

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA JAVA

ALTIERES HERCULANO BIANCHI DE QUEIROZ

AGENDA DE CONTATOS EM SISTEMA WEB COM
SINCRONIZAÇÃO VIA WEB SERVICE PARA CELULARES
COM SISTEMA OPERACIONAL ANDROID

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA - PR
2011

ALTIERES HERCULANO BIANCHI DE QUEIROZ

AGENDA DE CONTATOS EM SISTEMA WEB COM
SINCRONIZAÇÃO VIA WEB SERVICE PARA CELULARES
COM SISTEMA OPERACIONAL ANDROID

Monografia apresentada ao Curso de Especialização em Tecnologia Java da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Especialista” - Área de Concentração: Tecnologia Java.

Orientador: Prof. Paulo M. D. Bordin

Co-orientador: Prof. João Alberto Fabro

CURITIBA - PR
2011

RESUMO

A presente monografia apresenta um levantamento sobre as ferramentas para manipulação de contatos pessoais, geralmente denominados comercialmente de agenda de contatos, articulados entre ambientes distintos existentes atualmente no mercado brasileiro, destacando suas funcionalidades principais, bem como algumas ferramentas estrangeiras para tal finalidade. Além disto, são relatadas as dificuldades e necessidades dos usuários para este perfil de aplicativo. É demonstrada a possibilidade técnica na relação entre comunicação via Web Service com dispositivos móveis, mais especificamente fazendo uso das tecnologias Android para o ambiente móvel, através do desenvolvimento de um protótipo funcional, capaz de armazenar dados do usuário em um servidor *web* e disponibilizá-los para acesso via *internet*, bem como para sincronizar os dados com um aplicativo Android desenvolvido para este fim. Além disto, o sistema desenvolvido também permite alertar os usuários quando os seus contatos atualizarem o número do celular ou fixo no sistema *web*, proporcionando um tipo de “rede social de contatos telefônicos”.

Palavras-chave: Aplicação Java. Sistema Operacional Android. Web Service. Contatos de celular.

ABSTRACT

This work presents a survey of tools for handling personal contacts (address books) using mobile smartphones. The main applications available at the Brazilian market are listed, and their key features are highlighted. Some foreign tools are also evaluated. This work also proposes a new application, for Android based mobile phones, that allows the user to export/import its personal contacts to/from a web-based system, also developed by the author. The communication is achieved through Web Services, using the Java Web Service specification. A working prototype, involving the Android application, the web-based system, and the communication protocol, is presented, and is capable of storing user data on the web server, making it available to the user via a web interface, for consult, update and management. The Android application developed allows synchronization, so in case of loss of the mobile phone, the contacts are preserved and can be downloaded for a new mobile phone. Besides that, the developed system also allows the automatic update of the contacts phone numbers, if authorized by other users using the system, providing a kind of "social network" for personal phone contacts.

LISTA DE FIGURAS

Figura 1 – Método prioritário de ligação que os respondentes utilizam.....	14
Figura 2 – Problemas referenciados pelos usuários com relação a agenda de contatos dos aparelhos	15
Figura 3 - Local onde os pesquisados guardavam seus dados da agenda de contatos antes de possuir telefone celular	15
Figura 4 - Local onde atualmente os entrevistados guardam seus dados da agenda de contatos.....	16
Figura 5 - Média de tempo em que o pesquisado troca de aparelho	16
Figura 6 - Usuários pesquisados que possuem sistema de cópia de dados (backup) em seu celular	17
Figura 7 - Disposição dos pesquisados em pagar pelo serviço de cópia de segurança dos dados da agenda de contato na operadora	17
Figura 8 – Diagrama de Arquitetura do Sistema	30
Figura 9 – Diagrama de casos de uso - Sistema <i>Web</i>	31
Figura 10 – Diagrama de casos de uso - Aplicativo Android.....	31
Figura 11 – DER (Diagrama de Entidade e Relacionamento).....	36
Figura 12 – Diagrama de Classes do Sistema <i>Web</i> - Geral.....	37
Figura 13 – Diagrama de Classes <i>Web</i> – Pacote <i>webservice</i>	38
Figura 14 – Diagrama de Classes <i>Web</i> – Pacote <i>action</i>	38
Figura 15 – Diagrama de Classes <i>Web</i> – Pacote <i>core</i>	39
Figura 16 – Diagrama de Classes <i>Web</i> – Pacote <i>dao</i>	40
Figura 17 – Diagrama de Classes <i>Web</i> – Pacote <i>vo</i>	41
Figura 18 – Diagrama de Classes <i>Web</i> – Pacote <i>util</i>	41
Figura 19 – Diagrama de Classes do Aplicativo Android - Geral.....	42
Figura 20 – Diagrama de Classes Android – Pacote <i>util</i>	42
Figura 21 – Diagrama de Classes Android – Pacote <i>android</i>	43
Figura 22 – Diagrama de Classes Android – Pacote <i>dao</i>	44
Figura 23 – Diagrama de Classes Android – Pacote <i>util</i>	44
Figura 24 – Diagrama de Sequência Cadastrar Usuário	45
Figura 25 – Diagrama de Sequência Validar Usuário	45
Figura 26 – Diagrama de Sequência Visualizar Notificação.....	46

Figura 27 – Diagrama de Sequência Executar Notificação	47
Figura 28 – Diagrama de Sequência Visualizar Atualização	48
Figura 29 – Diagrama de Sequência Executar Atualização	48
Figura 30 – Diagrama de Sequência Sincronizar Contatos Android	49
Figura 31 – Diagrama de Sequência Sincronizar Contatos <i>Web</i>	50
Figura 32 – Tela do sistema para cadastro de novo usuário.....	51
Figura 33 – Tela do sistema para cadastro de novo contato.....	52
Figura 34 – Tela do sistema para visualizar a lista de notificações.....	52
Figura 35 – Tela do sistema para visualizar os dados da notificação	53
Figura 36 - Tela do sistema para visualizar a lista de atualizações.....	54
Figura 37- Tela do sistema para visualizar os dados da atualização	54
Figura 38 - Tela inicial do aplicativo Android com o <i>menu</i> de opções.....	55
Figura 39 - Tela do aplicativo para executar a sincronização dos contatos entre o aparelho Android e o sistema <i>web</i>	56

SUMÁRIO

1 INTRODUÇÃO	9
1.1 TEMA	9
1.2 PROBLEMA	9
1.3 JUSTIFICATIVA	10
1.4 OBJETIVO GERAL	10
1.5 OBJETIVOS ESPECÍFICOS	11
1.6 ESTRUTURA DO TRABALHO.....	12
2 ESTUDO PRELIMINAR	13
2.1 PRINCIPAIS SERVIÇOS PAGOS DE ARMAZENAMENTO E RECUPERAÇÃO DE CONTATOS TELEFÔNICOS	18
2.1.1 Claro Contatos	18
2.1.2 Oi Agenda	19
2.1.3 TIM Agenda.....	19
2.1.4 Vivo Agenda.....	19
2.2 SOLUÇÕES GRATUITAS PARA ARMAZENAMENTO E RECUPERAÇÃO DE CONTATOS TELEFÔNICOS	20
2.2.1 Contails	20
2.2.2 Google Sync.....	20
2.2.3 Microsoft My Phone	21
2.2.4 MIQ	21
3 REFERENCIAL TEÓRICO	22
3.1 JAVA	22
3.2 ANDROID.....	23
3.3 APACHE TOMCAT	24
3.4 POSTGRESQL.....	25
3.5 WEB SERVICE	26
4 PROJETO LÓGICO	28
4.1 REQUISITOS NÃO FUNCIONAIS	28
4.1.1 Sistema web.....	28
4.1.2 Aplicativo para dispositivos móveis com o sistema operacional Android	29
4.2 REQUISITOS FUNCIONAIS	29
4.3 DIAGRAMA DE ARQUITETURA.....	30
4.4 DIAGRAMA DE CASOS DE USO – SISTEMA WEB	31
4.5 DIAGRAMA DE CASOS DE USO - APLICATIVO ANDROID	31
4.6 CASO DE USO CADASTRAR USUÁRIO	32

4.7	CASO DE USO VALIDAR USUÁRIO.....	32
4.8	CASO DE USO VALIDAR CONTATO.....	33
4.9	CASO DE USO VISUALIZAR NOTIFICAÇÃO	33
4.10	CASO DE USO EXECUTAR NOTIFICAÇÃO	34
4.11	CASO DE USO VISUALIZAR ATUALIZAÇÃO.....	34
4.12	CASO DE USO EXECUTAR ATUALIZAÇÃO	35
4.13	CASO DE USO SINCRONIZAR CONTATOS.....	35
4.14	DER – DIAGRAMA DE ENTIDADE E RELACIONAMENTO.....	36
4.15	DIAGRAMA DE CLASSES SISTEMA WEB - GERAL.....	37
4.16	DIAGRAMA DE CLASSES WEB – PACOTE WEBSERVICE	38
4.17	DIAGRAMA DE CLASSES WEB – PACOTE ACTION	38
4.18	DIAGRAMA DE CLASSES WEB – PACOTE CORE.....	39
4.19	DIAGRAMA DE CLASSES WEB – PACOTE DAO	40
4.20	DIAGRAMA DE CLASSES WEB – PACOTE UTIL	41
4.21	DIAGRAMA DE CLASSES WEB – PACOTE UTIL	41
4.22	DIAGRAMA DE CLASSES APLICATIVO ANDROID - GERAL.....	42
4.23	DIAGRAMA DE CLASSES ANDROID – PACOTE UTIL.....	42
4.24	DIAGRAMA DE CLASSES ANDROID – PACOTE ANDROID	43
4.25	DIAGRAMA DE CLASSES ANDROID – PACOTE DAO.....	44
4.26	DIAGRAMA DE CLASSES ANDROID – PACOTE VO	44
4.27	DIAGRAMA DE SEQUÊNCIA CADASTRAR USUÁRIO.....	45
4.28	DIAGRAMA DE SEQUÊNCIA VALIDAR USUÁRIO.....	45
4.29	DIAGRAMA DE SEQUÊNCIA VISUALIZAR NOTIFICAÇÃO.....	46
4.30	DIAGRAMA DE SEQUÊNCIA EXECUTAR NOTIFICAÇÃO	47
4.31	DIAGRAMA DE SEQUÊNCIA VISUALIZAR ATUALIZAÇÃO	48
4.32	DIAGRAMA DE SEQUÊNCIA EXECUTAR ATUALIZAÇÃO.....	48
4.33	DIAGRAMA DE SEQUÊNCIA SINCRONIZAR CONTATOS ANDROID	49
4.34	DIAGRAMA DE SEQUÊNCIA SINCRONIZAR CONTATOS <i>WEB</i>	50
4.35	RESULTADOS.....	51
4.35.1	Tela Novo Usuário	51
4.35.2	Tela Cadastrar Contato.....	51
4.35.3	Tela Notificações.....	52
4.35.4	Tela Dados da Notificação	53
4.35.5	Tela Atualizações.....	53
4.35.6	Tela Dados da Atualização	54
4.35.7	Tela inicial do aplicativo Android.....	55
4.35.8	Tela Sincronizar Contatos	56

5 CONCLUSÃO	57
REFERÊNCIAS BIBLIOGRÁFICAS	58

1 INTRODUÇÃO

Neste capítulo são apresentados os objetivos da presente monografia, o problema a ser solucionado, bem como as informações referentes à solução proposta.

1.1 TEMA

Este trabalho apresenta um estudo sobre soluções existentes para armazenamento de listas de contatos em dispositivos móveis, como celulares e *smartphones*, e propõe o desenvolvimento de um protótipo de agenda de contatos como um sistema *web*, que pode ser acessado via *internet* e permite a sincronização via Web Service com aplicativo para *smartphones* com sistema operacional Android.

1.2 PROBLEMA

O mundo moderno exige cada vez mais das pessoas processar e absorver um número maior de informações, e para isto, surgem cada vez mais novas tecnologias visando facilitar a vida e liberá-las dessa obrigação (ABREU, 2005). Uma dessas tecnologias é o celular, pois permite o contato imediato, independente da distância ou barreiras físicas entre os interlocutores, ou mesmo a mobilidade destes. A capacidade de imediata e fácil conexão ganhou adeptos em todo mundo, pois permitiu aos usuários manter constante contato com suas redes sociais, contatos familiares ou mesmo profissionais (NICOLACI-DA-COSTA, 2006).

A agenda de contatos é uma das funções mais utilizada pelos usuários de aparelhos de celular. Se os usuários conseguirem usar com efetividade a agenda de contatos do seu celular, não terão mais a necessidade de memorizar tantos números diferentes, mesmo os que mais utilizem (ABREU, 2005).

Mas para que a agenda de contatos tenha sua funcionalidade utilizada de forma a facilitar a vida dos usuários, é imprescindível que os dados armazenados estejam sempre atualizados e disponíveis. Porém nem sempre isto ocorre, pois muitas vezes as pessoas trocam de número e não tem tempo ou disponibilidade para comunicar todos os seus contatos. Por exemplo, se for ligar para todos ou enviar mensagem de texto, isto demandaria tempo e custo para o usuário.

Outro exemplo é quando há a troca ou a perda do aparelho. Se os contatos foram gravados no chip, numa eventual troca de aparelho, ainda há como recuperar os contatos. Porém isso não seria possível numa eventual perda ou dano no aparelho. Levando em consideração essa hipótese, não havendo com recuperar os contatos do celular antigo ou não havendo como transferi-los para o novo aparelho, o proprietário terá que redigitar seus contatos um de cada vez. Isto ainda pode se agravar se o aparelho não possuir uma boa usabilidade, tornando essa tarefa ainda mais trabalhosa.

Para Abreu (2005), em sua pesquisa, os consumidores levam em consideração na hora de adquirir um novo celular a importância do seu benefício, depois a familiaridade e em terceiro a facilidade de uso. A pesquisa também revelou que 58% dos entrevistados gostariam que o aparelho possuísse cópia dos dados da agenda de contatos, porém metade não pagaria por este serviço.

1.3 JUSTIFICATIVA

Analisando os cenários apresentados anteriormente é possível perceber o quanto é difícil manter os contatos sempre disponíveis e mantê-los atualizados, o que justifica o desenvolvimento de um sistema que permita simplificar o armazenamento, a alteração e a realização de cópias de segurança dos dados de contatos dos aparelhos de comunicação.

1.4 OBJETIVO GERAL

Realizar levantamento sobre algumas ferramentas para manipulação de contatos pessoais e desenvolver um protótipo funcional que solucione os problemas mencionados. Os principais objetivos são:

- Contatos sempre disponíveis na web;
- Sincronização dos contatos do *smartphone* com o sistema *web*;
- Facilitar manipulação dos dados dos contatos através da página *web*;
- Associação entre os usuários e os contatos cadastrados.
- Tirar do usuário a responsabilidade de informar a alteração do seu número, tornando o processo automático através do sistema.

1.5 OBJETIVOS ESPECÍFICOS

A presente monografia tem como objetivo auxiliar as pessoas nos problemas apresentados, pois propõe desenvolver um protótipo de aplicação *web* no qual o usuário poderá cadastrar os seus contatos numa base única e poderá acessá-los de computadores conectados à *internet*. Além disso, propõe desenvolver um protótipo de aplicativo para *smartphones* com sistema operacional Android, o qual terá a opção de cadastrar e editar contatos e sincronizar com a aplicação *web* usando a solução Web Service, permitindo manter os dados atualizados, tanto no site como no aplicativo do *smartphone*. Há também a facilidade de receber uma notificação quando um contato alterar seu número de celular ou fixo, dando ao usuário a opção de atualizar o número alterado.

Através do aplicativo Android o usuário poderá cadastrar, editar, listar e excluir contatos, ficando estes armazenados no aparelho. A sincronização não é automática. Para sincronizar os contatos com o sistema *web*, o usuário deve acessar a opção de sincronização no aplicativo, o qual solicita a autenticação através do login e senha do sistema *web*. Se autenticado com sucesso, o aplicativo envia e busca as informações a serem sincronizadas através do Web Service, informando se a sincronização ocorreu com sucesso ou ocorreu alguma falha. A sincronização só pode ser feita através do aplicativo Android.

Através do sistema *web*, o usuário poderá cadastrar, editar, listar e excluir seus contatos. Também tem a opção de relacionar seus contatos com usuários cadastrados. Exemplo: Usuário A se cadastra no sistema. Usuário A cadastra um número de celular para o Contato A. Usuário B se cadastra no sistema. Sistema identifica que Usuário A tem o celular do Usuário B cadastrado como contato. Sistema exibe notificação para o Usuário B informando que Usuário A cadastrou seu celular como contato. Sistema exibe a opção de adicionar Usuário A como seu contato. Se Usuário B optar por adicionar como contato, o sistema busca as informações do Usuário A e o adiciona como contato do Usuário B.

Após inserir os dados do contato, o sistema associa o Usuário B ao contato do Usuário A, para que quando o Usuário B alterar seu número de telefone fixo ou celular, o sistema possa exibir um alerta para o Usuário A informando que o Usuário B atualizou seus dados.

Após Usuário B adicionar Usuário A como contato, o sistema identifica o relacionamento e informa ao Usuário A que o Usuário B tem seu celular cadastrado como contato. Neste caso, o sistema exibe para o Usuário A opção de permitir que suas atualizações sejam enviadas ao Usuário B e o mesmo possa manter seu contato atualizado.

O sistema identifica as atualizações e informa aos usuários, porém a atualização só é realizada após o usuário confirmar que deseja atualizar seu contato. O mesmo tem a opção excluir a atualização sem executá-la.

Tanto no sistema *web* como no aplicativo Android, os contatos adicionados, editados e excluídos serão disponibilizados para sincronização.

1.6 ESTRUTURA DO TRABALHO

No capítulo 2, é apresentado um estudo sobre algumas tecnologias que se propõem a resolver o problema mencionado e breve descrição das suas características. No capítulo 3 é apresentado um estudo sobre as técnicas utilizadas no desenvolvimento da solução proposta. No capítulo 4 são apresentados os requisitos do protótipo a ser desenvolvido. No capítulo 5 é apresentado o projeto lógico, contendo os diagramas para documentação do protótipo. No capítulo 6 é apresentada a conclusão do trabalho e as propostas para trabalhos futuros.

2 ESTUDO PRELIMINAR

O celular marca presença em vários setores da sociedade e teve impressionante crescimento desde sua entrada no Brasil, em 1990. Dessa data até dezembro de 2003, aumentou de 667 mil para 43 milhões de aparelhos móveis, ultrapassando nesse mesmo ano o número de usuários de telefonia fixa (ABREU, 2005). Em maio de 2009, já eram 157 milhões de telefones celulares habilitados, enquanto os fixos somavam 42 milhões (GUEDES e VASCONCELLOS, 2009).

Aos poucos a tecnologia de comunicação móvel foi sendo introduzida na vida das pessoas nas mais variadas situações, como nos contatos pessoais e em áreas de negócio. Aos poucos foi deixando de ser artigo de luxo para se tornar item básico, atingindo a sociedade moderna brasileira e criando forte dependência com essa tecnologia (ABREU, 2005).

O celular atingiu todas as faixas etárias (RESENDE, 2010), porém tiveram maior absorção pelos usuários mais jovens (NICOLACI-DA-COSTA, 2006), se tornando um instrumento para auxílio na organização das tarefas e planejamento das atividades dos jovens, bem como desenvolver a maturidade e autonomia (VERZA, 2008).

A perda ou quebra de aparelhos são dois dos grandes motivos que levam as pessoas a trocarem frequentemente de aparelho. Segundo uma pesquisa européia, 4,5 milhões de aparelhos são perdidos ou quebrados todos os anos no Reino Unido. Segundo o mesmo estudo, uma média de 810 mil aparelhos são esquecidos em bares e 540 mil em taxis e ônibus. A pesquisa levou em consideração apenas um país, mas a realidade não é muito diferente nas demais regiões do mundo (BAIO, 2008).

Os principais dados que os donos de celulares desejam gravar no novo aparelho, após trocá-los, são os contatos armazenados no aparelho antigo. A tarefa de redigitar todos os contatos no novo celular é, sem dúvida, a principal causa da insatisfação dos usuários. Numa pesquisa realizada por Abreu (2005) com usuários de aparelho de celular, 53% disseram utilizar a agenda do celular para ligar para um contato, como mostra a Figura 1. Quando questionados sobre os problemas da agenda de contatos, 34% apontaram a perda dos dados armazenados e 20% apontaram a migração para outro aparelho, conforme Figura 2. Sobre o local onde eram armazenados os contatos antes do celular, 75% responderam que

armazenavam em caderno telefônico, mas 71% responderam que também mantinham os contatos em uma agenda eletrônica (além do caderno), como mostrado na Figura 3. 93% responderam que hoje armazenam os dados dos seus contatos no telefone móvel, conforme Figura 4. Ao serem perguntados sobre como passavam os dados de um celular antigo para um novo, 53% disseram digitar os nomes um a um, e houve outras respostas como trocar o chip, digitar apenas alguns e digitação feita por outra pessoa. Sobre a média de tempo para troca de aparelho, 72% disseram trocar o aparelho no máximo em um ano, conforme a Figura 5. 75% dos entrevistados responderam não possuir sistema de cópia de segurança dos dados, conforme Figura 6. Ao serem perguntados se pagariam para que uma cópia dos dados de sua agenda de contatos ficasse armazenada na operadora, para ter a opção de recuperá-la em caso de troca, perda, roubo ou defeito do aparelho, metade não pagaria para ter esse recurso, como mostra a Figura 7.

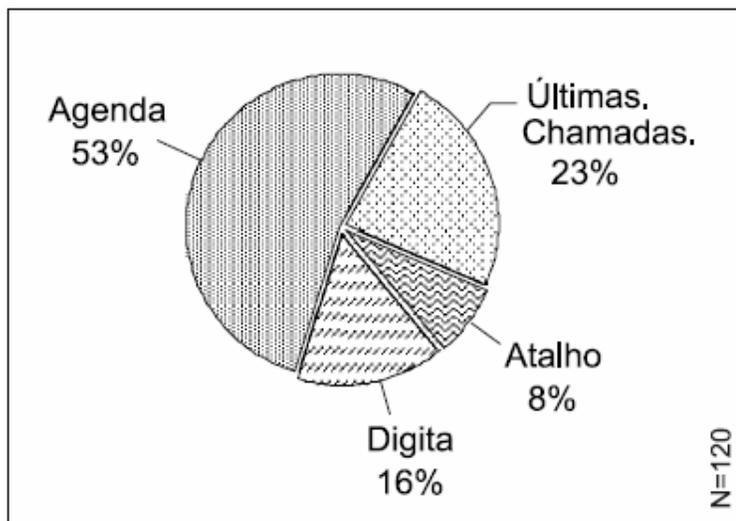


Figura 1 – Método prioritário de ligação que os respondentes utilizam
Fonte: Abreu (2005).

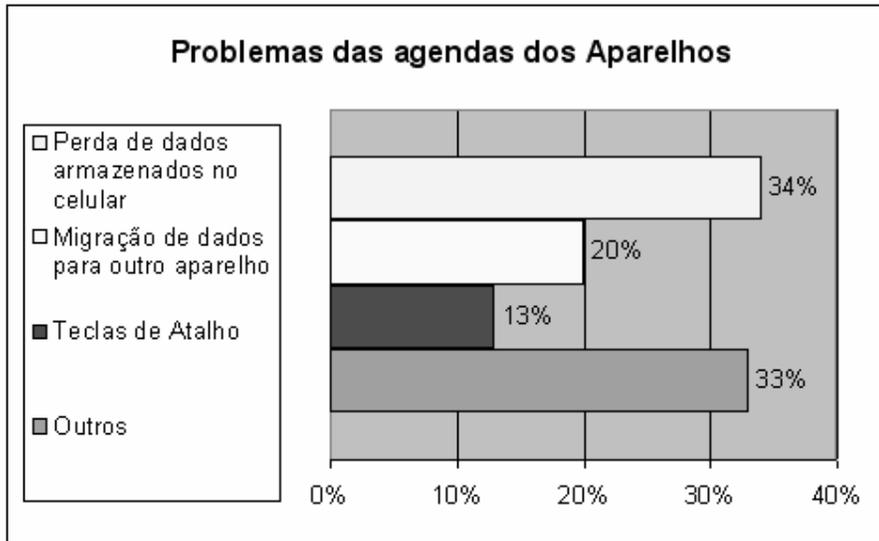


Figura 2 – Problemas referenciados pelos usuários com relação a agenda de contatos dos aparelhos
 Fonte: Autoria própria.

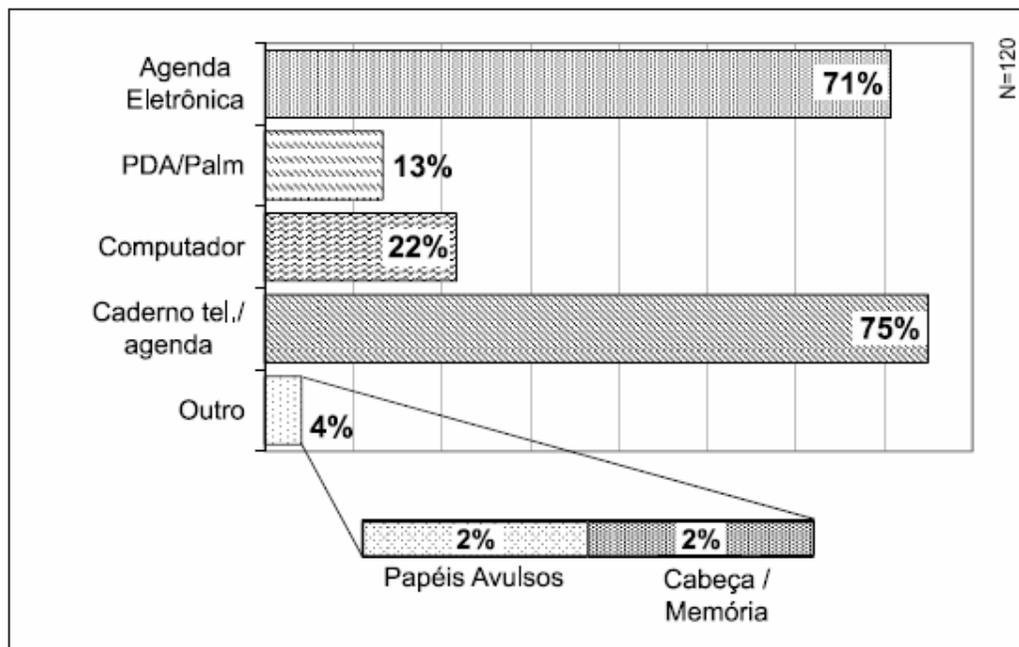


Figura 3 - Local onde os pesquisados guardavam seus dados da agenda de contatos antes de possuir telefone celular
 Fonte: Abreu (2005).

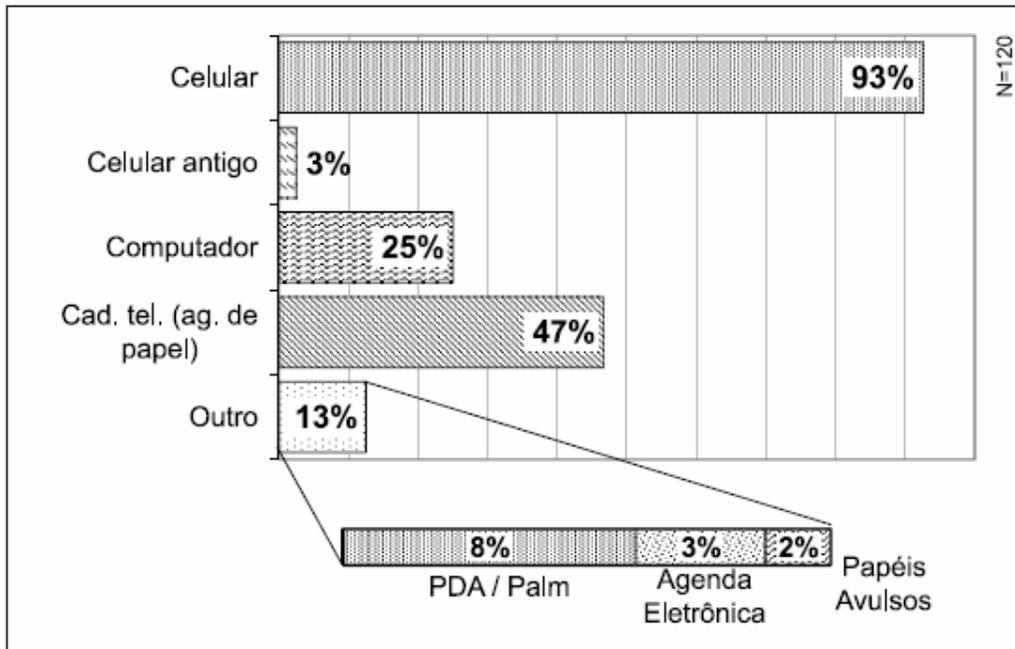


Figura 4 - Local onde atualmente os entrevistados guardam seus dados da agenda de contatos
Fonte: Abreu (2005).

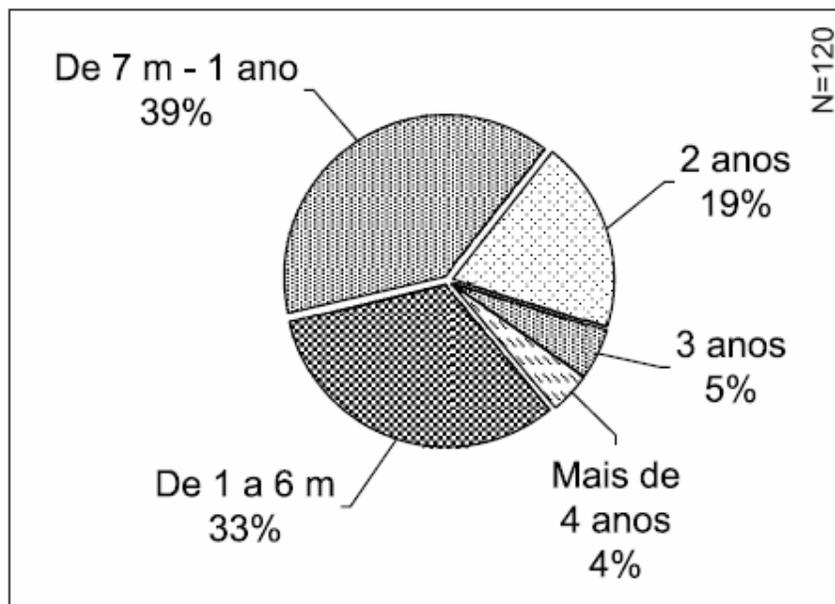


Figura 5 - Média de tempo em que o pesquisado troca de aparelho
Fonte: Abreu (2005).

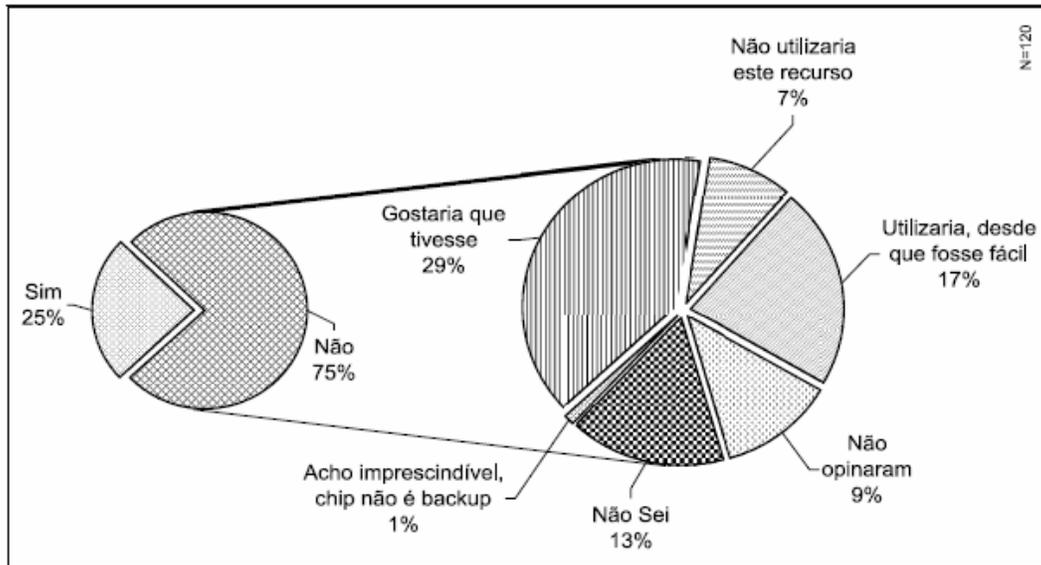


Figura 6 - Usuários pesquisados que possuem sistema de cópia de dados (backup) em seu celular
Fonte: Abreu (2005).

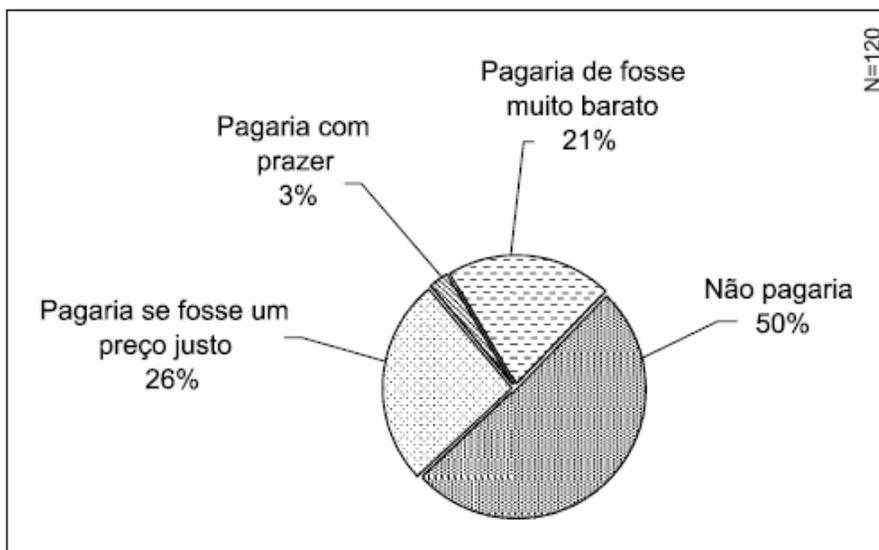


Figura 7 - Disposição dos pesquisados em pagar pelo serviço de cópia de segurança dos dados da agenda de contato na operadora
Fonte: Abreu (2005).

Desde o surgimento dos aparelhos de celular, foram criadas muitas soluções buscando resolver o problema do armazenamento dos contatos pessoais e a transferência desses dados para outro aparelho, mas nenhuma se tornou popular. Isto pode comprovar-se na maneira como cada aparelho gerencia os dados nele contidos. Para os usuários pesquisados, a transferência dos dados para um novo aparelho é complexa, como mostrado anteriormente na Figura 2.

Como solução para recuperação dos dados da agenda de contatos contidos nos aparelhos, as operadoras brasileiras passaram a oferecer serviços de sincronização e backup de dados. Em todos os serviços disponíveis, o funcionamento é parecido: é necessário realizar um cadastro no site da operadora e pagar uma taxa para adquirir o serviço. Sempre que houver uma nova atualização - como acrescentar um novo contato - uma nova cobrança é feita (BAIO, 2008).

As soluções apresentadas a seguir, atendem a proposta por eles oferecida, porém não atendem plenamente todas as necessidades dos usuários. As soluções das operadoras, por serem pagas, garantem uma maior segurança no armazenamento e na disponibilização dos dados, porém geram custos para o cliente. Já as soluções gratuitas não atendem completamente as necessidades dos usuários, pois não atendem a todos os aparelhos. As soluções existentes, pagas e gratuitas, atendem somente um usuário específico, não havendo relação entre os dados armazenados desse usuário com os dados de outro usuário, que é uma das propostas desta monografia.

2.1 PRINCIPAIS SERVIÇOS PAGOS DE ARMAZENAMENTO E RECUPERAÇÃO DE CONTATOS TELEFÔNICOS

A seguir, uma breve descrição dos serviços pagos disponibilizados pelas quatro principais operadoras de telefonia celular no Brasil, organizados em ordem alfabética.

2.1.1 Claro Contatos

Segundo Costa (2011), o serviço permite fazer um backup e sincronizar a agenda de contatos do chip Claro com o site do Claro Idéias. Permite manter uma cópia de segurança da agenda de contatos, a qual pode ser atualizada e restaurada quando o usuário desejar. Para isto, o usuário deve acessar o site Claro Contatos pelo computador ou pelo *menu* Claro Contatos do celular. Toda comunicação do serviço é feita por Torpedo. Para realizar a primeira sincronização, o usuário deve pagar uma taxa de R\$ 4,00. Para as demais sincronizações parciais, o valor é de R\$ 0,50 cada. E, se for necessário uma restauração completa, o usuário deverá pagar uma taxa no valor de R\$ 4,00.

2.1.2 Oi Agenda

Segundo o site da Oi (2011), com o Oi agenda é possível armazenar a agenda de contatos do Oi Chip no servidor da operadora. Depois, caso necessário, basta acessar o site da operadora e recuperar toda a agenda telefônica. O serviço custa R\$ 2,49 por sincronização ou recuperação da agenda.

2.1.3 TIM Agenda

Segundo o site da TIM (2011), o TIM Agenda faz uma cópia segura dos contatos salvos no TIM Chip. No caso de perda, roubo ou troca de aparelho, os contatos ficam salvos na internet. Para recuperá-los ou mantê-los atualizados, basta restaurar os contatos salvos na área de auto-atendimento do site TIM ou ligar para a central de atendimento e solicitar a restauração. A ação de sincronização salva todas as alterações e novos contatos incluídos no TIM Agenda. Também é possível atualizar os contatos sempre que desejar ou programar a atualização a cada contato adicionado ou a cada cinco ou dez contatos adicionados. O serviço custa R\$ 3,99 + impostos para a primeira sincronização. Demais atualizações custam R\$ 0,49 + impostos cada. O valor da restauração dos dados é de R\$ 3,99 + impostos.

2.1.4 Vivo Agenda

Segundo o site da Vivo (2011), o Vivo Agenda armazena com segurança os contatos do celular no servidor da Vivo, para que eles estejam à disposição em qualquer lugar ou aparelho. Também é possível consultar e editar os contatos pela Internet através da opção Meu Vivo. Alterações feitas no site também se refletirão no aparelho. O valor do serviço, para clientes GSM (*Groupe Special Mobile*) é de R\$ 2,99 para a primeira sincronização e as demais custam R\$ 0,49 cada. Para recuperação dos contatos é cobrado o valor de R\$ 2,99. Para clientes CDMA (*Code Division Multiple Access*), o preço é de R\$ 1,99 por mês mais o valor do tráfego utilizado para sincronizar a agenda, que pode variar conforme plano o contratado.

2.2 SOLUÇÕES GRATUITAS PARA ARMAZENAMENTO E RECUPERAÇÃO DE CONTATOS TELEFÔNICOS

A seguir são apresentadas algumas soluções gratuitas para armazenamentos dos contatos. Apesar de serem serviços gratuitos, quando houver transferência de dados, as operadoras podem cobrar por esse serviço. Os mesmos atendem a finalidade proposta, porém não atendem todos os tipos de aparelhos. São apresentados em ordem alfabética.

2.2.1 Contails

Contails para Polo (2009) é um facilitador. Após o cadastro, o site solicita o número de celular e o país (Brasil), podendo cadastrar vários celulares caso desejado. Depois de informar os dados, o sistema envia um SMS para verificar a autoria do número, instalando um sistema de sincronização que, depois de executado, enviará os dados dos contatos ao banco de dados online do Contails.

Os contatos ficam guardados online e o sistema detecta os números duplicados, oferecendo ao usuário a possibilidade de unificar os dados para sincronizar novamente com o celular. Após gravar os dados, é possível adicionar os contatos de outro telefone para unificá-los.

2.2.2 Google Sync

O Google Sync é um serviço para sincronia de compromissos e contatos oferecido pela Google. Com este serviço é possível sincronizar os dados do celular com a web usando o próprio Google. Não há a necessidade de baixar nenhum aplicativo para usar o serviço, bastando apenas configurar os recursos disponíveis no próprio aparelho. O serviço sincroniza contatos e compromissos (da Google Agenda), porém não sincroniza as tarefas armazenadas. A sincronia de e-mail funciona somente para iPhones e *smartphones* com Windows Mobile, ficando de fora os aparelhos com Symbian. Para sincronização é utilizado o protocolo Microsoft Exchange ActiveSync na maioria dos aparelhos. Ao configurar uma nova conta no aparelho, poderão ser removidos os dados da conta antiga (COSTA, 2010).

2.2.3 Microsoft My Phone

O My Phone é um serviço gratuito disponibilizado pela Microsoft. Com ele é possível efetuar cópia de segurança do celular automaticamente, compartilhar fotografias em sites de rede social, acessar os contatos, mensagens SMS, entre outros, online e gratuitamente, localizar o aparelho de celular perdido.

O aplicativo My Phone pode ser usado com telefones Windows Mobile 6 ou superior e é pré-carregado na maioria dos telefones Windows mais recentes. Se o software ainda não estiver no telefone, inserir <http://myphone.microsoft.com/install> no navegador do celular para baixar o software.

Depois de baixar o software no telefone, selecionar o aplicativo Microsoft My Phone no *menu* Programas ou Configurações para concluir a instalação. Após a sincronização, Quando desejar acessar os seus contatos, fotos, mensagens de texto entre outros, basta acessar o site do My Phone (MICROSOFT, 2009).

2.2.4 MIQ

O mIQ é um serviço gratuito que permite integração com plataformas diferentes: Windows Mobile, BlackBerry, Symbian e Android, o que engloba vários modelos de operadoras diferentes. O mIQ sincroniza o conteúdo de um *smartphone* na internet sem que o usuário tenha que se preocupar com o procedimento. São armazenados os contatos, mensagens de texto, ligações efetuadas e recebidas calendário, fotos e vídeos. Caso haja troca de aparelho, é possível transferir os dados para o novo aparelho com facilidade. Além de gerenciar contatos, o mIQ permite compartilhar imagens em redes sociais, enviar e receber mensagens de texto entre outras opções (MIQ, 2011).

3 REFERENCIAL TEÓRICO

Para desenvolvimento deste protótipo foram utilizadas as tecnologias Java, Android, Web Service, Apache Tomcat e PostgreSQL, as quais serão apresentadas a seguir.

3.1 JAVA

Para Serson (2007), o Java é uma linguagem de programação eficiente, versátil, segura e independente de plataforma. Foi criada para ser uma ferramenta de programação, como parte de um pequeno trabalho da Sun Microsystems, em 1991, por um grupo de 13 pessoas liderado por James Gosling. Tentavam planejar uma novidade na computação, para impulsionar o futuro dos dispositivos digitais. Mas a idéia acabou não dando certo, pois o controlador de dispositivos que foi criado estava à frente da tecnologia da época. Porém a *internet* estava preparada para essa nova tecnologia. Em 1995, foi lançado o navegador Netscape que tinha Java incorporado. Até 2007, o Java já havia atraído mais de quatro milhões de desenvolvedores e passou a ser usada nos principais setores pelo mundo e se hoje encontra presente na internet, laptops, telefones celulares, supercomputadores científicos, dispositivos para jogos, cartões de crédito, e em mais uma infinidade de soluções.

A linguagem Java foi desenvolvida para ser multiplataforma e rodar em uma maquina virtual que interpreta o código compilado, o que permite que qualquer programa rode em sistemas operacionais que possuam uma máquina virtual instalada.

Com o passar do tempo, a linguagem Java foi sendo ampliada, testada, refinada e experimentada por milhões de desenvolvedores de software, o que a tornou extremamente robusta, madura e extremamente versátil. Permite ao desenvolvedor criar um software em uma plataforma e executá-la em praticamente todas as outras, desenvolver aplicativos do lado do servidor, criar programas para execução em navegadores e serviços da web, combinar aplicativos ou serviços com o Java e criar serviços ou aplicativos altamente personalizados, criar aplicativos para celulares, processadores remotos e dispositivos com tecnologia digital, entre muitas outras soluções.

Para desenvolvimento deste protótipo, optou-se pelo Java por ser uma linguagem de programação multiplataforma, orientada a objetos, possuir código aberto e possuir extensa gama de documentação disponível.

3.2 ANDROID

Para Rogers (2009), Android é uma plataforma para celulares desenvolvida pela Google. Em 2007 foi lançada a primeira versão beta do SDK (*Software Development Kit*, ou Conjunto para Desenvolvimento de Sistema). Em poucos meses, mais de um milhão de pessoas haviam baixado as versões do SDK no *site* da Google.

O Android busca solucionar alguns problemas dos *smartphones* atuais, como: Fragmentação, ou seja, cada marca tem um ambiente de aplicação diferente. *Software Stack* proprietários, ou seja, os aparelhos têm o código fechado, permitindo apenas aos fabricantes adicionar um *driver*, por exemplo, ao aparelho. Redes fechadas, ou seja, as aplicações de terceiros são bloqueadas pelas operadoras de dispositivos móveis. As aplicações Android são um pouco diferentes das aplicações *desktop* e servidor. Essas diferenças são baseadas em conceitos exclusivos da telefonia móvel juntamente com os objetivos do Google para Android.

O Android define quatro tipos básicos de componentes para sua arquitetura: Atividades, ou seja, trechos de códigos executáveis instanciados pelo sistema ou usuário, ou executadas quando necessário. Serviços, ou seja, trechos de códigos executáveis, que geralmente são executados em segundo plano. Receptores de *Broadcast* (do Inglês, "transmitir") e Intenção, que respondem a solicitações de serviço de outra aplicação. Provedores de conteúdo, criados para compartilhar dados com outras Atividades ou Serviços.

O Android fornece um ambiente de usuário bastante flexível, uma vez que disponibiliza componentes para ser utilizado em aplicações, não sendo necessário reescrever o mesmo.

Para desenvolvimento deste protótipo, optou-se por esta tecnologia por estar presente na maioria dos *smartphones* lançados atualmente no mercado, oferecer os recursos necessários para que o protótipo possa ser instalado e utilizado pelo usuário e oferecer alguns componentes prontos que serão necessários ao desenvolvimento do aplicativo.

3.3 APACHE TOMCAT

Para Scherer (2002), o servidor web é um programa que roda em um computador que esteja conectado a *intranet* ou *internet*. É responsável por tornar as páginas *web* disponíveis para que sejam acessadas pelos usuários em um navegador de *internet*. Quando recebe uma requisição, realiza a busca pelo documento HTML (*HyperText Markup Language*, ou Linguagem de Marcação de Hipertexto) e o envia para o navegador que fez a solicitação.

São quatro as funções básicas de um servidor web: disponibilizar páginas web, rodar programas de interconexão e retornar seus resultados, realizar o controle de acesso ao servidor e realizar monitoração e fazer registros de estatísticas de acessos ao servidor. O modo de implementação dessas funções podem variar de um servidor para outro.

O Tomcat foi lançado em 1999, foi desenvolvido com o projeto Jakarta liderado pela Apache Software Foundation. O projeto Jakarta foi desenvolvido com o objetivo de produzir implementações de nível mundial das especificações de JSP (*Java Server Pages*) e *servlet* (do inglês, componente), e ao mesmo tempo tinha o objetivo de contar com a colaboração de pessoas interessadas em participar do desenvolvimento e aperfeiçoar o software, por ser uma plataforma aberta.

O Tomcat é um servidor de aplicações Java para web que implementa as especificações básicas de JSP e *servlets* desenvolvidas pela Sun. Inclui o servidor HTTP (*Hypertext Transfer Protocol*, ou Protocolo de Transferência de Hipertexto) baseado em Java e, por esse motivo, pode ser executado como um software independente. Possui também um módulo conector que suporta integração com outros servidores web, como o Apache e o Information Server da Microsoft. Possui independência de plataforma, pois foi desenvolvido em Java e possui as características de um programa desenvolvido nessa linguagem.

Para desenvolvimento deste protótipo, optou-se por esta tecnologia por ser gratuita, possuir código aberto, ser desenvolvida em Java e por implementar as especificações para JSP e *servlets*.

3.4 POSTGRESQL

Segundo Almeida (2004), o banco de dados PostgreSQL tem seu desenvolvimento derivado do SGBD (Sistema Gerenciador de Banco de Dados) Ingres e é um aplicativo de código aberto. O Ingres foi desenvolvido em 1977 na Universidade de Berkeley, na Califórnia. Em 1986 iniciou-se o desenvolvimento de um SGBD objeto-relacional sobre o Ingres, chamado inicialmente de Postgres, sendo chamado a partir de 1996 de PostgreSQL, que se tornou o mais avançado SGBD de código aberto.

O PostgreSQL, por possuir seu código aberto, possui fatores importantes na qualidade e na independência tecnológica. Estes fatores são defendidos pela comunidade de código aberto. No fator qualidade, pode-se observar uma constante preocupação em continuar o desenvolvimento, bem como corrigir defeitos, em comprometimento com o restante da comunidade. A qualidade dos produtos desenvolvidos pode ser observada pelo envolvimento de grandes empresas, seja no desenvolvimento como na utilização de um software livre. No fator independência tecnológica, a comunidade tem por objetivo fazer com que o PostgreSQL abranja o maior número possível de plataformas e sistemas operacionais.

O PostgreSQL é considerado, Dentre os SGBDs de código aberto, o mais robusto, pois possui suporte completo as especificações da SQL (*Structured Query Language*, ou Linguagem de Consulta Estruturada) desde as versões mais antigas, o que demonstra uma maior maturidade em comparação com outros SGBDs de código aberto.

Dentre as principais características do PostgreSQL, podem-se destacar: SGBD objeto-relacional (possui características de orientação a objetos), Integridade referencial, multi-plataforma (pode ser utilizado em diversos sistemas operacionais), grande diversidade de interfaces, suporte a linguagens procedurais e controle de concorrência.

Para o protótipo proposto neste trabalho, optou-se pelo PostgreSQL por possuir código aberto, ser gratuito, suportar um grande volume de dados simultaneamente e possuir grande gama de documentação disponível para consulta.

3.5 WEB SERVICE

Segundo Abinader Neto e Lins (2006), diversas tecnologias surgiram nos últimos anos destinadas ao desenvolvimento de aplicações para a Internet. Estas tecnologias possuem suas particularidades, o que impede a troca de informações de forma automatizada. Para possibilitar a comunicação entre estas aplicações heterogêneas, criou-se a solução Web Service e a linguagem XML (*Extended Markup Language*). Assim, independentemente de quais sejam as tecnologias envolvidas, a comunicação entre elas será feita em formato XML.

Tecnicamente, a solução Web Service é constituída de um sistema de baixo acoplamento, reutilizável e com componentes de fácil acesso pela *Internet*. O transporte dos dados é realizado geralmente via protocolo HTTP (o padrão não determina o protocolo de transporte). Os dados são transferidos no formato XML, encapsulados pelo protocolo SOAP (*Single Object Access Protocol*), que permite que os diferentes protocolos existentes sejam facilmente interpretados.

Algumas características podem ser destacadas como:

- Custo reduzido, pois como o modelo permite um alto grau de reutilização de componentes que podem ser acessados via rede, podem ser atingidas significativas reduções nos custos de desenvolvimento.
- Alto desempenho, pois as aplicações podem distribuir a carga de processamento entre diversos servidores e serem executadas em paralelo.
- Colaboração, pois as várias aplicações podem ser conectadas a mecanismo de computação distribuída através de padrões.
- Alta confiabilidade e disponibilidade, pois os servidores e as aplicações podem ser disponibilizados em conjuntos redundantes (conjunto de computadores que respondem de modo alternado a uma solicitação, dividindo a carga entre eles de modo que se um falhar haverá outros capazes de realizar a mesma solicitação, são os chamados *clusters*).
- Escalabilidade, que pode ser alcançada disponibilizando-se componentes distribuídos, reutilizáveis em servidores poderosos.

- Extensibilidade, que pode ser executada através da capacidade dinâmica de configuração e reconfiguração de uma aplicação distribuída entre os recursos disponíveis na rede.
- Alta produtividade e curto ciclo de desenvolvimento, pois quebra uma grande aplicação em componentes pequenos, os quais podem ser desenvolvidos de forma isolada por equipes pequenas.
- Reutilização, pois os componentes distribuídos podem executar diversos serviços, aos quais podem ser empregados vários clientes de diversas aplicações, economizando esforço repetitivo de desenvolvimento e aumentando a interoperabilidade entre componentes.

Optou-se pela solução Web Service para desenvolvimento do protótipo por permitir a comunicação entre sistemas desenvolvidos em tecnologias diferentes. Sendo assim, posteriormente poderão ser desenvolvidos aplicativos para sincronização dos dados do sistema web com aparelhos que possuam outros sistemas operacionais, não somente Android.

4 PROJETO LÓGICO

O protótipo desta monografia foi planejado para ser desenvolvido na linguagem de programação Java – o desenvolvimento para Android também é feito em Java – por ser uma linguagem que permite o desenvolvimento tanto estruturado como orientado a objetos.

A programação estruturada segue as estruturas de sequência (uma tarefa é executada após a outra, linearmente), de decisão (a partir de um teste lógico, determinado trecho de código é executado, ou não) e de iteração (a partir de um teste lógico, determinado trecho de código é repetido por um número finito de vezes). A programação orientada a objetos é formado pela implementação de um conjunto de classes que definem os objetos presentes num sistema de software, e cada classe determina o comportamento (definido pelos métodos) e estados possíveis (atributos) de seus objetos, bem como o relacionamento entre os eles.

Para este desenvolvimento, optou-se pela linguagem orientada a objetos, pois permite maior autonomia no desenvolvimento com o reaproveitamento de código, facilidade na manutenção e atende as necessidades dos sistemas, tanto para os pequenos quanto para os de grande porte.

A seguir é apresentada a especificação do protótipo desta monografia no desenvolvimento orientado a objetos.

4.1 REQUISITOS NÃO FUNCIONAIS

Para o desenvolvimento deste protótipo, foram levantados os requisitos não funcionais a seguir.

4.1.1 Sistema web

- Sistema desenvolvido em Linguagem Java.
- Banco de dados para armazenamento de dados.
- Aplicação disponível em servidor de aplicação web.
- Equipamento para acesso: Mínimo 256 MB de Memória RAM, com sistema operacional Windows (a partir da versão XP).

- Acesso ao sistema somente via internet.
- Velocidade mínima de conexão com a internet: 56 Kb/s.
- Acesso ao sistema através de navegadores web.

4.1.2 Aplicativo para dispositivos móveis com o sistema operacional Android

- Aplicativo desenvolvido para smartphone com sistema operacional Android.
- Smartphone deve conter tela sensível ao toque.
- Smartphone de conter acesso à internet.
- Para sincronização dos dados com o servidor, o aparelho deve estar conectado a internet.

4.2 REQUISITOS FUNCIONAIS

Para este protótipo, foram levantados os seguintes requisitos funcionais.

- R1 - Cadastrar Usuário
- R2 - Manter Usuário
- R3 - Manter Contato
- R4 - Visualizar Notificação
- R5 - Executar Notificação
- R6 - Visualizar Atualização
- R7 - Executar Atualização
- R8 - Sincronizar Contatos
- R9 - Autenticação de Usuário
- R10 – Validar Usuário
- R11 – Validar Contato
- R12 – Persistir de Dados

4.3 DIAGRAMA DE ARQUITETURA

O Aplicativo Android funciona independente do sistema *web*. Nesse aplicativo é possível realizar as ações de inserção, edição, listagem, exclusão e sincronização dos contatos. Os dados são salvos no próprio aparelho. Para sincronização dos dados com o sistema *web*, é solicitada a autenticação do usuário. Somente após a autenticação a sincronização é iniciada. A comunicação é feita via Web Service, por esse motivo, o aparelho deve estar conectado à *internet*.

O Sistema *web* também funciona independente do aplicativo Android. Os dados manipulados são salvos num banco de dados que pode estar localizado no mesmo servidor ou num servidor independente.

Para acessar o sistema *web*, o usuário deverá utilizar o navegador de um computador conectado à *internet*. Para utilizar o sistema, o usuário deverá realizar previamente um cadastro e validar o mesmo através de um e-mail enviado pelo sistema. Todas as ações realizadas no sistema requerem que o usuário esteja autenticado.

Através do sistema *web* não é possível sincronizar os contatos com o aplicativo Android. Essa sincronização é feita somente através do aplicativo, o qual envia as informações dos contatos para atualizar no servidor, bem como busca os dados do servidor para atualizar o aplicativo.

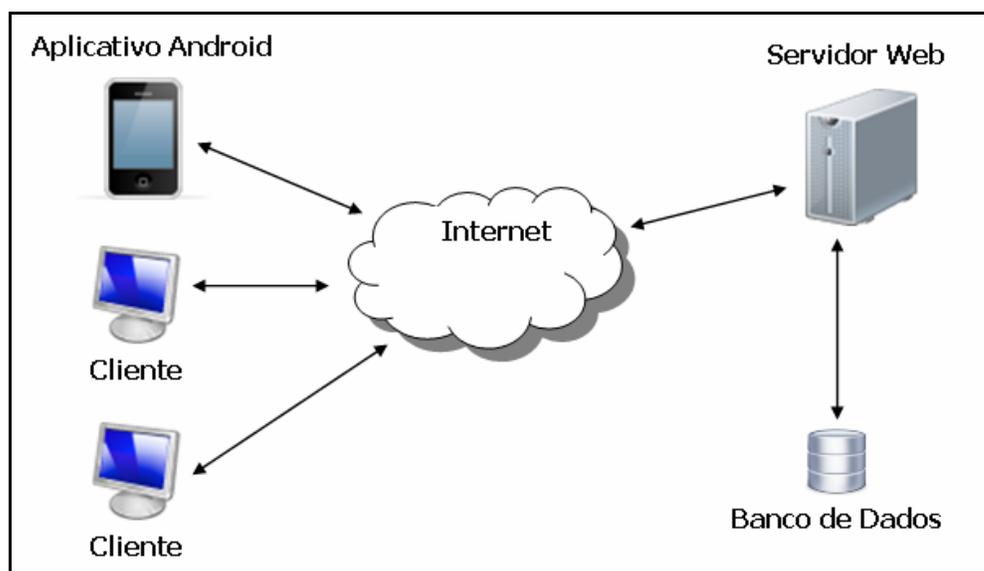


Figura 8 – Diagrama de Arquitetura do Sistema
Fonte: Autoria própria.

4.6 CASO DE USO CADASTRAR USUÁRIO

Atores: Administrador, Usuário

Nome: insertUsuario

Referências: R1, R9, R10, R12

Finalidade: Efetuar cadastro de novo usuário no sistema.

Sequência típica de eventos

Ação do ator	Resposta do sistema
1 – Este caso de uso começa quando o usuário clica no botão “Ainda não sou cadastrado” na tela de Login.	2 – Sistema exibe a tela de cadastro “Novo Usuário.”
3 – Usuário insere os dados nos campos disponíveis e clica no botão “Salvar”.	4 – Sistema realiza a validação do cadastro através do caso de uso validaUsuario.
	5 – Sistema realiza persistência de dados.
	6 – Sistema envia e-mail para ativação do cadastro e retorna mensagem de confirmação para o usuário.

4.7 CASO DE USO VALIDAR USUÁRIO

Atores: Sistema

Nome: validaUsuario

Referências: R9, R10

Finalidade: Efetuar a validação dos dados informados pelo usuário.

Sequência típica de eventos

Ação do ator	Resposta do sistema
	1 – Este caso de uso começa quando o sistema faz a validação dos dados do usuário informados pelo usuário.
	2 – Sistema valida os dados do formulário. – Caso os dados não estejam preenchidos corretamente, retorna mensagem de acordo com o erro encontrado.
	3 – Sistema retorna mensagem “SUCESSO”.

4.8 CASO DE USO VALIDAR CONTATO

Atores: Sistema

Nome: validaContato

Referências: R9, R11

Finalidade: Efetuar a validação dos dados do contato informados pelo usuário.

Sequência típica de eventos

Ação do ator	Resposta do sistema
	1 – Este caso de uso começa quando o sistema faz a validação dos dados do contato informados pelo usuário.
	2 – Sistema valida os dados do formulário. – Caso os dados não estejam preenchidos corretamente, retorna mensagem de acordo com o erro encontrado.
	3 – Sistema retorna mensagem “SUCESSO”.

4.9 CASO DE USO VISUALIZAR NOTIFICAÇÃO

Atores: Usuário, Administrador

Nome: getNotificaoByld

Referências: R4, R9

Finalidade: Visualizar as informações da notificação do sistema.

Sequência típica de eventos

Ação do ator	Resposta do sistema
1 - Este caso de uso começa quando o usuário clica em “Notificações” na tela Gerenciamento.	2 – Sistema exibe a tela Notificações com a listagem de notificações.
3 - Usuário clica em “Visualizar”.	4 – Sistema exibe a tela Dados da Notificação com as informações da notificação do sistema.

4.10 CASO DE USO EXECUTAR NOTIFICAÇÃO

Atores: Usuário, Administrador

Nome: executaNotificacao

Referências: R5, R9, R12

Finalidade: Adicionar os dados do usuário referenciado como contato do usuário autenticado e realizar a referência do contato com o usuário.

Sequência típica de eventos

Ação do ator	Resposta do sistema
1 - Este caso de uso começa quando o usuário clica em "Adicionar aos meus contatos" na tela Dados da Notificação. Caso o contato já esteja cadastrado, o botão será "Permitir visualizar minhas atualizações".	2 – Verifica se usuário referenciado já é um contato do usuário autenticado. - Se o contato ainda não estiver cadastrado, insere os dados do usuário referenciado como contato do usuário autenticado.
	3 – Sistema realiza a persistência dos dados associando o usuário autenticado com o contato do usuário referenciado (para que o sistema informe quando houver atualização do usuário autenticado) e retorna mensagem de confirmação para o usuário

4.11 CASO DE USO VISUALIZAR ATUALIZAÇÃO

Atores: Usuário, Administrador

Nome: getVisualizacaoByld

Referências: R6, R9

Finalidade: Visualizar as informações de atualização de contato.

Sequência típica de eventos

Ação do ator	Resposta do sistema
1 - Este caso de uso começa quando o usuário clica em "Atualizações" na tela Gerenciamento.	2 – Sistema exibe a tela Atualizações com a listagem de atualizações dos contatos.
3 - Usuário clica em "Visualizar".	4 – Sistema exibe a tela Dados da Atualização com as informações da atualização do sistema.

4.12 CASO DE USO EXECUTAR ATUALIZAÇÃO

Atores: Usuário, Administrador

Nome: executaNotificao

Referências: R7, R9, R12

Finalidade: Executar as informações da notificação do sistema.

Sequência típica de eventos

Ação do ator	Resposta do sistema
1 - Este caso de uso começa quando o usuário clica em "Atualizar Meu Contato" na tela Dados da Atualização.	2 – Sistema realiza a persistência dos dados atualizando os dados do contato (de acordo com a atualização do usuário de referência) e retorna mensagem de confirmação para o usuário.

4.13 CASO DE USO SINCRONIZAR CONTATOS

Atores: Usuário, Administrador

Nome: sincronizarContatos

Referências: R8, R9, R12

Finalidade: Sincronizar os contatos cadastrados do aplicativo Android com o sistema web.

Canais: Sistema Web, Aplicativo Android

Sequência típica de eventos

Ação do ator	Resposta do sistema
1 - Este caso de uso começa quando o usuário clica em "Sincronizar" na Sincronizar Contatos.	2 – Sistema envia os dados não sincronizados do aplicativo Android para o servidor web.
	3 – Servidor web realiza a persistência dos dados e retorna mensagem de confirmação.
	4 – Sistema busca os contatos não sincronizados no servidor web.
	5 – Sistema realiza persistência de dados no aplicativo Android e retorna mensagem de confirmação.

4.14 DER – DIAGRAMA DE ENTIDADE E RELACIONAMENTO

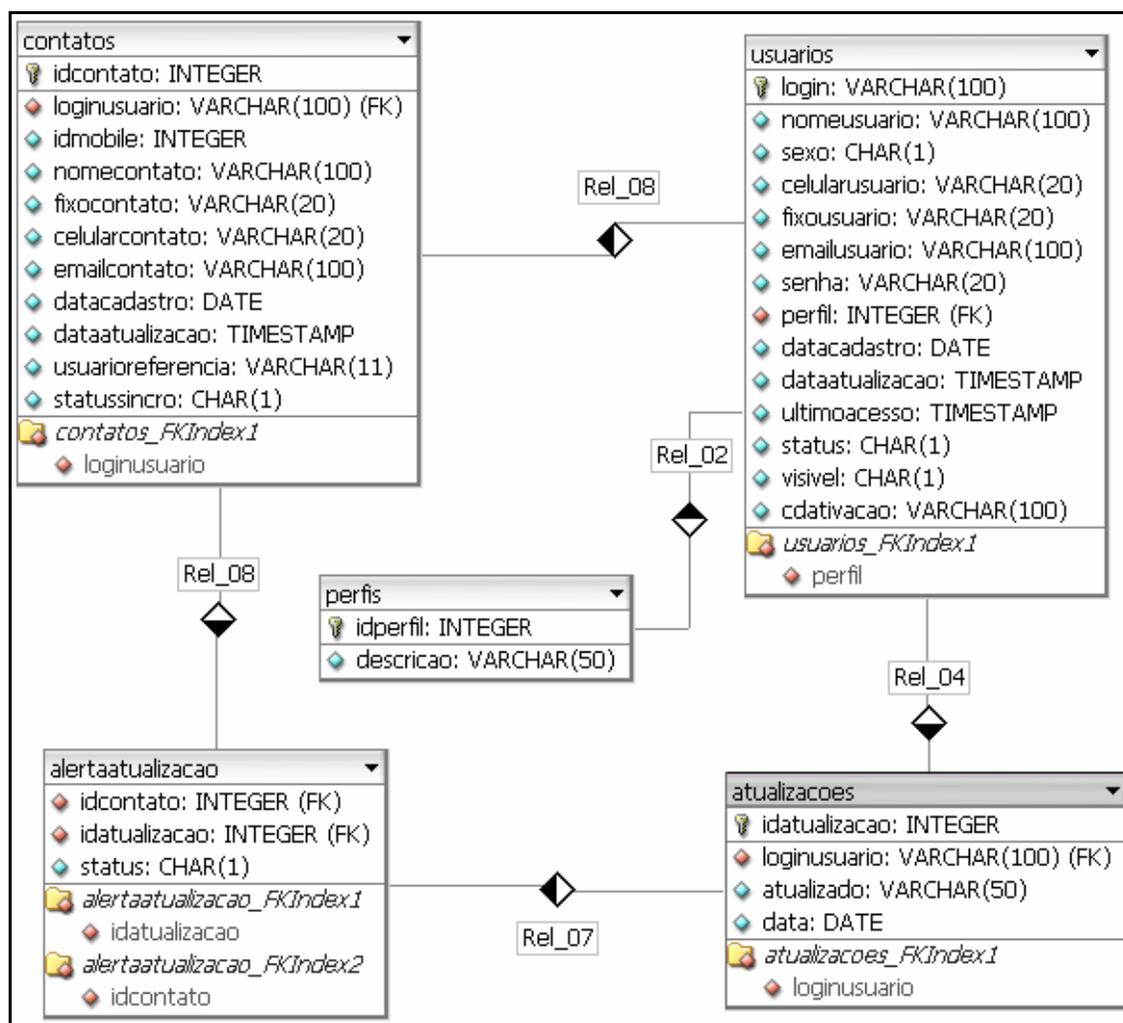


Figura 11 – DER (Diagrama de Entidade e Relacionamento).

Fonte: Autoria própria.

4.15 DIAGRAMA DE CLASSES SISTEMA WEB - GERAL

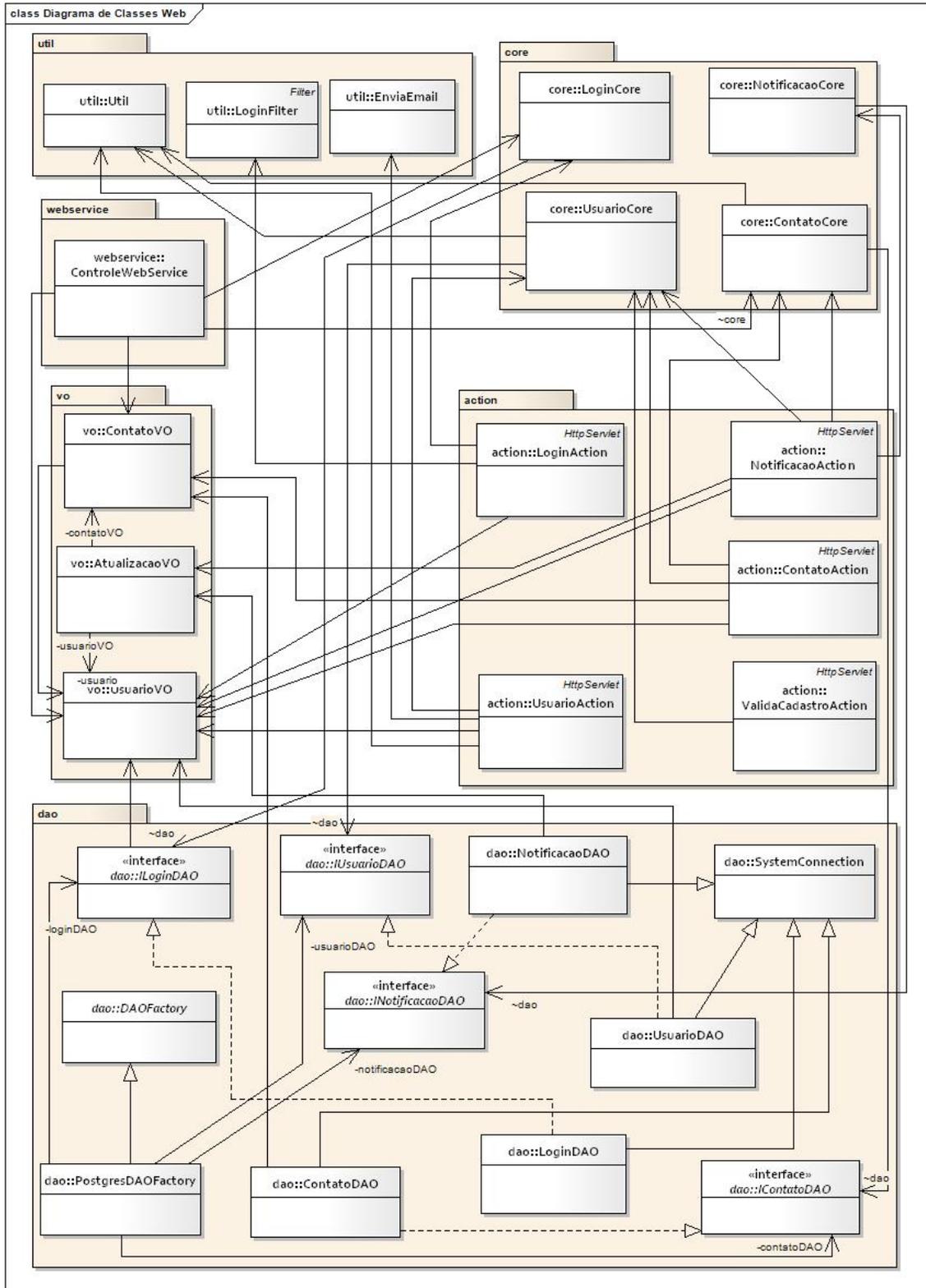


Figura 12 – Diagrama de Classes do Sistema Web - Geral

Fonte: Autoria própria.

4.16 DIAGRAMA DE CLASSES WEB – PACOTE WEBSERVICE



Figura 13 – Diagrama de Classes *Web* – Pacote webservice
Fonte: Autoria própria.

4.17 DIAGRAMA DE CLASSES WEB – PACOTE ACTION

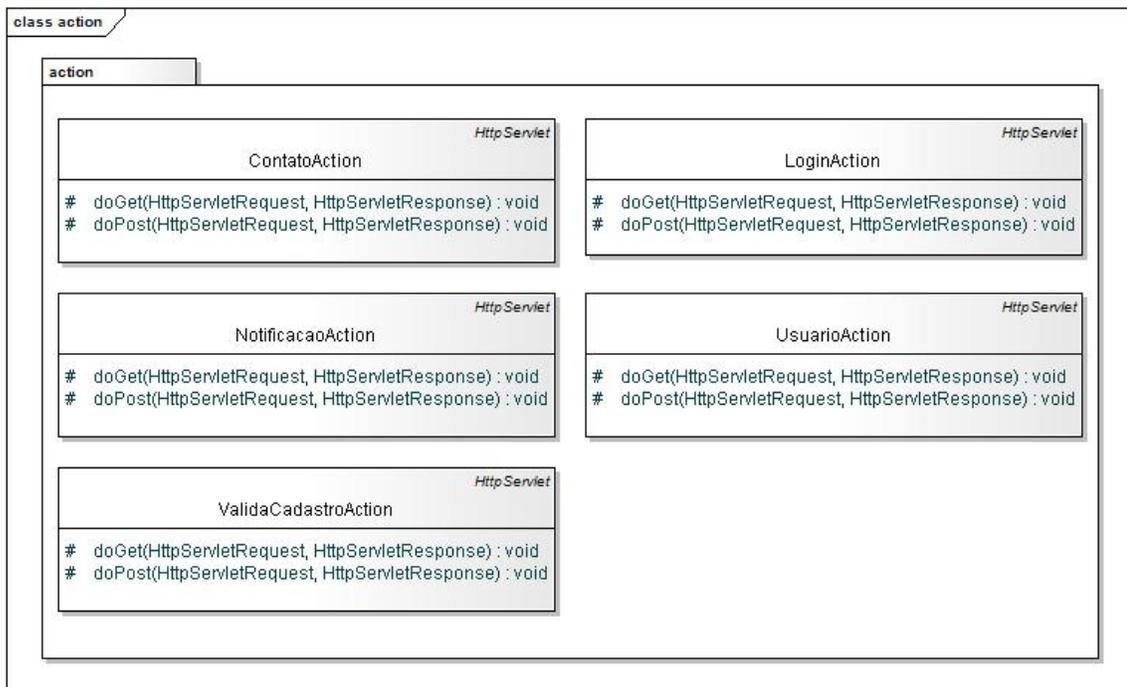


Figura 14 – Diagrama de Classes *Web* – Pacote action
Fonte: Autoria própria.

4.18 DIAGRAMA DE CLASSES WEB – PACOTE CORE

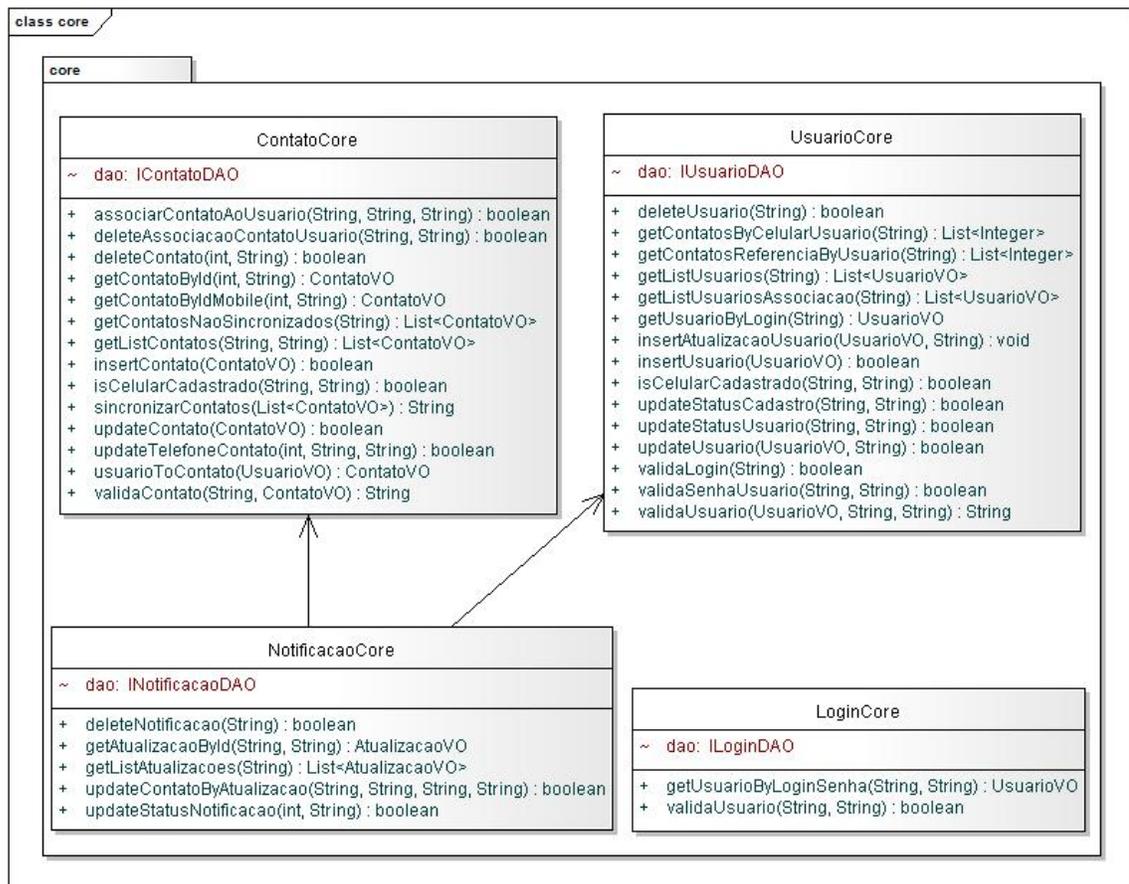


Figura 15 – Diagrama de Classes Web – Pacote core

Fonte: Autoria própria.

4.19 DIAGRAMA DE CLASSES WEB – PACOTE DAO

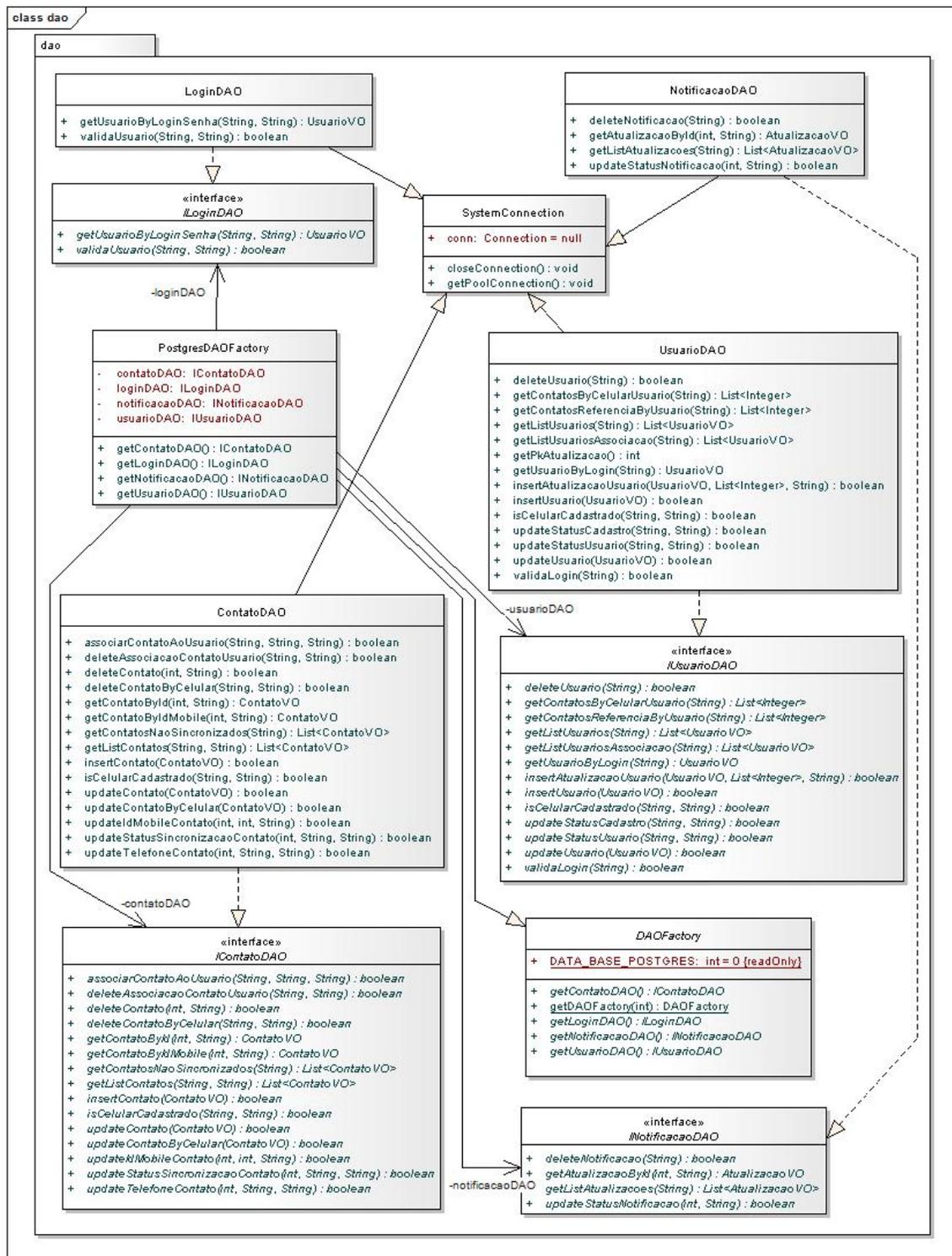


Figura 16 – Diagrama de Classes Web – Pacote dao

Fonte: Autoria própria.

4.20 DIAGRAMA DE CLASSES WEB – PACOTE UTIL

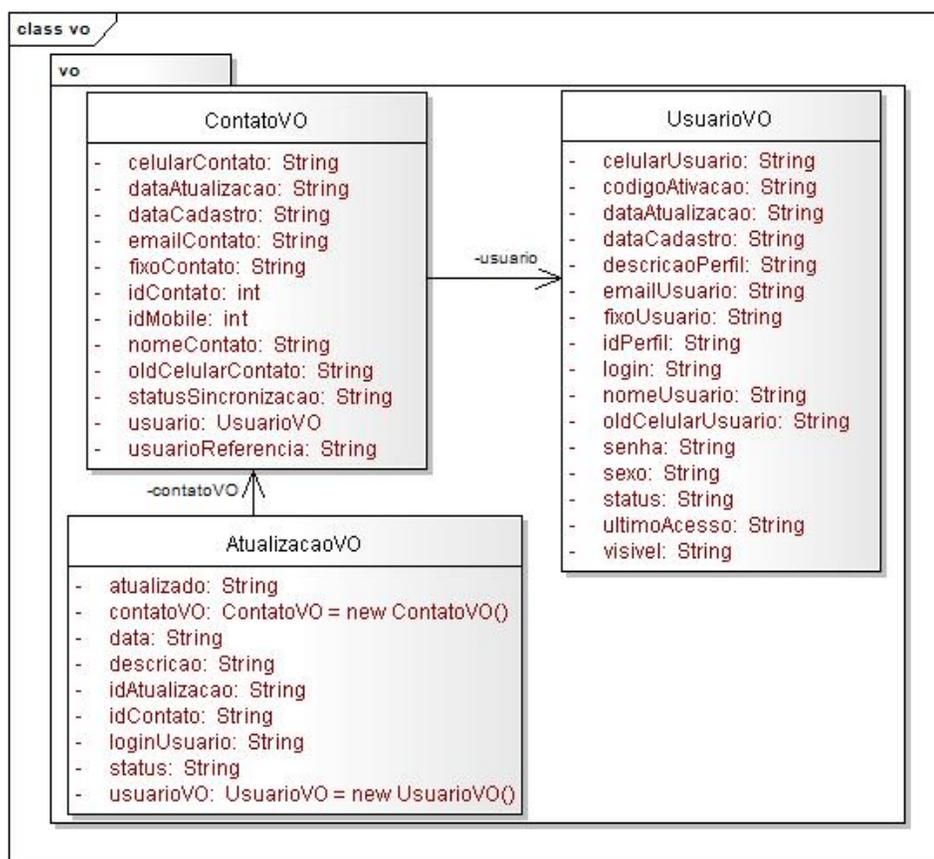


Figura 17 – Diagrama de Classes Web – Pacote vo

Fonte: Autoria própria.

4.21 DIAGRAMA DE CLASSES WEB – PACOTE UTIL

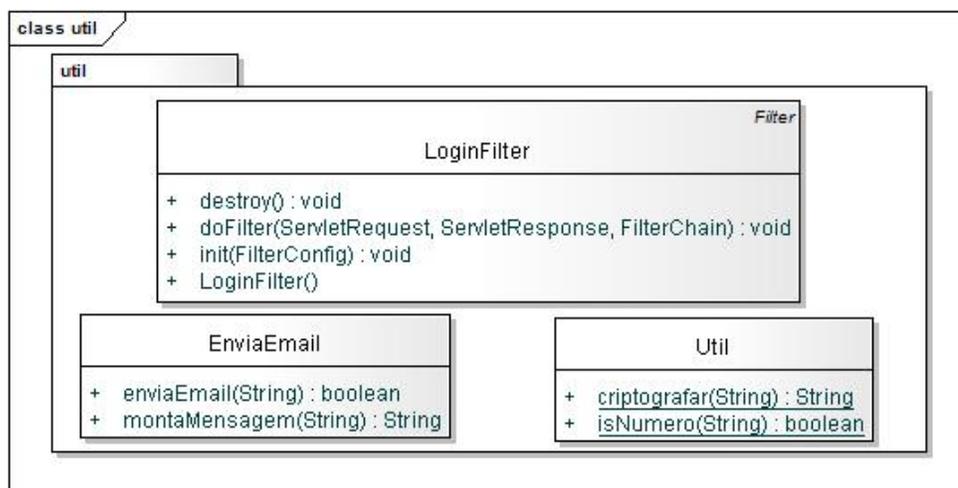


Figura 18 – Diagrama de Classes Web – Pacote util

Fonte: Autoria própria.

4.22 DIAGRAMA DE CLASSES APLICATIVO ANDROID - GERAL

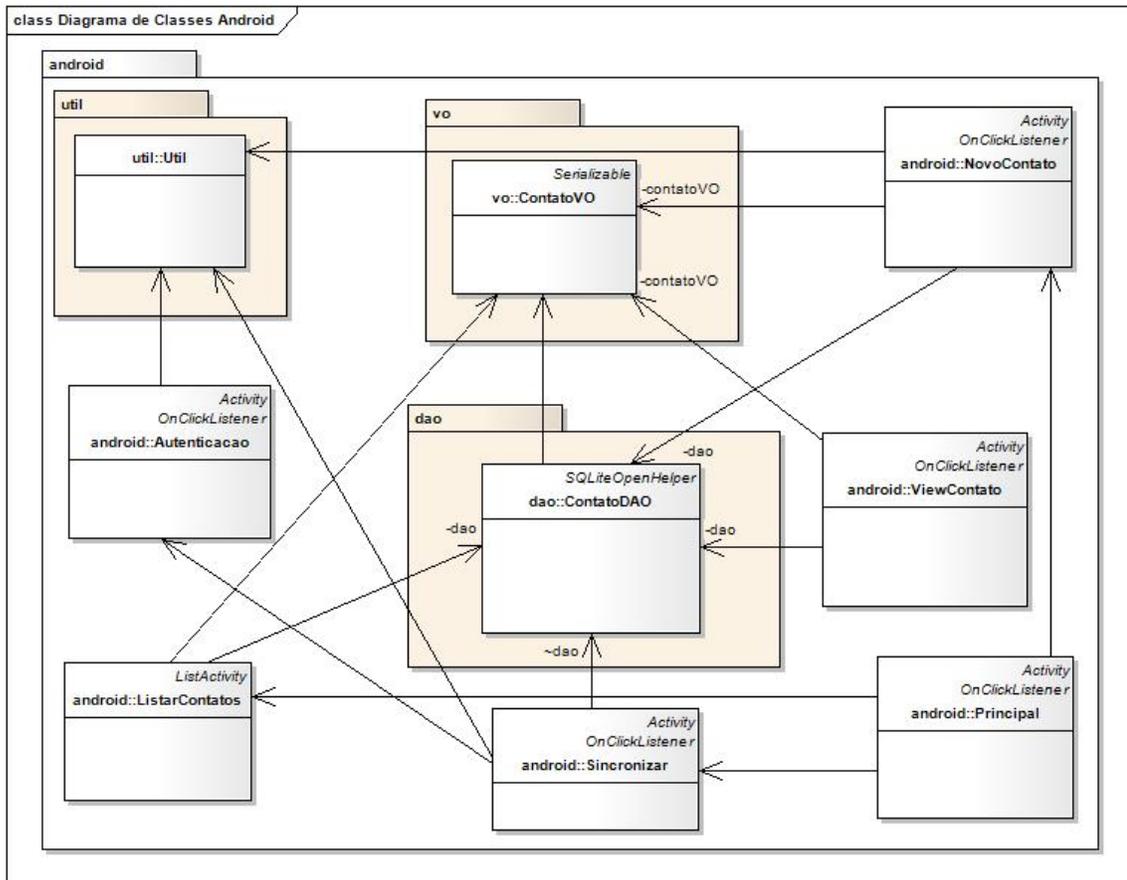


Figura 19 – Diagrama de Classes do Aplicativo Android - Geral

Fonte: Autoria própria.

4.23 DIAGRAMA DE CLASSES ANDROID – PACOTE UTIL

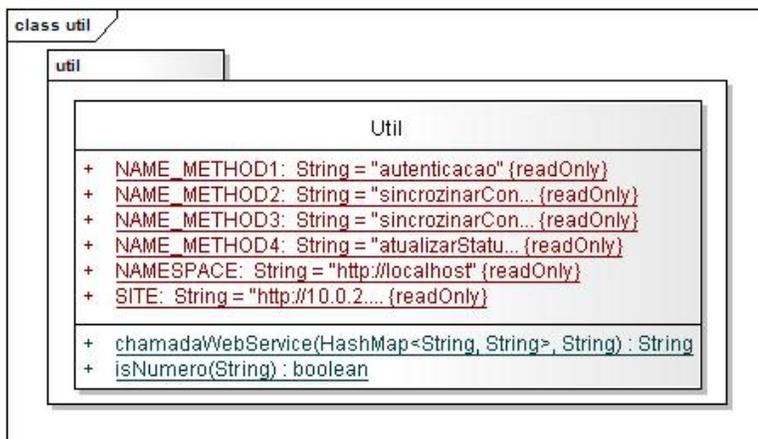


Figura 20 – Diagrama de Classes Android – Pacote util

Fonte: Autoria própria.

4.24 DIAGRAMA DE CLASSES ANDROID – PACOTE ANDROID

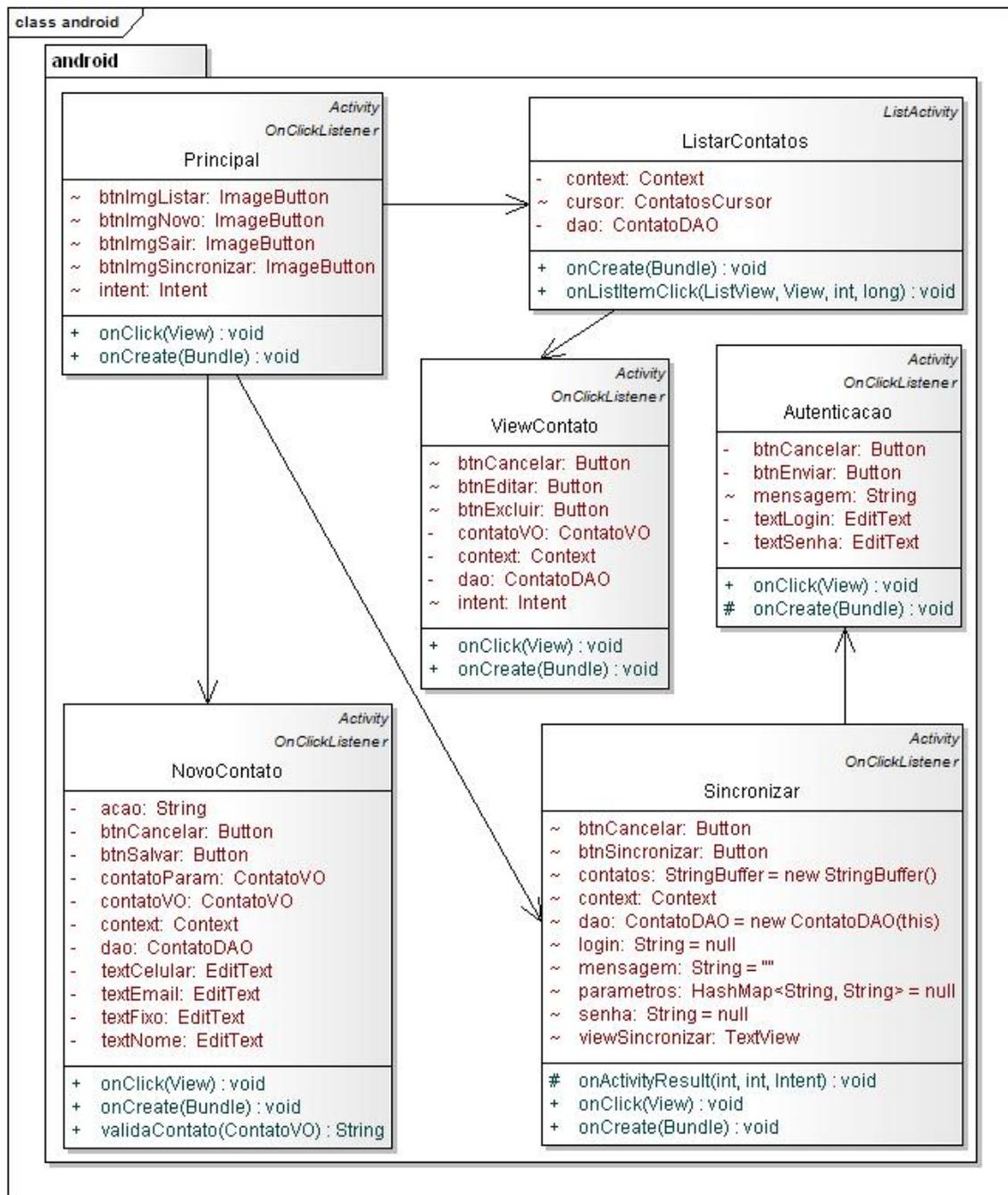


Figura 21 – Diagrama de Classes Android – Pacote android

Fonte: Autoria própria.

4.25 DIAGRAMA DE CLASSES ANDROID – PACOTE DAO

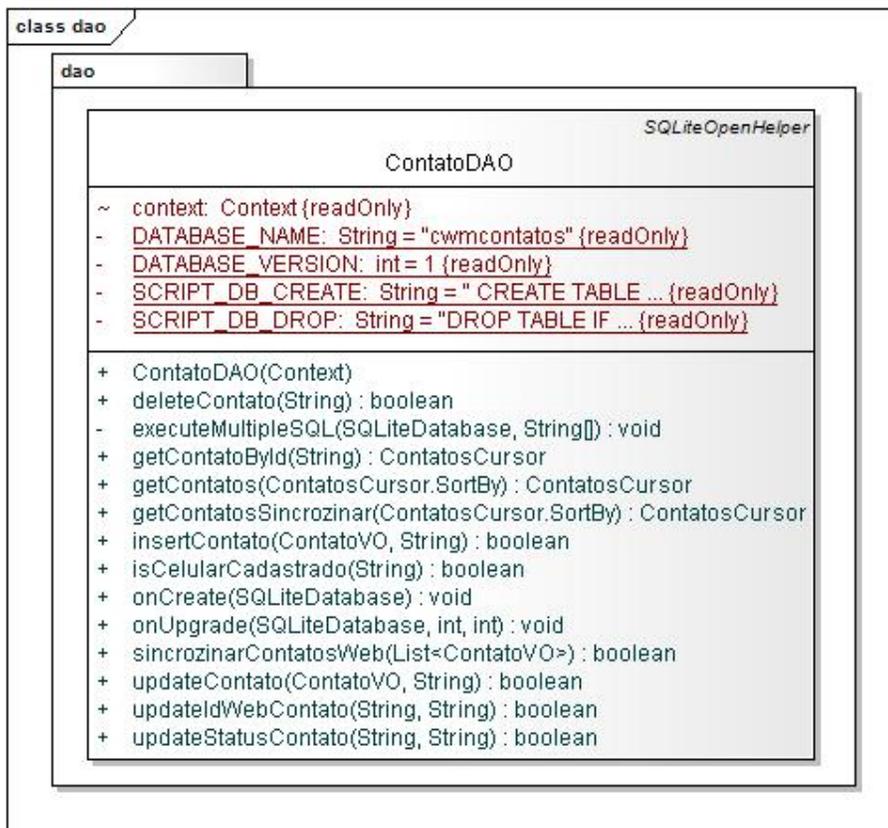


Figura 22 – Diagrama de Classes Android – Pacote dao
Fonte: Autoria própria.

4.26 DIAGRAMA DE CLASSES ANDROID – PACOTE VO

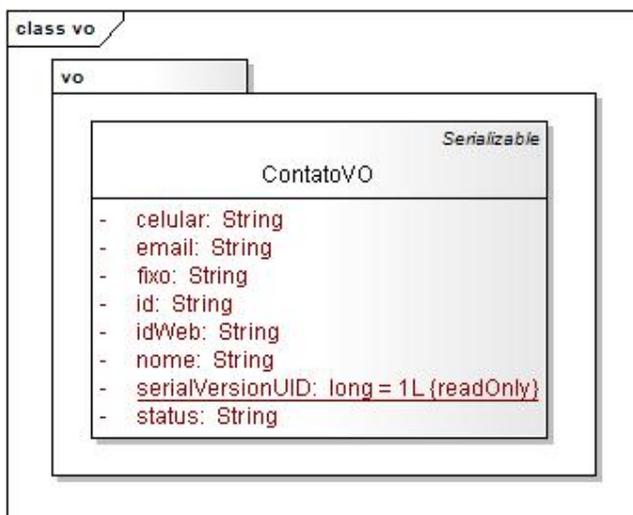


Figura 23 – Diagrama de Classes Android – Pacote util
Fonte: Autoria própria.

4.27 DIAGRAMA DE SEQUÊNCIA CADASTRAR USUÁRIO

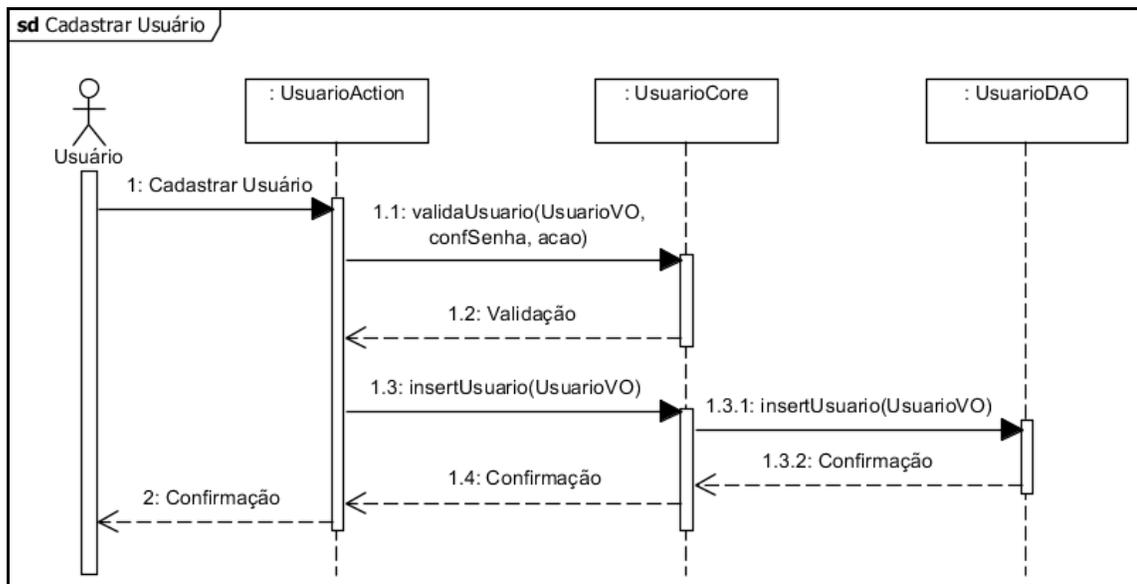


Figura 24 – Diagrama de Sequência Cadastrar Usuário
Fonte: Autoria própria.

4.28 DIAGRAMA DE SEQUÊNCIA VALIDAR USUÁRIO

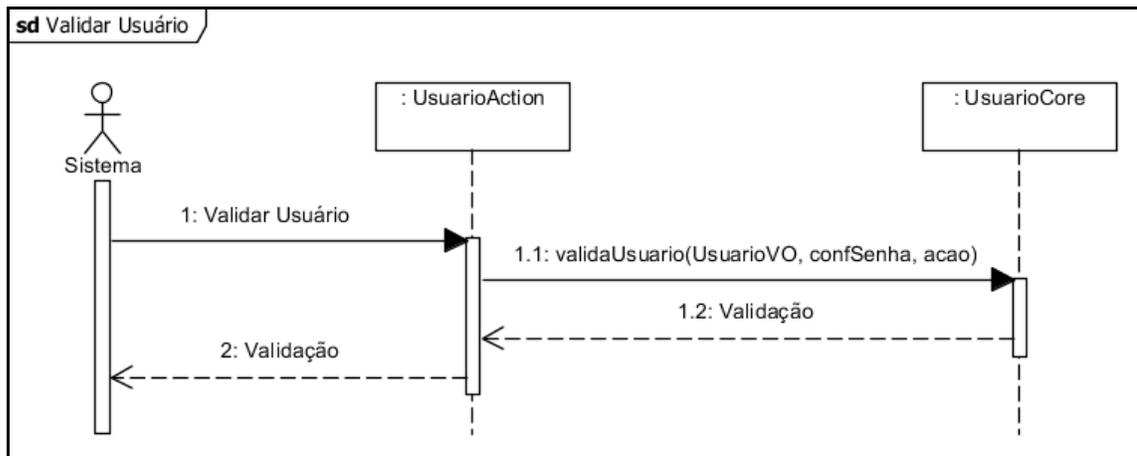


Figura 25 – Diagrama de Sequência Validar Usuário
Fonte: Autoria própria.

4.29 DIAGRAMA DE SEQUÊNCIA VISUALIZAR NOTIFICAÇÃO

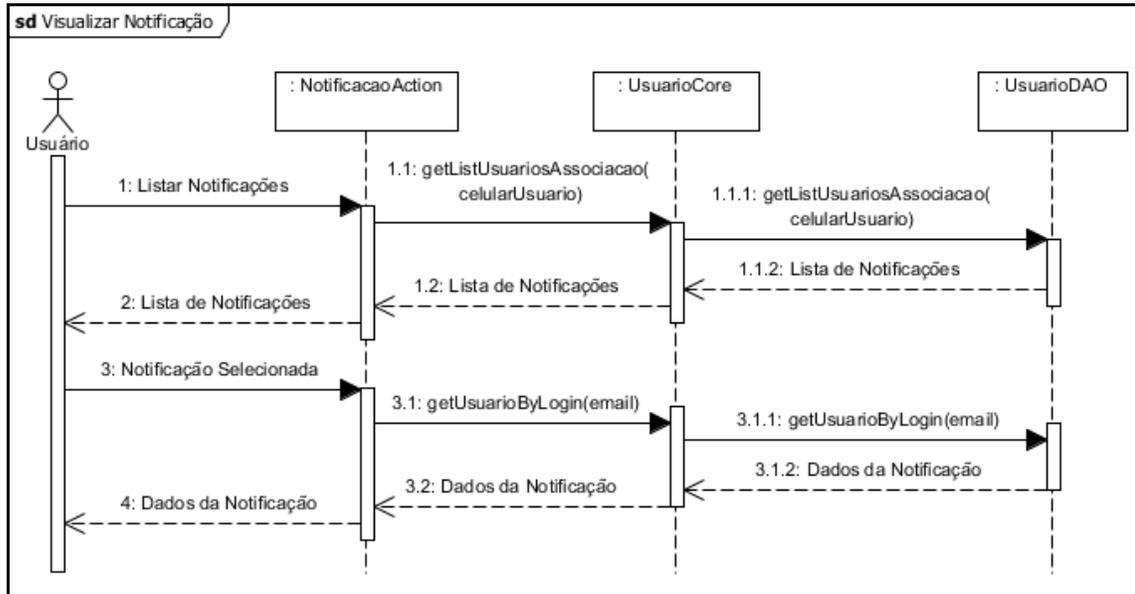


Figura 26 – Diagrama de Sequência Visualizar Notificação

Fonte: Autoria própria.

4.30 DIAGRAMA DE SEQUÊNCIA EXECUTAR NOTIFICAÇÃO

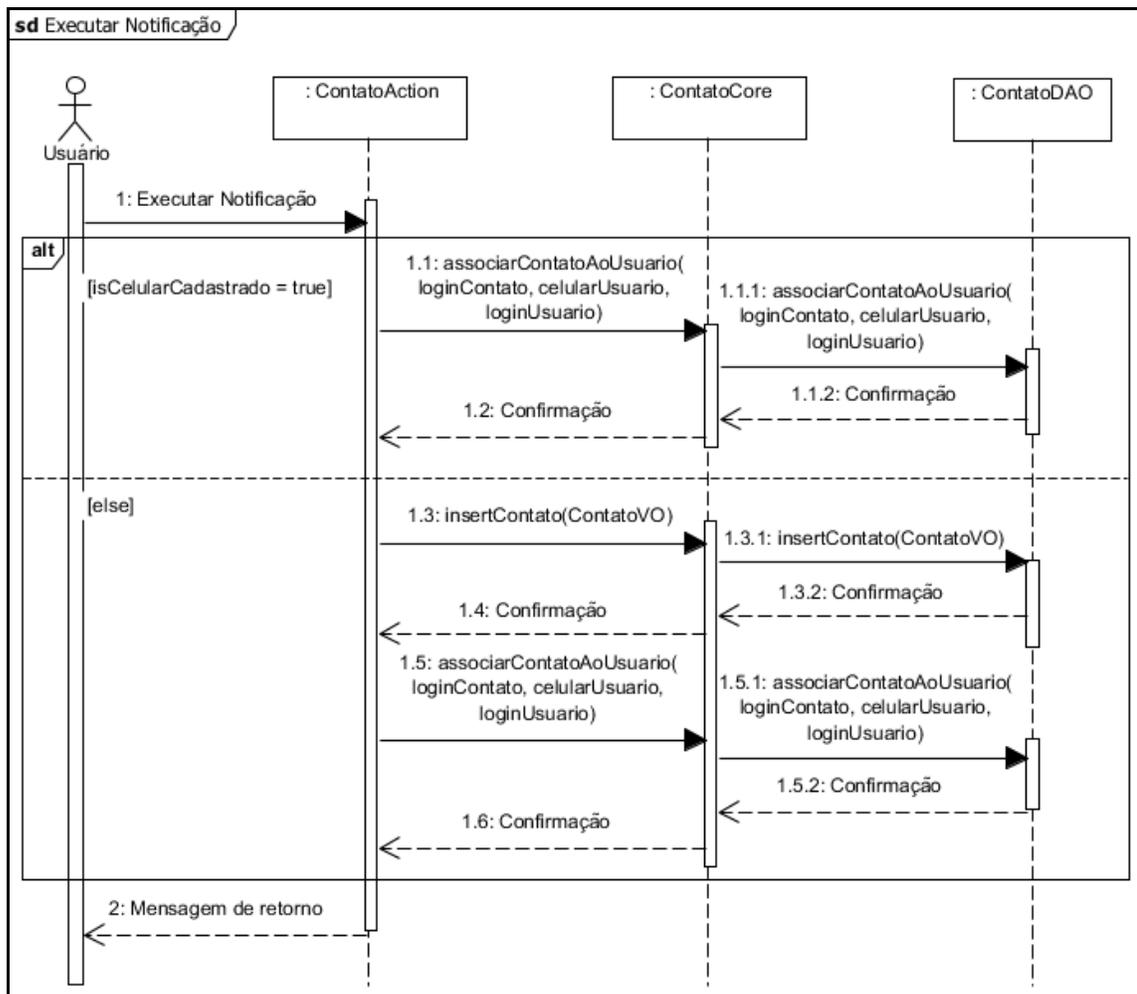


Figura 27 – Diagrama de Sequência Executar Notificação
 Fonte: Autoria própria.

4.31 DIAGRAMA DE SEQUÊNCIA VISUALIZAR ATUALIZAÇÃO

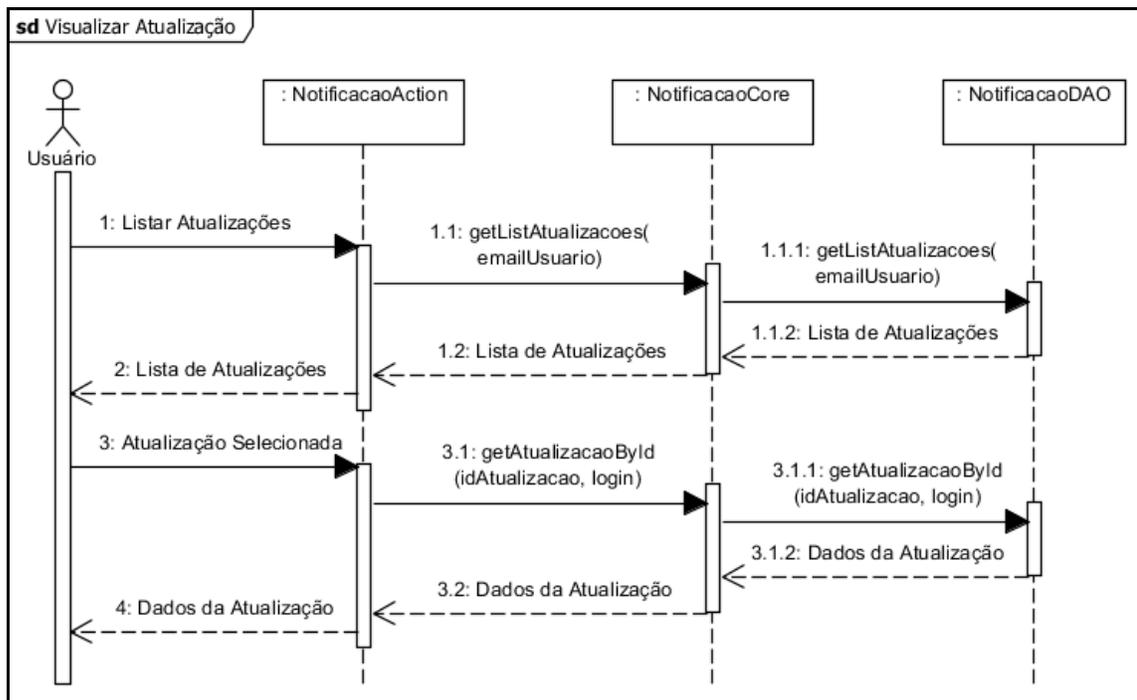


Figura 28 – Diagrama de Sequência Visualizar Atualização
Fonte: Autoria própria.

4.32 DIAGRAMA DE SEQUÊNCIA EXECUTAR ATUALIZAÇÃO

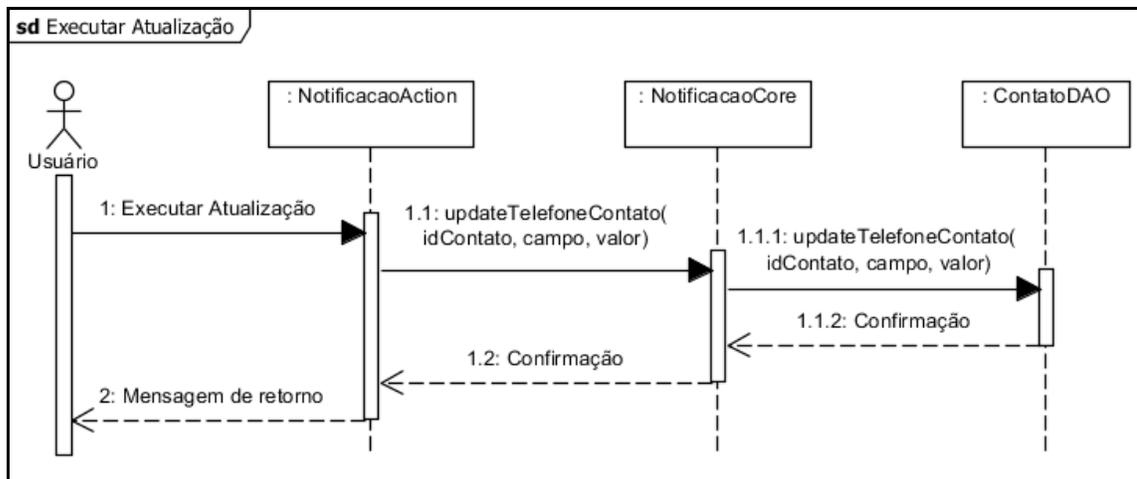


Figura 29 – Diagrama de Sequência Executar Atualização
Fonte: Autoria própria.

4.33 DIAGRAMA DE SEQUÊNCIA SINCRONIZAR CONTATOS ANDROID

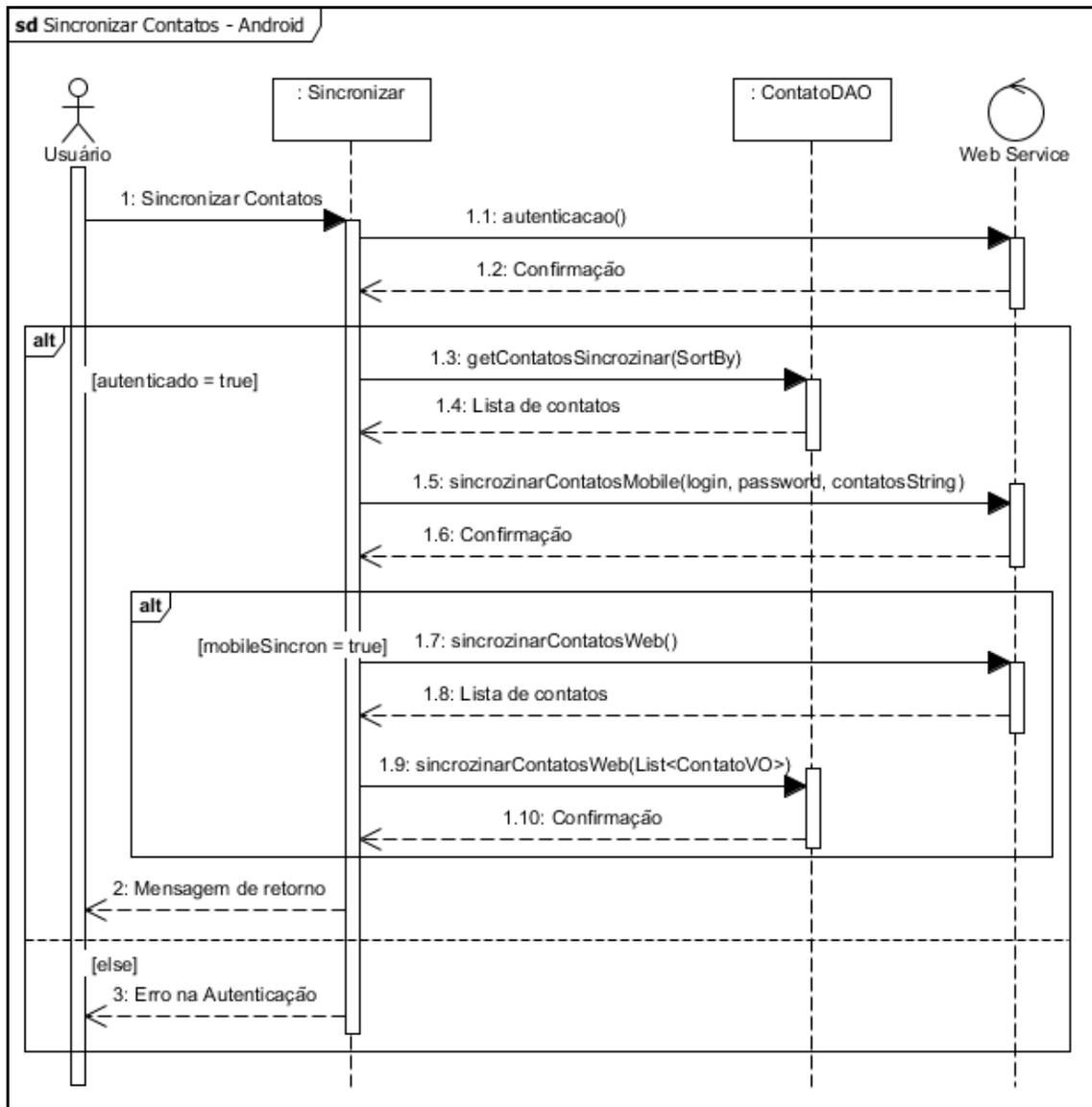


Figura 30 – Diagrama de Sequência Sincronizar Contatos Android

Fonte: Autoria própria.

4.34 DIAGRAMA DE SEQUÊNCIA SINCRONIZAR CONTATOS WEB

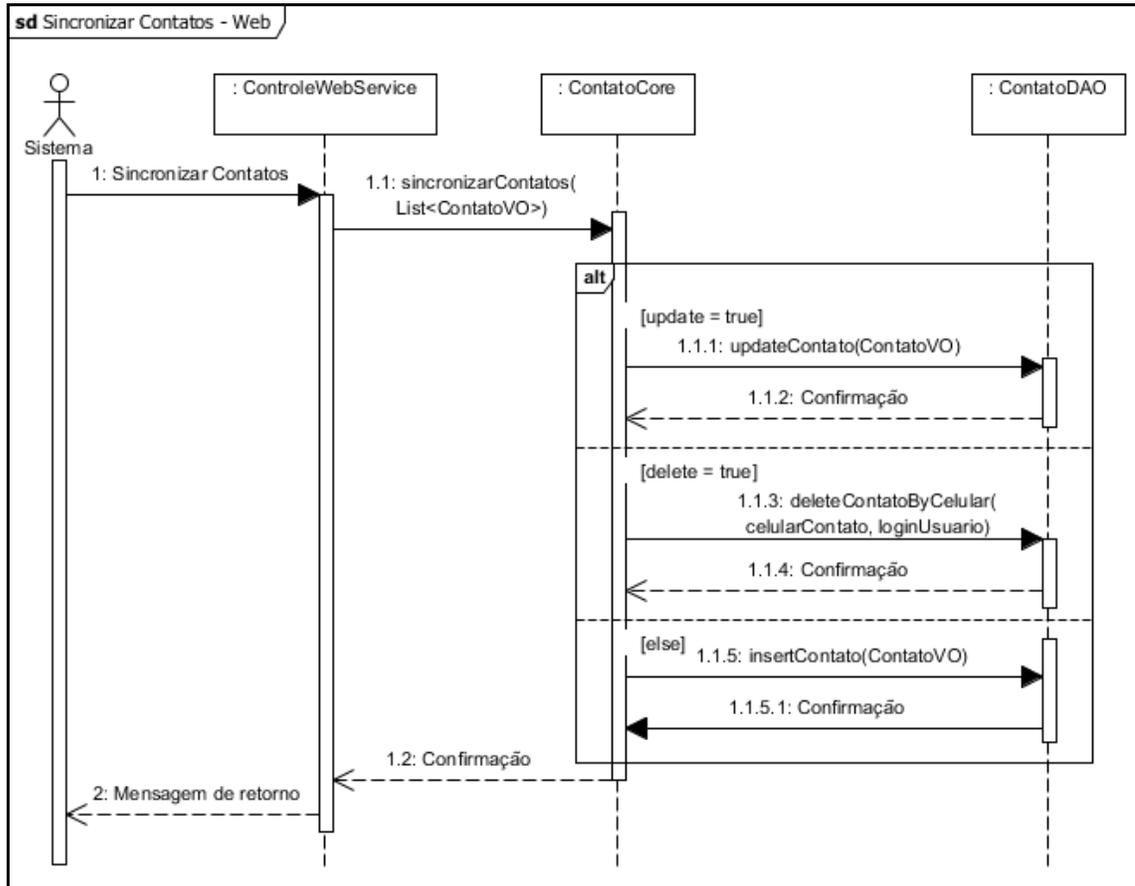


Figura 31 – Diagrama de Sequência Sincronizar Contatos Web

Fonte: Autoria própria.

4.35 RESULTADOS

Neste capítulo são apresentadas as telas do sistema.

4.35.1 Tela Novo Usuário

Através desta tela o usuário pode se cadastrar no site para que possa utilizá-lo. Os campos com asterisco são de preenchimento obrigatório. Após o cadastro o sistema envia um e-mail para que o cadastro possa ser validado. Somente após a validação é possível acessar o sistema.

A imagem mostra a interface de usuário para o cadastro de um novo usuário. No topo, há o logotipo "CWM Contatos" e o slogan ":: Mantenha seus contatos sempre atualizados ::". Abaixo, o título "Novo Usuário" está destacado. O formulário contém os seguintes campos e opções:

- Insira seus dados no formulário abaixo.
- Aguarde o recebimento de um e-mail para validação do seu cadastro.
- E-mail *:** Campo de texto com o placeholder "Informe um e-mail válido."
- Nome *:** Campo de texto.
- Sexo *:** Opções de rádio para "Masculino" e "Feminino".
- Celular *:** Campo de texto com o exemplo "Ex.: 9999999999".
- Fixo:** Campo de texto com o exemplo "Ex.: 9999999999".
- Senha *:** Campo de texto.
- Redigite a senha *:** Campo de texto para confirmação da senha.

Na base do formulário, há dois botões: "Salvar" e "Voltar".

Figura 32 – Tela do sistema para cadastro de novo usuário

Fonte: Autoria própria.

4.35.2 Tela Cadastrar Contato

Através desta tela o usuário pode cadastrar seus contatos no sistema web. Pode também editar os dados do mesmo após o cadastro. Tanto a inserção quanto a edição serão refletidos no celular após a sincronização. Se o contato estiver relacionado a outro usuário do sistema, também é possível visualizar o perfil desse usuário.

CWM Contatos :: Mantenha seus contatos sempre atualizados ::

[Início](#)
[Novo Contato](#)
[Meus Contatos](#)
[Meus Dados](#)
[Gerenciar](#)

Bem vindo(a) **Usuário 3** | [Sair](#)

Dados do Contato

Nome *:

Celular *: Ex.: 099999999999

Fixo: Ex.: 099999999999

E-mail:

Cadastrado em: 27/05/2011
 Atualizado em: 27/05/2011 00:28:47

Usuário Associado

Figura 33 – Tela do sistema para cadastro de novo contato
 Fonte: Autoria própria.

4.35.3 Tela Notificações

Através desta tela o usuário pode visualizar a lista de notificações do sistema. Essas notificações informam quais usuários cadastrados no sistema cadastraram o celular do usuário autenticado como contato.

CWM Contatos :: Mantenha seus contatos sempre atualizados ::

[Início](#)
[Novo Contato](#)
[Meus Contatos](#)
[Meus Dados](#)
[Gerenciar](#)

Bem vindo(a) **Usuário 3** | [Sair](#)

Notificações

Contatos que possuem meu número de celular cadastrado

	O usuário " Usuário 1 " tem você como contato.	Visualizar
	O usuário " Usuário 2 " tem você como contato.	Visualizar
	O usuário " Usuário 4 " tem você como contato.	Visualizar

Figura 34 – Tela do sistema para visualizar a lista de notificações
 Fonte: Autoria própria.

4.35.4 Tela Dados da Notificação

Através desta tela o usuário pode visualizar os dados do usuário que o adicionou como contato. Se ainda não o tem como contato, o usuário tem a opção adicioná-lo como um novo contato. Após inserir os dados do contato, o sistema associa o novo contato adicionado ao usuário de referência, para que possa receber alerta quando o usuário de referência alterar seu número de telefone. Se o usuário autenticado já possuir o celular do usuário de referência cadastrado como contato, o sistema identifica e somente faz a associação para alerta de atualização.



CWM Contatos :: Mantenha seus contatos sempre atualizados ::

🏠 Início 🧑 Novo Contato 👤 Meus Contatos 📄 Meus Dados ⚙️ Gerenciar

Bem vindo(a) **Usuário 3** | [Sair](#)

Dados da Notificação

Descrição: Este usuário tem seu número de celular cadastrado.

Nome: **Usuário 4**

Celular: **4198765432**

Fixo: 4133335050

E-mail: email@email.com

Cadastrado em: 04/05/2011

Obs.: Ao clicar no botão "Adicionar aos meus contatos", você irá adicionar este usuário aos seus contatos e ao mesmo tempo, permitir que receba suas atualizações.

Figura 35 – Tela do sistema para visualizar os dados da notificação

Fonte: Autoria própria.

4.35.5 Tela Atualizações

Através desta tela o usuário pode visualizar a lista de atualizações dos seus contatos que possuem associação com usuários cadastrados no sistema.

Atualizações

Atualizações dos meus contatos

Status	Atualização	Data	Ação
✉	Meu Contato 1 (Usuário 1) atualizou Celular.	27/05/2011 00:40	Visualizar
✉	Meu Contato 2 (Usuário 2) atualizou Celular e Fixo.	27/05/2011 00:40	Visualizar
✉	Meu Contato 2 (Usuário 2) atualizou Celular.	27/05/2011 00:42	Visualizar
✉	Meu Contato 4 (Usuário 4) atualizou Celular e Fixo.	27/05/2011 00:43	Visualizar

[Voltar](#)

Figura 36 - Tela do sistema para visualizar a lista de atualizações
Fonte: Autoria própria.

4.35.6 Tela Dados da Atualização

Através desta tela o usuário pode visualizar os dados da atualização de um dos seus contatos. O sistema exibe informações sobre seu contato e o usuário de referência, a data da atualização, quais dos telefones foram alterados e qual o novo número. Permite que usuário possa atualizar os dados do seu contato para o(s) número(s) novo(s). Tem também a opção de excluir a atualização para que não seja mais exibida na lista de atualizações.

Dados da Atualização

Meu Contato: **Meu Contato 2**

Referência: Usuário 2

Atualizado em: 27/05/2011 00:40

Descrição:

- Celular atualizado para 4112345670
- Fixo atualizado para 1111111112

[Atualizar Meu Contato](#) [Excluir Atualização](#) [Voltar](#)

Figura 37- Tela do sistema para visualizar os dados da atualização
Fonte: Autoria própria.

4.35.7 Tela inicial do aplicativo Android

Através da tela inicial o usuário pode acessar as telas para cadastrar, visualizar, editar, excluir e listar os contatos cadastrados no aplicativo Android. Essas telas não serão exibidas por serem de funcionamento comum como em outros aplicativos de cadastro. A principal opção é a de Sincronização, que fará a integração dos contatos cadastrados no aparelho com o sistema *web*. Essa funcionalidade é demonstrada na tela “Sincronizar Contatos”.



Figura 38 - Tela inicial do aplicativo Android com o *menu* de opções.
Fonte: Autoria própria.

4.35.8 Tela Sincronizar Contatos

Através desta tela o usuário pode sincronizar os dados dos contatos do aplicativo no *smartphone* Android com os dados do servidor *web*. O sistema exibe a lista dos contatos que serão sincronizados. Após clicar no botão Sincronizar, o aplicativo solicita autenticação no sistema *web* através de *login* e senha. Após a autenticação, o usuário deve clicar novamente no botão Sincronizar. O aplicativo envia os dados para sincronizar com os dados salvos no sistema *web*. Se a sincronização ocorrer com sucesso, o aplicativo busca no sistema *web* os dados não sincronizados para atualizar com os contatos do aplicativo Android.



Figura 39 - Tela do aplicativo para executar a sincronização dos contatos entre o aparelho Android e o sistema *web*.

Fonte: Autoria própria.

5 CONCLUSÃO

Apesar de existirem diversos aplicativos e soluções que auxiliem os usuários de telefones celulares na manutenção e recuperação da lista de contatos em caso de perda ou troca do aparelho, os mesmos não oferecem recursos que permita a sincronização dos dados com todos os modelos de aparelhos existentes, nem a atualização automática de informações.

O protótipo desenvolvido na presente monografia se propôs a realizar a sincronização dos dados cadastrados no sistema web com os cadastrados no aplicativo *web* para *smartphone* Android, para que ambas possam estar sempre atualizadas. Para essa atividade foi utilizada a solução Web Service, para que posteriormente possa ser realizada a sincronização em aparelhos com outros Sistemas Operacionais. Há ainda a facilidade de sincronização sem a necessidade de cabo USB (*Universal Serial Bus*) ou mesmo de instalação de software para sincronização, necessitando somente do aplicativo instalado no *smartphone* e acesso a internet.

Este trabalho também se propôs a realização de uma interação entre os usuários cadastrados no sistema web, relacionando os contatos com os usuários cadastrados através do número de celular, para que quando um usuário alterar seu número de celular ou fixo, os outros usuários que o possuem como contato possam ser alertados dessa alteração e possam atualizar nos seus cadastros, mantendo assim os dados dos seus contatos sempre atualizados.

Sendo assim, pode-se concluir que o objetivo inicial deste trabalho foi alcançado com sucesso, tanto para o site, que permite o cadastro de contatos em uma base de dados acessível de qualquer computador conectado a *internet*, quanto para aplicativo acessado via celular, que oferece ao usuário a opção de manutenção de seus contatos e a sincronização com a aplicação web, mantendo dessa forma, uma agenda de contatos sempre atualizada.

Algumas funcionalidades ainda podem ser incorporadas a este protótipo em trabalhos futuros, tais como: validação do cadastro através de mensagem de texto, desenvolvimento do aplicativo para aparelhos de celular ou *smartphone* com outros sistemas operacionais, alertar sobre a atualização de e-mail, permitir cadastro de mais de um número de celular ou fixo.

REFERÊNCIAS BIBLIOGRÁFICAS

ABREU, Leonardo Marques de. **Usabilidade de Telefones Celulares com Base em Critérios Ergonômicos**, 2005. Disponível em:

<http://www.maxwell.lambda.ele.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=6705@1>. Acessado em: 24 maio 2011, 00:40.

ALBINADER NETO, Jorge; LINS, Rafael Dueire. **Web Services em Java**. Rio de Janeiro: Brasport, 2006.

ALMEIDA, Eduardo Cunha de. **Estudo de Viabilidade de uma Plataforma de Baixo Custo Para Data Warehouse**, 2004. Disponível em:

<http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/662/Eduardo_Cunha_de_Almeida.pdf?sequence=1>. Acessado em: 06 junho 2011, 16:00.

BAIO, Cintia. **Backup de agenda: conheça alguns serviços para recuperar contatos telefônicos**, 2008. Disponível em:

<<http://tecnologia.uol.com.br/dicas/ultnot/2008/08/14/ult2665u370.jhtm>>. Acessado em: 24 abril 2011, 20:00.

COSTA, Anderson. **Claro Contatos: salve a agenda do seu Claro Chip na internet**, 2011. Disponível em:

<http://www.claroblog.com.br/conteudo.asp?post_id=640>. Acessado em: 28 abril 2011, 20:30.

COSTA, Eric, **Google Sync**, 2010. Disponível em:

<<http://info.abril.com.br/downloads/webware/google-sync>>. Acessado em: 01 junho 2011, 13:40.

GUEDES, Luís Fernando Ascensão; VASCONCELLOS, Liliana. Uma Proposta de Modelo Preditivo para o Crescimento da Telefonía Celular no Brasil. **Future Studies Research Journal**. São Paulo, v. 1, n. 1, jan. 2009. Disponível em:

<<http://revistafuture.org/index.php/FSRJ/article/viewFile/2/2>>. Acessado em: 30 maio 2011, 21:40.

MICROSOFT. **My Phone**, 2009. Disponível em:

<<http://myphone.microsoft.com/mkweb/Install.po?mkt=pt-BR>>. Acessado em: 28 abril 2011, 23:55.

MIQ. **About**, 2011. Disponível em: <<http://miqlive.com/about>>. Acessado em: 02 fev. 2011, 00:15.

NICOLACI-DA-COSTA, Ana Maria. Celulares: A Emergência de um Novo Tipo de Controle Materno. **Scientific Electronic Library Online**. v. 1, n. 1, julho 2006.

Disponível em: <<http://www.scielo.br/pdf/%0D/psoc/v18n3/a13v18n3.pdf>>. Acessado em: 23 maio 2011, 23:50.

OI. **Oi Agenda - Você não vai mais querer viver sem**, 2011. Disponível em: <<http://loja.oi.com.br/oiinternet/staticContent.do?path=/html/oi-agenda/index.html>>. Acessado em: 28 abril 2011, 20:20.

POLO, Juan Diego. **Contails – Sincronizando e guardando seus contatos do celular pela Internet**, 2009. Disponível em: <<http://br.wwwwhatsnew.com/2009/02/contails-sincronizando-e-guardando-seus-contatos-do-celular-pela-internet/>>. Acessado em: 24 abril 2011, 21:15.

RESENDE, Ricardo. **Percepção e Uso do Telefone Celular Entre Consumidores de Mais Idade**, 2010. Disponível em: <http://www.maxwell.lambda.ele.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=15595@1>. Acessado em: 24 maio 2011, 01:30.

ROGERS, Rick et al. **Android – Desenvolvimento de Aplicações Android**. 1. ed. Tradução: Lia Gabriele Regius. São Paulo: Novatec, 2009.

SCHERER, Adriana Paula Zamin. **GURU - Uma ferramenta para administrar banco de dados através da Web**, 2002. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/3888/000345169.pdf?sequence=1>>. Acessado em: 03 junho 2011, 13:10.

SERSON, Roberto Rubinstein. **Programação Orientada a Objetos com Java 6**. Rio de Janeiro: Brasport, 2007.

TIM. **TIM Agenda**, 2011. Disponível em: <http://www.tim.com.br/portal/site/PortalWeb/menuitem.d884721d78543193b5508210703016a0/?vgnnextoid=ef87152d7c060110VgnVCM100000a22e700aRCRD&wfe_pweb_area=24&wfe_pweb_estado=13&&wfe_pweb_oid=eaaa2183efe63110VgnVCM100000a22e700a____&>. Acessado em: 28 abril 2011, 21:10.

VERZA, Fabiana. **O Uso do Celular na Adolescência e Sua Relação com a Família e Grupo de Amigos**, 2008. Dissertação (Mestrado em Psicologia), Pontifícia Universidade Católica do Rio Grande do Sul, 2008. Disponível em: <http://tede.pucrs.br/tde_arquivos/20/TDE-2009-04-17T063754Z-1833/Publico/411130.pdf>. Acessado em: 30 maio 2011, 21:00.

VIVO. **Vivo Agenda**, 2011. Disponível em: <<http://www.vivo.com.br/portal/para-voce-servicos-vivo-agenda.php?WT.ac=paravoce.servicos.menu lateral.vivoagenda>>. Acessado em: 28 abril 2011, 20:45.