração dos parâmetros envolvidos na comunicação do computador com a câmera. Estes parâmetros incluem o adaptador de vídeo, o formato de captura da imagem (largura, altura e número de bits de cor por *pixel*), a taxa de quadros por segundo e a definição do evento que inicia a captura. Foi utilizada uma resolução de imagem de 160 X 120, com 24 bits de cor por *pixel*.

Para realizar a captura contínua, o laço principal do programa tem uma estrutura de repetição. A cada repetição um quadro é captado da câmera, a imagem é codificada em JPEG e transmitida através da porta serial para o outro computador. O encerramento do laço de

captura da imagem é comandado pelo usuário pressionando a tecla “q”.

3.2.2 Codificação e Decodificação em JPEG

O processo de codificação e decodificação é realizado utilizando as funções do Matlab para operação com arquivos. Para a codificação, a matriz RGB obtida na captura da imagem é convertida no formato JPEG e salva no disco rígido do computador transmissor com a função `imwrite()`. Através da função `fread()` de leitura binária, a imagem codificada é recuperada do disco rígido e armazenada em um vetor o qual contém a imagem no formato JPEG. No receptor, o arquivo JPEG recebido é salvo em disco na forma binária com a função `fwrite()` e posteriormente lido pela função `imread()` que decodifica a imagem para o formato *bitmap* (Figura 12).

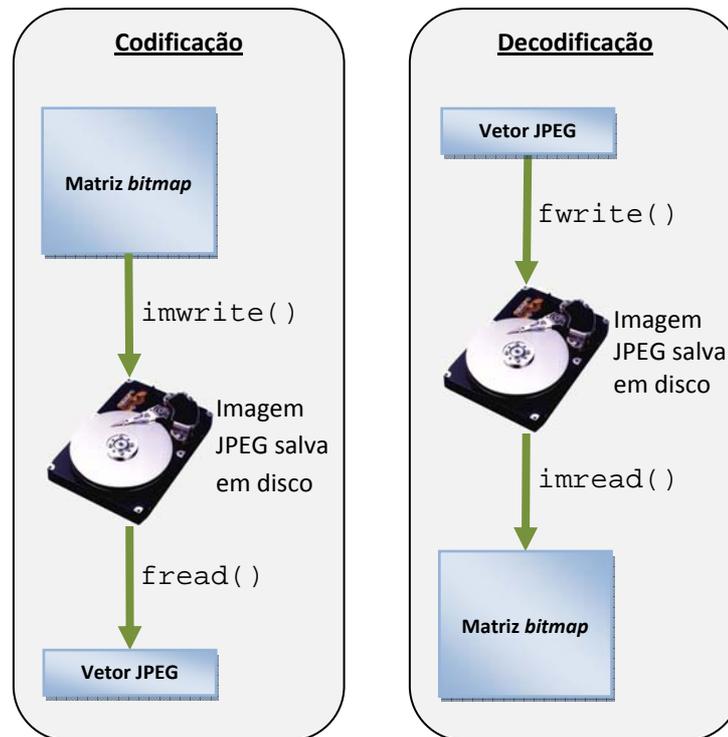


Figura 12 – Processo de codificação e decodificação JPEG utilizado

3.2.3 Transmissão e Recepção via Interface Serial

Para o uso da interface serial, inicialmente é necessário configurar os parâmetros da comunicação, tais como taxa de transmissão, bits de paridade, bits de parada e tipo de controle de fluxo. Para a transmissão (escrita) e recepção (leitura) de dados, o Matlab dispõe

de dois modos diferentes: o modo binário e o modo formatado.

As funções de escrita e leitura no modo binário (`fwrite` e `fread`) transmitem os dados somente na forma de bytes sem a possibilidade do envio de caractere de fim para o receptor. Por isso, a recepção ocorre enquanto existirem dados no *buffer*² de entrada da interface. Neste modo, é possível também indicar o número de bytes que devem ser lidos do *buffer*, sendo esta uma opção para a transmissão de seqüências de bytes quando o tamanho da seqüência de dados é previamente conhecido.

Outra opção é o uso das funções de escrita e leitura formatada (`fprintf` e `fscanf`), as quais permitem o envio para a porta serial de dados com tipo. Nos sistemas computacionais, os dados podem ser interpretados como vários tipos. Um dado pode ser representado como um número inteiro ou um caractere, por exemplo. Para que seja possível a recepção de seqüências de bytes sem ter o prévio conhecimento dos seus tamanhos, é necessário que o fim de cada seqüência seja informado. O caractere de fim padrão usado é `0dh`³. Quando esse caractere é recebido, a leitura é interrompida e os dados recebidos são armazenados em uma variável do aplicativo para serem tratados. Se necessário, a leitura é reiniciada para a aquisição da próxima seqüência a ser recebida.

Para a transmissão de texto, é utilizado um código padrão de caracteres (padrão ASCII) no qual cada letra corresponde a um byte. Além de letras, este padrão também representa caracteres de controle tais como o de fim de dados. Estes caracteres especiais não são usados em textos. Exemplo de uso de caracteres especiais da tabela ASCII, é o controle de fluxo de dispositivos e a formatação de textos.

Na transmissão byte-a-byte de uma imagem codificada em JPEG, é possível que os dados transmitidos imitem o caractere de fim, resultando na interrupção da transmissão. Em virtude deste problema optou-se pela transmissão de modo binário dos dados, sendo desenvolvido um protocolo para sinalizar o tamanho dos dados transmitidos (ver Figura 13).

O método usado consiste em, antes do envio da imagem, transmitir o total de bytes que compõe o dado. Esta informação é transmitida através de um cabeçalho contendo apenas um número decimal de 15 dígitos codificado em ASCII, sendo completado com zeros à esquerda do número quando todos os dígitos não forem utilizados. Após este cabeçalho, a imagem codificada em JPEG é transmitida byte-a-byte. No receptor, a primeira leitura de cada imagem

² Buffer é uma memória intermediária usada para armazenar temporariamente dados enquanto eles estão sendo movidos de um lugar para outro.

³ `0dh` é a notação hexadecimal do número inteiro 13.

será sempre do cabeçalho, e a leitura subsequente dos bytes, até o tamanho definido da imagem.



Figura 13 – Quadro de transmissão de imagens no sistema CaTICIS

Para sinalizar para o receptor que a transmissão de imagens será encerrada, é transmitida uma mensagem com número negativo no cabeçalho. Desta forma, o receptor interrompe a recepção ao receber um número negativo no campo de tamanho da mensagem.

3.2.4 Exibição da Imagem na Tela

Para a exibição do fluxo de imagens na tela, a função padrão de exibição de imagens do Matlab (`imshow`) foi utilizada. Cada nova imagem recebida é exibida sobrepondo a anterior, dando a sensação ao sistema visual humano de uma reprodução de vídeo.

3.2.5 Dificuldades Encontradas e Soluções Adotadas

No início do projeto pretendia-se implementar as rotinas de codificação e decodificação da imagem em memória, e para tal foi necessário realizar um estudo detalhado da norma JPEG (ISO; ITU-T, 1992). Devido a restrições de tempo, optou-se pela utilização do artifício, de compressão e descompressão de imagens, descrito na Seção 2.1, no qual a imagem é codificada e decodificada usando funções de escrita e leitura em arquivos do Matlab. A consequência dessa escolha é um maior tempo de codificação em comparação com uma rotina que opere diretamente na memória, a qual evitaria a necessidade de acesso ao disco e sistema de gerenciamento do Windows.

Outra dificuldade foi definir na linguagem Matlab um mecanismo de encerramento do laço principal do aplicativo, uma vez que a leitura do teclado é desabilitada durante a sua execução. Como solução foi criado um objeto `figura`, configurando seus atributos para deixá-lo invisível ao usuário. Quando o usuário pressiona a tecla “q”, ela é registrada no atributo “caractere atual” do objeto `figura`. Então, a cada repetição do laço de transmissão de imagens, o atributo em questão é testado quanto ao caractere registrado, e o encerramento é dado caso seja o caractere definido como indicador de encerramento da aplicação (ALOE,

2006).

3.2.6 Resultados

O objetivo da elaboração do Sistema CaTICIS foi realizar um estudo inicial para a elaboração do Sistema EDGE2JAVA, foco principal do projeto. Neste sistema, as etapas de captura da imagem da câmera, codificação em JPEG, transmissão pela interface serial e recepção no computador remoto, bem como a sua decodificação e exibição da imagem foram realizadas com êxito.

Com a resolução utilizada, foi obtido um tamanho médio de 4 kB por imagem.

3.3 Sistema RICS (Recepção de Imagem no Celular de um Servidor HTTP).

Este sistema resume-se a exibir sequencialmente na tela de um aparelho celular as imagens armazenadas em um servidor HTTP, tendo como principal objetivo desenvolver um aplicativo java para avaliar a compatibilidade do aparelho celular para leitura e exibição das imagens. O aplicativo desenvolvido foi chamado de `http2java`. Na Figura 14 é mostrada a estrutura física utilizada nos testes do sistema. Como as imagens armazenadas no servidor HTTP são acessíveis através da rede IP, utiliza-se um aplicativo em código java para receber as imagens no aparelho celular, exibindo-as seqüencialmente na tela. Para a conexão entre o aparelho celular e o servidor HTTP a classe `connector` é realizada. As etapas do funcionamento do sistema são mostradas no diagrama da Figura 15 e descritas nas próximas seções.

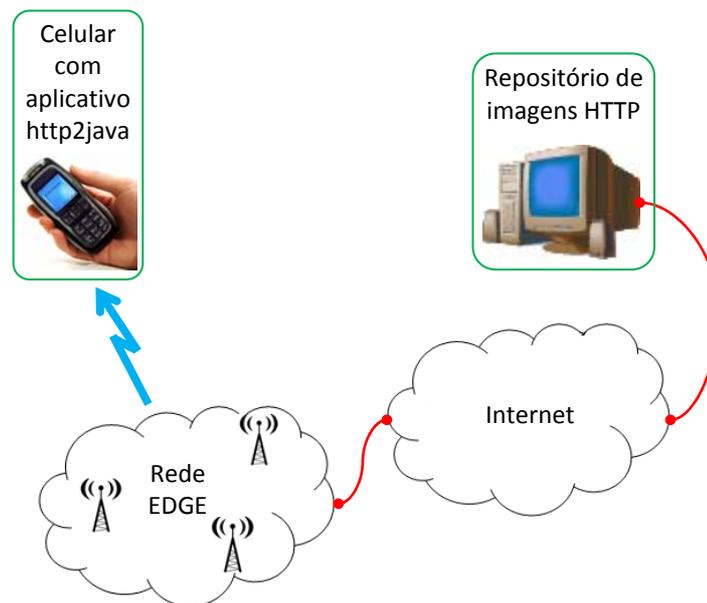


Figura 14 – Interligação dos equipamentos no Sistema RICS.

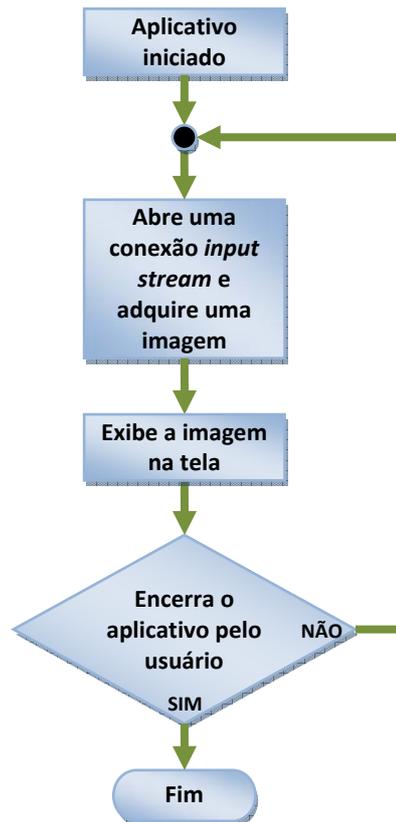


Figura 15 – Fluxograma de funcionamento do Sistema RICS

3.3.1 Armazenamento da Imagem no Servidor HTTP

Para realizar esta etapa, as imagens a serem exibidas no aparelho celular devem estar armazenadas em formato JPEG em um servidor HTTP. O URL usado nos testes foi <http://sj.cefetsc.edu.br/~emydios/nome da imagem>, conforme o exemplo a seguir:

http://endereço_da_imagem.com/imagem1.jpeg

http://endereço_da_imagem.com/imagem2.jpeg

...

http://endereço_da_imagem.com/imagemN.jpeg

3.3.2 Abertura de Conexão HTTP/TCP/IP/EDGE ou GPRS

Esta etapa consiste no acesso a web pelo aparelho celular através da tecnologia EDGE ou GPRS, seguido do estabelecimento de uma conexão de fluxo de dados com a URL na qual se encontram as imagens a serem exibidas. A abertura da conexão HTTP é feita através do aplicativo `http2java` previamente carregado e instalado no aparelho celular.

3.3.3 Recepção e exibição da imagem no aparelho celular

O aplicativo consiste basicamente em um laço infinito no qual requisições de páginas HTTP, compostas apenas de imagens, são realizadas em seqüência. Após cada requisição de página é feito o *download* de uma imagem através da conexão de fluxo de dados (*input stream connection*). A imagem recebida é montada em um objeto do tipo `ImageItem` utilizando os comandos de manipulação de imagens do java ME, e exibida na tela do aparelho celular.

3.3.4 Resultados

O sistema RICS funcionou como esperado nos diferentes aparelhos celulares testados (modelos Nokia 5200, 5310, 6265, Motorola C650 e Siemens C75).

O aplicativo `http2java` e o código fonte desse programa estão disponíveis no endereço <http://www.sj.cefetsc.edu.br/wiki/index.php/Edge2java>, e as imagens estão depositadas no endereço: <http://sj.cefetsc.edu.br/~emydios/>. Atualmente qualquer usuário que tenha um aparelho celular compatível com o java MIDP 2.0 pode, após a instalação do aplicativo e a configuração do acesso à internet, verificar o funcionamento do sistema RICS.

O sistema apesar de ser concebido apenas como um teste pode ser utilizado para a transmissão de imagens através de um servidor HTTP, para tal basta que imagens sejam armazenadas no servidor HTTP utilizado.

3.4 Sistema EDGE2JAVA – Captura e Transmissão de Imagem entre um Computador e Unidade Móvel via Rede GSM/EDGE

O Sistema EDGE2JAVA é o resultado final deste projeto, sendo composto pelo Sistema CaTICIS e Sistema RICS adaptados para trabalharem em conjunto. Para o seu funcionamento é necessário instalar o aplicativo Edge2javaRX em um aparelho celular e a executar o código desenvolvido no Matlab Edge2javaTX no computador transmissor. Para a execução do aplicativo java Edge2javaRX no aparelho celular é necessário que este tenha java MIDP 2.0 e a API de mensagens sem fio 1.1 (*wireless messaging API 1.1 – WMA 1.1*) especificado pela JSR 205. Em alguns modelos de aparelhos celulares pode ser ainda necessário configurar um perfil de acesso à web, definindo parâmetros como o APN e senha de acesso.

Os aplicativos Edge2javaRX e Edge2javaTX e o código fonte desses aplicativos estão disponíveis no endereço <http://www.sj.cefetsc.edu.br/wiki/index.php/Edge2java>.

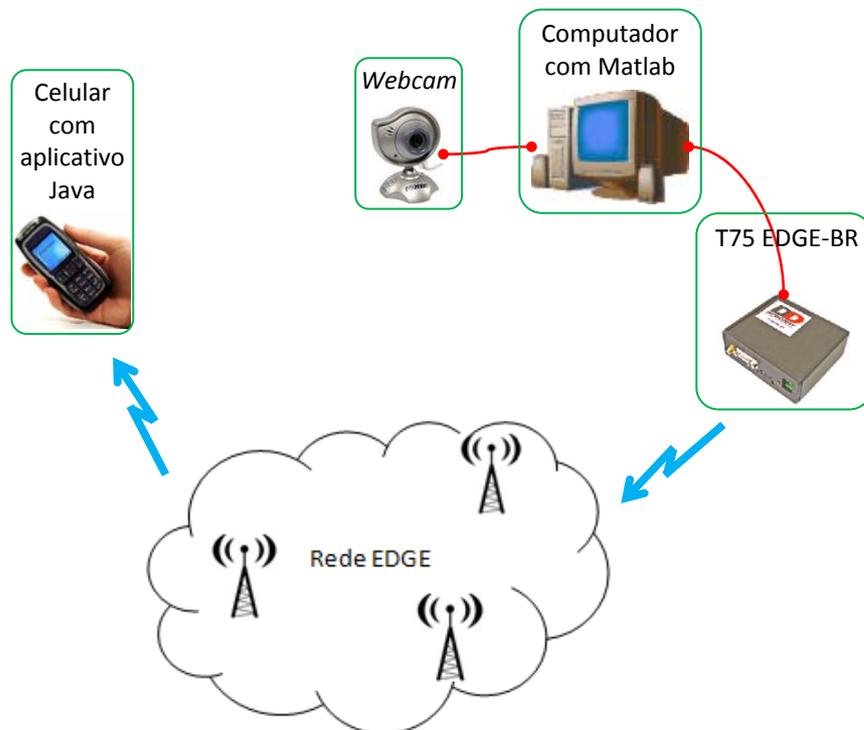


Figura 16 – Interligação dos equipamentos no Sistema EDGE2JAVA.

A Figura 16 apresenta a estrutura física usada no sistema EDGE2JAVA. O aplicativo Edge2javaRX desenvolvido é instalado num aparelho celular com capacidade de acesso à rede EDGE. O *software* Matlab e o aplicativo Edge2javaTX são instalados no computador ao qual a *webcam* e o terminal T75 EGE-BR são conectados.

A Figura 17 apresenta o funcionamento em blocos do sistema EDGE2JAVA. Neste diagrama alguns dos blocos são idênticos aos descritos nos sistemas CaTICIS e RICS. Nas seções a seguir são apresentadas apenas as alterações feitas nos sistemas CaTICIS e RICS para a implementação do sistema EDGE2JAVA.

3.4.1 Configurações Iniciais – Configuração do Módulo MC75

Nesta etapa, é feita a configuração da câmera de vídeo, da interface serial e do módulo MC75. As duas primeiras seguem o procedimento realizado no sistema CaTICIS, sendo que a resolução escolhida das imagens (160 X 120) é adequada à dimensão das telas da maioria dos celulares disponíveis. A configuração do MC75 consiste basicamente na ativação do servidor TCP no equipamento, que resulta no seu registro na rede GSM/EDGE. Para tal são usados os comandos AT de operação do MC75. A seguir cada etapa dessa configuração é descrita.

a) Registro na Rede GSM e Abertura do Servidor TCP

Como a rede GSM/EDGE das operadoras de telefonia celular possui uma faixa de endereçamento IP limitado, normalmente os dispositivos móveis não estão registrados o tempo todo na rede para se comunicar usando a rede de dados. Para permitir o acesso de aplicações móveis à rede GSM/EDGE é necessário executar o procedimento de registro na rede.

Para registrar o módulo MC75, devem ser configurados três perfis: o perfil de configuração de conexão com a internet (*internet connection setup profile* - SICS), o perfil de configuração de serviço de internet (*internet service setup profile* - SISS) e o perfil de abertura de serviço de internet (*internet service open* - SISO).

No perfil SICS são configuradas as informações de acesso à rede GSM/EDGE tais como ponto de acesso, usuário e senha. No perfil SISS são especificados os tipos de serviço, tais como soquetes clientes ou servidores e a porta de acesso à aplicação. No perfil SISO é feita a associação dos perfis SICS e SISS. Nesta forma podem-se ter vários perfis SISO formados por uma combinação de diferentes perfis SISS e SICS, o que permite oferecer diferentes tipos

de serviços aos usuários.

Após a criação de um SISO, o mesmo é ativado através de um comando AT, sendo permitida a ativação simultânea de vários perfis SISO. Uma vez ativado um SISO, o módulo MC75 efetua o registro na rede celular e ativa os serviços configurados. No sistema EDGE2JAVA o perfil SISO utilizado é um soquete servidor TCP com acesso à internet pela rede EDGE.

O módulo suporta até 4 conexões do tipo soquete, pois só é possível criar 4 perfis de serviço soquete. No sistema EDGE2JAVA, as conexões de clientes TCP são feitas de forma sequencial. Desta forma, um cliente só pode se conectar quatro vezes. Caso o limite de quatro conexões seja atingido, o aplicativo Edge2javaTX reinicia o módulo e reabre o servidor TCP.

3.4.2 Inicialização do Sistema

Após a abertura do servidor é enviado ao telefone celular, cadastrado para receber as imagens, um SMS que contém no corpo da mensagem o endereço IP de acesso ao servidor e a porta de destino da mensagem. Devido à necessidade de se escolher a porta de destino do SMS optou-se pela mensagem SMS no modo PDU (ver Seção 2.4 - Mensagens SMS e Protocolo PDU).

3.4.3 Abertura da conexão TCP do cliente com servidor

Ao receber o SMS na porta especificada, o aplicativo Edge2javaTX instalado no aparelho celular é iniciado. Após solicitar a confirmação do usuário, o endereço IP contido no corpo mensagem é salvo no aparelho celular através do sistema de gerenciamento de gravações (*record management system* - RMS). O aplicativo Edge2javaTX então inicia uma conexão cliente-servidor TCP com o MC75. Como o endereço IP é armazenado na memória do aparelho celular e não é perdido quando o aparelho celular é desligado, o aplicativo também poderá ser posteriormente executado de modo manual.

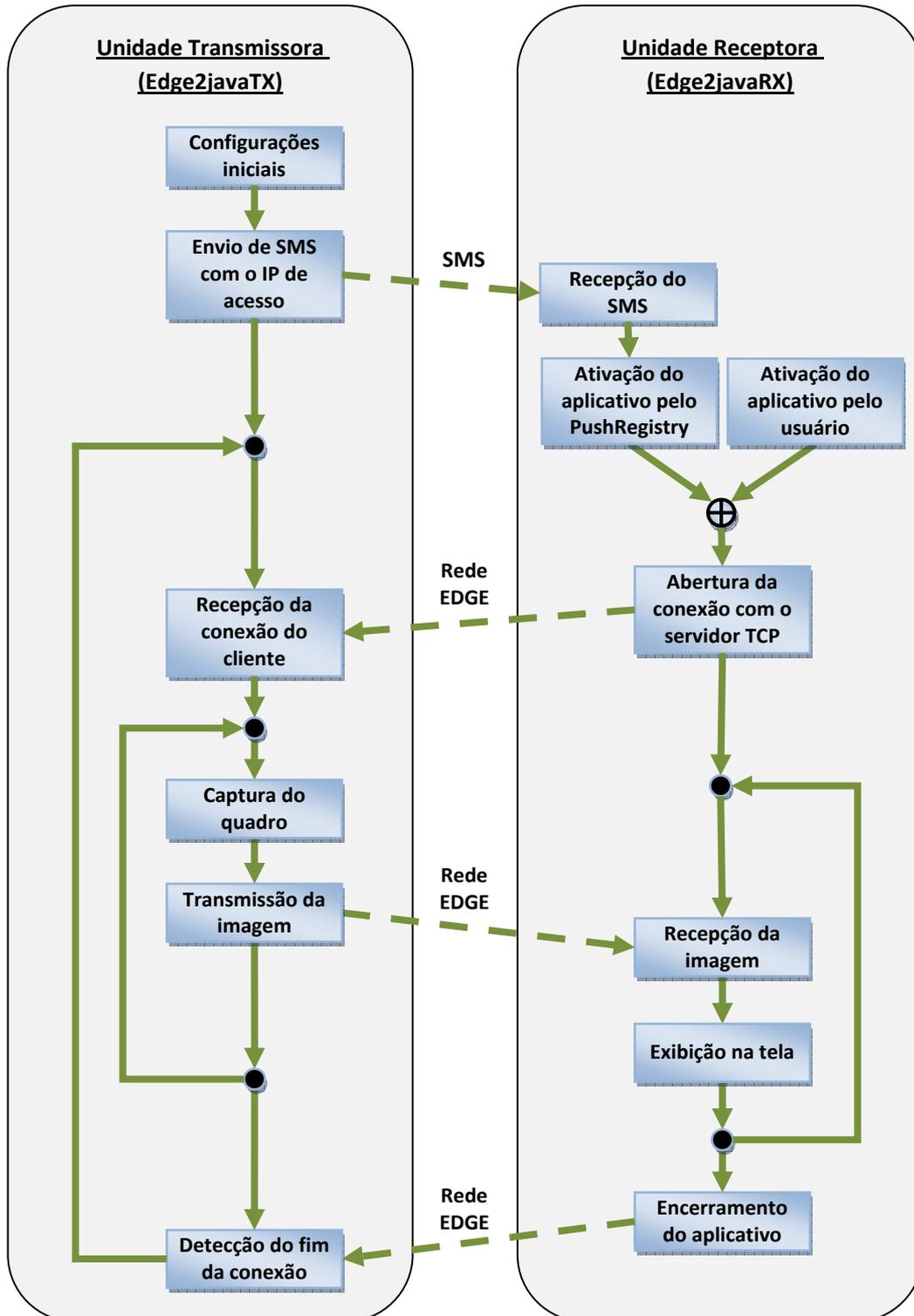


Figura 17 – Fluxograma de funcionamento do Sistema EDGE2JAVA.

3.4.4 Transmissão e Recepção da Imagem

Após a captura da imagem na câmera, ela deve ser processada e enviada para o MC75,

que a envia para o aparelho celular através da rede GSM/EDGE. O modo como a transmissão e a recepção das imagens são feitas será explicado a seguir em detalhes.

a) Transmissão da Imagem para o Módulo MC75

Após a captura e codificação da imagem, o Edge2javaTX envia os dados através da interface serial para o MC75, que realiza a sua transmissão. Como o *buffer* de entrada do MC75 tem apenas 1500 bytes, cada imagem JPEG com tamanho maior que isto é dividida em blocos de 1400 bytes pelo Edge2javaTX, como mostra a Figura 18.

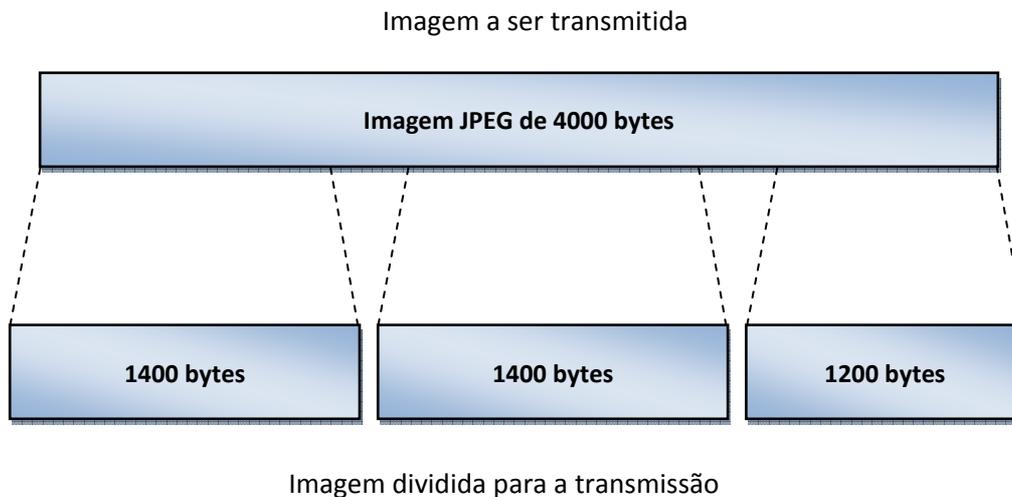


Figura 18 – Divisão em blocos da imagem codificada em JPEG para transmissão.

b) Encapsulamento dos Pacotes TCP/IP no Padrão EDGE

No módulo MC75 os dados são empacotados no quadro TCP/IP e transmitidos pela rede EDGE. Tanto o protocolo TCP/IP como a sua interface com a rede EDGE é totalmente transparente para o desenvolvedor de aplicativos e para o usuário do sistema. Caso haja redução na qualidade do sinal recebido pelo MC75, a sua classe EDGE é também reduzida automaticamente para reduzir a taxa de transmissão.

c) Recepção da Imagem e Exibição na Tela

O aplicativo java Edge2javaRX, ao receber os dados da rede EDGE, para bytes e monta os pacotes TCP recebidos em uma variável do tipo `Image`. Em seguida os dados são transformados em imagem JPEG pelo método `createImage`. Finalmente a imagem é exibida na tela através de um objeto `ImageItem`. Cada nova imagem recebida sobrescreve a anterior,

dando ao usuário a sensação de receber um vídeo. Este ciclo é repetido até que seja interrompido pelo usuário.

3.4.5 Problemas e Soluções Adotadas

Na proposta do projeto a imagem codificada seria transmitida através de bytes, como no sistema CaTICIS. No entanto, descobriu-se que o MC75 interpretava todos os bytes recebidos de acordo com a tabela ASCII, resultando em erros na transmissão dos bytes correspondentes aos caracteres de controle de fluxo dos dispositivos (17d, 18d, 19d e 20d). Isto ocorre pois esses caracteres são tratados pelo MC75 como comandos de controle de fluxo, resultando num travamento da comunicação entre o MC75 e o microcomputador. Como solução para este problema foi desenvolvida a codificação “hexadecimal codificado em ASCII” (HCA).

A codificação HCA converte o valor inteiro de cada byte da imagem JPEG para seus respectivos valores em hexadecimal. Em seguida o valor hexadecimal é dividido em dois algarismos, que são convertidos em caracteres da tabela ASCII, conforme ilustra a Figura 19. Desta forma, por exemplo, para transmitir o byte com valor decimal 093d, são transmitidos na seu lugar dois bytes com os valores decimais de 053d e 068d, respectivamente. O processo de decodificação no aparelho celular é inverso do processo ao mostrado na Figura 19.

Desta forma, limitamos os caracteres ASCII a serem enviados como 0-9, e A-F. Esta estratégia foi usada para evitar os travamentos do MC75, mas reduz a taxa efetiva de transmissão pela metade, já que para cada 8 bits da imagem são transmitidos 16 bits. Como uma alternativa melhor ainda não foi encontrada, este formato HCA continua sendo usado.

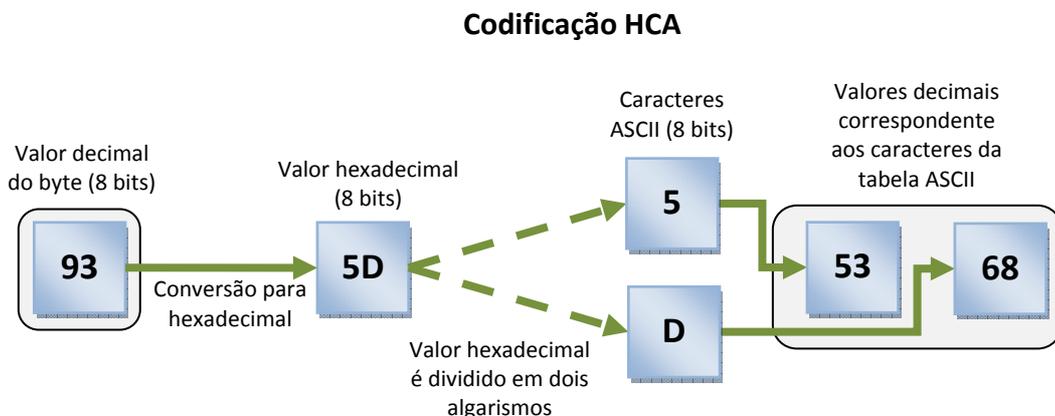


Figura 19 – Codificação HCA.

Em relação à rede celular foram encontrados diversos problemas relacionados com a falta

de resposta a solicitações de serviços, baixos níveis de sinais, e a demora na transmissão de mensagens de SMS do MC75 para a rede GSM. Foram tentados vários contatos com o suporte técnico das operadoras TIM, Brasil Telecom e Claro para obter esclarecimentos, mas sem sucesso, pois as respostas obtidas não eram técnicas.

Para a obtenção do endereço IP da rede celular na criação do servidor TCP apenas a operadora Claro forneceu sempre endereços públicos. A operadora TIM às vezes fornecia endereços públicos e outras vezes privativos. Já para cartões SIM da operadora Brasil Telecom os endereços IP sempre eram privativos. Diante do exposto acima, foi optado por utilizar um cartão SIM pré-pago da operadora Claro no MC75, e um cartão de qualquer operadora de telefonia celular no aparelho celular a ser testado. Com essa opção, não foi necessário adquirir nenhum serviço adicional para obtermos um endereço IP público da rede.

Outra dificuldade encontrada nos testes do Sistema EDGE2JAVA foi que alguns aparelhos celulares inicialmente não conseguiam se conectar ao MC75 através do aplicativo Edge2javaRX. Foi descoberto que a causa disso era a necessidade de configurar um perfil de conexão do celular para o aplicativo poder acessar a internet. Neste perfil são configurados parâmetros de conexão, como senha e APN. Cada operadora de telefonia celular tem um APN e uma senha próprios. O problema foi solucionado através de um site (JMOBI, 2008) que explica como configurar os perfis de conexão para vários modelos de aparelho celular e fornece o APN e a senha de cada operadora. A solução foi testada em alguns aparelhos celulares, obtendo-se sucesso nestes.

4 Conclusões e Comentários Finais

O presente trabalho apresentou três sistemas. Os dois primeiros (CaTICIS e RICS) foram utilizados para testes, e serviram de base para indicar o que seria necessário pesquisar além do que se esperava inicialmente, em termos de rede celular, programação JME e captura e processamento de imagens. O terceiro sistema (EDGE2JAVA) é o que realmente representa a proposta.

O sistema atingiu o esperado, conseguindo-se visualizar em um aparelho celular imagens capturadas por uma câmera em tempo real. Porém as taxas obtidas ficaram aquém dos valores teóricos e esperados. Esta limitação da rede GSM/EDGE repercutiu em uma baixa taxa de quadros das imagens recebidas, não sendo recomendada quando seja primordial a visualização de imagens com baixo tempo de espera entre duas imagens subsequentes.

Concluída a implementação, foram realizados alguns levantamentos de custos de uso do sistema e da taxa de transmissão das seqüências de imagem. Eles não puderam ser exaustivamente testados, devido à falta de tempo, e ausência de suporte das operadoras, uma vez que o servidor do sistema EDGE2JAVA apenas funcionou com cartões SIM da operadora Claro. O funcionamento do sistema na execução de um *handoff*⁴ no aparelho celular foi verificado, não havendo erros que causassem o encerramento da conexão. A seguir são relatados os resultados obtidos.

Para iniciar automaticamente o aplicativo no aparelho celular utilizou-se a classe java chamada PushRegistry. Assim poderia se acionar o aplicativo através da abertura de uma de conexão, mas tivemos problemas para saber o IP do aparelho celular. Por isso optamos por enviar um SMS para uma porta de destino específica no dispositivo móvel.

Foi necessário implementar o formato de mensagem no modo PDU para suprir a necessidade de endereçamento de porta do SMS. Foi constatado com isso que não são todos os aparelhos celulares disponíveis no mercado que têm suporte a esse modo de envio de SMS,

⁴ Handoff é o procedimento empregado em redes sem fio para tratar a transição de um aparelho celular de uma célula para outra de forma transparente ao utilizador.

pois em alguns aparelhos celulares testados esse formato de mensagem não chegava ao seu destino. Procuramos contato técnico com operadoras para comentar o ocorrido, e obtendo respostas genéricas, concluímos que os suportes técnicos das empresas não conseguem explicar o que ocorre dentro de seus equipamentos, em se tratando de protocolos de comunicação e roteamento de dados.

4.1 Custo do uso do sistema

Os custos para a transmissão da imagem podem ser classificados em custos fixos e variáveis. Como custo fixo considera-se o custo do envio do SMS pelo servidor Edge2javaTX. A Tabela 5 trás uma relação do custo do envio de mensagens SMS para cada operadora. Em custos variáveis estão incluídos os valores gastos com a transmissão das imagens pelo servidor e os valores gastos com a recepção das imagens no aparelho celular. A Tabela 6 apresenta um resumo do custo da transmissão de dados para as operadoras da rede celular disponíveis na grande Florianópolis. Ambas as tabelas são informações do dia 04 de setembro de 2008 (TIM, 2008), (CLARO, 2008), (BRASIL TELECOM, 2008).

Tabela 5 – Custo do envio de mensagens SMS

| Operadora | Custo por mensagem (R\$) |
|----------------|--------------------------|
| Tim | 0,39 |
| Claro | 0,35 |
| Brasil Telecom | 0,38 |

Tabela 6 – Custo da transmissão de dados

| Operadora | Tipo Tráfego | Custo mensal | Custo por imagem ^{**} (centavos de real) |
|----------------|----------------------|--------------|--|
| TIM | Ilimitado até 7 Mbps | R\$ 159,90 | Depende do tráfego |
| | Ilimitado até 1 Mbps | R\$ 99,90 | Depende do tráfego |
| | Limitado em 1 GB | R\$ 69,90 | 0,053 |
| | Limitado em 250 | R\$ 39,90 | 0,125 |
| | Limitado em 40 MB | R\$ 29,90 | 0,584 |
| Claro | Limitado em 2 GB | R\$ 99,90 | 0,038 |
| | Limitado em 500 | R\$ 79,90 | 0,125 |
| | Limitado em 100 | R\$ 49,90 | 0,390 |
| | Limitado em 40 MB | R\$ 39,90 | 0,779 |
| Brasil Telecom | Ilimitado | * | 24 |

* Até o dia 04/09/08, a operadora Brasil Telecom não fornecia um plano específico para transmissão de dados com a tecnologia GSM/GPRS/EDGE.

** Tamanho médio das imagens de 4 kBs

Observando as tabelas verifica-se que, tanto para o lado cliente quanto o servidor, a operadora Claro se mostrou a opção mais econômica, utilizando o plano Limitado em 2 GB. O limite de 2 GB aplicado é mais que suficiente para a transmissão das imagens, permitindo, para imagens de tamanho médio de 4 kB e codificação HCA, o envio de aproximadamente 250 mil imagens. Como o sistema suporta até quatro conexões sem precisar de um novo envio de SMS, o custo fixo de R\$ 0,35 para a operadora Claro pode ser dividido por quatro, totalizando R\$ 0,0875 de custo fixo por conexão.

4.2 Taxa de Transmissão

De acordo com a especificação do sistema celular GSM/EDGE, esperava-se uma taxa de transmissão próxima aos 473,6 kbps (SVERZUT, 2005), a qual permitiria a transmissão de até sete quadros por segundo considerando imagens com 4 kB cada, e codificação HCA. Este número foi obtido através do seguinte cálculo: $473,6 \text{ kbps} / (4 \text{ kB} \times 8 \text{ bits/byte} \times 2 \text{ HCA})$. Segundo testes realizados, a rede GSM/EDGE apresenta uma grande variação de taxa, gastando para a transmissão de uma imagem, desde três segundos até nove minutos. Como resultado, uma taxa de transmissão final próxima a 1 kbps foi obtida.

Para reduzir o tempo de espera entre duas imagens subseqüentes, foi reduzida a resolução da imagem dos 160 x 120 para 54 x 40 *pixels*, resultando em, na média, imagens de 1,5 kB. É importante ressaltar que mesmo com essa redução, o tempo de atraso continua sofrendo grandes variações.

4.3 Implementações Alternativas

Um método alternativo para o envio das imagens para o aparelho celular é utilizar um servidor HTTP, nos quais as imagens capturadas pela câmera seriam transmitidas pelo módulo para este servidor. Este servidor ficaria sempre conectado à internet, disponibilizando as imagens no celular. Uma vantagem desta implementação é que, usando-se um servidor intermediário, é facilitada a implementação de um sistema multiusuário. Uma desvantagem é a adição dos custos relacionados à manutenção do servidor ao custo final da aplicação.

Outra possibilidade seria a visualização das imagens capturadas diretamente do servidor através de uma página *web*, sem a necessidade do aplicativo java instalado no aparelho celular. Isto representa a possibilidade de se utilizar qualquer equipamento com acesso e visualização do conteúdo da internet, como computadores, *Ipods* e os próprios aparelhos celulares, para obtenção das imagens capturadas pela câmera. Como problemas da alternativa citada, têm-se a falta de segurança envolvida, já que para a recepção das imagens basta que o endereço IP seja conhecido. Qualquer pessoa teria então acesso às informações que, dependendo da aplicação, sejam sigilosas.

4.4 Aplicações do Sistema Desenvolvido

O trabalho desenvolvido pode ser incorporado a sistemas de vigilância eletrônica e monitoramento de ambientes, para possibilitar aos usuários a visualização em tempo real em um aparelho celular das imagens capturadas pelas câmeras.

O sistema EDGE2JAVA pode também ser adaptado à transmissão de dados, para a utilização em aplicações de telemetria (medições à distância sem a necessidade de uma pessoa para realizar a tarefa). Como exemplo, medições de temperatura de um termômetro, medições de rotações por minuto de um motor, medições de níveis de tensão de um transformador, etc.

Outra aplicação seria a construção de portais de informação de vídeo, na qual câmeras captando imagem em diversos lugares tais como, praias, festas, pontos turísticos, ruas e avenidas, permitiriam aos usuários a visualização de tais eventos.

4.5 Trabalhos Futuros

Para elevar a taxa de transmissão alcançada pode-se trocar o módulo EDGE utilizado por outro com suporte a tecnologias mais recentes. O acesso em pacote com elo descendente em alta velocidade (*high speed downlink packet access* - HSDPA) é um novo protocolo de telefonia móvel, considerado de geração 3,5. Ele atinge taxas teóricas máximas de 14,4 Mbps (3GPP, 2008).

Pode-se também implantar um canal de retorno no sistema EDGE2JAVA, possibilitando assim, a interação entre o aparelho celular e o servidor. Isso permitiria, por exemplo, o controle da câmera, o disparo de alarmes sonoros, o acendimento de luzes, o desligamento de equipamentos, entre outras possibilidades.

Outro desenvolvimento do projeto pode ser o de embarcar todo o processo realizado atualmente por um computador em um “arranjo de portas programável em campo” (*field programmable gate array* - FPGA) (FPGA, 2008), que é um dispositivo utilizado no processamento de informações digitais. Isto reduziria o custo da aplicação, e ao mesmo tempo, diminuiria consideravelmente o tamanho dos equipamentos envolvidos.

Um melhoramento do aplicativo Edge2javaRX poderia ser realizado, acrescentando uma opção para o usuário salvar no aparelho celular as imagens que estão sendo exibidas. Esta funcionalidade poderia ser utilizada em sistemas de monitoramento, servindo como prova da ocorrência de um determinado evento.

Referências Bibliográficas

3GPP. 3GPP TS 25.308 V8.2.0 (2008-05) - 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; High Speed Downlink Packet Access (HSDPA); Overall description; Stage 2 (Release 8). 2008.

3GPP. 3GPP TS 27.007 V6.3.0 (2003-06) – 3rd Generation Partnership Project; Technical Specification Group Terminals; AT command set for User Equipment (UE) (Release 6). 2003.

ALOE, Greg. dbloop.m: Dynamically enter debug mode during a loop. 2006. Disponível em: <<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=9736&objectType=file>>. Acesso em: Maio 2008.

BRASIL TELECOM. Custo Básico dos Dados. Disponível em: <<http://gsm.brasiltelecom.com.br/gsm/site/home/ServicosHome.do?acao=detalhes&srvPk=E DGE>>. Acesso em 28 Ago. 2008.

BRUCE, Robert, V. Bell: Alexander Graham Bell and the Conquest of Solitude Publicado por Cornell University Press, 1990. 564 p.

CLARO. Planos de Dados da Claro. Disponível em: <<http://www.claro.com.br/portal/planoinside.do?method=showDadosGSM>>. Acesso em 28 Ago. 2008.

Eclipse. Eclipse – an open development platform. Site oficial do Eclipse. Disponível em: <<http://www.eclipse.org/>>. Acesso em: 06 Ago. 2008.

EIA (Electronics Industries Association). EIA Standard RS-232-C Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Data Interchange. Agosto 1969. Reimpresso em Telebyte Technology Data Communication Library. Greenlawn NY, 1985.

ETSI. TS 100 900 V7.2.0 (1999-07) Digital cellular telecommunications system (Phase 2+);

Alphabets and language-specific information (GSM 03.38 version 7.2.0 Release 1998). 1999.

ETSI; ETSI TS 100 901 V7.5.0 (2001-12) Technical Specification: Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point-to-Point (PP); (3GPP TS 03.40 version 7.5.0 Release 1998). Zurique: ETSI, 2001.

FLANAGAN, David. Java In A Nutshell – A Desktop Quick Reference. O’Reilly, 2005. 1254 p.

FPGA. FPGA – Wikipédia, a enciclopédia livre. Disponível em: <http://en.wikipedia.org/wiki/Field-programmable_gate_array>. Acesso em: 23 Ago. 2008.

GSM, W. GSM World. GPRS Class Type. Disponível em: <<http://www.gsmworld.com/technology/gprs/class.shtml>>. Acesso em: 06 Ago. 2008.

HOPSON, K, C; INGRAM, Stephen, E. Developing Professional Java Applets. Publicado por Sams.net, 1996. 528 p.

ISO; ITU-T. ISO/IEC IS 10918-1 | ITU-T Recommendation T.81: Information technology – digital compression and coding of continuous-tone still images – requirements and guidelines. Zurique: ISO|ITU-T, 1992.

JMOBI. Configuração GPRS. Disponível em: <<http://www.jmobi.com.br/jmobi-gprs/>>. Acesso em: 19 Ago. 2008.

JOHNSON, T. M. Java para Dispositivos Móveis: Desenvolvendo Aplicações com J2ME. São Paulo: Novatec, 2007. 334 p.

JORNAL CIDADE. Jornal Cidade online. Disponível em: <<http://jornalcidade.uol.com.br/paginas.php?id=30705>>. Acesso em 23 Ago. 2008.

KUROSE, James F. Computer networking : a top-down approach featuring the internet. Segunda edição. Boston: Addison Wesley, 2003. 752p.

NetBeans. NetBeans IDE 6.1. Site oficial do NetBeans. Disponível em: <<http://www.netbeans.org/>>. Acesso em: 06 Ago. 2008.

PALUDO, LAURIANA. Um Estudo Sobre As Tecnologias Java De Desenvolvimento De Aplicações Móveis. 2003. 118 p.

SIEMENS. Wireless Module MC75. Disponível em: <<http://www.siemens.com.br/templates/produto.aspx?channel=250&produto=16248>>. Acesso em: 20 Ago. 2008.

SUN. Javax.microedition.io: Class PushRegistry. Classe PushRegistry. Disponível em: <<http://java.sun.com/javame/reference/apis/jsr118/javax/microedition/io/PushRegistry.html>>. Acesso em: 15 jul. 2008.

SUN. The Java ME Platform. Plataforma JME. Disponível em: <<http://java.sun.com/javame/index.jsp>>. Acesso em: 15 jul. 2008.

SVERZUT, J. U. Redes GSM, GPRS, EDGE e UMTS - Evolução a Caminho da Terceira Geração (3G). São Paulo: Erica, 2005. 454 p.

TAURION, Cezar. Internet Móvel Tecnologias, Aplicações e Modelos. Rio de Janeiro-RJ. Editora Campus, 2002.

TELECO. O Modelo OSI de Interconexão de Sistemas Abertos. Disponível em: <<http://www.teleco.com.br/tutoriais/tutorialosi/default.asp>>. Acesso em: 09 jun. 2008.

TIM. Custo Básico dos Dados. Disponível em: <http://www.tim.com.br/portal/site/PortalWeb/menuitem.d884721d78543193b5508210703016a0/?vgnextoid=719ae20042070110VgnVCM100000a22e700aRCRD&wfe_pweb_oid=7ad97200dbce8110VgnVCM100000a22e700a____&wfe_pweb_area=56&wfe_pweb_estado=24&>. Acesso em 28 Ago. 2008.

TIM. TIM Web Banda Larga. Disponível em: <http://www.tim.com.br/portal/site/PortalWeb/menuitem.8419ab3f80fb36423da5faa7703016a0/?vgnextoid=74cafd4cb3a43110VgnVCM100000a22e700aRCRD&wfe_pweb_area=56&wfe_pweb_estado=24&&wfe_pweb_oid=cf35fd4cb3a43110VgnVCM100000a22e700a____&>. Acesso em: 28 Ago. 2008.

Lista de Abreviaturas e Siglas

| | | | |
|-----------------|--|-----------|---|
| AMS | Software de Controle de Aplicação (<i>Application Management Software</i>) | DNS | Serviço de Nomes de Domínio (<i>Domain Name Service</i>) |
| APN | Nome do Ponto de Acesso (<i>Access Point Name</i>) | EDGE | Taxas de Dados Melhorada para a Evolução do GSM (<i>Enhanced Data Rates for GSM Evolution</i>) |
| ARP | Protocolo de Resolução de Endereço (<i>Address Resolution Protocol</i>) | EDGE2JAVA | Sistema de Captura e Transmissão de Imagem entre um Computador e Unidade Móvel via Rede GSM/EDGE |
| ASCII | Código Americano Padrão para troca de Informações (<i>American Standard Code for Information Interchange</i>) | EIA | <i>Electronics Industries Association</i> |
| AT | Comandos de Atenção (<i>Attention Commands</i>) | ETSI | Instituto Europeu de Padrões de Telecomunicações (<i>European Telecommunications Standards Institute</i>) |
| bps | Bits por Segundo | FDDI | Interface de Dados de Fibra Distribuída (<i>Fiber distributed data interface</i>) |
| Bps | Bytes por segundo | FPGA | Arranjo de Portas Programáveis em Campo (<i>Field-Programmable Gate Array</i>) |
| BSD | Distribuição de Software Berkeley (<i>Berkeley Software Distribution</i>) | FTP | Protocolo de Transferência de arquivos (<i>file transfer protocol</i>) |
| CaTICIS | Sistema de Captura e Transmissão de Imagem entre dois Computadores via Interface Serial | GB | Giga Bytes |
| CDC | Configuração de Dispositivo Conectado (<i>Connected Device Configuration</i>) | GIF | Formato de Intercâmbio de Gráficos (<i>Graphics Interchange Format</i>) |
| CDMA | Acesso múltiplo por Divisão de Código (<i>Code Division Multiple Access</i>) | GPRS | Serviço De Rádio De Pacote Geral (<i>General Package Radio Service</i>) |
| CDMA2000 1xRTT | – Acesso Múltiplo por Divisão de Código, Tecnologia de Transmissão de Rádio, Uma Vez (<i>Code Division Multiple Access, 1 times Radio Transmission Technology</i>) | GSM | Sistema Global para Comunicações Móveis (<i>global system for mobile communications</i>) |
| CDMA2000 1xEVDO | – Acesso Múltiplo por Divisão de Código, Otimização da Evolução dos Dados, Uma Vez (<i>Code Division Multiple Access, 1 times Evolution-Data Optimized</i>) | HCA | Hexadecimal Codificado em ASCII |
| CLDC | Configuração de Dispositivo Conectado Limitado (<i>Connected Limited Device Configuration</i>) | HSDPA | Acesso em Pacote com Elo Descendente em Alta Velocidade (<i>High Speed Downlink Packet Access</i>) |
| CSD | Dados por Comutação de Circuitos (<i>Circuit Switched Data</i>) | HTTP | Protocolo de Transferência de Hipertexto (<i>Hypertext Transfer Protocol</i>) |
| CTS | Pronto para Enviar (<i>Clear to Send</i>) | ICMP | Protocolo de Mensagens de Controle da Internet (<i>Internet Control Message Protocol</i>) |
| DA | Endereço de Destino (<i>Destination Address</i>) | IDE | Ambiente de Desenvolvimento Integrado (<i>Integrated Development Environment</i>) |
| DCS | Esquema de Codificação de Dados (<i>Data Coding Scheme</i>) | | |
| DCT | Transformada Discreta do Cosseno (<i>Discrete Cosine Transform</i>) | | |

| | | | |
|-------|---|------|---|
| IP | Protocolo de Internet (<i>Internet Protocol</i>) | PP | Perfil Pessoal (<i>Personal Profile</i>) |
| ISO | Organização Internacional para Padronização (<i>International Organization for Standardization</i>) | PSK | Modulação por Deslocamento de Fase (<i>Phase Shift Keying</i>) |
| ITU-T | União Internacional de Telecomunicações - Padronização de Telecomunicações (<i>International Telecommunication Union - Telecommunication Standardization</i>) | RAM | Memória de Acesso Aleatório (<i>Random Access Memory</i>) |
| JAD | Descrição de Aplicativo Java (<i>Java Application Descriptor</i>) | RARP | Protocolo de Resolução Reversa de Endereços (<i>Reverse Address Resolution Protocol</i>) |
| JAR | Arquivo Java (<i>Java Archive</i>) | RD | Rejeição de Duplicatas (<i>Reject Duplicate</i>) |
| JEE | Edição para Empresas do Java (<i>Java Enterprise Edition</i>) | RGB | Vermelho Verde Azul (<i>Red Green Blue</i>) |
| JPEG | Grupo de Especialistas em Fotografia (<i>Joint Photographic Experts Group</i>) | RICS | Sistema de Recepção de Imagem no Celular de um Servidor http |
| JSE | Edição Padrão do Java (<i>Java Standard Edition</i>) | RMS | Sistema de Gerenciamento de Gravações (<i>Record Management system</i>) |
| JSR | Requisitos de Especificação Java (<i>Java Specification Request</i>) | RP | Caminho de retorno (<i>Reply Path</i>) |
| JVM | Máquina Virtual Java (<i>Java Virtual Machine</i>) | RTP | Protocolo de Transporte em Tempo Real (<i>Real-time Transport Protocol</i>) |
| kB | Kilo Byte | RTPC | Rede de Telefonia Pública Comutada |
| kbps | Kilo bits por segundo | RTS | Requisição para Enviar (<i>Request to Sent</i>) |
| KVM | Máquina Virtual Kilobyte (<i>Kilobyte Virtual Machine</i>) | SC | Centro de Serviço (<i>Service Center</i>) |
| MB | Mega bytes | SCA | Endereço do Centro de Serviço (<i>Service Center Address</i>) |
| Mbps | Mega bits por segundo | SICS | Perfil de Configuração de Conexão com a Internet (<i>Internet Connection Setup Profile</i>) |
| MIDP | Perfil de Dispositivo móvel (<i>Mobile Information Device Profile</i>) | SIM | Módulo de Identificação do assinante (<i>Subscriber Identity Module</i>) |
| MPEG | Grupo de Especialistas em Imagem em Movimento (<i>Moving Pictures Experts Group</i>) | SISO | Perfil de Abertura de Serviço de Internet (<i>Internet Service Open</i>) |
| MR | Referência da mensagem (<i>Message Reference</i>) | SISS | Perfil de Configuração de Serviço de Internet (<i>Internet Service Setup Profile</i>) |
| MTI - | Indicador de Tipo da Mensagem (<i>Message Type Indicator</i>) | SMMO | Mensagem Curta Originada pelo Móvel (<i>Short Message Mobile Originated</i>) |
| OSI | Interconexão de Sistemas Abertos (<i>Open Systems Interconnection</i>) | SMMT | Mensagem Curta Terminada no Móvel (<i>Short Message Mobile Terminated</i>) |
| PDA | Assistente Pessoal Digital (<i>Personal Digital Assistant</i>) | SMS | Serviço de Mensagens Curtas (<i>Short Message Service</i>) |
| PDU | Protocolo de Unidade de Dados (<i>Protocol Data Unit</i>) | SMTP | Protocolo Simples de Transferência de Cartas (<i>Simple Mail Transfer Protocol</i>) |
| PID | Identificação do Protocolo (<i>Protocol Identifier</i>) | | |
| PNG | Gráficos Portáteis para Rede (<i>Portable Network Graphics</i>) | | |
| POP3 | Protocolo de Correio versão 3 (<i>Post Office Protocol</i>) | | |

| | | | |
|------|--|------|--|
| SNMP | Protocolo Simples de Gerência de Rede (<i>Simple Network Management Protocol</i>) | UDP | Protocolo de Datagramas (<i>User Datagram Protocol</i>) |
| SP2 | SP2 – Pacote de Serviços 2 (<i>Service Pack 2</i>) | UMTS | Sistema Universal de Comunicações Móveis (<i>Universal Mobile Telecommunications System</i>) |
| SRR | Requisição de Relatório de Estado (<i>Status Report Request</i>) | URL | Localizador Uniforme de Recursos (<i>Uniform Resource Locator</i>) |
| TCP | Protocolo de Controle de Transmissão (<i>Transmission Control Protocol</i>) | USB | Barramento Serial Universal (<i>Universal Serial Bus</i>) |
| TFTP | Protocolo Trivial de Transferência de Arquivo (<i>Trivial File Transfer Protocol</i>) | VP | Período de Validade (<i>Validity Period</i>) |
| UD | Dados do Usuário (<i>User Data</i>) | VPF | Formato do Período de Validade (<i>Validity Period Format</i>) |
| UDH | Cabeçalho dos Dados do Usuário (<i>User Data Header</i>) | WMV | Mídia de Vídeo do Windows(<i>Windows Media Video</i>) |
| UDHI | Identificador do Cabeçalho dos Dados do Usuário (<i>User Data Header Identifier</i>) | | |
| UDL | Comprimento dos Dados do Usuário (<i>User Data Length</i>) | | |