

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE SISTEMAS DE INFORMAÇÃO

Automação e Gestão de Serviços de Restaurante usando
Tecnologia Wireless em um PDA

AUTOR: Felipe Simas Silva

Florianópolis, Janeiro de 2007

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Automação e Gestão de Serviços de Restaurante usando
Tecnologia Wireless em um PDA**

AUTOR: Felipe Simas Silva

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do grau
de Bacharel em Sistemas de Informação.

Orientador: João Bosco Manguiera Sobral

Banca Avaliadora: José Leomar Todesco

Lucas Guardelben

Florianópolis, Janeiro de 2007

"A diferença fundamental entre o homem comum e o guerreiro, é que o guerreiro encara tudo como desafio, enquanto o homem comum encara tudo como bênção ou maldição."

Carlos Castañeda, antropólogo

RESUMO

O objetivo deste trabalho é automatizar o processo de Gestão de Serviços em um restaurante. Para alcançar essa meta é feito um estudo comparativo entre as mais variadas tecnologias existentes na área de comunicação sem fio, *PDA*s e *Data Warehouse*. A partir deste estudo é possível ter um conhecimento do que utilizar e como desenvolver um sistema capaz de auxiliar nos processos de um restaurante. O aplicativo desenvolvido utiliza a plataforma *.NET Compact Framework* em um Pocket PC. Estas tecnologias foram escolhidas devido à sua viabilidade e praticidade. A comunicação sem fio é feita através da rede IEEE 802.11b, este padrão foi o escolhido porque é o mais utilizado, possui um baixo custo e tem um alto alcance. Outra tecnologia usada no desenvolvimento deste trabalho é *Data Warehouse*, ela pode-se responder muitas questões que dificilmente são respondidas com banco de dados operacionais simples. Após o desenvolvimento desse projeto pode-se ter uma melhora considerável no atendimento dos clientes de um restaurante, pois ocorre um aumento da velocidade na entrega dos pedidos e através de relatórios e gráficos as informações relativas ao restaurante tornam-se muito mais claras e visíveis.

Palavras-chave: Wi-fi, Bluetooth, Data Warehouse, IEEE 802.11, Pocket PC, Palm Top, J2ME, .NET Compact Framework

SUMÁRIO

RESUMO	4
LISTA DE FIGURAS	7
LISTA DE TABELAS	9
GLOSSÁRIO	10
1. Introdução	12
1.1. Objetivo geral.....	13
1.2. Objetivos Específicos.....	13
1.3. Justificativa	14
1.4. Estrutura do trabalho.....	15
2. Tecnologias que dão suporte aos PDAs	16
2.1. Comunicação sem fio (Wireless)	18
2.1.1. Técnicas de modulação	18
2.1.1.1. Frequency Hopping Spread Spectrum (FHSS)	19
2.1.1.2. Direct Sequence Spread Spectrum (DSSS).....	19
2.1.1.3. Orthogonal Frequency Division Multiplexing (OFDM).....	21
2.1.2. Padrões Wireless	21
2.1.2.1. Bluetooth.....	22
2.1.2.2. IEEE 802.11 (Wi-Fi).....	25
2.2. Programação em PDAs	27
2.2.1. J2ME (Java 2 Micro Edition).....	27
2.2.2. NET Compact Framework	35
2.2.3. Comparativo J2ME X .NET Compact Framework.....	39

3. Desenvolvimento da Aplicação	40
3.1. Pocket PC (Interface e funções).....	43
3.2. Data Warehouse	46
3.2.1. Modelagem Multidimensional	50
3.2.2. Carga de dados	52
3.2.3. Front end (apresentação dos dados).....	53
4. Conclusão	61
5. Referências Bibliográficas	62
Anexos A – Especificação da Modelagem Multidimensional	63
Anexos B – Dados contidos nas tabelas de dimensão usado na geração dos gráficos	65
Anexos C – Artigo	68

LISTA DE FIGURAS

Figura 1 – Palmone (Tungsten T5).....	17
Figura 2 – Pockect PC (Pocket PC h4150).....	17
Figura 3 – Espalhamento do sinal na DSSS.....	20
Figura 4 – Piconet e Scatternet.....	24
Figura 5 – Arquitetura J2ME.....	28
Figura 6 – Relação da configurações no Java.....	29
Figura 7 – Exemplo de um MIDlet.....	32
Figura 8 – J2ME Wireless Toolkit 1.0.4.....	33
Figura 9 – Sun ONE Studio Mobile Edition 4.....	34
Figura 10 – Arquitetura .net CF.....	37
Figura 11 – Smart Device Extensions for Visual Studio .NET.....	38
Figura 12 – Chamada de um Web Service através do VB.NET.....	45
Figura 13 – Exemplo de um Web Service.....	46
Figura 14 – Estrutura do Sistema.....	47
Figura 15 – Área 1 - Restaurante.....	48
Figura 16 – Área 2 - Restaurante.....	48
Figura 17 – Tela Check-in.....	49
Figura 18 – Tela Cozinheiro.....	49
Figura 19 – Tela Visualizar Pedidos.....	49
Figura 20 – Tela Realizar Pedidos.....	49
Figura 21 – Esquema Estrela (Star Schema).....	51
Figura 22 – Total vendido por hora e dia da semana.....	54
Figura 23 – Demanda das comidas em relação aos meses no ano de 2005.....	55

Figura 24 – Influência do tempo no consumo de bebida e comida.....	56
Figura 25 – Movimento nos feriados dos últimos 3 anos.....	58
Figura 26 – Consumo de bebida quando somente 1 casal ocupa a mesa.....	59
Figura 27 – Consumo de bebida quando um casal e duas crianças ocupam a mesa.....	59
Figura 28 – Tipos de clientes que freqüentaram o restaurante em 2006.....	60

LISTA DE TABELAS

Tabela 1 – Comparação entre Padrões IEEE 802.11.....	26
Tabela 2 – Comparação entre J2ME e .Dot CF.....	39
Tabela 3 – Quantidade de bebidas vendida em dia de sol.....	55
Tabela 4 - Classificação dos garçons pela eficiência.....	56
Tabela 5 - Clientes mais valiosos.....	59

GLOSSÁRIO

.NET CF: Dotnet Compact Framework

BD: Banco de Dados

BPSK: Binary Phase Shift Keying

CLR: Common Language Runtime

DSSS: Direct Sequence Spread Spectrum

DW: Data warehouse

FCC: Federal Communication Commission

FH-CDMA: Frequency Hopping - Code-Division Multiple Access

FHS: Frequency Hopping-Synchronization

FHSS: Frequency Hopping Spread Spectrum.

IEEE: Institute of Electrical and Electronics Engineers

ISM: Industrial, Scientific, Medical

J2ME: Java 2 Platform Micro Edition

J2SE: Java 2 Platform Standard Edition

JCE: Java Cryptography Extension

JCP: Java Community Process

KB: Kilobyte

OFDM: Orthogonal Frequency Division Multiplexing

OLAP: On Line Analytical Processing

OLTP: On Line Transactional Processing

SO: Sistema Operacional

PC: Personal Computer

PDA: Personal Digital Assistant

PIM: Personal Information Manager

QPSK: Quadrature Phase Shift Keying

SQL: Structured Query Language

SSM: Spread Spectrum Modulation

VB: Visual Basic

VM: Virtual Machine

Wi-Fi: Wireless Fidelity

XML: Extensible Markup Language

1. Introdução

Com o constante avanço das tecnologias voltadas aos dispositivos móveis, muitos processos operacionais, nas mais diversas áreas, se tornam cada dia mais rápidos e eficientes. O propósito deste trabalho é apresentar uma solução eficaz que auxilie no processo de automação e gestão de serviços de um restaurante.

Antes de dar início ao projeto é feita uma análise dos processos em um restaurante que podem ser automatizados. O processo de pedidos é o que pode sofrer maiores melhorias. Pois uma vez que este processo é automatizado o tempo economizado com a soma de todos os pedidos é muito grande, podendo gerar assim um lucro maior para o restaurante. Outro ponto que deve ser considerado é que a maioria dos restaurantes utiliza um sistema que não é capaz de registrar todas as informações sobre seus processos. E essas informações podem ser muito valiosas e capazes de gerar uma excelente fonte de conhecimento sobre o restaurante.

Para chegar em uma solução é feito um estudo sobre as tecnologias mais recentes citando as principais características de cada uma. Existem diversas tecnologias que podem ser utilizadas para tal finalidade, no entanto o trabalho se concentra em algumas delas. Muitos fatores estão envolvidos na escolha das ferramentas, elas possuem características distintas e atendem soluções específicas.

O sistema permitirá que os pedidos em um restaurante possam ser feitos através de um PDA (utilizado por um garçom) e enviados para cozinha sem o auxílio de nenhum cabo (*Wireless*). Com os dados capturados pelo PDA, serão gerados relatórios e gráficos com o intuito de responder as mais variadas perguntas que auxiliem na área de planejamento estratégico dos negócios do restaurante. Com isso pretende-se entender mais sobre o comportamento dos clientes e conseqüentemente propor melhorias para o restaurante.

1.1. Objetivo geral

Automatizar o processo de gestão de negócios e serviços em um restaurante através do desenvolvimento de um aplicativo que utiliza a tecnologia *Wireless* em um *PDA e Data Warehous*.

1.2. Objetivos Específicos

- Apresentar características das tecnologias *J2ME* e *.NET Compact Framework*.
Identificando as vantagens e desvantagens de cada uma;
- Apresentar um comparativo entre os padrões de transmissão *Wireless*, citando as características de cada uma;
- Propor uma ferramenta que possa agilizar o processo de pedido em um restaurante;
- Responder questões relevantes à um restaurante através da tecnologia *Data Warehouse*.

1.3. Justificativa

A maioria dos restaurantes possui algum tipo de deficiência no atendimento aos seus clientes, muitas vezes quando o cliente faz um pedido, o pedido demora para chegar ou chega errado, isto acontece geralmente quando esse processo é manual e exige um certo deslocamento físico tanto da parte do garçom quanto do cozinheiro. A automatização deste processo pode reduzir bastante essas falhas, garantindo uma maior segurança e rapidez nos pedidos. Outra deficiência existente é que os dados relativos ao que está acontecendo com o restaurante dificilmente são registrados como por exemplo, que tipo de cliente fez determinado pedido, se é a primeira vez do cliente, ou como estava o tempo na hora do pedido, enfim, quanto mais informações forem registradas, mais eficiente será uma futura inferência. Este trabalho propõe uma solução que auxilie nesse sentido.

O projeto dará todas as condições a um garçom realizar pedidos e enviar à cozinha do restaurante sem precisar se locomover, e quando o pedido estiver pronto o cozinheiro por sua vez enviará uma notificação ao garçom, tudo isto sem o auxílio de nenhum cabo. Toda essa troca de informação, junto com algumas informações referente aos clientes serão armazenadas em uma base de dados. Com uma base de dados robusta e com o auxílio de um DW será possível fazer diversas inferências sobre o restaurante e o comportamento de seus clientes. Nesse sentido, acontecerão melhorias na qualidade do restaurante, pois o mesmo terá o processo de pedidos automatizado e será visto de uma nova forma facilitando as tomadas de decisões.

1.4. Estrutura do trabalho

Para alcançar os objetivos deste trabalho, o desenvolvimento do mesmo terá 3 etapas. Na primeira, capítulo 2, 3 e 4, será feito um estudo sobre PDAs e tecnologias que dão suporte para esses dispositivos. Serão comparados os padrões de comunicação sem fio disponíveis atualmente, e também as atuais tecnologias existentes para o desenvolvimento de aplicativos para dispositivos móveis.

Na segunda etapa será abordado o uso de uma *Data Warehouse*, citando as vantagens que essa ferramenta pode trazer à uma empresa com as valiosas informações estratégicas. Nessa etapa também constam alguns itens que fazem parte do conceito *Data Warehouse* como ferramenta de *front-end*, extração e *metadados*.

A terceira etapa é o desenvolvimento de uma aplicação com base em todos os conceitos abordados. A aplicação dará suporte ao processo de gestão de negócios e serviços em um restaurante. Uma parte do desenvolvimento será um software para um PDA que supra as necessidades de um restaurante e ao mesmo tempo capture informações relevantes para uma futura análise. Outra parte terá o foco na apresentação dos dados, visando responder as mais variadas perguntas para dar suporte às tomadas de decisões para isso será utilizado um *Data Warehouse* e um ferramenta *front-end*.

Do ponto de vista dos procedimentos técnicos, a pesquisa é de cunho: bibliográfico, elaborada a partir de material já publicado (sites e livros); documental, a partir de materiais que não receberam tratamento analítico (Manual de dispositivos móveis); e experimental, através da determinação do objeto de estudo, da seleção das variáveis, das formas de controle e da observação dos efeitos produzidos. (GIL, 1991).

2. Tecnologias que dão suporte aos PDAs

PDAs são pequenos computadores que podem ser transportados dentro do bolso, e que possuem características muito semelhante a de um PC, no entanto com algumas limitações. Existem diversos tipos de PDA, os mais atuais possuem tela colorida, acesso a Internet via *Wireless*, *Touch Screen* (mecanismo que possibilita o envio de comandos através de toques na tela), *Bluetooth*, câmera digital, MP3 player, controle remoto e até mesmo GPS. Eles também podem ser utilizados como calendário, relógio, agenda, jogos, calculadora e o uso de qualquer aplicativo dentro das suas limitações de memória e processamento.

As principais vantagens desses aparelhos é o fácil transporte e a conveniência de não precisar o auxílio de nenhum cabo, tanto para conectar-se à rede de dados quanto a um suporte de energia.

O número de *PDAs* vem crescendo progressivamente no mundo inteiro, empresas e usuários comuns estão cada vez mais adotando essa tecnologia, e tendências indicam que em pouco tempo os *Smartphones* (fusão entre um telefone celular e um PDA) serão maioria absoluta no mercado. (HAYKIN, 2003)

No mercado atual os principais tipos de PDAs existentes são o *Pocket PC* (Figura 1), desenvolvido pela *Microsoft* e o *PalmOne* (Figura 2), desenvolvido pela *PalmSource, Inc*

O *PalmOne* utiliza o sistema operacional *Palm OS*, a versão mais atual existente é a 6.0 *Cobalt*, sua base (*Kernel*) nas próximas versões deverá ser baseada no *SO Linux*. Este sistema é muito leve, rápido e seguro e tem uma imensa variedade de programas disponíveis na Internet tais como editor de texto, planilha eletrônica, agenda, calculadora, programas de acesso à Internet, envio de email, entre outros. (DAVE2005). A linguagem de programação mais utilizada para criação de softwares nesses dispositivos é a J2ME.

Já o *Pocket PC* utiliza o SO *Windows Mobile*, a versão mais atual desse sistema é a versão 5.0. O *Windows Mobile* é um Sistema Operacional compatível com o *Windows desktop*. Nele é possível “rodar” vários softwares da *Microsoft* parecidos com os que existem no *Windows* tradicional, tais como *Pocket Internet Explorer (HTML e WAP)*, *MSN Messenger*, *Pocket Word*, *Pocket Excel*, *Windows Media Player*, entre outros softwares. E também ele é capaz de sincronizar informações com um PC que utiliza *Windows* como SO.

Os *Pocket PCs* rodam quase que exclusivamente na arquitetura *intel*. Os processadores *Intel® PCA Application*, como o processador *Intel® PXA250 Application* estão nos dispositivos mais recentes e são baseados na tecnologia *Intel XScale®*. Os *Pocket PCs* estão disponíveis em mais de 30 empresas, dentro das principais estão: *Casio*, *Dell*, *HP*, *Compaq* e *Toshiba*.

Para criar softwares para o *Pocket PC* a tecnologia mais utilizada é a *.NET Compact Framework*, esta é bem parecida com *.NET Framework* que é utilizada para *Windows desktop*, porém um software desenvolvido para *Pocket PC* não funciona em um sistema operacional *Windows desktop*, e vice-versa. Apesar de ambos serem da *Microsoft* são sistemas totalmente diferentes.



Figura 1: Palmone (Tungsten T5)
Fonte: www.pnnewswire.com



Figura 2: Pockect PC (Pocket PC h4150)
Fonte: www.hp.com

2.1. Comunicação sem fio (Wireless)

Uma rede de comunicação sem fio pode utilizar sinais de rádio frequência ou infravermelho para transmissão de dados, na maioria das vezes trata-se de rádio frequência, essas ondas operam em uma frequência aberta e não necessitam de qualquer tipo de licença ou autorização.

Os sinais de rádio frequência que carregam informações de um ponto ao outro são chamados de onda portadora. E a forma de inserção de uma informação no sinal é chamado de modulação. A modulação permite que as informações sejam transportadas embutida nos parâmetros de frequência, amplitude ou fase da portadora.

São dois, os principais padrões para redes de comunicação sem fio, *Bluetooth* (redes *AdHoc*) e *IEEE 802.11* (redes infra-estruturadas). Eles possuem algumas características distintas que serão citadas em capítulos posteriores.

2.1.1. Técnicas de modulação

Na modulação, os bits são codificados através de símbolos. Ela mapeia cada seqüência de bits em um comprimento preestabelecido de um determinado símbolo. O conjunto de símbolos gerado pela modulação é conhecido como constelação, sendo que cada tipo de modulação gera uma constelação diferente. E é essa constelação que será transmitida pela onda portadora.

A maioria das redes sem fio usam a modulação SSM (*spread spectrum modulation*) para transmissão de dados, com essa tecnologia é possível obter a utilização máxima dos canais de rádio frequência. O tipo de SSM empregado é a maior diferença entre os padrões wireless.

2.1.1.1 Frequency Hopping Spread Spectrum (FHSS)

FHSS é uma técnica SSM que se baseia na rápida variação da frequência de transmissão de um sinal usando um padrão pseudoaleatório. Cada frequência é utilizada para uma transmissão somente durante um intervalo de tempo, esse intervalo é chamado de tempo de permanência ou *dwell time*. O *dwell time* é muito curto, menor que 400 ms, e pode ser ajustado. A variação rápida da frequência de transmissão é conhecida por saltos de frequência. Esses saltos são definidos por um gerador com números pseudoaleatórios que são instalados em cada uma das estações do sistema. Cada número pseudo aleatório corresponde a uma frequência.

No entanto, para que o sistema funcione as estações devem estar funcionando simultaneamente e com a mesma seqüência pseudoaleatória de frequências, para que ocorra assim um sincronismo e compatibilidade entre elas. Isto só é possível se os geradores de números pseudoaleatórios de todas as estações forem o mesmo, desta forma a seqüência gerada sempre será igual para todos.

2.1.1.2 Direct Sequence Spread Spectrum (DSSS)

DSSS tem algumas vantagens em relação ao método FHSS. O DSSS pode atingir uma operação mais eficaz e com velocidade superiores a 2 Mbps. Para a transmissão de 1 Mbps é usada a técnica BPSK (*Binary Phase Shift Keying*) e para a transmissão de 2 Mbps a técnica utilizada é a QPSK (*Quadrature Phase Shift Keying*), nessa técnica utiliza-se 4 rotações: 0, 90, 180 e 270 graus enquanto no BPSK são apenas 2, ou seja, o QPSK codifica 2 bits de informação no mesmo espaço que o BPSK codifica 1.

O FCC (*Federal Communication Commission*) determina 11 canais em 2,4 GHz para o uso do DSSS, mas com uma potência um pouco menor que o FHSS.

De acordo com esta técnica, como mostra a Figura 3, os “bits” de informação são multiplicados por uma seqüência de espalhamento, essa seqüência é chamada de código de *Barker*. Este código é uma seqüência de 11 bits (1 0 1 1 0 1 1 1 0 0 0). Cada bit do sinal original passa a ser representado no sinal transmitido. Com essa modulação tem-se um aumento da banda ocupada pelo sinal, o que significa o espalhamento deste sinal na frequência. Cada bit é codificado por 11 bits do código *Barker*, onde cada grupo de 11 chips codificados representa um bit do dado.

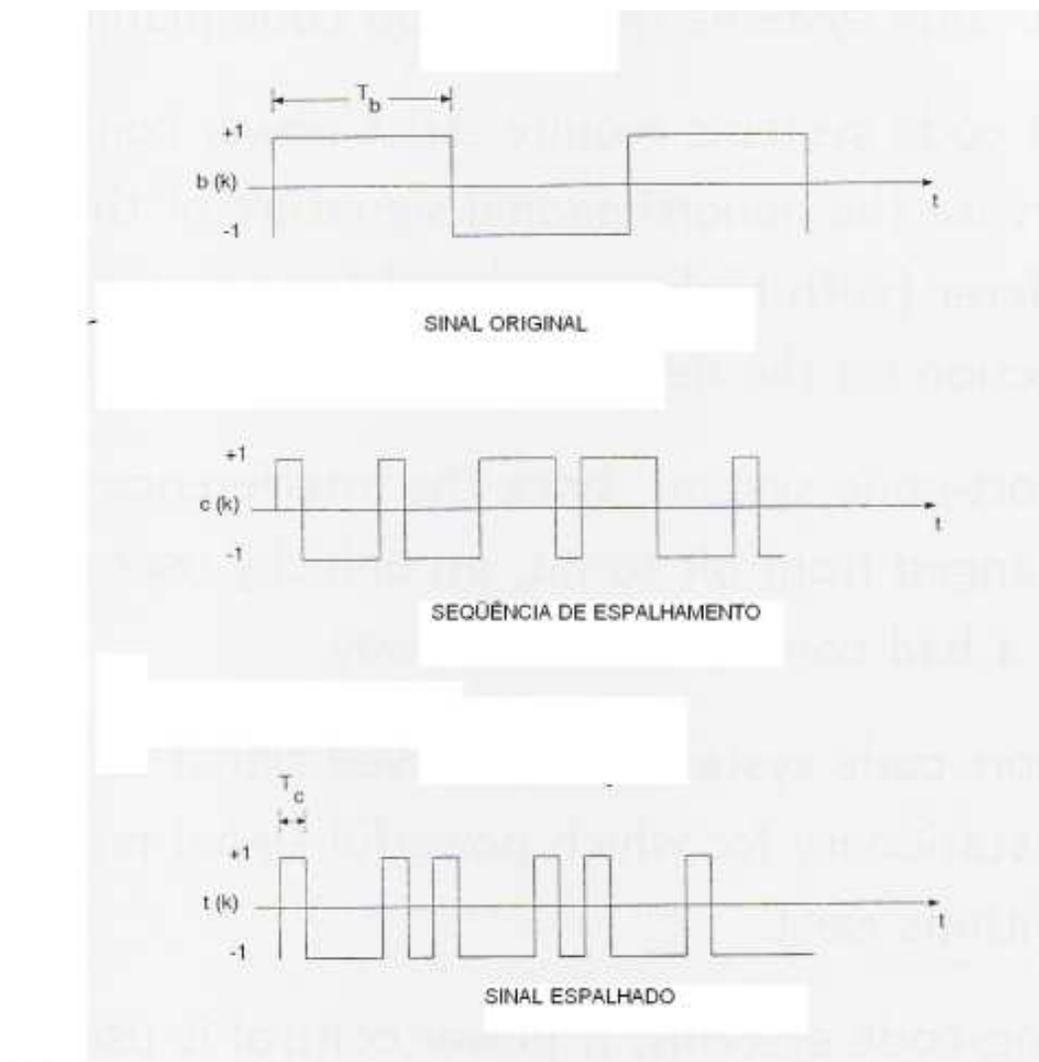


Figura 3: Espalhamento do sinal na DSSS
 Fonte: Principles of Wireless Communication

2.1.1.3. Orthogonal Frequency Division Multiplexing (OFDM)

A técnica de modulação DSSS é um pouco suscetível a obstáculos como paredes, móveis ou muros. Esse problema no DSSS causa a diminuição da taxa efetiva de transmissão. Para amenizar este problema criou-se a OFDM, esta técnica possui múltiplas portadoras ortogonais na qual as informações são moduladas. Para transmitir os dados, o canal de transmissão é dividido em vários sub-canais, cada um utilizando uma portadora independente. E antes de ocorrer estas modulações, os sinais passam por diversas etapas de processamento fazendo com que a performance alcançada pelo OFDM melhore ainda mais.

2.1.2. Padrões Wireless

O IEEE (*Institute of Electrical and Electronics Engineers*) é uma organização que foi fundada em 1963 nos EUA pela fusão do Instituto Americano de Engenheiros Elétricos com o Instituto de Engenheiros de Rádio. O IEEE é responsável por padronizar as camadas de enlace e físicas para redes sem fio, ele tem o objetivo de orientar empresas que produzem equipamentos que utilizam a tecnologia *Wireless*, buscando uma interoperabilidade entre eles.

Os principais padrões utilizados em soluções na computação móvel é o padrão *IEEE 802.11*, que possuem redes infraestruturadas e padrão *bluetooth*, com redes *Adhoc*.

2.1.2.1. Bluetooth

Esta tecnologia tem como objetivo a comunicação sem fio em um curto alcance entre aparelhos eletrônicos como telefones celulares, laptops, teclados, joysticks, impressoras, entre outros dispositivos móveis. Utiliza frequência de rádio para transmissão de dados e opera em uma faixa ISM entre 2.40 Ghz e 2.50 Ghz, esta faixa varia de país para país, em alguns locais é necessária uma adaptação, no entanto já existe iniciativas para criar um valor único. Existem 79 frequências diferentes de 1 MHz nas quais um dispositivo *bluetooth* pode estar transmitindo. A velocidade de transmissão pode chegar até 721 Kbps e a largura da banda é de 1 Mbit/s.

Para que ocorra a comunicação entre dois dispositivos através dessa tecnologia, os mesmos devem conter um chip bluetooth e operar na mesma frequência. Essa comunicação é feita através de um canal FH-CDMA (*Frequency Hopping - Code-Division Multiple Access*) que funciona da seguinte forma: o transmissor envia sinais na forma de frequências de rádio e sobre uma série randômica. O receptor captura estes sinais através de uma sincronia com o transmissor. A mensagem só é recebida se o receptor conhecer a série de frequências na qual o transmissor utiliza para enviar os sinais.

Três elementos estão ligados no estabelecimento de uma conexão entre dispositivos *bluetooth*: *scan*, *page* e *inquiry*.

O primeiro elemento, o *scan*, é utilizado para economizar energia, se algum dispositivo estiver ocioso, ele entra em *stand-by*, este é um modo onde o dispositivo “dorme” (praticamente zero o consumo de energia) e acorda periodicamente para verificar se há algum dispositivo tentando estabelecer uma conexão. Esse intervalo ocorre em uma faixa de 10 ms.

O *page* é usado no dispositivo que deseja estabelecer uma conexão, a cada 1,25 ms são transmitidos dois pedidos seguidos em diferentes portas e conseqüentemente é verificado duas vezes se há alguma resposta de outro dispositivo.

O *inquiry* é um mecanismo onde são enviadas mensagens com o objetivo de determinar quais dispositivos estão em sua área e qual é a configuração de cada um deles. Quando um dispositivo recebe esse tipo de mensagem, ele retorna um pacote chamado FHS (*Frequency Hopping-Synchronization*), esse pacote contém sua identidade e informações para que possa ocorrer um sincronismo entre os dispositivos.

A principal vantagem do *Bluetooth* em relação a outras tecnologias de comunicação sem fio é a transmissão de um sinal robusto e seguro; mesmo em ambientes com grandes ruídos é possível manter uma comunicação estável e sem interferências. Além disso todos os dados transmitidos são criptografados e possuem métodos de correção de erros. Sua principal desvantagem é que a transmissão só pode ser feita em um curto alcance, para manter uma comunicação estável a distância máxima é de 10 metros.

Sua primeira fabricante foi a *Ericsson*, na década de 90, hoje o chip *BlueTooth* é desenvolvido dentro de muitas companhias como *HP*, *3Com*, *Motorola*, *Dell*, *Sony*, *Samsung*, *Philips* entre outras. Estas fabricam os mais variados dispositivos com o chip bluetooth tais como *notebooks*, *PCs*, *palmtops*, *impressoras*, *handhelds*, *scanners*, celulares. Com o bluetooth é possível conectar um microfone à um celular, teclados à um computador e até mesmo interligar eletro-domésticos de uma casa.

A forma que dispositivos *Bluetooth* se comunicam chama-se *piconet*, na qual até 8 dispositivos podem ser conectados entre si. Um dispositivo é o “mestre” (*master*), sendo os demais “escravos” (*Slaves*). E ainda pode-se sobrepor vários *piconets*, multiplicando os pontos de comunicação. Esse método é chamado *Scatternet* como mostra a Figura 4. A figura seguinte mostra um exemplo desse método.

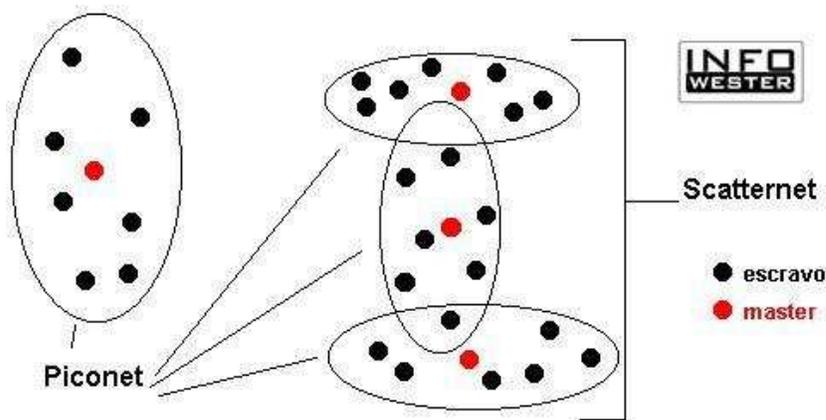


Figura 4: Piconet e Scatternet
 Fonte: Info Wester (2005)

Os dispositivos se comunicam através de canais, se ambos estiverem sintonizados no mesmo canal, a comunicação se estabelece. O canal é definido por uma seqüência de freqüências, o dispositivo mestre da piconet é quem cria esta seqüência, e os dispositivos escravos apenas seguem, isso é feito através do sincronismo. O dispositivo mestre pode mudar sua freqüência até 1600 vezes, isto faz com que diminua as interferências mesmo que exista diversos dispositivos dentro de um mesmo ambiente, isto porque cada dispositivo utiliza uma seqüência diferente. Para evitar colisões entre as múltiplas transmissões de dispositivos escravos, uma técnica chamada *polling* é utilizada pelo dispositivo mestre, que permite somente ao dispositivo indicado no slot *master-to-slave* transmitir no slot *slave-to-master*.

O *bluetooth* é classificado em três classes de acordo com o alcance e a potência e alcance:

- 100 mW, com alcance de até 100 m, cobre uma casa ou alguns andares de um prédio
- 2,5 mW e alcance até 10 m, cobre uma sala ou escritório
- 1 mW e alcance de 1 m, pouco utilizada

2.1.2.2. IEEE 802.11 (Wi-Fi)

Protocolo criado em 1997. É também conhecido como *Wi-Fi*. Em sua concepção original, sua velocidade era entre 1 e 2 Mbps, dois anos depois surgiu a versão 802.11b, esta pode atingir até 11 Mbps, e em 2003 foi criada a versão 802.11g, podendo atingir velocidades de até 54 Mbps.

Os três principais padrões na família IEEE 802.11x são:

- **IEEE 802.11 b**

Criado em 1999, este padrão atingi uma velocidade teórica de até 11 Mbps, no entanto pode chegar até 22 Mbps quando criada por fabricante não padronizados. Por funcionarem em uma frequência de 2.4 GHz, este padrão possui interferências, pois esta banda é equivalente a de dispositivos *Bluetooth*, telefones móveis, forno microondas, entre outros. Para amenizar esta interferência, este padrão utiliza a técnica de modulação DSSS. O baixo custo é a principal vantagem deste padrão.

- **IEEE 802.11 a**

Foi criado na mesma época do padrão *IEEE 802.11 b* e opera em 3 diferentes faixa de frequência, uma vai de 5,15 GHz até 5,25 GHz, outra faixa de 5,25 GHz até 5,35 GHz e outra 5,725 GHz até 5,825 GHz. Por possuírem uma alta frequência, seu alcance é baixo, no entanto não há muita interferência. Podem alcançar uma velocidade teórica de até 54 Mbps ou 108 Mbps por fabricantes que não seguem a padronização. Sua principal vantagem é a alta velocidade, se comparado com o padrão *IEEE 802.11 b* e também suportam um maior número de usuários simultaneamente, possuem até 8 canais não sobrepostos, contra 3 no padrão *IEEE 802.11 b*. Porém possuem como desvantagem a incompatibilidade com os padrões *IEEE 802.11 b* e *IEEE 802.11 g*, pois operam em diferentes faixas. Só em 2002, produtos com o padrão 802.11 a começaram a aparecer no comércio.

- **IEEE 802.11 g**

Foi criado em 2003 e pode atingir uma velocidade de até 54Mbps. Sua grande vantagem é a compatibilidade com o padrão 802.11b, ou seja, com pontos de acesso sem fios, oferece conectividade simultânea para ambos os equipamentos 802.11b e 802.11g, com isso é possível aproveitar instalações já feitas. IEEE 802.11g opera com a modulação DSSS quando a velocidade é de até 11 Mbps, acima disso é usado a modulação OFDM. Uma desvantagem é que ele não pode atingir um grande número de usuários. A modulação OFDM atinge uma alta velocidade, mas a largura de sua banda permanece a mesma, isto porque esse padrão só pode utilizar 3 canais em 2,4 GHz, enquanto no padrão 802.11a, 8 canais estão disponíveis em 5 GHz.

Tabela de comparação entre os Padrões IEEE 802.11:

	802.11b	802.11a	802.11g
Compatibilidade	802.11g	802.11a	802.11b
Velocidades Máxima	11Mbps	54 Mbps	54 Mbps
Frequência	2,4 GHz	5,8 GHz	2,4 GHz
Técnica de modulação	DSSS	OFDM	DSSS-OFDM
Custo	Baixo	Alto	Baixo
Distância	100 a 300 m	30 a 100 m	100 a 300 m
Uso	Mais usado	Aplicação Específica	Mais Promissor

Tabela 1: Comparação entre Padrões IEEE 802.11

Fonte: Communication System (Edição 4) Autor: Simon Haykin – Editora Wiley

2.2 Programação em PDAs

Construir uma aplicação por completo requer muito tempo e um conhecimento específico dos dispositivos alvos. A alternativa para isso é utilizar um *framework*. Com um *framework*, o programador dispensa a programação de baixo nível e se concentra mais na parte lógica do negócio.

Os principais *Frameworks* utilizados na programação em PDAs são *J2ME* e *.Net Compact Framework*. Estas tecnologias disponibilizam recursos para o desenvolvimento de softwares para dispositivos móveis com pouco recurso de memória e processamento.

2.2.1. J2ME (Java 2 Micro Edition)

J2ME ou *Java ME* é uma tecnologia que foi desenvolvida pela empresa *Sun*. Ela é a plataforma *Java* para dispositivos de pequeno porte. Seu ambiente oferece diferentes máquinas virtual, cada uma delas é otimizada para os mais variados tipos de processador e memórias. Ela é uma coleção de APIs definidas através da JCP (*Java Community Process*).

A JCP adotou duas abordagens para conceituar a programação para diferentes tipos de dispositivos. Primeiro foi definido o ambiente de execução, o *Java*, e um conjunto de classes chamada *core*, que operam sobre cada dispositivos, isso foi denominado Configuração. A segunda abordagem é o perfil que consiste em um conjunto de classes que são específicas para determinado tipo de aplicação.

Na Figura 5, como mostra a seguir, a arquitetura da plataforma *J2ME* consiste em 2 *stacks* (pilhas): Configuração (CDC e CLDC) e perfil. (Personal Profile, Foundation Profile e MID Profile).

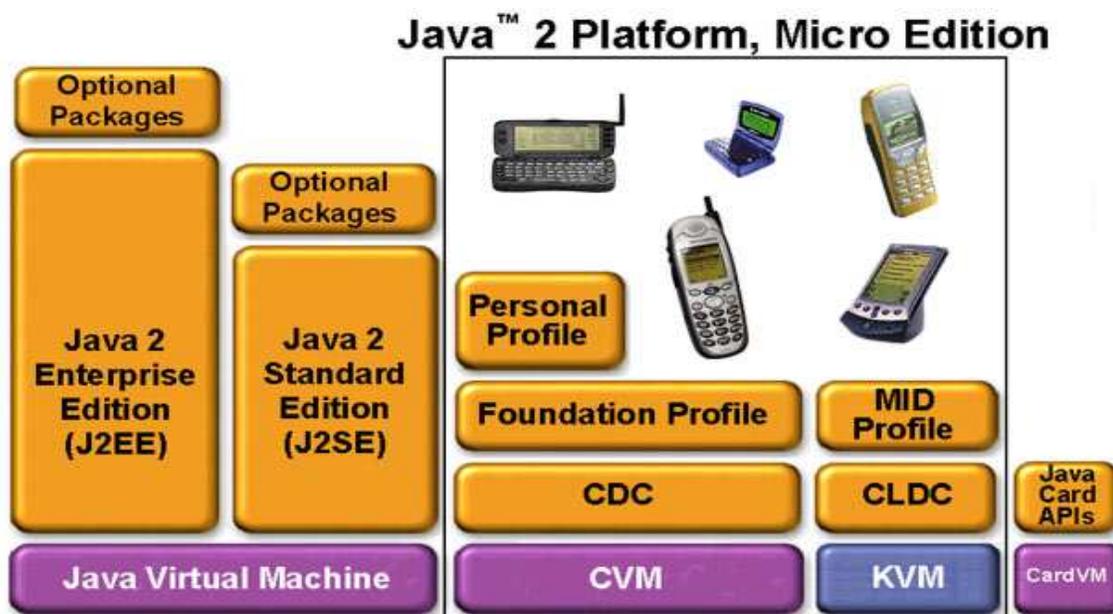


Figura 5: Arquitetura J2ME
 Fonte: java.sun.com

A Configuração inclui uma biblioteca de classes limitada, uma máquina virtual e uma interface para as funcionalidades nativas do dispositivo. Existem dois tipos de configurações, o CLDC (Connected Limited Device Configuration), este tipo tem um foco em dispositivos com fonte de energia limitada, *display* reduzido e com pouca capacidade de processamento e memória (de 128 a 256 *kilobytes*), é utilizado em *paggers*, telefones celulares, Palm Top e outros dispositivos que possuem um interface simples. O outro tipo é o CDC (*Connected Device Configuration*), este é voltado à utilização em dispositivos mais potentes e com uma capacidade maior de processamento e memória (de 2 a 4 megabytes), eles estão em conversores de TV por assinatura, computadores de bordo e os demais dispositivos que possuem uma interface mais complexa e conexões de rede de alta velocidade. As duas

configurações têm máquinas virtuais e livrarias diferentes. Cada uma foi criada com base nas características de seus aparelhos alvo.

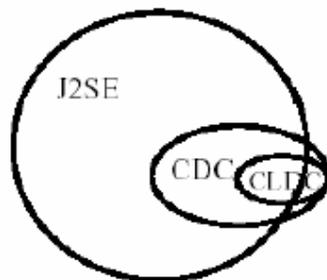


Figura 6: Relação da configurações no Java
Fonte: Autor

CLDC (Connected Limited Device Configuration)

O CLDC é voltado para dispositivos limitados em relação ao processamento e memória.

As principais características deste tipo de dispositivo são:

- Pelo menos 128 KB de memória não volátil está disponível para as bibliotecas CLDC e para a máquina virtual;
- Pelo menos 32 KB de memória volátil está a disposição para o *runtime* da máquina virtual;
- O processador é de 16 ou 32 bits;
- Consumo baixo de energia e através de bateria;
- Algum tipo de conectividade, na maioria das vezes sem fios e com a largura da banda limitada.

Existem dois perfis definidos para o CLDC, o perfil MIDP (Mobile Information Device Profile) e o perfil IMP (Information Module Profile). O IMP é mais recente que o

MIDP, um é basicamente o subconjunto do outro. A grande diferença é que o IMP não tem APIs relativas à interface com o utilizador.

A máquina virtual da CLDC é a KVM (*Kjava Virtual Machine*), esta é uma versão mais enxuta da máquina virtual JVM. A KVM é compacta e desenhada exclusivamente para dispositivos pequenos e com limitações. Quando foi criada esta máquina virtual o objetivo foi manter ela com os mesmos aspectos centrais da linguagem *Java*, mas que usando apenas alguns *KiloBytes*. [COURTNEY2002]

CDC (Connected Device Configuration)

Esta configuração esta voltada para dispositivos mais potentes e que estão ligados à uma rede, mesmo do que uma forma intermitente.

Suas principais características são:

- 32 bits de processador;
- 2 MB de memória disponíveis para o ambiente Java;
- Forte conectividade à rede;
- Alguma sofisticação na interface com o utilizador.

A Máquina virtual do CDC é a CVM (máquina virtual escrita em C) Segundo a *Sun* esta Máquina Virtual é determinística, portátil, e eficiente no espaço. O processo de execução do CVM é parecido com o processo de execução do JVM (*Java Virtual Machine*), ele suporta quase que integralmente as especificações e bibliotecas da JVM na versão 1.3 do J2SE e inclui também suporte para *multiithreading*, *threading*, *threads* de grupo e até tratamento de erros e exceções.

Na Figura 5, uma camada acima das configurações está o Perfil, ele tem o papel de garantir uma interoperabilidade entre famílias de dispositivos ou domínio definido pela plataforma Java para determinado tipo de mercado. No perfil estão inclusas bibliotecas de classes que são mais específicas para determinado tipo de domínio. Eles são um conjunto de classes que adicionam funcionalidades não abrangidas pelas configurações. Funcionam como um complemento para as configurações, ou seja, uma série de APIs padrões combinadas com alguma configuração provém um serviço muito mais completo para as aplicações.

Constantemente novos perfis são definidos. Abaixo constam alguns deles:

- Game Profile (CDC);
- Personal Profile (CDC);
- RMI Profile (CDC);
- Foundation Profile (CDC);
- PDA Profile (CDC);
- Personal Basis Profile (CDC);
- MIDP – Mobile Information Device Profile (CLDC). Este é o único perfil

usado na CLDC. Ele contém classes para um armazenamento local de dados. Existe a versão 1.0 e 2.0 para este perfil, as principais diferenças entre elas é que a versão 2.0 possui mais opções de conectividade (interface serial e HTTPS), classe para o desenvolvimento de jogos, uma maior variedade de classes para interface, entre outros recursos. (HIGHTOWER, 2006).

Uma aplicação MIDP é também conhecida como um MIDlet. Um suite MIDlet é uma ou várias MIDlets que são empacotados em conjunto, ele pode ser colocado no dispositivo e depois executado.

A Figura 7 representa um código MIDlet no qual sua função é aparecer um texto dizendo “Hello MIDlet”.

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class HelloMIDlet extends MIDlet implements CommandListener {
    public HelloMIDlet() {
        display = Display.getDisplay(this);
        exitCommand = new Command("Exit", Command.SCREEN, 2);
    }
    public void startApp() {
        TextBox t = new TextBox("Hello MIDlet", "Wireless Internet", 256, 0);
        t.addCommand(exitCommand);
        t.setCommandListener(this);
        display.setCurrent(t);
    }
    public void pauseApp() { }

    public void destroyApp(boolean unconditional) { }

    public void commandAction( Command c, Displayable s) {
        If (c == exitCommand) {
            destroyApp(false); notifyDestroyed();
        }
    }
}

```

Figura 7: Exemplo de um MIDlet
Fonte: Autor

Ferramentas de Desenvolvimento do J2ME:

- Java 2 Micro Edition Wireless Toolkit 1.0.4

Essa ferramenta está orientada para o desenvolvimento de aplicações J2ME MIDP (MIDlets), ela facilita a automatização o processo de colocação, depuração, compilação e execução dos MIDlets.

Esse *toolkit* pode ser unida à outras ferramentas de desenvolvimento como por exemplo o Forte para IDE Java, esta inclui muitas funcionalidade não existentes neste *toolkit*.

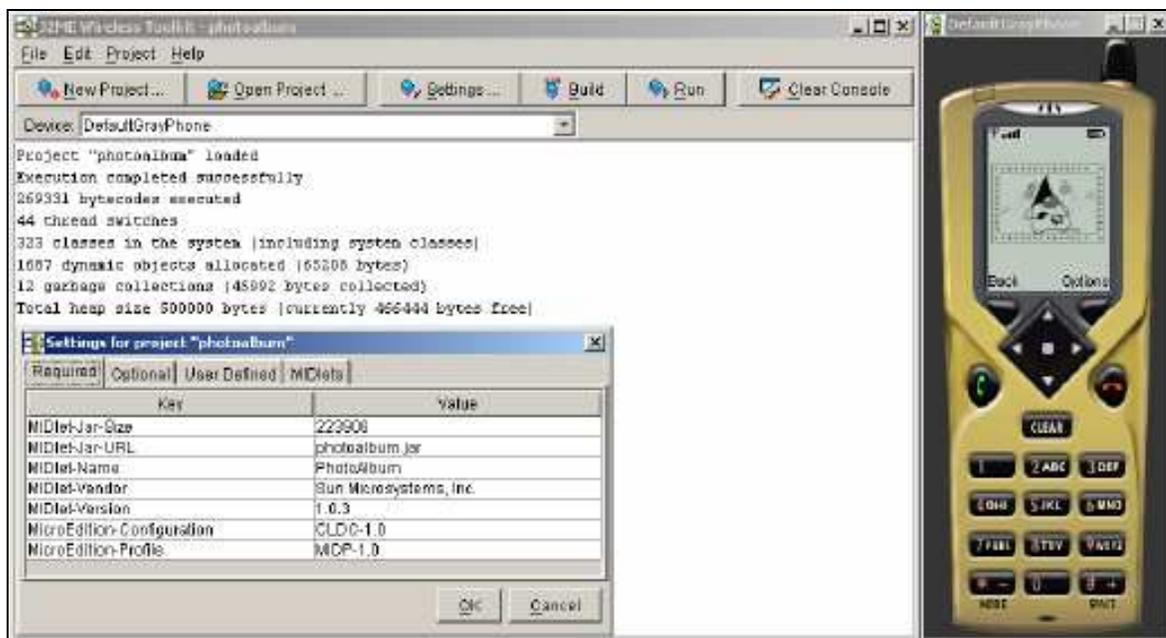


Figura 8: J2ME Wireless Toolkit 1.0.4

Fonte: java.sun.com

- Sun ONE Studio Mobile Edition 4

Esta é a ferramenta de desenvolvimento da *Sun* para construir aplicações J2ME. Ela permite o desenvolvimento de aplicações J2ME MIDP/CLDC. Nela inclui templates de classes, visualizador de classes, *wizards* para criar métodos e classes, um compilador, um editor de código inteligente, um depurador de error e vários emuladores para os mais variados aparelhos.

É uma ferramenta gratuita e que suporta várias plataformas tais como, *Linux*, *Solaris e Windows*.

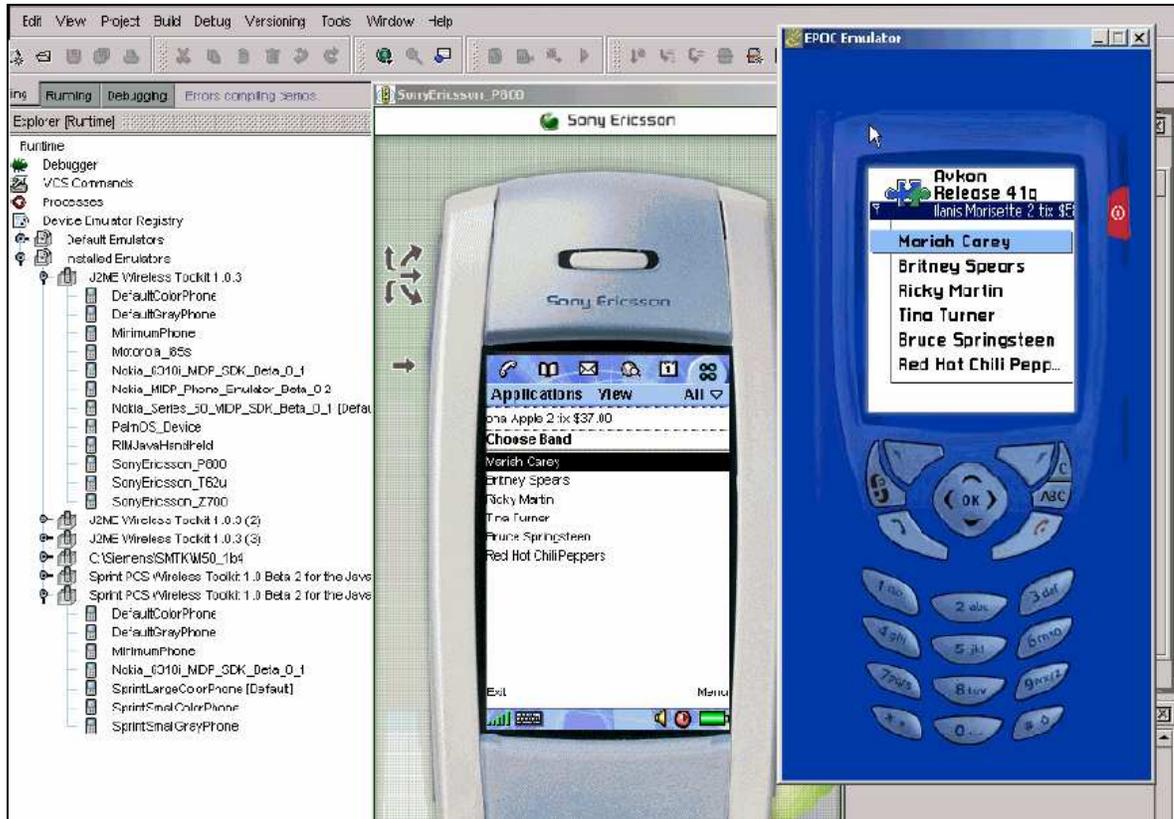


Figura 9: Sun ONE Studio Mobile Edition 4
 Fonte: developers.sun.com

2.2.2. NET Compact Framework

.Net Compact Framework é um subconjunto do *.Net Framework*, ele traz um poderoso ambiente para o desenvolvimento de aplicativos para pequenos dispositivos móveis. Todos programadores que conhecem ou desenvolvem aplicativos com o *framework .Net Framework* podem facilmente utilizar o *.Net CF*, pois são duas plataformas com muitas características em comum. Logo, uma empresa que já possui o conhecimento em *.Net Framework* pode passar a construir aplicativos para *Pocket PC* apenas com algumas adaptações. O investimento será bastante reduzido devido aos funcionários não precisarem de muito treinamento.

Dentro desta plataforma as principais vantagens nas funcionalidades são:

- Executa softwares que são independente de SO e *hardware*;
- Suporta diversos protocolos de rede e faz um perfeita conexão com serviços Web XML;
- Fornece ao desenvolvedor modelos que auxiliam e direcionam seus aplicativos;
- Possui uma otimização de recursos limitados do sistema;
- A geração de código nativo é obtida usando o método de compilação JIT (*Just-In-Time*). Esse método possui uma ótima performance se comparado com outros métodos de compilação nessa mesma área.

O *.NET CF* pode compartilhar códigos através de computadores ou de múltiplos aparelhos. Esse código pode fazer parte da camada de acesso a dados, pode ser uma lógica de negócio ou parte da camada de *XML Web Service*. Isso simplifica consideravelmente o desenvolvimento de aplicações rodam em múltiplos dispositivos.

A plataforma oferece um ambiente seguro e robusto para a execução do código no cliente. O modelo *Managed Code* (código gerenciado) aumenta a confiabilidade do código; a execução desse modelo assegura que o comportamento da aplicação não irá travar o aparelho e também que códigos maliciosos não tenham nenhuma permissão para acessar os recursos de segurança do sistema.

A integração do *.NET CF* com *XML Web Services* é bastante simples e eficiente, funciona como no *.NET Framework*, invocar um *XML Web Service* é tão simples com chamar uma função. *XML Web Services* são muito usado em *Smart Devices*, eles permitem que as aplicações se comuniquem uma com a outra sem levar em consideração o SO ou a linguagem de programação.

Os desenvolvedores *.NET CF* possuem um ampla portabilidade e reusabilidade para escrever para o modelo de programação comum, mas ele é também desenhado para ser estendido com *Class Library* que expõem características únicas para cada família de aparelhos, ou para um modelo de aparelho específico. Isto permite aproximar desenvolvedores para criar as melhores possibilidades para cada aparelho, enquanto usa novamente conhecimentos e códigos.

O *.NET CF* suporta o desenvolvimento em *C#, VB .NET, C++ e J#* além de que no futuro irá suportar outras linguagens devido ao *Common Language Interface (CLI)*.

Outra característica importante é que qualquer aplicativo desenvolvido nesta linguagem pode ser *debugado* com um emulador rodando na própria máquina de desenvolvimento ou num dispositivo físico anexado à ela através de uma porta serial ou USB. Isso facilita muito os testes do aplicativo e evita futuros problemas.

O *.Net CF* faz parte da seguinte arquitetura:

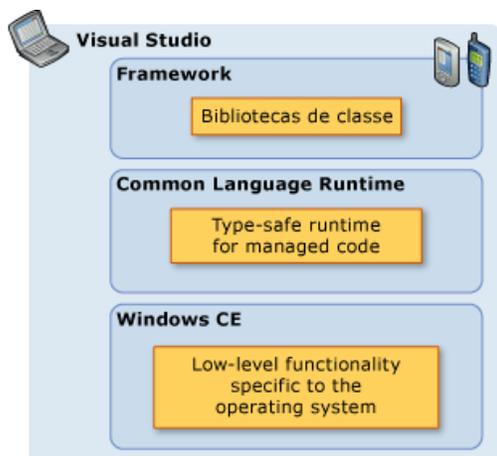


Figura 10: Arquitetura .net CF
 Fonte: www.microsoft.com

O Windows CE é a camada de nível mais baixo, este é o SO usado pelo .NET CF. Possui vários recursos específicos tais como elementos gráficos, serviços da Web, Windows Forms. Esses foram criados de uma maneira um pouco diferente do .NET Framework completo.

O Windows CE possui as seguintes características:

- Compatibilidade com a segurança nativa;
- Integração completa com programas de instalação nativos;
- Interoperabilidade com o código nativo usando COM.

O CLR (Common Language Runtime) está na segunda camada e também é um pouco diferente do CLR do .NET Framework completo. Ele foi reconstruído para que possa executar com mais eficiência certas funcionalidades, pois a memória utilizada por ele é limitada e a energia provem de um bateria.

Na terceira camada encontra-se o .Net CF. Ele fornece uma imensa quantidade de recursos que facilita o desenvolvimento de aplicativos para dispositivos móveis.

Ferramentas de Desenvolvimento do .NET Compact Framework

- Smart Device Extensions for Visual Studio .NET

O Visual Studio .NET é um ambiente de desenvolvimento para o .NET CF e também para o *.NET Framework*. As extensões para dispositivos móveis (*Smart Device Extensions*) são opcionais. Ela incluem classes do *.NET CF*, SQL CE e um emulador para Pocket PC. Essa ferramenta possui um editor de código, *wizards* para métodos e classes, um manuseador de classes, um ambiente para desenhar interfaces e um depurador.

Neste momento é a única ferramenta que suporta .NET CF. A versão mais atual do Visual Studio .NET é a versão 2005

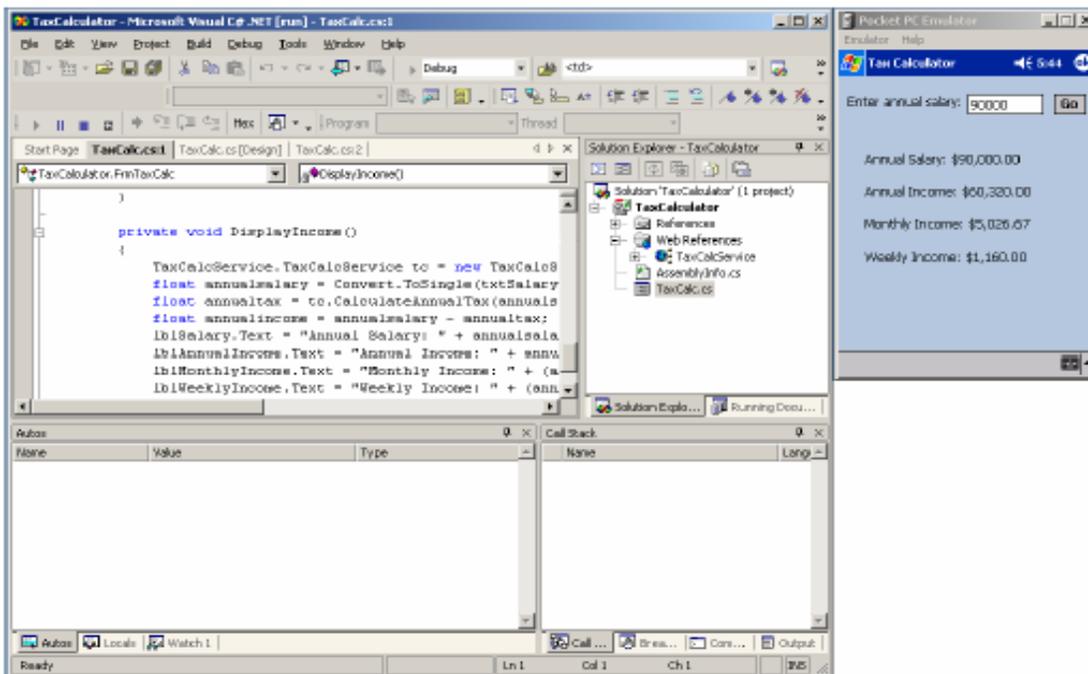


Figura 11: Smart Device Extensions for Visual Studio .NET
Fonte: www.microsoft.com

2.2.3. Comparativo J2ME X .NET Compact Framework

	.Net CF	J2ME CDC	J2ME CLDC
Cross-language	Sim	Não	Cross-language
Criptografia	Third-party APIs	JCE and third-party libraries	Bibliotecas externas
Interface	Sub-sistema do Windows Forms	Sub-sistema do AWT (Abstract Windowing Toolkit),	MIDP liquid crystal display UI, PDA Profile sub-sistema do AWT
Database API	Sub-sistema do ADO.Net, DataGrid	Sub-sistema do JDBC	Vendor-specific JDBC-like APIs
Mobile database	SQL Server CE, Sybase iAnywhere Solutions	IBM DB2 Everyplace, iAnywhere Solutions, PointBase, Oracle9i	Vendor-specific relational implementation over RMS, Oracle SODA
Base de Dados Remota	Any ADO.Net compatible	Qualquer JDBC compatível	Vendor-specific JDBC-like API bridge
Linguagem	VS.Net 2005	Command line, vendor SDKs, CodeWarrior.	Command line, vendor SDKs, all major Java IDEs
Ferramentas Web services	Integrated with VS.Net	kSOAP	kSOAP
Custo	Alto	Alto	Médio
Email and PIM	P/Invoke Outlook APIs	JavaPhone e APIs externa	PDA Profile
SMS	P/Invoke device SMS stack	Wireless Messaging API	Wireless Messaging API
Disponibilidade	Privado	Público	Público
Modelo de Segurança	Simplified .Net model	Full Java security manager	Limited Java 2 model supplemented by OTA
Compatibilidade Byte-code	Standard .Net CLR	Standard Java 2	Não compatível com J2SE ou CDC
Padronização EMCA	Sim	Não	Não
Instalação Cliente	ActiveSync, Internet Explorer download	Sync, download	Formal OTA specification
Multimedia	Windows Media Player APIs	Sub-sistema do JMF	Embutido MIDP plus J2ME multimedia APIs

Tabela 2: Comparação entre J2ME e .Dot CF.

Fonte: <http://www.javaworld.com/javaworld/jw-02-2003/jw-0221-wireless.html>

3. Desenvolvimento da Aplicação

Para o desenvolvimento dessa aplicação foram utilizados o *PDA Pocket PC* com o *Wi-Fi IEEE 802.11b*, a plataforma *.Net Compact Framework*, a linguagem de programação VB.NET e o banco de dados SQL Server 2000. Esta decisão foi baseada na viabilidade e facilidade de adquirir e aprender essas novas tecnologias, uma vez que o autor já possui um conhecimento do Framework *.Net Framework* e também pelas eficientes e robustas ferramentas que existem disponíveis para estas tecnologias. Na última parte do desenvolvimento do aplicativo será criado um *Data Mart* para ajudar na visualização dos dados relativos ao restaurante, o front-end utilizado para este *Data Mart* será o *PivotTable and PivotChart* do *Microsoft Excel*.

Uma parte da aplicação será localizada em um servidor, e outra parte em um *Pocket PC*. A comunicação entre o PDA e o servidor ocorrerá através de *Web Services*, este permite que diferentes aplicações localizadas em diferentes máquinas possam trocar informações em formato *XML*. No aplicativo em questão, o garçom registrará as informações no *Pocket PC*, e este por sua vez enviará as informações para o servidor através de funções *Web Services*.

O exemplo na Figura 12 mostra a chamada de um *Web Service* para a inclusão de um pedido. O cliente (*Pocket*) fará a chamada para o servidor com os dados que deseja incluir.

```
Dim r as new Restaurante
Dim retorno as Boolean
Retorno = r.IncluirPedido(32, 1, 3, 5, 5, 1, 15, 20, 11, 2006, 1, 1, 1, 0, 5, 32)
```

Figura 12: Chamada de um Web Service através do VB.NET
Fonte: Autor

Note que o uso de um *Web Service* é igual a chamada de uma função comum.

Na Figura 13 está um exemplo de *Web Service* escrito com a linguagem de programação *Visual Basic* que inclui o pedido.

```

Imports System
Imports System.Web.Services
Imports System.Data
Imports System.Data.Odbc
Imports System.Data.SqlClient
Imports System.IO

Public Class Restaurant :Inherits WebService
Public Function IncluirPedido (ByVal Restaurante As Int16, ByVal Mesa As Int16, ByVal
Garcon As Int16, ByVal Item As Int16, ByVal Feriado As Int16, ByVal Hora As Int16,
ByVal Dia As Int16, ByVal Mes As Int16, ByVal Ano As Int16, ByVal NU_Item As Int16,
ByVal Mulher As Int16, ByVal Homem As Int16, ByVal Crianca As Int16, ByVal
Dia_Semana As Int16, ByVal Semana_Ano As Int16) As Boolean
Dim Retorno as Boolean
Dim l_rs As SqlDataReader
    Connect ()
    l_rs = executeInstruction("Insert into Fato_Pedido values(" + Restaurante + "," +
Mesa + "," + Garcon + "," + Item + "," + Feriado + "," + Hora + "," + Dia + "," + Mes + "," +
Ano + "," + NU_Item + "," + Mulher + "," + Homem + "," + Crianca + "," + Dia_Semana +
"," + Semana_Ano + ")")
    Retorno = l_rs
    l_rs.Close()
    disconnect()
    Return Retorno
End Function

Public Shared Function executeInstruction(ByRef sql As String) As SqlDataReader
    On Error GoTo erro
    Dim catCMD As SqlCommand = CNsqlServer.CreateCommand
    catCMD.CommandText = sql
    Return catCMD.ExecuteReader
Exit Function
erro:
    Console.Write("Err: " & Err.Description, MsgBoxStyle.Critical)
End Function
Public Function connect() As SqlConnection
    If CNsqlServer.State <> ConnectionState.Open Then
        CNsqlServer = New SqlConnection("Data Source=192.168.200.3; Initial
Catalog=Orkut;")
        CNsqlServer.Open()
    End If
    Return CNsqlServer
End Function
Public Sub disconnect()
    CNsqlServer.Close()
End Sub
End Class

```

Figura 13: Exemplo de um Web Service
Fonte: Autor

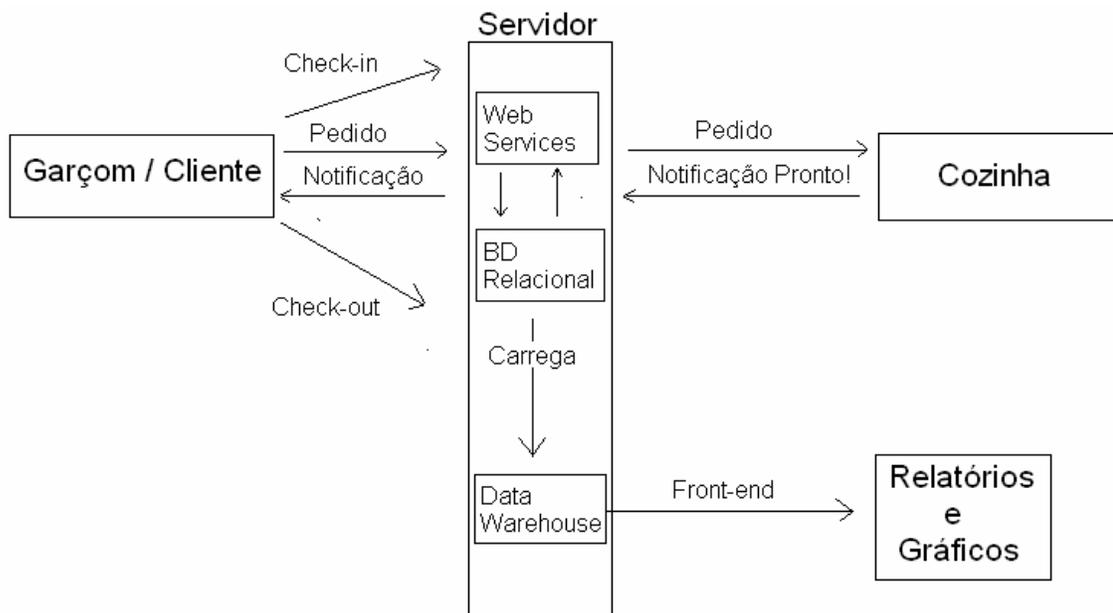


Figura 14: Estrutura do sistema
Fonte: Autor

Na figura 14 consta um esboço de como será o sistema. O garçom efetuará a operações de check-in, pedidos e check-out e também possíveis alterações. A cozinha irá apenas receber o pedido e notificar quando o pedido estiver pronto. Os dados relativos as operações são registrados em uma base de dados relacional com ajuda de *Web Services*. A partir deste BD operacional é feito uma carga diaria em outro BD que tem como estrutura um modelo dimensional (*Data Warehouse*). E o *Data Warehouse* junto com um *front-end* gera-se relatórios e gráficos.

3.1. Pocket PC (Interface e funções)

Cada garçom terá um *Pocket PC*, no qual irá registrar quando o cliente entra no restaurante, seus pedidos e o check-out.

A usabilidade é um fator crucial nesse projeto, as interfaces projetadas visam um entendimento por intuição pelos funcionários e com o objetivo dos mesmos executarem todas as funções em um menor tempo possível. Para realizar um pedido, um garçom precisa de, em média, 7 clicks, para fazer um *check-in*, 4 clicks, *check-out*, 3. Esses números são relativamente baixos, qualquer processo é tranqüilamente feito em menos de 10 segundos

Na primeira tela (Figura 15 e 16) o garçom visualizará um desenho do restaurante, identificando as mesas ocupadas e pedidos pendentes. A cor verde na mesa indica que a mesa está livre e cor vermelha indica que a mesa está ocupada. O contorno amarelo sobre o vermelho indica que a mesa está aguardando algum pedido.

Nestas telas é possível fazer o check-out clicando na mesa desejada e depois na figura do canto inferior direito da tela. Bastam 2 clicks para fazer o check-out.

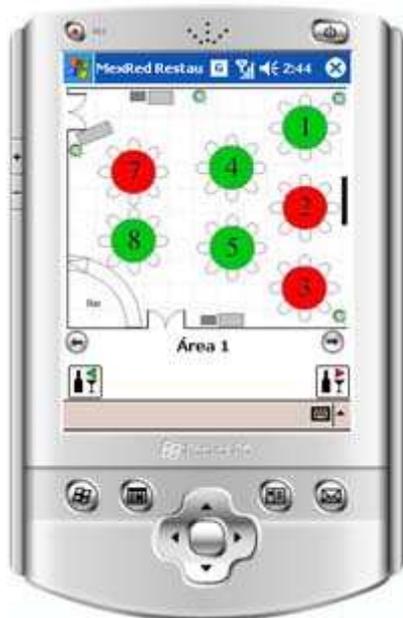


Figura 15: Área 1 - Restaurante
Fonte: Microsoft Device Emulator

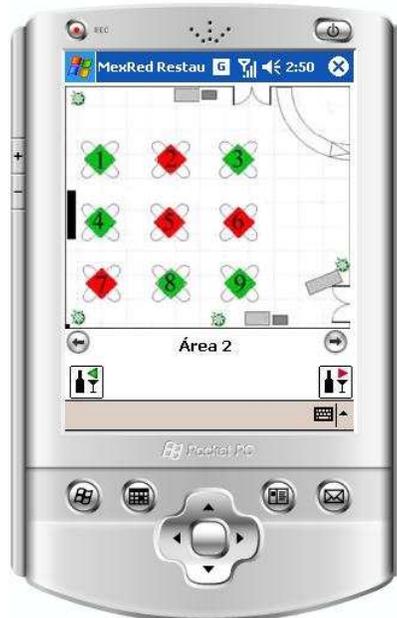


Figura 16: Área 2 - Restaurante
Fonte: Microsoft Device Emulator

O *check-in* é um pouco mais complexo. No momento em que o cliente entra no restaurante e senta na mesa, o garçom deverá, através do *Pocket PC*, registrar alguns dados, como mostra a Figura 17. As informações capturadas na hora do *check-in* são as seguintes:

- A hora que o cliente chegou no restaurante; esta informação é registrada implicitamente, no momento no *check-in* a hora é gravada.
- Quantidade de pessoas que estão na mesa; o garçom informará ao sistema quantos homens, mulheres e crianças estão ocupando a mesa.
- Se é ou não a primeira vez do cliente no restaurante.
- O tempo/clima na hora em que foi feito o *check-in*. Esta informação será acrescentada futuramente com base na hora em que foi feito o *check-in*.
- E qualquer outra observação relevante relativa ao clientes como idade, alguém de aniversário, etc. Essas informações podem ser colocadas no campo OBS.

Alguns campos em certas circunstâncias poderão ser deixados em branco, no entanto quanto mais informações forem adquiridas melhor será a qualidade dos relatórios futuramente.

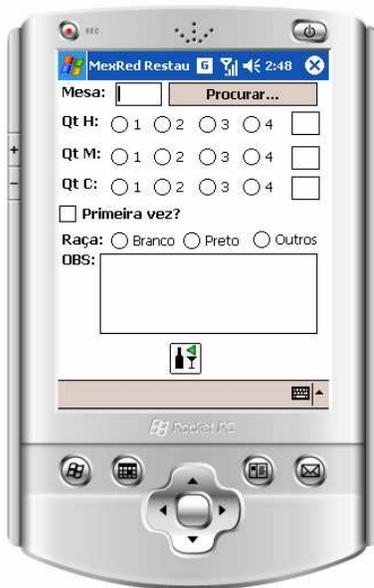


Figura 17: Tela de Check-in
Fonte: Microsoft Device Emulator

A Figura 18 é a tela que o cozinheiro ou outro funcionário qualquer, notificará o garçom que o pedido está pronto. A tela para realizar essa função é simples. Haverá um lista com os pedidos pendentes, quando o pedido estiver pronto, o “cozinheiro” clicará no pedido e consequentemente no botão “pronto”.



Figura 18: Tela Cozinheiro
Fonte: Autor

As Figuras 19 e 20 mostram respectivamente a tela para visualizar os pedidos de um determinado cliente e a tela para fazer um pedido.



Figura 19: Tela Visualizar Pedidos
Fonte: Autor

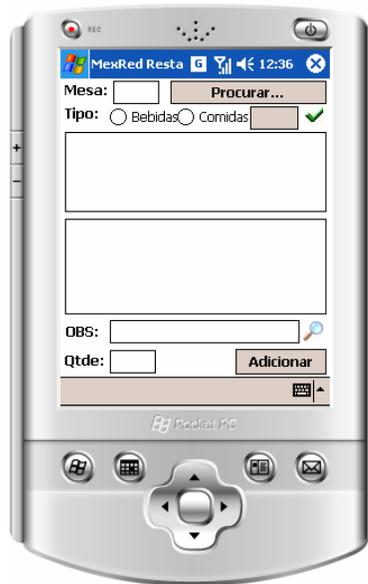


Figura 20: Tela Realizar Pedidos
Fonte: Autor

3.2. Data Warehouse

Segundo Willian H. Inmon “*Data Warehouse* é um banco de dados orientado por assunto, integrado, não volátil e histórico, criado para suportar o processo de tomada de decisão” (Inmmon, 1998). Kimball define um *Data warehouse* da seguinte forma “fonte de dados para consulta da organização” (Kimball, 1998)

DW poder ser definido com um Banco de Dados especializado, no qual ele integra o fluxo de informações a partir de um BD corporativo e fontes de dados externas. Um *DW* é construído para que os dados possam ser armazenados e acessados de tal forma que não tenham limites por tabelas e linhas relacionais. Sua função é tornar as informações corporativas visíveis e acessíveis para o seu entendimento, uso e gerenciamento. Ele oferece fundamentos e recursos necessários para um Sistema de apoio a Decisão, fornecendo dados históricos e integrados que servem desde a alta direção até a gerência de baixo nível. (Wilson, 2005)

Segundo Taurion (1998), uma maneira simples de saber se um Data warehouse será útil a uma empresa é responder um pequeno número de questões. Quanto mais respostas "sim", maior será o aproveitamento do Data Warehouse pela empresa. Algumas das questões são as seguintes:

- A empresa se baseia em informações para tomada de decisões?

Se o comportamento de seus clientes ou do mercado é necessário para uma decisão, a resposta deve ser "sim". A resposta "não" se refere apenas quando a decisão é baseada fortemente na intuição;

- O ramo da empresa é caracterizado por fortes concorrências e mudanças?

A resposta “sim” se trata de uma empresa que possui concorrentes e/ou seu ramo sofre constantes mudanças. Uma resposta "não" corresponderia a um segmento estável e de pouca competição;

- Os clientes são diversificados e numerosos?

Um exemplo de resposta "sim" pode ser um supermercado, onde o número de clientes é grande e são bastantes diversificados.. O "não" serviria para um empresa que fornece produtos para um único cliente como plataformas de petróleo;

- Os dados estão armazenados em diferentes locais?

O "sim" é proveniente de empresas que possuem diversas fontes de dados como banco de dados, papéis ou até mesmo dados relativos à fornecedores que não estão armazenados em nenhum lugar. O "não" será o caso de poucas empresas que possuem somente uma fonte de dados;

- Os mesmos dados estão em mais de um lugar em com formato diferentes?

Um exemplo claro disso é o "endereço do cliente" aparecer em mais de um lugar e com conteúdo diferente;

Segundo Amaral (1999), as modalidades de DW são as seguintes:

- **DWC (Data Warehouse Corporativo)**

Esta é uma modalidade bastante robusta, geralmente adotadas por grandes empresas. Características desta modalidade:

- Tempo de resposta rápido;
- Alta capacidade de armazenamento. (entre 100 GB e 1 TB);
- Dados não-voláteis, relacionais e bastante desnormalizados;
- Informações organizadas por área de análise.

- **ODS (Operational Data Store)**

Modalidade rápida. Opera diretamente com os dados operacionais, com característica de diminuir o tempo de resposta. Suas principais características são:

- O tempo de resposta varia de segundo a alguns minutos;
- Dados voláteis, relacionais e desnormalizados;
- Informações organizadas por área de análise (históricos de 30 a 60 dias)
- Seus usuários finais são consumidores de informação.

- **DM (Data Mart)**

Esta modalidade é utilizada em setores específicos da empresa, segundo Kimball que é um defensor do Data Mart, afirma que “Data Mart são subconjuntos de um DW completo”. Ou seja, o Data Mart tem uma visão apenas para um departamento ou área de interesse. Resolve apenas um problema empresarial particular. No caso do sistema desenvolvido nesse trabalho trata-se de um Data Mart, pois ele tem um único foco que é o comportamento dos clientes em um restaurante. O tamanho não importa para distinguir um *Data Mart* de um DW, o que importa é o “foco”. No DM as questões de projeto, planejamento e implementação são as mesmas encontradas nas fases de construção de um DW.

As seguintes características são atribuídas a um *Data Mart*:

- Fácil gerenciamento por serem menores;
- Foco para tomada de decisão em um nível departamental;
- Dados não voláteis, relacionais ou multidimensionais;
- Baixo custo quando comparado com um DW.

Arquitetura do Data Warehouse

A arquitetura do *DW* representa a estrutura completa de dados, processamento, resultados e comunicação. Suas principais camadas são: (RICHARD, 1997)

- **Camada de Acesso aos Dados:** Esta camada permite que a camada de acesso a informações se comunique com a camada operacional. A linguagem mais utilizada para esse fim é o SQL.
- **Camada de Diretório de Dados (Meta Dados):** Meta Dados são dados sobre um dado de uma empresa. Para que um DW seja completamente funcional, é preciso que a corporação tenha uma vasta quantidade de Meta Dados disponíveis.
- **Camada de Acesso a Informações:** Representa as ferramentas *front end* que o usuário final utiliza para acessar as informações. Exemplo dessas ferramentas são: *Lotus, Access, Excel* ou pode ser até mesmo um software ou uma página Web criados para exibição desses dados. As formas mais utilizadas para a visualização dessas informações são relatório e gráficos.
- **Camada de Gerenciamento de Processos:** Esta camada é responsável por manter o DW. Nela constam o agendamento de várias tarefas ou processos que carregam o DW.
- **Camada do Data Warehouse:** É o núcleo, onde os dados analíticos residem. Há uma redundância de dados, de forma a aumentar a flexibilidade e facilitar o acesso.
- **Camada de Organização dos Dados:** A organização dos dados inclui todos os processos necessários para selecionar, editar, combinar, resumir e carregar o Data warehouse

3.2.1. Modelagem Multidimensional

Modelagem Multidimensional é uma modelagem de dados onde os dados são estruturados em diversas dimensões e também desnormalizados. O tipo de modelagem dimensional mais utilizado é o *Star Schema*, em português, Esquema Estrela, este esquema foi desenvolvido por Ralph Kimball, um dos precursores da *Data Warehouse*.

A principal característica do *Star Schema* é a alta redundância de dados, que tem a finalidade de melhorar o desempenho na busca de informações. Este modelo é composto basicamente por dois tipos de tabelas: as de Fato e as de Dimensão. As tabelas de fato são grandes tabelas centrais e armazenam os fatos ocorridos e as chaves para características correspondentes nas tabelas dimensionais, que armazenam as descrições do negócio. Cada tabela de Dimensão possui uma única chave primária, e o conjunto dessas chaves primárias formará a chave composta da tabela de Fato. Cada negócio ou assunto possui sua tabela de Fato e suas respectivas tabelas de Dimensão.

O modelo dimensional deste trabalho terá uma tabela fato que se chamará “Fato Pedido” e também 6 tabelas dimensionais. A Figura 21 mostra o *Star Schema* feito para este trabalho, em anexo estão detalhes desta modelagem.

Star Schema

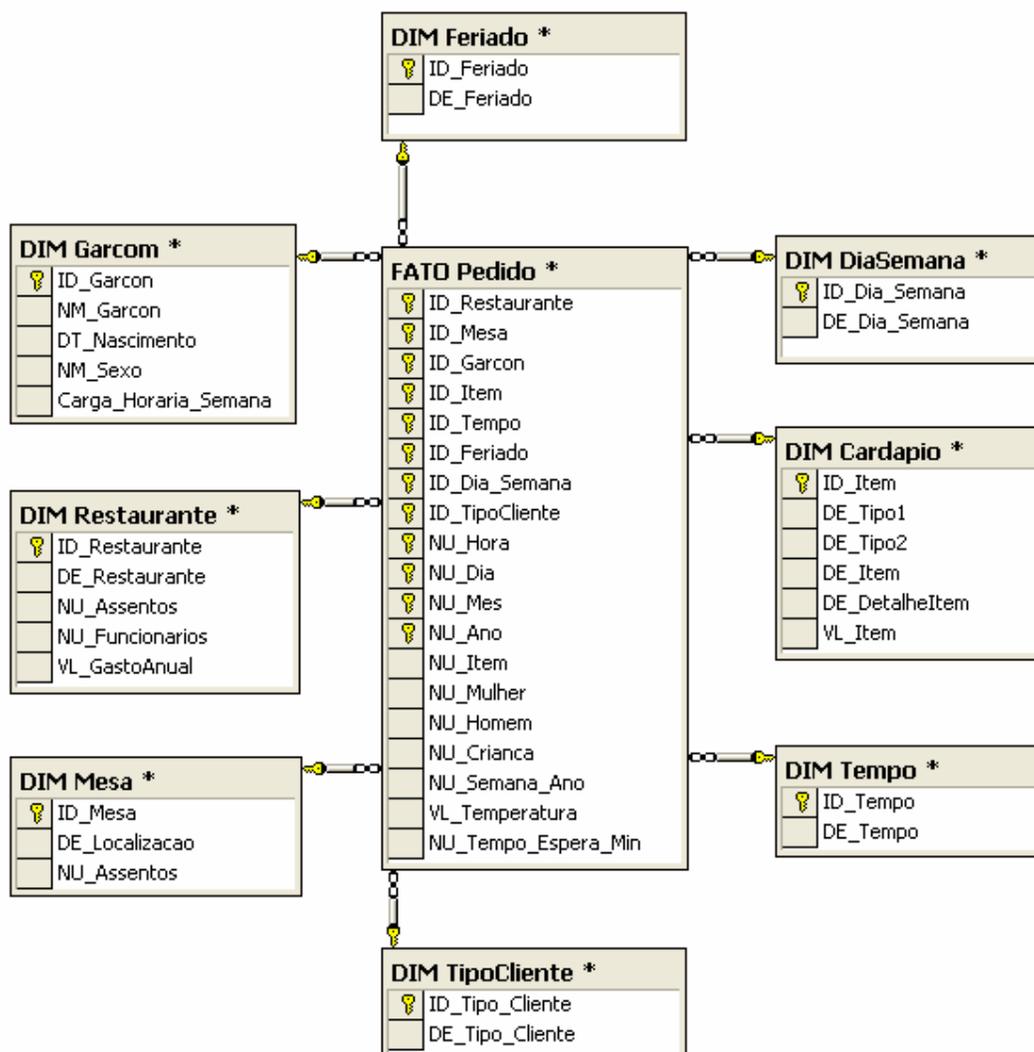


Figura 21: Star Schema

Fonte: Autor

3.2.2. Carga de dados

A carga de dados no *Data Mart* é feita uma vez por dia assim que o expediente se encerra. Os dados que alimentam o *DM* são provenientes de um BD operacional que é carregado pelo Pocket PC e por um sistema simples. E alguns dados são fixos e inseridos apenas uma vez manualmente, estes dados são relativos as dimensões “DIM DiaSemana”, “DIM Feriado”, “DIM Tempo” e “DIM TipoCliente”. Toda a carga de dados do *Data Mart* é feita com comandos SQL tradicionais.

Os dados das dimensões “DIM Cardapio”, “DIM Garcon”, “DIM Mesa” e “DIM Restaurante” são registrados ou modificados diretamente através de um sistema que o restaurante já possui para que no final do dia possam ser transferidos para o *Data Mart*.

E a maioria dos campos da tabela de fato (FATO Pedido) são primeiramente atualizados no BD operacional a partir das informações registradas pelo garçom no *Pocket PC*. Isto será feito através de um *Web Service*. Este *Web Service* conectará o servidor no qual o BD do sistema antigo se encontra através do *Pocket PC*, o código desse processo encontram-se na Figura 12 e 13.

Os campos “VL_temperatura” (campo que contém a temperatura na hora do pedido) e “ID_Tempo” (campo contém o estado do tempo na hora do pedido) da tabela “FATO Pedido” serão inserido também uma vez por dia a partir de um *Web Service* localizado no servidor de um Website especializado em tempo (www.weather.com). Este *Web Service* fornece as condições do tempo em qualquer momento do dia, e essa informação é linkada com o pedido. O código da chamada deste *Web Service* está em anexo.

Em anexo também estão os dados de todas as tabelas de dimensão

3.2.3. Front end (apresentação dos dados)

A maneira pela qual a alta gerência irá analisar os dados relativos ao restaurante é de extrema importância. O objetivo é conseguir eliminar todos os dados que não interessem a determinada questão e somente apresentar o que interesse e de uma forma clara.

Nesta aplicação *Front end* utilizado é *PivotTable and PivotChart* do *Microsoft Excel*. É claro que existem ferramentas de *front end* muito melhores e com mais funcionalidades, mas para o problema em questão o *Excel* atende todas as necessidades. Com essa ferramenta pode-se gerar relatórios e gráficos a partir da escolha de certas variáveis e dimensões. E o relatório pode ser manipulado dinamicamente, ou seja, é possível trocar dimensões e variáveis a qualquer momento, e isso com um simples arraste do mouse (*drag and drop*)

Com o *Data Mart* carregado e uma ferramenta *front end* é possível responder as mais variadas perguntas a respeito do restaurante, o projeto apenas se foca em algumas. São elas:

- Quais os horários e dias de maior demanda?
- Quais as comidas mais pedidas por dia e por horário?
- Qual a interferência do tempo no consumo de bebida e comida?
- Quais e quantas bebidas foram vendidas em dia de sol (detalhe no suco)?
- Como foi a movimentação nos feriados dos últimos 3 anos?
- Classificar garçons pela eficiência.
- Quais os tipo de clientes do restaurante e o que eles consomem?

A seguir estão as resposta para cada uma delas em forma de relatórios e gráficos.

- **Quais os horários e dias de maior demanda?**

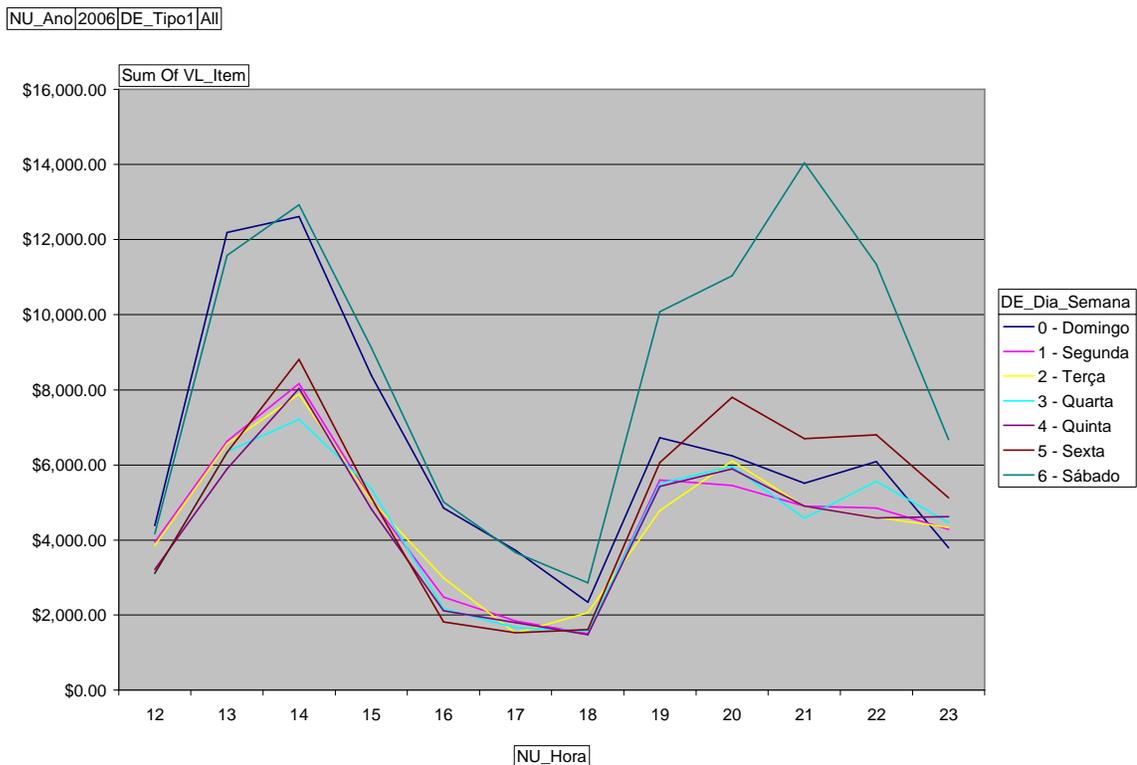


Figura 22: Total vendido por hora e dia da semana.
 Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor.

O gráfico (Figura 22) mostra que os horários de maior movimentação no restaurante é às 14:00 no sábado e no domingo, e às 21:00 no sábado. No eixo y consta o total vendido e no eixo x o horário. E cada linha define um dia da semana. O gráfico é relativo ao total vendido no ano de 2006. Com esta informação o restaurante pode tomar algumas medidas como: mudar os horários dos funcionários de tal forma que nos horários de pico tenha um maior volume de empregados para poderem assim melhor atender os clientes.

Com apenas alguns clicks é possível gerar um relatório parecido com o anterior e responder a seguinte pergunta:

- **Quais as comidas mais pedidas por dia e por horário?**

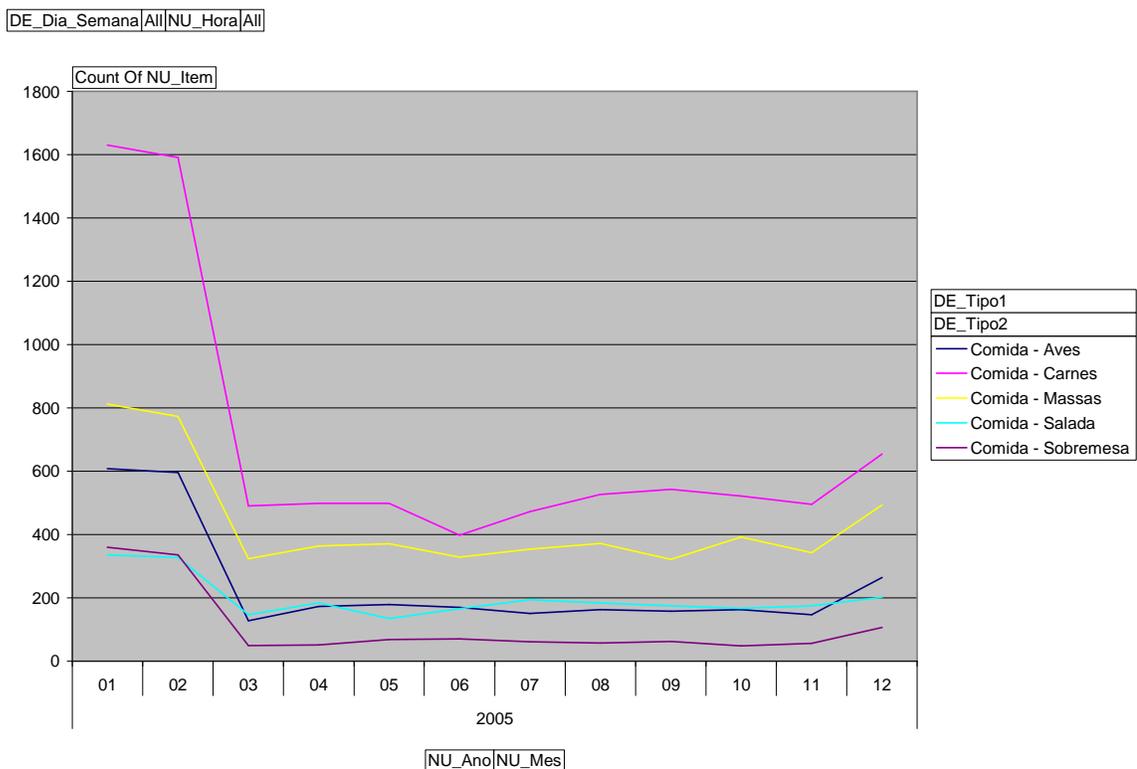


Figura 23: Demanda das comidas em relação aos meses no ano de 2005.

Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor.

O relatório, na Figura 23, mostra que no mês de Janeiro de 2005 o consumo de carne foi um pouco mais que o dobro do que o consumo de massa. Com essa informação o restaurante pode se preparar melhor no mês de janeiro para o fornecimento de carne aos seus clientes.

Esse tipo de relatório gerado pelo *Microsoft Excel* é totalmente flexível. A partir deste relatório é possível obter qualquer tipo de detalhe referente às variáveis que estão disponíveis como por exemplo saber quais os pratos com carne que são mais vendidos ou o dia da semana que mais se come carne ou massa. Pode-se até mesmo acrescentar novas dimensões que estão contidas no cubo e a partir dela fazer novos filtros.

- Qual a interferência do tempo no consumo de bebida e comida?

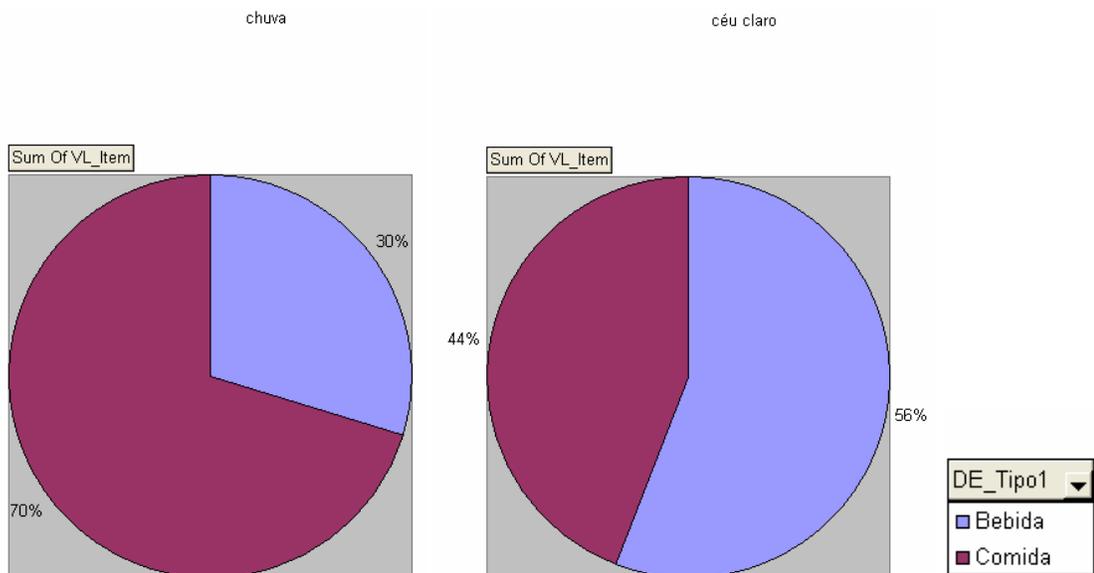


Figura 24: Influência do tempo no consumo de bebida e comida

Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor.

O gráfico na Figura 24 mostra que o tempo está fortemente relacionado com o consumo de bebida em relação à comida. Em dia chuvoso o consumo de bebida é de 30% e em dia ensolarado é de 45%.

- Quais e quantas bebidas foram vendidas em dia de sol (detalhe no suco)?

DE_Tipo1	DE_Tipo2	DE_Item	QT. Vendida
Bebida	Champagnes		205
	Drinks		414
	Refrigerante		2539
	Suco	Abacaxi	306
		Goiaba	147
		Laranja	303
	Suco Total		756
	Vinhos Importados		83
	Vinhos Nacionais		397
	Whisky Importado		202
	Whisky Nacional		122
	Bebida Total		4718

Tabela 3 – Quantidade de bebidas vendida em dia de sol

Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor.

Nesta tabela é possível visualizar detalhes da venda de bebidas em dia de sol, tendo um foco nos sucos. Essa informação pode ser muito útil para o restaurante, uma vez que o restaurante pode prever o que vai vender em determinado dia e se preparar para isso.

- **Classificar garçons pela eficiência.**

Garçom	N. Pedidos	Tot. Gasto	Méd. Min. espera	Gorjeta (%)
Antonio	543	11560.45	10.94	23.91
Marcia	743	14390.36	11.50	25.65
Maria	658	13982.24	11.99	22.44
Lucas	415	16565.22	12.43	26.51
Jomar	534	12618.81	12.55	22.31
Carlos	563	13164.23	12.71	23.66
Roberto	467	15482.07	12.89	22.13
Joao	531	15134.81	13.05	21.62
Joaquim	597	16483.70	13.45	23.38
Fernando	608	14824.33	13.67	20.03
Cláudia	566	15751.52	13.81	21.96
Lucas	503	15210.18	13.99	23.47
Media	561	14597.33	12.75	23.09

Tabela 4: Classificação dos garçons pela eficiência

Fonte: Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor.

Com esta tabela é possível fazer diversas inferências sobre a eficiência dos garçons, como por exemplo, qual garçom atende mais rápido, quem recebe mais gorjeta, ou quem faz os clientes gastarem mais, enfim, pode-se conhecer muito mais a respeito dos garçons.

- Como foi a movimentação nos feriados dos últimos 3 anos?

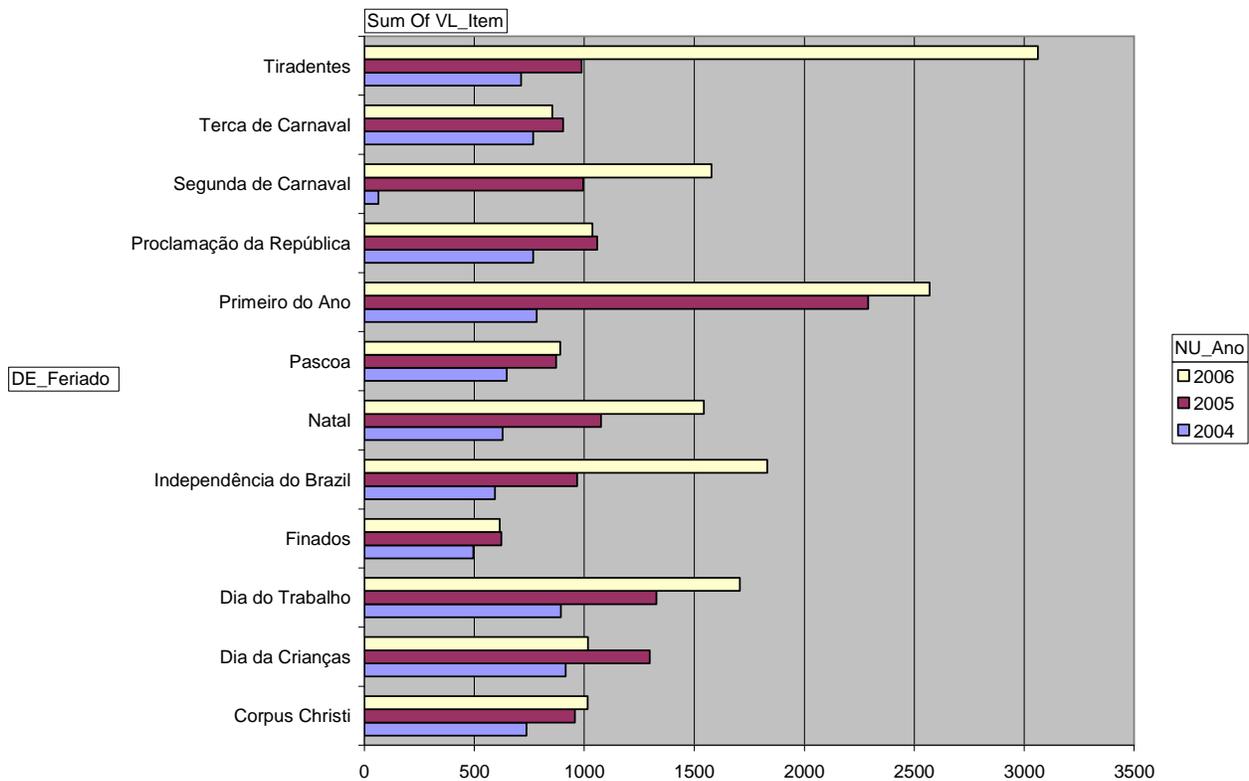


Figura 25: Movimento nos feriados dos últimos 3 anos

Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor com base nos dados do restaurante *Maxim*

Gráfico na Figura 25 mostra o total vendido em reais nos feriados dos últimos 3 anos. Com esta informação é possível fazer um previsão do faturamento ou preparar o restaurante para receber um determinado número de cliente.

• **Quais os tipo de clientes do restaurante e o que eles consomem?**

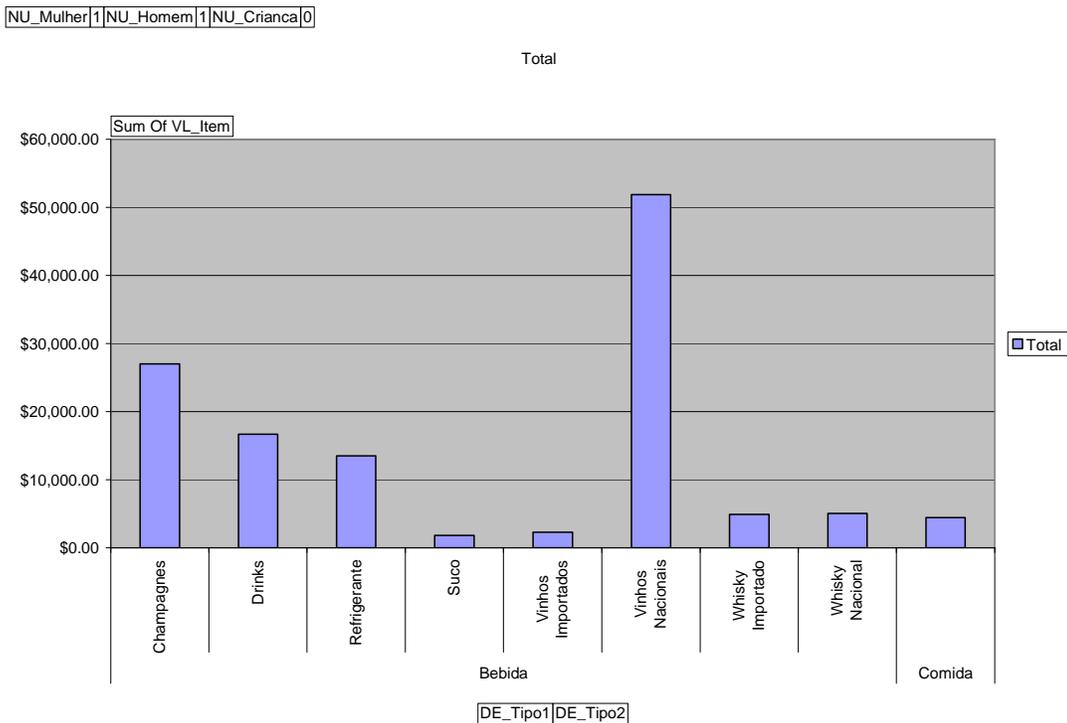


Figura 26: Consumo de bebida quando somente 1 casal ocupa a mesa
 Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor com base nos dados do restaurante *Maxim*

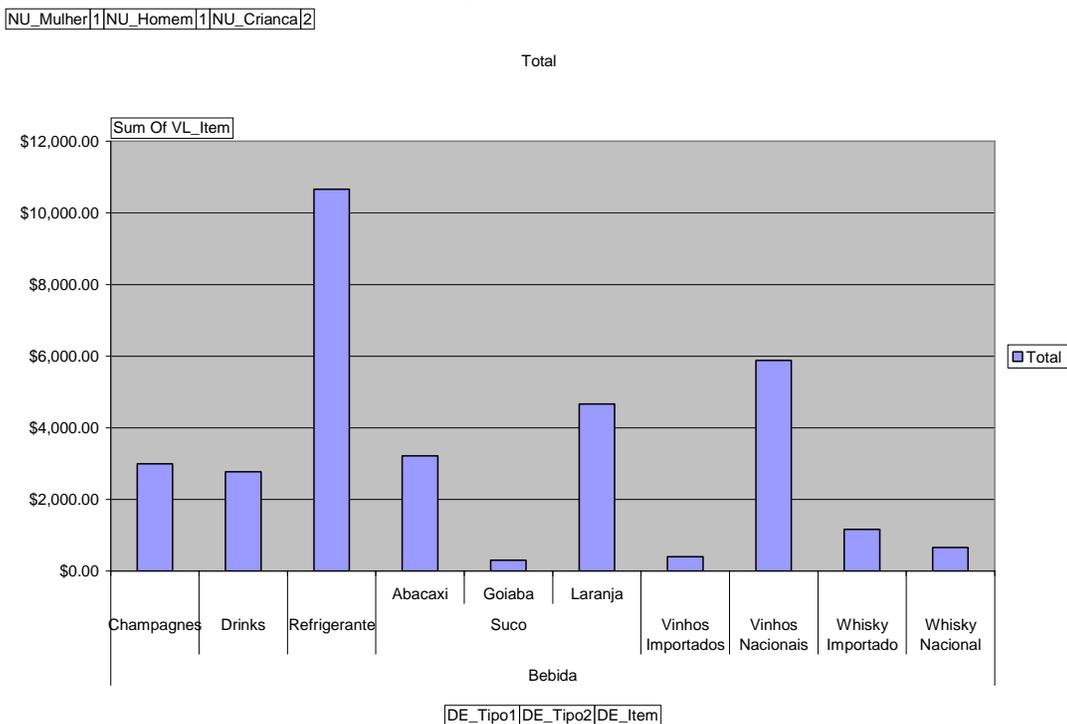


Figura 27: Consumo de bebida quando um casal e duas crianças ocupam a mesa.
 Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor com base nos dados do restaurante *Maxim*.

Tipo	Consumo médio por pessoa
Casais	R\$ 37,88
Grupo Homens e Mulheres	R\$ 36,55
Grupo Homens	R\$ 25,92
Famílias	R\$ 23,43
Grupo Mulheres	R\$ 22,24

Tabela 5: Clientes mais valiosos

Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor com base nos dados do restaurante *Maxim*

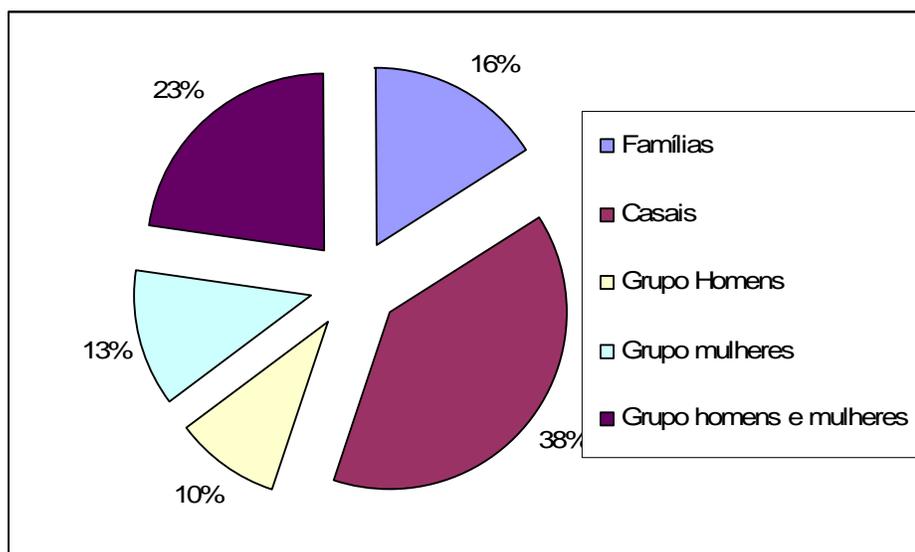


Figura 27: Tipos de clientes que frequentaram o restaurante em 2006.

Fonte: Cubo OLAP criado no *Microsoft Excel* pelo autor com base nos dados do restaurante *Maxim*

Nos últimos gráficos e relatórios gerados é possível perceber a diferença de consumo entre um casal e uma família. A bebida que os casais mais gastam dinheiro consumindo é o vinho, já as famílias consomem mais refrigerante.

Os clientes mais valiosos do restaurante são os casais, ele consomem em média R\$ 37,88 por pessoa e os menos valiosos são grupo de mulheres, onde o consumo médio é de R\$ 22,24.

4. Conclusão

A realização desse trabalho abriu muito a visão que o autor tinha sobre as tecnologias existentes na área dos PDAs, assim como os benefícios que esses pequenos computadores podem trazer. Utilizando essas tecnologias dentro de um restaurante pode-se obter grandes resultados como a melhora do atendimento aos clientes e o aumento da visão sobre o comportamento dos clientes.

Os ambientes de programação *J2ME* e *.Net Compact Framework* oferecem uma programação madura e excelentes soluções para o desenvolvimento de aplicativos voltados para dispositivos móveis. São tecnologias difíceis de ser comparadas, este tipo de análise é muito complexo, pois para fazer uma comparação que seja eficiente é preciso incluir dimensões não tecnológicas como a viabilidade, aceitação do mercado, ferramentas de teste e desenvolvimento, padronização e coerência da plataforma.

A escolha da tecnologias *Wireless* a ser utilizada envolve basicamente três fatores, distância, qualidade da transmissão e economia. Cada padrão *Wireless* atende uma solução específica. No desenvolvimento do sistema para o restaurante, o padrão que mais se adapta é o IEEE 802.11b, este tem um alto alcance (de até 300 metros) e possui um baixo custo.

Com este trabalho também foi possível ter um conhecimento das vantagens que um *DW* pode trazer para um restaurante, principalmente quando se fala em apoio a tomada de decisão. Com ele é possível elaborar conclusões e respostas muito mais rápidas e eficazes. Com o *Data Mart* desenvolvido no aplicativo em questão pode-se ter uma visão muito mais detalhada sobre o comportamento dos clientes do restaurante, fazendo com que a alta gerência possa tomar decisões baseadas nas tais respostas.

5. Referências Bibliográficas

SIQUEIRA, JOSÉ ROBERTO. Programação do Pocket PC com eMbedded Visual Basic, 2005.

JOHNSON, DAVE. and Rick Broida. How to Do Everything with Your Palm Handheld, Fifth Edition, 1998.

RICHARD, D. H. Como usar o *Data Warehouse*. Rio de Janeiro: Infobook, 1997.

INMON, W. H. Building the *Data Warehouse*, Terceira edição, Wiley, 2002.

WILSON, JOSÉ DE OLIVEIRA. Editora Visual Books, 2004.

GAST, MATTHEW. 802.11-Wireless Network: The Definitive Guide .O'REILLY, 2002.

KIMBALL, R. WILEY. The Data Warehouse Lifecycle Toolkit experts methods for designing, developing, and deploying Data Warehouse, 1998.

HAYKIN, SIMON. Editora Wiley - Communication Systems (4th edition), 2003

LOUREIRO, ANTONIO A. F. Comunicação sem fio e computação móvel: Tecnologias, desafios e oportunidades. Artigo SBC, 2003.

SANTOS, DANIEL DOS. Mundo sem fio, PCWORLD n° 167 – IDG Brasil, 2006

SUN MICROSYSTEM; mobile Information Device Profile (MIDP); JSR 37, JSR 118 Overview. Disponível em: <http://java.sun.com/products/midp/>, 2006

CEFETSP. Análise das modalidades de um data warehouse (DW). Disponível em: <http://www.cefetsp.br/edu/sinergia/6p9c.html> acessado em 23/08/2006;

SCIELO. Gestão do conhecimento: uma revisão crítica orientada pela abordagem da criação do conhecimento. Disponível em: <http://www.scielo.br> acessado em 08 /09/2006;

WIKI. <http://pt.wikipedia.org>, 2006

ELLIS, JON. YOUNG, MARK . JSR 172: J2ME Web Services Specification. Disponível em <http://jcp.org/>, acessado em 14/5/2005

TRELLIS, Exemplo de rede sem fio (A Empresa que Mudou o Conceito de Tecnologia Nacional). Disponível em: <http://www.trellis.com.br>. Acessado em: 13/11/2005.

PINHEIROS, Jose Mauricio Santos, **Vulnerabilidades em redes wireless**, www.redesdecoputadores.com 2005.

Anexos A – Especificação da Modelagem Multidimensional

Fato Pedido

ID_Restaurante	Integer(4)	- Chave estrangeira para a dimensão Restaurante
ID_Mesa	Integer(4)	- Chave estrangeira para a dimensão Mesa
ID_Garcon	Integer(4)	- Chave estrangeira para a dimensão Garçon
ID_Item	Integer(4)	- Chave estrangeira para a dimensão Item
ID_Tempo	Integer(4)	- Chave estrangeira para a dimensão Tempo
ID_Feriado	Integer(4)	- Chave estrangeira para a dimensão Feriado
ID_Dia_Semana	Integer(4)	- Chave estrangeira para a dimensão Dia Semana
ID_TipoCliente	Integer(4)	- Chave estrangeira para a dimensão TipoCliente
NU_Hora	Integer(2)	- Hora do Pedido
NU_Dia	Integer(2)	- Dia do Pedido
NU_Mes	Integer(2)	- Mês do Pedido
NU_Ano	Integer(2)	- Ano do Pedido
NU_Item	Integer(4)	- Item do Cardápio
NU_Mulher	Integer(4)	- Quantidade de mulheres na mesa
NU_Homem	Integer(4)	- Quantidade de homens na mesa
NU_Crianca	Integer(4)	- Quantidade de crianças na mesa
NU_Semana_Ano	Integer(2)	- Semana do ano do pedido (de 0 à 54)
VL_Temperatura	Integer(2)	- Temperatura na hora do pedido
NU_Temp_Espera	Integer(2)	- Tempo de espera do Cliente no pedido

Dimensão Cardapio

ID_Item	Integer(4)	- Identificação do Item do Cardápio
DE_Tipo1	VarChar(50)	- Descrição da Categoria 1 do Item
DE_Tipo2	VarChar(50)	- Descrição da Categoria 2 do Item
DE_Item	VarChar(50)	- Descrição do Item
DE_Detalhe	VarChar(50)	- Detalhes do Item
VL_Item	Decimal(2)	- Preço do Item

Dimensão DiaSemana

ID_Dia_Semana	Integer(4)	- Identificação do Dia da Semana
DE_Dia_Semana	VarChar(50)	- Descrição do Dia da Semana

Dimensão Feriado

ID_Feriado	Integer(4)	- Identificação do Feriado
DE_Feriado	VarChar(50)	- Descrição do Feriado

Dimensao Garcom

ID_Garcon	Integer(4)	- Identificação do Garçon
NM_Garcon	VarChar(50)	- Nome do garçon
DT_Nascimento	Date	- Data de nascimento do garçon
NM_Sexo	char(10)	- Sexo do garçon
Carga_Horario	int(4)	- Quantas horas o garçon trabalha por semana

Dimensao Cliente

ID_Tipo_Cliente Integer(4) - Identificação do Tipo de Cliente
DE_Tipo_Cliente Varchar(50)- Descrição do Tipo de Cliente

Dimensao Mesa

ID_Mesa Integer(4) - Identificação da mesa
DE_Localizacao Varchar(50)- Localização da mesa
NU_Assentos Integer(4) - Número de assentos que tem a mesa

Dimensao Restaurante

ID_Restaurante Integer(1) - Identificação do restaurante
DE_Restaurante Varchar(50)- Descrição do restaurante
NU_Assentos Integer(4) - Número de assentos que o restaurante possui
NU_Funcionarios Integer(1) - Número de funcionários existentes no Restaura

Dimensao Tempo

ID_Tempo Integer(4) - Identificação do tempo
DE_Tempo Varchar(50)- Descrição do tempo

Anexos B – Dados contidos nas tabelas de dimensão usado na geração dos gráficos

DIM Tempo

ID_Tempo	DE_Tempo
1	céu claro
2	poucas nuvens
3	nublado
4	pancadas de chuva
5	chuva
6	geada
7	neve

DIM Garcom

ID	NM_Garcom	DT_Nascimento	NM_Sexo	Carga_Horária
1	Antonio Pereira	28/01/73	Masculino	45
2	Jomar dos Santos	10/07/78	Masculino	45
3	Lucas Almeida	01/11/81	Masculino	45
4	Carlos Andrade	20/09/77	Masculino	40
5	Fernando da Silva	19/03/84	Masculino	45
6	Joaquim Vieira	11/02/80	Masculino	45
7	Roberto Lopes	01/10/70	Masculino	20
8	Marcia Garcia	10/02/84	Feminino	40
9	Maria Simas	29/01/82	Feminino	45
10	Joao Carlos	18/08/79	Masculino	45
11	Claudia Ferraz	02/10/83	Feminino	45

DIM Mesa

ID	DE_Localizacao	NU_Assentos
1	Área 1	8
2	Área 1	8
3	Área 1	8
4	Área 1	8
5	Área 1	8
6	Área 1	8
7	Área 1	8
8	Área 1	8
9	Área 2	4
10	Área 2	4
11	Área 2	4
12	Área 2	4
13	Área 2	4
14	Área 2	4
15	Área 2	4
16	Área 2	4

DIM Cardápio

ID	DE_Tipo1	DE_Tipo2	DE_Item
1	Bebida	Drinks	Marguerita
2	Bebida	Drinks	Piña Colada
3	Bebida	Drinks	Cuba Libre
4	Bebida	Drinks	Caipirinha
5	Bebida	Drinks	Caipiroska
6	Bebida	Drinks	Cocktail de frutas
7	Bebida	Refrigerante	Sprite
8	Bebida	Refrigerante	Água com Gás
9	Bebida	Whisky Nacional	"Teache□□s"
10	Bebida	Whisky Nacional	Bell's
11	Bebida	Whisky Importado	"Johnnie Walker [Red Label] "
12	Bebida	Whisky Importado	Johnnie Walker [Black Label]
13	Bebida	Whisky Importado	Ballantine's
14	Bebida	Champagnes	"De Greville"
15	Bebida	Champagnes	M. Chandon
16	Bebida	Champagnes	Georges Albert
17	Bebida	Vinhos Nacionais	"Liebfraumilch"
18	Bebida	Vinhos Nacionais	Almaden Tinto
19	Bebida	Vinhos Nacionais	Almaden Branco
20	Bebida	Vinhos Nacionais	Chateau Duvalier
21	Bebida	Vinhos Nacionais	Miolo Branco
22	Bebida	Vinhos Nacionais	Miolo Tinto
23	Bebida	Vinhos Importados	Vinho do Porto
24	Bebida	Refrigerante	Água sem Gás
25	Bebida	Refrigerante	Fanta
26	Bebida	Refrigerante	Coca Cola
27	Bebida	Refrigerante	Pepsi
28	Bebida	Suco	Abacaxi
29	Bebida	Suco	Laranja
30	Bebida	Suco	Goiaba
31	Comida	Salada	Salada Maxim's
32	Comida	Salada	Salada Mista
33	Comida	Carnes	Filé à parmegiana
34	Comida	Carnes	Filé ao molho de cogumelos
35	Comida	Carnes	Filé ao poivre
36	Comida	Carnes	Picanha na chapa
37	Comida	Massas	Penne ao sugo
38	Comida	Massas	Penne ao molho pesto
39	Comida	Massas	Penne ao molho funghi
40	Comida	Aves	Frango tailandês
41	Comida	Aves	Frango à cubana
42	Comida	Sobremesa	Sorvete de Creme
43	Comida	Sobremesa	Sorvete de Chocolate

DIM_Feriado

ID_Feriado	DE_Feriado
0	Não feriado
1	Primeiro do Ano
2	Segunda de Carnaval
3	Terça de Carnaval
4	Pascoa
5	Tiradentes
6	Dia do Trabalho
7	Corpus Christi
8	Independência do Brazil
9	Dia da Crianças
10	Finados
11	Proclamação da República
12	Natal

DIM_Dia_Semana

ID_Dia_Semana	DE_Dia_Semana
0	0 - Domingo
1	1 - Segunda
2	2 - Terça
3	3 - Quarta
4	4 - Quinta
5	5 - Sexta
6	6 - Sábado

Anexos C – Artigo

Automação e Gestão de Serviços de Restaurante usando Tecnologia Wireless em um PDA

Felipe Simas Silva

Universidade Federal de Santa Catarina

fel_simas@hotmail.com

RESUMO

O objetivo deste trabalho é automatizar o processo de Gestão de Serviços em um restaurante. Para alcançar essa meta é feito um estudo comparativo entre as mais variadas tecnologias existentes na área de comunicação sem fio, PDAs e Data Warehouse. A partir deste estudo é possível ter um conhecimento do que utilizar e como desenvolver um sistema capaz de auxiliar nos processos de um restaurante. Com o desenvolvimento desse projeto pôde-se ter uma melhora considerável no atendimento dos clientes de um restaurante, pois ocorre um aumento da velocidade na entrega dos pedidos e através de relatórios e gráficos as informações relativas ao restaurante tornam-se muito mais claras e eficientes.

Palavras-Chave

Wi-fi, Bluetooth, Data Warehouse, IEEE 802.11, .NET Compact Framework

1. INTRODUÇÃO

Com o constante avanço das tecnologias voltadas aos dispositivos móveis, muitos processos operacionais, nas mais diversas áreas, se tornam cada dia mais rápidos e eficientes. O propósito deste trabalho é apresentar uma solução eficaz que auxilie no processo de automação e gestão de serviços de um restaurante. Será feito um estudo sobre as tecnologias mais recentes citando as principais características de cada uma. Existem diversas tecnologias que podem ser utilizadas para tal finalidade, no entanto o trabalho se concentra em algumas delas, nas quais possibilitam a viabilidade do desenvolvimento de um sistema que atenda aos objetivos desse trabalho. Muitos fatores estão envolvidos na escolha das ferramentas, elas possuem características distintas e atendem soluções específicas.

O sistema permitirá que os pedidos em um restaurante possam ser feitos através de um PDA (utilizado por um garçom) e enviados para cozinha sem o auxílio de nenhum cabo (*Wireless*). Com os dados capturados pelo PDA, serão gerados relatórios e gráficos com o intuito de responder as mais variadas perguntas que auxiliem na área de planejamento estratégico dos negócios do restaurante. Com isso pretende-se entender mais sobre o comportamento dos clientes e conseqüentemente propor melhorias para o restaurante.

2. O Problema

A maioria dos restaurantes possui algum tipo de deficiência no atendimento aos seus clientes, muitas vezes quando o cliente faz um pedido, o pedido demora para chegar ou chega errado, isto acontece geralmente quando esse processo é manual e exige um certo deslocamento físico tanto da parte do garçom quanto do cozinheiro. A automatização deste processo pode reduzir bastante essas falhas, garantindo uma maior segurança e rapidez nos pedidos. Outra deficiência existente é que os dados relativos ao que está acontecendo com o restaurante dificilmente são registrados como por exemplo, que tipo de cliente fez determinado pedido, se é a primeira vez do cliente, ou como estava o tempo na hora do pedido, enfim, quanto mais informações forem registradas, mais eficiente será uma futura inferência. Este trabalho propõe uma solução que auxilie nesse sentido.

3. A Metodologia Utilizada

Para alcançar os objetivos deste trabalho, o desenvolvimento do mesmo terá 3 etapas. Na primeira, será feito um estudo sobre PDAs e tecnologias que dão suporte para esses dispositivos. Serão comparados os padrões de comunicação sem fio disponíveis atualmente, e também as atuais tecnologias existentes para o desenvolvimento de aplicativos para dispositivos móveis.

Na segunda etapa será abordado o uso de uma *Data Warehouse*, citando as vantagens que essa ferramenta pode trazer à uma empresa com as

valiosas informações estratégicas. Nessa etapa também constam alguns itens que fazem parte do conceito *Data Warehouse* como ferramenta de *front-end*, extração e *metadados*.

A terceira etapa é o desenvolvimento de uma aplicação com base em todos os conceitos abordados. A aplicação dará suporte ao processo de gestão de negócios e serviços em um restaurante. Uma parte do desenvolvimento será um software para um PDA que supra as necessidades de um restaurante e ao mesmo tempo capture informações relevantes para uma futura análise. Outra parte terá o foco na apresentação dos dados, visando responder as mais variadas perguntas para dar suporte às tomadas de decisões para isso será utilizado um *Data Warehouse* e um ferramenta *front-end*.

Do ponto de vista dos procedimentos técnicos, a pesquisa é de cunho: bibliográfico, elaborada a partir de material já publicado (sites e livros); documental, a partir de materiais que não receberam tratamento analítico (Manual de dispositivos móveis); e experimental, através da determinação do objeto de estudo, da seleção das variáveis, das formas de controle e da observação dos efeitos produzidos. (GIL, 1991).

4. PDA's

PDAs são pequenos computadores que podem ser transportados dentro do bolso, e que possuem características muito semelhante a de um PC, no entanto com algumas limitações. Existem diversos tipos de PDA, os mais atuais possuem tela colorida, acesso a Internet via *Wireless*, *Touch Screen* (mecanismo que possibilita o envio de comandos através de toques na tela), *Bluetooth*, câmera digital, MP3 player, controle remoto e até mesmo GPS. Eles também podem ser utilizados como calendário, relógio, agenda, jogos, calculadora e o uso de qualquer aplicativo dentro das suas limitações de memória e processamento.

As principais vantagens desses aparelhos é o fácil transporte e a conveniência de não precisar o auxílio de nenhum cabo, tanto para conectar-se à rede de dados quanto a um suporte de energia.

O número de PDAs vem crescendo progressivamente no mundo inteiro, empresas e usuários comuns estão cada vez mais adotando essa tecnologia, e tendências indicam que em pouco tempo os *Smartphones* (fusão entre um telefone celular e um PDA) serão maioria absoluta no mercado. (HAYKIN, 2003)

No mercado atual os principais tipos de PDAs existentes são o *Pocket PC* (Figura 1), desenvolvido pela *Microsoft* e o *PalmOne* (Figura 2), desenvolvido pela *PalmSource, Inc*

O *PalmOne* utiliza o sistema operacional *Palm OS*, a versão mais atual existente é a 6.0 *Cobalt*, sua base (*Kernel*) nas próximas versões deverá ser baseada no *SO Linux*. Este sistema é muito leve, rápido e seguro e tem uma imensa variedade de programas disponíveis na Internet tais como editor de texto, planilha eletrônica, agenda, calculadora, programas de acesso à Internet, envio de email, entre outros. (DAVE2005). A linguagem de programação

mais utilizada para criação de softwares nesses dispositivos é a J2ME.

Já o *Pocket PC* utiliza o SO *Windows Mobile*, a versão mais atual desse sistema é a versão 5.0. O *Windows Mobile* é um Sistema Operacional compatível com o *Windows desktop*. Nele é possível “rodar” vários softwares da *Microsoft* parecidos com os que existem no *Windows* tradicional, tais como *Pocket Internet Explorer (HTML e WAP)*, *MSN Messenger*, *Pocket Word*, *Pocket Excel*, *Windows Media Player*, entre outros softwares. E também ele é capaz de sincronizar informações com um PC que utiliza *Windows* como SO.

Os *Pocket PCs* rodam quase que exclusivamente na arquitetura *intel*. Os processadores *Intel® PCA Application*, como o processador *Intel® PXA250 Application* estão nos dispositivos mais recentes e são baseados na tecnologia *Intel XScale®*. Os *Pocket PCs* estão disponíveis em mais de 30 empresas, dentro das principais estão: *Casio*, *Dell*, *HP*, *Compaq* e *Toshiba*.

Para criar softwares para o *Pocket PC* a tecnologia mais utilizada é a *.NET Compact Framework*, esta é bem parecida com *.NET Framework* que é utilizada para *Windows desktop*, porém um software desenvolvido para *Pocket PC* não funciona em um sistema operacional *Windows desktop*, e vice-versa. Apesar de ambos serem da *Microsoft* são sistemas totalmente diferentes.

5. Comunicação sem fio

Uma rede de comunicação sem fio pode utilizar sinais de rádio frequência ou infravermelho para transmissão de dados, na maioria das vezes trata-se de rádio frequência, essas ondas operam em uma frequência aberta e não necessitam de qualquer tipo de licença ou autorização.

Os sinais de rádio frequência que carregam informações de um ponto ao outro são chamados de onda portadora. E a forma de inserção de uma informação no sinal é chamado de modulação. A modulação permite que as informações sejam transportadas embutida nos parâmetros de frequência, amplitude ou fase da portadora.

São dois, os principais padrões para redes de comunicação sem fio, *Bluetooth* (redes *AdHoc*) e *IEEE 802.11* (redes infra-estruturadas). O padrão utilizado no trabalho é o *IEEE 802.11b*

6. .NET Compact Framework

.Net Compact Framework é um subconjunto do *.Net Framework*, ele traz um poderoso ambiente para o desenvolvimento de aplicativos para pequenos dispositivos móveis. Todos programadores que conhecem ou desenvolvem aplicativos com o *framework .Net Framework* podem facilmente utilizar o *.Net CF*, pois são duas plataformas com muitas características em comum. Logo, uma empresa que já possui o conhecimento em *.Net Framework* pode passar a construir aplicativos para *Pocket PC* apenas com algumas adaptações. O investimento será bastante reduzido devido aos funcionários não precisarem de muito treinamento.

Dentro desta plataforma as principais vantagens nas funcionalidades são:

- Executa softwares que são independente de SO e *hardware*;
- Suporta diversos protocolos de rede e faz um perfeita conexão com serviços *Web XML*;
- Fornece ao desenvolvedor modelos que auxiliam e direcionam seus aplicativos;
- Possui uma otimização de recursos limitados do sistema;
- A geração de código nativo é obtida usando o método de compilação *JIT (Just-In-Time)*. Esse método possui uma ótima performance se comparado com outros métodos de compilação nessa mesma área.

O *.NET CF* pode compartilhar códigos através de computadores ou de múltiplos aparelhos. Esse código pode fazer parte da camada de acesso a dados, pode ser uma lógica de negócio ou parte da camada de *XML Web Service*. Isso simplifica consideravelmente o desenvolvimento de aplicações rodam em múltiplos dispositivos.

A plataforma oferece um ambiente seguro e robusto para a execução do código no cliente. O modelo *Managed Code* (código gerenciado) aumenta a confiabilidade do código; a execução desse modelo assegura que o comportamento da aplicação não irá travar o aparelho e também que códigos maliciosos não tenham nenhuma permissão para acessar os recursos de segurança do sistema.

A integração do *.NET CF* com *XML Web Services* é bastante simples e eficiente, funciona como no *.NET Framework*, invocar um *XML Web Service* é tão simples com chamar uma função. *XML Web Services* são muito usado em *Smart Devices*, eles permitem que as aplicações se comuniquem uma com a outra sem levar em consideração o SO ou a linguagem de programação.

Os desenvolvedores *.NET CF* possuem um ampla portabilidade e reusabilidade para escrever para o modelo de programação comum, mas ele é também desenhado para ser estendido com *Class Library* que expõem características únicas para cada família de aparelhos, ou para um modelo de aparelho específico. Isto permite aproximar desenvolvedores para criar as melhores possibilidades para cada aparelho, enquanto usa novamente conhecimentos e códigos.

O *.NET CF* suporta o desenvolvimento em *C#, VB .NET, C++ e J#* além de que no futuro irá suportar outras linguagens devido ao *Common Language Interface (CLI)*.

7. O Projeto

O projeto dará todas as condições a um garçom realizar pedidos e enviar à cozinha do restaurante sem precisar se locomover, e quando o pedido estiver pronto o cozinheiro por sua vez enviará uma notificação ao garçom, tudo isto sem o auxílio de nenhum cabo. Toda essa troca de informação, junto com algumas informações referente aos clientes serão armazenadas em uma base de dados. Com uma base de dados robusta e com o auxílio de um DW será possível fazer diversas inferências sobre o restaurante e o comportamento de seus clientes. Nesse sentido, acontecerão melhorias na qualidade do restaurante,

pois o mesmo terá o processo de pedidos automatizado e será visto de uma nova forma facilitando as tomadas de decisões.

7.1 Avaliação

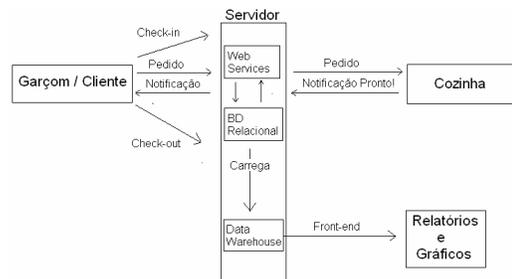
Antes de dar início ao projeto é feita uma análise dos processos em um restaurante que podem ser automatizados. O processo de pedidos é o que pode sofrer maiores melhorias. Pois uma vez que este processo é automatizado o tempo economizado com a soma de todos os pedidos é muito grande, podendo gerar assim um lucro maior para o restaurante.

Outro ponto que deve ser considerado é que a maioria dos restaurantes utiliza um sistema que não é capaz de registrar todas as informações sobre seus processos. E essas informações podem ser muito valiosas e capazes de gerar uma excelente fonte de conhecimento sobre o restaurante.

7.2 Desenvolvimento

Para o desenvolvimento dessa aplicação serão utilizados o PDA Pocket PC com o Wi-Fi IEEE 802.11b, a plataforma .Net Compact Framework, a linguagem de programação VB.NET e o banco de dados SQL Server 2000. Esta decisão foi baseada na viabilidade e facilidade de adquirir e aprender essas novas tecnologias, uma vez que o autor já possui um conhecimento do Framework .Net Framework e também pelas eficientes e robustas ferramentas que existem disponíveis para estas tecnologias. Na última parte do desenvolvimento do aplicativo será criado um Data Mart para ajudar na visualização dos dados relativos ao restaurante, o front-end utilizado para este Data Mart será o PivotTable and PivotChart do Microsoft Excel.

Uma parte da aplicação será localizada em um servidor, e outra parte em um Pocket PC. A comunicação entre o PDA e o servidor ocorrerá através de Web Services, este permite que diferentes aplicações localizadas em diferentes máquinas possam trocar informações em formato XML. No aplicativo em questão, o garçom registrará as informações no Pocket PC, e este por sua vez enviará as informações para o servidor através de funções Web Services.



7.3 Modelagem Dimensional

Modelagem Multidimensional é uma modelagem de dados onde os dados são estruturados em diversas dimensões e também desnormalizados. O tipo de modelagem dimensional mais utilizado é o *Star*

Schema, em português, Esquema Estrela, este esquema foi desenvolvido por Ralph Kimball, um dos precursores da *Data Warehouse*.

A principal característica do *Star Schema* é a alta redundância de dados, que tem a finalidade de melhorar o desempenho na busca de informações. Este modelo é composto basicamente por dois tipos de tabelas: as de Fato e as de Dimensão. As tabelas de fato são grandes tabelas centrais e armazenam os fatos ocorridos e as chaves para características correspondentes nas tabelas dimensionais, que armazenam as descrições do negócio. Cada tabela de Dimensão possui uma única chave primária, e o conjunto dessas chaves primárias formará a chave composta da tabela de Fato. Cada negócio ou assunto possui sua tabela de Fato e suas respectivas tabelas de Dimensão.

O modelo dimensional deste trabalho terá uma tabela fato que se chamará “Fato Pedido” e também 6 tabelas dimensionais.

7.4 Front end (apresentação dos dados)

A maneira pela qual a alta gerência irá analisar os dados relativos ao restaurante é de extrema importância. O objetivo é conseguir eliminar todos os dados que não interessem a determinada questão e somente apresentar o que interesse e de uma forma clara.

Nesta aplicação *Front end* utilizado é *PivotTable and PivotChart* do *Microsoft Excel*. É claro que existem ferramentas de *front end* muito melhores e com mais funcionalidades, mas para o problema em questão o *Excel* atende todas as necessidades. Com essa ferramenta pode-se gerar relatórios e gráficos a partir da escolha de certas variáveis e dimensões. E o relatório pode ser manipulado dinamicamente, ou seja, é possível trocar dimensões e variáveis a qualquer momento, e isso com um simples arraste do mouse (*drag and drop*).

Com o *Data Mart* carregado e uma ferramenta *front end* é possível responder as mais variadas perguntas a respeito do restaurante, o projeto apenas se foca em algumas. São elas:

- Quais os horários e dias de maior demanda?
- Quais as comidas mais pedidas por dia e por horário?
- Qual a interferência do tempo no consumo de bebida e comida?
- Quais e quantas bebidas foram vendidas em dia de sol (detalhe no suco)?
- Como foi a movimentação nos feriados dos últimos 3 anos?
- Classificar garçons pela eficiência.
- Quais os tipo de clientes do restaurante e o que eles consomem?

8. Conclusão

A realização desse trabalho abriu muito a visão que o autor tinha sobre as tecnologias existentes na área dos PDAs, assim como os benefícios que esses pequenos computadores podem trazer. Utilizando essas tecnologias dentro de um restaurante pode-se

obter grandes resultados como a melhora do atendimento aos clientes e o aumento da visão sobre o comportamento dos clientes.

Os ambientes de programação *J2ME* e *.Net Compact Framework* oferecem uma programação madura e excelentes soluções para o desenvolvimento de aplicativos voltados para dispositivos móveis. São tecnologias difíceis de ser comparadas, este tipo de análise é muito complexo, pois para fazer uma comparação que seja eficiente é preciso incluir dimensões não tecnológicas como a viabilidade, aceitação do mercado, ferramentas de teste e desenvolvimento, padronização e coerência da plataforma.

A escolha da tecnologias *Wireless* a ser utilizada envolve basicamente três fatores, distância, qualidade da transmissão e economia. Cada padrão *Wireless* atende uma solução específica. No desenvolvimento do sistema para o restaurante, o padrão que mais se adapta é o IEEE 802.11b, este tem um alto alcance (de até 300 metros) e possui um baixo custo.

Com este trabalho também foi possível ter um conhecimento das vantagens que um *DW* pode trazer para um restaurante, principalmente quando se fala em apoio a tomada de decisão. Com ele é possível elaborar conclusões e respostas muito mais rápidas e eficazes. Com o *Data Mart* desenvolvido no aplicativo em questão pode-se ter uma visão muito mais detalhada sobre o comportamento dos clientes do restaurante, fazendo com que a alta gerência possa tomar decisões baseadas nas tais respostas.

9. Referências Bibliográficas

[1] SIQUEIRA, JOSÉ ROBERTO. Programação do Pocket PC com eMbedded Visual Basic, 2005.

[2] JOHNSON, DAVE. and Rick Broida. How to Do Everything with Your Palm Handheld, Fifth Edition, 1998.

[4] RICHARD, D. H. Como usar o *Data Warehouse*. Rio de Janeiro: Infobook, 1997.

[5] INMON, W. H. Building the *Data Warehouse*, Terceira edição, Wiley, 2002.

[6] GAST, MATTHEW. 802.11-Wireless Network: The Definitive Guide .O'REILLY, 2002.

[7] KIMBALL, R. WILEY. The Data Warehouse Lifecycle Toolkit experts methods for designing, developing, and deploying Data Warehouse, 1998.

[8] HAYKIN, SIMON. Editora Wiley - Communication Systems (4th edition), 2003

[9] LOUREIRO, ANTONIO A. F. Comunicação sem fio e computação móvel: Tecnologias, desafios e oportunidades. Artigo SBC, 2003.

[10] SANTOS, DANIEL DOS. Mundo sem fio, PCWORLD n° 167 – IDG Brasil, 2006
SCIELO. Gestão do conhecimento: uma revisão crítica orientada pela abordagem da criação do conhecimento. Disponível em: <http://www.scielo.br> acessado em 08 /09/2006;

[11] ELLIS, JON. YOUNG, MARK . JSR 172: J2ME Web Services Specification. Disponível em <http://jcp.org/>, acessado em 14/5/2005

[12] TRELIS, Exemplo de rede sem fio (A Empresa que Mudou o Conceito de Tecnologia Nacional). Disponível em: <http://www.trellis.com.br>. Acessado em: 13/11/2005.