

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

**SISTEMA DE TRANSFERÊNCIA DE ARQUIVOS PARA
DISPOSITIVOS MÓVEIS BASEADOS EM WEB SERVICES**

FERNANDO MICHELS

BLUMENAU
2010

2010/1-12

FERNANDO MICHELS

**SISTEMA DE TRANSFERÊNCIA DE ARQUIVOS PARA
DISPOSITIVOS MÓVEIS BASEADOS EM WEB SERVICES**

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
Ciências da Computação — Bacharelado.

Prof. Francisco Adell Péricas – Orientador.

**BLUMENAU
2010**

2010/1-12

SISTEMA DE TRANSFERÊNCIA DE ARQUIVOS PARA DISPOSITIVOS MÓVEIS BASEADO EM WEB SERVICES

Por

FERNANDO MICHELS

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, Mestre – Orientador – FURB

Membro: Prof. Paulo Fernando da Silva, Mestre – FURB

Membro: Prof. Sérgio Stringari, Mestre – FURB

Blumenau, 05 de Julho de 2010.

Dedico este trabalho a todos que me ajudaram e de alguma forma colaboraram para a conclusão deste trabalho. De forma especial a Deus, pois sem ele nada conseguiria.

AGRADECIMENTOS

A Deus, pelo seu imenso amor e graça.

À minha esposa, pelo seu amor que sempre me apoiou.

Ao meu Orientador, Francisco Adell Péricas, por ter me ajudado e colaborado com este trabalho.

RESUMO

Este trabalho apresenta a especificação e desenvolvimento de um sistema de controle de acesso de arquivos para dispositivos móveis utilizando *Web Services*. Apresenta um estudo sobre os *Web Services*, *.Net Framework*, dispositivos móveis e transferência de arquivos. É mostrado como a plataforma *.Net Compact Framework* possui recursos para desenvolvimento em *smartphones* e como a comunicação do dispositivo móveis com o servidor pode ser feita de maneira simples.

Palavras-chave: Dispositivos móveis. *Web service*. *.Net framework*.

ABSTRACT

This work presents the system's specification and development to control the file access in mobile devices using Web Services. Presents a study about Web Services, .Net Framework, mobile devices and file transfer. It's show how .Net Compact Framework platform have resources to development for smartphones e how the communication between mobile devices and server can be done easily.

Keywords: Mobile device. Web service. .Net framework

LISTA DE ILUSTRAÇÕES

Figura 1 - Evolução dos dispositivos móveis.....	15
Figura 2 - <i>Smartphone</i>	16
Figura 3 - Arquitetura do <i>.Net Framework</i>	19
Figura 4 - Arquitetura do <i>Web Service</i>	22
Figura 5 - Arquitetura de pilhas do <i>Web Service</i>	23
Figura 6 - Principais elementos de um WSDL.....	26
Quadro 1 - Prefixos de <i>namespaces</i> mais usados no WSDL	26
Figura 7 - Casos de uso do usuário.....	31
Figura 8 - Casos de uso da aplicação cliente.....	32
Figura 9 - Casos de uso da aplicação servidor	32
Figura 10 - Diagrama de Classes.....	33
Figura 11 - Diagrama da classe do <i>Web Service</i>	33
Figura 12 - Diagrama da classe <i>Tree</i>	34
Figura 13 - Diagrama da classe <i>Directory</i>	35
Figura 14 - Diagrama da classe <i>File</i>	35
Figura 15 - Diagrama de atividades do processo de autenticação.....	36
Figura 16 - Diagrama de atividades do processo de recebimento.....	37
Figura 17 - Diagrama de sequência do processo de autenticação	38
Figura 18 - Diagrama de sequência do processo de recebimento.....	39
Figura 19 - MER do banco de dados	40
Figura 20 - Ambiente de desenvolvimento Visual Studio	41
Figura 21 - Emulador de <i>smartphone</i>	42
Figura 22 - <i>Internet Information Services</i>	43
Figura 23 - <i>Web Service</i> publicado.....	44
Figura 24 - Tela de logon do sistema.....	45
Figura 25 - Acessar parâmetros do sistema.....	45
Figura 26 - Telas para fazer o recebimento de arquivos	46
Quadro 2 - Comparativo com trabalhos correlatos.....	47

LISTA DE SIGLAS

CE - *Compact Edition*

CLR - *Common Language Runtime*

ECMA - *European Computer Manufacturers Association*

FTP - *File Transfer Protocol*

GPS - *Global Position System*

HTTP - *Hypertext Transfer Protocol*

HTTPS - *Hypertext Transfer Protocol Secure*

IIS - *Internet Information Services*

IL - *Intermediate Language*

MER - MODELO ENTIDADE RELACIONAL

MSIL - *Microsoft Intermediate Language*

PDA - *Personal Digital Assistant*

PE - *Portable Executable*

RF - Requisitos Funcionais

RNF - Requisitos Não Funcionais

SMTP - *Simple Mail Transfer Protocol*

SOAP - *Simple Object Access Protocol*

TCP - *Transmission Control Protocol*

UDDI - *Universal Description, Discovery, and Integration*

UML - *Unified Modeling Language*

WSDL - *Web Service Definition Language*

XML - *eXtensible Markup Language*

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVOS DO TRABALHO	12
1.2 ESTRUTURA DO TRABALHO	12
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 DISPOSITIVOS MÓVEIS	14
2.1.1 Benefícios.....	15
2.1.2 Smartphones	16
2.2 .NET FRAMEWORK	17
2.2.1 CLR.....	18
2.2.2 .Net Compact Framework	19
2.2.3 Asp.Net	19
2.2.4 C#.....	20
2.3 WEB SERVICES.....	20
2.3.1 Arquitetura	21
2.3.2 SOAP	24
2.3.3 WSDL	25
2.4 TRANFERÊNCIA DE ARQUIVOS	27
2.5 TRABALHOS CORRELATOS	27
3 DESENVOLVIMENTO	29
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO	29
3.2 ESPECIFICAÇÃO	30
3.2.1 Diagrama de casos de uso	30
3.2.2 Diagrama de classes.....	32
3.2.3 Diagrama de atividades	36
3.2.4 Diagrama de sequência	37
3.2.5 Modelo Entidade Relacional (MER)	39
3.3 IMPLEMENTAÇÃO	40
3.3.1 Técnicas e ferramentas utilizadas	40
3.3.2 Operacionalidade da implementação	42
3.4 RESULTADOS E DISCUSSÃO	46
4 CONCLUSÕES	48

4.1 EXTENSÕES 49

1 INTRODUÇÃO

O uso de dispositivos móveis vem crescendo aceleradamente. Está cada vez maior o número de pessoas e empresas que têm necessidade de flexibilidade no acesso a dados e a troca de informações rapidamente.

Segundo dados do Gartner, o segmento que mais está crescendo na indústria de celulares, é a venda de *smartphones*. Segundo especialistas da área, esta é uma tendência natural, pois diariamente está se buscando por convergência de mídias, acesso rápido a internet, facilidade de uso e aplicativos úteis e interessantes.

Para acompanhar o aumento na popularidade destes dispositivos, surgem muitas demandas na área de sistemas e aplicativos para os dispositivos móveis. Para que se possa tirar o máximo de proveito dos recursos disponíveis, como conexão fácil à internet, câmera digital integrada, dispositivo Global Position System (GPS), capacidade de armazenamento, surge no mercado um novo setor destinado ao desenvolvimento de software voltado a área de dispositivos móveis.

Segundo Burégio (2003, p. 7), as aplicações para dispositivos móveis podem variar desde uma simples aplicação *stand-alone* a sistemas que permitam a troca de dados remotamente através da internet utilizando tecnologias como, por exemplo, *Web Services*. Atualmente muitas empresas investem na compra ou desenvolvimento de aplicativos com o objetivo de facilitar e agregar valor ao dia-a-dia dos funcionários. Esses aplicativos podem ser sistemas de vendas, sistemas de controle de estoque ou até mesmo de comunicação integrada com os recursos internos da empresa.

As áreas possíveis de desenvolvimento para os dispositivos móveis são as mais abrangentes possíveis, podendo se comparar com a de microcomputadores e servidores. Analisando todos os recursos que estes aparelhos oferecem, começam a surgir necessidades de sistemas que façam uso das tecnologias existentes, tendo como público alvo os usuários corporativos ou residenciais.

Segundo Gratsanuncios (2009), estes aparelhos têm se tornado populares para uma variedade de clientes no Brasil: “Os utilizadores finais têm demonstrado muito interesse em *smartphones*, não apenas no segmento corporativo, mas no segmento residencial”.

Tendo conhecimento dessas informações expostas acima, surgiu a idéia de explorar essa área de mercado que está em evidência, através da criação de uma ferramenta para transferência de arquivos entre dispositivos móveis e fixos.

A ferramenta que foi desenvolvida é, portanto, um repositório para disponibilização de arquivos para dispositivos móveis via web, implementado através de um *Web Service*. Para ambientes corporativos, pode-se fazer com que a empresa disponibilize para seus consultores e representantes externos, arquivos diários com informações pertinentes ao seu trabalho ou para que os funcionários externos possam enviar seus arquivos para o repositório da empresa. Para usuários residenciais, as utilidades também são expressivas, pois ele poderá facilmente navegar e acessar os arquivos do seu computador pessoal através do *smartphone*, fazer cópias de segurança e compartilhar arquivos com outras pessoas.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um sistema que, com o auxílio de *Web Services* através da internet, seja capaz de fazer o envio e recebimento de arquivos do dispositivo móvel para um servidor de arquivos ou computador pessoal.

Os objetivos específicos do trabalho são:

- a) criar dois módulos, um a ser executado no servidor que irá fazer a gerência do sistema e a disponibilização dos *Web Services* e o outro a ser executado no dispositivo móvel com Windows Mobile;
- b) realizar a transferência de arquivos do dispositivo móvel para o servidor, utilizando o *Web Service*;
- c) fazer o controle de acesso dos usuários, garantindo que apenas usuários autorizados utilizem o sistema.

1.2 ESTRUTURA DO TRABALHO

A estrutura deste trabalho está apresentada em três partes, a fundamentação teórica, o desenvolvimento do trabalho e as conclusões.

No capítulo sobre a fundamentação teórica, são detalhadas todas as informações necessárias para o entendimento da solução e desenvolvimento do sistema. São abordados assuntos relacionados aos dispositivos móveis e às tecnologias atuais, transferência de

arquivos entre dois pontos, tecnologias utilizadas para a programação do software e publicação do serviço de transferência dos dados entre os dispositivos.

No capítulo seguinte são detalhados todos os passos do desenvolvimento da ferramenta, bem como os requisitos, especificações e casos de uso. Também são exibidas as telas e detalhes de códigos de programas utilizados. Ao final, são detalhados os dados do resultado e discussões. E o último capítulo trata das considerações finais e sugestões para outros trabalhos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados conceitos e detalhes sobre assuntos relacionados ao trabalho. Por primeiro, é abordado o tema dispositivos móveis, posteriormente o *Framework* da Microsoft, o *.NET Framework*, a versão para dispositivos móveis (*Compact*) e a versão completa. Após trata sobre os *Web Services* e a tecnologia *Simple Object Access Protocol* (SOAP), transferência de arquivos e o emprego da tecnologia para ajudar o cotidiano das pessoas. Por fim, são apresentados alguns trabalhos correlatos com características em comum ao tema abordado.

2.1 DISPOSITIVOS MÓVEIS

Segundo Netto (2005), o termo mobilidade é utilizado para identificar dispositivos que podem ser operados a distância ou através de uma conexão sem fio. Estes dispositivos podem ser desde um simples Bip, até os mais modernos *Pockets* ou *Smart Phones*. O avanço tecnológico nos últimos anos está gerando uma grande transformação de conceitos. Paradigmas que antes impediam que uma nova tecnologia chegasse ao mercado estão sendo quebrados, e assim está se inovando cada dia mais.

De acordo com Pekus Consultoria e Desenvolvimento (2002), Pocket PCs e Palms são categorias de dispositivos móveis frequentemente utilizados em processos de computação móvel. Muito mais do que assistentes pessoais ou agendas eletrônicas, estes dispositivos 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.

Pekus Consultoria e Desenvolvimento (2002) também afirma que, para aqueles que consomem grande parte do seu tempo trabalhando remotamente, estes equipamentos são versáteis, dedicados, multifuncionais e de uso genérico. Do ponto de vista empresarial, eles são ótimos geradores de informações, podendo ser utilizados desde na automação de processos até na coleta de informações estratégicas.

Conforme Schaefer (2004, p. 20), as grandes inovações trazidas pelas tecnologias sem fio fizeram com que a indústria deste setor tenha tido um crescimento expressivo nos últimos

anos, tornando-se uma das mais eficientes e rápidas áreas tecnológicas do mundo, permitindo que as pessoas comuniquem-se de forma barata e fácil, sem ficarem presas aos seus telefones ou computadores de mesa. A figura 1 apresenta a evolução dos dispositivos móveis.



Figura 1 - Evolução dos dispositivos móveis

Segundo TelecomWeb (2004), um estudo realizado pela empresa de pesquisa Canalsys 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 Canalsys, 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%.

No Brasil com atraso em relação aos Estados Unidos, os usuários estão descobrindo que um *smartphone* é bem mais que uma agenda eletrônica de alto custo. A tendência é cada vez mais haver serviços voltados para estes dispositivos.

2.1.1 Benefícios

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.

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 conseqüentemente 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.

É crescente o número de empresas que investem em oferta de serviços através de dispositivos móveis, atualmente muitas tarefas já podem ser feitas diretamente através do *smartphone*, como acesso a conta bancárias, acompanhamento de notícias, divulgação de preços e até mesmo a concretização do processo de venda.

2.1.2 Smartphones

O *smartphone* (Figura 2) 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 *Personal Digital Assistants* (PDA), 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.



Figura 2 - *Smartphone*

Segundo Guimarães (2004), a transição para o *smartphone* no Brasil começou com os

*palm*s 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.

2.2 .NET FRAMEWORK

“A *.NET Framework* é um ambiente para a criação, instalação e execução de serviços da web e outros aplicativos. Consiste em três partes principais: a *Common Language Runtime* (CLR), as classes *Framework* e o ASP.NET.” (MICROSOFT CORPORATION, 2001, p. 108).

O *.NET Framework* é um conjunto de classes e bibliotecas da Microsoft que têm como objetivo criar uma plataforma única de desenvolvimento. Todo código que é gerado a partir deste *Framework* pode ser executado em qualquer plataforma ou dispositivo que tenha o *.NET Framework* instalado. Com isso, o desenvolvedor não precisa mais escrever o código direcionado para um dispositivo específico, bastando simplesmente escrever para a plataforma *.NET*.

A Microsoft lançou através do *Framework .NET* uma nova interface de programação para os serviços do Windows e expandiu a possibilidade de integração com novas tecnologias. Um dos maiores objetivos no lançamento deste ambiente é a integração de sistemas existentes com o universo internet. Por isso, essa plataforma possui recursos muito valiosos para o desenvolvimento de aplicativos e sistemas baseados na Internet.

Segundo Scribner e Stiver (2002, p. 53), partindo da perspectiva do *Web Service*, o aspecto mais importante do *.NET Framework* é a adoção de padrões de Internet como *eXtensible Markup Language* (XML), SOAP e *Web Service Definition Language* (WSDL).

O paradigma de programação da plataforma *.NET* é voltado para a orientação a objetos e conforme Microsoft Corporation (2001, p. 3), nos últimos anos, a programação orientada a objetos vem surgindo como a principal metodologia para o desenvolvimento de sistemas que atendem as necessidades do mercado, que deve ser escalável, resistente e reutilizável. Usar programação orientada a objetos facilita a compreensão de sistemas de larga

escala, simplifica a depuração e agiliza a atualização.

2.2.1 CLR

O *Common Language Runtime* é o coração da plataforma .NET. Trata-se de uma aplicação similar a uma máquina virtual da linguagem JAVA, que se encarrega de executar os aplicativos desenvolvidos para o .NET. É o CLR que é responsável pela interoperabilidade entre as linguagens suportadas pela plataforma.

Segundo Richter (2002, cap. 1), a Microsoft está dando suporte a diversas linguagens que tem como alvo o código intermediário *Microsoft Intermediate Language* (MSIL) do CLR: C++, C#, Visual Basic e J#. Ainda, segundo o mesmo autor, outras empresas estão disponibilizando compiladores para o CLR, nas mais diversas linguagens: Alice, APL, COBOL, *Component Pascal*, Eiffel, Fortran, Haskell, Mercury, ML, Mondrian, Oberon, Perl, Python, RPG, Scheme e Smalltalk.

No .NET *Framework*, independente da linguagem e do compilador utilizado, o resultado é sempre o mesmo: um módulo gerenciado, também conhecido como *assembly* .NET. Este módulo é composto por:

- a) um *Portable Executable* (PE): padrão para os arquivos executáveis no ambiente Windows desde a versão NT;
- b) *CLR header*: cabeçalho que contém informações sobre o módulo gerenciado;
- c) *metadata*: tabelas que descrevem tipos e dados definidos ou referenciados pelo código fonte;
- d) *intermediate language* (IL): o código intermediário que é produzido por um compilador da plataforma Microsoft .NET.

A figura 3 apresenta como está estruturado o Framework da Microsoft.

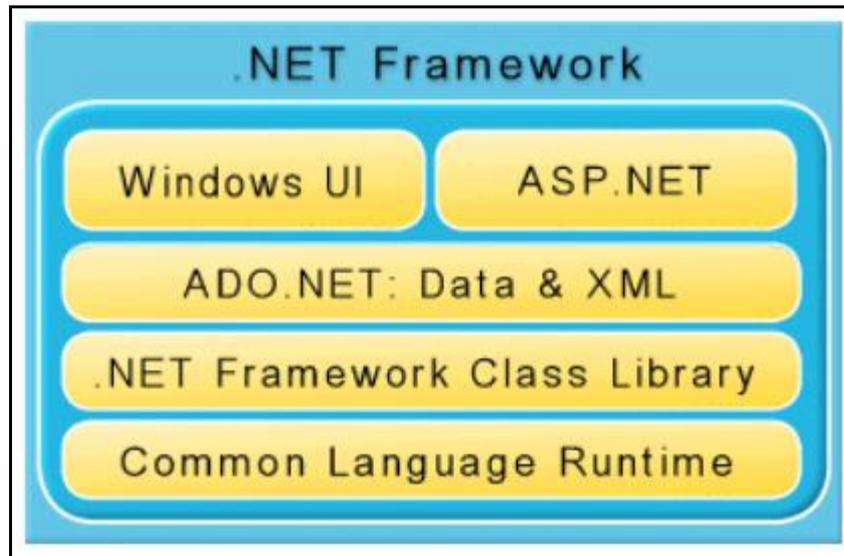


Figura 3 - Arquitetura do .Net Framework

2.2.2 .Net Compact Framework

“A Microsoft .NET Compact Framework é um subconjunto da Microsoft .NET Framework, criada para possibilitar o desenvolvimento de aplicações para dispositivos inteligentes (*smart devices*) da mesma forma com que aplicativos para Windows *desktop* são desenvolvidos.” (NETTO, 2005).

Esta versão do .NET é voltada para rodar em sistemas com Windows Compact Edition (CE), que normalmente são dispositivos de pequeno porte ou menor capacidade de processamento. É composta por algumas classes do .NET Framework e também possui algumas bibliotecas adicionais, específicas para uso em dispositivos móveis.

Complementando a categoria de soluções para equipamentos menores, existe também o produto de banco de dados da Microsoft, o *SQL Server Compact Edition*, que em parceria com o .NET Compact Framework, torna-se possível fazer um sistema completo e de alto nível.

2.2.3 ASP.NET

ASP.NET é a plataforma da Microsoft voltada à criação de conteúdo WEB dinâmico e também aplicativos WEB mais elaborados e confiáveis como os *Web Services*. Tem como objetivo reduzir as dificuldades no desenvolvimento WEB, tornando-se assim parte essencial

da plataforma no que diz respeito a aplicativos WEB (CEMBRANELLI, 2003, p. 17).

Fazendo parte do *.NET Framework*, todas as características e vantagens da plataforma *.NET* podem ser atribuídas ao ASP.NET, como o uso da biblioteca de classes *.NET*, compilação do código em MSIL, orientação a objetos entre outras.

Conforme Cembranelli (2003, p. 42), aplicações ASP.NET podem ser definidas como um conjunto de arquivos, páginas, módulos e códigos executáveis que podem ser acessados por meio de um diretório virtual ou subdiretório em um servidor WEB. As aplicações ASP.NET são executadas totalmente independentes, onde cada aplicação é tratada dentro de um *pool* de aplicativo, podendo assim cada aplicativo WEB manter suas próprias características de processamento e configuração.

Por ser uma tecnologia onde a execução é feita diretamente no servidor, e não no cliente, como o Javascript, o ASP.NET necessita da plataforma *.NET* instalada no servidor e de servidor Web para publicar a aplicação. Isso o torna dependente do *Internet Information Services* (IIS), servidor Web da Microsoft encontrado nos sistemas operacionais Windows 2000 Server ou superior.

2.2.4 C#

O C# (pronunciado como “csharp”) é uma linguagem de programação que utiliza o paradigma de orientação a objetos, que está no *Framework* da Microsoft. A linguagem C# é bastante similar à linguagem Java, e surgiu com o advento da plataforma Microsoft *.NET*.

A linguagem C# é uma especificação aberta, certificada pelo *European Computer Manufacturers Association* (ECMA), que pode ser consultada livremente através da Internet. Ela possui uma sintaxe baseada na linguagem C, e foi desenvolvida com o objetivo de combinar as possibilidades de programação da linguagem C++ com a simplicidade da linguagem Visual Basic.

2.3 WEB SERVICES

Os *Web Services* são serviços oferecidos via web, com o objetivo de integração e comunicação entre aplicações isoladas ou distribuídas. Atualmente é vastamente utilizado

como parte integrante de sistemas distribuídos, deixando a comunicação transparente e simples.

Conforme Scribner e Stiver (2002, p. 10), *Web Services* podem ser descritos como qualquer funcionalidade que está acessível pela internet, geralmente utilizando uma ou mais mensagens XML no protocolo de comunicação. *Web Services* usa o conceito de operação, que representa a associação de uma requisição para zero ou mais respostas. Quando estas operações são combinadas para satisfazer algum propósito específico, tem-se uma interface.

Segundo WEB service (2006), com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis.

A base para a construção de um *Web Service* são os padrões SOAP e XML. O transporte é feito totalmente via *Hypertext Transfer Protocol* (HTTP), através do formato XML e encapsulado no protocolo SOAP. A através da linguagem WSDL, os métodos do *Web Service* são descritos.

2.3.1 Arquitetura

O *Web Service* se baseia na integração de três entidades: provedor de serviços (*service provider*), servidor de registro (*service registry*) e cliente do serviço (*service consumer*). Esta iteração serve para que haja publicação, busca e execução das operações.

O provedor de serviços representa a entidade que hospeda o *Web Service*. É esta a entidade responsável por disponibilizar o serviço para que alguém possa utilizá-lo. Para que isto seja possível, seu formato deve estar descrito em um formato padrão, que seja compreensível por qualquer um que precise usar esse serviço. O provedor de serviços deve também publicar os detalhes sobre seu *Web Service* em um servidor de registros que esteja disponível.

O cliente do serviço pode ser um aplicativo em busca de um método ou uma pessoa acessando através de um browser, ou até mesmo outro *Web Service*. O cliente do serviço deve conhecer a interface de comunicação do *Web Service* para que possa consultá-lo. A descrição da interface é disponibilizada pelo provedor de serviços e recuperada a partir de uma pesquisa no servidor de registro. Através desta pesquisa, o consumidor de serviços pode obter o mecanismo para ligação com este *Web Service*.

Servidor de registro representa o registro e busca os *Web Services* baseados em

arquivos de descrição de serviços que foram publicados pelos provedores de serviços. Quando um cliente busca por um serviço no servidor de registro ele recupera informações referentes a interface de comunicação.

A figura 4 mostra a ligação destas três entidades.

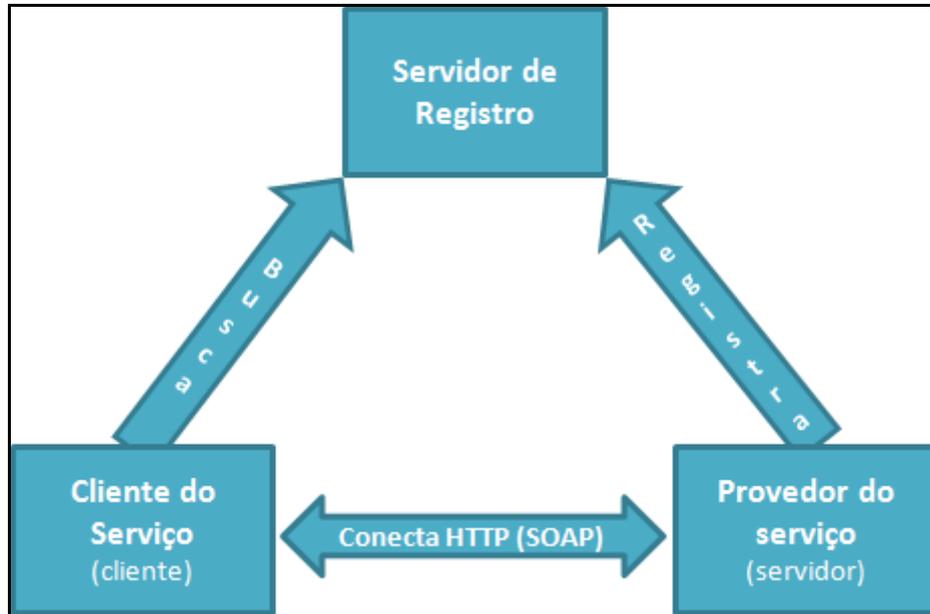


Figura 4 - Arquitetura do *Web Service*

Todo esse processo faz uso do protocolo HTTP para se comunicar. Como normalmente os *firewalls* não bloqueiam esta porta, não existem restrições para o seu funcionamento em redes internas ou internet.

A comunicação e troca de mensagens é possível por causa do pacote de camadas de um *Web Service*. Ela é implementada através de cinco tipos de tecnologias, organizadas em camadas definidas uma sobre a outra. A figura 5 mostra esta estrutura em camadas.

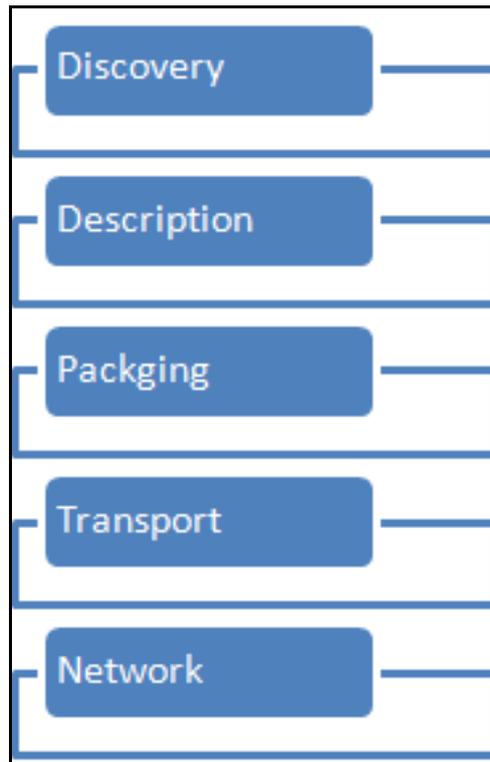


Figura 5 - Arquitetura de pilhas do *Web Service*

As camadas *packging*, *description* e *discovery* da pilha de tecnologias do *Web Service* são essenciais para o funcionamento do modelo de três entidades (*service provider*, *service registry* e *service consumer*).

Como cada camada da pilha do *Web Service* se dirige a um problema em separado, a implementação destas camadas também é independente. Ou seja, quando uma nova camada tiver que ser criada ou alterada, não haverá necessidade de alteração nas outras camadas.

A primeira camada da pilha (*discovery*), fornece o mecanismo que um consumidor de *Web Service* necessita para buscar a interface e descrições do provedor de serviço. A especificação usada e reconhecida para a criação desta camada é a *Universal Description, Discovery, and Integration* (UDDI). UDDI é uma especificação técnica que tem como objetivo descrever, integrar e descobrir *Web Services* (SNELL; TIDWELL; KULCHENKO, 2002, p. 7, tradução nossa).

Assim como as demais tecnologias do *Web Service*, o UDDI é baseado em XML, a qual fornece uma plataforma de dados neutra e permite descrever relações hierárquicas (RECKZIEGEL, 2006).

A segunda camada da pilha de tecnologias *Web Service* (*description*), é a camada responsável por descrever o que o *Web Service* pode oferecer ao cliente. De tal maneira que o cliente do serviço possa fazer uso do serviço (SNELL; TIDWELL; KULCHENKO, 2002, p. 7, tradução nossa). É nesta camada que se descreve os serviços externos, ou interfaces que são

oferecidos por um determinado Web Service. Para prover estas informações é usada uma linguagem especificada em XML, a WSDL. Neste XML, entre vários elementos, estão definidos os tipos de dados, parâmetros de entrada e saída de um serviço e protocolo usado para comunicação (RECKZIEGEL, 2006).

Para que os dados trafeguem pela rede, eles devem ser “empacotados” em um formato que todos entendam, ou seja, um formato padrão. A terceira camada (*packaging*) é a camada responsável por este empacotamento de dados, onde o SOAP é o padrão mais comum de empacotamento, baseado em XML (SNELL; TIDWELL; KULCHENKO, 2002, p. 8, tradução nossa).

A camada de transporte (*transport*) tem como objetivo mover dados entre dois ou mais locais em uma rede. Nela incluem várias tecnologias que permitem a comunicação entre aplicações. Entre estas tecnologias estão protocolos como *Transmission Control Protocol* (TCP), HTTP, *Simple Mail Transfer Protocol* (SMTP) entre outros. O *Web Service* pode ser construído sobre quase todos os protocolos de transporte (SNELL; TIDWELL; KULCHENKO, 2002, p. 8, tradução nossa).

A camada de rede (*network*) da pilha de tecnologias do *Web Services* é exatamente a mesma do modelo de rede de TCT/IP. É sobre esta camada que os protocolos de transporte trafegam.

2.3.2 SOAP

SOAP é um protocolo para troca de mensagens em ambiente distribuído, baseado em definições XML. Utilizado para acessar *Web Services*, seu papel básico é de definir mecanismos que expressem o modelo semântico das aplicações, assim como o modelo em que os dados são decodificados. Esse protocolo empacota as chamadas e retornos aos métodos dos *Web Services*, sendo utilizado principalmente sobre HTTP (RECKZIEGEL, 2006).

Conforme Seely (2002, p. 43), o SOAP é um mecanismo de comunicação entre as aplicações em uma plataforma descentralizada e distribuída. A especificação do SOAP é dividida em quatro partes principais:

- a) SOAP *envelope*: define uma plataforma para descrição do que há na mensagem e como processá-la. Ela guarda todos os dados da mensagem e é a única parte do protocolo que é obrigatória;

- b) SOAP encoding rules: define um mecanismo de serialização que pode ser usado para a troca de instâncias de tipos definidos pela aplicação;
- c) SOAP RPC style: define uma convenção que pode ser usada para representar chamadas e respostas remotas a procedimentos, nada mais que a dupla requisição/resposta, que não é obrigatória;
- d) link SOAP-HTTP: define um protocolo que liga o SOA e o HTTP. Ele descreve como mensagens SOAP são transmitidas via HTTP.

2.3.3 WSDL

Web Service Definition Language define um sistema para descrição de serviços. Através desta linguagem são descritos os serviços e interfaces oferecidos por uma determinada aplicação, assim como a sua localização. Como outras tecnologias para *Web Services*, sua especificação também é baseada no XML (RECKZIEGEL, 2006).

Em um documento WSDL existem elementos que formam sua especificação. Estes elementos servem para definir os parâmetros de um determinado serviço.

Para Reckziegel (2006), os elementos deste documento estão definidos como:

- a) types (tipos): container de definição de tipos de dados;
- b) message (mensagem): define parâmetros de entrada e saída de um serviço;
- c) operation (operações): são as definições dos métodos, relação entre parâmetros de entrada e saída;
- d) port type (tipo de porta): descreve o agrupamento lógico das operações, ou seja, das definições dos métodos;
- e) binding (vínculo): especifica o protocolo e formato de dado a ser usado para um dado port type;
- f) service (serviço): define a localização real do serviço. É uma coleção que pode conter várias portas, e cada uma é especificada para um tipo de vínculo.

Estes elementos podem ser mais bem compreendidos se observada a figura 6, que demonstra os principais elementos de um documento WSDL.

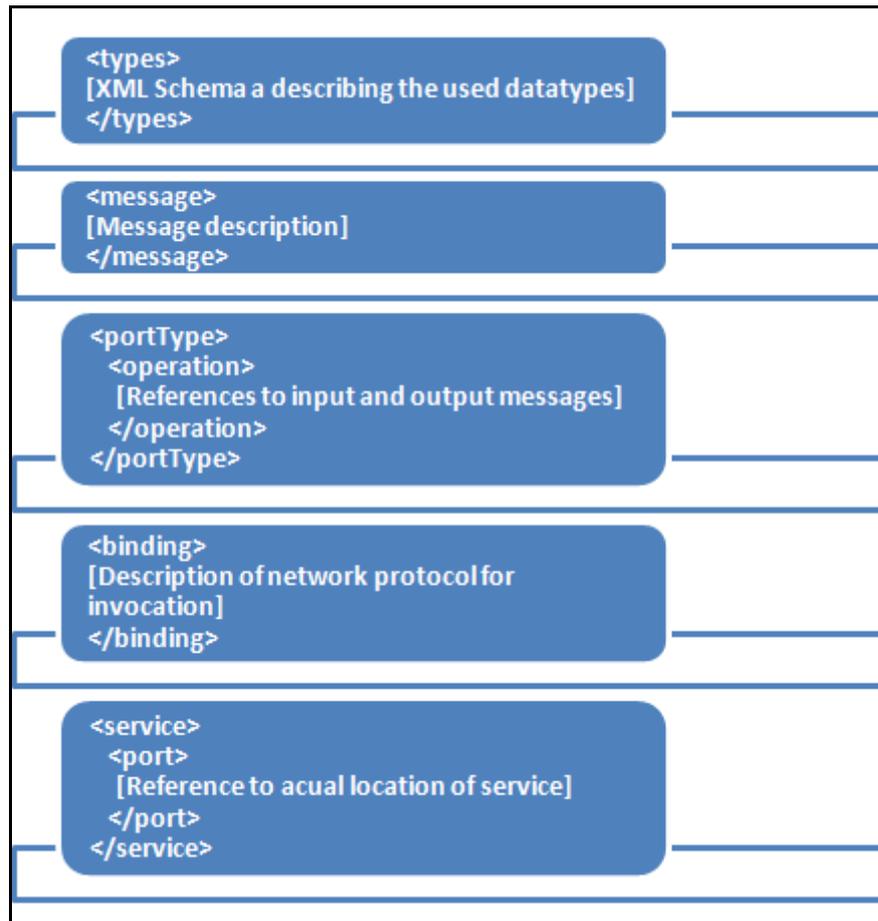


Figura 6 - Principais elementos de um WSDL

Como em qualquer XML, o WSDL também precisa de um elemento raiz. Este elemento chama-se *definitions*.

O WSDL utiliza *namespaces* para aumentar a reutilização dos elementos/componentes definidos em seu documento. Estes *namespaces* são espaços para nomes definidos no interior de um XML, o que permite sua unicidade de nomes.

Os principais *namespaces* usados em um documento WSDL estão indicados no Quadro 1.

Prefixo	URI do Namespace	Descrição
WSDL	http://schemas.xmlsoap.org/wsdl	Namespace de WSDL para Framework WSDL
HTTP	http://schemas.xmlsoap.org/wsdl/http	Namespace de WSDL para WSDL HTTP GET e vínculo POST
MIME	http://schemas.xmlsoap.org/wsdl/mime	Namespace de WSDL para vínculo MIME DE WSDL
XSD	http://www.w3.org/2001/XMLSchema	Namespace do esquema conforme definido pelo esquema XSD

Fonte: adaptado de Seely (2002).

Quadro 1 - Prefixos de *namespaces* mais usados no WSDL

Quando observado um documento WSDL é preciso lembrar que ele é apenas um

documento XML. Ele contém elementos que derivam de seu *namespace* e *schema*, definido por um XSD (SEELY, 2002, p. 167, tradução nossa).

2.4 TRANFERÊNCIA DE ARQUIVOS

A transferência de arquivos é um dos recursos mais utilizados pelos usuários da internet. Encontram-se espalhados nos computadores pelo mundo milhares de documentos ou aplicações científicas, demonstrativos de jogos, aplicativos para diversos sistemas operacionais, diversos artigos e dicas sobre determinados programas ou periféricos existentes no mercado. Todos estes recursos e muitos outros não mencionados estão ao alcance dos usuários da Internet, desde que, é claro, possuam as ferramentas adequadas para a transferência de arquivos entre máquinas

Segundo *FILE transfer protocol* (2006), a transferência de arquivos ocorre através de um computador chamado cliente, que inicia a conexão fazendo uma requisição e um servidor, que recebe a requisição e recebe ou envia o arquivo solicitado. O usuário pode selecionar quais arquivos enviar ou receber do servidor. Para existir uma conexão ao servidor, caso o servidor exija, o utilizador informa um nome de usuário e uma senha, bem como o nome do servidor ou seu endereço IP. Se os dados foram informados corretamente, a conexão pode ser estabelecida, utilizando-se dois canais de comunicação, chamados de portas. Tais portas são conexões no qual é possível trocar dados. No caso da comunicação *File Transfer Protocol* (FTP), são utilizadas duas portas. Uma de controle (porta 21) e outra para transferência de arquivos (porta 20).

A transferência de arquivos também pode ocorrer através de *Web Services*, entre um computador cliente e um servidor. A principal vantagem da transferência via FTP, é a segurança. Com o uso de *Web Services*, a transferência pode ocorrer de maneira segura, com o uso de certificados digitais e totalmente através da porta padrão de internet, a porta 80.

2.5 TRABALHOS CORRELATOS

A seguir são apresentados alguns trabalhos com características em comuns ao tema

proposto. Estes trabalhos têm características específicas do seu escopo, mas englobam muitas áreas relacionadas.

O protótipo desenvolvido por Galvin (2004) teve como objetivo a criação de um protótipo de um software para a integração e troca de dados com um aplicativo cliente/servidor de uma empresa através de dispositivos móveis. A aplicação visa auxiliar o trabalho de relacionamento entre empresa e cliente diretamente no campo, disponibilizando informações importantes para os usuários sobre os seus clientes. A ferramenta foi desenvolvida em Visual Studio .NET e com banco de dados SQL Server 2000, na versão Compact Edition. O protótipo da ferramenta obteve os resultados esperados, com exceção do desempenho que ficou um pouco baixo, conforme afirma o autor.

Outro trabalho analisado é o protótipo de um sistema para controle e monitoração residencial à distância através de dispositivos móveis utilizando a plataforma .NET (VENTURI, 2005). Teve como objetivos principais controlar e monitorar objetos de uma residência, criar a comunicação da casa com o dispositivo móvel e desenvolver um simulador de um ambiente residencial. Este trabalho foi desenvolvido com o Visual Studio .NET com a linguagem Visual C# e utilizando *Web Services* para a comunicação de dados via Internet.

Junges (2006) criou um software para controle de entregas utilizando dispositivos móveis e *Web Services* sobre a plataforma .NET. Teve como objetivo a exploração das tecnologias envolvidas, no sentido de troca de dados e integração de dispositivos móveis com *Web Services*. Com isso, desenvolveu um software para controlar e facilitar a entrega de encomendas. O sistema foi desenvolvido com o Visual Studio .NET, banco de dados SQL Server 2000 e com o auxílio de emuladores para dispositivos móveis.

3 DESENVOLVIMENTO

O sistema deste trabalho é formado por dois módulos, um aplicativo e um *Web Service*. O aplicativo é executado no dispositivo móvel e faz a interação do sistema com o usuário. Através deste que o usuário fará todas as operações no sistema. O *Web Service* é executado no servidor e é responsável por executar as requisições do aplicativo cliente. Toda comunicação entre o aplicativo e o servidor é realizada através do *Web Service*, pelo protocolo HTTP.

Na sequência serão apresentados os requisitos do sistema.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O sistema foi desenvolvido considerando os seguintes Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF):

- a) permitir que o usuário do dispositivo móvel envie e receba arquivos do servidor (RF);
- b) permitir que o usuário do dispositivo móvel remova arquivos e pastas no servidor (RF);
- c) permitir que o usuário do dispositivo móvel renomear arquivos e pastas no servidor (RF);
- d) permitir que o usuário do dispositivo móvel crie novas pastas no servidor (RF);
- e) permitir que o usuário do dispositivo móvel bloqueie a alteração ou sobreposição de arquivos no servidor (RF);
- f) permitir que o usuário cadastre os usuários do sistema (RF);
- g) permitir que o usuário do dispositivo móvel receba informações sobre arquivos do servidor, tais como tamanho, data de alteração e último usuário que modificou (RF);
- h) permitir que o usuário receba dados estatísticos sobre a área de armazenamento no servidor (RF);
- i) ter um método de autenticação, garantido somente acessos autorizados (RF);

- j) garantir que os arquivos cheguem de forma íntegra e consistente ao servidor, utilizando métodos de verificação de consistência (RNF);
- k) ser implementado com o .NET Framework da Microsoft (RNF);
- l) utilizar tecnologia de *Web Services* para a comunicação entre os clientes e o servidor (RNF);
- m) utilizar banco de dados SQL Server para armazenar as informações do sistema no servidor (RNF).

3.2 ESPECIFICAÇÃO

Nesta seção são apresentadas as especificações do sistema apresentado. Para especificação dos requisitos foram utilizadas as técnicas da *Unified Modeling Language* (UML) através da ferramenta Enterprise Architect, para descrição dos casos de uso, diagramas de classes e diagramas de sequências.

3.2.1 Diagrama de casos de uso

São demonstrados os principais casos de uso do usuário do sistema, do aplicativo e do *Web Service*.

O usuário possui seis casos de uso (figura 7):

- a) autenticar: responsável pela autenticação do usuário no aplicativo do dispositivo móvel, para garantir a autenticidade do usuário e segurança do uso do aplicativo;
- b) enviar e receber arquivos: responsável pelas operações de enviar e receber arquivos do dispositivo móvel para o servidor ou do servidor para o dispositivo móvel;
- c) remover arquivo ou pasta: responsável pelas operações de remover arquivos ou pastas do servidor através do aplicativo no dispositivo móvel;
- d) renomear ou criar nova pasta: responsável pelas operações de renomear arquivos ou pastas do servidor através do aplicativo no dispositivo móvel;
- e) bloquear ou desbloquear arquivo: responsável pela operação de bloquear ou desbloquear um arquivo no servidor;
- f) ver relatórios: responsável pela visualização dos relatórios do sistema ao usuário.

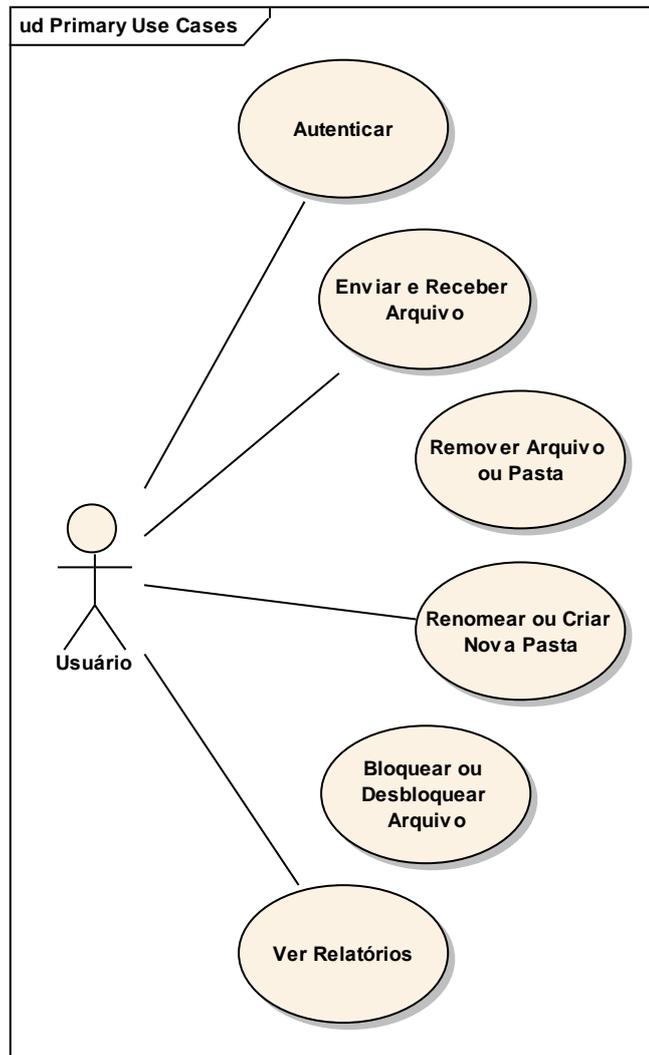


Figura 7 - Casos de uso do usuário

O aplicativo cliente que é executado no dispositivo móvel possui 2 casos de uso (figura 8):

- a) envia requisição para o servidor: Responsável por processar as solicitações do usuário, se comunicar com o aplicativo servidor e fazer as chamadas aos serviços do *Web Service*;
- b) recebe resposta do servidor: responsável por processar as respostas do servidor das requisições enviadas. Exibe ao usuário as mensagens de retorno, confirmação ou erro nas operações solicitadas.

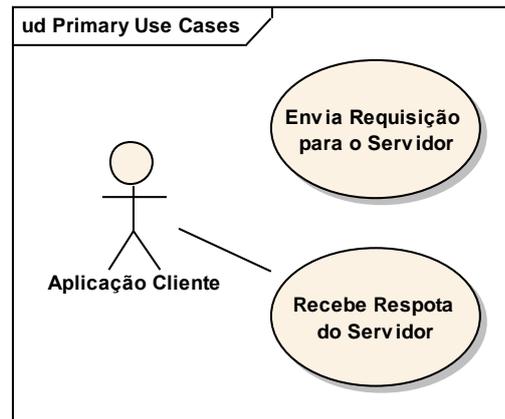


Figura 8 - Casos de uso da aplicação cliente

O aplicativo servidor possui 2 casos de uso (figura 9):

- a) recebe requisição do cliente: responsável pelo recebimento das requisições do cliente, que são feitas através dos serviços do *Web Services*.
- b) executa requisições e retorna resposta ao cliente: responsável pela execução das requisições recebidas do cliente através do *Web Service*. Após a execução, o servidor retorna para o cliente um resultado, proveniente da operação executada.

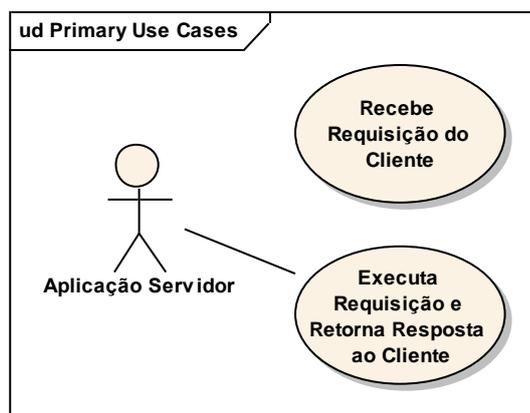


Figura 9 - Casos de uso da aplicação servidor

3.2.2 Diagrama de classes

A especificação foi desenvolvida baseada na orientação a objeto, utilizando o diagrama de classes da UML como linguagem de modelagem.

A figura 10 a seguir apresenta o modelo de classes do módulo servidor:

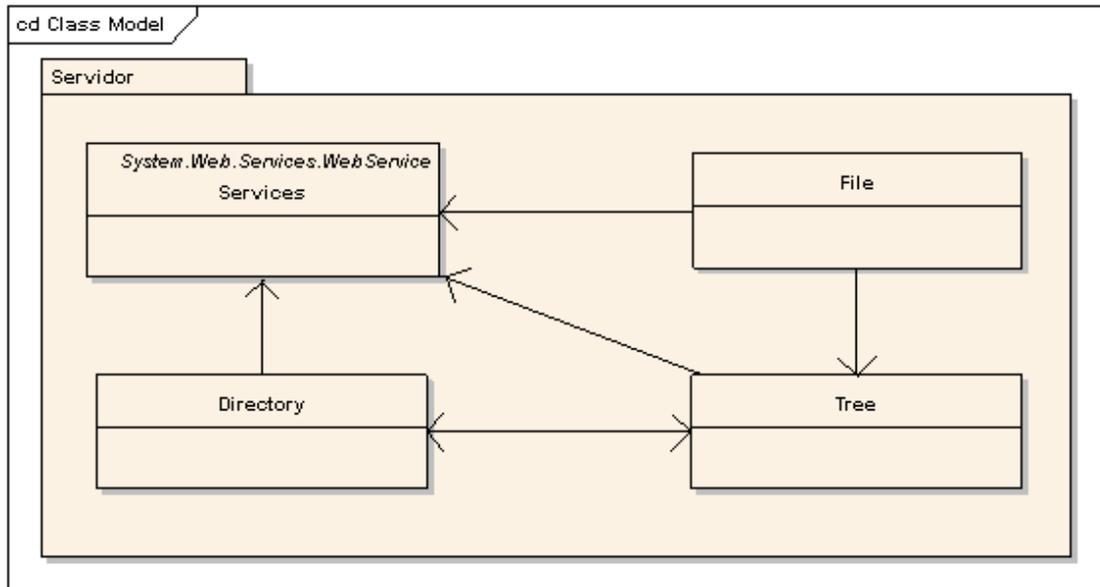


Figura 10 - Diagrama de Classes

A figura 11 apresenta os detalhes da classe *Services*, que possui os métodos do *Web Service*. Todos os métodos apresentados abaixo executam as operações com os arquivos e pastas no servidor, sendo necessário informar o diretório inicial das operações. A classe *Services* depende das classes *Tree*, *File* e *Directory* para armazenar informações sobre a estrutura hierárquica de pastas e arquivos e atributo de pastas e arquivos.



Figura 11 - Diagrama da classe do *Web Service*

A figura 12 apresenta os detalhes da classe *Tree*. Esta classe é auxiliar para o *Web*

Service, pois através desta estrutura é possível gravar a árvore de arquivos e pastas a partir do diretório inicial informado. Esta classe depende das classes *Directory* e *File*, para armazenar informações sobre os diretórios e arquivos da árvore.



Figura 12 - Diagrama da classe `Tree`

A figura 13 apresenta os detalhes da classe *Directory*. Esta classe é auxiliar para o *Web Service*, pois através desta estrutura é possível gravar informações sobre pastas a partir do diretório inicial informado.



Figura 13 - Diagrama da classe Directory

A figura 14 apresenta os detalhes da classe File. Esta classe é auxiliar para o Web Service, pois através desta estrutura que o sistema representa um arquivo.

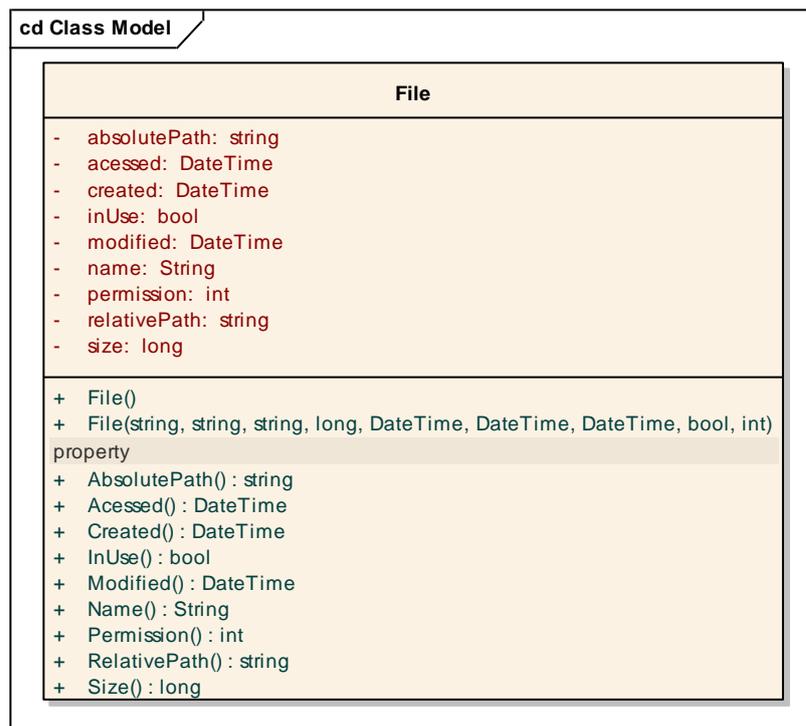


Figura 14 - Diagrama da classe File

3.2.3 Diagrama de atividades

O diagrama de atividades foi desenvolvido para facilitar a visualização do processo de determinadas situações do sistema. O diagrama da figura 15 mostra as atividades do processo do caso de uso autenticar.

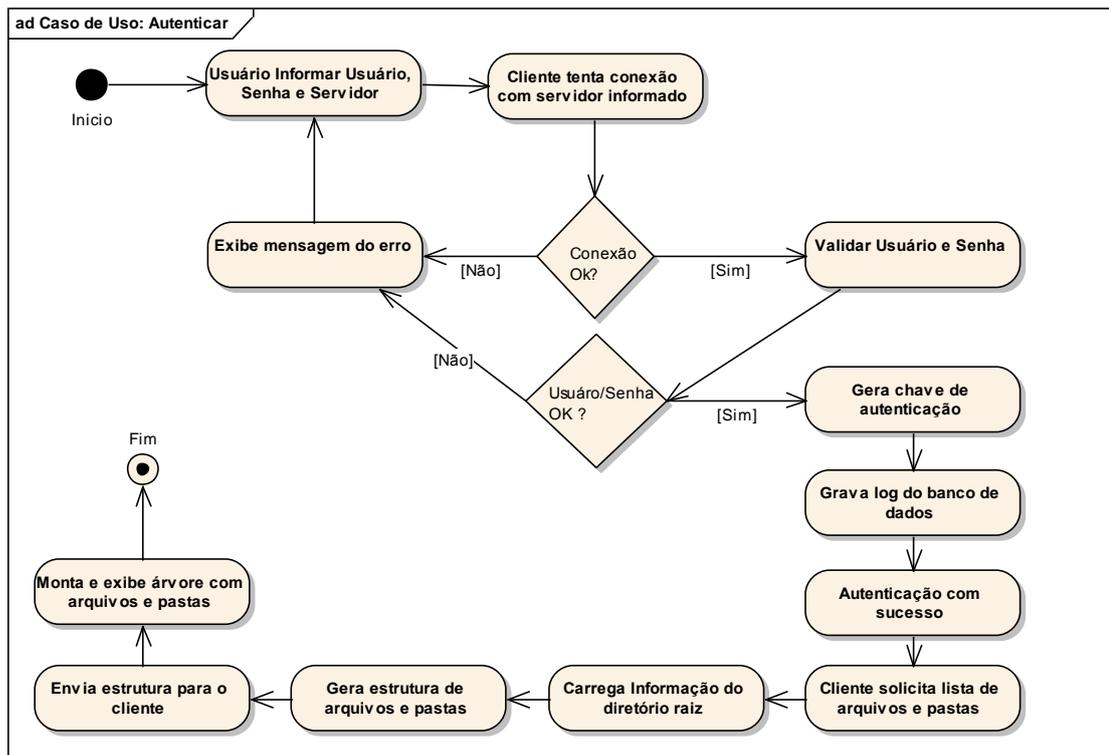


Figura 15 - Diagrama de atividades do processo de autenticação

No diagrama apresentado, o usuário informa ao sistema no dispositivo móvel o seu nome de usuário, senha e servidor. O sistema no dispositivo móvel tenta conexão com o servidor informado para validar as informações de nome de usuário e senha. O módulo no servidor recebe a solicitação de autenticação através do *Web Service Login*, gera a chave de autenticação, grava o registro no banco de dados e responde ao cliente com a chave de autenticação. Após estar autenticado, o cliente solicita ao servidor a árvore de arquivos e pastas. O servidor recebe a solicitação através do *Web Service GetTree*, faz geração recursiva de árvore de arquivos e pastas e retornar uma instância do objeto *Tree* para o cliente. O módulo cliente abre o objeto *Tree* recebido do servidor e monta a visualmente a estrutura de arquivos e pastas na tela do dispositivo móvel.

O diagrama representado na figura 16 detalha o processo do caso de uso receber arquivo pelo dispositivo móvel no aplicativo cliente.

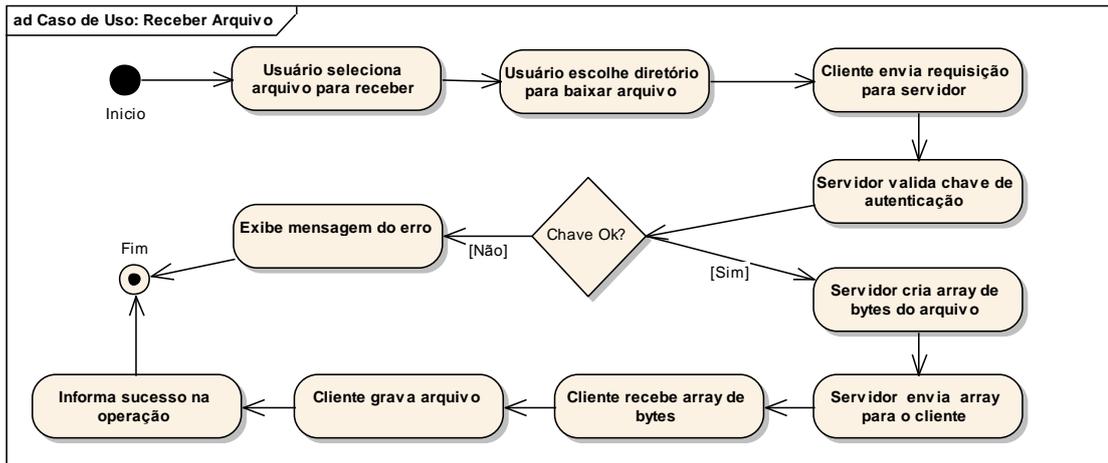


Figura 16 - Diagrama de atividades do processo de recebimento

O usuário navega pela árvore de arquivos e pastas, seleciona um arquivo e solicita o recebimento. O sistema solicita ao usuário que selecione a pasta que deseja salvar o arquivo e envia a requisição do *download* do arquivo junto com a chave de autenticação para o servidor. O servidor faz a validação de chave de acesso para verificar se é uma sessão ativa, localiza e faz a abertura do arquivo informado, gera um *array* de *bytes* do arquivo informado e retorna o objeto ao cliente. O cliente monta o arquivo a partir do *array* de *bytes* recebido e informa o sucesso da operação para o usuário.

3.2.4 Diagrama de sequência

O diagrama de sequência é um diagrama usado pela UML e representa a sequência dos processos e as mensagens transmitidas entre os objetos.

A figura 17 apresenta o diagrama de sequência do caso de uso autenticar.

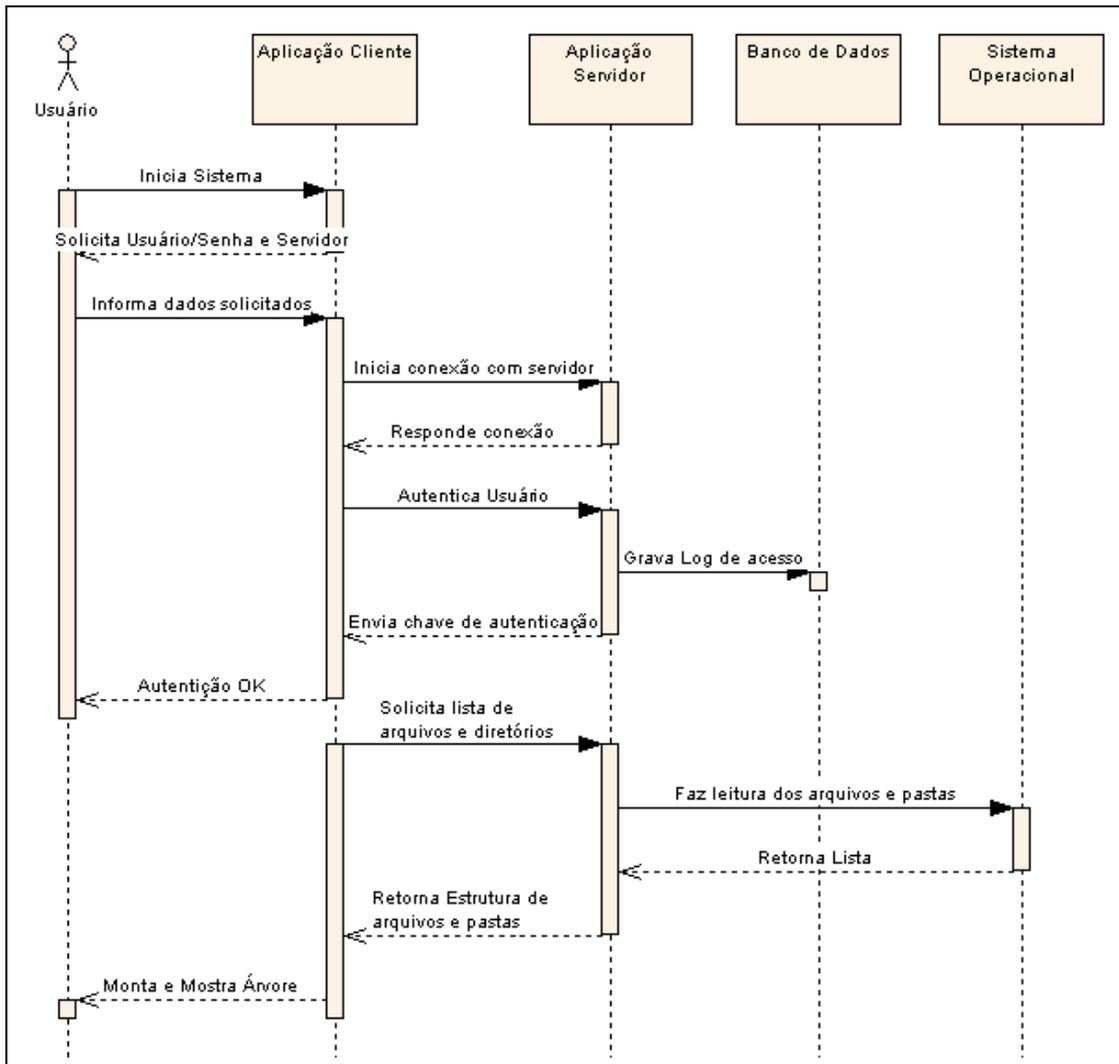


Figura 17 - Diagrama de sequência do processo de autenticação

A figura 18 apresenta o diagrama de sequência do caso de uso receber arquivo.

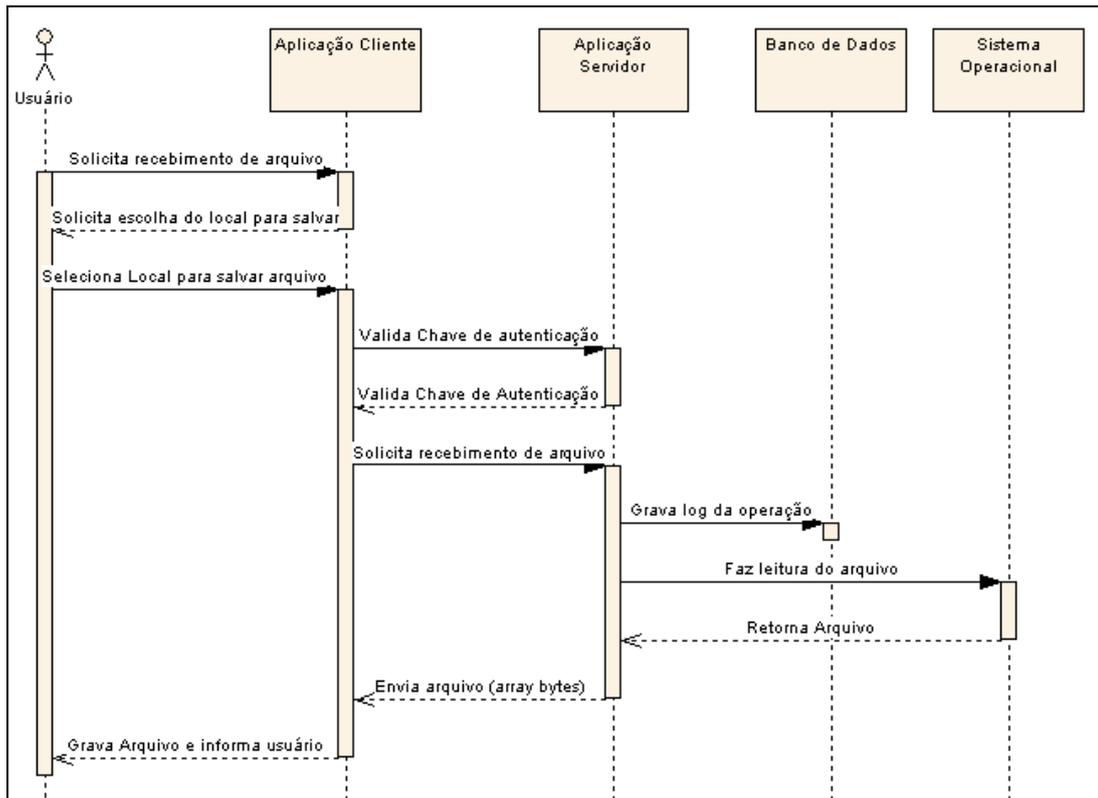


Figura 18 - Diagrama de sequência do processo de recebimento

3.2.5 Modelo Entidade Relacional (MER)

O MER tem como objetivo representar de forma visual as estruturas de dados de um banco de dados. Para desenhar o MER deste sistema foi utilizado a ferramenta *DB Designer* da fabFORCE.net.

Como banco de dados, é utilizado o *Microsoft SQL Server* na versão *Express* e para fazer o gerenciamento e criação dos objetos é utilizado o *Microsoft SQL Server Studio Management*. A figura 19 mostra o MER da estrutura de banco de dados utilizada neste trabalho.

Na tabela `Log` são armazenadas todas as operações dos usuários no dispositivo, tais como acesso ao sistema, envio ou recebimento de arquivos, remoção de arquivo. A tabela `Parametro` armazena os parâmetros de uso geral do sistema, tais como diretório padrão para os usuários, tempo limite do aplicativo cliente. A tabela `Usuario` é responsável por armazenar os usuários cadastrados para acessar o sistema no dispositivo móvel. Na tabela `Bloqueio` são gravados os bloqueios aos arquivos efetuados pelos usuários.

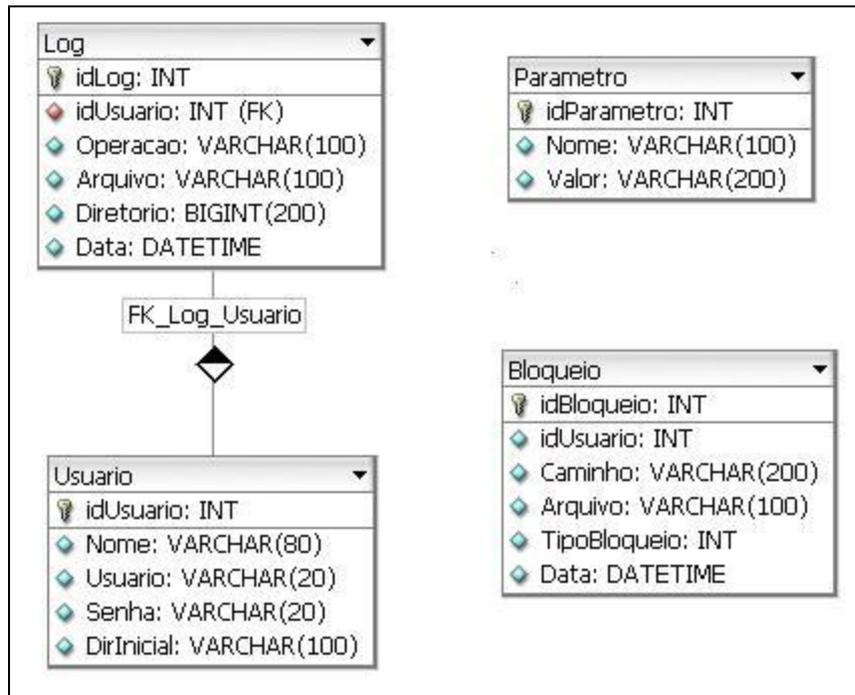


Figura 19 - MER do banco de dados

3.3 IMPLEMENTAÇÃO

A seguir serão conceituadas as técnicas e ferramentas utilizadas na análise, especificação e implementação do trabalho.

3.3.1 Técnicas e ferramentas utilizadas

O desenvolvimento do trabalho foi realizado através da linguagem C# encontrada no *.NET Framework* da Microsoft. A mesma linguagem foi utilizada para fazer o módulo servidor e módulo cliente. Como ambiente de desenvolvimento, foi utilizado o Microsoft Visual Studio. Com esta ferramenta foi possível desenvolver todos os módulos deste trabalho.

Microsoft Visual Studio é uma ferramenta que permite a criação de projetos de *web sites*, aplicativos *desktops*, *Web Services*, banco de dados e para dispositivos móveis. A figura 20 mostra uma imagem do ambiente de desenvolvimento neste projeto e uma parte do código fonte utilizado na criação do *Web Service*.

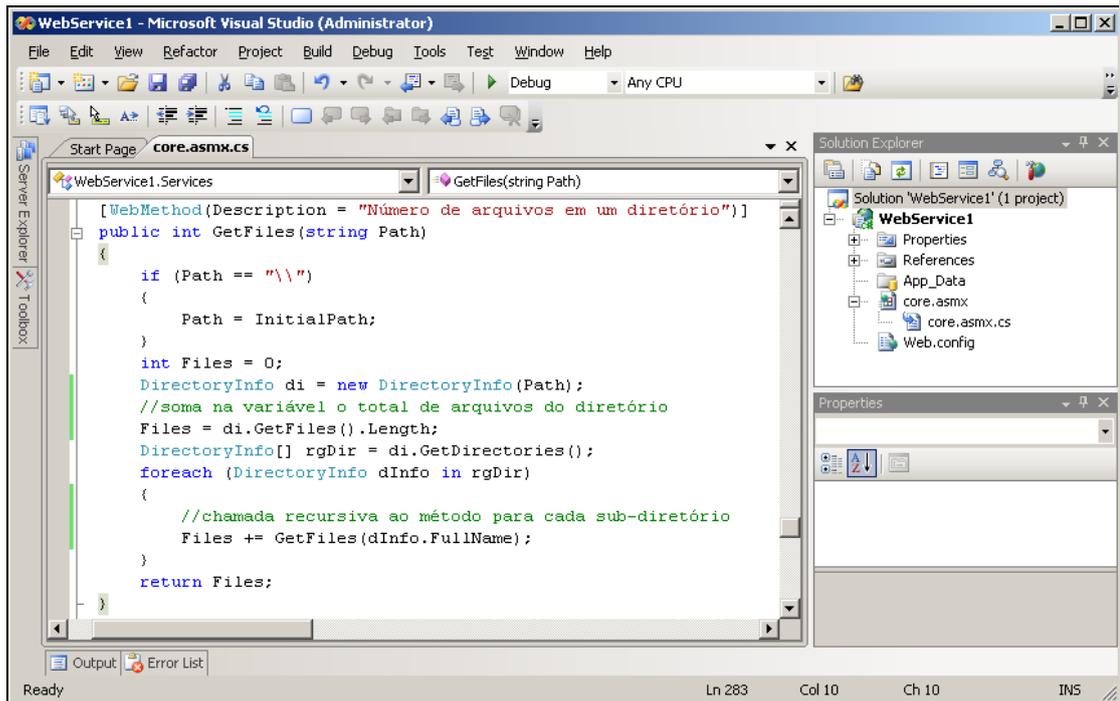


Figura 20 - Ambiente de desenvolvimento Visual Studio

Para o aplicativo no dispositivo móvel, durante o processo de desenvolvimento foi utilizado o emulador de dispositivos do Microsoft Visual Studio. Ele proporciona um ambiente virtual de um dispositivo que possui os mesmos recursos e funcionalidades de um aparelho físico. Com o emulador, é possível conectar o aparelho virtual na internet, simular recebimento de mensagens de texto e ligações pelo celular. A grande vantagem de usar o emulador durante o processo de desenvolvimento, é a possibilidade de executar um programa passo a passo, facilitando a localização de erros e resolução de problemas.

A figura 21 mostra o emulador executando o sistema desenvolvido neste trabalho. Para manusear o emulador, é necessário clicar com o mouse sobre o teclado do aparelho e se necessário, também é possível utilizar o teclado para facilitar na digitação de textos.



Figura 21 – Emulador de *smartphone*

3.3.2 Operacionalidade da implementação

Nesta seção são apresentadas algumas funcionalidades do sistema desenvolvido e requisitos para configuração do servidor. Primeiramente é necessário configurar o servidor que hospedará o *Web Service*. Para isso é necessário que o servidor esteja com o recurso IIS habilitado e com o *.NET Framework* instalado. O IIS é o serviço que provê a funcionalidade de servidor web, através dele vai-se publicar o *Web Service* do sistema. A figura 22 mostra o servidor *web* IIS configurado e a figura 23 mostra a tela do *Web Service*.

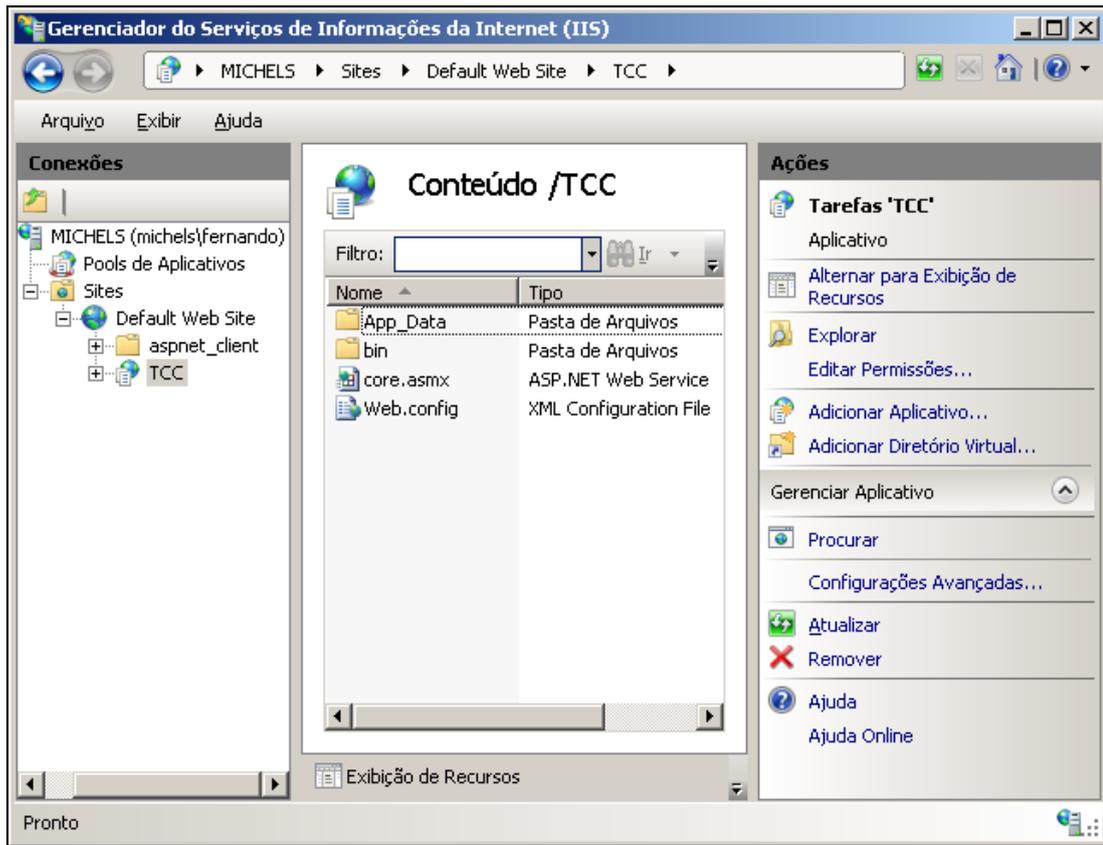


Figura 22 - Internet Information Services

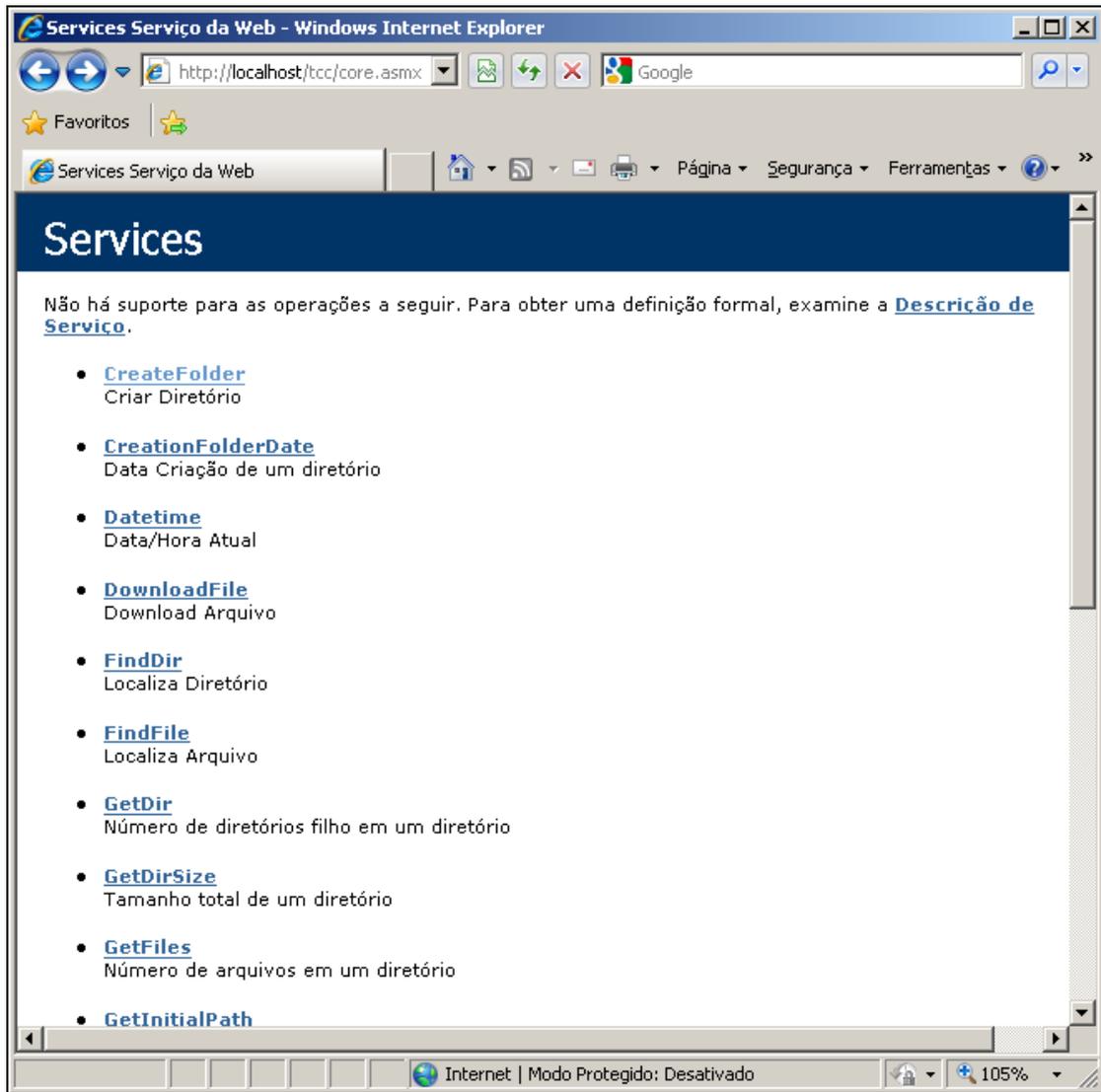


Figura 23 - Web Service publicado

Para instalar o módulo cliente no dispositivo móvel, basta copiar o arquivo executável para dentro do aparelho e executá-lo. O dispositivo móvel precisa estar com o *.NET Compact Framework* instalado. Ao acessar o sistema no aparelho, será apresentado a tela de autenticação e o campo para preencher o endereço do servidor, que já vem com um valor padrão correto, conforme mostra a imagem 24. A usuário e senha padrão são admin/admin. A manutenção das contas de usuários de acesso pode ser feito pelo área de administração.



Figura 24 - Tela de *logon* do sistema

Após efetuar o logon, o sistema automaticamente monta na tela do dispositivo móvel a árvore de pastas e arquivos a partir do diretório inicial cadastrado para o usuário. Caso o usuário queira alterar o seu diretório inicial, o mesmo pode ser feito pelo Sistema, conforme mostra a imagem 25.



Figura 25 - Acessar parâmetros do sistema

Para fazer o recebimento de um arquivo, o usuário precisa selecionar o arquivo, ir até o menu Operações e clicar em Receber. Após isso, o sistema irá abrir uma tela para que o usuário possa escolher o local do arquivo, conforme mostra a figura 26.



Figura 26 - Telas para fazer o recebimento de arquivos

Após a confirmação do local para salvar o arquivo, o sistema irá solicitar e receber o arquivo do servidor, gravar no local indicado e exibir uma mensagem de sucesso.

3.4 RESULTADOS E DISCUSSÃO

Este trabalho atingiu o seu objetivo proposto. O sistema é capaz de fazer a transferência de arquivos entre o servidor e o dispositivo móvel. Possui um sistema de autenticação para evitar acessos indevidos, e registra no banco de dados as operações executadas pelos usuários.

O desempenho do sistema no dispositivo móvel não se mostrou muito satisfatório, havia uma demora no momento de carregar o aplicativo e carregar a árvore de arquivos e pastas. O principal motivo encontrado é que o aplicativo está rodando em um sistema emulado e não diretamente em um *smartphone*. Quanto ao desempenho do *Web Service* no servidor, este se mostrou extremamente rápido. A operação que se acreditava ser demorada, que é no momento de fazer a varredura recursiva nos arquivos e pastas do sistema operacional, se mostrou muito eficaz e apresentou ótimos resultados.

Além das operações de envio e recebimento, o sistema ainda é capaz realizar outras operações, como a criação de novas pastas no servidor, renomear arquivos ou pastas, remover arquivos ou pastas, visualizar as propriedades de um arquivo ou pasta. Também é possível bloquear um arquivo para alteração, deixando-o somente leitura.

O quadro 2 mostra um comparativo com os trabalhos correlatos. Não foi encontrado nenhum trabalho que tivesse o mesmo objetivo deste, então o critério foi o uso das mesmas tecnologias e funcionalidades próximas.

	Este trabalho	Galvin (2004)	Venturi (2005)	Junges (2006)
Plataforma de desenvolvimento	Visual Studio	Visual Studio	Visual Studio	Visual Studio
Linguagem de Programação	C#	VB	C#	VB
Banco de Dados	SQL Server	SQL Server	SQL Server	SQL Server
<i>Web Services</i>	Sim	Sim	Sim	Sim
Transferência de arquivos	Sim	Não	Não	Não
Objetivo Principal	Controle no acesso a arquivos pelo dispositivo móvel	Aplicativo CRM pelo dispositivo móvel	Controle e monitoramento de residências pelo dispositivo móvel	Controle de entregas pelo dispositivo móvel

Fonte: Adaptado de Galvin (2004), Venturi (2005), Junges (2006).

Quadro 2 - Comparativo com trabalhos correlatos

4 CONCLUSÕES

Com o auxílio das ferramentas descritas e das pesquisas realizadas, foi possível alcançar todos os objetivos propostos. Certamente houve dificuldades, mas foram superadas e ao final do projeto o resultado foi positivo.

Foi verificado que o desenvolvimento para dispositivos móveis está mais acessível. Apesar de ter componentes visuais ainda limitados, com as ferramentas utilizadas neste trabalho, como o *.NET Compact Framework*, o Microsoft Visual Studio e o seu emulador de dispositivo móveis não há barreiras para a criação de projetos nesta linha.

A utilização de *Web Services* como solução para disponibilização dos serviços foi fundamental. Com esta tecnologia é possível a comunicação de uma aplicativo servidor com qualquer outro que seja capaz de se conectar ao *Web Service*. Outra vantagem da utilização deste recurso é a facilidade de encontrar documentação.

A maior dificuldade encontrada foi no momento da criação do módulo para o dispositivo móvel. Ainda não há muito componentes visuais no *.NET Compact Framework*. Com isso, foi necessária a criação de um componente para deixar o usuário selecionar o diretório que deseja salvar o arquivo e também para abrir uma janela para selecionar um arquivo a ser enviado.

Outra dificuldade encontrada foi no momento de criar a rotina para exibir a árvore de arquivos e pastas lidas do servidor para o dispositivo móvel. O componente *TreeView* do *.NET Compact Framework* possui diferenças e incompatibilidades com o mesmo componente da versão do *Framework* para aplicativos desktops. Com isso, foi necessária a criação de uma nova estrutura que armazenasse essas informações e posteriormente fazer a transferência para o dispositivo móvel.

As principais vantagens deste sistema são:

- a) os usuários com o dispositivos móveis podem ter acesso aos seus arquivos a partir de qualquer lugar. A distância geográfica não pode limitar o acesso;
- b) o uso compartilhado de arquivos por usuários remotos é controlado e facilitado, pois o sistema tem recursos de bloquear o acesso ao arquivo e registrar a leitura e gravação dos mesmo;
- c) com o uso de *Web Services*, este sistema pode ser integrado com outros aplicativos, *plug-ins* ou extensões para dispositivos móveis ou *desktops*.

No momento dos testes foi encontrada uma limitação durante o recebimento de

arquivos. Quando o arquivo a ser recebido é maior que 10 MB, o dispositivo móvel não consegue fazer o recebimento do *array* de *bytes*, devido a sua capacidade limitada em memória.

4.1 EXTENSÕES

Nesta seção são apresentadas sugestões de extensões e modificações para este trabalho, que estão descritas a seguir:

- a) implementar a interface visual do aplicativo no dispositivo móvel, adicionando ícones aos botões, mensagens com imagens e novos componentes visuais;
- b) criar rotina para fazer o tratamento durante o recebimento de arquivos grandes, fazendo com que o arquivo seja fragmentado durante a recepção no dispositivo móvel e unido as partes após o recebimento;
- c) permitir a abertura de arquivos diretamente pelo sistema no dispositivo móvel. Atualmente é necessário primeiro fazer o recebimento do arquivo e depois fazer a abertura pelo *Windows Mobile*;
- d) implementar segurança do sistema durante a comunicação com os *Web Services*, com o uso de certificados digitais, permitindo assim que os dados sejam criptografados através do protocolo *Hypertext Transfer Protocol Secure* (HTTPS).

REFERÊNCIAS BIBLIOGRÁFICAS

BURÉGIO, Vanilson A. A. **Desenvolvimento de aplicações para dispositivos móveis com .NET**. 2003. 69 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. Disponível em: <<http://www.cin.ufpe.br/~tg/2003-1/vaab.PDF>>. Acesso em: 15 mar. 2010.

CEMBRANELLI, Felipe. **ASP.NET: guia do desenvolvedor**. São Paulo: Novatec, 2003.

DOCTORSYS Tecnologia da Informação. **Dispositivos móveis**. São Paulo, 2004. Disponível em: <<http://www.doctorsys.com.br/dispmoveis.asp>>. Acesso em: 09 maio 2010.

FILE transfer protocol. In: WIKIPÉDIA, a enciclopedia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <http://pt.wikipedia.org/wiki/File_Transfer_Protocol>. Acesso em: 09 maio 2010.

GALVIN, Daleon. **Protótipo de sistema CRM para dispositivos móveis utilizando a tecnologia .NET**. 2004. 89 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

GRATISANUNCIOS. **Aumento no consumo de Smartphones no Brasil**. [S.l.], [2009]. Disponível em: <<http://www.gratisanuncios.com.br/Blog/aumento-no-consumo-de-smartphones-no-brasil>>. Acesso em: 10 mar. 2010.

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: 09 maio 2010.

JUNGES, Ivan C. **Software de controle de entregas usando dispositivos móveis e Web Services sobre a plataforma .NET**. 2006. 84 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MICROSOFT CORPORATION. **Microsoft .NET framework e aplicativos web**. Rio de Janeiro: Campus, 2001.

MIRANDA, Luiz Henrique. **Introdução ao mundo móvel**. Goiás, [2004]. Disponível em: <<http://www.devgoiania.net/artigos.aspx>>. Acesso em: 25 mar 2010.

NETTO, Max M. **Microsoft .NET compact framework: conheça a plataforma para dispositivos móveis criada pela Microsoft**. [S.l.], [2005]. Disponível em: <<http://www.linhadecodigo.com.br/Artigo.aspx?id=646>>. Acesso em: 15 mar. 2010.

_____. **Mobilidade e dispositivos móveis.** [S.l.], [2004]. Disponível em: <<http://www.linhadecodigo.com.br/Artigo.aspx?id=206>>. Acesso em: 15 mar. 2010.

PEKUS Consultoria e Desenvolvimento Ltda. **Dispositivos móveis.** São Paulo, 2002. Disponível em: <<http://www.pekus.com.br/palmtops.htm>>. Acesso em: 15 set. 2008.

RECKZIEGEL, Mauricio. **Entendendo os web services.** [São Paulo], 2006. Disponível em: <http://www.imasters.com.br/artigo/4245/webservices/entendendo_os_webservices>. Acesso em: 09 mar. 2010.

RICHTER, Jeffrey. **Applied Microsoft.NET framework programming.** Redmond: Microsoft Press, 2002.

SCHAEFER, Carine. **Protótipo de aplicativo para transmissão de dados a partir de dispositivos móveis aplicado a uma empresa de transportes.** 2004. 53 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SCRIBNER, Kenn; STIVER, Mark C. **Applies SOAP: implementing .NET XML Web Services.** Indianapolis: Sams, 2002.

SEELY, Scott. **SOAP: cross platform Web Services development using XML.** New Jersey: Prentice Hall, 2002.

SNELL, James; TIDWELL, Doug; KULCHENKO, Pavel. **Programing web services with SOAP.** Sebastopol: O'Reilly & Associates, 2002.

TELECOMWEB. **Mercado de dispositivos móveis cresce 45% em 12 meses.** [S.l.], 2004. Disponível em: <<http://www.telecomweb.com.br>>. Acesso em: 08 mar. 2010.

VENTURI, Eli. **Protótipo de um sistema para controle e monitoração residencial através de dispositivos móveis utilizando a plataforma .NET.** 2005. 68 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

WEB service. In: WIKIPÉDIA, a enciclopédia livre. [S.l.]: Wikimedia Foundation, 2006. Disponível em: <http://pt.wikipedia.org/wiki/Web_service>. Acesso em: 09 mar. 2010.