UNIVERSIDADE REGIONAL DE BLUMENAU

CENTRO DE CIÊNCIAS EXATAS E NATURAIS

CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

PROTÓTIPO DE SOFTWARE PARA ENVIO DE MENSAGENS CRIPTOGRAFADAS PARA UM DISPOSITIVO MÓVEL UTILIZANDO A PLATAFORMA .NET

ROBSON RAMOS

BLUMENAU
2004

2004/2-42

ROBSON RAMOS

PROTÓTIPO DE SOFTWARE PARA ENVIO DE MENSAGENS CRIPTOGRAFADAS PARA UM DISPOSITIVO MÓVEL UTILIZANDO A PLATAFORMA .NET

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Ciência da Computação — Bacharelado.

Prof. Francisco Adell Péricas - Orientador

BLUMENAU
2004

2004/2-42

PROTÓTIPO DE SOFTWARE PARA ENVIO DE MENSAGENS CRIPTOGRAFADAS PARA UM DISPOSITIVO MÓVEL UTILIZANDO A PLATAFORMA .NET

Por

ROBSON RAMOS

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente:	Prof. Francisco Adell Péricas – Orientador, FURB
Membro:	Prof
Membro:	Prof.

Blumenau, 10 de novembro de 2004

Dedico este trabalho a minha família, amigos e todas as pessoas que direta ou indiretamente me ajudaram na realização deste.

“Não é a espécie mais forte que sobrevive,
nem a mais inteligente, e sim a mais receptiva
a mudanças.”

Charles Darwin

AGRADECIMENTOS

A Deus, que me acompanha em todos os momentos da minha vida, me fornecendo paz, alegria e esperança.

Aos meus pais Valmor (in memoriam) e Osnilda que forneceram todos os valores necessários para formação de meu caráter.

À minha irmã Grazielle que sempre me acompanhou e incentivou está minha caminhada.

À minha querida esposa Helena, que agüentou esses anos de faculdade longe de seu convívio e sempre me encorajou a seguir em frente.

Aos meus amigos, pela ajuda prestada neste trabalho.

Ao meu orientador, Francisco Adell Péricas, por ter me ajudado na conclusão deste trabalho.

RESUMO

Este trabalho apresenta um estudo realizado sobre a plataforma .NET, com o foco voltado para implementação de softwares para dispositivos móveis. O protótipo desenvolvido tem por objetivo proteger as informações das empresas até o usuário final através de criptografia por chave secreta. O protótipo possui dois softwares independentes, sendo um instalado no *desktop* e o outro no dispositivo móvel, onde a ligação da empresa com o aparelho ocorrerá através de um *Web service*. É sabido que a informação atualmente é o bem mais precioso da empresa, e este protótipo vem sugerir um meio de se unir as novas tecnologias com a segurança da criptografia.

Palavras chaves: Plataforma .NET, .NET *Framework*, .NET *Compact Framework*, Criptografia, Dispositivos Móveis.

ABSTRACT

This work presents a study about the platform NET, with the focus directed toward implementation of softwares for mobile devices. The archetype is intended to protect the information of companies until the final user through cryptography with private key. The archetype has two independent softwares, being one installed in desktop and the other in the mobile device, which link of the company with the device will occur through a Web service. The information currently is the most precious one of the company, and this archetype comes to suggest a way of joining the new technologies with the security of the cryptography.

Key-Words: Platform .NET; .NET *Framework*; .NET *Compact Framework*; Cryptography, Mobile Devices.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

CLR – *Common Language Runtime*

MSIL – *Microsoft Intermediate Language*

DFD – Diagrama de Fluxo de Dados

MER – Modelo de Entidade de Relacionamento lógico

SMS - *Short Message Service*

HTML – *Hypertext Markup Language*

XML – *Extensible Markup Language*

SOAP - *Simple Object Access Protocol*

XSLT – *Extensible Stylesheet Language for Transformations*

JIT – *Just-In-Time*

GSM – *Global System for Mobile*

CDMA – *Code-Division Multiple Acess*

WAP – *Wireless Application Protocol*

WML – *Wireless Markup Language*

CTS - *Common Type System*

DES - *Data Encryption Standard*

IDEA - *International Data Encryption Algorithm*

SUMÁRIO

1 INTRODUÇÃO

Nos tempos atuais a influência dos dispositivos móveis na sociedade é cada vez mais intensa, uma vez que possibilitam maior agilidade na obtenção das informações necessárias para a tomada de decisões. Fortes (2004, p. 63) cita como exemplo a “empresa farmacêutica Eurofarma, os vendedores levam o PDA e um *modem*. Antes de fechar o pedido, eles se conectam a um telefone 0800 para consultar a posição dos estoques e a situação do cliente”.

Segundo Dorman (2001), a tecnologia sem fio possibilita novas oportunidades de negócio, onde a mais importante trata das tecnologias baseadas na localização. Seguindo nesta linha de raciocínio pode-se dizer que dentre os dispositivos móveis o celular se destaca como um recurso de grande importância, pois deixou de ser um simples aparelho utilizado para realizar conversas e está se tornando um computador portátil.

Já é possível executar no celular vários softwares que não estão escritos na linguagem nativa do aparelho, sendo que para isso, torna-se necessária à presença de uma máquina virtual, que é utilizada para compilar o aplicativo para a linguagem nativa do aparelho em tempo de execução. A máquina virtual mais utilizada é a do Java e está presente em vários dispositivos móveis. Outra máquina virtual que está ganhando espaço no mercado é o .NET *Framework* da plataforma .NET de propriedade da Microsoft.

Este trabalho descreve a construção do protótipo de um aplicativo para celular, utilizando os recursos provenientes da plataforma .NET. Este software por sua vez transmitirá mensagens criptografadas de um servidor para dispositivos móveis.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho foi construir um protótipo de um software para transmissão de mensagens criptografadas para dispositivos móveis (celular), de forma segura.

Os objetivos específicos do trabalho são:

a) transmitir mensagens para dispositivos móveis (celular) de modo sigiloso;
b) obter mensagem no celular através de aplicativo rodando na plataforma .NET.

1.2 ESTRUTURA DO TRABALHO

Este trabalho primeiramente apresentará uma introdução, bem como seus objetivos a serem alcançados e os trabalhos correlatos. Na seqüência será mostrado um estudo sobre as vantagens e perspectivas para o futuro em relação aos aparelhos móveis (SmartPhone).

Após as considerações iniciais será apresentado um estudo mais detalhado sobre a plataforma .NET, para em seguida exibir informações sobre o protótipo, ferramentas utilizadas, especificação e implementação do trabalho.

O último capítulo tratará das considerações finais e sugestões para trabalhos futuros.

1.3 TRABALHOS CORRELATOS

Santos (2003) desenvolveu um software baseado na plataforma .NET para dispositivos móveis (PDA), com o objetivo de buscar notícias em um servidor através de uma conexão *Web*, para isso foi implementado um software cliente que busca as notícias no servidor e em seguida exibe-as na tela do PDA.

Outro trabalho de desenvolvimento de software para dispositivos móveis é do acadêmico Depiné (2002), que implementou um aplicativo em Java, que possibilita efetuar os cálculos de tempo e deslocamento necessários para realização de um rally de regularidade. O software foi construído utilizando a linguagem Java (J2ME) para equipamentos portáteis, e instalado em um dispositivo móvel (celular).

Pode-se citar ainda o trabalho da acadêmica Schaefer (2004), que desenvolveu um software em Java que coleta dados em um dispositivo móvel (celular) e depois os envia através de e-mail, onde um aplicativo construído em Delphi instalado no *PC* se conecta no servidor e ao baixar os e-mails, grava-os em uma base de dados para consultas futuras. O programa que roda no celular foi desenvolvido na plataforma J2ME.

2 DISPOSITIVOS MÓVEIS

Com o avanço da tecnologia, os dispositivos móveis estão se tornando cada vez mais poderosos e ao mesmo tempo mais acessíveis aos consumidores. Nos últimos anos, uma crescente quantidade destes dispositivos têm estado disponível no mercado.

De acordo com Pekus (2002), os aparelhos móveis são mais do que assistentes pessoais ou agendas eletrônicas, “são computadores que podem ser facilmente levados a qualquer lugar, criados para atender profissionais em movimento que necessitem de rapidez, facilidade e segurança no acesso a informações corporativas e pessoais”.

Segundo Guimarães (2004), atualmente os dispositivos móveis encontram-se de vez na ordem de prioridades das empresas, pois proporcionam agilidade e segurança no acesso das informações.

Com os *PocketPCs* ocupando o bolso de milhares de usuários em todo o mundo, começaram os desconfortos causados pela necessidade de se carregar um *PDA* em um bolso e um telefone celular em outro. A solução encontrada pelo mercado foi permitir o uso do *PocketPC* como telefone móvel, assim, foram criados os primeiros cartões e expansões com este objetivo.

Segundo Miranda (2004), diversos problemas existiam com as primeiras soluções, desde a falta de suporte interno das aplicações nativas dos *PDAs*, até a baixa capacidade da bateria, uma vez que estes cartões e expansões possuíam alto consumo de energia.

Mas atualmente um equipamento está ganhando mercado rapidamente, pois este aparelho reúne a mobilidade do telefone celular e a capacidade de processamento dos computadores; é conhecido como celular inteligente ou SmartPhone.

Conforme Guimarães (2004), o SmartPhone faz tudo o que um computador é capaz de fazer, pois gerencia e-mails, arquivos de texto, áudio e vídeo, tira fotos, navega em alta velocidade pela internet, transfere dados para outros equipamentos e o mais importante, possui o tamanho de um celular comum.

2.1 VANTAGENS DOS DISPOSITIVOS MÓVEIS

De acordo com Pekus (2002), os profissionais que desenvolvem seu trabalho fora das dependências da empresa ganham em produtividade com a utilização de aparelhos móveis, pois são versáteis, multifuncionais e de uso genérico. Para as empresas, estes equipamentos são excelentes concentradores de informações, podendo ser utilizados desde a automação de processos até a coleta de informações estratégicas.

Segundo a DoctorSys (2004), os dispositivos móveis apresentam inúmeras vantagens de uso em comparação a aparelhos fixos (*desktops*). Algumas são:

a) baixo custo dos aparelhos clientes (celulares, *Pockets PCs* e *Handhelds*);
b) portabilidade e pequeno tamanho para transporte;
c) facilidade de uso e consequentemente baixo custo de treinamento;
d) ideal para equipes de vendas (representantes e vendedores), setor de saúde (comunicação entre a equipe médica, acesso a prontuários) e soluções de colaboração.

Por serem dispositivos mais compactos e econômicos, o consumo de energia e tempo de recarga são menores e a autonomia é maior. Como a carga de aplicações embutidas nestes dispositivos é inferior quando comparado a outros equipamentos, é possível que um vendedor durante o tempo de deslocamento de um cliente para outro, possa preparar as informações básicas de um pedido e deixar disponível na tela de negociação de preços. Ao chegar no cliente, basta ligar o equipamento e a tela de negociação estará disponível imediatamente, liberando assim tempo para visitar novos clientes.

2.2 SMARTPHONE

O SmartPhone (Figura 1) é a evolução do telefone celular, pois, além de possuir todas as funções de um celular convencional, contém um ótimo suporte de dados, isto é, possui diversos atributos antes encontrados somente em *PDAs*, como maior capacidade de memória e processamento, display maior, facilidades para acesso à internet e um sistema operacional exclusivo, que é uma variação da interface utilizada nos *PocketPCs*. Este sistema é chamado de Windows Mobile for SmartPhone.

Fonte: Microsoft Corporation (2004)
Figura 1: SmartPhone Motorola Mpx200

Miranda (2004) cita que “apesar da evolução em relação a celulares comuns, ainda são muito limitados quando comparados a um PDA”.

Segundo Guimarães (2004) a transição para o SmartPhone no Brasil começou com os *palms* que permitem conversar por meio de um fone de ouvido. Exemplo é o Tungsten W, da Palm-One, lançado em parceria com a operadora Claro em outubro de 2003.

Conforme Miranda (2004), o maior objetivo de um SmartPhone é ser um dispositivo móvel pequeno e permitir um tempo mínimo de uso sem recarga de bateria. Devido a isto o processador não pode possuir um clock muito elevado, pois significará maior consumo de energia.

2.3 PERSPECTIVAS PARA O FUTURO

O mercado está cada vez mais voltado para diminuição de custos e praticidade, fazendo com que os dispositivos móveis sejam cada vez mais utilizados, pois permitem maior integração entre a empresa e o seu colaborador. Miranda (2004) cita que uma análise feita por “empresas de pesquisa apontam que as vendas de SmartPhones ultrapassarão a venda de PDAs em um futuro próximo.”

Segundo a TelecomWeb (2004), um estudo realizado pela empresa de pesquisa Canalys, mostrou que o mercado de dispositivos móveis cresceu 45% no segundo trimestre de 2004, em comparação ao mesmo período do ano passado. De acordo com a Canalys, os

aparelhos para acesso a dados e informações, como *palms*, tiveram um aumento de 26% no período, enquanto os celulares e dispositivos de voz sofisticados como SmartPhones, cresceram 70%.

De acordo com Guimarães (2004) dos 100 vendedores da Master Boi (processadora de carnes de Recife (PE)) “equipados com computadores de mão, 30 já usam o recurso de voz. ‘A vantagem é que eu não preciso pagar um celular e um *palm* para cada vendedor. Fica mais barato ter um equipamento só ’, afirma Lima gerente comercial da Master Boi”.

No Brasil com atraso de pelo menos dois anos em relação aos Estados Unidos, os usuários descobriram que o computador de mão é bem mais que uma agenda eletrônica cara. Conforme Guimarães (2004) no ano passado, o Brasil tinha 46,3 milhões de celulares, base que deve crescer 25% neste ano, acima da média mundial de 13%. “‘Deste total, de 8 milhões a 10 milhões de usuários têm o perfil adequado ao SmartPhone’, diz Alexandre Szapiro, vice-presidente de vendas da PalmOne Brasil”.

A tendência é cada vez mais termos serviços voltados para estes dispositivos. De acordo com a InfoExame (2004), “a Visa colocará no ar a venda de ingressos para cinema e teatro via celular, com pagamento por cartão de crédito ou Visa Electron. Totens instalados nos estabelecimentos receberão o comprovante da compra, via Bluetooth, e imprimirão o ingresso para o usuário”, a criação deste serviço está prevista para dois ou três meses após o lançamento comercial do MPx220 (lançamento da Motorola em conjunto com a Microsoft).

3 .NET

Neste capítulo serão abordados os temas: Microsoft .NET, *Microsoft Intermediate Language* (MSIL), .NET *Framework*, .NET *Compact Framework*, Windows para SmartPhone, SmartPhone *Hardware*, *Common Language Runtime* (CLR), segurança e criptografia.

3.1 MICROSOFT .NET

Microsoft .NET é uma plataforma de software que fornece várias ferramentas e tecnologias necessárias para o desenvolvimento de aplicativos voltados para os padrões de *Web Services* XML. As aplicações desenvolvidas para .NET podem comunicar-se com vários clientes, tais como celulares e *palms*.

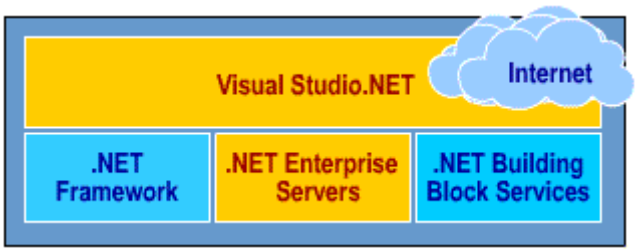
Segundo Deitel, Deitel e Nieto (2004), a tecnologia .NET permite uma independência de linguagem e sistema operacional, pois possibilita ao desenvolvedor a escolha da linguagem de programação no momento da criação do aplicativo bastando que a linguagem seja suportada pelo .NET.

Na prática um projeto pode ser escrito em Visual Basic .NET e C#, pois ambos serão compilados para linguagem intermediária Microsoft (*Microsoft Intermediate Language* - MSIL). Depois da compilação para MSIL poderá ser realizada a unificação dos programas, o que fará surgir um único programa. Outro fator que possibilita a integração é o fato que os componentes utilizados não precisam ser registrados, basta copiar o programa para outra pasta e o mesmo continuará funcionamento normalmente.

Segundo Guimarães (2003), os sistemas desenvolvidos em .NET são auto-explicativos, pois cada programa compilado contém em si informações necessárias em *metatags* que faz com que o *runtime* não precise procurar as informações no registro do Windows. Por exemplo, quando cria-se um sistema, o mesmo pode ser executado em qualquer máquina que suporte a plataforma .NET sem necessidade de instalação. Todas as informações necessárias para que o sistema seja executado são encontradas nas *metatags* dentro dele mesmo. Com isso temos um sistema auto-explicativo, pois as informações necessárias estão armazenadas dentro dele e não no registro do sistema operacional.

De acordo com Deitel, Deitel e Nieto (2004) “a estratégia .NET incorpora a idéia de reutilização de software”. Com isso um único aplicativo pode gerenciar pagamentos de contas, empréstimos e investimentos, usando os *Web services* de diversas empresas.

A plataforma .NET (Figura 2) é composta pelas seguintes tecnologias: .NET *Framework*, .NET *Enterprise Servers*, *Building Block Services* e o Microsoft Visual Studio .NET que é um ambiente de desenvolvimento para aplicações baseadas na tecnologia .NET.



Fonte: MSDN Library (2003)

Figura 2: estrutura do Microsoft .NET

3.2 MSIL

A linguagem intermediária Microsoft “é uma mistura de linguagem de baixo nível, semelhante a uma linguagem de máquina e linguagem de objeto superior” (TURTSCHI et al, 2002, p. 15).

O MSIL possui instruções para carregamento, armazenamento, chamadas de métodos, operações aritméticas e lógicas e outras informações necessárias para execução do programa. No momento da criação do MSIL são também produzidos os metadados. Estes por sua vez descrevem os tipos contidos no código, incluindo a definição de cada tipo, suas assinaturas, referências e outros dados que são utilizados durante a execução.

Após a compilação para o MSIL o código passa a ser do tipo gerenciado, isto é, o programa para ser executado precisa ser gerenciado por outro ambiente, que ficará responsável pelo gerenciamento de segurança e memória, entre outras funcionalidades.

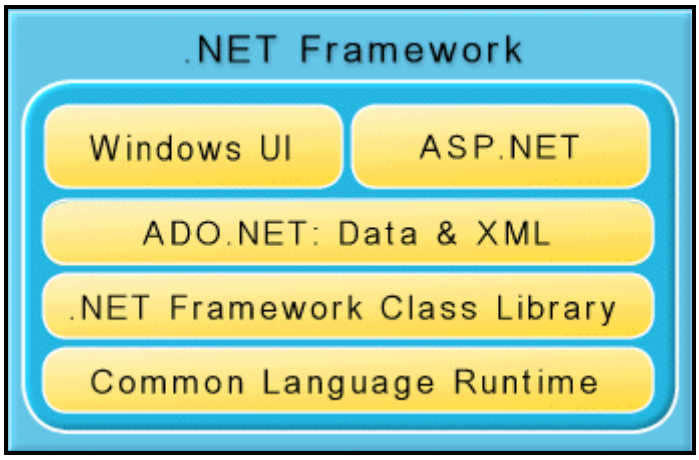
3.3 .NET FRAMEWORK

O .NET *Framework* é uma biblioteca de componentes de linguagem-neutra e ambiente de execução, sendo que esta tecnologia é responsável pelo gerenciamento e execução dos aplicativos desenvolvidos em .NET. Em outras palavras, é a máquina virtual da plataforma.

Segundo a Microsoft Corporation (2003), atualmente o *Framework* suporta as linguagens Perl, Python, COBOL, JScript, Eiffel, Haskell, Pascal, ML, Ada, APL, C, C++, Visual Basic, C#, SmallTalk, Oberon, Scheme, Mercury, Oz e Objective Caml. Além disto suporta todas as tecnologias existentes para internet incluindo *Hypertext Markup Language* (HTML), *Extensible Markup Language* (XML), *Simple Object Access Protocol* (SOAP), *Extensible Stylesheet Language for Transformations* (XSLT), *Xpath* e outros.

As classes do .NET *Framework* estão disponíveis para que os programadores possam estendê-las caso seja necessário. Uma classe somente não poderá ser estendida, se por ventura vir a ser fechada pela fabricante.

A arquitetura do .NET *Framework* (Figura 3) consiste dos seguintes componentes: *Common Language Runtime* (CLR), .NET *Framework class library*, ADO.NET e ASP.NET.



Fonte: MSDN Library (2003)

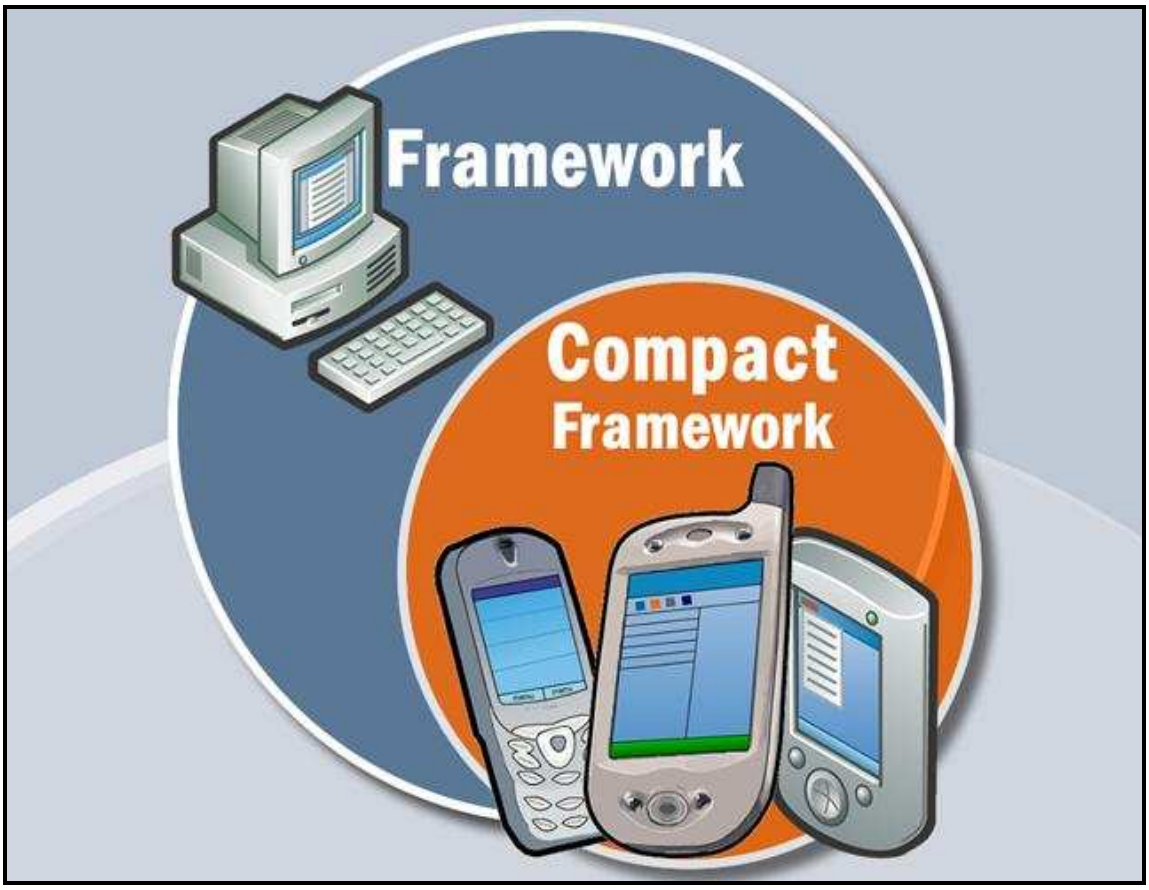
Figura 3: estrutura do .NET Framework

Como na plataforma Java, que possue uma versão mais compacta da máquina virtual para ser utilizada em dispositivos móveis, devido ao espaço que estes aparelhos possuem, a Microsoft também desenvolveu um *framework* específico para estes aparelhos, denominado .NET Compact Framework

3.4 .NET COMPACT FRAMEWORK

O .NET *Compact Framework* (Figura 4) representa um subconjunto do .NET *Framework* que introduz a plataforma .NET no mundo dos dispositivos móveis. É um novo conjunto de APIs (Interface de Programação de Aplicações) totalmente orientadas a objetos que podem ser utilizadas em diversos tipos de dispositivos móveis com requerimentos

mínimos de hardware, onde a infra-estrutura da .NET é assegurada através de código gerenciado. Sendo que o .NET *Compact Framework* não possui todas as bibliotecas disponíveis no .NET *Framework*, pois precisa ser instalado em equipamentos com pouco espaço de armazenamento.



Fonte: Microsoft Corporation (2004)

Figura 4: .NET *Framework* com o subconjunto .NET Compact Framework

No .NET *Compact Framework* como no .NET Framework, o núcleo é composto pelo *Common Language Runtime* (CLR), que executa aplicações compostas pelo *Microsoft Intermediate Language* (MSIL) através do JIT (*Just-In-Time Compiler*), gerando o código nativo, de acordo com o dispositivo móvel onde a aplicação esta sendo utilizada.

Algumas características do .NET *Compact Framework* podem ser vistas na tabela 1.

Tabela 1 – Características do .NET *Compact Framework* (CF)

Característica	**Descrição**
Portabilidade entre diversos dispositivos	Os componentes *Native Support Libraries* (NSL) e *Platform Adaption Layer* (PAL) garantem a arquitetura de execução das aplicações em dispositivos diferentes, respeitando-se as limitações e recursos existentes em cada dispositivo, como dimensões de tela, recursos de sistema, opções de hardware, etc.
Suporte a tipos de dados nativos da Common Type System (CTS)	Boolean, Sbyte, Char, String, Int16, Uint16, Int32, Uint32, Int64, Uint64, Float, Double.
Class Loader and Verifier	Quando uma aplicação CF é carregada, o *Class Loader and Verifier* é responsável por carregar 'fisicamente' os arquivos e dependências da aplicação, bem como liberar sua utilização.
Just-In-Time (JIT) Compiler	Elemento chave para a boa performance da execução das aplicações CF. Executa tarefas de plena importância, pois determina instruções nativas de acordo com o *hardware* e recursos do dispositivo.
Garbage Collection (GC)	Responsável por 'recolher' alocações de memória já não utilizadas pela aplicação CF. Por mais que recursos de memória de aplicações móveis sejam normalmente inferiores se comparados com aplicações Desktop, o trabalho do GC na *Compact Framework* é ainda mais delicado (se não mais complicado) devido ao ambiente restrito ao qual ele deve administrar, que envolve velocidade na busca/avaliação/liberação de memória não utilizada.
Multithreading	O desenvolvedor continua tendo a oportunidade de processar (sempre que possível) tarefas em paralelo dentro das aplicações baseadas na CF.
Tratamento de Exceções	A CF oferece a mesma estrutura de tratamento de exceções existente na.NET *Framework* (alguns técnicos internacionais afirmam que este foi um dos maiores desafios para a equipe de engenheiros da Microsoft).

Fonte: Pekus Cons. e Desenvolvimento Ltda (2002).

Segundo a Microsoft Corporation (2003), o .NET *Compact Framework* só poderá ser instalado em dispositivos que rodem o Windows CE .NET ou sistema operacional similar que suporte .NET, como o *Windows Mobile for SmartPhone*.

Na tabela 2 encontram-se algumas das diferenças entre o .NET *Framework* e o .NET *Compact Framework* citadas pela Microsoft Corporation (2003).

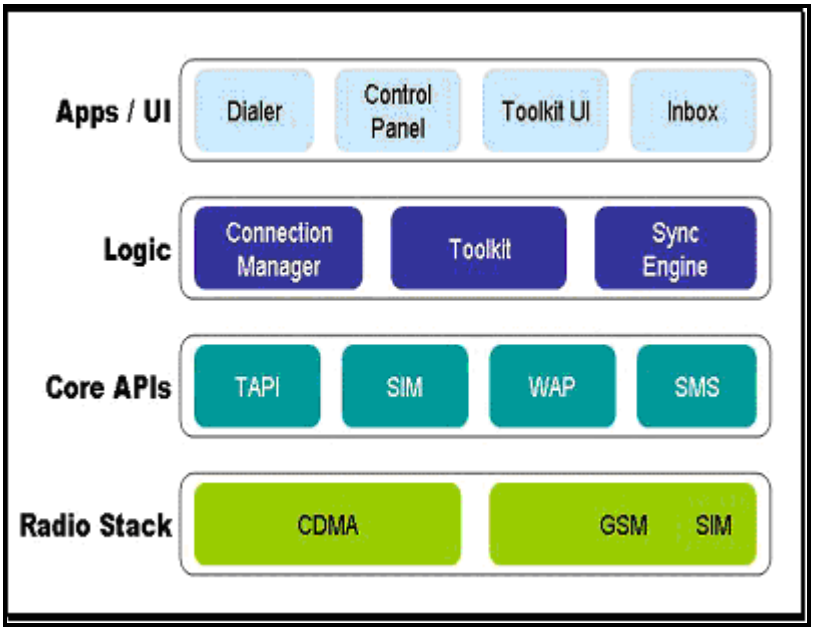
Tabela 2 – Diferenças entre .NET *Framework* e .NET *Compact Framework*

Característica	**Diferença**
Application Domains	O .NET *Compact Framework* não suporta o carregamento de assembly na área central para uso de múltiplos *Application Domains*.
ASP .NET	O .NET *Compact Framework* é destinado a aplicações consideradas residentes no device, e que não requerem conexão contínua a servidores centrais.
Biblioteca de Classes	O .NET *Compact Framework* suporta um subconjunto do .NET *Framework* apropriado para aplicações direcionadas para dispositivos com poucos recursos.
Exceptions	Foram feitas reduções em algumas informações obtidas através de objetos tipo *Exception* (para se poupar espaço), porém o desenvolvedor pode distribuir um assembly adicional para suprir algumas deficiências.
Segurança	Existem várias modificações em termos de segurança, incluindo a ausência de criptografia nativa no .NET *Compact Framework.*
Serialização	Não existe suporte para serialização usando *BinaryFormatter* e *SoapFormatter* (somente para transmissão de dados em SOAP para XML *WebServices*).
XML	Não há suporte para *schema validation* e *XPath Queries* em documentos XML.

3.5 WINDOWS PARA SMARTPHONE

A plataforma do SmartPhone é baseada no sistema Microsoft Windows CE 3.0, e possui muitas aplicações disponíveis em dispositivos *Pocket*, como e-mail, ferramentas de PIM e *Pocket Internet Explorer Web*. O sistema suporta os formatos HTML, WAP (WML) e XML. Conforme a Microsoft Corporation (2003), devido ao SmartPhone ser construído na plataforma Windows, os desenvolvedores de aplicação podem usar tecnologias baseadas em Windows e disponíveis no SmartPhone SDK, fornecendo um ambiente de desenvolvimento mais amplo e propiciando a criação de novos produtos ou serviços.

Como a arquitetura do Microsoft SmartPhone (Figura 5) é baseada no sistema operacional Windows CE 3.0, contém muitas das características e funções baseadas em Win32, incluindo Win32 APIs como TAPI e Winsock. A arquitetura fornece uma série de serviços que tornarão abstratos uma variedade de ligações subjacentes para serviços de voz e dados, onde as aplicações desenvolvidas não precisam saber qualquer coisa sobre as conexões subjacentes.



Fonte: Microsoft Corporation (2003)

Figura 5: Arquitetura do SmartPhone

Os componentes da arquitetura do SmartPhone são:

a) Applications/UI: este nível refere-se ao SmartPhone *shell* e aplicações cliente, como o *Pocket Internet Explorer*, o *Inbox*, o painel de controle, e o discador do telefone;

b) Logic: este nível contém a lógica da aplicação do sistema que pode ser usada pela camada de aplicação. Os exemplos deste incluem o controle de conexões de rede e de potencialidades da sincronização;

c) Core APIs: proporciona a interface entre os componentes da arquitetura de baixo nível (Sistema Operacional, software do SmartPhone e a *radio stack*) e as camadas de aplicação e lógica. Desenvolvendo as aplicações desejadas nesta camada ou acima, os programadores não necessitam saber os detalhes do nível mais baixo;

d) Radio Stack: refere-se em geral, aos componentes da arquitetura responsáveis pelo controle da voz, dados e a transmissão de dados.

Conforme a Microsoft Corporation (2003), as aplicações que originalmente foram desenvolvidas para funcionar com a tecnologia *Global System for Mobile* (GSM), não precisam sofrer alterações para trabalharem em uma rede *Code-Division Multiple Acess* (CDMA). Devido a esta flexibilidade, os desenvolvedores poderão utilizar-se de seus conhecimentos, para gerar softwares em uma larga escala de redes, com opções de plataforma e de configurações.

3.5.1 SMARTPHONE HARDWARE

O Microsoft SmartPhone combina características de telefone com as funções encontradas tipicamente em um *PDA*. Com uma conexão de dados *wireless*, um SmartPhone fornece uma ferramenta portátil de voz e transmissão de dados.

Segundo a Microsoft Corporation (2003), os estudos realizados sobre usabilidade indicam que os usuários esperam robustez no desempenho e simplicidade para operar o dispositivo. O usuário pode mover-se enquanto opera o telefone e pode-se utilizar o dispositivo em condições inadequadas para outros aparelhos, como em espaços aglomerados ou mal iluminados. Para diferenciar o produto, a Microsoft propôs um conjunto de setas para manipulação da navegação, discagem, e acesso a dados no SmartPhone.

Na Figura 6 mostra-se a sugestão de layout e controle do hardware para SmartPhones que é baseado na pesquisa de usabilidade. A colocação de *soft keys*, *home*, *back keys*, e *a five-way d-pad* são altamente recomendas.

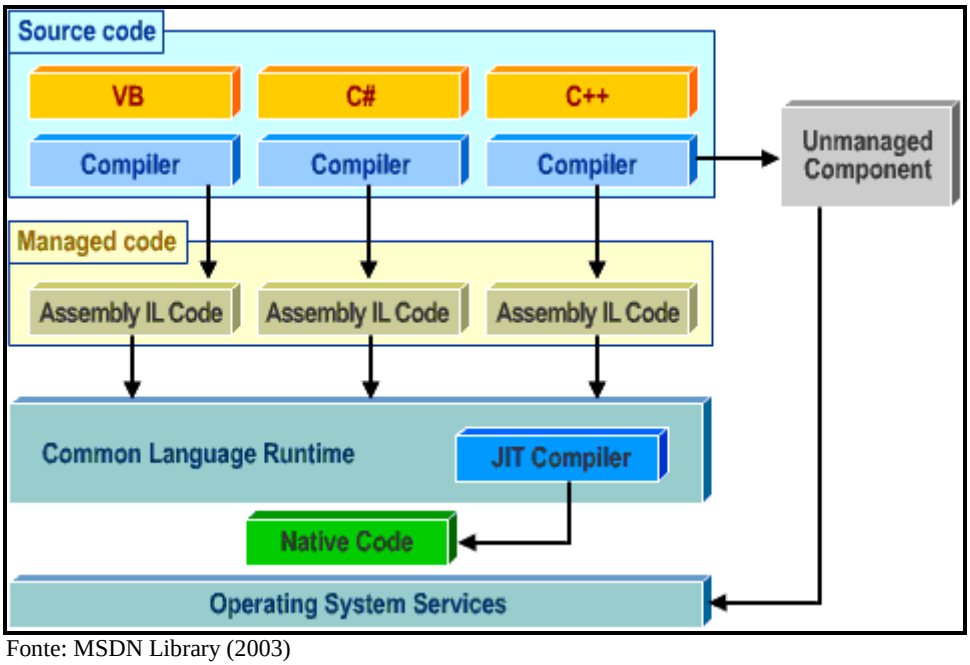
Fonte: Microsoft Corporation (2003)

Figura 6: Sugestão de layout para SmartPhone

O SmartPhone possui uma tela de LCD na orientação de retrato, sendo que sua resolução padrão é de 0.20 mm *dot pitch* em uma definição de 176x220, podendo ser em escala de cinza ou colorido (MICROSOFT, 2003). O display é projetado para a navegação *one-handed* utilizando-se para isso a *five-way d-pad*, que é melhor para o usuário que não precisa ficar tocando na tela. A pesquisa realizada pela Microsoft (2003) indica que o usuário prefere este tipo de navegação por diversas razões, incluindo o de poder operar o telefone com uma mão, o consumo de potência mais baixo e uma maior durabilidade devido ao não toque na tela.

3.6 CLR

O componente CLR (Figura 7) é a parte mais importante do .NET *Framework*, pois é responsável pela execução do código MSIL. O CLR fornece uma execução segura, isto é, realiza o "gerenciamento de segurança e de memória, segurança do tipo e interoperabilidade de interlinguagem. Um código não gerenciado pode gravar em áreas da memória que ele não possui, executar instruções em locais arbitrários da memória e apresentar qualquer outro mal comportamento que não possa ser gerenciado ou evitado pelo CLR" (TURTSCHI et al, 2002, p. 14).



Fonte: MSDN Library (2003)

Figura 7: estrutura do CLR.

O CLR utiliza a tecnologia de compilação *Just In Time* (JIT) para transformar o código MSIL no código nativo do aparelho, sendo que possui três tipos de JIT:

a) Pré-JIT: o código MSIL é compilado para o código nativo de uma única vez;
b) Econo-JIT: o código MSIL é compilado bit a bit para o código nativo, liberando recursos da memória *cache* quando necessário;
c) Normal-JIT: é o JIT padrão do .NET. O código MSIL é compilado para o código nativo somente quando for utilizado e o resultado armazena na memória *cache*.

4 SEGURANÇA

No estágio atual em que se encontram os negócios nas empresas, sua multiplicidade, com uma significativa interação entre os sistemas, a segurança de informações não pode mais ser pensada como um conceito teórico e sim como um fato presente no cotidiano.

A segurança no ambiente corporativo deve ser vista como proteção ao patrimônio da empresa e aos investimentos realizados em software, pessoal e equipamentos. As informações sobre a empresa podem ter alto valor no mercado, por isso devem ser protegidas de espiões. Para que esta proteção seja efetiva e permanente, deve-se desenvolver uma política de segurança.

De acordo com Laudon e Laudon (1999, p.270), é fundamental para política de segurança de dados restringir o acesso em termos de “precisar conhecer”, ou seja, permitir que as pessoas tenham acesso somente ao tipo de dados que elas precisam para as suas funções.

Os sistemas de informação adquiriram extrema importância para a sobrevivência das organizações modernas, já que sem computadores e redes de comunicação, a manipulação de dados e informações torna-se difícil comprometendo a condução dos negócios (DIAS, 2000, p. 50).

4.1 CRIPTOGRAFIA

A criptografia protege os dados da observação ou de serem modificados e fornece um canal seguro de comunicação. Por exemplo, os dados podem ser cifrados para serem transmitidos em um estado criptografado e mais tarde ser decriptografados pela parte autorizada. Se uma terceira parte interceptar os dados cifrados, terá grande dificuldade em decifrar os dados.

De acordo com O’Brien (2001), a criptografia de dados tornou-se uma maneira importante de proteger dados e outros recursos de redes de computadores. O uso da criptografia é imprescindível na comunicação de dados, pois protege informações sensíveis contra revelação, principalmente em transações comerciais.

A criptografia é usada para conseguir os seguintes objetivos:

a) Confidencialidade: para proteger uma identidade ou dados da leitura;
b) Integridade de dados: para proteger dado de ser alterado;
c) Autenticação: para assegurar que os dados originam do lugar correto.

Conforme Péricas (2003, p.138), existem dois tipos de algoritmos, de onde são feitas combinações para criar algoritmos mais complexos. Os tipos são:

a) criptografia por chave secreta ou criptografia simétrica: este tipo de encriptação usa uma única chave secreta para criptografar e decriptografar os dados;
b) criptografia por chave pública ou criptografia assimétrica: este tipo de encriptação usa um par de chaves publica/privada para cifrar e decifrar os dados.

4.1.1 CRIPTOGRAFIA POR CHAVE SECRETA

Os algoritmos de encriptação por chave secreta utilizam-se de uma mesma chave para criptografar e decriptografar os dados, ou seja, tanto o remetente quanto o destinatário usam a mesma chave, sendo que os sistemas de uma chave são mais rápidos que os sistemas de chave pública.

Segundo Lucchesi (1986, p. 50), quando utiliza-se a mesma chave para criptografar e decriptografar, a chave deve ser enviada para o usuário autorizado antes que as mensagens sejam enviadas. A chave é o "segredo", sendo de grande importância que a mesma não seja conhecida por outrem além do destinatário, a fim de manter a integridade dos dados.

Os algoritmos de chave secreta efetuam a criptografia por bloco. A encriptação efetuada por algoritmos como RC2, DES, TripleDES, e Rijndael, realiza-se através da transformação de um bloco de entrada de n bytes em um bloco da saída de bytes cifrados. Para cifrar ou decifrar uma seqüência de bytes, tem-se que fazê-la bloco por bloco, uma vez que o tamanho de n é pequeno (n = 8 bytes para RC2, DES; para TripleDES, n = 16 [*default*] ou n = 24; e n = 32 bytes para Rijndael), os valores maiores do que n tem que ser cifrados um bloco por vez (MICROSOFT CORPORATION, 2003).

Para realização deste trabalho utilizou-se o Algoritmo Blowfish.

4.1.2 BLOWFISH

Blowfish é um algoritmo de chave simétrica de domínio público, desenvolvido por Bruce Schneier em 1993. O algoritmo foi construído para ser uma alternativa grátis e eficiente de criptografia de dados, podendo ser utilizado tanto em ambientes domésticos, como em empresas (SCHNEIER, 2004).

Segundo Schneier (2004), o algoritmo possui as seguintes características:

a) cifragem em blocos de 64 bits;
b) chave de tamanho variável: 32 à 448 bits;
c) mais rápido que o Data Encryption Standard (DES) e o International Data Encryption Algorithm (IDEA);
d) não patenteado e totalmente grátis;
e) não necessita de licença.

Desde que foi desenvolvido, o mesmo vem sendo analisado e esta sendo considerado como um algoritmo forte. Foi realizado um teste de velocidade em computador Pentium 150 entre os algoritmos Blowfish, DES e IDEA, onde os resultados são demonstrados na tabela 3.

Tabela 3 - Comparações de velocidade de criptografia de dados em um Pentium

Algoritmo	***Clock cycles per round***	***# of rounds***	***# of clock cycles per byte encrypted***	**Notas**
Blowfish	9	16	18	não patenteado
DES	18	16	45	56-bit key
IDEA	50	8	50	patenteado pela Ascom-Systec

Fonte: Schneier (2004).

5 DESENVOLVIMENTO DO TRABALHO

O protótipo deste trabalho é formado por dois aplicativos autônomos e um *Web Service*, onde o objetivo é realizar o cadastro de mensagens em um banco de dados no *desktop*, para futuramente serem carregadas em um dispositivo móvel (SmartPhone) através do *Web Service.*

Um protótipo será executado no *desktop* e tem a função de cadastrar as mensagens para os respectivos usuários finais da informação.

O aplicativo do SmartPhone será executado através do emulador SmartPhone SDK da Microsoft que emula o ambiente do Windows para SmartPhone e suporta a plataforma .NET. Este protótipo tem por finalidade se conectar ao *Web Service* e solicitar o envio das mensagens ao dispositivo, sendo que as mesmas se encontrarão criptografadas. Quando as mensagens estiverem no aparelho, elas serão decriptografadas pelo programa através de uma chave secreta.

O *Web Service* realizará a conexão entre o banco de dados e o SmartPhone. Antes de enviar as mensagens, o *Web Service* irá criptografar com a chave secreta do usuário.

5.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Devido ao protótipo ser dividido em aplicativos independentes, os requisitos funcionais (RF) e requisitos não funcionais (RNF) foram descritos para cada um.

Os requisitos do aplicativo do SmartPhone são:

a) verificar se o colaborador está cadastrado na empresa através do *Web Service* (RF);
b) receber as mensagens enviadas pelo *Web Service* (RF);
c) decriptografar as mensagens (RF);
d) mostrar para o usuário as mensagens decriptografadas (RF);
e) mostrar para o usuário as mensagens criptografadas, ou seja, visualizar as mensagens conforme foram enviadas pelo *Web Service* (RNF);
f) permitir o cadastramento da chave secreta (RF);
g) rodar o sistema em qualquer celular que suporte a tecnologia .NET. (RNF);

Os requisitos do software do *desktop* são:

a) permitir o cadastramento do colaborador com a sua devida chave secreta (RF);
b) permitir o cadastramento do usuário do sistema (RF);
c) mostrar as mensagens cadastradas (RF);
d) rodar o sistema em qualquer desktop que suporte a tecnologia .NET (RNF);
e) permitir a inclusão das mensagens (RF).

O requisito do *Web Service* é criptografar as mensagens enviadas para o dispositivo móvel (RF);

5.2 ESPECIFICAÇÃO

Para realizar a especificação do aplicativo utilizou-se da análise estruturada. Como os softwares que formam o protótipo são independentes, a especificação dos aplicativos foi realizada separadamente.

5.2.1 LISTA DE EVENTOS

A Lista de Eventos foi separada em duas tabelas uma para o software do *desktop* e a outra para o aplicativo contido no dispositivo móvel, aonde pode ser identificado qual sistema deve responder e uma indicação da pessoa ou sistema que inicia o evento.

Tabela 4 – Lista de Eventos do software do desktop

Nº	**Nome do Evento**
1	Usuário é cadastrado
2	Colaborador é cadastrado
3	Mensagem é cadastrada para o colaborador
4	Sistema mostra mensagens

Tabela 5 – Lista de Eventos do software do dispositivo móvel

Nº	**Nome do Evento**
1	Chave secreta é cadastrada
2	Colaborador é validado
3	Mensagem recebida do Web Service
4	É momento de decriptografar a mensagem
5	É momento de mostrar a mensagem ao colaborador

5.2.2 DIAGRAMA DE CONTEXTO

Nas Figuras 8 e 9, são mostrados os diagramas de contexto para os aplicativos existentes, sendo que através do diagrama é possível visualizar quais são relacionamentos

existentes com entidades externas. Foi utilizada a ferramenta Microsoft Visio 2003, para realizar a construção dos diagramas.

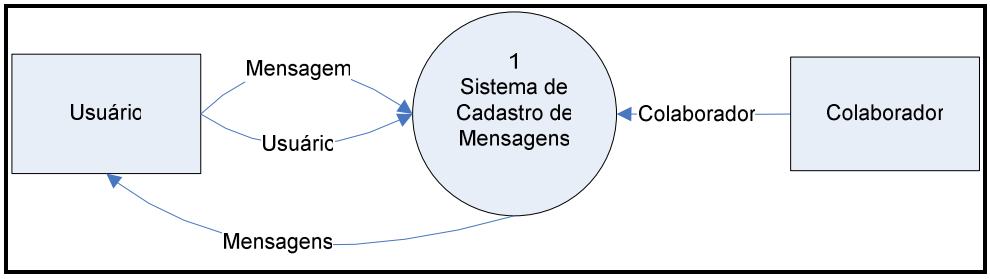


Figura 8: Diagrama de contexto do software do desktop

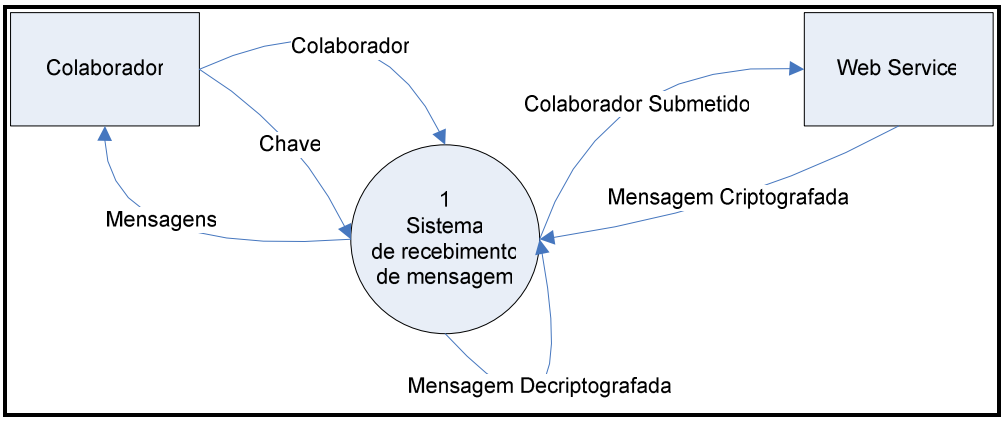


Figura 9: Diagrama de contexto do software do dispositivo móvel

5.2.3 DFD

As Figuras 10 e 11 representam através do diagrama de fluxo de dados, as principais funcionalidades de cada aplicativo.

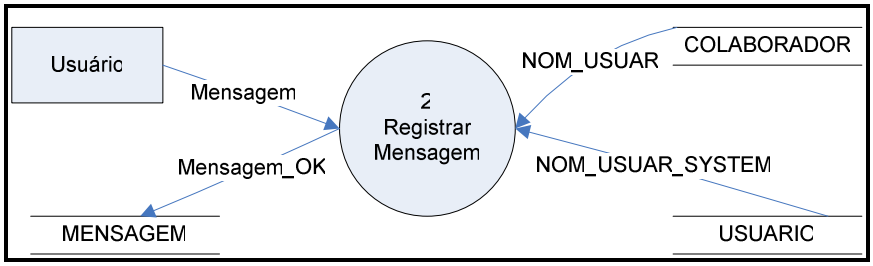


Figura 10: Diagrama de Fluxo de Dados do aplicativo do desktop

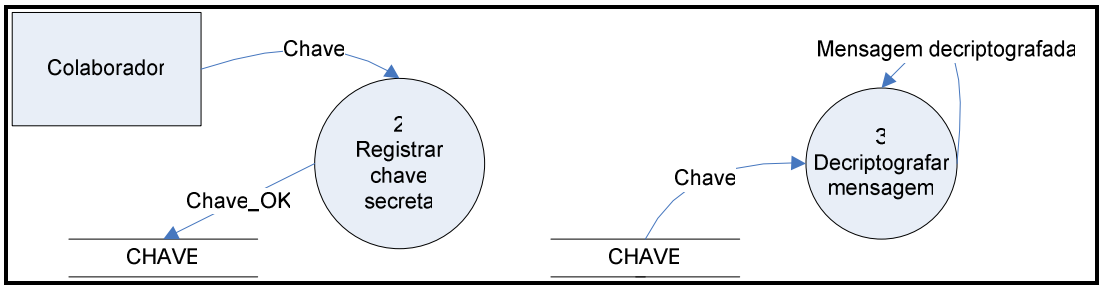


Figura 11: Diagrama de Fluxo de Dados do aplicativo do dispositivo móvel

5.2.4 MER

O MER (Figura 12) apresenta as entidades que fazem parte do protótipo, e proporcionar um entendimento mais completo do funcionamento geral.

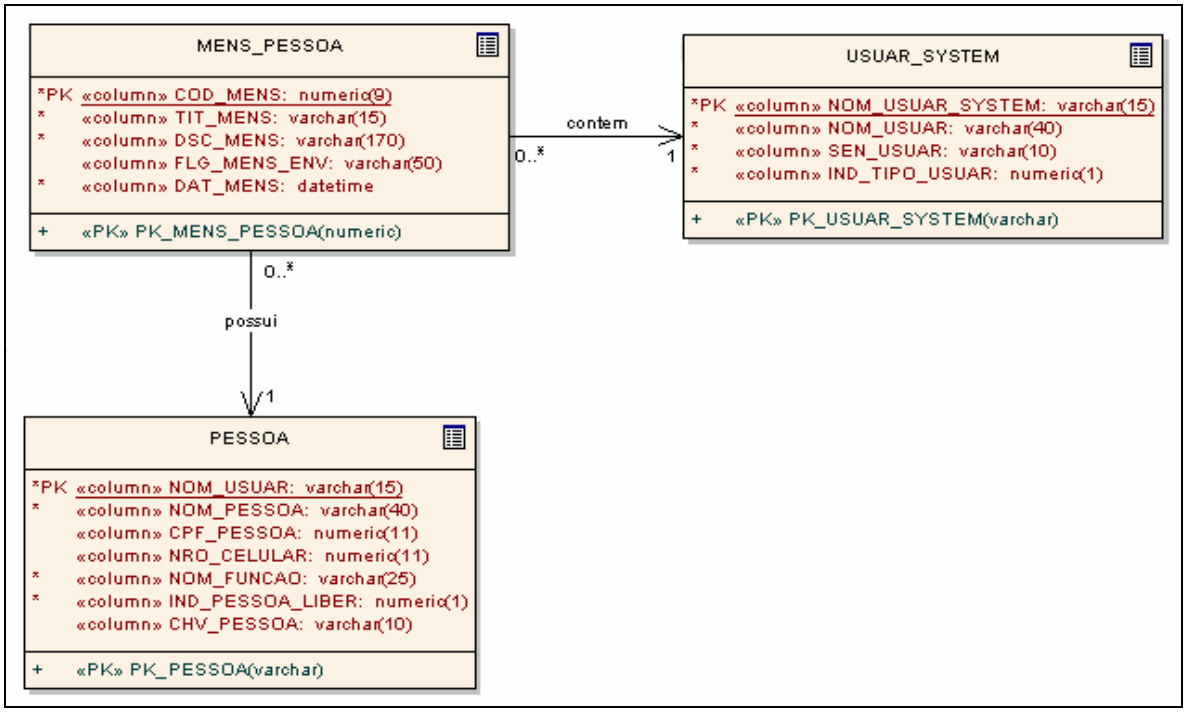


Figura 12: Modelo Entidade Relacionamento Lógico

A Figura 12 apresenta as tabelas utilizadas para armazenar os dados do aplicativo localizado no *desktop*. A tabela MENS_PESSOA contém as mensagens cadastradas para o colaborador, onde a mensagem deve conter um colaborador (tabela PESSOA) e um usuário de sistema (tabela USUAR_SYSTEM).

5.3 IMPLEMENTAÇÃO

Nas próximas seções serão abordadas as ferramentas utilizadas na implementação dos protótipos, e o funcionamento do aplicativo.

5.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

O desenvolvimento do trabalho foi realizado através da linguagem Visual Basic .NET, que foi empregada na construção do aplicativo contido no dispositivo móvel, *desktop* e *Web Service*. O ambiente empregado para desenvolver os protótipos foi o Visual Studio 2003 (Figura 13) da Microsoft, sendo que para implementação do aplicativo para o dispositivo móvel foi necessário ainda o .NET *Compact Framework* (máquina virtual para aparelhos móveis), a biblioteca Microsoft SMARTPHONE 2003 SDK e o Windows *Mobile* 2003 *Second Edition Emulator Images for SmartPhone* (Figura 14) que contém as imagens para emular o aplicativo.

O SGBD utilizado foi o SQL Server 2000 e para manipular os dados utilizou-se o *Enterprise Manager*, ferramenta contida no pacote de instalação do SGBD.

Como o .NET *Compact Framework* não possui criptografia nativa, foi adaptado o código escrito por Acheson (2003) na linguagem Visual Basic .NET para o .NET *Framework*, afim de que o mesmo pudesse ser utilizado em aparelhos móveis. O código implementa o algoritmo de criptografia por chave secreta BlowFish.

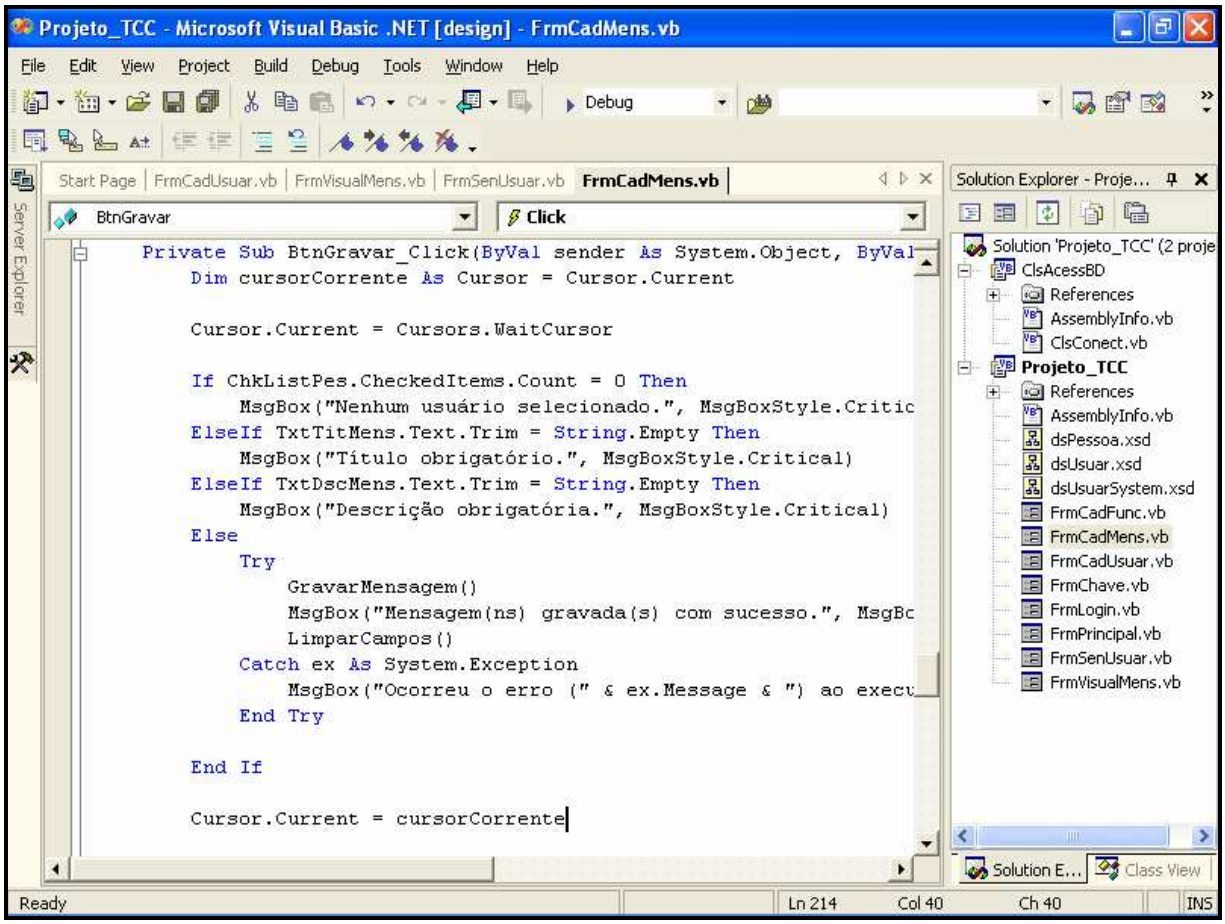


Figura 13: Microsoft Visual Studio 2003

Figura 14: Emulador para SmartPhone da Microsoft

Em seguida são mostradas as principais funções e procedures desenvolvidas nos protótipos e no *Web Service*.

Através do Quadro 1 pode-se visualizar a função MontarCheckList do protótipo do *desktop*.

```
  Private Sub MontarCheckList()
    Dim clsBD As New ClsAcessBD.ClsBD  'Classe de Conexão com o banco
    Dim pesDataSet As DataSet = New DataSet
    Dim pesDataRow As DataRow
    Dim querySQL As String

    querySQL = "Select nom_usuar, nom_pessoa, chv_usuar "
    querySQL += "From  Pessoa "
    querySQL += "where IND_USUAR_LIBER = 1"  'Usuário Liberado

    'Retorna usuários liberados para o envio de mensagem
    pesDataSet = clsBD.RetornarDataSet(querySQL, "PESSOA")

    For Each pesDataRow In pesDataSet.Tables("PESSOA").Rows
      ChkListPes.Items.Add((pesDataRow("NOM_USUAR").ToString() & _
                    " - " & pesDataRow("NOM_PESSOA").ToString()))
    Next

  End Sub
```

Quadro1: Procedure MontarCheckList do aplicativo do PC

A função MontarCheckList é responsável por montar a lista de colaboradores liberados para o envio de mensagem, com isso se evita que mensagens sejam cadastradas sem necessidade, isto é, somente será incluída uma mensagem se o usuário estiver apto a recebê-la.

A seguir é mostrado, no Quadro 2, a função RetornarMensagem construída no *Web Service*, que tem como objetivo retornar as mensagens criptografadas para o colaborador especificado através do parâmetro “username” e atualizar a flag “flg_mens_env”, a fim de informar ao sistema que a mensagem foi enviada ao usuário.

```
    <WebMethod(Description:="Retorna as mensagens criptografadas para o colaborador
")> _
    Public Function RetornarMensagem(ByVal username As String) As DataSet

      Dim clsBD As New ClsAcessBD.ClsBD
      Dim daTCC As SqlDataAdapter = New SqlDataAdapter
      Dim dsTCC As DataSet = New DataSet
      Dim drTCC As DataRow
      Dim querySQL As String
      Dim strCripto As String

      querySQL = "Select p.nom_pessoa, mp.cod_mens, mp.tit_mens, mp.dsc_mens,
p.chv_usuar"
      querySQL += " from Mens_Pessoa mp, Pessoa p"
      querySQL += " where p.nom_usuar = mp.nom_usuar"
      querySQL += " and   mp.nom_usuar = '" & username.ToUpper.Trim() & "'"
      querySQL += " and   mp.flg_mens_env is null"

      dsTCC = clsBD.RetornarDataSet(querySQL, "TCC")

      For Each drTCC In dsTCC.Tables("TCC").Rows
        strCripto = CriptoMensagem(drTCC("TIT_MENS").ToString(),
drTCC("CHV_USUAR").ToString())
        drTCC("TIT_MENS") = strCripto
        strCripto = CriptoMensagem(drTCC("DSC_MENS").ToString(),
drTCC("CHV_USUAR").ToString())
        drTCC("DSC_MENS") = strCripto
      Next

      dsTCC.AcceptChanges()

      querySQL = "Update Mens_Pessoa set "
      querySQL += " flg_mens_env = 'S'"
      querySQL += " where nom_usuar = '" & username.ToUpper.Trim() & "'"
      querySQL += " and   flg_mens_env is null"

      clsBD.ExecutarSQL(querySQL)

      Return dsTCC
    End Function
```

Quadro2: Função RetornarMensagem do Web Service

A função CriptoMensagem, mostrada no Quadro 3, serve para criptografar a mensagem.

```
   Private Function CriptoMensagem(ByVal mensagem As String, ByVal chave As String)
As String

     Dim aKey() As Byte
     Dim strCripto As String ' Usado para armazenar a msg criptografada

     aKey = cv_BytesFromString(chave)    ' Transforma a chave em bytes
     blf_KeyInit(aKey)                   ' Inicializa a chave criptografica
     strCripto = blf_StringEnc(mensagem) ' Criptografa a mensagem

     Return strCripto                    ' Retorna a mensagem Criptografa

   End Function
```

Quadro3: Função CriptoMensagem do *Web Service*

A função DataSetDecripto implementada no protótipo do aparelho móvel, tem por finalidade solicitar as mensagens para o *Web Service*, em seguida chama a função para decriptografar as mensagens, sendo que por último move os dados para a tela. A função pode ser vistao no Quadro 4.

```
  Private Sub DataSetDecripto()
    Try
      Dim strChave As String

      strChave = RetornarChave()
      dsMensCripto = wsTCC.RetornarMensagem(nomUsuar)
      dsMensDecripto = wsTCC.RetornarMensagem(nomUsuar)

      For Each drMensagem In dsMensDecripto.Tables("TCC").Rows
        LblNomPessoa.Text = drMensagem("NOM_PESSOA").ToString()
        drMensagem("TIT_MENS") =
DecriptoMensagem(drMensagem("TIT_MENS").ToString(), strChave.Trim())
        drMensagem("DSC_MENS") =
DecriptoMensagem(drMensagem("DSC_MENS").ToString(), strChave.Trim())
      Next

      dsMensDecripto.AcceptChanges()
      nomPessoa = LblNomPessoa.Text

    Catch ex As System.Exception
      MsgBox("Ocorreu o Erro: " & ex.Message, MsgBoxStyle.Critical)
    End Try

  End Sub
```

Quadro4: Função DataSetDecripto do aplicativo do dispositivo móvel

A função DecriptoMensagem, mostrada no Quadro 5, realiza a decodificação da mensagem através da chave secreta.

```
  Private Function DecriptoMensagem(ByVal mensagem As String, ByVal chave As
String) As String
    Dim aKey() As Byte
    Dim strDecripto As String ' Used to store ciphertext

      aKey = cv_BytesFromString(chave)
      blf_KeyInit(aKey)
      strDecripto = blf_StringDec(mensagem)

      Return strDecripto

  End Function
```

Quadro5: Função DecriptoMensagem do aplicativo do dispositivo móvel

5.3.2 OPERACIONALIDADE DA IMPLEMENTAÇÃO

Neste tópico é visto a operacionalização do protótipo, começando pelo software instalado no *desktop*.

O aplicativo possui dois tipos de usuários que são:

a) usuário avançado: possui permissão de administrador, isto é, pode efetuar inclusão, exclusão e alteração de colaboradores, usuários do sistema, senhas para serem utilizadas na criptografia e cadastrar mensagens para os colaboradores;
b) usuário padrão: possui permissão para cadastrar colaboradores e mensagens, mas não podem incluir, alterar ou excluir a chave de criptografia do mesmo e pode somente consultar os usuários do sistema que se encontram cadastrados.

Ao iniciar o software é exigido o *username* e a senha do usuário. Após os dados serem validados o sistema abre a tela principal (Figura 15) onde são mostradas as mensagens que já encontram-se cadastradas. Quando novas mensagens são incluídas o *grid* é automaticamente atualizado.

Figura 15: Tela principal do sistema

Como as mensagens podem ter até 170 caracteres, a mensagem não aparece por inteira no *grid*, mas com um dublo *click* na coluna em azul do lado esquerdo, abre-se uma tela com os dados completos da mensagem (Figura 16).

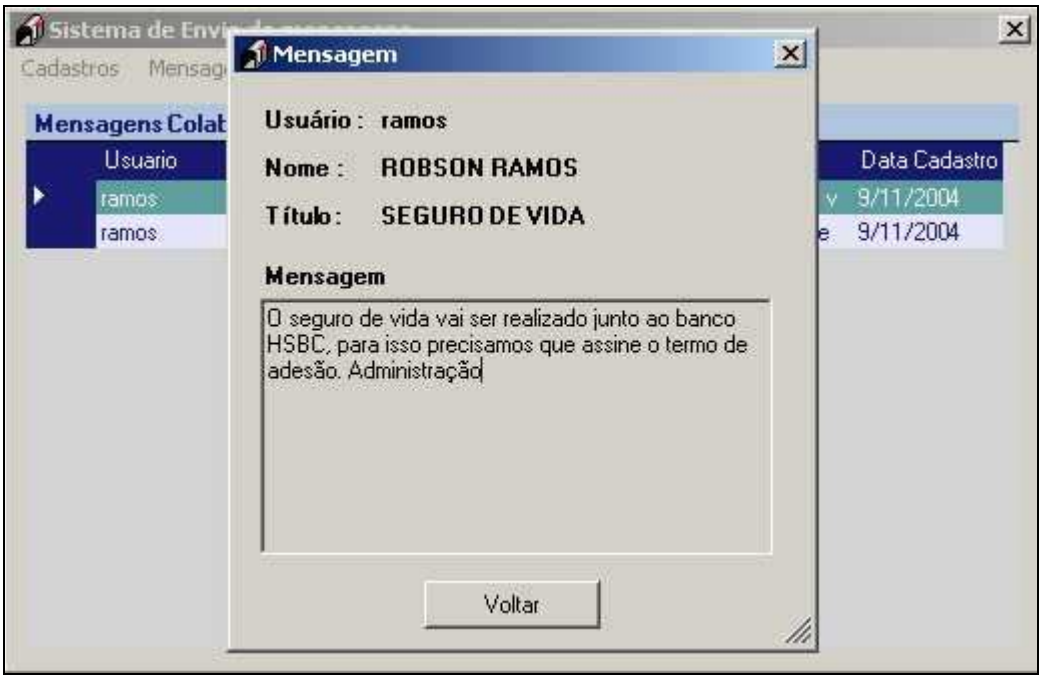


Figura 16: Tela com os dados da mensagem

Através do menu é possível acessar as telas do sistema que são:

a) cadastro de colaboradores;
b) cadastro de usuários do sistema;
c) cadastro de mensagens.

Na tela de cadastro de colaboradores o usuário padrão não tem acesso aos campos relacionados à chave de criptografia do colaborador, conforme demonstrado na Figura 17.

Figura 17: Tela com os dados da mensagem

A seguir é apresentada a tela mais importante do aplicativo que é a de cadastro de mensagens (Figura 18). Ao carregar-se, preenche a lista dos usuários aptos a recebem mensagens automaticamente, com isso evita-se de cadastrar mensagens a colaboradores que não estejam preparados para receber as informações. Na tela deve-se digitar o título e a descrição da mensagem e marcar os colaboradores que devem receber a informação.

Após os dados serem salvos, a tela retorna ao estado inicial de abertura para que o usuário possa cadastrar novas mensagens.

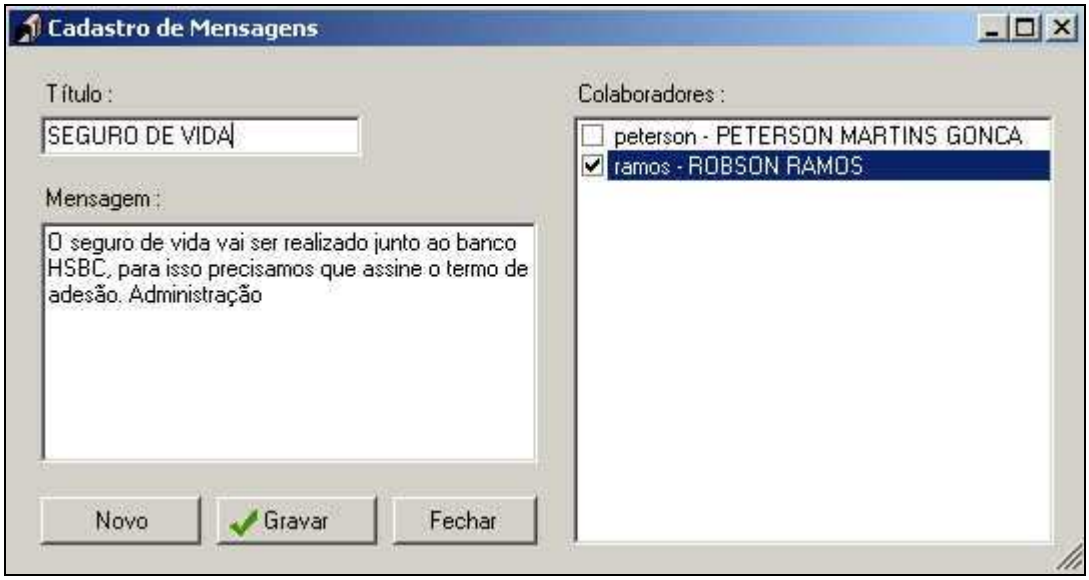


Figura 18: Tela com os dados da mensagem

O software instalado no dispositivo móvel verifica se existe mensagem cadastrada para o usuário através de uma conexão com o *Web Service*. Caso exista mensagem para o colaborador, o *Web Service* criptografa a mensagem e envia para o aparelho móvel, onde o software decriptografa a mesma, e exibe para o colaborador a mensagem original, o que torna o processo transparente para o usuário final.

Mas para que a decriptografia funcione é necessário que a chave secreta já esteja instalada no dispositivo, pois é através dela que o sistema consegue voltar o texto criptografado para o original. Devido a está necessidade, o sistema, ao ser ativado, verifica se a chave está cadastrada. Caso esteja é aberta à tela de *login* (Figura 19), senão, abre a tela para cadastro da chave (Figura 20) e se a mesma não for digitada o sistema é encerrado.

Após a digitação da chave, o sistema solicita o *username* do colaborador para efetuar a validação e retornar as mensagens cadastradas para o mesmo. Se não existir nenhuma mensagem o sistema emite um aviso.

Figura 19: Tela de *login*

Figura 20: Tela de cadastro da chave secreta

A tela a seguir mostra as mensagens decifradas (Figura 21) e o nome do colaborador, para visualizar as outras mensagens é necessário somente pressionar a seta para a esquerda ou direita. A partir desta tela é possível ir para a tela de cadastro da chave ou visualizar as mensagens antes de serem decriptografadas, ou seja, as mensagens aparecem criptografadas como pode ser visto na Figura 22.

Figura 21: Tela para visualizar as mensagens decriptografadas

Figura 22: Tela para visualizar as mensagens criptografadas

Após o aplicativo ser fechado, as mensagens são apagadas do dispositivo móvel e se o colaborador desejar verificar se existem novas mensagens o sistema deve ser encerrado e executado novamente.

5.4 RESULTADOS E DISCUSSÃO

Este trabalho atingiu seu objetivo de realizar a transmissão de mensagens de um *desktop* para um dispositivo móvel de modo sigiloso, onde a mesma é obtida através de um software implementado em Visual Baisc .NET rodando na plataforma .NET.

Primeiramente tinha-se a intenção de utilizar o SMS como meio para enviar a mensagem do PC ao SmartPhone, mas devido ao emulador não realizar está operação de recebimento como realmente acontece na prática optou-se pelo *Web Service*. Através desta opção é possível emular o funcionamento de todo o processo do trabalho, desde o cadastro da mensagem até o recebimento da mensagem no aparelho móvel, e apesar do meio de envio ter sido alterado, o objetivo do trabalho que é enviar informação criptografada para o dispositivo móvel não foi alterado.

A interface do aplicativo do SmartPhone foi desenvolvida o mais simples possível, pois como se trata de um celular, a digitação é limitada uma vez que possui somente 12 teclas. Mesmo com esta limitação, a interface ficou de fácil manuseio para o usuário.

Na realização dos testes verificou-se que as mensagens foram incluídas corretamente no banco de dados. No momento que a mensagem é solicitada pelo aparelho móvel, a mesma é criptografada usando o algoritmo de chave secreta e ao chegar ao seu destino, é decriptografada voltando assim ao seu estado inicial. Notou-se que a resposta do *Web Service* para primeira solicitação do aparelho, demorou em torno de 5 a 10 segundos, enquanto nas requisições posteriores, a resposta foi quase imediata.

Verificou-se também que caso a chave não seja a mesma que está cadastrada no PC e no celular, a mensagem não é decriptografada, sendo que aparece dados inconsistentes para o colaborador.

Na tabela 6, pode ser visualizado algumas comparações feitas entre este trabalho e os trabalhos correlatos.

Tabela 6 – Comparação entre o trabalho e os trabalhos correlatos

Acadêmicos / **Itens**	**Este Trabalho**	**Santos (2003)**	**Depiné (2002)**	**Schaefer (2004)**
Plataforma de desenvolvimento	.NET	.NET	JAVA	JAVA
Linguagem de programação	Visual Basic .NET	C#	JAVA (J2ME)	JAVA(J2ME)
Dispositivo Móvel	SmartPhone	PDA	Celular convencional	Celular convencional
Acesso a Web	Sim	Sim	Não	Sim
Criptografia	Sim	Não	Não	Não
Objetivo principal do trabalho	Segurança na transmissão de mensagens para dispositivos móveis através de criptografia	Visualizar notícias da *Web* através de um PDA	Realizar cálculos de tempo e deslocamento necessários em um Rally	Coletar Informações e envia-lás pra um *desktop* para futura análise dos dados

6 CONCLUSÕES

Através deste trabalho foi possível ter uma visão sobre a atual tendência do mercado em relação aos dispositivos móveis, bem como o crescente número de aplicações que estão surgindo para estes aparelhos, devido ao seu baixo custo e alta portabilidade em relação a outros equipamentos de informática como o notebook.

Também realizou-se um estudo sobre a plataforma .NET, onde foi descrita a sua estrutura, característica e portabilidade, sendo que a mesma oferece vários recursos para o desenvolvimento de aplicativos destinados a aparelhos móveis. Outro aspecto importante no Microsoft .NET é sua independência de plataforma, ou seja, o programa pode ser construído na linguagem que o desenvolvedor tem maior conhecimento e o mesmo rodará em qualquer ambiente que possua o .NET *Framework* instalado.

Uma dificuldade encontrada durante o desenvolvimento do protótipo foi a falta de criptografia nativa no .NET *Compact Framework*. Para resolver este tipo de problema foi necessário implementar um algoritmo de criptografia sem utilizar as classes *Security* do .NET *Framework*. Neste trabalho foi utilizado o código aberto escrito por Acheson (2003) para a linguagem Visual Basic .NET, sendo que o mesmo foi adaptado para trabalhar no .NET *Compact Framework*.

Outro problema encontrado foi o alto custo do SmartPhone, onde inviabilizou que o protótipo fosse executado em um ambiente real. Devido a isto, o mesmo somente foi testado e executado no emulador da Microsoft.

Este trabalho mostrou como é possível integrar as novas tecnologias para efetuar a proteção de dados de uma empresa, sendo que durante todo o processo de desenvolvimento, foram adquiridos novos conhecimentos sobre a tecnologia utilizada para realização deste trabalho.

6.1 EXTENSÕES

Como extensões deste trabalho sugerem-se:

a) maior iteração entre o usuário e a empresa, isto é, possibilitar o envio de informações do dispositivo móvel para a empresa funcionando como um coletor de dados, onde as mesmas seriam atualizadas no banco de dados em tempo real, com

envio de arquivos e atualização automática da chave de encriptação;

b) desenvolver o protótipo utilizando criptografia por chave pública através de um mecanismo de autenticação.

REFERÊNCIAS BIBLIOGRÁFICAS

ACHESON, Todd**. Cryptography software code in Visual Basic and C**. [S.l.], [2003]. Disponível em: <http://www.di-mgt.com.au/crypto.html>. Acesso em: 24 out. 2004.

DEITEL, H. M.; DEITEL, P. J.; NIETO, T. R. **Visual Basic .NET**: como programar. Tradução Célia Yumi Okano Taniwaki. São Paulo: Pearson Education do Brasil, 2004. 1088 p.

DEPINÉ, Fabio Marcelo. **Protótipo de software para dispositivos móveis utilizando Java me para cálculo de regularidade em rally**. 2002. 55 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

DIAS, Claudia. **Segurança e auditoria da tecnologia da informação**. Rio de Janeiro: Axcell Books, 2000. 218 p.

DOCTORSYS Tecnologia da Informação. **Dispositivos móveis.** São Paulo, 2004. Disponível em: <http://www.doctorsys.com.br/dispmoveis.asp>. Acesso em: 10 out. 2004.

DORMAN, Andy. **Wireless communication**: o guia essencial de comunicação sem fio. Tradução Fábio Freitas. Rio de Janeiro: Campus, 2001. 304 p.

FORTES, Débora. O PDA vai para a rua. **Info Exame**, São Paulo, n. 216, p. 65-69, mar. 2004.

GUIMARÃES, Camila. **As empresas detectam vantagens no celular que é também computador**, [s.l.], 2004. Disponível em: <http://www.netmarkt.com.br/noticia2004/2306.html>. Acesso em: 30 ago. 2004

INFOEXAME**. Visa venderá ingressos no smartphone MPx220**, São Paulo, 2004. Disponível em: <http://www.infoexame.com.br>. Acesso em: 28 set. 2004.

LAUDON, Kenneth C.; LAUDON, Jane Price. **Sistemas de informação.** Tradução Dalton Conde de Alencar. Rio de Janeiro: LTC, 1999.

LUCCHESI, Cláudio Leonardo. **Introdução à criptografia computacional.** Campinas: UNICAMP/Papirus, 1986. 132 p.

MIRANDA, Luiz Henrique. **Introdução ao mundo móvel.** Goiás, [2004]. Disponível em: <http://www.devgoiania.net/artigos.aspx>. Acesso em: 20 set. 2004.

MICROSOFT CORPORATION. **Microsoft .NET.** [S.l.], [2003]. Página oficial do produto. Disponível em: <http://www.microsoft.com/brasil/dotnet/default.asp>. Acesso em: 30 mar. 2004.

MICROSOFT CORPORATION. **PDC2004.** [S.l.], [2004]. Professional Developers' Conference 2004. Disponível em: <http://www.microsoft.com/brasil/msdn/Eventos/pdc2004>. Acesso em: 15 set. 2004.

MSDN LIBRARY. **Microsoft MSDN**. [S.l.], [2004]. Página de ajuda da Microsoft Corporation. Disponível em: <http://msdn.microsoft.com/library/default.asp>. Acesso em: 25 mar. 2004.

O'BRIEN, J.A. **Sistemas de informação e as decisões gerenciais na era da internet**. São Paulo: Saraiva, 2001.

PEKUS CONS. E DESENVOLVIMENTO LTDA. **Dispositivos móveis.** São Paulo, 2002. Disponível em: <http://www.pekus.com.br/palmtops.htm>. Acesso em: 18 ago. 2004.

PÉRICAS, Francisco Adell. **Redes de computadores:** conceitos e arquitetura internet. Blumenau: ediFURB, 2003. 158 p.

SANTOS, Michael Schuenck dos. **Utilização de web services na plataforma .NET para a criação de um aplicativo visualizador de notícias para dispositivos móveis**. 2003. 89 f. Trabalho de Conclusão de Curso (Sistemas de Informação) - Centro de Ciências Exatas e Naturais, Centro Universitário Luterano de Palmas, Palmas.

SCHAEFER, Carine. **Protótipo de aplicativo para transmissão de dados a partir de dispositivos móveis aplicado a uma empresa de transportes**. 2004. 52 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SCHNEIER, Bruce. **Blowfish.** [S.l.], [2004]. Disponível em: <http://www.schneier.com/blowfish.html>. Acesso em: 07 dez. 2004.

TELECOMWEB. **Mercado de dispositivos móveis cresce 45% em 12 meses.** [S.l.], 2004. Disponível em: <http://www.telecomweb.com.br>. Acesso em: 27 set. 2004.

TURTSCHI, Adrian et al. **C# .NET**, guia do desenvolvedor Web. Tradução Marcos Vieira. Rio de Janeiro: Alta Books, 2002. 741 p.