

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
II CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA JAVA E
DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

DEMIS LEANDRO DESTRO DA SILVA

DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA
CONDUÇÃO DE PROCESSOS LICITATÓRIOS DE MODALIDADE
PREGÃO ELETRÔNICO

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA

2014

DEMIS LEANDRO DESTRO DA SILVA

**DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA
CONDUÇÃO DE PROCESSOS LICITATÓRIOS DE MODALIDADE
PREGÃO ELETRÔNICO**

Monografia apresentada no II Curso de Especialização em Tecnologia Java e Desenvolvimento para Dispositivos Móveis do Departamento Acadêmico de Informática – DAINF – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do Título de Especialista.

Orientadora: Prof^a. Dra. Maria Cláudia F. P. Emer.

CURITIBA

2014

RESUMO

Destro da Silva, Demis. Desenvolvimento de um Aplicativo Android para Condução de Processos Licitatórios de Modalidade Pregão Eletrônico. 2014. Monografia (Especialização em Tecnologia Java e Desenvolvimento para Dispositivos Móveis) – Programa de Pós-Graduação em Tecnologia, Universidade Federal Tecnológica do Paraná – UTFR. Curitiba, 2014.

O acesso a *Tablets* e *Smartphones* está em ritmo cada vez maior. Com isso, foram surgindo diversos softwares para estes dispositivos, cada um direcionado para uma determinada finalidade. Entre estas finalidades, surgiu a idéia de desenvolver um projeto utilizando a linguagem *Java* e a plataforma *Android* para empresas licitantes conduzirem suas participações nos processos de licitação eletrônica por meio destes dispositivos. O objetivo do aplicativo proposto é auxiliar as empresas fornecedoras a conduzirem seus processos licitatórios de modalidade Pregão Eletrônico durante o envio de lances, utilizando um *Web Service .Net* como meio de comunicação com a base de dados. Neste trabalho foi realizado um estudo sobre as principais tecnologias e ferramentas que poderiam ser empregadas no desenvolvimento desta aplicação móvel e a implementação de uma ferramenta de condução de processos de licitação eletrônica. Desta forma, é esperado um aumento na participação dos fornecedores nos processos de licitação eletrônica, evitando a perda de oportunidades que possam adquirir e tornando-as independentes de suas infra-estruturas de comunicação.

Palavras-chave: Android, Tecnologia Java, Web Services, .Net Framework, Processos, Pregão Eletrônico.

ABSTRACT

Destro da Silva, Demis. Developing an Application Android Procedures for Conducting Bidding Mode Electronic Trading. 2014. Monografia (Especialização em Tecnologia Java e Desenvolvimento para Dispositivos Móveis) – Programa de Pós-Graduação em Tecnologia, Universidade Federal Tecnológica do Paraná – UTFR. Curitiba, 2014.

Access to Tablets and Smartphones is growing. Thus arose several software for these devices, each directed to a particular purpose. Among these, arose the idea of developing a project using Java and the Android platform for bidding companies conduct their participation in the processes of electronic bidding through these devices. The goal of the proposed application is to assist suppliers to conduct their bidding processes Electronic Trading mode when sending bids, using a NET Web Service as a means of communication with the database. In this paper a study about the key technologies and tools that could be employed in the development of mobile application and implementation of a tool for conducting electronic bidding processes was conducted. Thus, it is expected to increase the participation of suppliers in the electronic bidding processes, avoiding the loss of opportunities to acquire and making them independent of their communications infrastructure.

Keywords: Android, Java Technology, Web Services, .Net Framework, Processes, Electronic Trading.

DEDICATÓRIA

Dedico este trabalho a todos aqueles que contribuíram para a elaboração deste, principalmente à minha orientadora Prof^a. Dra. Maria Cláudia Emer no apoio pelos caminhos da pesquisa com paciência e maestria, aos meus familiares que contribuíram com sua força e motivação e à minha querida esposa e filho por toda a paciência, compreensão, carinho e amor por me ajudarem muitas vezes a achar soluções quando elas pareciam não aparecer, muito obrigado a todos.

LISTA DE FIGURAS

FIGURA 1 – ARQUITETURA DO ANDROID.....	09
FIGURA 2 – FORMULÁRIO DO SDK ANDROID.....	11
FIGURA 3 – ESTRUTURA DE UM PROJETO ANDROID GERADO PELA IDE ECLIPSE SDK.....	12
FIGURA 4 – HIERARQUIA DE COMPONENTES DE INTERFACE	13
FIGURA 5 – CÓDIGO FONTE DE UMA ACTIVITY GERADO AUTOMATICAMENTE	14
FIGURA 6 – ESTRUTURA DE UMA MENSAGEM SOAP	16
FIGURA 7 – ARQUITETURA WEB SERVICE BASEADO EM SOAP.....	17
FIGURA 8 – TROCA DE MENSAGENS ENTRE CLIENTE E SERVIDOR UTILIZANDO SOAP.....	18
FIGURA 9 – CONTEÚDO DO ARQUIVO .ASMX (DIRETIVA WEBSERVICE .NET).....	20
FIGURA 10 – ARQUIVO CODE-BEHIND WEBSERVICE.CS	20
FIGURA 11 – FLUXO DE OPERAÇÕES DO SISTEMA	23
FIGURA 12 – ARQUITETURA DE HARDWARE DO SISTEMA	24
FIGURA 13 – VERSÕES DO ANDROID HOMOLOGADAS	25
FIGURA 14 – MODELO DE CASO DE USO DO SISTEMA	28
FIGURA 15 – <i>USE CASE</i> DE USO AUTENTICAÇÃO FORNECEDOR	29
FIGURA 16 – MODELO DE CLASSE AUTENTICAÇÃO FORNECEDOR.....	30
FIGURA 17 – DIAGRAMA DE SEQUENCIA AUTENTICAÇÃO DO FORNECEDOR.....	31
FIGURA 18 – TELA DE AUTENTICAÇÃO E SELEÇÃO DO FORNECEDOR.....	32
FIGURA 19 – <i>USE CASE</i> MANTER PARTICIPAÇÃO DO PROCESSO	33
FIGURA 20 – MODELO DE CLASSE MANTER ACEITE DE PARTICIPAÇÃO DO PROCESSO.....	35
FIGURA 21 – DIAGRAMA DE SEQUENCIA MANTER PARTICIPAÇÃO DO PROCESSO.....	35
FIGURA 22 – TELA DE PARTICIPAÇÃO DO PROCESSO LICITATÓRIO	36
FIGURA 23 – <i>USE CASE</i> MANTER PROPOSTAS	37
FIGURA 24 – MODELO DE CLASSE MANTER PROPOSTAS.....	39
FIGURA 25 – DIAGRAMA DE SEQUENCIA MANTER PROPOSTAS	40
FIGURA 26 – TELA DE ENVIO DE PROPOSTAS	41
FIGURA 27 – <i>USE CASE</i> MANTER LANCES	42
FIGURA 28 – MODELO DE CLASSE MANTER LANCES.....	44
FIGURA 29 – DIAGRAMA DE SEQUENCIA MANTER LANCES	45
FIGURA 30 – TELA DE ENVIO DE LANCES	46
FIGURA 31 – MÉTODO CHAMADO VIA HTTP.....	47
FIGURA 32 – PÁGINA DE TESTES DA EXECUÇÃO DO MÉTODO	47
FIGURA 33 – PÁGINA DE RESULTADOS DA EXECUÇÃO DO MÉTODO	48
FIGURA 34 – MÉTODO QUE RETORNA A LISTA DE FORNECEDORES.....	49
FIGURA 35 – FUNCIONAMENTO DA TELA ENVIO DE LANCES.....	50

LISTA DE REDUÇÕES

SIGLAS

ADT	Android Developer Tools
API	Application Program Interface
GPS	Global Positioning System
GSM	Global System Mobile
GPRS	General Packet Radio Service
HTTP	Hyper Text Transport Protocol
IDE	Integrated Development Environment
PE	Pregão Eletrônico
PMC	Prefeitura Municipal de Curitiba
PMI	Project Management Institute
SQL	Structured Query Language
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
UMTS	Universal Mobile Telecommunications System
XML	Extensible Markup Language
WSDL	Web Service Description Language
W3C	World Wide Web Consortium

SUMÁRIO

1 INTRODUÇÃO	1
1.1 ESTUDO DO DOMÍNIO DO PROBLEMA	1
1.2 OBJETIVOS	3
1.2.1 OBJETIVO GERAL	3
1.2.2 OBJETIVOS ESPECÍFICOS	3
1.3 MOTIVAÇÃO E JUSTIFICATIVA	3
1.4 ESCOPO	5
1.5 ESTRUTURA DO TRABALHO	5
2 TECNOLOGIAS APLICADAS	7
2.1 CARACTERÍSTICAS DO PROJETO MOBILE	7
2.2 ANDROID	8
2.2.1 SDK ANDROID	11
2.2.2 ECLIPSE ADT	14
2.3 WEB SERVICE	14
2.3.1 WEB SERVICE DESCRIPTION LANGUAGE - WSDL	17
2.3.2 SERVIDOR WEB SERVICE	17
2.3.3 KSOAP2	18
2.4 BANCO DE DADOS – SQLITE	21
3 DESENVOLVIMENTO DO PROJETO	22
3.1 DESCRIÇÃO DO PROBLEMA	22
3.2 ANÁLISE DE REQUISITOS	24
3.2.1 REQUISITOS NÃO-FUNCIONAIS	24
3.2.2 REQUISITOS FUNCIONAIS	26
3.3 CASOS DE USO	26
3.3.1 AUTENTICAÇÃO DO FORNECEDOR	28
3.3.2 MANTER ACEITE DE PARTICIPAÇÃO DA SELEÇÃO DO PROCESSO DE MODALIDADE PREGÃO ELETRÔNICO (PE)	32
3.3.3 MANTER PROPOSTAS	36
3.3.4 MANTER SESSÃO DE LANCES	41
3.3.5 MANTER OPERAÇÕES <i>WEB SERVICE</i>	46
3.4 VALIDAÇÃO DA PROPOSTA	49
3.5 RESULTADOS OBTIDOS	51
4 CONCLUSÕES E TRABALHOS FUTUROS	52
REFERÊNCIAS BIBLIOGRÁFICAS	55

1 INTRODUÇÃO

1.1 ESTUDO DO DOMÍNIO DO PROBLEMA

O desenvolvimento de aplicativos móveis está ganhando cada vez mais espaço no dia a dia das organizações que visam facilitar o trabalho de seus funcionários para agilizar os processos para seus clientes.

A mobilidade está cada vez mais integrada ao mundo corporativo e com o avanço digital, que promoveu uma verdadeira migração para as tecnologias móveis, como *tablets* e *smartphones*, os aplicativos vêm ganhando cada vez mais espaço dentro das empresas, facilitando o desenvolvimento do trabalho em diversas áreas e agilizando a entrega de resultados. A computação móvel pode ser aplicada a praticamente todas as atividades e/ou segmentos de negócio que lidam com informações, permitindo ao usuário se comunicar a qualquer momento e em qualquer lugar (Stefanini, 2014).

Smartphones e Tablets com suporte a *Widgets*, Banda Larga Móvel e *Touchscreen*, fazem crescer a demanda por novos tipos de serviço e aplicações nas mais diversas áreas. Com o crescimento tecnológico do mercado móvel, o empresário está proporcionando tudo isso ao seu cliente. Atualmente, com o crescimento no uso de tecnologias móveis, a facilidade de uma pessoa obter um celular (*Smartphone*) de qualidade, faz com que sejam criados serviços que podem ser implementados de forma bastante abrangente (LARA, Guilherme, *Mobile Marketing*, 2009).

Com base neste cenário fica claro que um dos principais viabilizadores de uma empresa que tem foco em uma arquitetura baseada em tempo real é a mobilidade: ao tornar móveis as informações de qualquer natureza, rompendo os laços que prendíamos usuários à infra-estrutura da empresa, cria-se a possibilidade de trabalhar em tempo real. Um ambiente móvel permite a um colaborador manter-se sempre conectado às aplicações de negócio, em relação aos indicadores mais adequados as suas atividades. Além de permanecer sempre conectado, o colaborador móvel tem a oportunidade de usar a tecnologia para aplicações de coleta de dados e/ou visualização de dados, como por exemplo, o acesso móvel a sistemas que incluem serviços

como consulta bancárias, previsão do tempo, notícias, redes sociais e localização em sistema *GPS* ou *Global Positioning System* (Sistema de Posicionamento Global), indicando em tempo real a sua exata localização (PINGARILHO, CARLOS - *Promon Business & Technology Review*,).

Entre esses meios de acesso a informações, estão também as tecnologias utilizadas para gerar e manter estas informações, sendo a linguagem de desenvolvimento Java, tecnologia *Asp.Net*, sistemas de gerenciamento de banco de dados (*MySQL*, *SQL Server*, *SQLite*), dentre outros. O conjunto destas tecnologias envolvidas proporcionam diversas formas de implementação entre diferentes soluções para acesso às informações (ROGERS e MEIKE, 2009).

Em virtude do aumento do uso das tecnologias móveis e da necessidade de disponibilizar soluções diversas para o acesso a informação, surge também a necessidade de serviços a serem disponibilizados para empresas que pretendem ter acessos a processos de compra de bens na modalidade de PE – Pregão Eletrônico - e também na contratação de serviços de forma mais simples, rápida e econômica.

O Pregão na forma eletrônica, já está consolidado como o melhor modelo de aquisições governamentais, sendo a eficiência, a celeridade, transparência e a economia características marcantes que se destacam em relação às modalidades licitatórias tradicionais, previstas na Carta de Licitação (Lei 8.666, 21 de junho de 1993).

A palavra Pregão provém do latim *praiconium*, do verbo *praeconari*, que significa apregoar, proclamar notícia. Refere-se à proclamação de lances em alta voz nas hastas públicas. Ainda é utilizado para designar as sessões das bolsas de valores, em que são negociadas ações abertamente, em público. Portanto, desde logo se percebe que a palavra *Pregão*, num sentido comum, tem haver com algo que é dito em alta voz, direcionado ao público geral (BOSELI, 2009).

O Pregão é uma modalidade de licitação para aquisição de bens e serviços comuns no âmbito da União, Estados e Municípios. Garante economia imediata nas aquisições de bens e serviços comuns e permite ainda agilidade nas aquisições, pois desburocratiza os procedimentos realizados durante as etapas da licitação (LEI 10.520/02).

O Pregão, na forma eletrônica, como modalidade de licitação do tipo menor preço item, realizar-se-á quando a disputa pelo fornecimento de bens ou serviços comuns for feita à distância, em sessão pública, por meio de um sistema que promova a comunicação pela internet (DEC. 5.450/05).

Nesse contexto, a idéia deste trabalho é a criação de um aplicativo móvel de condução de licitação eletrônica para os usuários fornecedores participantes (habilitados no Pregão) via *Smartphones*, proporcionando as empresas fornecedoras mais um meio de participação aos processos eletrônicos.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Desenvolver um aplicativo cliente *Android* para que os fornecedores participantes de licitação eletrônica na modalidade de Pregão Eletrônico possam conduzir seus processos.

1.2.2 OBJETIVOS ESPECÍFICOS

- Auxiliar, facilitar e agilizar a condução dos processos pelos fornecedores participantes.
- Desenvolver um aplicativo móvel cliente para usuários fornecedores licitantes em processos de Pregão Eletrônico.
- Desenvolver um *Web Service* para comunicação entre o aplicativo móvel e a base de dados.
- Validar o aplicativo móvel.

1.3 MOTIVAÇÃO E JUSTIFICATIVA

Os benefícios provenientes da tecnologia móvel oferecem mais recursos para a melhoria do processo de negócio, redução de custos e diminui a latência de comunicação (Yeung 2008). O dispositivo móvel se tornou uma ferramenta tão eficiente que deixou de ser utilizado apenas para ligações ou envio de SMS. Segundo LEMOS (2004, p.6) “O celular passa a ser um ‘teletudo’ um

equipamento que é ao mesmo tempo telefone, máquina fotográfica, televisão, cinema, receptor de informações jornalísticas, difusor de e-mails e SMS7, WAP8, atualizador de sites (*moblogs*), localizados por GPS, tocador de música (*mp3* e outros formatos), etc.”.

Como justificativa para o desenvolvimento do trabalho pode-se salientar que será possível atender a um grande número de fornecedores participantes nos processos de modalidade Pregão Eletrônico. Também há a necessidade de dar vazão a um alto volume de informações pertinentes aos processos de licitação eletrônica de critério de julgamento menor preço item (disputa individual de um determinado item de processo).

Além disso, o desenvolvimento de um aplicativo móvel cliente para sistemas *Android* torna as empresas fornecedoras independentes de suas infraestruturas atuais de comunicação, como por exemplo, a internet, proporcionando mais um meio de acesso para condução de processos licitatórios na forma eletrônica. O acesso por meio do dispositivo móvel (*Smartphones*) utiliza como infraestrutura o acesso a rede móvel 2G (*Internet Wap*), 3G (*GSM/GPRS/EDGE*) e 4G (*UMTS/GSM/LTE*). Esta nova forma de acesso aos Pregões Eletrônicos fará com que as empresas fornecedoras possam planejar o gerenciamento de suas disputas nos processos a qual tenham acesso, como por exemplo, monitorando e conduzindo suas disputas de qualquer lugar.

As vantagens aplicadas ao problema proposto são evidenciadas da seguinte forma:

- Melhorar a condução de processos de modalidade Pregão Eletrônico, com critério de julgamento menor preço item por meio do *Smartphone* (transmissão *online*);
- Disponibilizar as informações de forma rápida e eficiente aumentando a participação das empresas fornecedoras na condução dos processos licitatórios de modalidade Pregão Eletrônico;
- Evitar perdas de oportunidades para as empresas fornecedoras por não terem acesso ao Portal de Compras *Web* de onde possam estar.

A utilização dependerá apenas de um ambiente que tenha cobertura de sua operadora ou dispositivo com suporte a rede *Wireless*.

1.4 ESCOPO

O escopo de desenvolvimento do aplicativo cliente *Android* para que as empresas fornecedoras participem de seus processos licitatórios de modalidade Pregão Eletrônico com critério de julgamento menor preço item, contém os seguintes módulos:

- 1-) Autenticação de Usuários
- 2-) Listagem de Fornecedores;
- 3-) Listagem de Processos;
- 4-) Termo de Aceite de Participação;
- 5-) Envio de Propostas;
- 6-) Envio de Lances;

Para que seja possível a comunicação entre o aplicativo móvel e a retaguarda, em que as informações estarão armazenadas, será utilizado um serviço *Web Service* desenvolvido em *.Net*, que integra a aplicação *Android* por meio do padrão *KSOAP2*, que serão detalhados mais adiante. Além desta integração, a aplicação *Android* poderá contar com armazenamento temporário de informações no próprio dispositivo com uso do *SQLite*.

1.5 ESTRUTURA DO TRABALHO

O trabalho está organizado da seguinte forma:

1º Capítulo: Neste capítulo são apresentados os conceitos relacionados aos dispositivos móveis, suas características e aplicações, bem como, o problema proposto, suas justificativas para a elaboração do projeto e o seu escopo.

2º Capítulo: Neste capítulo são apresentadas as tecnologias utilizadas para o desenvolvimento da solução do projeto proposto, no que se refere à sua análise e implementação.

3º Capítulo: Neste capítulo é apresentado o desenvolvimento do projeto e tecnologias utilizadas, bem como, a validação do projeto.

4º Capítulo: Neste capítulo são apresentadas as conclusões do trabalho, suas contribuições e sugestões para trabalhos futuros.

2 TECNOLOGIAS APLICADAS

A finalidade deste capítulo é descrever algumas características dos sistemas móveis e das tecnologias utilizadas nos processos envolvidos para o desenvolvimento do projeto. As ferramentas utilizadas para o desenvolvimento do aplicativo móvel também são abordadas.

2.1 CARACTERÍSTICAS DO PROJETO MOBILE

Um erro normal de quem deseja um aplicativo para *smartphones* é querer portar tudo que se tem na web para o aparelho. Com os novos aparelhos com excelente capacidade de processamento e armazenamento, os desenvolvedores atuais acham natural fazer um aplicativo móvel que tenha todas as funcionalidades que uma aplicação web pode ter (JÚNIOR, 2011).

O problema desta abordagem é que, por mais desempenho que o aparelho tenha, ele ainda tem várias limitações se comparado a um computador normal: tamanho de tela, quantidade de informações que pode mostrar por vez, teclado, enfim, uma série de restrições que, se não consideradas, podem fazer o orçamento do projeto aumentar (JÚNIOR, 2011).

Portanto, é necessário observar a definição de dispositivos móveis e suas características antes de entender as diferenças entre suas aplicações. Firtman (2010) define dispositivo móvel como sendo aquele que possui as seguintes características:

- **Portátil:** que possa ser carregado sem muito esforço, como um molho de chaves ou uma carteira.
- **Pessoal:** cada pessoa tem seu próprio dispositivo, com suas customizações e normalmente não se compartilha sua utilização.
- **Companheiro:** É possível estar com o dispositivo o tempo todo.
- **Fácil de usar:** No sentido de condições de uso, como por exemplo, não é necessário utilizar o dispositivo sentado, ou num lugar com ótima iluminação, ou ter que esperar dois minutos para que o sistema esteja pronto para uso.
- **Conectado:** Deve ser possível conectar na internet quando necessário.

Por conta do avanço da plataforma *mobile*, é normal que as empresas queiram possuir aplicativos para atender em suas tarefas diárias. A chave aqui é sempre ter em mente o que o *smartphone* oferece: mobilidade, informação em qualquer lugar.

Assim sendo, para o desenvolvimento de um aplicativo móvel é necessário considerar: o que é importante para o usuário de um aplicativo móvel ter na rua, na fila do banco e quando ele não tem um computador disponível. Com essa pergunta respondida podem ser evitadas funcionalidades desnecessárias, bem como, podem ser reduzidos custos e prazos (JÚNIOR, 2011).

Segundo Pinheiro (2004), alguns pontos devem ser considerados para projetos de sistemas móveis: características do ambiente, energia, interface, processamento e segurança.

Uma aplicação móvel utiliza uma arquitetura baseada em “*Native apps*”, isto é, uma aplicação específica para uma determinada plataforma. Esta aplicação é desenvolvida usando código nativo “*SDKs*” e tem um código relacionado com a plataforma sobre a qual foi escrita (NOKIA *Developer*, 2013).

Existem diversas plataformas de dispositivos móveis disponíveis no mercado, entre elas, as mais populares são *Android* e *iOS* (PUC-RIO, Certificação Digital Nº 1012653/CA). Cada uma destas plataformas possui um *SDK* próprio com a sua linguagem de programação e arquitetura bem definidas. Neste trabalho, a plataforma *Android* é utilizada para o desenvolvimento do aplicativo móvel por ser de código aberto e baseada no sistema operacional *Linux*.

2.2 ANDROID

O *Android* é um sistema operacional de código aberto construído para dispositivos móveis. Desenvolvido com base no *Linux*, o *Android* usa *Java* como linguagem padrão de desenvolvimento, mas *C/C++* pode ser usado para desenvolver aplicativos nativos usando o *NDK* (*Native Development Kit*). Pertence ao Google e a *Open Handset Alliance* (*OHA*), uma aliança voltada

para dispositivos móveis open source, composta com aproximadamente 30 empresas (Intel, Motorola, Samsung e outros) (FARIA, 2011).

Lecheta (2009) comenta as características do sistema operacional para aplicativos móveis criado pela Google, “Inclui um sistema operacional baseado no Linux e diversas aplicações, com uma rica interface gráfica, um browser para navegar na Internet, integração com o Google *Maps*, suporte a multimídia, *GPS*, banco de dados integrado, jogos em 3D, entre outras.”.

A arquitetura do *Android*, conforme a Figura 1 é composta de cinco camadas.

A camada de Aplicação contém todos os aplicativos que são acessados pelo usuário. Estes aplicativos são escritos em *Java* e rodam em máquina virtual *Dalvik*, criada para rodar as aplicações de forma otimizada nos dispositivos móveis.

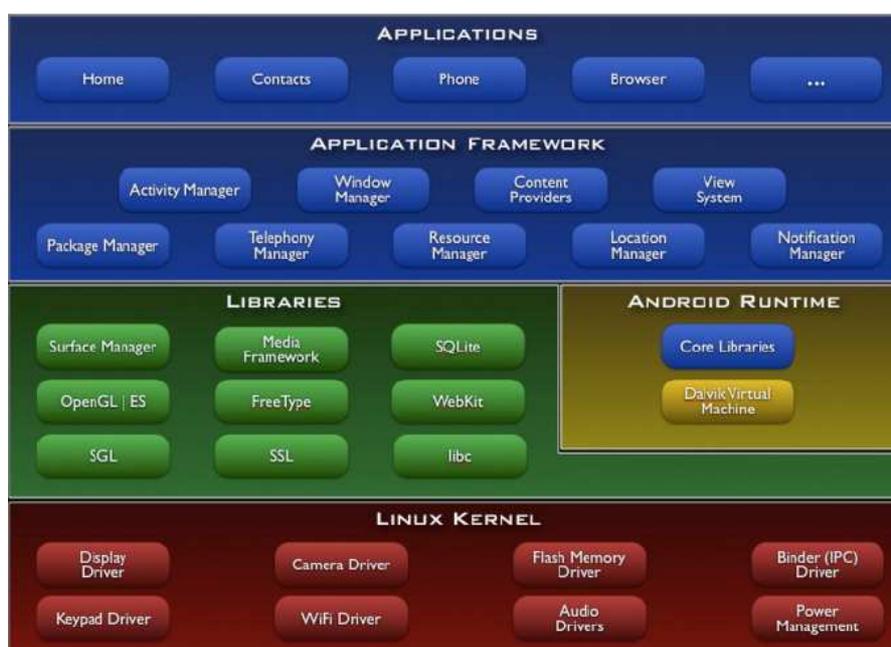


Figura 1 – Arquitetura do *Android*. Fonte: Faria (2011).

Na camada de Framework estão disponíveis todas as *APIs* e recursos utilizados pelos aplicativos. Dentre eles, podemos citar os controles de telas, provedores de serviço (*Content Providers*), os elementos visuais como botões, caixas de texto, etc. Outro importante serviço desta camada é o gerenciador de atividades que controla o ciclo de vida das aplicações.

A camada de Bibliotecas contém todas as bibliotecas *C/C++* que são utilizadas pelo sistema, entre elas a biblioteca *C* padrão, bibliotecas de

multimídia, navegação web, acesso a banco de dados *SQLite*, renderização 3D (*OpenGL ES*) e outras.

Usando a versão 2.6 do Kernel do Linux, a camada Linux Kernel é responsável pelos principais serviços do sistema operacional, tais como, comunicação com o hardware e gerenciamento de memória, processos e energia.

O *Android* possui algumas variantes de versões. A versão mais recente é a versão 4.4 (*KitKat*). Antes da versão 4.0, havia distinção entre versões para *tablets* e *smartphones* com diferenças significativas no *layout* visual dos aplicativos de cada uma das versões.

A arquitetura das aplicações *Android* é aplicada ao reuso de componentes. Sendo assim, os desenvolvedores possuem acesso irrestrito às mesmas *APIs* usadas pelas aplicações centrais, permitindo que componentes de aplicação possam ser substituídos pelo próprio usuário, como por exemplo, o aplicativo de envio de *SMS*, que pode ser substituído por qualquer outro aplicativo que tenha a mesma finalidade. Referente à questão do reuso, podem ser destacados dois componentes importantes:

1. Atividade (*Activity*): São trechos de códigos independentes, executados de acordo com a necessidade da aplicação. Geralmente são associadas a uma tela de entrada ou listagem de informações.
2. Intenções (*Intents*): Permitem que uma aplicação selecione uma atividade com base na ação desejada pelo usuário e pelas informações que operam. Esse recurso possibilita, por exemplo, que ocorra a troca de dados entre atividades diferentes, sendo uma forma de comunicação entre processos.

Dada a diferença de linguagem, arquitetura e versionamento entre as plataformas, desenvolver um aplicativo para cada plataforma demandaria muito esforço, então optou-se por utilizar o *SDK* do *Android*.

2.2.1 SDK ANDROID

Os *SDKs* (*Software Development kit*) surgem como uma alternativa que permite o uso de uma linguagem em comum entre diversas plataformas (PUC-Rio – Certificação Digital Nº 1012653/CA). Os principais fatores que influenciaram na escolha do *SDK* foram:

1. As plataformas móveis que são suportadas.
2. A linguagem de programação que será usada para desenvolvimento.
3. Flexibilidade de estender e criar novas funcionalidades.
4. Comunidade de desenvolvedores para discussão.
5. Custo de licença do *SDK*.

Como o objetivo do framework é permitir que cada aplicação defina a sua própria interface, o *SDK Android* (kit de desenvolvimento), inclui documentação, código e utilitários para que os programadores consigam desenvolver as suas aplicações de acordo com um padrão de desenvolvimento para o sistema operativo em questão (*ANDROIDZ - FORUM, 2013*). O *SDK Android* também proporciona ferramentas e chamadas via *API* na linguagem Java. Resumindo, são programas para celulares com um sistema operacional, *middleware* e interface. A Figura 2 apresenta o *SDK Android*, contendo seus principais componentes.

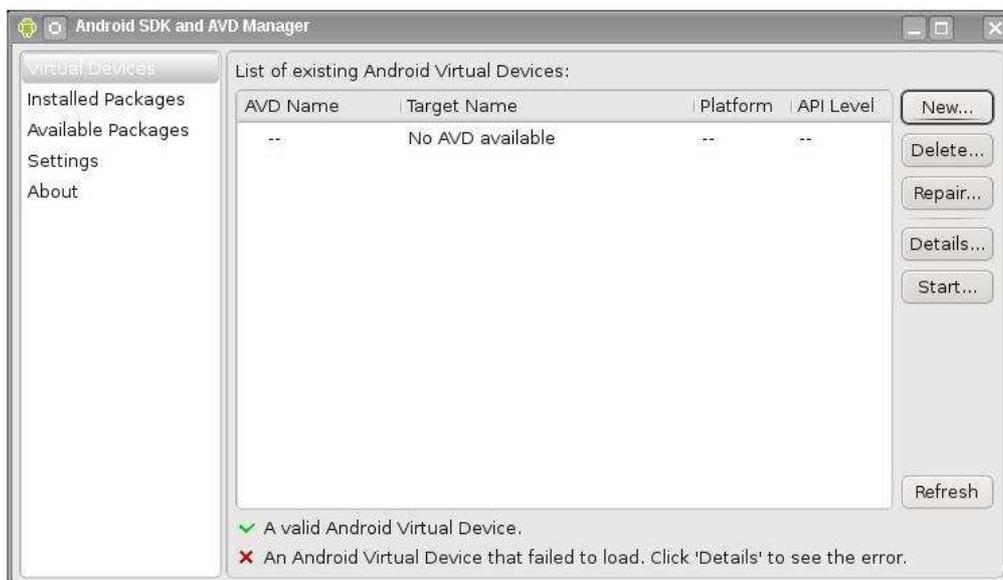


Figura 2 – Formulário do SDK Android. Fonte: Faria (2011).

Os principais recursos desta plataforma *mobile* são a máquina virtual otimizada e o *plugin* para o Eclipse (*ADT*). Vasconcelos (2011 b) explica que “o *Android* automaticamente verifica seus *resources* e ajusta de acordo com as configurações fornecidas pelo aparelho”. A Figura 3 apresenta a estrutura de um projeto *Android* gerado pela *IDE Eclipse ADT SDK*.



Figura 3 – Estrutura de um projeto *Android* gerado pela IDE Eclipse SDK.

A estrutura apresentada na Figura 3, segundo Vasconcelos (2011 b) é definida como:

1. A pasta *src*, na qual fica o código fonte da aplicação.
2. A pasta *gen* (*Generated source folder*) possui somente uma única classe gerada automaticamente pelo *plugin* do eclipse chamada ‘R’ que contém os ids para todos os itens da pasta *res*, para que seja possível usar todos os *resources* da aplicação.
3. Na pasta *res* ficam os *resources* da aplicação, em *res/drawable* ficam as imagens, na qual a divisão entre *hdpi* (*High dpi*), *mdpi* (*Medium dpi*), *ldpi* (*Low dpi*) é usada para guardar as imagens em resoluções diferentes, em *res/layout* ficam os arquivos *XML* que representam o *layout* das

Activities, e em *res/values* ficam os *XML* que guardam *Strings* que podem ser usadas na aplicação.

4. Na pasta *assets* ficam arquivos que podem ser abertos por *stream* dentro da aplicação.

Segundo Rabelo (2009), “Em *Android*, todos os componentes de interface gráfica são representados por subclasses de *android.view.View*, que representam os componentes gráficos (os chamados *widgets*) e a classe *android.view.ViewGroup*, que representa um *container* de *Views* e também *ViewGroups*”, conforme Figura 4.

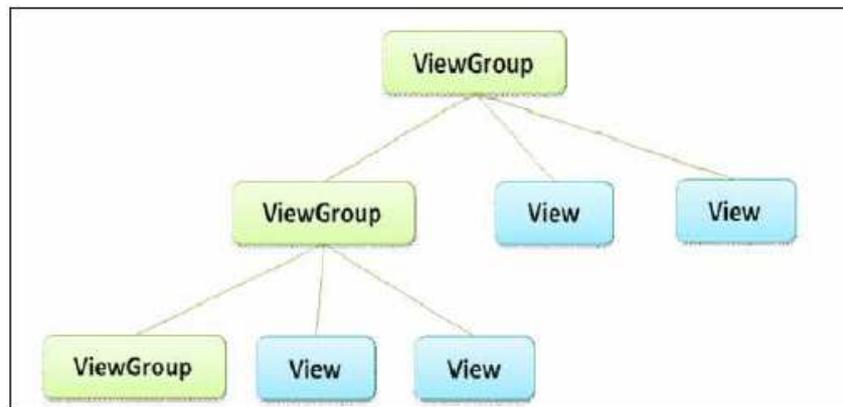
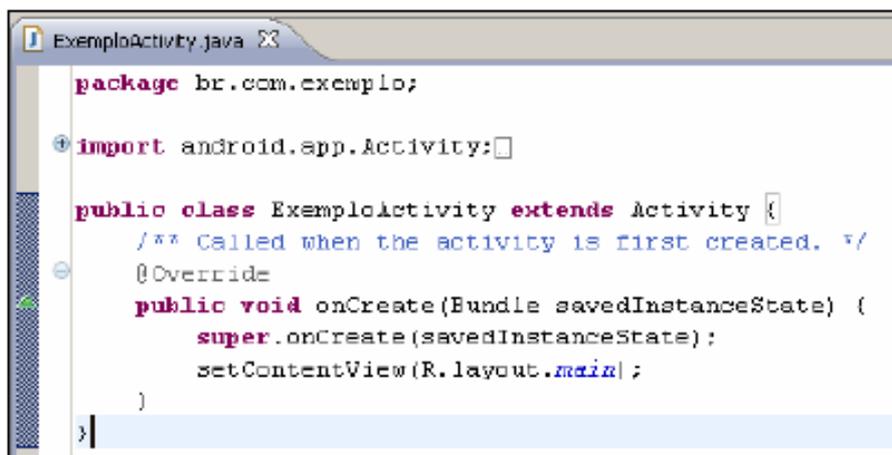


Figura 4 – Hierarquia de Componentes de Interface. Fonte: RABELO (2009).

Quando um projeto é criado por meio do Eclipse utilizando o *Android SDK*, é gerada uma classe *Activity*. A Figura 5 apresenta o código fonte gerado na classe *Activity*.



```
ExemploActivity.java
package br.com.exemplo;

import android.app.Activity;

public class ExemploActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Figura 5 – Código fonte de uma Activity gerado automaticamente.

2.2.2 ECLIPSE ADT

O Eclipse é uma ferramenta de integração da linguagem de programação *Java*. Nele é também possível integrar o *Android SDK* para o desenvolvimento de aplicações *Android*. O sistema *Android* disponibiliza um *plugin* para o Eclipse chamado *ADT (Android Development Tools)*. Com ele é possível executar uma aplicação *Android* usando um dispositivo virtual, que nada mais é um emulador. Este dispositivo virtual poderá emular um celular com diferentes versões do *Android*, testando o comportamento de diferentes aplicações nas diversas plataformas *Android*. O *ADT* estende os recursos do Eclipse para permitir que o desenvolvedor possa configurar rapidamente projetos *Android*, criar aplicações, adicionar componentes baseados na *API* do framework *Android*, depurar aplicações usando as ferramentas do *SDK* e até exportar projeto para sua distribuição (FERNANDES, Maicris, 2013).

Neste projeto, a aplicação cliente que acessa o *Web Service*, possui rotinas de código implementadas para a plataforma *Android SDK* utilizando a biblioteca para acesso ao *Web Service .Net* chamada *KSOAP2*.

2.3 WEB SERVICE

A principal característica dos aplicativos móveis é que eles normalmente precisam se conectar aos aplicativos das organizações, que ficam em uma estrutura fixa e que contém as informações da solução em questão. A maneira

mais utilizada para a integração de dados entre aplicativos de plataformas diferentes são os *Web Services*, que tem a possibilidade de transmitir dados com padrões específicos que qualquer plataforma pode utilizar (KNUTSEN, 2009).

Web Services são conjuntos de métodos e ferramentas presentes em uma determinada máquina, que podem ser acessados facilmente por aplicações presentes em outras máquinas, conforme Basiura *et al.* (2001), Wigley & Whellwright (2003) e MSDN (2005).

Web Services permitem que todas as transferências sejam realizadas no padrão *XML*, o que é bastante completo e de fácil aproveitamento por aplicações externas. Têm custo reduzido, pois podem ser implementados com ferramentas baratas, e não são uma tecnologia proprietária. Em contrapartida, aumentam a quantidade de tráfego transmitido na conexão, e causam um *overhead* adicional no processamento das máquinas, devido a vários clientes acessarem o mesmo *Web Service* simultaneamente (BASIURA *et al.*, 2001).

Kopack (2003) define *Web Service* como um sistema que pode ser acessado remotamente, e semelhante a sites *Web* ele é identificado na rede por uma *URL* e fornece ao cliente uma resposta a uma requisição. De acordo com a especificação do protocolo *SOAP*, tanto a requisição quanto a resposta, são formadas por um arquivo *XML*.

O *SOAP*, ou protocolo de acesso simples a objetos, habilita o transporte das requisições e respostas do *Web Service* por meio de informações encapsuladas num envelope e que são escritas em *XML* (BENZ e DURANT 2003). O uso deste formato visa atender a necessidade de trafegar as requisições com facilidade sobre *firewalls* das empresas e sobre a internet, ao mesmo tempo em que os dados fossem descritos de forma detalhada. Segundo Benz e Durant (2003), o protocolo (Figura 6) é composto por:

1. *SOAP Endpoint*: um ponto de extremidade que pode ser definido por uma *Uniform Resource Locators (URL)* padrão, com um *Uniform Resource Identifier (URI)* opcional e um *SOAP Action* (ação definida para o envelope *SOAP*);
2. Métodos (para chamadas via *HTTP POST*) para serem invocados em um *Web Service*;

3. Um cabeçalho *HTTP*, que indica o método invocado pela chamada;
4. O envelope *SOAP* descreve o conteúdo da mensagem e como utilizá-la;
5. *SOAP Encoding Rules*: Regras de codificação do *SOAP* que descrevem a serialização dos dados baseados em tipos de dados genéricos ou específicos da aplicação;
6. *Body*: O corpo do protocolo que contém a chamada ou mensagem de resposta que é descrita pelo envelope e formatada de acordo com as regras de codificação.

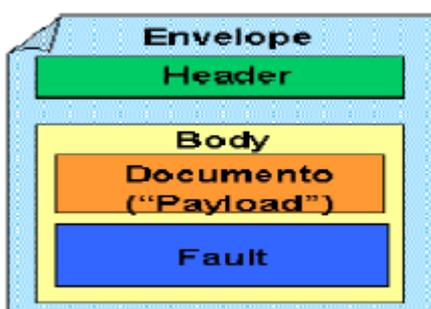


Figura 6 - Estrutura de uma mensagem SOAP. Fonte: Alonso e Bussler (2004).

Benz e Durant (2003), ainda destacam que assim como o cabeçalho *HTTP* numa requisição *SOAP* o envelope baseia-se em valores declarados no arquivo *WSDL* do *Web Service*, o qual menciona instruções de como estão estruturadas as respostas e requisições. Dessa forma, tanto o cliente como o provedor do serviço devem ter as ferramentas necessárias para receber o pacote (*package*) de acordo com a especificação do protocolo e passar a informação adiante para ser recuperada (Figura 7).

Segundo Kalin (2010), o *Web Service* trata-se de uma aplicação distribuída, tipicamente oferecida por meio de *HTTP* (*Hyper Text Transport Protocol*), em que seus componentes podem ser executados em dispositivos distintos.

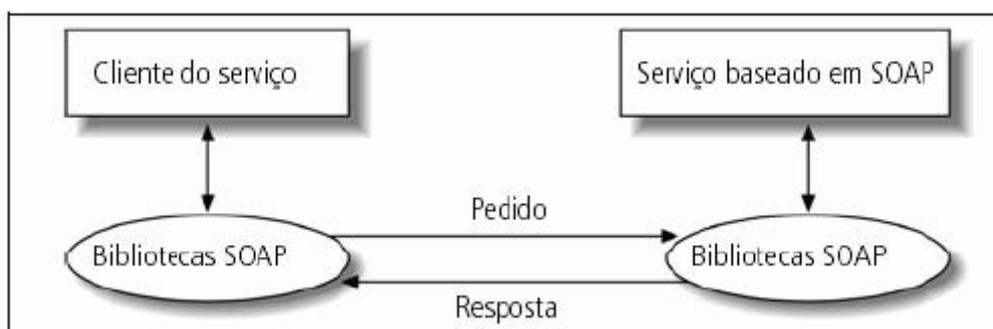


Figura 7 – Arquitetura *Web Service* baseado em SOAP. Fonte: Kalin (2010).

2.3.1 WEB SERVICE DESCRIPTION LANGUAGE - WSDL

Conforme definição de Cerami (2002) a linguagem conhecida pela sigla *WSDL* é uma especificação do que descreve os *Web Services* utilizando uma linguagem comum baseada em *XML*. *WSDL* representa um contrato entre o requerente do serviço e seu provedor e possibilita ao cliente localizar um *Web Service* e invocar qualquer de seus serviços expostos publicamente, bem como identificar os tipos de dados utilizados para as chamadas e respostas, obter informações sobre o protocolo utilizado para transporte e o endereço para localizar um determinado serviço (CERAMI, 2002).

Entre as vantagens oferecidas pelo uso do *WSDL* segundo Tidwell, Snell e Kulchenko (2001), é possível destacar a facilidade em escrever e manter os serviços, proporcionando uma abordagem mais estruturada para definição dos mesmos, a diminuição da quantidade de código a ser implementado pela aplicação cliente, além da rapidez para a codificação de alterações nos serviços publicados.

2.3.2 SERVIDOR WEB SERVICE

Segundo Vitor Fernando Pamplona (2010), para comunicar com o *Web Service*, é necessário a implementação do protocolo *SOAP* (*Simple Object Access Protocol*) definido no *W3C* (*World Wide Web Consortium*). Este protocolo é responsável pela independência que o *Web Service* precisa. Na Figura 8 encontra-se um diagrama mostrando as mensagens trocadas entre cliente e servidor em uma comunicação *SOAP*. Existem duas aplicações se

comunicando, um cliente *Wrapper* e um *Server Wrapper* que estão disponibilizando a transparência para as aplicações. Entre eles só trafega o *XML*, seguindo o protocolo *SOAP* sobre *HTTP*.

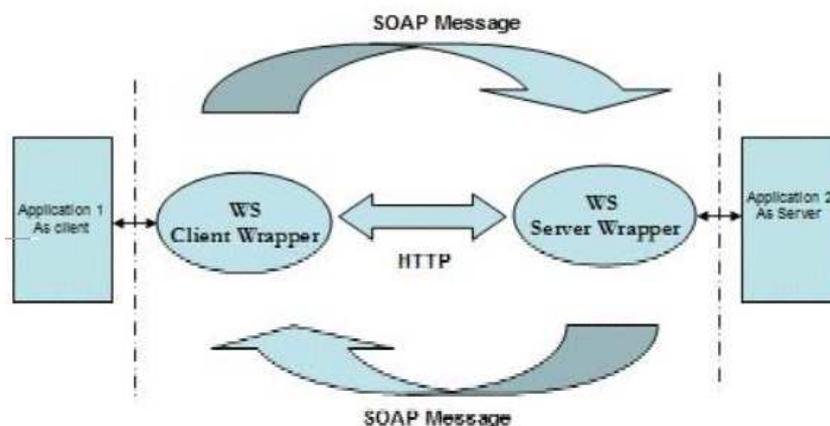


Figura 8 – Troca de mensagens entre cliente e servidor utilizando SOAP.

Fonte: PAMPLONA (2010).

Um *Web Service* será publicado, e para que os usuários possam utilizá-lo é necessário definir como ele é, como deve ser acessado, e que valores ele retornará. Estas definições são descritas em um arquivo *XML* de acordo com a padronização *Web Service Description Language (WSDL)*. Este arquivo deve ser construído para que os usuários do serviço possam entender o funcionamento do *Web Service* e, logicamente, será de acesso público (PAMPLONA, Vitor Fernando, 2010).

2.3.3 KSOAP2

O *Android SDK* oferece ao desenvolvedor um conjunto de bibliotecas que disponibilizam vários recursos para o desenvolvimento de sistemas *Android*, porém existem problemas que a plataforma ainda não resolve como acesso aos padrões das tecnologias *Web Services* (KNUTSEN, 2009; LECHETA, 2010).

Dessa forma, o *Android*, não possui nativamente suporte aos *Web Services* (KNUTSEN, 2009), sendo necessário escrever classes de comunicação para cada informação específica da aplicação, utilizando algum protocolo específico como o *TCP/IP*. Esta, contudo, é uma solução trabalhosa e que gera aumento da codificação, podendo até criar duplicação das regras de negócio programadas na aplicação. Para resolver este tipo de problema na

plataforma *Android* existem algumas bibliotecas que implementam suporte aos *Web Services*. Uma delas é a biblioteca *KSOAP2*, que encapsula métodos para a conexão, envio e recebimento de dados de *Web Service*. A biblioteca *KSOAP2* tem a característica de ser mais flexível quanto à geração do código de suporte, necessário ao cliente para acessar o *Web Service*.

O *KSOAP2* é originado do *KSOAP* que é uma biblioteca utilizada para acesso à *Web Service* por aplicações *Java ME* desde 2001, sem a necessidade de geração de código ou uso de *proxies* dinâmicos.

Rabello (2009) explica que o *Android* ainda não possui uma *API* embutida que trata especificamente de *Web Service*, para isso ele possui uma capacidade de integrar as mesmas bibliotecas de *JAVA SE* com a versão do *KSOAP2*.

Originalmente a implementação do *KSOAP2* para o *Android* surgiu no grupo de desenvolvedores do Google e várias versões do projeto tiveram sua manutenção encerrada em 2009/2010. Atualmente o projeto foi retomado e novas versões corretivas da biblioteca já foram lançadas, atualmente está na versão 2.5.5 (*KSOAP2-ANDROID*, 2014).

No desenvolvimento do presente trabalho, para que a aplicação cliente em *Android* possa acessar o *Web Service* foi utilizada a biblioteca “*Ksoap2-android-full-2.1.2.jar*”, que possui as classes necessárias para a comunicação com o *Web Service*.

A aplicação cliente *Android* deste projeto irá “consumir” um serviço *Web* desenvolvido em *ASP.NET* (plataforma Microsoft de desenvolvimento de aplicações *Web*), apresentando uma estrutura para facilitar na conectividade entre aplicações. O serviço *Web* desenvolvido por meio do *Framework ASP.NET* vai cuidar de fazer a conexão de baixo nível entre o *SOAP* e o *XML* transmitido. O servidor *Web Service* implementado em *ASP.NET* utiliza servidores *HTTP*, embora isso não seja uma restrição para o funcionamento do protocolo *SOAP*. As mensagens *SOAP* são documentos *XML* que aderem a uma especificação fornecida pelo órgão *W3C*.

No *ASP.NET* um *Web Service* é construído de um arquivo “.asmx” e uma classe *code-behind* que fornece a funcionalidade requerida. O conteúdo deste arquivo é apenas uma linha que contém a diretiva “*WebService*” idêntica a “*@Page*” usada em um arquivo “.aspx”, conforme Figura 9:

```
<%@ WebService Language="C#" CodeBehind="~/App_Code/WebService.cs" Class="WebService" %>
```

Figura 9 – Conteúdo do arquivo .asmx (Diretiva WebService .Net).

O arquivo “Code-Behind” *WebService.cs* é uma classe que herda de “*System.Web.Services.WebService*”. Na Figura 9 pode ser observado o atributo “*WebService*” na definição da classe, este atributo permite definir o *namespace* para o *Web Service*. Por padrão o *namespace* é definido para “*http://tempuri.org/*”, mas pode ser alterado para uma *URL* representando um projeto específico.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Web;
5 using System.Web.Services;
6
7 /// <summary>
8 /// Summary description for WebService
9 /// </summary>
10 [WebService(Namespace = "http://tempuri.org/")]
11 [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
12 // To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
13 // [System.Web.Script.Services.ScriptService]
14 public class WebService : System.Web.Services.WebService {
15
16     public WebService () {
17
18         //Uncomment the following line if using designed components
19         //InitializeComponent();
20     }
21
22     [WebMethod]
23     public string HelloWorld() {
24         return "Hello World";
25     }
26
27 }
```

Figura 10 – Arquivo Code-Behind WebService.cs.

O atributo “*WebServiceBinding*” indica que o *WebService* está em conformidade com a especificação *WS-I BP 1.1 – Web Service Interoperability Basic Profile Specification*. Esta especificação garante a integração entre todas as plataformas que venham utilizar este padrão, permitindo assim a troca de informações. No ambiente “.Net” o *XML* foi adotado por já ser um padrão da indústria e ser independente da plataforma adotada, sendo ainda controlado pelo *W3C* (DURÃES, 2012).

Como a aplicação cliente foi projetada e desenvolvida como um aplicativo para dispositivos móveis compatível com sistema operacional *Android*, e o mesmo realiza diversas consultas ao Serviço *Web Service*,

verificou-se a possibilidade de armazenar algumas destas informações no próprio dispositivo. Portanto, a tecnologia de banco de dados *SQLite*, poderá ser utilizada para realizar este armazenamento de informações retornadas pelo *Web Service*.

2.4 BANCO DE DADOS – SQLITE

Para armazenar os dados do aplicativo desenvolvido o *Android* possui embutido o *SQLite*, que é uma biblioteca que implementa um mecanismo de banco de dados, sem servidor e com código de domínio público (LECHETA, 2010).

O *SQLite* dá suporte a maioria das funções da *SQL92*, a terceira revisão do padrão *SQL*. Armazena as informações em um arquivo único e o tamanho de seu código é pequeno, por isso, pode ser utilizado em ampla gama de aparelhos, que disponham de recursos limitados (*SQLITE*, 2012). Seu uso é recomendado em sistemas e aplicativos para dispositivos móveis, não sendo apropriado para aplicações de maior porte (*SQLITE*, 2012).

Conforme Kreibich (2010), apesar de possuir restrições se comparado a um banco de dados corporativo, é mais vantajoso optar pelo uso do *SQLite* em aplicações de pequeno porte tendo em vista as garantias fornecidas em relação à atomicidade, consistência, isolamento e durabilidade dos dados (*ACID*), possibilidade do uso da maioria dos recursos da linguagem de consulta (*SQL*), além de permitir a criação de procedimentos armazenados (*stored procedures*) e uso de *constraints* entre objetos de banco de dados.

Com a utilização deste banco de dados, o armazenamento, consulta e manutenção das informações se torna mais rápido e prático. Os celulares mais antigos proviam o armazenamento em arquivos simples.

Esse banco de dados pode ser instalado em outras plataformas, como o *Windows*, *Linux* e *IOS*, facilitando o desenvolvimento, podendo-se testar as consultas e comandos *SQL* em um ambiente gráfico. O limite desse banco de dados será dado pelo tamanho da memória do dispositivo, podendo chegar a *terabytes* de armazenamento (*SQLITE*, 2012).

3 DESENVOLVIMENTO DO PROJETO

Este capítulo aborda o desenvolvimento do projeto, apresentando a forma de como foi realizado o seu desenvolvimento, destacando os casos de uso do sistema.

3.1 DESCRIÇÃO DO PROBLEMA

Para o desenvolvimento do trabalho foi utilizado o sistema portal de compras web de licitação de uma prefeitura na forma de Pregão Eletrônico. O público alvo do projeto proposto são as empresas fornecedoras de diversos itens mercantis que estarão habilitadas a fornecer para a Prefeitura.

O Pregão Eletrônico é realizado atualmente por um portal web denominado Portal de Compras da Prefeitura. Por meio deste portal, acessando o módulo “portal fornecedor”, as empresas fornecedoras podem participar dos processos licitatórios. Para que as empresas fornecedoras possam participar dos processos licitatórios é necessário que estejam previamente cadastradas no sistema, sendo que a autenticação da empresa é controlada pelo “portal fornecedor”, no qual *login* e senha são escolhidas pelas próprias empresas.

Uma vez que o fornecedor tenha se cadastrado e sido liberado pelos administradores do sistema do portal de compras (Auditores podem aprovar ou reprovar a empresa fornecedora), a empresa passa a ter acesso ao módulo do “portal fornecedor”, que permite a visualização de suas informações de cadastro e de seus processos licitatórios de modalidade Pregão Eletrônico.

Neste contexto, a idéia é desenvolver um aplicativo para dispositivos móveis que realize a condução de processos licitatórios de modalidade Pregão Eletrônico “PE”. Os dispositivos móveis alvos são os *smartphones* que possuem o sistema operacional *Android* com versão 2.1 ou superior. Este aplicativo realiza consultas a um *Web Service* que retorna informações do Pregão Eletrônico para ser conduzido no dispositivo móvel.

Esse processo licitatório é um procedimento que permite as empresas fornecedoras licitantes encaminhar seus lances exclusivamente por meio eletrônico. Durante o transcurso da sessão pública, os licitantes são

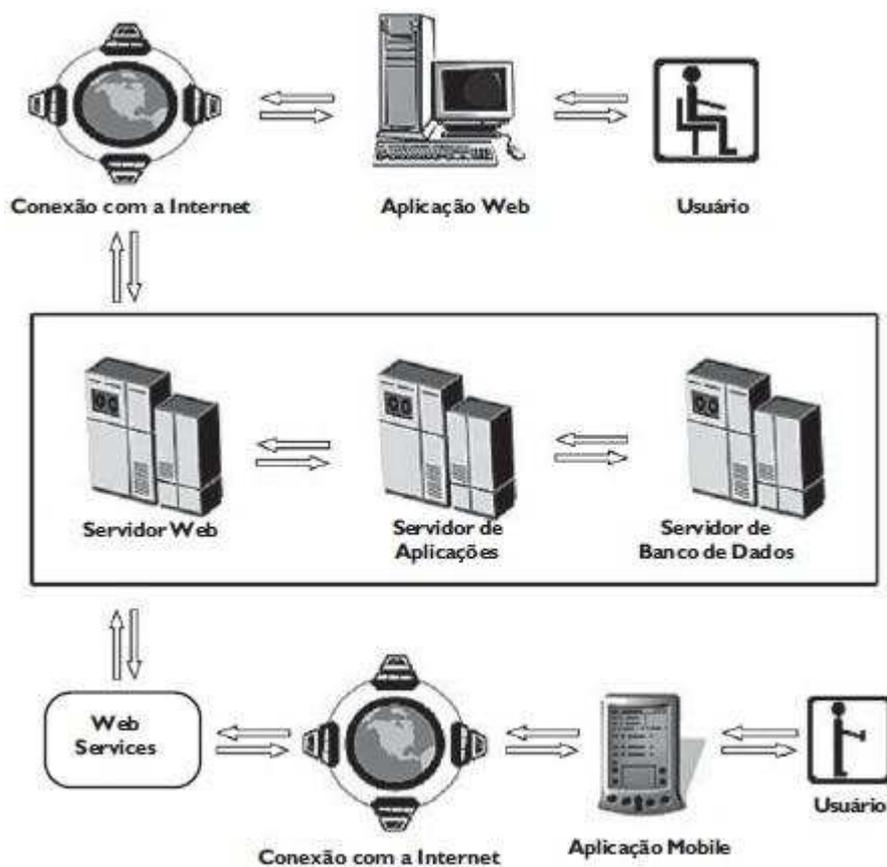


Figura 12 – Arquitetura de hardware do sistema.

3.2 ANÁLISE DE REQUISITOS

Esta seção apresenta os requisitos não-funcionais e funcionais para o aplicativo móvel desenvolvido.

3.2.1 REQUISITOS NÃO-FUNCIONAIS

O sistema foi homologado para ser executado em qualquer versão do *Android* 2.2 ou superior. A Figura 13 apresenta as versões disponíveis do sistema operacional *Android* para um dispositivo móvel.

Versão	Distribuição	Número da API
2.2	Froyo	8
2.3.x	Gingerbread	9-10
3.x.x	Honeycomb	11-13
4.0/4.03/4.04	Ice Cream Sandwich	14-15
4.1.x/4.2.x/4.3	Jelly Bean	16
4.3.x/4.4	KitKat	19

Figura 13 – Versões do *Android* homologadas para rodar o sistema.

O sistema proposto foi desenvolvido em linguagem Java, utilizando um SDK, disponibilizado pelo *Google*, para auxiliar no desenvolvimento. Por meio deste SDK podem ser integradas a IDE Eclipse junto a um emulador para testes do sistema.

A arquitetura aplicada no projeto é classificada em 3 (três) camadas: Apresentação, Negócio e Dados.

A camada de apresentação é a camada responsável pelas telas, interfaces entre o sistema e o usuário.

A camada de negócio é a camada intermediária, na qual as funcionalidades, a estrutura e as regras são implementadas. Esta camada realiza o processamento das informações trocadas entre o sistema e o usuário. As informações que estão armazenadas no banco de dados são passadas da camada de negócio para a camada de dados por meio da integração entre o *Web Service* e a aplicação cliente *Android*.

A camada de dados é a camada na qual as informações são efetivamente gravadas e recuperadas de um banco de dados remoto.

No requisito de segurança, o sistema possui um procedimento de autenticação de usuários, nos quais tenham que se identificar usando um *login* e senha de acesso. Somente usuários autorizados tem acesso às informações do processo licitatório, por meio de uma senha criptografada armazenada no banco de dados.

3.2.2 REQUISITOS FUNCIONAIS

Os requisitos funcionais apresentam os serviços que o sistema deve oferecer e suas funcionalidades, como devem se comportar as entradas nas mais variadas situações (Sommerville, 2007). Os requisitos funcionais para o sistema proposto são:

- O sistema deve permitir a validação de *login* e senha dos usuários;
- O sistema deve listar as empresas vinculadas ao usuário autenticado;
- O sistema deve listar os processos na qual a empresa selecionada deve estar habilitada a participar;
- O sistema deve permitir ao fornecedor autenticado um termo de aceite de participação de processo;
- O sistema deve permitir o envio de propostas para o item mercantil do processo (disputa fechada);
- O sistema deve permitir o envio de lances para o item mercantil do processo (disputa aberta).

3.3 CASOS DE USO

Os requisitos funcionais para o sistema proposto implementam os seguintes casos de uso:

1. **Autenticar Fornecedor:** este caso de uso trata da validação das informações (*login* e senha) passadas pelo usuário. Após a captura destas informações as mesmas serão avaliadas por meio de um *Web Service*, concedendo ou não acesso ao sistema.
2. **Selecionar Empresas (Fornecedores):** Este caso de uso efetua a consulta das empresas vinculadas ao usuário fornecedor autenticado no sistema. A descrição da Razão Social e o CNPJ das empresas são apresentadas na lista de fornecedores. É a partir da empresa fornecedora selecionada que é apresentado os processos na qual a empresa estará habilitada para participar.
3. **Selecionar Processos:** Este caso de uso efetua a consulta dos processos licitatórios na qual o usuário fornecedor autenticado tem

acesso, isto é, se a empresa está habilitada ao item mercantil do processo licitatório que pode fornecer. É apresentada na lista de processos a descrição do processo e o status do andamento do processo.

4. **Manter Aceite de Participação:** Este caso de uso permite que os usuários (Fornecedores) leiam e aceitem os termos para participação no processo de modalidade Pregão Eletrônico (PE).
5. **Manter Propostas:** Este caso de uso é responsável pela pesquisa do item mercantil do processo licitatório para que o usuário possa efetuar o encaminhamento das propostas. Somente a última proposta encaminhada que será válida para o processo, antes do início da fase de lances.
6. **Manter Lances:** Este caso de uso permite que os usuários (Fornecedores) possam efetuar o encaminhamento de seus lances para o item mercantil que atende ao processo licitatório. Somente está disponível no processo o item que possui proposta gerada. O melhor lance para o item que atende ao processo é visualizado, considerando o seu critério de julgamento (menor preço ítem). As atualizações do melhor lance encaminhado ocorrem durante o certame do processo (fase de envio de lances).
7. **Manter Operação Web Service:** Este caso de uso consiste na execução dos métodos implementados do *Web Service*, avaliando as informações de retorno e resposta no formato *XML* do resultado. Este caso de uso é utilizado pelos demais casos de uso citados anteriormente.

O diagrama de casos de uso do sistema proposto (Figura 14) ilustra os módulos implementados no sistema, atendendo as necessidades dos usuários fornecedores que participam dos processos licitatórios de modalidade Pregão Eletrônico (PE).

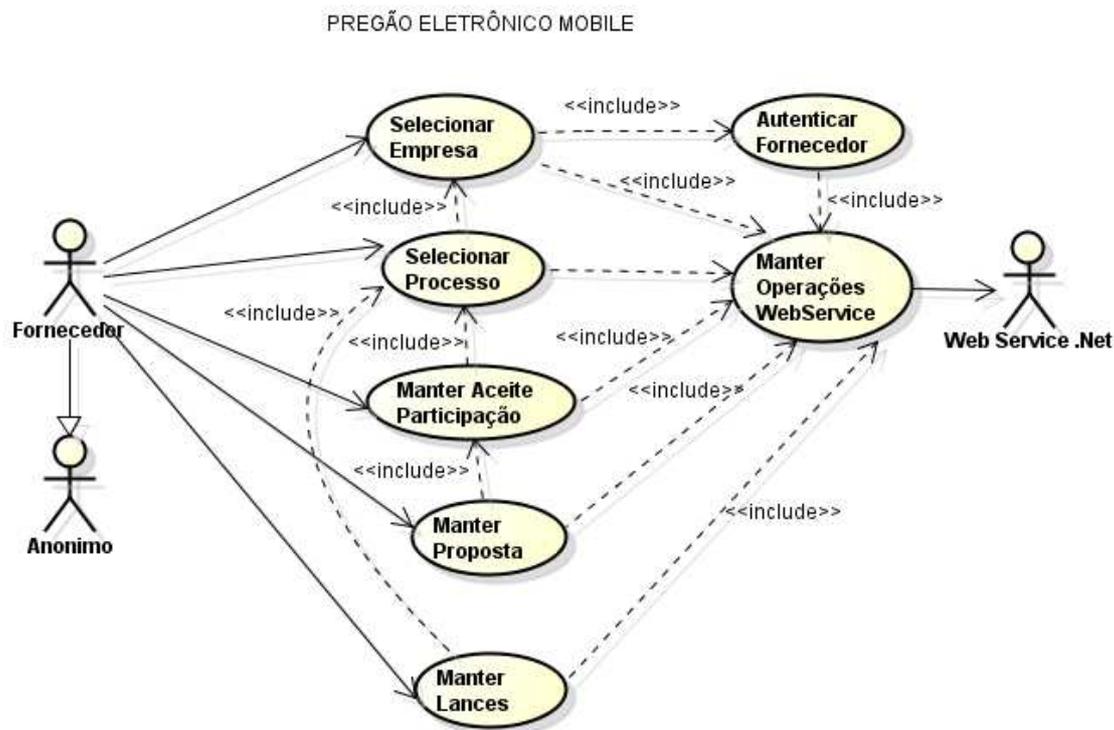


Figura 14 – Modelo de caso de uso do sistema de condução de processos de modalidade Pregão Eletrônico (PE).

A seguir cada caso de uso é explicado detalhadamente. Também são apresentados um fragmento do diagrama de classes, o diagrama de seqüências e a tela projetada para a aplicação, referentes a cada caso de uso.

3.3.1 AUTENTICAÇÃO DO FORNECEDOR

Os usuários do sistema proposto representam as empresas fornecedoras que participam de processos licitatórios de modalidade Pregão Eletrônico (PE). Os usuários (fornecedores) antes de acessar a aplicação cliente *Android* deverão estar cadastrados no sistema Portal de Compras da Prefeitura, sendo que a autenticação do usuário fornecedor é controlada por meio da comunicação com o *Web Service* no sentido da validação dos dados por meio de *login* e senha.

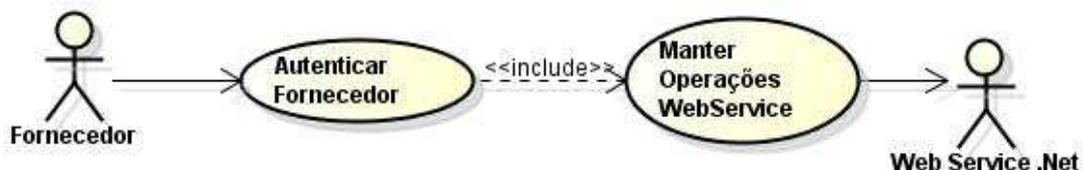


Figura 15 – Use Case Autenticação do Fornecedor.

FLUXO DE CASO:

Fluxo Principal:

1. O sistema exibe a tela de *login* e senha (tela inicial);
2. O usuário preenche os campos de *login* e senha e clica em OK;
3. O sistema valida os dados, e direciona o usuário para a tela de seleção de fornecedores.

Fluxo Alternativo:

1. O usuário não informou todos os campos obrigatórios (E1);
2. Os dados informados não são válidos (E2);
3. O usuário não tem nenhuma empresa vinculada (E3);
4. O usuário fornecedor clicou em “Sair” (E4).

PRÉ-CONDIÇÕES:

1. O usuário deve estar previamente cadastrado no sistema Portal de Compras.

PÓS-CONDIÇÕES:

1. *Login* e Senha Validada.

EXCEÇÕES:

1. E1: Sistema exibe mensagem informando obrigatoriedade dos campos (*Login* e Senha);
2. E2: Sistema exibe mensagem informando que o *login* ou senha estão incorretos;

3. E3: Sistema exibe mensagem informando que o usuário não possui vínculo com nenhuma empresa cadastrada no sistema Portal de Compras da Prefeitura;
4. E4: O sistema exibe mensagem informando ao usuário que o sistema será encerrado.

O fragmento do diagrama de classes referente a autenticação do fornecedor é apresentado na Figura 16:

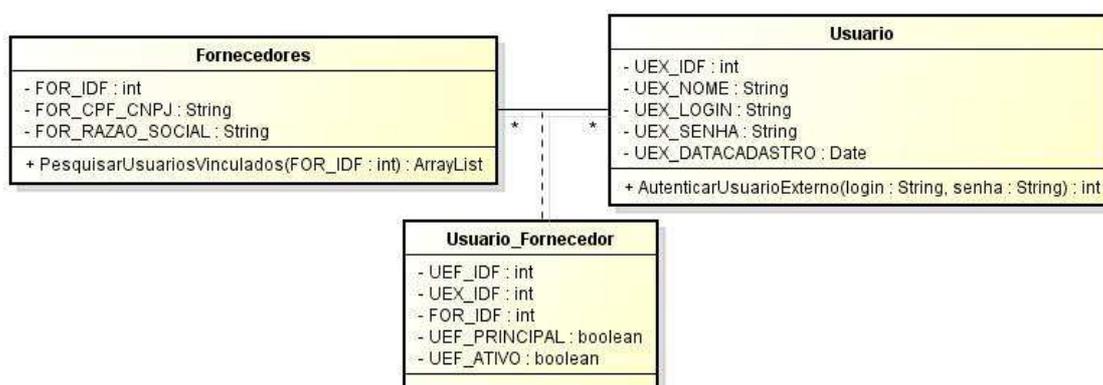


Figura 16 – Modelo de Classe Autenticação do Fornecedor.

O processo de autenticação do usuário fornecedor realiza a validação dos dados informados pelo usuário. A Figura 17 apresenta a situação na qual o usuário possa informar o seu *login* e senha para acesso ao sistema.

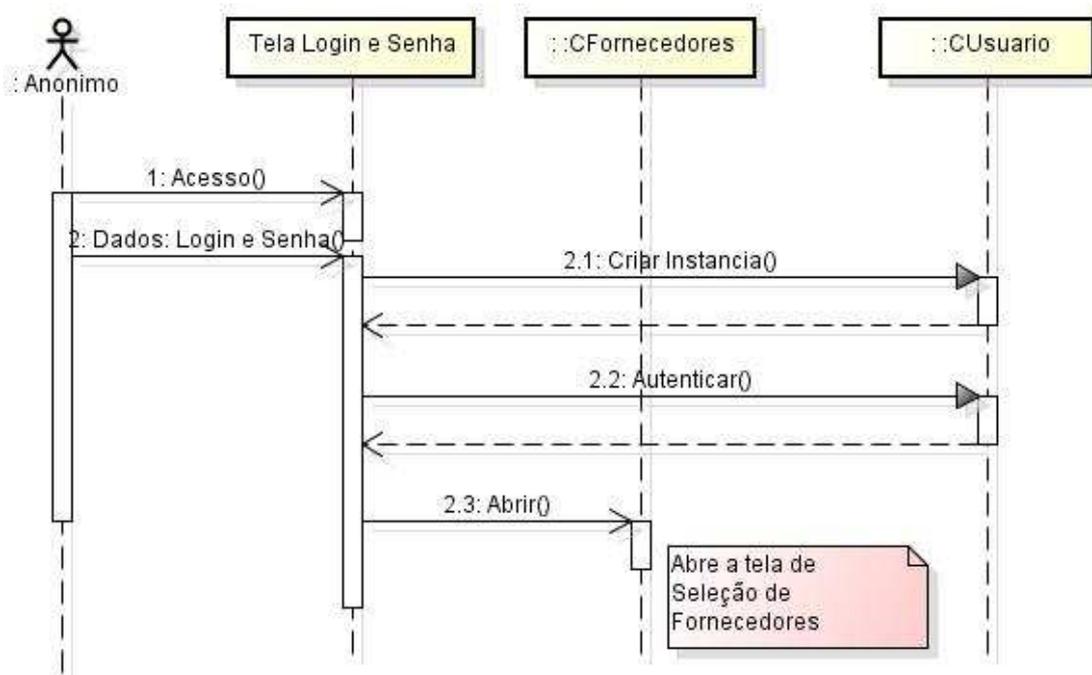


Figura 17 – Diagrama de Seqüência Autenticação do Fornecedor.

Na Figura 18 são apresentadas as telas de Autenticação de Fornecedores e Listagem de Fornecedores. Após a autenticação do fornecedor no sistema por meio do *login* validado, é apresentada uma lista de empresas fornecedoras vinculados ao usuário autenticado que conduzirá o certame no sistema.



Figura 18 – Tela de Autenticação / Seleção do Fornecedor.

3.3.2 MANTER ACEITE DE PARTICIPAÇÃO DA SELEÇÃO DO PROCESSO DE MODALIDADE PREGÃO ELETRÔNICO (PE)

Após o usuário fornecedor selecionar a empresa que deve ser representada durante a condução do processo licitatório o sistema apresenta a lista de processos licitatórios na qual poderá participar. Para que o usuário fornecedor possa participar do Pregão, o processo licitatório deverá estar na fase de encaminhamento de “propostas” ou “sessão de lances” (caso já tenha encaminhado lances). Durante a condução do processo licitatório selecionado o usuário fornecedor deverá ler e aceitar os termos para participação no processo em questão. Neste aceite será verificado se o usuário fornecedor está bloqueado ou penalizado (suspensão ou inidôneo) para participação do processo licitatório selecionado.

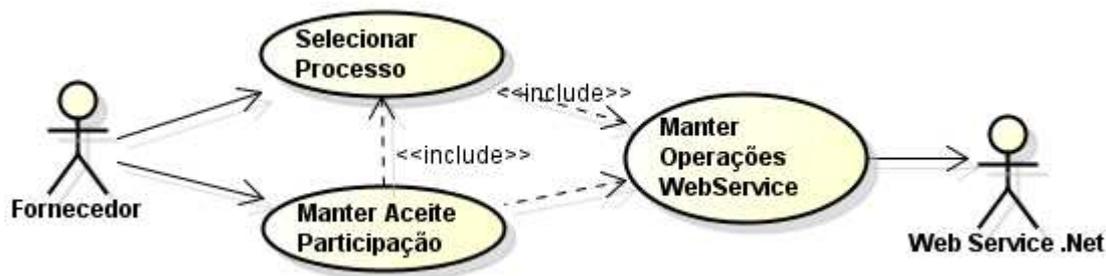


Figura 19 – Use Case Manter Participação do Processo.

FLUXO DE CASO:

Fluxo Principal:

1. O fornecedor clica no ícone “Acesso”, no *grid* de processos licitatórios em andamento na tela do sistema;
2. O sistema exibe a tela com o termo de aceite para participação do processo;
3. O Fornecedor clica em “Aceitar”.
4. O sistema valida se o usuário fornecedor pode participar do processo e direciona para a tela de propostas

Fluxo Alternativo:

1. O processo não está na fase de propostas (E1);
2. O usuário fornecedor clicou em “Recusar” (E2);
3. O usuário fornecedor clicou em “Sair” (E3);
4. O sistema não validou a participação do usuário fornecedor (E4, E5, E6).

PRÉ-CONDIÇÕES:

1. O usuário fornecedor deve estar autenticado.

PÓS-CONDIÇÕES:

1. O usuário deve selecionar o processo na lista de processos em andamento.

EXCEÇÕES:

1. E1: O sistema exibe mensagem informando ao usuário que o período para o aceite do processo já foi encerrado, redirecionando para a tela de listagem de processos em andamento;
2. E2: O sistema exibe mensagem informando ao usuário fornecedor o fim da participação no processo e atualiza o fornecedor como recusado, redirecionando para a tela de listagem de processos em andamento;
3. E3: O sistema exibe mensagem informando ao usuário fornecedor que o sistema será encerrado;
4. E4: O sistema exibe mensagem informando ao usuário fornecedor que está penalizado, redirecionando para a tela de listagem de processos em andamento;
5. E5: O sistema exibe mensagem informando ao usuário fornecedor que está reprovado na operação de auditoria de fornecedores, redirecionando para a tela de listagem de processos em andamento;
6. E6: O sistema exibe mensagem informando ao usuário fornecedor que o processo já foi recusado anteriormente, redirecionando para a tela de listagem de processos em andamento.

O fragmento do diagrama de classes referente a participação dos fornecedores no processo licitatório é apresentado na Figura 20:

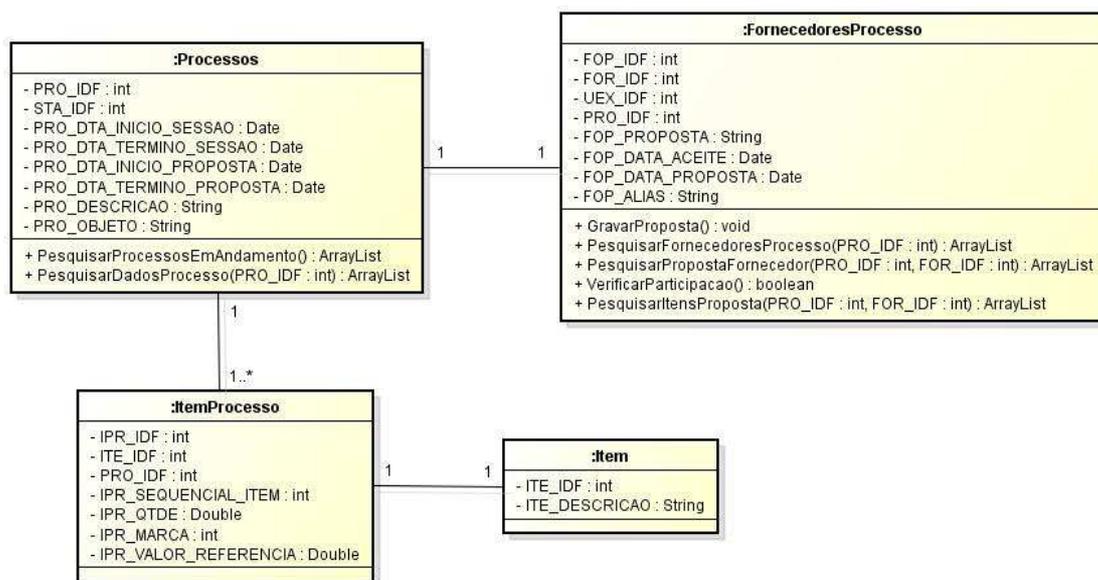


Figura 20 – Modelo de Classe Manter Aceite de Participação do Processo.

O termo de aceite de participação dos fornecedores nos processos licitatórios na forma eletrônica é uma etapa do sistema prevista por lei (§ 2º do artigo 32 da lei 8.666/1993). Ao recusar a participação do processo o fornecedor não estará mais permitido a participar do processo posteriormente. A recusa em participar de um processo é de total responsabilidade da empresa participante. Na Figura 21 é apresentada a situação na qual o usuário fornecedor poderá ou não aceitar a sua participação no processo.

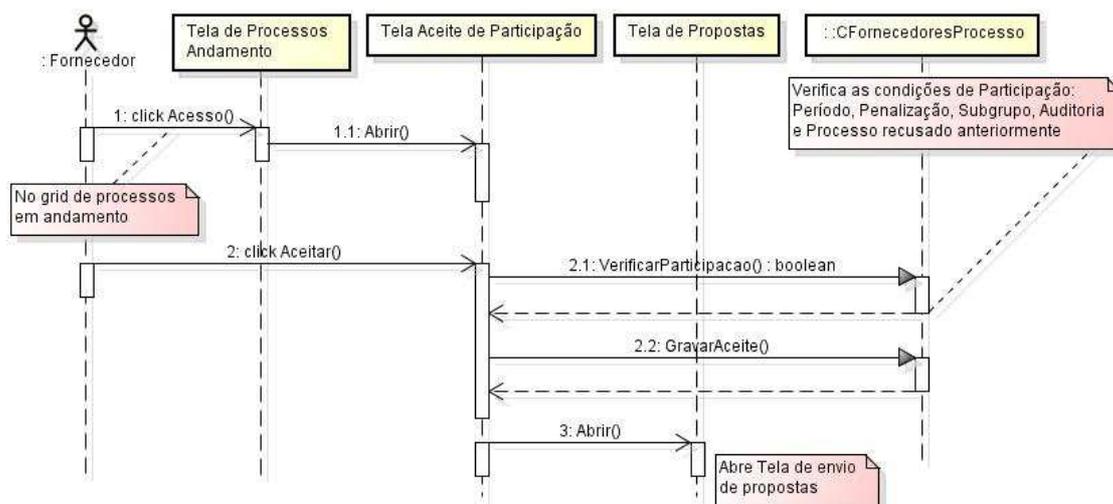


Figura 21 – Diagrama de Seqüência Manter Participação do Processo.

Na Figura 22 são apresentadas as telas de Listagem de Processos em Andamento (somente para os status: andamento de processo, fase de propostas e fase de sessão de lances) e Tela de Termo de Aceite de Participação. A tela de Lista de Processos em Andamento apresenta os processos na qual a empresa está habilitada a participar, sendo que logo após selecionar um processo durante o status de sessão de propostas é apresentado uma tela na qual a empresa é obrigada a credenciar a sua participação no processo.



Figura 22 – Telas para participação do processo licitatório.

3.3.3 MANTER PROPOSTAS

O usuário fornecedor ao aceitar a participação do processo licitatório o sistema redirecionará para a tela de envio de propostas. Nesta etapa do sistema os usuários fornecedores deverão encaminhar suas primeiras propostas para o item mercantil pertencente ao processo licitatório selecionado. Estas propostas poderão ser encaminhadas mais de uma vez, lembrando que será considerada a última enviada. O encaminhamento de propostas pelos

usuários fornecedores participantes é realizado sem que haja interação entre os participantes do processo, isto é, sendo uma forma de disputa “fechada” entre as empresas participantes. Para cada processo licitatório apresentado durante esta etapa haverá um período de tempo determinado em que os usuários fornecedores poderão encaminhar suas propostas, respeitando o critério de julgamento estabelecido no processo. Após o término deste período os usuários fornecedores que não encaminharam suas propostas não poderão mais participar do processo licitatório selecionado.

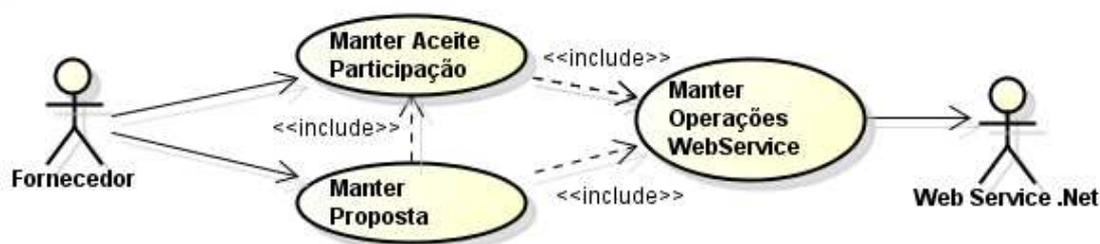


Figura 23 – Use Case Manter Propostas.

FLUXO DE CASO:

Fluxo Principal:

1. O usuário clica no botão “Aceitar” na tela de aceite de participação de processo licitatório;
2. O sistema exibe a tela de encaminhamento de propostas;
3. O usuário fornecedor preenche os campos (Marca/Modelo e Valor) e clica em “Enviar Proposta”;
4. O sistema valida os dados e insere a proposta para o fornecedor (RN1, RN2);
5. O sistema exibe a tela de confirmação de envio de proposta.

Fluxo Alternativo:

1. O tempo para o envio de proposta foi encerrado (E1);
2. O sistema não validou as informações (E2, E3);
3. A proposta já foi inserida anteriormente (E4).
4. O sistema exibe tela de confirmação de proposta.

PRÉ-CONDIÇÕES:

1. O usuário fornecedor deve estar autenticado;
2. O caso de uso “Manter Aceite Participação” deve ter sido realizado.

PÓS-CONDIÇÕES:

1. Nenhuma.

REGRA DE NEGÓCIO:

1. RN1: Se o item estiver acima do valor de referência, o sistema exibe mensagem informando ao usuário fornecedor que o valor deve estar abaixo do valor de referência (R\$ 0,00), isto é, o critério de julgamento do processo é menor preço item.
2. RN2: Se o tempo para envio de proposta estiver encerrado, o sistema exibe mensagem informando ao usuário fornecedor que o tempo para envio de proposta foi encerrado.

EXCEÇÕES:

1. E1: O sistema exibe mensagem informando ao usuário fornecedor que o tempo para envio de propostas encerrou;
2. E2: O sistema exibe mensagem informando ao usuário fornecedor que os campos do item do processo devem ser preenchidos;
3. E3: O sistema exibe mensagem informando ao usuário fornecedor que o valor preenchido deve ser menor que o valor de referência do item;
4. E4: O sistema exibe mensagem informando ao usuário fornecedor que o valor preenchido já foi encaminhado anteriormente.

O fragmento do diagrama de classes referente ao encaminhamento de propostas é apresentado na Figura 24:

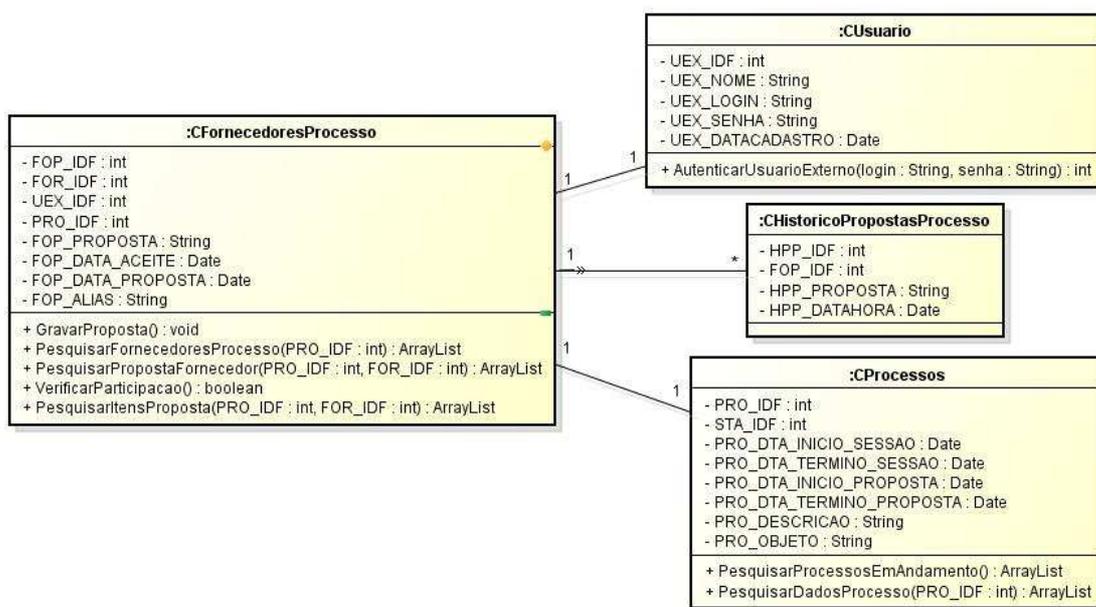


Figura 24 – Modelo de Classe Manter Propostas.

Para os processos licitatórios que exigem cadastramento de propostas os usuários fornecedores deverão encaminhar suas primeiras propostas. Estas propostas podem ser encaminhadas mais de uma vez, porém somente a última proposta encaminhada é que será considerada. Como o critério de julgamento adotado para os processos licitatórios selecionados é menor preço item, os usuários fornecedores deverão encaminhar suas propostas abaixo do valor de referência do item mercantil do processo. Na Figura 25 é mostrada a situação em que o usuário fornecedor poderá encaminhar suas propostas.

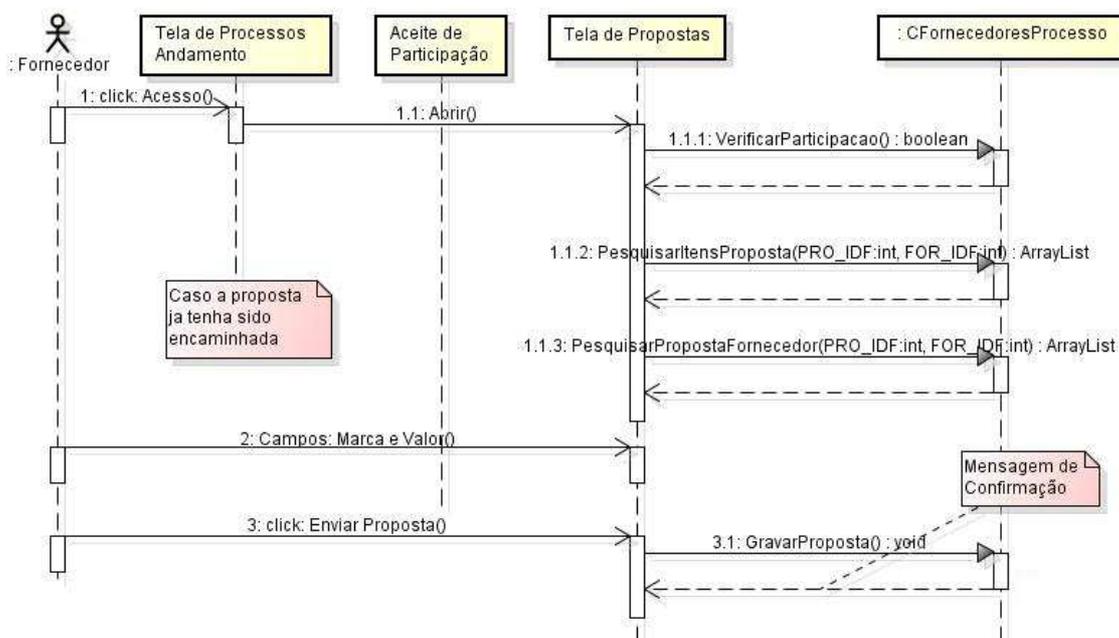


Figura 25 – Diagrama de Seqüência Manter Propostas.

Na Figura 26 é apresentado a Tela de Envio de Propostas. Durante o envio de propostas o sistema apresenta ao usuário fornecedor um período válido para encaminhar as suas propostas (tempo de encerramento). O usuário fornecedor que não encaminhar propostas estará impedido de participar da fase de envio de lances, isto é, estará automaticamente eliminado do certame.



Figura 26 – Tela de envio de propostas para o processo licitatório.

3.3.4 MANTER SESSÃO DE LANCES

Após a fase de encaminhamento de propostas, serão consideradas como primeiro lance (lance inicial) das empresas participantes na fase de sessão de lances a última proposta encaminhada. Nesta fase as empresas poderão reduzir seus valores até a data/hora de encerramento do processo.

O usuário licitante que obtiver o menor valor para o item mercantil do processo será considerado vencedor. Será mostrada na tela durante esta fase, a descrição do fornecedor que estiver vencendo por meio de um “alias” e o valor do lance vencedor atual para o item mercantil em disputa, considerando o critério de julgamento “menor valor item”. A cada 15 segundos as informações referentes ao vencedor atual do processo licitatório serão atualizadas na tela.

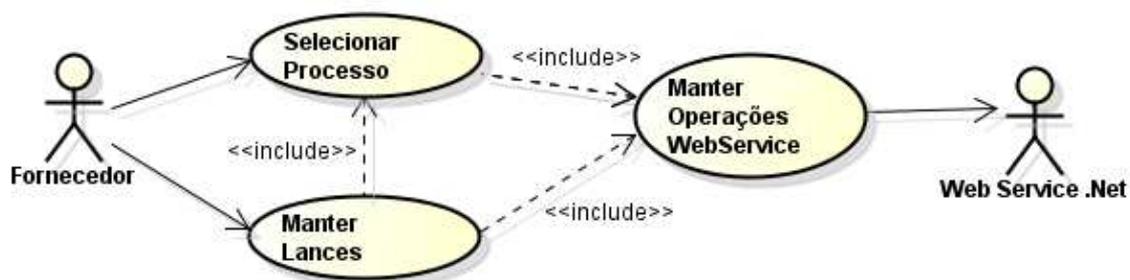


Figura 27 – Use Case Manter Lances.

FLUXO DE CASO:

Fluxo Principal:

1. O usuário fornecedor clica no ícone “Acesso”, no *grid* de processos licitatórios em andamento da tela inicial do sistema (para lances já encaminhados anteriormente);
2. O usuário fornecedor clica em “OK” na mensagem de finalização de envio de propostas;
3. O sistema exibe a tela de envio de lances para o processo selecionado;
4. O sistema carrega o item para o processo;
5. O fornecedor preenche o campo “Valor” e clica em “Enviar Lance”;
6. O sistema valida a informação e insere o lance (RN1, RN2);
7. O sistema exibe mensagem de lance encaminhado com sucesso (E3).

Fluxo Alternativo:

1. Não foi encaminhada proposta para o processo selecionado (E1);
2. Não foi realizado o aceite do processo selecionado (E2);
3. O processo não está mais na fase de sessão de lances (E4);
4. O usuário fornecedor clica em “OK” na mensagem de finalização da sessão de envio de lances, redirecionando o usuário fornecedor para a tela processos em andamento.

PRÉ-CONDIÇÕES:

1. O usuário fornecedor deve estar autenticado;
2. O caso de uso “Manter Aceite de Participação” deve ter sido realizado;

3. O usuário fornecedor ter encaminhado a proposta.

PÓS-CONDIÇÕES:

1. Nenhum.

REGRA DE NEGÓCIO:

1. O valor unitário do item deve ser preenchido e de valor menor ou igual ao valor de referência do item;
2. O valor unitário do item já foi encaminhado anteriormente.

EXCEÇÕES:

1. E1: O sistema exibe mensagem informando ao usuário fornecedor que a proposta não foi encaminhada anteriormente;
2. E2: O sistema exibe mensagem informando ao usuário fornecedor que o aceite de participação para o processo selecionado não foi realizado;
3. E3: O sistema exibe mensagem informando ao usuário fornecedor que o lance foi encaminhado com sucesso.
4. E4: O sistema exibe mensagem informando ao usuário fornecedor que o processo está na fase de julgamento.

O fragmento do diagrama de classes referente ao encaminhamento de lances é apresentado na Figura 28:

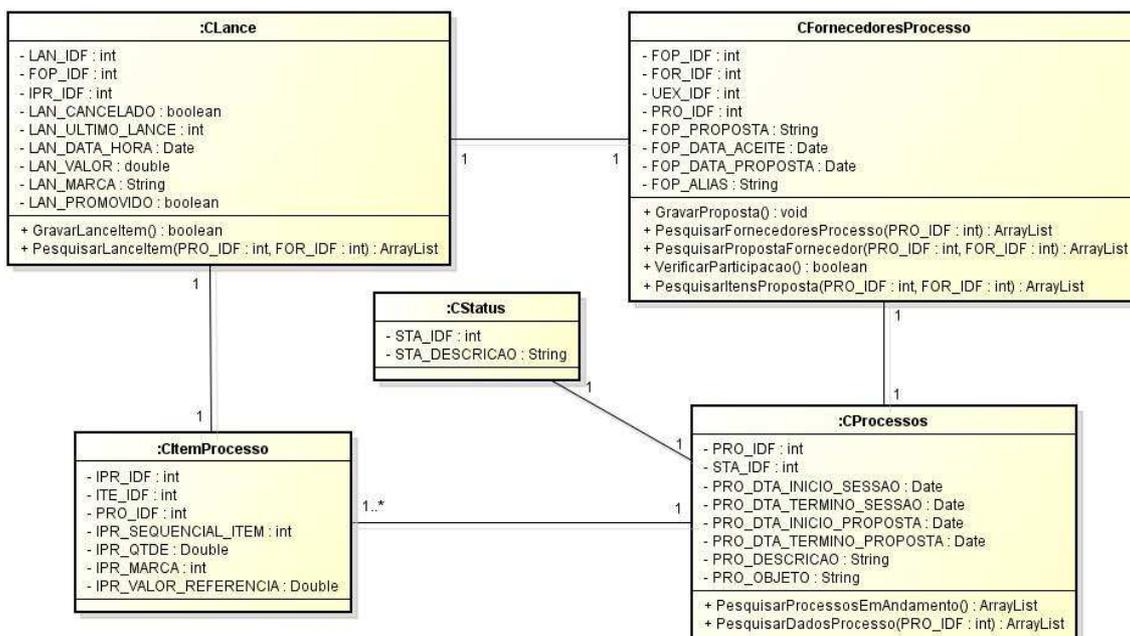


Figura 28 – Modelo de Classe Manter Lances.

Durante a fase de sessão de lances, os usuários fornecedores encaminham seus respectivos lances do processo selecionado. Só poderão participar da fase de sessão de lances somente os usuários fornecedores que encaminharam propostas. Também durante a fase de sessão de lances não será permitida a troca da marca/modelo do item mercantil do processo. Na Figura 29 é apresentada a situação em que o usuário fornecedor encaminha seus lances.

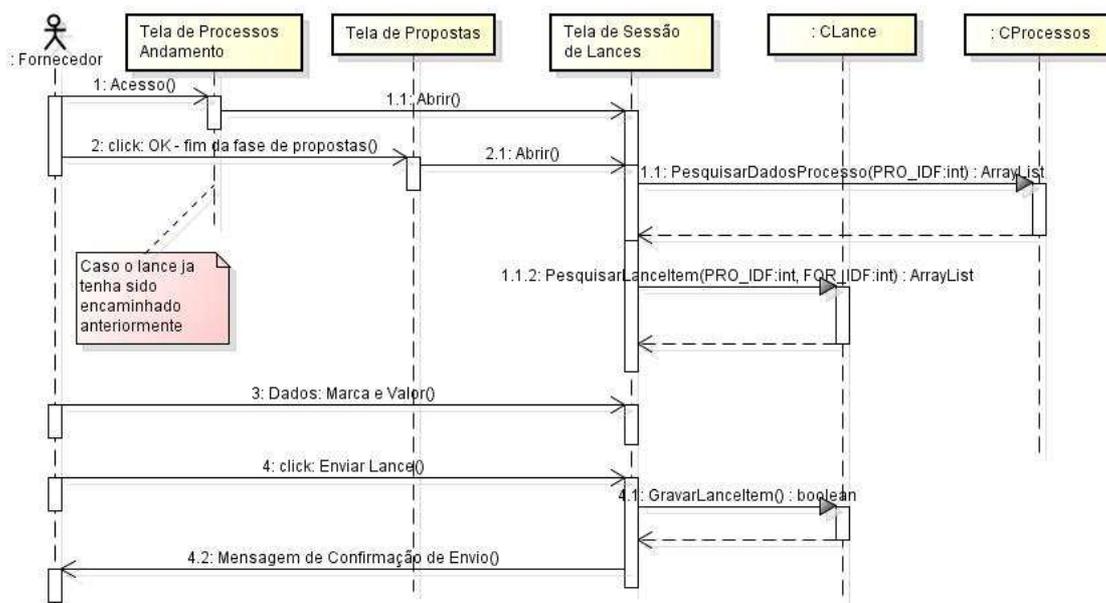


Figura 29 – Diagrama de Seqüência Manter Lances.

Na Figura 30 é apresentado a Tela de Envio de Lances. É durante esta etapa que o usuário fornecedor pode reduzir seus respectivos lances ao mínimo possível para obtenção do melhor lance, respeitando o critério de julgamento menor preço item. Também é nesta etapa que o sistema apresenta quem é o vencedor atual durante a disputa do certame, mantendo a transparência da disputa, caracterizado como uma disputa aberta.



Figura 30 – Tela de envio de lances para o processo licitatório.

3.3.5 MANTER OPERAÇÕES *WEB SERVICE*

A camada de acesso ao *Web Service* do sistema proposto proporciona a troca de informações entre o dispositivo móvel *Android* e o servidor *Web Service*. Esta camada é composta por uma classe que realiza a conexão com o servidor *Web Service*, o envio de requisições e o processamento das informações de retorno.

Esta classe chamada “*WebServiceKSOAP*” trata a recepção dos parâmetros e nome do serviço a ser executado, realiza a chamada aos métodos e devolve as informações para cada método executado.

O “ator” Servidor *Web Service .NET* é responsável pelo gerenciamento das informações distribuídas para a aplicação cliente. Na Figura 31 é apresentado um trecho de código do sistema da operação “lista de empresas vinculadas ao usuário autenticado” detalhando esta comunicação:

```

#region RetornarFornecedoresVinculadosAndroid
/// <summary>
///Através do login do usuário é retornado os fornecedores vinculados.
/// </summary>
[WebMethod(Description = @"Através do LOGIN do usuário ele retorna os fornecedores vinculados", EnableSession = true)]
public FornecedoresAndroid[] RetornarFornecedoresVinculadosAndroid(string LOGIN)
{
    return FornecedoresAndroid.RetornarFornecedoresVinculadosAndroid(LOGIN);
}
#endregion

```

Figura 31 – Método chamado via http.

No trecho de código da Figura 31 o nome do atributo *[WebMethod]*, que sinaliza o sistema de *runtime*, indica que é um método chamado via *HTTP*. A classe que implementa o método acima herda da classe *System.Web.Services.WebServices* que fornece acesso direto aos objetos *ASP.NET*. As rotinas de suporte ao *WebService* a ser executado irão criar automaticamente a página para testá-lo, conforme a Figura 32:

RetornarFornecedoresVinculadosAndroid
Através do LOGIN do usuário ele retorna os fornecedores vinculados

Test
To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
LOGIN:	<input type="text"/>

Invoke

SOAP 1.1
The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```

POST /EcomprasSGPService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://ici.curitiba.org.br/webservices/RetornarFornecedoresVinculadosAndroid"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
<soap:Body>
  <RetornarFornecedoresVinculadosAndroid xmlns="http://ici.curitiba.org.br/webservices">
    <LOGIN>string</LOGIN>
  </RetornarFornecedoresVinculadosAndroid>
</soap:Body>
</soap:Envelope>

```

Figura 32 – Página para testes da execução do método.

Na Figura 33 é apresentado o resultado da execução do método implementado na classe:

```

<?xml version="1.0" encoding="utf-8" ?>
- <ArrayOfFornecedoresAndroid xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://ici.curitiba.org.br/webservices">
- <FornecedoresAndroid>
  <Codigo>71348</Codigo>
  <RazaoSocial>TESTE FORNECEDOR</RazaoSocial>
  <CNPJCPF>07307946000144</CNPJCPF>
</FornecedoresAndroid>
- <FornecedoresAndroid>
  <Codigo>73307</Codigo>
  <RazaoSocial>TESTE ICI</RazaoSocial>
  <CNPJCPF>72229553000121</CNPJCPF>
</FornecedoresAndroid>
- <FornecedoresAndroid>
  <Codigo>75982</Codigo>
  <RazaoSocial>RAZAO TESTE ICI</RazaoSocial>
  <CNPJCPF>42908116000173</CNPJCPF>
</FornecedoresAndroid>
</ArrayOfFornecedoresAndroid>

```

Figura 33 – Página do resultado da execução do método.

A descrição do serviço no formato *XML* é feita por meio do padrão *WSDL*. Esta é a forma utilizada para criar automaticamente uma classe “*Proxy*” que chamará o *WebService*. O *XML* acima é o retorno para os métodos *HTTP-GET* e *HTTP-POST*, possibilitando sua utilização para qualquer linguagem voltada para *WEB*.

O *SDK* do *Android* não oferece uma solução embutida para consumir *Web Services*. Portanto, uma biblioteca chamada “*KSOAP2*”, que permite o consumo de *Web Services* sem a necessidade de uso de *proxy* dinâmico, foi empregada. Logo, com o uso dos recursos da biblioteca foi necessário implementar uma classe de encapsulamento para definir a coleção de objetos, como por exemplo, “*Fornecedores*”. Uma lista de objetos do tipo “*Fornecedores*” foi criada e o resultado do *WebService* foi obtido. A Figura 34 apresenta um fragmento da rotina de código implementado na classe “*WebServiceKSOAP*” para o retorno das informações de “*Fornecedores*” do *WebService .NET*.

```

/**
 * Método que retorna uma lista de objetos de Fornecedores.
 * @param login autenticado do usuário fornecedor
 * @return ArrayList
 * @throws Exception
 */
public List<Fornecedores> RetornarFornecedores(String login) throws Exception
{
    //Instancia da lista de objetos Fornecedores
    List<Fornecedores> listaFornecedores = new ArrayList<Fornecedores>();

    //Objeto composto pelo namespace do WS e pelo método que será chamado.
    //Representa o encapsulamento da requisição, ou seja, o que vai dentro do envelope.
    //Se caracteriza por ser um objeto dinâmico simples que pode ser usado para construir
    //as chamadas do Ksoap2
    SoapObject soap1 = new SoapObject(NAMESPACE, "RetornarFornecedoresVinculadosAndroid");

    //Cria o envelope (parâmetro: versão do envelope SOAP)
    //Este envelope representa a estrutura de uma mensagem SOAP.
    SoapSerializationEnvelope envelapel = new SoapSerializationEnvelope(SoapEnvelope.VER11);

    //Especifica que o web service utilizado é um .NET;
    envelapel.dotNet = true;

    //Adiciona a propriedade classe "SoapObject" utilizada como (parâmetros) do método que será executado.
    soap1.addProperty("LOGIN", login);

    // Envia a requisição ao web service;
    envelapel.setOutputSoapObject(soap1);

    // Cria a comunicação com o local do web service.
    // Realiza a serialização e desserialização das mensagens SOAP
    HttpTransportSE httpTransport = new HttpTransportSE(URL);
}

```

Figura 34 – Método que retorna a coleção de objetos “Fornecedores”.

3.4 VALIDAÇÃO DA PROPOSTA

Os testes de validação realizados no sistema móvel proposto permitiram verificar a interação entre o sistema cliente *Android* e a aplicação portal de compras de licitação eletrônica durante a condução de um processo licitatório. Para a realização dos testes foi preparado um cenário em que 2 usuários fornecedores pudessem disputar quem teria o menor lance encaminhado durante o certame. Um usuário fornecedor acessando o portal de compras web e outro usuário fornecedor acessando o aplicativo móvel, após realizar todas as operações necessárias para a disputa, os usuários ficaram encaminhando seus respectivos lances até o encerramento do certame. No final da disputa um dos usuários saiu vencedor por ter encaminhado o menor lance possível.

Com a realização da validação preliminar do sistema móvel com dois usuários, foi possível observar que:

- O sistema proposto apresenta de forma clara e objetiva as informações ao usuário para a condução de processos de licitação eletrônica na modalidade de Pregão.
- As funcionalidades apresentadas foram implementadas, já que o sistema móvel disponibiliza ao usuário fornecedor todas as possíveis etapas para a condução do certame de um determinado processo licitatório, desde o seu acesso ao envio de lances.
- Os resultados da utilização do sistema móvel podem ser visualizados diretamente no portal de compras, mostrando desta forma, que a condução de um processo pode ser realizada tanto pelo sistema portal web quanto pelo sistema móvel.
- A utilização do sistema móvel por diversos usuários ao mesmo tempo ainda precisa ser estudada para a completa implementação do trabalho proposto por envolver questões relacionadas ao dimensionamento do servidor *WebService* para poder suportar um número de requisições cada vez maior.

Na Figura 35 é apresentado o emulador SDK do *Android* do funcionamento real do sistema durante a fase de envio de lances.



Figura 35 – Funcionamento do sistema durante o envio de lances.

É importante ressaltar que a integração com a base de dados é realizada por meio de um *Web Service*, transformando as informações em objetos de negócio significativos para o sistema móvel. No entanto, a responsabilidade referente ao controle das informações durante a condução do processo cabe ao portal de compras por meio de um serviço de monitoramento de status e recebimento de propostas, tornando o sistema móvel ainda mais “leve” na questão de requisições a serem realizadas no servidor.

3.5 RESULTADOS OBTIDOS

Com o desenvolvimento do sistema móvel para condução de processos licitatórios na forma eletrônica podem ser citados os seguintes resultados obtidos:

- O sistema móvel apresenta ao usuário fornecedor as informações necessárias para condução de processos de licitação eletrônica na modalidade de Pregão Eletrônico (PE);
- O sistema móvel apresenta ao usuário fornecedor todas as etapas necessárias para efetivar a participação na condução de processos de licitação eletrônica;
- Os resultados finais da utilização do sistema móvel podem ser visualizados diretamente no portal de compras;
- O sistema móvel acessa as informações armazenadas em um banco de dados remoto por meio da utilização de um *Web Service* implementado em *.NET*.

4 CONCLUSÕES E TRABALHOS FUTUROS

Para a elaboração deste trabalho optou-se no desenvolvimento de um aplicativo móvel para condução de processo de licitação eletrônica de modalidade Pregão Eletrônico, que possibilita um incremento importante para as empresas fornecedoras de materiais de bens de consumo, a terem uma participação mais efetiva nas licitações eletrônicas promovidas por órgãos públicos.

O sistema móvel proposto pode agilizar e facilitar para as empresas a participação e a condução dos processos na modalidade de Pregão Eletrônico, aumentando a concorrência entre as empresas participantes em diversos itens mercantis. Desta forma, os órgãos públicos (Prefeituras) que promovem licitações eletrônicas podem adquirir itens mercantil com um menor preço possível, gerando mais economia nas aquisições de materiais de bem de consumo.

Este aplicativo disponibiliza de forma dinâmica o acesso às informações dos processos de licitação eletrônica na modalidade de Pregão Eletrônico, como também a integração entre o dispositivo *Android* e o *Web Service .Net*. Para realizar a integração entre estas tecnologias foi utilizada uma biblioteca específica para a plataforma *Android*, denominada *KSOAP2*, integrando-a no sistema para a utilização de suas *APIs*, proporcionando uma comunicação mais direta e com menos capacidade de processamento.

No sistema foram desenvolvidas funcionalidades para a condução de processos de licitação eletrônica na modalidade Pregão Eletrônico, com critério de julgamento menor preço item. Durante a utilização do sistema, o usuário fornecedor realiza diversas operações, como por exemplo, autenticação por meio de *login* e senha, seleção de empresas habilitadas a participar de processos de licitação, seleção do processo licitatório, termo de aceite para participação, envio de propostas e envio de lances.

Com relação a trabalhos futuros, podem ser enumeradas as seguintes extensões:

- O sistema além da condução de processos na modalidade de Pregão Eletrônico poderia conduzir processos de outras modalidades, como por

exemplo, “Leilão Eletrônico - LE” e “Dispensa Eletrônica de Licitação – DE/DL”;

- O sistema proposto realiza o envio de lances somente para um único item mercantil pertencente ao processo, no futuro a aplicação também poderia conduzir processos com mais itens mercantis;
- Os processos conduzidos no sistema que possui como critério de julgamento “menor preço item”, contém lances encaminhados para o item mercantil específico e sempre abaixo do seu valor de referência. O sistema também poderia conduzir processos com outros critérios de julgamento, como por exemplo, “menor preço item lote”, sendo neste caso, processos disputados na forma agrupada de itens, na qual o vencedor é aquele que possui a menor soma dos lances encaminhados item a item do lote;
- Para a condução de processos com mais de um item mercantil e também na forma agrupada de itens para um determinado processo, a arquitetura do sistema poderia ser aprimorada para que as informações sejam persistidas também no dispositivo móvel. Desta forma, poderia ser utilizado um banco de dados local, utilizando o *SQLite* nativo para dispositivos *Android*, o que reduziria a necessidade de acessos frequentes ao *Web Service*.

Como o desenvolvimento de aplicativos móveis para esse segmento ainda é pequeno, faz-se necessário o aumento de pesquisas para o desenvolvimento de sistemas nesta área.

É importante ressaltar que a implementação deste projeto proporcionou um aprofundamento no aprendizado da linguagem Java, bem como, do *Android*. Além desse aprofundamento, proporcionou também, um amplo estudo na obtenção de conhecimentos necessários no desenvolvimento de aplicativos móveis. O sistema utiliza um conjunto de bibliotecas únicas de componentes, utilizados pela própria plataforma *Android*. Esta padronização facilita o aprendizado e ajuda a popularizar cada vez mais o uso da tecnologia.

No entanto, um problema enfrentado durante o desenvolvimento da aplicação *Android* foi a ocorrência de mudanças de versões de releases do *SDK Android*, o que em alguns momentos implicou severamente ou

parcialmente na compatibilidade dos componentes utilizados no sistema proposto.

REFERÊNCIAS BIBLIOGRÁFICAS

Alonso e Bussler (2004) – SOAP Exchange Patterns (v1.2). Disponível em http://www.ime.usp.br/~reverbel/SOD-06/Slides/2004_edbt_web_services_part_1.pdf, Acessado em 14 de agosto de 2014.

ANDROIDZ - FORUM, 2013: O que é o SDK do Android. Disponível em: <http://www.androidz.com.br/portal/o-que-e-o-sdk-do-android.html> > Acesso em: 04 de agosto de 2014.

STEFANINI – ARTIGO, 2014 Aplicativos: As ferramentas da mobilidade. Disponível em: [http:// stefanini.com.br/2013/09/aplicativos-as-ferramentas-da-mobilidade/](http://stefanini.com.br/2013/09/aplicativos-as-ferramentas-da-mobilidade/) Acessado em: 04 de agosto de 2014.

ROSAS DA SILVA, FRANCISCO – Instrutor do Curso de Formação de Pregoeiros / 8ª ICEx (Inspetoria de Contabilidade e Finanças do Exército): Apostila Curso de Formação de Pregoeiros. ENAP, 2003.

BENZ, B. E DURANT, J. XML Programming Bible. New York: Wiley Publishing Inc, 2003.

BOSELI, Paulo. Pregão – Capacitação para Pregoeiros e Licitantes. Editora Negócios Públicos, 2009.

DURÃES, Ramon 2012. *.NET+WEBSERVICES* – Web Services para Iniciantes. Disponível em “www.imasters.com.br/artigo/3561” Acessado em 15 de agosto de 2014.

FARIA, Alessandro O. Como instalar o Android SDK. 2011. Disponível em: http://pt.opensuse.org/ARTIGOS:Como_instalar_o_Android_SDK, Acessado em 14 de agosto de 2014.

FERNANDES, Maicris, 2013. , Blake – Android: Desenvolvimento de Aplicações Android, São Paulo: Novatec Editora Ltda Disponível em: pontov.com.br/site/android, Acessado em 14 de agosto de 2014.

Google Developers, 2014: Google For Education Guidelines. Disponível em <<https://developers.google.com/edu/guidelines>> Acesso em: 04 de agosto de 2014.

KNUTSEN, J. *Web Service Clients on Mobile Android Devices*, 2009.

Kreibich, J. *Using SQLite*. O' Reilly Media, 2010.

KSOAP2 ANDROID – Google Project Hosting. Disponível em: <<http://code.google.com/p/ksoap2-android>>. Acesso em: 04 de agosto de 2014.

LECHETA, Ricardo R. *Google ANDROID - Aprenda a criar aplicações para dispositivos móveis com o Android SDK - 3ª Edição* - São Paulo: Novatec, 2010.

Maxwell: Repositório Institucional da Pontifícia Universidade Católica do Rio de Janeiro. Disponível em <http://www.maxwell.lambda.ele.puc-rio.br/21806/21806_4.PDF> Acesso em: 04 de agosto de 2014.

PINGARILHO, CARLOS/FARO, LUIZ, 2007 – Artigo: Mobilidade: A grande tendência do futuro. Disponível em: http://www.teleco.com.br/promon/pbtr/Mobilidade_4Web.pdf Acessado em: 04 de agosto de 2014.

JÚNIOR, AFONSO, 2011 – ARTIGO: Dicas para criar o seu projeto móbil. Disponível em <<http://www.mobits.com.br/2011/7/20/dicas-para-criar-o-seu-projeto-mobile>> Acesso em: 04 de agosto de 2014.

NOKIA *Developer*, 2013 – Arquitetura de Aplicações multiplataforma para dispositivos móveis. Disponível em: http://developer.nokia.com/community/wiki/Arquitetura_de_aplica%C3%A7%C3%B5es_multiplataforma_para_dispositivos_m%C3%B3veis, Acessado em: 14 de agosto de 2014.

PAMPLONA, Vitor Fernando 2010. JavaFree.org: Web Services. Disponível em: <<http://javafree.uol.com.br/artigo/871485/Web-Services-Construindo-disponibilizando-e-acessando-Web-Services-via-J2SE-e-J2ME.html>> Acesso em: 04 de agosto de 2014.

PUC-RIO, Certificação Digital Nº 1012653/CA. Disponível em: http://www.maxwell.vrac.puc-rio.br/21806/21806_4.PDF, Acessado em 14 de agosto de 2014.

ROGERS, Rick; MEIKE, Blake. Android: Desenvolvimento de Aplicações Android, São Paulo: Novatec Editora Ltda, 2009.

SANT'ANNA, MAURO. SOAP E WebServices – Linha de Código. Disponível em <<http://www.linhadecodigo.com.br/artigo/38/soap-e-webservices.aspx>> Acesso em 04 de agosto de 2014.

SQLite. Disponível em <<http://sqlite.org/>> Acesso em 04 de agosto de 2014.

SOMMERVILLE, Ian. Processos de Engenharia de Requisitos. In: Engenharia de Software – 8ª Edição – São Paulo: Pearson Addison-Wesley, 2007.

VILLA, Felipe Junges. Apostila Estágio de Formação de Pregoeiros. 3ª ICFEx. 2011. Apostila Curso de Formação de Pregoeiros. ENAP, 2003.

WIGLEY, Andy & WHELLWRIGHT, Stephen. *Microsoft. NET Compact Framework – Core Reference. Microsoft Press, 2003.*