Universidade Federal do Rio de Janeiro

Escola Politécnica

Departamento de Eletrônica e de Computação

Sistema de Sincronizaçãode Dados entre Computadores e Celulares

Autor:

Camilla Pinheiro Gueiros

Orientador:

Prof. Marcelo Luiz Drumond Lanza.

Examinador:

Prof. Aloysio Aloysio de Castro Pinto Pedroza

Examinador:

Prof . Maurus Campello Queiroz

DEL

Março de 2011

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Escola Politécnica – Departamento de Eletrônica e de Computação
Centro de Tecnologia, bloco H, sala H-217
CidadeUniversitáriaRio de Janeiro – RJ
CEP 21949-900

AGRADECIMENTOS

Dedico este trabalho ao povo brasileiro que contribuiu de forma significativa à minha formação e estada nesta Universidade. Agradeço a todos os educadores que contribuíram com a minha formação. Este projeto é uma pequena forma de retribuir o investimento e a confiança em mim depositados. Agradeço à minha família que sempre me apoiou mesmo à distância, principalmente à minha irmã Isabella que sempre esteve comigo durante essa jornada. Agradeço também ao meu orientador Marcelo Luiz Drumond Lanza que tornou este trabalho possível sempre me encorajando e me orientando a buscar as melhores soluções e inovações nas mais diversas áreas.

RESUMO

Este projeto teve como objetivo implementar um sistema que permita realizar trocas de dados entre computadores e dispositivos móveis, geralmente celulares, visando manter a sincronização entre os dados armazenados nestes equipamentos. As trocas em questão são realizada sutilizando-se o protocolo SyncML (*Synchronization Markup Language*) desenvolvido pela OMA (*Open Mobile Alliance*). Este projeto deve atender a todas as requisições contidas na definição do protocolo SyncML.

Apesar do protocolo SyncML, datado de 2001, ser a escolha fundamental para as trocas de dados, existem outros dois componentes de grande relevância. O primeiro deles é um programa para computadores pessoais desenvolvido utilizando-se a linguagem Java que permite o armazenamento das informações desejadas em um sistema de banco de dados (no caso deste projeto o MySQL). Este programa é responsável por realizar a autenticaçãodo usuário, permitindo que o mesmo tenha acesso às informações desejadas, apenas quando a autenticação for realizada com sucesso. As informações armazenadas pelos usuários podem ser do tipo *contatos*ou do tipo *tarefas*. Informações do tipo *contatos* incluem o nome, o sobrenome, o endereço (rua, cidade, estado, pais) eo endereço eletrônico de um contato. Por outro lado, informações do tipo *tarefas* incluem o assunto e a data da tarefa. O segundo componente do sistema é um servidor. Foi utilizado como base para a implementação deste servidor, o software desenvolvido por Nicolas Bougues[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"]e que está disponível na Internet. Este servidor foi implementado utilizando a versão 1.01 do protocolo SyncML. Por consequinte, foram realizadas algumas modificações em seu código para que o mesmo ficasse compatível com o *SyncClient* (cliente executado no celular). Estas modificações dizem respeito aos campos de *DataStores* existentes no código. Principalmente no campo *config*/*configuration*. Além disso, foram realizados testesutilizando-se outros dois outros servidores de sincronização de dados, o Memotoo[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"] e o Funambol[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"] visando-se obter dados para comparações e para o melhor entendimento do protocolo SyncML. Os estudos e as diferenças entre os

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" http://nicolas.bougues.net/syncml/

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" http://www.memotoo.com/

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" http://my.funambol.com/c/portal/layout?p_l_id=default

servidores despontaram como uma grande ferramenta para o entendimento do protocolo SyncML, principalmente no que tange à retransmissão de pacotes.
Palavras-chave: SyncML.Servidor. Sincronia. Informação.

ABSTRACT

The Project Sincronia has the purpose to exchange data between the computer and the mobile system (often represented by cellular, PDA´s). This accomplishment is providing by the SyncML protocol developed by Open Mobile Alliance. In this manner, the project should attend to all requisition necessary and sufficient to proper operation. The reference document is dated from 2001 and is diffused between sponsor and third parts, pointing out to be a new era of technology. However, throughout the protocol represents a fundamental requisite to exchange the information there are other two part of great relevance. The first of them is the desktop program. It was develop in Java language with a MySQL database. This system is responsible for the authentication of the user with the purpose of connect him to his information. The data information can be divided in two different aspects. The Contacts type has the following field: name, last name, address (street, city, state, and country), e-mail. The other type is referred to the tasks (description, and the calendar dates). The second system is the server more specifically the *SyncServer*. The project uses the *open source* Nicolas Bouges Server available on the Internet. The server was developed based on the 1.01 version. Hence some modifications were made in the original code to achieve full compatibility. Those modifications comprised the DataStore fields more especially the config/cofiguration. Besides all that, other analysis turns out to be a great instrument to the study of the protocol. Two servers were study to complete the full observation over the packets transaction. This server were Memotoo and the Funambol. This study in particular showed the benefits of using encrypting files to transmit the messages through the net.

Keywords: SyncML. Server. Sincronia. Packets .

SIGLAS

API – Aplication Programming Interface

AT&T – American Telephone and Telegraph

CEP – Código de Endereçamento Postal

CMD – Comamnd

CMDID – Command Identity

CTCAPS –Content Type Capabilities

DEVINT – Device Information

DNS – Domain Name System

GML – Generalized Markup Language

GPL – General Public License

GUID – Globally Unique Identifier

HTML – Hypertext Markup Language

HTTP – HypperText Transfer Protocol

IBM – International Bussiness Machine

IDE – Integrated Development Environment

ISP – Provedores de serviços de Internet

JDK – Java Development Kit

JDBC –Java Database Connectivity

JRE – Java Runtime Enviroment

LOCNAME – Local Name

LUID – Locally Unique Identifier

MSGID – Messege Identity

MYSQL – My Structure Query Language

OMA – Open Mobile Alliance

PDA – Personal Digital Assitant

PDU – Protocol Data Unit

PHP – Hipertext Protocol

SOURCEREF – Source Reference

SYNC – Synchronize

SYNCCAP –Synchronization Capabilities

SYNCML – Synchronization Markup Language

SGML – Standard Generalized Markup Language

SQL –Structure Query Language

TARGETREF – Target Reference

TCP – Transmission Control Protocol

UFRJ – Universidade Federal do Rio de Janeiro

URI – Universal Identifier Resource

URL – Uniform Resource Locator

URN – Uniform Resource Name

VERDTD –Version Document Type Definition

VERPROTO – Version Protocol

VPN – Virtual Private Network

XML – Extensible Markup Language

WXML – WeatherObjects Markup Language

Sumário

Lista de Figuras

1 – SyncML Framework..... 10

2 – Maneira Simplória de uma requisição por parte do cliente servidor 12

3 – Esquemático de uma transmissão *two-way sync*..... 14

4 – Esquemático de transição de pacotes..... 19

5 – Simulador Blackberry Curve 8520..... 23

6 – Tabelas de Banco de Dados Diagrama 30

6 .1 – Diagrama de Classes..... 31

7 – Tela Inicial..... 40

8 – Tela Nova Conta 40

9 – Tela Contato..... 41

10 – Tela Adicionar..... 41

11 – Tela Tarefa..... 42

12 – Tela Nova Tarefa..... 42

13 – Tabelas de Armazenamento Servidor Nicolas Bougues..... 46

13.1 – Diagrama de 46

Classes.....

14 – SyncClient..... 54

15 – Tela de Sincronização..... 55

16 – Wireshark – Analisador de Protocolo..... 59

Lista de Tabelas

1 – Cenário de Sincronização 13

2 – ID Mapping of Data Item..... 15

3 – Apache -Inicialização por módulo..... 27

Capítulo 1 - Introdução

1.1 Tema

O tema deste trabalho é a manipulação da inguagem de marcação SyncML (*Synchronization Markup Language*) e de suas aplicações. A cada dia, aumenta a necessidade de integrar as mais variadas informações em um só dispositivo, geralmente celulares. SyncML é um padrão de sincronização de informações definido pela OMA (*Open Mobile Alliance*) e independente de plataforma. Ela funciona como um método de sincronização entre celulares e computadores, de maneira a mantê-los, isto é, atualizados com as mesmas informações.

1.2 Delimitação

Este projeto foi desenvolvido a partir de um servidor web desenvolvido por Nicolas Bougues[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp". Foram realizadas adaptações com o intuito de permitir a conexão deste servidor com o programa implementado em Java (*standalone*). O MySQL foi utilizado como servidor de banco de dados. Neste banco de dados são armazenadas as tabelas do servidor e do programa *standalone*.

Para os testes foram utilizados um simulador de celular *BlackBerry*[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"e um simulador de Internet Móvel[1]. Para o serviço de *SyncClient* (agente responsável por enviar a primeira modificação para o servidor e também receber repostas do mesmo) foi utilizado o cliente FUNAMBOL[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp".

---

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" http://nicolas.bougues.net/syncml/

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" http://docs.blackberry.com/ptbr/developers/deliverables/16728/BlackBerry_Smartphone _Simulator-1001926-0618115637-012-5.0-PT.pdf

1 http://us.blackberry.com/developers /javaappdev/javadevenv.jsp

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" http://www.altn.com/Products/Free-Windows-Mail-Server/

1.3 Justificativa

O uso do celular nos dias de hoje vem se tornando indispensável. Sempre buscando estar em contato com novas tecnologias, principalmente com aquelas que despontam como inovadoras, torna-se interessante o entendimento e o desenvolvimento desse projeto. É papel da universidade e dever dos seus alunos englobar novos conhecimentos contribuindo com enriquecimento de todos. Além de se conseguir integrar o computador ao celular através da Internet, este projeto abre caminho para outras idéias em que haja a fusão destes dois importantes mecanismos do novo século.

1.3.1 Ambiente computacional

O celular, nos últimos 30 anos, se tornou um aparelho "multi-tarefas" e essencial. No princípio, ele era utilizado apenas como um telefone móvel. Ninguém ousaria dizer que o celular, aparelho mais ou menos do tamanho de uma maleta, desenvolvido em 1956 pela Ericcson, poderia vir a se tornar tão diminuto e altamente capacitado. Hoje, ele é um verdadeiro gerenciador de documentos, funcionando como um minicomputador portátil. Nesse contexto, a SyncML é uma linguagem nova e promete ser revolucionária na integração de informação entre esses vários recursos.

Podemos analisar cronologicamente a evolução das linguagens até o concebimento da SyncML. Nos primórdios da década de 60 surgiu a GML[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"] (*Generalized Markup Language*) desenvolvida pela IBM. A partir dela surgiu mais tarde a SGML[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"] (*Standard Generalized Markup Language*). A grande utilidade da SGML seria compartilhar documentos que permitissem a leitura por máquinas em projetos de grandes dimensões governamentais e na indústria aeroespacial durante vários anos. A seguir passou a ser utilizada

---

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" Ref Charles F. Goldfarb (1996). "The Roots of SGML - A Personal Recollection"

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" Ref ISO. "JTC 1/SC 34 – Document description and processing languages". ISO. http://www.iso.org/iso/iso_technical_committee.html?commid=45374. Retrieved 2009-12-25.

por outros setores. Não podemos deixar de mencionar que a SGML é uma norma ISO de 1986. As linguagens HTML (Hipertext Transfer Protocol) e XML[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"](Extensible Markup Language) foram derivadas do SGML. Enquanto que a HTML é um aplicação da SGML, a XML é um perfil, um subconjunto específico da SGML. Assim, como exposto acima, a SyncML é uma linguagem de marcação e é baseada nas linguagens XML e WXML (WeatherObjects Markup Language).

1.4 Objetivos

O objetivo deste projeto foi implementar um programa capaz de armazenar dados de um celular em um computador com o intuito de sincronizá-los, utilizando-se para tanto o protocolo SyncML. Os dados selecionados para a sincronização incluem a agenda de contatos e a agenda de tarefas disponíveis no celular. Para atender a este objetivo os seguintes pontos foram atendidos:

a) Fazer um programa que possua uma ligação com um banco de dados capaz de gerenciar uma conta para cada usuário, armazenando os dados deste usuário e mantendo a sincronia entre os dados armazenados no celular e no computador.

b) Sincronizar o celular com o computador a partir da Internet.

c) Fazer a transmissão de dados XML utilizando-se diferentes servidores de sincronização a fim de comparar os resultados com aqueles obtidos com o sistema implementado pela autora.

1.5 Metodologia

Este trabalho visou a implementação de um sistema de sincronização de dados entre dois dispositivos,um sistema central fixo e outro móvel (por exemplo, um celular). O projeto foi dividido em duas partes: a transmissão celular-computador, e a programação dos aplicativos correspondentes ao servidor e ao cliente (sistema *standalone*).

---

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp" Ref Bosak, Jon; Tim Bray (May 1999). "XML and the Second-Generation Web". *Scientific American*. Online at XML and the Second-Generation Web

A escolha do protocolo SyncML para a transmissão de dados talvez tenha sido a escolha fundamental do projeto. Assumindo como referência o protocolo criado pelo consórcio OMA, datado de 2001, este projeto procurou respeitar a todas as definições do protocolo.Uma vez definido o primeiro passo, foram analisadas e pesquisadas as ferramentas auxiliares para a criação e a validação do protocolo, como por exemplo, o simulador de celular que seria utilizado.Independente das ferrramentas utilizadas, a arquitetura deste projeto é a tradicional cliente-servidor. Um analisador de pacotes (o *Wireshark*http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp") foi utilizado para validar a comunicação entre os componentes do sistema. A existência de servidores que implementam este protocolo permitiu a comparação dos resultados obtidos com o sistema desenvolvido pela autora e aqueles apresentados por sistemas comerciais ou não encontrados na Internet.

Os dados que podem ser transferidos entre computadores e celulares são limitados pelas características do protocolo SyncML. Tanto a implementação do programa *standalone*, como a definição das tabelas do banco de dados utilizadas neste projeto devem respeitar as definições do protocolo SyncML. Dentre os servidores e bibliotecas disponíveis na Internet, destacou-se o programa de Nicolas Bougues, escrito em PHP.

Para a implementação do programa *standalone* foi escolhida a linguagem de programação JAVA. Com relação ao servidor de banco de dados, o Navicat da Premiumhttp://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp",(MySQL) foi preferido devido à sua ampla capacidade de armazenamento e ao alto desempenho durante as operações de busca. Com este objetivo traçado, com os protocolos e as linguagens definidas, o próximo passo foi a estruturação das classes e a definição das tabelas internas do projeto.

1.6 Estrutura da Documentação

A documentação consta de oito capítulos. O primeiro remete aos estudos e definições e mteodologia servindo de introdução para os demais. O segundo capítulo mostra a definição e conceitos do protocolo SyncML, ele é uma poderosa arma e mesmo assim possui um código bastante simples. Alguns conceitos com âncoras de sincronismo, mapa de identidades são

---

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"Ref CHAPPELL, Laura. Wireshark network: The official Wireshark Certified Network Analyst study Guide. Laura Chappell Univeersity, Local: editora, 2010

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"Ref http://navicat.com/en/products/navicat_premium/premium_release.html

explanados. O capítulo 3 lista todas as ferramentas e recursos empregados para o desenvolvimento do projeto. Assim sendo, podemos citar a IDE Netbeans, o analisador de protocolos Wireshark responsável pela leitura da comunicação, o simulador balckberry curve (8520) entre outras ferramentas. Os próximos capítulos 4 e 5 explanam as características de delimitações do projeto para o alcance do objetivo. O capítulo 4 é o programa standalone feito na linguagem JAVA com dados armazenados em um banco MySQL. O capítulo 5 por sua vez, é o servidor Nicolas Bougues responsável por atender aos post do celular. Neste capítulo o código é minunciosamente detalhado e também ressalva a necessidade de pequenas modificações devido incompatibilidade de versões. No intuito de abranger a pesquisa sobre o protocolo e realizar comparações, outros dois servidores foram avaliados. Dessa maneira, o capítulo 7 detalha os resultados e avaliações provenientes não só do capítulo 6 mas também dos demais. Para finalizar, tem-se as considerações finais na qual conclui-se o trabalho atingindo a meta pretendida.

Capítulo 2 – Arquitetura do Sistema

2.1 Introdução

O sistema de comunicação entre o celular e o computador foi realizadoutilizando-se o protocolo SyncML. Todavia, antes de identificar os aspectos técnicos, é proposta uma visão macro do processo, identificando-se a função de cada dispositivo.

O protocolo SyncML foi projetado para utilizar uma rede segura, já que normalmente é utilizado para uma intensa troca de informações. Utilizando a topologia cliente/servidorpromove-se segurança no tráfego de dados e velocidade no processamento como ressalta Julio Ross:

> A grande vantagem de se ter um servidor dedicado é a velocidade das respostas às solicitações do cliente, isso acontece porque além de ser especializado na tarefa em questão, normalmente ele não executa outras tarefas.Outra vantagem da rede cliente/servidor é a forma centralizada de administração e configuração, o que melhora a segurança e a organização da rede. (ROSS, 2005, p.4).

2.2 O ProtocoloSyncML

2.2.1 O Framework SyncML

Pela definição de Dirk Riehle em sua tese de doutorado “Framework Desing”, temos:

> Frameworks model a specific domain or an important aspect thereof. They represent the domain as an abstract design, consisting of abstract classes (or interfaces). The abstract design is more than a set of classes, because it defines how instances of the classes are allowed to collaborate with each other at runtime. Effectively, it acts as a skeleton, or a scaffolding, that determines how framework objects. A framework comes with reusable implementations in the form of abstract and concrete class implementations.(RIEHLE, 2000, p.1).

RIEHLE(2000) afirma que *frameworks* representam todo o domínio ao qual a teoria será aplicada, assim como, o *design* abstrato representa todas as classes abstratas (interfaces). O *design* abstrato engloba não somente as classes, como também, todas as suas instâncias que

irão operar entre si. Dessa maneira, o *framework* é o domínio real e imaginário de todo o projeto.

Baseado nessa definição podemos entender o *Framework SyncML*como o conjunto de classes, instâncias e pré-definições necessárias afim de compor o domínio da plataforma para o uso do protocolo SyncML.

A figurafigura 1 mostra uma construção em blocos fornecida pela OMA (*Data Synchronization Work Group*), um consórcio de empresas que se uniram para gerar esta nova tecnologia.Pode-se observar que o *framework* está utilizando o MIME *Media TypeApplication* e que o MIME *Subtype Name* é o "*vnd.syncml*". A própria OMA através de uma mensagem eletrônica ressaltou a peculiaridade do MIME *Type* onde inicialmente identifica-se o tipo de caractere utilizado para especificar a aplicação *syncml*(8bits) (OMA DATA SYNCHRONIZATION WORKING GROUP). Este mnemônico permite a autenticação do remetente originário, além de permitir a execução de comandos remotamente. Todo o cuidado deve ser tomado para execuções remotas afim de ser atingir o objetivo final.

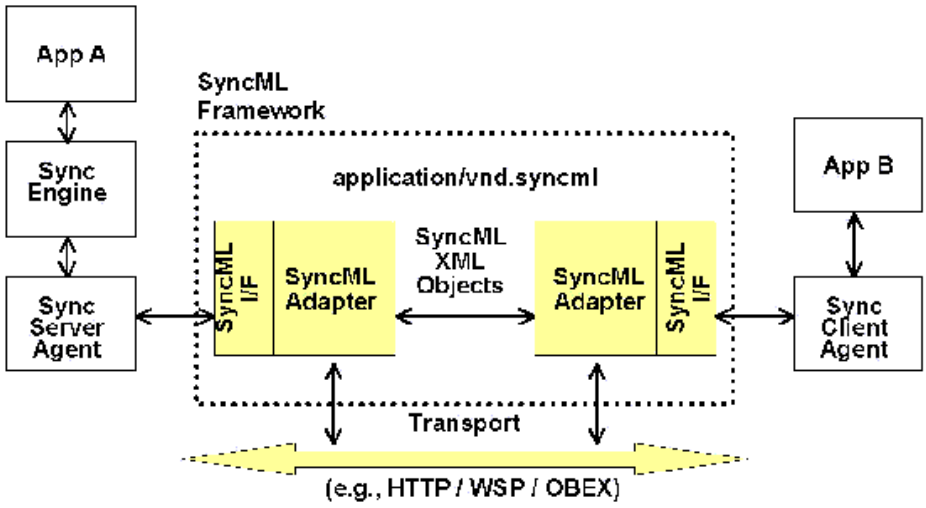


Figura 1 –*Framework SyncML*
Fonte: (Copyright © Ericsson, IBM, Lotus, Matsushita Communication Industrial Co., LTD, Motorola, Nokia, Palm, Inc., Psion,Starfish Software (1999 - 2001) All Rights Reserved)

Com o intuito de minimizar os parâmetros de segurança e utilizar menos *flags* de controle, a OMA buscou utilizar uma linguagem simples que pudesse escrever variados tipos de dados, facilitando a programação dos clientes. A linguagem escolhida foi a XML definida por Havey M. Deitel, em seu livro "XML Como programar", sendo:

> XML é uma tecnologia para criar linguagens de marcação que descrevem dados praticamente de qualquer tipo de uma forma estruturada. Diversamente da HTML, que limita o autor do documento ao uso de um conjunto fixo de marcas, a XML permite que os autores de documentos descrevam dados de formas mais precisas através da criação de novas marcas. Pode ser utilizada para criar linguagens de

> marcação para descrição de dados em quase qualquer campo. (DEITEL, 2003, p. 152).

No diagrama temos que a aplicação “A” (App A) representa um serviço de rede responsável por prover serviço de dados de sincronização para outras aplicações, neste caso aplicação “B” (App B), em outros dispositivos de rede. O serviço é conectado através de um protocolo de transporte de rede comum como o HTTP (HipperText Transfer Protocol).

Na figura 1, temos que a *Sync Engine* (Engrenagem de Sincronização) é completamente dependente do servidor, mesmo que algum tipo de cliente apresente uma engrenagem de sincronização, é obrigatória a funcionalidade deste ao lado do servidor. A *Sync Engine* será o analisador de sincronização como veremos mais adiante.O servidor SyncML (*Sync Server Agent*) e o cliente SyncML (*Sync Client Agent*) utilizam a interface SyncMLI/F para a comunicação um com o outro.

2.2.2 Papel dos Dispositivos

Para melhor entendimento de como estatopologia cliente/servidor operaos agentes envolvidos na configuração são apresentados a seguir (figura 2):

- *SyncML Client*

Este é o dispositivo que contém osdados do cliente o qual envia a primeira modificação para o servidor. O cliente também deve possuir a capacidade de receber respostas SyncML do servidor. Além disso,este será sempre responsável por enviar as modificações primeiro. Em alguns casos, o servidor poderá ter o papel de iniciar a sincronização.

- *SyncML Server*

Este dispositivo contém o agente servidor e a engranagem de sincronização. Geralmente, ele espera até o cliente começar a sincronização para posteriormente enviar as modificações. O servidor é responsável por processar a análise de sincronização (*sync analysis- sync engine*) ao receber as modificações do cliente. Além disso, o servidor também poderá iniciar a sincronização se o protocolo de transporte permitir chamadas não solicitadas.

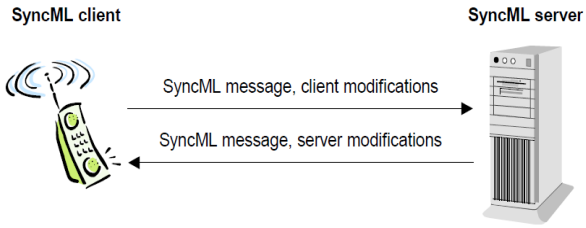


Figura 2 - Maneira simplória uma requisição por parte do cliente para o servidor.
Fonte: (Copyright © Ericsson, IBM, Lotus, Matsushita Communication Industrial Co., LTD, Motorola, Nokia, Palm, Inc., Psion,Starfish Software (1999 - 2001) All Rights Reserved)

2.3 –Iteração Cliente-Servidor

Existem maneiras distintas de fazer a sincronização entre cliente e servidor. Dependendo das condições do meio, esteja ele congestionado, ou livre de perturbações, deve-se escolher a conexão mais segura. Nem sempre uma conexão com altas taxas de troca de dados entre os agentes, é a melhor opção. Além do que, aumentando o tráfego na rede aumenta-se o tempo de sincronização, ficando a mercê da qualidade da conexão. Em contrapartida, uma sincronização rápida, pode acarretar perdas de dados ou mensagens interrompidas forçando uma retransmissão de pacotes, atrapalhando a reconfiguração do sistema. Como podemos ver, o meio é um dos principais fatores que influenciam a programação da sincronização. O quadro abaixo, também fornecido pela OPEN MOBILE ALLIANCE (2001b), fornece o cenário de sincronização:

Cenário de Sincronização	Descrição
Two-way sync	Tipo de sincronização normal onde o cliente e o servidor trocam informações sobre os dados modificados nos dispositivos. O cliente envia as modificações primeiro.
Slow sync	Uma forma derivada do *two-way sync* onde todos os itens são comparados com o outro em um modelo de verificaçãocampo a campo básica. Na prática, isto significa que o cliente envia todos os dados de seu banco para o servidor, e o servidor faz a análise de sincronização (campo a campo), desta informação com a informação retida em seu próprio banco de dados.
One-way sync from client only	É um tipo de configuração onde o cliente envia suas modificações para o servidor, mas, o servidor não envia suas modificações de volta para o cliente.
Server	Um tipo de sincronização onde o servidor alerta o cliente realizar a

Alerted Sync	sincronização. Isto é, o servidor informa ao cliente para iniciar a sincronização.

Tabela 1 – Cenário de Sincronização.
Fonte: OPEN MOBILE ALLIANCE, 2001b.

Mais a frente,serãoressaltadosos cenários *two-way sync*, *slow sync*,*one-way sync* e a condição *server alerted sync*. Entretanto, convém primeiramente, mostrar como ocorre o processo chamado âncoras de sincronização, responsável pelo gerenciamento de informação. Além do que, este é o cerne do protocolo. Estas âncoras são efetuadas nas primeiras trocas de informações onde são identificadas as interfaces, o usuário, o servidor, e as últimas atualizações.

2.3.1 Âncoras de Sincronização

Âncoras de sincronização, como dito anteriormente, são o cerne do protocolo SyncML. São utilizadas como controle para a verificação da sincronização. Existem, por definição, duas âncoras *Last* e *Next* que sempre são enviadas no inicio da sincronização. A âncora *Last* sempre determina o último evento (instante) quando a base de dados foi sincronizada, do ponto de emissão de dados. Já a âncora *Next* descreve o evento atual, também do ponto de emissão de dados. Ambos, cliente e servidor, enviam suas âncoras, um ao outro. O dispositivo receptor tem que retransmitir a âncora *Next* de volta para o dispositivo transmissor no campo *Status* do comando *Alert* (elemento predefinido dentro do comando *Status* no XML).

A utilização das âncoras de sincronização torna o processo viável, somente se, o *SyncServer* armazenar em seu *Storage* a âncora *Next*. Desta maneira, esta âncora *Next* passará a ser a âncora *Last* na próxima sincronização. Se na próxima sincronização não houver compatibilidade entre as âncoras, o servidor retornará um pedido de solicitação (*request*) especial (*Slow Sync*) de maneira a manter a sincronização sempre correta. Se não houvesse esse sistema de compatibilidade poder-se-ia ter, facilmente, dados corrompidos. O diagrama a seguir (figura 3) ilustra a transmissão *two-way sync*:

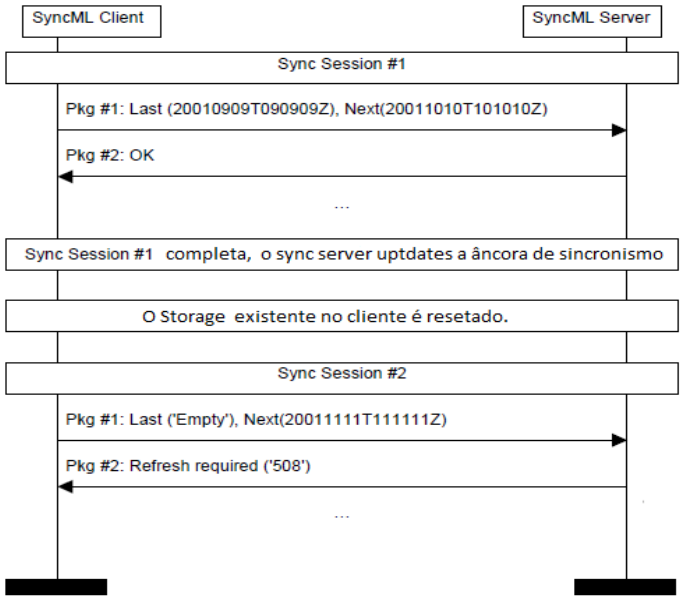


Figura 3 – Esquemático de uma transmissão two-way sync.
Fonte: OPEN MOBILE ALLIANCE, 2001b.

Pode-se observar na figura 3 que a primeira *Sync Session* ocorre de maneira correta. Entretanto, quando o *Storage* do cliente é resetado, passa a existir uma não compatibilidade entre o *Storage* do servidore a âncora *Last* do cliente. Dessa maneira, o servidor retransmite ao emissor um pedido de *Refresh Require*. Será iniciada a *Slow Sync*.

2.3.2 Mapa das identidades dos Itens

Uma característica importante do protocolo SyncML é a necessidade de, tanto o servidor quanto o cliente, possuírem ID´s (identificadores) para a identificação dos *Data Item* (âncoras). Todavia, não há necessidade das partes possuírem o mesmo identificador. Basta existir um correlação entre o ID do cliente (LUID – *Locally Unique Identifier*) e o ID do servidor (GUID – *Globally Unique Identifier*) sempre apontando para o mesmo endereço (ID).

Para que esta técnica seja empregada de maneira suficiente e satisfatória, tem-se a presença de uma tabela de mapeamento no servidor. Esta tabela liga os GUIDs com seus respectivos LUIDs. O *SyncML Sync Protocol*, version 1.0.1, exemplifica a técnica descrita acima através do diagrama mostrado na figura 3 (OMA, 2001b).

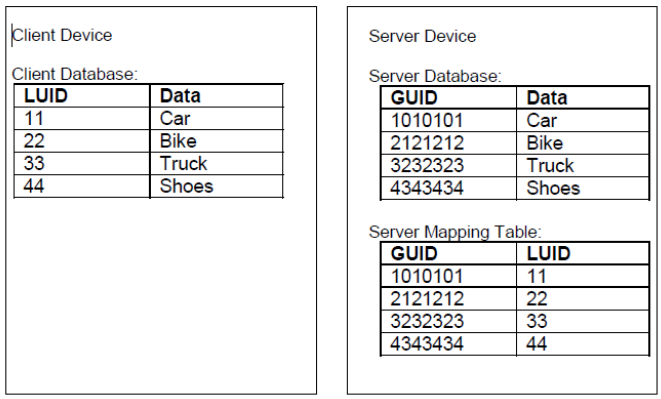


Tabela 2 – Mapeamento de dados
Fonte:OPEN MOBILE ALLIANCE, 2001b.

Na tabela acima, tem-se uma informação *Car* escrita na base de dados do Cliente sob o LUID 11. Por outro lado, no banco de dados do servidor este mesmo identificador está sobre o GUID 1010101. Com o intuito de não permitir erros de comparação de informação (*sync analisys*), temos uma tabela de mapeamento informando a correlação entre o GUID e o LUIDno servidor.

Mesmo que o servidor necessite inserir novos itens no cliente, supondo uma atualização de dados a partir do servidor, será necessário um tipo de requisição de dados para o cliente. Isto significa que o servidor envia novos itens ao cliente e fica aguardando os novos LUIDs para preencher a tabela de mapeamento.

2.3.3 Endereçamento

O código syncML é basicamente composto de duas partes. O SyncML *header* e o SyncML *body*. O SyncML *header* contém todos os itens necessários para a identificação da interface, seja ela cliente ou servidor, além da informação sobre o dispositivo e para a autenticação do usuário, caso necessário. Já o SyncML *body* contém as estruturas de controle de execução remota, neste protocolo traduzidos pelos campos*Alert* , *CmID* (*Command ID*), *Data*.

Para que todas as informações, de execução e de identificação, sejam efetuadas com sucesso é necessário que o endereçamento de origem e de destino seja realizado de maneira correta e não dúbia.

O sistema de endereçamento é realizado utilizando-se um esquema de URI (*Universal Idetifier Resource*). Ninguém melhor indicado que T. Bernes - Lee para se ter o conceito do

URI.Inventor da *World Wide Web* ele define no RFC 1630 (*Request for Coments* 1630) o que seria URI :

> *Universal Resource Identifier* (URI) is a member of this universal set of names in registered name spaces and addresses referring to registered protocols or name spaces. A *Uniform Resource Locator* (URL), defined elsewhere, is a form of URI which expresses an addresswhich maps onto an access algorithm using network protocols. Existing URI schemes which correspond to the (still mutating) concept of IETF URLs are listed here. The *Uniform Resource Name* (URN) debate attempts to define a name space (and presumably resolution protocols) for persistent object names. This area is not addressed by this document, which is written in order to document existing practice and provide a reference point for URL and URN discussions.(LEE, 1994).

Berners informa que URI é um identificador universal de recurso, membro de universo de nomes registrados, podendo ser espaços de nomes ou endereços, referentes a protocolos registrados. Portanto, o URL (*Uniform Resource Locator*) é uma forma de URI que expressa os endereços e que será mapeado sobre um algoritmo de acesso usando protocolos da rede. Já o URN (*Uniform Resource Name*) tenta definir um espaço de nomes para objetos constantes.

O serviço de endereçamento da SyncML (OMA, 2001b) pode ser feito baseado no endereçamento URI da seguinte maneira:

```
<Source>
<LocURI>http://www.syncml.org/sync-server</LocURI>
</Source>
```

Para dispositivos conectados temporariamente, geralmente clientes (celulares, PDA, etc.) é preferível utilizar a seguinte configuração (OMA, 2001b):

```
<Source>
<LocURI>IMEI:493005100592800</LocURI>
</Source>
```

2.3.3.1 Endereço do banco de dados

Uma vez configurados a origem e o destino, o protocolo permite especificar o diretório para o banco de dados do cliente e do servidor. Para isso utiliza-se o campo *Target* na estrutura de controle. (OMA, 2001b).

```
<Sync>...
```

```
<Target>
<LocURI>./calendar/james_bond</LocURI>
</Target>
...</Sync>
```

Para o servidor, tem-se:

```
<Sync>...
<Target>
<LocURI>
http://www.syncml.org/sync-server/calendar/james_bond
</LocURI>
</Target>
...</Sync>
```

2.4 Autenticação

O protocolo SyncML foi criado pensando na necessidade do provedor de serviços validar os seus usuários (autenticação de usuários). Para isso, existe por definição, *flags* de autorização permitindo o estabelecimento da conexão para a inicialização da sessão. Caso não seja autorizada, a sessão é encerrada mantendo a integridade dos dados intacta. Se o código deresposta for igual a 407(Não Autorizada), será necessário fazer a autenticação.A programação varia de acordo com o desejado pelo analista/ desenvolvedor do sistema.

Ambos, cliente e servidor, podem requerer autenticação (*Authentication Challenge*).

Se a resposta do *challange* for igual a 201 (*Status*) a autenticação foi aceita.

No caso da autenticação falhar (identificador ou senha do usuário incorreta),as credenciais deverão ser enviadas novamente para o servidor.

Abaixo, segue o código com as credenciais para a autenticação (cliente) (OMA, 2001b):

Pkg #1 (with credentials) from Client

```
<Cred>
<Meta><Type xmlns='syncml:metinf'>syncml:auth-
basic</Type></Meta>
<Data>QnJ1Y2UyOk9oQmVoYXZl</Data>
<!—base64 formatting of "userid:password"-->
</Cred>
```

2.5 Opções de Sincronização

Existem vários métodos de sincronização entre o *SyncClient* e o *SyncServer*. Todavia, há dois métodos a serem destacados: *two- way sync*e o *slow sync*.

Uma vez realizada a sincronização inicial (âncoras de sincronização), o protocolo *SyncML* estará apto a transferir informações. O método *two way sync* é o padrão, enquanto que o método *slow sync* só é utilizado caso o primeiro seja corrompido.

2.5.1 *Two-Way Sync*

Se fosse necessário fazer uma tradução para *two-way sync* seria equivalente à sincronização de dois caminhos. O procedimento remete ao seu título já que para a finalização do procedimento existe uma transmissão nos dois sentidos.

Supondo que a sincronização inicial tenha sido estabelecida, será necessário transmitir os dados (por exemplo, contatos, tarefas). Primeiramente, o cliente inicia o procedimento informando ao servidor sobre as últimas modificações desde a última sincronização. Se for a primeira, somente os novos dados deverão ser enviados.O servidor, por sua vez, recebe todas as informações e começa a análise de dados de maneira à comparar e a unificar toda a informação. Depois que todo o conteúdo é analisado, o servidor atualiza o seu banco e envia de volta a informação ao cliente. Somente assim, este último poderá atualizar o seu banco de informações.

Uma vez o que o cliente tenha atualizado os seus dados, este deverá emitir todos os seus LUIDs para que o servidor possa atualizar a tabela de mapeamento correspondente. Isto é muito importante, pois o servidor necessita das informações de correspondência para futuras análises de dados.

Para finalizar o processo, o servidor envia uma informação de *acknowledge* ao cliente para informá-lo que recebeu o pacote. Esta informação é enviada mesmo que não haja mudanças nos LUID antigos. Sem ela o processo estará incompleto e será desconsiderado.

Caso haja uma quebra na comunicação ou qualquer outro problema de armazenamento de dados, os bancos de ambas as partes não serão atualizados e um outro pedido de

sincronização será feito. Vale ressaltar, que neste caso, o *two-way sync* será descartado, e o *slow sync*será acionado.

A figura 4 mostra um exemplo do procedimento descrito acima.

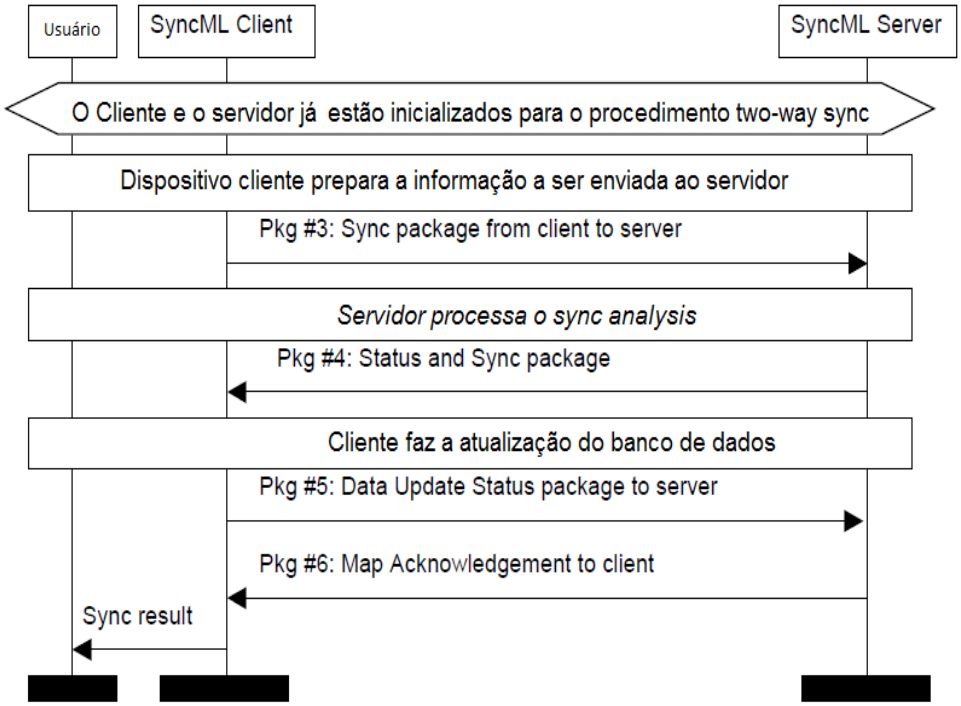


Figura 4 - Esquemático da transição dos pacotes.
Fonte:OPEN MOBILE ALLIANCE, 2001b.

2.5.1.1 Slow Sync

Segundo a OMA em seu protocolo SyncML versão 1.1 temos:

> *The slow sync can be desired for many reasons, e.g., the client or the server has lost its change log information, the LUID's have wrapped around in the client, or the sync anchors mismatch. The slow sync is a form of the two-way synchronization in which all items in one or more databases are compared with each other on a field-by-field basis. In practise, the slow sync means that the client sends all its data in a database to the server and the server does the sync analysis (field-by-field) for this data and the data in the server. After the sync analysis, the server returns all needed modifications back to the client. Also, the client returns the Map items for all data items, which were added by the server.*(OMA, 2001b).

Pode-se extrair deste trecho que vários são os motivos para uma possível troca da sincronização de dados para a forma *Slow Sync*. Estes motivos incluem a falta de correlação

entre as âncoras, a perda no registro de mudança, etc. Além disso, tanto o cliente (*SyncClient*)quanto o servidor (*SyncServer*) podem requerer o método. Ainda assim, quando chamado, ele realiza uma comparação campo à campo de um ou mais banco de dados. Se a requisição for realizada pelo cliente, o servidor enviará as modificações de volta depois de ser realizada a apuração dos dados.De maneira bastante sucinta temos que o *slow sync* é derivado do *two way sync*, porém com uma análise de dados mais trabalhada.

2.5.1.2 *Server Alerted Sync*

Está é uma opção diferente de todas as citadas até agora, pois o usuário enviará uma mensagem através do servidor para o celular (*SyncClient*) iniciar uma sincronização. Na realidade, existe outro software cliente que envia a mensagem de requisição, geralmente um POST, ao servidor. Ao invés do servidor responder a este software, a mensagem é destinada ao *SyncClient*. Para que os trâmites ocorram de maneira correta, é necessário que as informações do dispositivo[http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"] estejam disponíveis no software cliente. Além da informação dos dispositivos ainda é acrescentado o tipo de sincronização.

2.5.2 *One Way Sync*

Este tipo de sincronização é diferente do *two-way sync* pois as modificações só ocorrem em um sentido. O *one way sync from client* envia ao servidor todas as modificações contidas no banco de dados do S*yncClient* (celular).Entretanto, ao invés de receber as modificações do servidor, ele recebe um pacote de *Status*. Dessa maneira, o servidor fica atualizado com todos os contatos do *SyncClient*, enquanto que o *SyncClient* não obtém informação nenhuma do servidor.

Outro tipo de *one way sync* é o proveniente do servidor (*one way sync from Server*). O princípio de funcionamento é o mesmo mudando apenas o remetente e o destinatário.

---

http://us.blackberry.com/developers%20/javaappdev/javadevenv.jsp"

O *SyncClient* envia uma requisição de sincronização ao servidor. Desta vez, a atuador é o*SyncServer,*responsável por enviar as modificações presentes em seu banco de dados ao receptor. O *SyncClient* recebe as modificações e envia um sinal de *Status Update*. Por fim, o servidor transmite ao cliente dados correspondentes ao mapeamentode identificadores (*map acknowledge*), conforme descrito na seção 2.3.2, informando o fim da transição.

Capítulo 3 – Estudo Preliminares

3.1 Ferramentas de Apoio

Algumas ferramentas foram utilizadaspara integrar e construir a estrutura básica de desenvolvimento do projeto. O sistema pode ser dividido em duas partes distintas:

- *Software* escrito utilizando a linguagem JAVA e servidor de banco de dados MySQL;
- Transmissão de dados via protocolo SyncML para um servidor escrito em PHP.

Maiores detalhes e especificaçõessobre o *software*esobre a transmissão cliente-servidor, serão fornecidos no capítulo 4. Esta sessão tem o intuito de apresentar as ferramentas de apoio, seus benefícios e possibilidades de uso, além das características técnicas que serão necessárias para o melhor entendimento da análise do projeto.

3.1.1 Simulador Blackberry Curve (8520)

Como cliente foi utilizado o simulador do *Smartphone* Curve (8520) da Blackberry (figura 5). A empresa fornece o software para desenvolvimento de novas aplicações e torna pública suas possibilidades de utilização e serviços como ressaltados no Guia do Desenvolvedor (RESEARCH IN MOTION BLACKBERRY, 2010).

> O BlackBerry® permite executar aplicativos do dispositivo BlackBerry no computador. O BlackBerry *Smartphone Simulator* inclui os aplicativos que normalmente estão disponíveis nos dispositivos BlackBerry e permite que você carregue e teste seus próprios aplicativos. É possível simular e testar várias alterações de estado e conectividade usando o BlackBerry *Smartphone Simulator*. Ao usar o simulador para realizar testes, convém simular também outros serviços BlackBerry. O BlackBerry MDS *Simulator* e o BlackBerry *Email Simulator* estão disponíveis para essa finalidade.

Como consta no trecho acima utilizaremos o serviço MDS (Mobile Data System) para fazer a conexão com a Internet. Desta maneira, para que a instalação ocorra corretamente é

necessária a instalação do *Java SE Development Kit* fornecido pela *Sun Systems,* como será explicado mais tarde.

Figura 5 -Simulador Blackberry Curve 8520.
Fonte: RESEARCH IN MOTION BLACKBERRY, 2010.

3.1.2 Serviço Blackberry -*Mobile Data System* MDS

O que torna possível a execução do sistema é emulação da rede *Edge* através do serviço *Blackberry* de MDS.

> To allow a BlackBerry® Java Application access to resources behind the corporate firewall, the BlackBerry® Enterprise Server includes the BlackBerry® Mobile Data System. The BlackBerry MDS provides HTTP and TCP/IP proxies for a BlackBerry Java Application, which allow the BlackBerry device to communicate with application and web servers behind the corporate firewall without additional VPN software. Applications that send data using the BlackBerry Enterprise Server as a gateway can capitalize on the simplified enterprise connectivity, data encryption and compression, and wireless network-independence that the BlackBerry® Enterprise Solution offers. BlackBerry MDS also provides an open interface, allowing server-side applications behind the corporate firewall to push content to applications on BlackBerry devices.(RESEARCH IN MOTION BLACKBERRY, 2008).

Doo trecho acima vemosque,com o intuito de possibilitar o estabelecimento de conexões entre o simulador e a *Internet*, a empresa fornece um serviço de *proxies* para HTTP (Hyppertext Transfer Protocol) e TCP (Transmission Control Protocol) /IP (Internet Protocol) para aplicações JAVA. Isso permite que o dispositivo *BlackBerry* estabeleça conexões com servidores WEB, mesmo quando estiver em uma rede protegida por um *firewall*corporativo e a sem necessidade de softwares de VPN (Virtual Private Network). Aplicações que enviam dados utilizando o Servidor *BlackBerry Enterprise*c omo *gateway* podem simplificar a

conectividade, a encriptação de dados, a compressão de dados, além simplificar o uso de redes sem fio independentes oferecidas pelo próprio serviço *BlackBerry*. O MDS também provêuma interface aberta, permitindo que servidores protegidos por *firewalls*se comuniquem com aparelhos BlackBerry.

3.1.2.1 Software cliente/Servidor (FUNAMBOL)

Apesar prover os recursos acima, por não ser um membro do consórcio OAM, oscelulares da *BlackBerry*, não possuem o *SyncML Client* em sua programação básica. Como solução para atender a esse requisito, neste projeto utilizou-se recursos da FUNAMBOL.

A FUNAMBOL é uma corporação norte-americana que trabalha com um modelo de negócio de licença duplo, que inclui uma licença para software comercial e uma licença para software livre. Seu principal projeto, voltado para a sincronização de dados móveis, chama-se *Funambol Core Project.*

O *Funambol Core Project* é um software livre e *Open Source* de troca de dados móveis entre os clientes (geralmente celulares e PDA´s) e servidores. Os serviços oferecidos incluem: sincronização de *e-mails*, sincronização de calendários, sincronização de listas de contatos(PIM) e gerenciamento de dispositivos sem fio utilizando o protocolo SyncML. Além disso, as aplicações podem ser programadas nas linguagens C++ e JAVA, facilitando o desenvolvimento.

Neste projeto, utilizou-se uma de suas APIs (significado) para o cliente *BlackBerry* que utilize o sistema operacional com versão 4.7 ou superior e programado em JAVA. O *Sync Client* atende a todos os requisitos, podendo realizar a sincronização de contatos, calendários e tarefas.

Além do *Sync Client*, foram realizados alguns testes com o *Sync Server* da FUNAMBOL visando análises do protocolo SyncMLutilizando o analisador de pacotes (*Wireshark*). Os resultados serão apresentados no Capítulo 6.

3.1.3 Memotoo

Outro servidor utilizado para os testes do protocolo SyncML foi o Memotooo. Este servidor é aplicação francesa que presta serviço de sincronização móvel desde 2001. Ele permite contas de livre acesso para testes com capacidade reduzida. A versão *Premium* é paga.

3.1.4 Wireshark

Uma vez definida a infraestrutura de transmissão de dados foi necessário definir uma ferramenta que fosse capaz de acompanhar a transmissão em questão. Para isso, foi escolhido o *Wireshark (ver fig 16, p59).*

Este analisador de pacotes é livre,código aberto e licenciado sob a *GNU General Public License* (GPL).Pode ser utilizado para inúmeros e distintos fins. Por exemplo, administradores de redes o utilizam para investigar possíveis problemas na rede, engenheiros de segurança de rede o utilizam para examinar problemas de segurança. Já desenvolvedores o empregam para a depuração de implementações, enquanto que pessoas em geral podem utilizá-lo para aprendizado sobre protocolos de rede.

O programa é capaz de captar grande parte do fluxo de informação que trafega na rede. Entretanto, dependendo do tamanho da rede e da quantidade de usuários, esta opção é indesejada. Para contornar esta situação foram desenvolvidas técnicas de filtragem. Filtros, como o próprio nome sugere, são regras pré-programadas responsáveis por realizar uma espécie de busca somente nas opções desejadas, como acontece em *queries* programadas no banco de dados. Empregando este recurso, tem-se a visualização somente da comunicação de interesse. É possível definir filtros para a captura direcionada entre dois IP específicos, o que, para o desenvolvimento da comunicação deste projeto, foi imprescindível.

3.1.5 Servidor em IP Dinâmico (NO–IP)

A *No–IP* é uma empresa que promove serviços pagos ou não, de DNS (Domain Name System) dinâmico, monitoramento de rede, certificados, filtros de *e-mails*, entre outros. O principal serviço desta empresa é o DNS dinâmico que pode ser utilizado de forma gratuita por um período (caso a conta fique inutilizada por 45 dias o sistema remove o DNS)

Os provedores de serviço de Internet (ISP – Provedores de serviço de Internet), comumente atribuem ao usuário um endereço IP dinâmico. Muitas vezes, isso se dá porque a quantidade de clientes que contrataram o serviço é maior do que a faixa de endereços IP disponível. Para que um servidor possa ser acessado na Internet, é necessário que o mesmo possua um endereço IP estático, de forma que cliente possa acessar o servidor a partir de um nome de máquina (por exemplo, www.memotoo.com) utilizando o serviço de DNS. Quando o cliente possui um endereço IP dinâmico,a resolução de nomes pode ser tornar mais complexa.

Visando resolver o problema criado pelo uso de endereçamento IP dinâmico, foi desenvolvido o DNS dinâmico, que torna possível a hospedagem de uma página HTML ou a criação de qualquer servidor IP dentro deste contexto.

O princípio de funcionamento do DNS dinâmico é bastante simples. O usuário instala um executável, fornecido pela empresa correspondente (por exemplo, NO-IP), na sua máquina e escolhe um subdomínio e nome que será utilizado para referenciar a sua aplicação. Este executável é responsável por identificar o endereço IP atribuído pelo ISP, e por enviá-lo para o endereço da empresa de DNS dinâmico. No servidor da empresa existe uma tabela a qual faz a correspondência entre os endereços IPe a URL´s (significado). Toda vez que houver uma troca no endereço IP do usuário, esta correspondência será atualizada. É necessário habilitar a função *"refresh"* do executável para que toda vez que a máquina for ligada, o servidor central receba esta informação.

Para o projeto foi criada a URL sincronia.no-ip.org. E o redirecionamento foi feito para a porta 8080.

3.1.6 MySQL

O MySQL é um banco de dados antigo,de código aberto, desenvolvido para combater os bancos de dados comerciais. Com o passar dos anos acabou se tornando umaferramenta poderosa, robusta e flexível,para se adaptar à concorrência. Por manter parte do seu código aberto está bastante difundida em vários outros aplicativos como ressalta Ulmann (2006, p.54).

DYER (2008) revela que uma das principais característica do MySQL está na relação velocidade por escalabilidade. O MySQL é famoso por sua escalabilidade, suportando dezenas de milhares de tabelas e chegando a possuir bilhões de fileiras de informação. Mesmo nos casos em que armazena volumes muito grandes de dados, as buscas (*queries*) são executadas rapidamente.

O aplicativo utilizado para a construção das tabelas do bancos de dados foi o *MySQL Workbench*. Ele é um *software* livre, de código aberto, mutiplataforma, empregado para o projeto de banco de dados (MySQL). Está disponível para as plataformas *Windows* e sistemas operacionais derivados do *Unix* (BSD's, Linux, etc.).

Este sistema foi desenvolvido a partir do *Generic Runtime Environment*, o que o torna facilmente expansível para o aperfeiçoamento de cada um que desejar ser desenvolvedor. Para os fins deste projeto não iremos entrar neste mérito,mas sim, ratificar a compatibilidade deste recurso com o *Navicat* que será nosso banco de dados.

Todavia, apesar do *MySQL Workbench* ser uma excelente ferramenta para o projeto de tabelas de banco de dados, ele não possui um sistema de armazenamento. Como sistema de gerenciamento de banco de dados utikizou-se o *MySQLNavicat* versão *Lite*. O *MySQLNavicat* é um aplicativo da *Premium Software* que oferece a versão *Lite* com algumas restrições quando comparado com as versões*Stand* e *Premium*, mas ainda assim, é bastante funcional e prático, além de ser livre.

3.1.7 Java

Java, senão é a linguagem mais popular dos dias atuais, com certeza, está entre as mais utilizadas no mercado. Porém, apesar de bastante difundida, ainda ocorrem sérias confusões em torno da linguagem de programação Java, da máquina virtual Java e da plataforma Java. O

autor Flanagan (2005, p.1) em seu livro tenta explicar esta diferença de uma maneira bastante simplória. Flanagan diz que a linguagem Java é escrita em seus aplicativos, sejam eles *applets, servelets,* ou outros componentes, para posteriormente serem compiladas.

Após os códigos serem compilados, o algoritmo escrito passa a ser uma linguagem de máquina (conjunto de bytes).Estes, por sua vez, serão interpretados pela *Java Virtual Machine* a fim de se obter resultados interpretados e executados. Por fim, a plataforma Java consta no conjunto predefinido de classes existentes em toda a instalação Java. Muitas vezes a plataforma Java pode ser chamada de *Java Runtime Environment* (JRE).

Para que novos aplicativos possam sercriados,é necessário obter o *Java Development Kit* (JDK) disponilizado pela *Suns Microsystems*.Encontram-se disponíveis versões do JDK para os sistemas Operacionais Unix, Windows, e Solaris. O JRE funciona somente para executar aplicações existentes (*Flanagan – Java in a Nutshell pg 7/1147*).

3.1.7.1 NetBeans IDE

Com o objetivo de facilitar a programação e a utilização de recursos, emprega-se a IDE (*Integrated Development Environment*) *NetBeans*. Ele pode funcionar como um *framework* de aplicações *standalone* ou como uma IDE, a qual abrange não somente a linguagem Java, mastambém as linguagens C++, Java Script, PHP, Python, entre outras. O *NetBeans* possui vários recursos que tornam a programação mais flexível, facilitando o reconhecimento de recursos. Além disso, existe um grande de banco de informações sobre o *NetBeans* à disposição.

Para a conexão entreo *NetBeans* e o *Navicat* foi utilizado um conector, ou seja, o“mysql-connector-java-5.1.13”.

3.1.7.2 Programação Orientada a Objeto

A programação orientada a objeto é um recurso fundamental na programação moderna. Apesar de ser uma técnica bastante utilizada no século XXI ela é datada do início dos anos 70 com a criação da primeira linguagem completamente orientada a objeto, a

SmallTalk, da Xerox (PARC). Posteriormente, nos anos 80 foi desenvolvida, pela AT&T, a linguagem C++, que dominaria o mercado. (SANTOS, 2009).

A teoria da orientação a objetos provém de um dos primeiros pensadores, Aristóteles. Em seu estudo chamadoTeoria das Categorias, um dos seis trabalhos lógicos que compõem o livro Organon (XAVIER, 2008),Aristóteles experimenta "categorizar" o mundo. Para isso, ele define padrões que sigam um mesmo comportamento (métodos) e características (atributos). A estes padrões ele chama de objetos. Quaisquer regras utilizadas para regimentar estes objetos são chamadas de classes, dessa maneira pode existir mais de um objeto em uma classe. Definindo estas regras o homem conseguiu quantificar e classificar, coisas (agora objetos), através de classes e famílias.

O desenvolvimento do *software* é organizado da mesma maneira. Inicialmente devem-se definir os objetos e necessidades que o aplicativo deve possuir para posteriormente desenvolver suas classes. Um dos principais recursos utilizados na orientação a objeto na programação é a herança entre as classes. Uma vez que a classe é filha de outra classe, ele herda todos os métodos e atributos de seu "genitor" sem que haja a cópia do código. Isso simplifica o código e o torna enxuto.

3.1.7.3 Conector MySQL/JAVA

Outra ferramenta de vital importância para este projeto é o conector *mysql-connector-java-5.1.13*.Por definição da própria MySQL, este software fornece conectividade para aplicações cliente desenvolvidas em linguagem JAVA via JDBC (*Java Database Connectivity*) driver.

Uma aplicação JDBC é um padrão industrial para banco de dados independentes entre a linguagem de programação JAVA e a maciça variedade de banco de dados SQL ou não. Esta API fornece acesso aos bancos (ORACLE COMPANY).

```
public conexaobanco(){
this.driver = "com.mysql.jdbc.Driver";
        this.url = "jdbc:mysql://localhost:3306/banco1";
        this.login = "root";
        this.senha = "";
    }
public void conectar() throws Exception {
       Class.forName(this.driver).newInstance();
```

this.conexao = DriverManager.getConnection(url, login,senha);
}
Fonte: Código em Anexo

Aqui se tem uma demonstração de como o conector atua. Supondo que os atributos tenham sido previamente definidos, eles serão instanciado na parte inicial do código. A *stringDriver* recebe o nome definido por padrão do software. Já o campo URL remete ao diretório da máquina (*localhost* - 127.0.0.1), 3306 é porta correspondente e banco1 é nome do *Database* dentro do Navicat. A autenticação é feita através de um *login* e de uma senha.Cada usuário deverá ter sua conta e consequentemente sua própria autenticação

3.1.8 Linguagem PHP

Outra linguagem utilizada no projeto é o PHP (Hypertext Preprocessor). Esta é uma linguagem *server-side*, ou seja, o código reside no lado do servidor ao invés de ser descarregado para o computador do cliente. Além disso, ele permiteacesso a códigos armazenados em um servidor remoto ou local, além de permitir a recepção e processamento de informações HTML (BROOKS, 2008).

A definição da linguagem PHP pode sercomplementada por Carlos Costa em seu livro *Desenvolvimento para Web*. Carlos afirma –"PHP éuma linguagem Open-Source, que é executado ao lado do servidor, embebida no HTML e de *scripting* compatível com os principais servidores de Web, nomeadamente Apache".

Em nosso projeto o servidor do Nicolas Bouges está escrito em PHP. Como foi necessário aprender as funções e fazer modificações no código, o PHP foi empregado como ferramenta auxiliar.

3.1.9 Servidor HTTP Apache

O servidor HTTP Apache foi desenvolvido pela *Apache Foundation Software* a partir de 1999 nos EUA.Éum servidor web baseado no protocolo HTTP. Como o *software* é um programa complexo e com vários atributos, o estudo desta ferramenta foi focado somente para as opções de controle necessárias para o funcionamento no sistema operacional utilizado (*Windows* 7) e nas modificações de redirecionamento de porta (DNS Dinâmico).

O servidor Apache foi projetado para ser flexível de maneira a trabalhar com uma variedade de plataformas diferentes, em diferentes ambientes. Plataformas distintas e diferentes ambientes, geralmente requerem configurações especiais para cada item, ou definem maneiras diversas para se obter a mesma configuração. Para suplantar este quesito e, assim ser flexível, o Apache desenvolveu o projeto por módulos (*Multi-Processing Module*).Neste esquema, o *webmaster* decide quais módulos deverão ser selecionados para a inclusão no servidor no tempo de compilação, ou quando o programa já estiver rodando (*Apache Sofware Foundation*- Apache HTTP Server 2.2 Server Administration, 2010).

Podemos ver na tabela 3 os módulos que serão carregados na compilação caso não haja interferência do administrador:

BeOS	beos
Netware	mpm_netware
OS/2	mpmt_os2
Unix	prefork
Windows	mpm_winnt

Tabela 3 - Apache inicialização por módulos.
Fonte:FULTUS CORPORATION, 2010.

Como o sistema operacional utilizado para o desenvolvimento foi o *Windows 7,* pode-se retirar os outros módulos.

O arquivo de configuração do Apache chama-se httpd.config e está no seguinte diretório: \...\apache\Apache2.2.11\conf\httpd.conf.

O arquivo de configuração contém as diretivas de funcionamento, além dos módulos de multi-processamento definidos acima. Entre estas diretivas, está a diretiva responsável por fazer a escuta da porta do programa. Ela está definida através do comando *Listen* no arquivo de configuração. Por definição, a porta padrão do Apache é a 80. Entretanto, na tentativa de evitar conflitos com outros programas, ela pode ser modificada para outra porta, como por exemplo a porta 81 ou 8080. É de suma importância fazer esta alteração, uma vez que, este

arquivo será carregado no *Firewall* do sistema para liberar o tráfego. Caso contrário, o servidor não irá funcionar.

Capítulo 4 - Projeto

4.1 Especificação

Após as ferramentas terem sido definidas e explicadas, esta sessão contémas especificações do programa.

Inicialmente, o projeto foi idealizado e pré-definido através das tabelas do banco de dados criadas com a ajuda *Workbench*, atendendo às limitações e às exigências da programação do cliente. Isto quer dizer que os tipos dos campos definidos no celular foram levados em consideração para evitar incompatibilidades futuras.

Uma vez definidos os campos e as interações entre as tabelas, foi possível categorizar os objetos e, conseqüentemente, as classes. O recurso da herança foi bastante utilizado para tornar o código mais limpo e sucinto.

Por fim, foram implementadas as interfaces gráficas a fim de proporcionar ao usuário um conhecimento intuitivo, evitando a necessidade de complexos recursos de ajuda.

Ao longo deste capítuloserão expostos gráficos e tabelas que têm como principal objetivo facilitar o entendimento do sistema.

4.1.1 Tabelas do Banco de Dados

As tabelas do banco de dados (figura 6 - Tabela de Banco de Dados) mostram como está disposto o armazenamento de dados o qual foi utilizado para criar o diagrama de classes.

4.1.2 Diagrama de Classes

Como o conceito e as definições de objeto já foram previamente explicadas pode-se definir os objetos do sistema e suas interfaces. Dessa maneira, delimita-se o comportamento e as características para a estruturação das classes.

Os objetos são: login (pai), usuário (filho), contato, data tarefa (Pai), tarefa (filho), tipo contato (pai), contato(filho), endereço (filho), estado(filho), país (pai),calendário, tela inicial, tela contatos, nova conta, alterar/adicionar, conexão banco.

Para cada um desses objetos e interfaces foi proposta uma classe com métodos (comportamento) e características (atributos). As tabelas de banco de dados (figura xxx tabela 4, p.30) representam o conjunto de tabelas do MySQL sobre as quais as classes irão atuar, como estão dispostas e se comunicando.

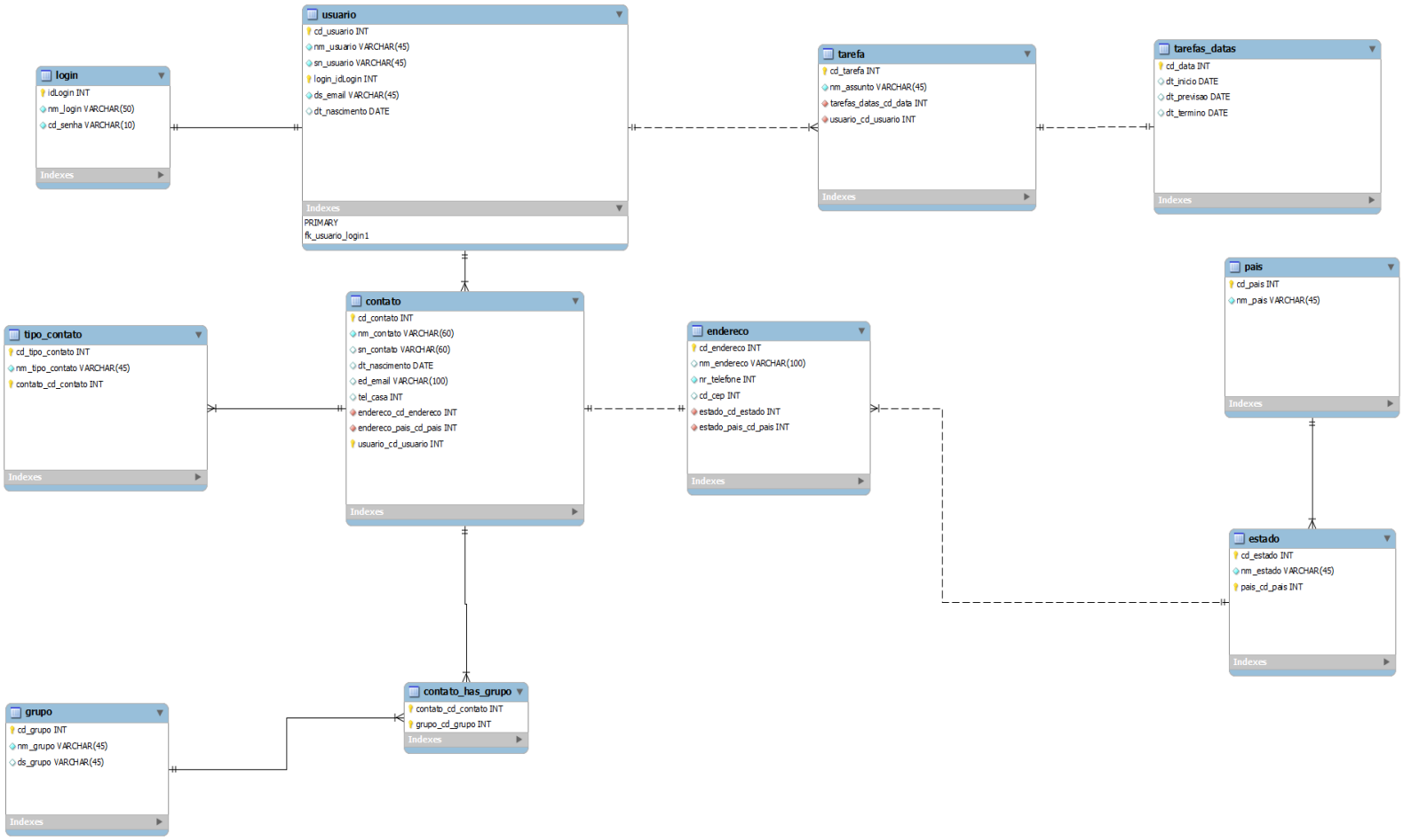
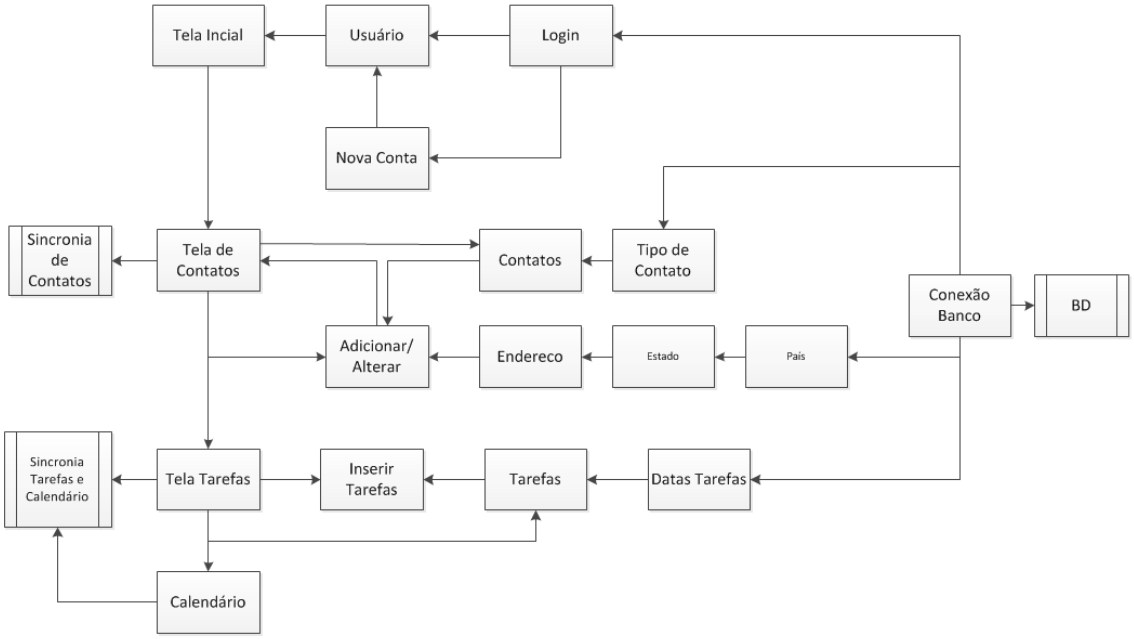


Figura 6a - Tabelas de Banco de Dados.

Diagrama de Classes



nente.
outro
ibuto,
classe


```
        return nm_login;    }
    public void setNome(String nome) {
        this.nm_login = nome;}
    public String getSenha() {
        return senha; }

    public void setSenha(String senha) {
        this.senha = senha; }
```

```
public int checarLogin(loginUsuario log) throws Exception
    {   Boolean resultadoParcialBusca ;
        Boolean naoPossuiLogin;
          String sql = "Select cd_senha , cd_usuario from usuario where nm_login = ? and cd_senha
= ?";
        PreparedStatement comandoSQL = conexao.prepareStatement(sql);
comandoSQL.setString(1, log.getUsuario());
        comandoSQL.setString(2, log.getSenha());
ResultSet rs = comandoSQL.executeQuery();
        bol = rs.first();
        if(resultadoParcialBusca==true){
        log.setCdUsuario(rs.getInt("cd_usuario"));
       return log.getCdUsuario();
}
            return naoPossuiLogin;
        }

    }
```

Código: Em negrito, primeiramente tem-se os métodos secundários gets & sets. Posteriormente, também em negrito, temos um método principal *checarLogin*.

Login

Esta classe contém dois atributos do tipo *String* .São estes: nm_login (*nickname)*e cd_senha (*password*) codigoLogin. Além disso, esta classe possui dois métodos principais *checarLogin* e *criptarSenha*. O primeiro diz respeito à verificação da conta através da validação do *nickname* no banco de dados. O segundo método corresponde aencriptação da senha fornecida para a comparação da senha registrada com o intuito de garantir a validação do usuário.

Usuário

Esta classe é filha da classe Login e, portanto, herda seus atributos e métodos. Além disso, existem mais cinco*private* atributos, são eles: *nome do usuário, sobrenome do usuário, campo de c-mail, data de nascimento, código do usuário (chave-primaria)*. Enquanto seu método principal é *cadastrarUsuário*. Este tem por definição criar uma conta nova caso o usuário não possua registro no sistema. Novamente a classe utiliza get & set para qualificar e quantificar os atributos. Como a herança da classe anterior totalizam-se 6 atributos e três métodos principais.

Tipo Contato

Esta classediz respeito à qualificação do tipo de informação. Cada contato só poderá ter um registro no sistema com o mesmo nome, sendo os tipos de opções do campo telefone: *residencial, celular, trabalho, fax*. Ou seja, caso eu tenha que inserir um contato chamado João Silva ele só pode ter um telefone residencial com esse nome. Caso tenha um celular será necessário inserir outra entrada com nome diferente. Os atributos são código do tipo, nome do tipo, código do contato. Só possui métodos secundários *get* e *set,* porém interagem com a interface Adicionar/Alterar como veremos mais a frente. É a classe pai da classe Contatos.

Classe Contato

Esta classe diz respeito a todos os atributos que o programa deve guardar de um contato (correspondem aos campos das tabelas do MySQL). São eles: nome do contato, sobrenome do contato, número do telefone, e-mail, cidade, estado, país, endereço, CEP, data e código do usuário. Possuem seus respectivos métodos secundários,*get* e *set*. Quanto aos seus métodos principais, são quatro: cadastrar contato, remover contato, editar contato, listar contato. Cadastrar contato representa inserir um novo contato na tabela contatos do banco de dados sob o código do usuário em questão. Remover contato significar apagar o contato da tabela contatos. Editar contato significa modificar um dos campos anteriores a fim de sempre

manter as tabelas atualizadas. Por fim, listar contatos importando todos os contatos da tabela para uma lista no aplicativo. Esta lista pode ser feita pelo código do usuário somente, ou pelo código do usuário e tipo de telefone. Novamente, esta classe herda os atributos da classe tipo contato podemos ainda acrescentar a informação do tipo contato e os métodos secundários.

Classe País

Esta classe diz respeito ao atributo referente à região. Possui os campos código do país e nome do país. Ela possui uma filha, a classe Estado. Além disso, tem os métodos secundários *get e set*.

Classe Estado

Esta classe diz respeito ao atributo referente à região. Possui os campos código do país, nome do estado e código do estado. Por ser filha da classe País ela terá seus estados pré-determinados de acordo com o código do País na tela Adicionar/Alterar contatos. Além disso, tem os métodos secundários *geteset*.

Classe Endereço

Filha da classe Estado, ela herda atributos e métodos das classes Estado e País. Além desses atributos, possui ainda: código de endereço, nome do endereço, número de CEP e nome da cidade em que reside. Possui os métodos secundário *geteset*, além dos métodos principais *cadastrarEndereco, editarEndereco, excluirEndereco e verificarCodigoContato.*

CadastrarEndereco significa armazenar novas informações nas tabelas do banco de dados sob o código do Contato. *ExcluirEndereco* representa apagar o registro do endereço do contato. *VerificarCodigoContato* é um método principal de validação, este busca verificar a existência do código do contato. Por fim, *editarEndereco* corresponde a fazer uma alteração em qualquer um dos campos mencionados. Uma ressalva deve ser feita comrelação a esta classe. Ela só pode ser executada caso o método *cadastrarContato* tenha sido realizado com êxito. Portanto o método *verificarCódigoContato* deve ser executado antes de *cadastrarEndereco*.

Classe DiaMesAno

Funciona como uma classe de apoio à qualificação das Datas em prol de executar busca com parâmetros corretos. O banco de dados possui o tipo pré-definido *date*. Este tipo trabalha com a ordem Ano/Mês/Dia ao invés da ordem Dia/Mês/Ano. Um fator positivo deste tipo de armazenamento é poder manusear os números, mais facilmente devido sua ordem

numérica ser crescente. Por exemplo, 20100317 é menor que 20100516. Em contrapartida, utilizando a convenção normal, tem-se 17032010 é maior que 16052010. Pelo modelo convencional a premissa é uma falácia, o que acarretaria erros nos parâmetros.

Para esta classe foram criados 3 atributos, são estes *dataParametroInferior, dataParametroSuperior*e*CódigoData*. Como método principal temos *transfIntEmDate e* comométodos secundários *gets* e*Sets*.

Classe Datas das Tarefas

Esta classe diz respeito às datas das tarefas referentes ao início, previsão de término e término.Possui quatro atributos dataInicio, dataPrevisao, dataTermino e codigoDatasTarefa. O método principal é*inserirDatasTarefas* responsável por alimentar o banco de dados com as datas. Os métodos secundários *gete set* também estão presentes. Esta classe tem uma relação bastante estreita com a classe DiaMesAno devido à manipulação de campos *date*.Além disso, esta classe será pai da classe Tarefas, passando seus atributos e métodos como herança para a classe Tarefas.

Classe Tarefas

Classe Tarefas é segunda classe mais importante do programa diz respeito àmanipulação das tarefas e como elas interagem com o usuário. Possui os seguintes atributos: *codigoTarefa, NomeTarefa, flagDeControle e variavelComparadora*. Como métodos principais possui: *inserirTarefa, excluirTarefa, editarTarefa e listarTarefas*. Além disso, herda os atributos e métodos da *ClasseDataTarefas* muito importantes para a completa funcionalidade do projeto. O método *inserirTarefa, excluirTarefa* e *editarTarefa* são métodos de acesso ao banco de dados, sempre alimentado o banco comas informações recentes ou até mesmo eliminando as informações antigas. O método *listarTarefas* é o único diferente, no sentido de realizar a função inversa, alimenta o usuário com as informações contidas no banco de dados. Este método será utilizado nas classes de interface do usuário como veremos mais adiante, mas consta no diagrama de classes (Figura 6, p. 31).

Classe Grupo

A classe Grupo é definida pela necessidade do usuário ter grupos de Tarefas ou Contatos diferentes. Ela herda atributos e métodos da classe Contatos e Tarefas. Além disso, possui três novos atributos: codigoGrupo, nomeGrupo e descricaoGrupo. Quanto aos seus métodos principais temos: *inserirGrupo*, *inserirContatoNoGrupo*, *inserirTarefaNoGrupo*, *excluirGrupo*, *excluirContatoNoGrupo*, *listarGrupos*, *listarContatosDoGrupo* e *listarTarefasDoGrupo*.

O método *inserirGrupo* diz respeito à criação de um novo grupo, nele éobrigatoriamente necessário o preenchimento do atributo nome do grupo. Além disso, é obrigatória a existência de pelo menos um contato. O método *inserirContatoNoGrupo*, primeiramente valida a existência do grupo e verifica se aquele contato já existe, posteriormente adiciona um novo contato. O método *inserirTarefaNoGrupo* segue o mesmo protocolo que inserirContatoNoGrupo, mas com o campo tarefa ao invés do contato.

O método *excluirGrupo* apaga o grupo indicado e todas as entradas de contatos que estiverem presentes. Já os métodos *excluirContatoNoGrupo e excluirTarefasNoGrupo* são responsáveis por excluir somente os itens indicados. Mesmo que o usuário apague todos os contatos e tarefas, o grupo ainda permanecerá existente.

Por fim, os métodos *listarGrupolistarContatosDoGrupo*e *listarTarefasDoGrupo* são os métodos que fornecem ao usuário as informações do armazenamento. Caso não haja informação, uma mensagem do sistema informa que não existem entradas no banco.

Classe Calendário

Esta classe funciona como suporte para as classe *Datas das Tarefas* e *Tarefas*. Ela se utiliza da classe *DiaMesAno* para melhor ordenação das datas. Possui três atributos dia, mês e ano.Seus métodos *Get* e *Set*. Os Métodos principais são *listarDiasDoMes* e *listarMesDoAno*. Pode ser invocada quando o usuário requerer.

Classe Conexão Banco

Esta classe é responsável por fazer a conexão entre a IDE do *NetBeans* e o *NavicatMySQL*. Está classe utiliza uma configuração pré-estabelecida pelo conector MySQL/JAVA. Como o estabelecimento da conexão foi explicado na sessão *conector MySQL/JAVA 3.1.1.1* (p.25), pode-se ater ao funcionamento das *queries* de busca disponibilizadaspela instância *PreparedStatment*, ou seja: *Insert*, *Delete*e*Select*.

Para efetuar buscas no banco de dados utiliza-se o método *createdStatment( )*, que cria uma instância do tipo *Statment* (vide Apêndice). Uma vez instanciada, pode-se fazer seleções de *queries* chamando *executeQuery*(*String*). Está *string* geralmente é o comando SQL. Os resultados, por sua vez são armazenados em um objeto *ResultSet*.

O *ResutlSet* é um objeto que mantém o cursor apontando para a linha da tabela referida. Inicialmente o cursor está apontado para a primeira linha da tabela. O método *next* é movido para a próxima linha, e retorna zero quando não existem mais linhas no obejto *ResultSet*, pode ser usado em um laço de repetição *while* (ORACLE COMPANY, 2003).

O código abaixo é um trecho da Classe Tarefa onde a técnica acima mencionada é empregada. Este recurso é muito utilizado para os métodos que necessitam listar nomes (campos em geral) do banco de dados.

```
String sql = "select nm_assunto from tarefa where tarefas_datas_cd_data = ?";
PreparedStatement comandoSQL = conexao.prepareStatement(sql);
comandoSQL.setInt(1,camposTabela.getCd_data());
ResultSet rs = comandoSQL.executeQuery();
if(rs.next()== false)
{
        camposTabela.setNm_assunto("");
    }
    else{
    camposTabela.setNm_assunto(rs.getString("nm_assunto"));
    }
    return camposTabela;
```

Fonte: Código em Anexo

Interfaces Gráficas

Aqui serão apresentadas as interfaces. Elas também são classes, porém com a função de agregar métodos de outras classes e estabelecer conexão visual com o usuário tornando o programa amigável e simples.

Ainda assim, as interfaces do sistema estão expostas (p.40, p.41, p.42) de maneira a tornar mais fácil o entendimento das especificações abaixo descritas.

Tela Inicial

Aclasse Tela Inicial (figura 7,p.40) utiliza as classes Usuário e Login, além de possuirdois eventos muito importantes. O primeiro deles é liberar o acesso para a próxima Tela Contato (figura 9, p.41),através do botão "*Entrar"*.Ao ser acionado, o botão roda um algoritmo equivalente ao *cookie* na Internet,se fazendo utilizar do método público *guardarCódigoUsuário*. Assim, evita-se a necessidade de autenticação do usuárioa cada nova tela. O segundo evento diz respeito à necessidade de um novo usuário requerer uma conta. Para isso cria-se uma conexão com a *Tela Nova Conta* através do atalho *Desejo me Cadastrar* (figura 7 –Tela Inicial , p40).

Tela Nova Conta

A classe Tela Nova Conta (figura 8, p.40) como descrito acima, é a classe responsável por receber os dados do usuário e enviar um registro de abertura de conta ao administrador do sistema (*root*) para uma possível validação. O administrador deve decidir se aceita ou não o

usuário, para incluí-lo no banco de dados. Este evento é acionado pelo clique do botão "*Cadastrar*". Caso o pedido seja aceito, o administrador enviará via correio eletrônico o informativo de liberação da conta.

Tela Contato

A Tela Contato (figura 9, p.41) é a classe responsável pela interação entre as classes contato e tipo contato. Utilizando o método listar contatos elamostra ao usuário seus contatos existentes.

É composta por cinco eventos de associação com *action* sendo o*click* do *mouse* nos botões Adicionar, Alterar, Tarefas, Sincronizar e Criar. Os botões Adicionar, Alterar e Tarefas são chamadas para suas respectivas interfaces Tela Adicionar (p.41), Tela Alterar (p.41), Tela Tarefas (p.42). Os outros dos botões são eventos da própria classe. O botão Criar Grupo gera grupos diferentes, caso o cliente deseje classificar diferentes pessoas. Um exemplo disso é criar um grupo de pessoas do trabalho e outro grupo de familiares a fim de não misturar telefones e tarefas de ambientes distintos. Por fim, o último evento é o de *Sincronizar*. Este evento será explicado mais adiante na sessão "Iniciar Sincronia".

Existem ainda 26 eventos secundários com todas as letras do abecedário na Tela Contato. Cada botão lista todos os contatos da biblioteca que iniciam pela letra do botão em questão.

Tela Adicionar Contato

A Tela Adicionar (p.41) contém todos os campos que fazem referência aos atributos das classes Contatos, Tipo Contato, Endereço, País, Estados e Grupo. Após as informações serem preenchidas, utiliza-se o botão Adicionar que será responsável pela inserção dos dados no banco. Como se pode notar só existem dois eventos, o primeiro foi descrito acima e o segundo é a conexão com a tela anterior (Tela Contatos).

A tela Adicionar não foi exposta, uma vez que também se faz uso da classe Tela Adicionar. Todavia elas diferem em seu conteúdo. Na tela adicionar, o usuário informa os dados a serem inseridos. Já na Tela Alterar, um contato, previamente selecionado, tem suas informações expostas na interface. Dessa maneira, o cliente poderá atualizar os novos dados.

Tela Nova Tarefa

É uma interface (figura 12, p.42) simples que utiliza a classe Nova Tarefa como sua principal classe. Ela é proviniente de uma chamada da classe Tela Tarefa e existe com o intuíto do usuário de inserir novas informações no sistema. Possui três campos simples epré-

configurados em um recurso chamado *choice* e a ação *scroll-lock,* com as opções de dias e meses do calendário. A única digitalização necessária remete ao assunto da tarefa a ser inserida. Após todas as informações serem postas de maneira correta, o botão Atualizar iniciará a inserção do novo registro no sistema.

Tela Tarefa

Talvez esta seja a interface mais complexa do sistema. A tela Tarefa (figura 11, p.42)é composta por vários botões acionados pelos cliques, e se inter-relaciona com as classes *Tarefas*, *DatasTarefas* e*DiaMesAno*. Os botões são *Atualizar, Excluir, Nova Tarefa, Calendário*e *Importar.*

O botão *Importar* trabalha com o recurso *choice* e é acionado pelo *scroll-lock* de maneira a filtrar o determinado mês, utilizando atributos da classe *DiaMesAno* para realizar a busca. O resultado é exposto em dois painéis, um referindo-se aos nomes das atividadese o outro especificando as datas.

O botão *Excluir* funciona pré-selecionando um dos itens dos painéis. Após a confirmação da ação, o registro é excluído do banco.

O botão *Atualizar* permite fazer alterações nas tarefas já existentes. Similar ao botão *Alterar*, faz uso da mesma classe *Tela Nova Tarefa* para realizar modificações no conteúdo antes escrito.

O botão Calendário expõe os meses dos anos pré-selecionados pelo recurso *choice*.

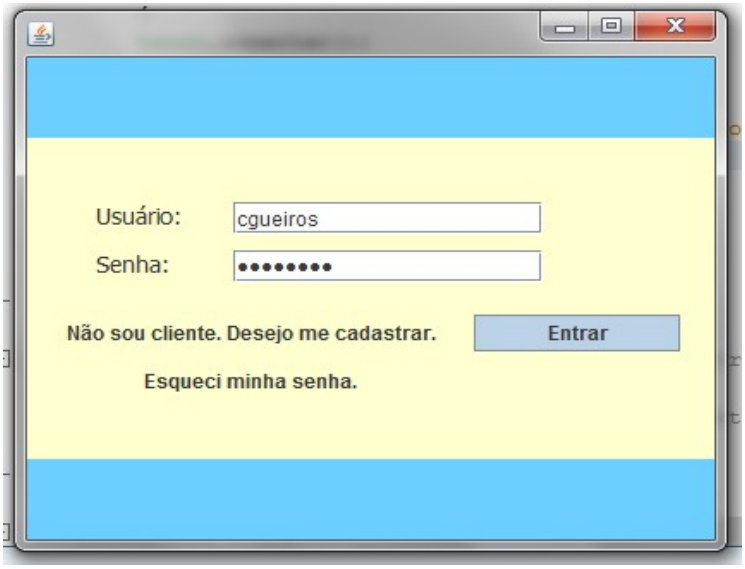


Figura 7 - Tela inicial.

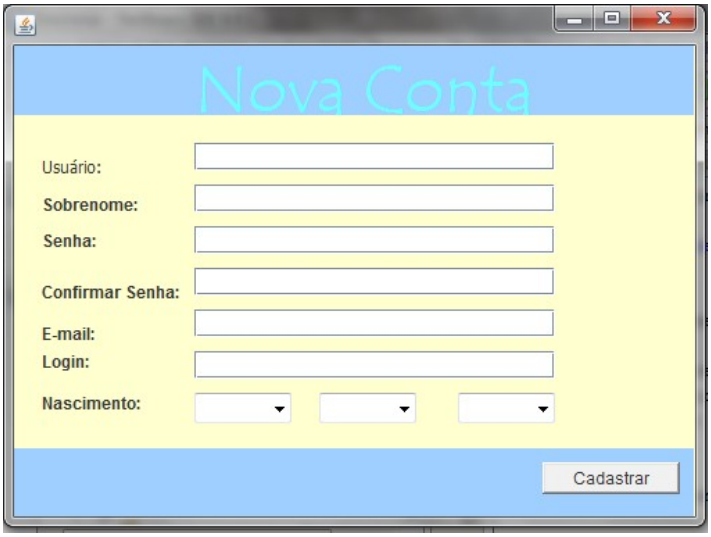


Figura 8 - Tela Nova Conta.

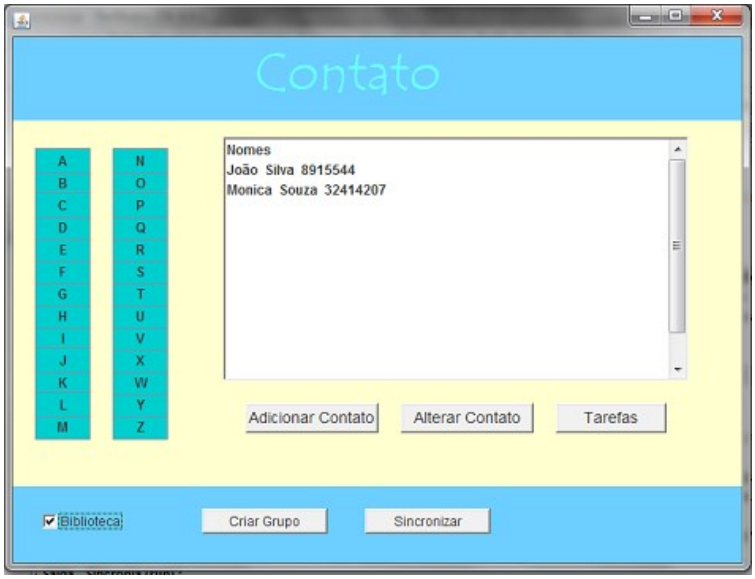


Figura 9 - Tela Contato.

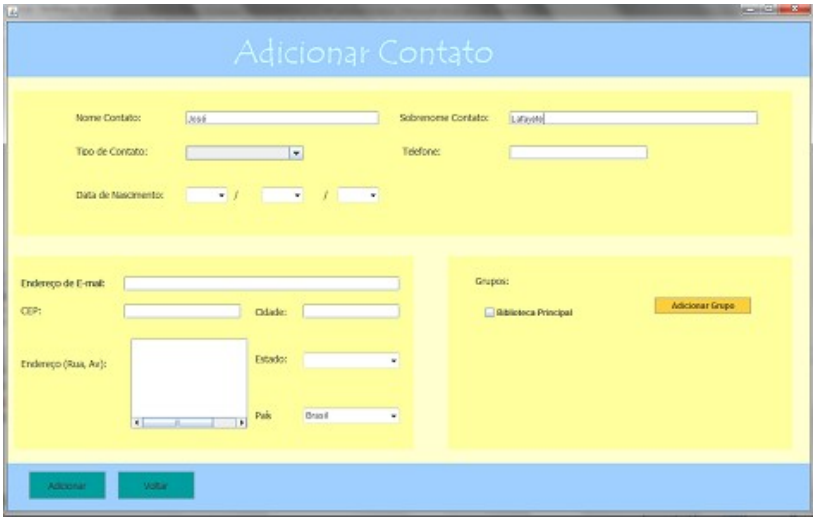


Figura 10 - Tela Adicionar.

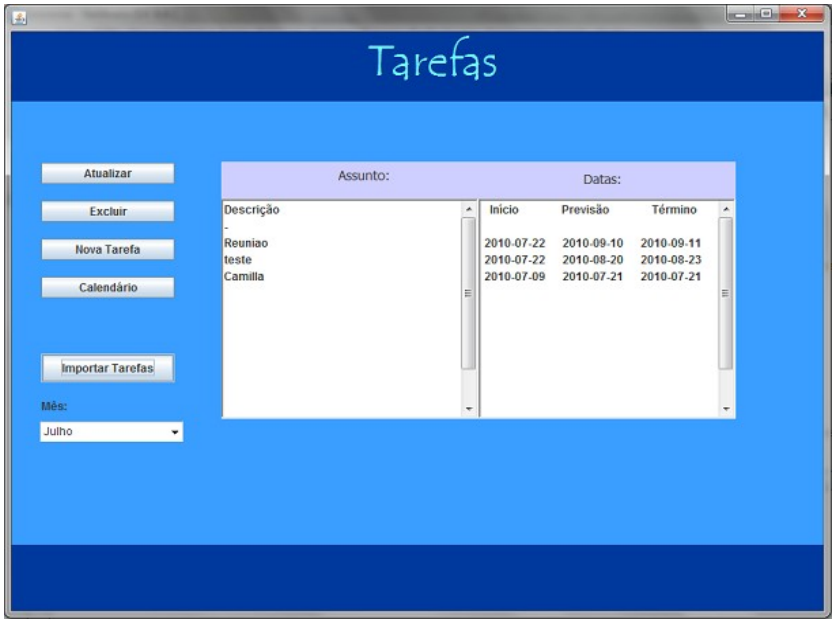


Figura 11 - Tela Tarefa.

Figura 12 - Tela Nova Tarefa.

4.2 SERVIDOR PHP

O servidor do projeto é um de código aberto e livre desenvolvido por Nicolas Bouges.Este servidor está disponível em seu website particular (BOUGUES, 2004).

Como o projeto está feito e funcionando, suas especificações não condizem com as especificações anteriores. As tabelas do servidor são diferentes das tabelas da sessão anterior, uma vez que segue o padrão pré-estabelecido pelo protocolo. Para fazer a conexão entre os dois programas foram criados classes de manipulação de dados melhor detalhadas a seguir (seção 4.2.2,Descrição das Classes).

Além dessas classes de manipulação criadas, serão definidas as classes utilizadas e desenvolvidas por Nicolas Bouges para melhor aprendizado do sistema.Pequenas modificações foram realizadas para atender ao cliente Funambol Blackberry. Além disso, possíveis divergências de versões serão destacadas mais a frente.

4.2.1 Tabelas de Armazenamento

No Capítulo 2 vimos especificações técnicas relacionadas como mapeamento entre GUID e LUID.Estas especificações devem ser levadas em conta para a criação das tabelas de armazenamento e para o funcionamento do projeto. No total existem sete tabelas: *users, vcalendars,vcalendars_maps, vcalendars_sync, vcards, vcards_maps, vcards_syncs, vnotes, vnotes_card e, vnotes_syncs*.

A tabela *users*é correspondente à tabela Login. Fazendo as devidas mudanças, não há necessidade de mantê-la. Entretanto, as restantes são de fundamental importância.

As tabelas *vcards*, *vnotes*e *vcalendars* possuem dois campos principais:a chave primária e o campo da informação. Para a tabela *vcards*, os campos incluem:*id_vcards* e *vcard* (contato/txt). Para a tabela *vcalendars* tem-se:*id_vcalendars*(data/txt). Por fim, a tabela *vnotes* inclue os campos: *id_vnote* e *vnotes*(tarefa/txt). Os demais campos são campos de informação de atualização e dados de criação.

As tabelas *vcards_maps, vcalendars_maps e vnotes_maps* representam respectivamente o mapeamento das tabelas principais destacadas acima. Dessa maneira, *device_int* representa o GUID, o *remote_id* representa o LUID e por fim tem-se o código da respectiva função para a identificação da informação (chave primária da tabela anterior).

Para as âncoras de sincronizaçãoas tabelas são separadas. Elas são: *vcalendars_sync, vcards_syncs* e *vnotes_syncs*. Cada uma das tabelas fica remetente ao código do dispositivo (GUID)visando correlacionar as âncoras*remote_last_ancor* e *local_last_anchor.*

4.2.2 Descrição das Classes

Em uma tentativa de facilitar o entendimento e a explicação das classes, as mesmas foram divididas em classes funcionais e classes auxiliares. Classes funcionais são responsáveis pelas funções de sincronização, apagar e atualizar. Já as classes auxiliares são pequenas engrenagens para lidar com objetos XML e autenticações do usuário.

Classe Xml_tree_node_c: possui três atributos: *name*, *children* e *attribs*. Seus métodos são: *get_child, &get_data, &XML_tree_node_c, &add_child, &new_child, new_child_data, get_name, print_node*.

Esta classe tem por fim montar o documento XML. Seus métodos são instanciados com valor NULL. Os mais utilizados no código são *&new_child,* quando há a necessidade de se criar um filho na árvore, e *new_child_data*, quando é necessário instanciar um novo objeto para guardar informação. Toda vez que uma nova árvore XML é requisitada, o comando *&XML_tree_node_c* deve ser implementado.

Class Xml_tree_text_node_c: possui um atributo *data*. Seus métodos são &XML_text_node_c, get_data, print_node e text_to_xml. Está classe tem como principal função transformar os objetos texto em XML. Além disso, o *get_data* é bastante utilizado para buscar uma informação armazenada no sistema depois de transformada em texto.

Classe Auth: possui quatro atributos globais *db_server, db_username, dB_password* e *dB_name.* Sua única função é verificar a existência do nome no servidor. Caso falso, o usuário não pode entrar no sistema.

Classe Generic_mysql: uma das principais classes do sistema e também uma das mais complexas. Possui apenas um atributo *data*, e vários métodos.

O primeiro delesé o *dbconnect*. Este faz a validação do usuário e da senha para guardar os valores das variáveis globais definidas na Classe Auth.

Outro método,*start_sync* deve retornar um par específico de usuário e LUID do dispositivo.Além disso, retorna também a âncora *Last* do servidor e do usuário (verificação para um possível comprometimento do banco caso haja falha).

O método *get_anchors* é responsável por devolver *o local_next_anchor (âncora next* do servidor) e o *remote_next_anchor* (âncora *Next* do dispositivo).

O *sync_add* tenta adicionar um novo item ao banco. Primeiramente, ele verifica a existência do dado (418 - item jáexistente). Caso não exista, o valor é inserido (201- OK).

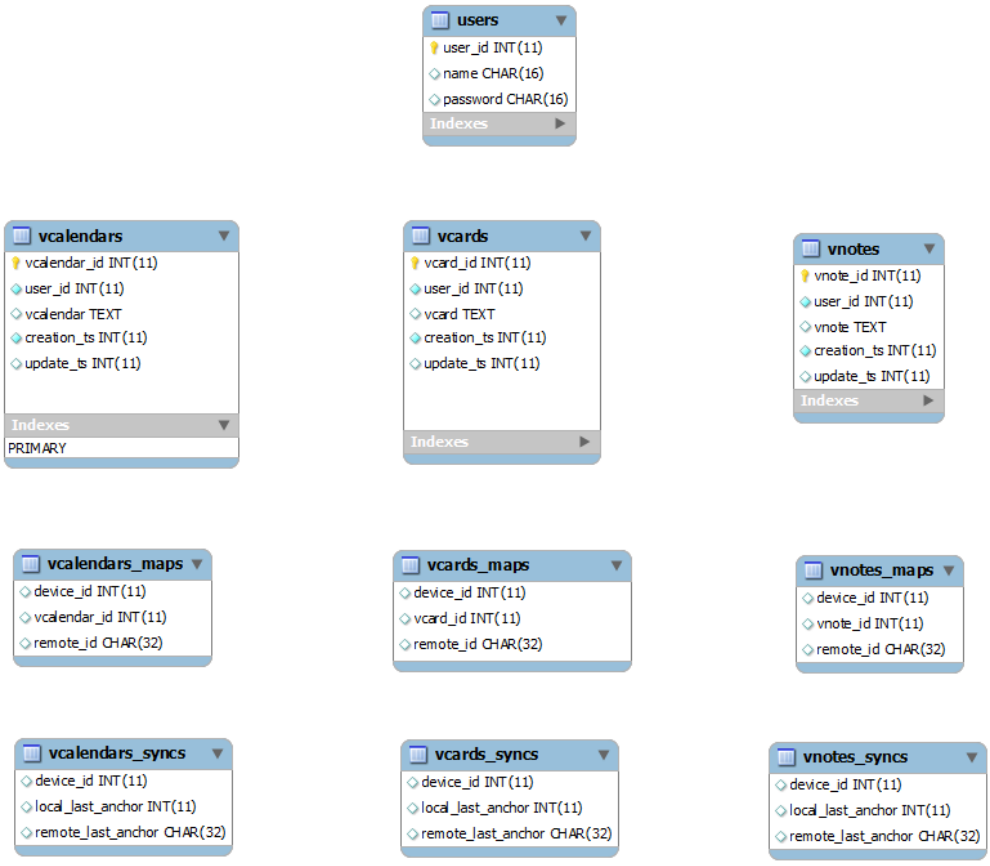


Figura 13 – Tabelas de armazenamento servidor Nicolas Bougues.
Fonte:BOUGUES, Nicolas. 2004

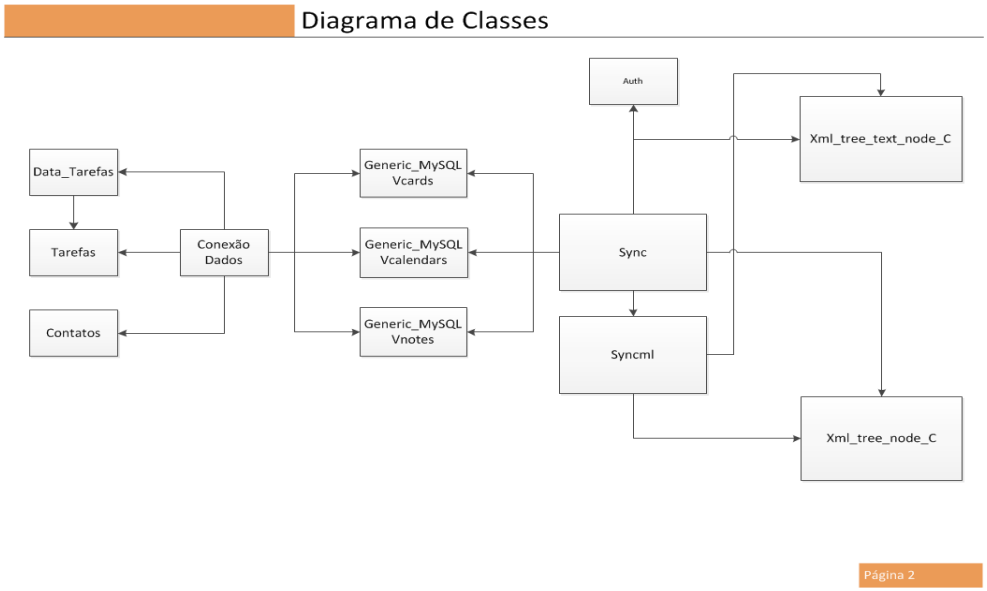


Figura 13.1 - diagrama de Classes.

Em contrapartida ao *add_sync* temos o *delete_sync*. Este método tem como principal meta apagar uma entrada no banco. Primeiramente, é executada uma busca à procura do item desejado. Caso o resultado da busca seja positivo, a informação, (*vcard, vnote, ou vcalendar*) e seu respectivo GUID serão apagados. O autor faz uma ressalva que somente o GUID da informação será apagado, os outros GUID permanecerão caso futuramente queiram ser apagados.

O *Server_delete* serve como um método "limpador". Ele apaga todas as entradas no mapa de correlação que não possuam entradas correspondentes de informação. Por fim, o *end_sync* deve ser invocado quando o processo de sincronização está sendo finalizado, pois ele salva as âncoras de sincronização para a próxima sessão.

Classe *VCards*, Classe *VCalendar* e **Classe *Vnotes***:todas estas três classessão herdeiras da classe generic_mysql. Além disso, todas possuem um atributo *data* já instanciado com o valor de *"vcard", "vnote"*ou *"vcalendar"*. As classes possuem uma lista dos *ctCaps* (informações da capacidade do dispositivo), além dos métodos *get_datastore_caps* e get_ctcaps. Estes métodos não são obrigatórios para o funcionamento do sistema. Funcionam listando tudo que existe dentro do banco do cliente. Em versões mais recentes não são expostos.

Classe Syncml: está classe é uma das fundamentais do projeto. Ela possui todos os comandos para a construção da resposta XML.Possui os seguintes atributos: *_data_store, _rcvd_mdg_id, next_cmd_id, _syncml_version, authenticated, username, session_id, _target_uri*e *_source_uri.* Quanto aos métodos, temos: *handle_next_header, make_response_header, handle_comand, handle_command_put, handle_comand_alert*e *handle_comand_sync*. Como todos os métodos são utilizados maciçamente, é fundamental definir-se um a um.

Handle_next_header é responsável por formar o *SyncHeader*(cabeçalho) da resposta ao Cliente. É composto por uma estrutura com:VerDTD (*Version of Device Information*),*VerProto* (*Version Protocol*), *SessionID* (identificador de sessão – fornecido pelo servidor, função do PHP), *MsgID* (Message Identifier), *Target* (LocURI/LOCNAME) e Source (LocURI).

Make_header_response é o primeiro comando a ser executado no *SyncBody* (corpo) da resposta. Todas as estruturas do *SyncBody* possuem os mesmos campos. Ocorre somente uma mudança quando se trata do campo <Cmd> (Comando) devido às diferentes funções que este pode assumir (PUT, GET, ALERT, SYNC), neste caso *SyncHeader* (ver código abaixo) :

```
<SyncBody>
<Status>
<CmdID>1</CmdID>
<MsgRef>1</MsgRef>
<CmdRef>0</CmdRef>
<Cmd>SyncHdr</Cmd>
…
<Data>212</Data>
</Status>
```

A nova estrutura do “Cmd” terá filhos para indicar a referência da origem (celular), e a referência do destino (servidor) funcionando como um *acknolegde*.

Handle_command é ummétodo que funciona como um *switch* de programação, alternado as chamadas dos métodos a seguir de acordo com a ordem do *post*(PUT, GET, ALERT, SYNC, SYNCHEADER).

Handle_command_Put, realiza duas funções. Na primeira, ele armazena a informação obtida pela requisição do cliente em suas variáveis locais. Na segunda, ele utiliza a estrutura do código acima (referência) para preencher os campos com esta informação. O campo *Command* éalterado para PUT, tendo na subrede o campo DevInf (*device Information*) correspondente à versão do dispositivo (necessáriopara averiguar compatibilidade entre o protocolo SYNCML do servidor com cliente).

Handle_command_ Alert, este método escreve duas estruturas diferentes na resposta do servidor. O princípio de funcionamento é o mesmo descrito pelo método *handle_comand_put*. Ele armazena as informações e posteriormente posta na sub-árvore.

Novamente a estrutura do código acima é empregada. Entretanto, no campo *Command* ocorre uma mudança para *Alert*. Neste caso, a sub-estrutura é um pouco maior. Os campos designados para origem e destino (*SourceRef, TargetRef*) são novamente incorporados com os dados recebidos. Todavia, após a validação da informação recebida, o servidor informa a âncora *Next* recebida.

A segunda parte é a parte ativa. Após ter-se seguido as regras ditas acima, o servidor deve informar suas âncoras (vide seção 2.3.1). Para isso ele cria a estrutura (ALERT) com suas âncoras *Next* e *Last* e os novos endereços de origem e destino. Todas essas estruturas são realizadas com as classes auxiliares, principalmente a classe XML.

Handle_comand_Get é um método que realiza duas funções como descritas acima, armazenando informações e postando-as mais tarde na estrutura do código.

Exceção do campo *Command* que desta vez terá a informação GET. Este novo dado informa ao cliente o *devInf* do servidor.

Handle_command_sync é um método que é utilizado para o cliente informar qual o tipo de sincronia exigida (ver Tabela). Por definição, o cliente utiliza a *Two Way Sync* como sincronização padrão. Entretanto, qualquer possível comprometimento na conexão gera uma quebra de confiabilidade fazendo com que o procedimento seja reiniciado desde o princípio. Desta vez, a opção de sincronização obrigatoriamente é a *Slow Sync*.

Classe Sync

Esta é a classe principaldo projeto. Énela que todas as outras classes interagem como podemos ver no diagrama de classes 2 (figura13, p.46).Há uma programação à parte feita por Nicolas Bouges que não diz respeito ao protocolo SyncML, mas sim àdisponibilidade de utilização da função *session* no PHP. Para que não haja necessidade de haver autenticação humana a cada intervalo de tempo não respondido, principalmente caso ocorra algum erro de transmissão, o PHP permite que o usuário se autentique apenas uma vez. Esta autenticação, entretanto, não se refere à autenticação através das tabelas do banco de dados, nem a autenticação referente ao protocolo.

Apesar do sync.php ser o principal arquivo, ele é curto e simples. Único ponto a ser ressaltado é que, esteprojeto só não aceita nenhum tipo de *Content type txt/plain*, mas somente *application/vnd.syncml+xml.*

```
<php
If (@$_SERVER["CONTENT_TYPE"] != "application/vnd.syncml+xml")
{ ?>
<html>I only speak SyncML.</html>
<?php
  exit();
}
```

Fonte: Código retirado do arquivo sync.php Nicolas Bouges – www.nicolasbouges.net

Classe Conexão Dados

Para fazer a ligação entre as informações SyncML e as informações Java é necessário existir uma classe de conexão dos dados. Isso porque as informações do banco de dados do celular estão na forma de *vcards (text)*, *vcalendars(text)* e*vnotes (text)* (vide Apêndice). Entretanto as tabelas do usuário possuem as informações divididas em campos e tipos diferentes.

Outra característica marcante é que os códigos das informações não são os mesmos entre o banco Java e banco SyncML, mesmo problema evidenciado pelo protocolo SyncML.

Para resolver esse problema foi utilizada uma tabela de conexão semelhante à tabela de mapeamento (vide seção 2.3.2).

```
BEGIN:VCARD
VERSION:2.1
REV:20061203T164021Z
N:Donovan;Luc;;Mr;
FN:Luc Donovan
BDAY:19820714
EMAIL;INTERNET;HOME:luc@yahoo.fr
EMAIL;INTERNET;WORK:luc@work.com
ADR;HOME:;;Rue dom;Ville dom;;Cp dom;Pays dom
TEL;VOICE;HOME:3304658877
TEL;CELL;HOME:0604658877
TEL;CELL;WORK:0125789963
ORG:Entreprise;
TITLE:Emploi
URL;HOME:http://www.google.fr
NOTE:Test vCard file
PHOTO;TYPE=PNG;ENCODING=BASE64:
    iVBORw0KGgoAAAANSUhEUgAAAHgAAAA46aAAABxVBMVEX///8AAAAd
   CAgQEBAaGhotFg8xFxIhISEvIRhLHQAnJyczJiQ2Jx04KStiIwB1IABQKg8zMzN5
   CAgQEBAaGhotFg8xFxIhISEvIRhLHQAnJyczJiQ2Jx04KStiIwB1IABQKg8zMzN5
   BeXkBkZBXSSQBqahpmZmbjUQBtbW12dg3vVgB0dER1dTV3dyt0dFRycnJ7eyt5
   eXV6emWBgRx7e3v2ZQfUbin/ZgDxahKCgnqMjBGMjCGEhITscyD/cQCLi3iUlA6
   lBaMjIyMjJ+NjaGOkJKamhCbmwnShVSvjnehoQeZmZmcnHiqmY2Yn6Senqelpa
   paWqqnisrI3Jo4mirbWtra2zs3mxsYy5uXC2tp+1tbW2tsi9vXu5ucywvcXCwnvKdsf
   u7G8vMy4v8O9vb3MzAC+vsLAwLjMzDO4w8m5xMvIyHXMzGbKw7/FxcXFxdS
   0lfEztbMzMzMzN3a2kTU0M3G1N3J09ji4gTL09nS0uPW1tbl5TLn5yPY2OvR3OP
    z8kNYTyzBNGHBrtRFlXKGLqDksWErdpVj3quisfSQYhFLILYsbbRud2/1zlJ00lad
   up2qmW8+n/y+fre3Nzfnd87vnPuoIGTIkCFDhv8l/P2h7Sm19M/NP3/9fv3kmxHas
Xv95c69oA0NZwq9h6Dudlrny4P2H773voceefHZtbe3o0acef+BwdX1PeBuq1Tbcf
END:VCARD
```

Fonte: Documentação da API (Memotoo-Documentação da API para programadores 2001-2011http://www.memotoo.com/index.php?rub=api#contactsVCF.php

Métodos:

InserirContatoNoSyncML:toda vez que um novo contato é inserido na tabela do servidor (novo GUID) a suaentrada correspondente deverá ser automaticamente criada na tabela de conexão e posteriormente na tabela do Java. Como o cd_contato da tabela Contato é auto incrementado e nunca pode se repetir, esta chave será herdada na tabela do servidor.

InserirContatoNoJava:toda vez que um novo contato é inserido na tabela do Java a sua entrada correspondente deverá ser automaticamente criada na tabela de conexão. Para a tabela do SyncML o sistema também é auto incrementado. O restante das informações serão populadascom métodos que ainda serão explicados.

VerificandoInformacao: este método compara as informações por *byte* verificando se há duplicação. Pode ser executado em qualquer tabela.

VerificandoCampos: este método compara os campos da tabela conexão. Procura por duplicidade nos códigos de informação, sejam eles GUID ou cd_contato. Caso haja duplicidade a entrada mais recente será apagada.

Text2Int:este método transforma o texto em inteiros de acordo com as especificações do projeto.

Text2Byte: este método utiliza o texto em forma de byte para se encaixar nas especificações do sistema.

Text2Field: este método separa o texto em campos individuais. Caso não haja informações o campo será preenchido com NULL ou zero.

Sincronização: com este método o usuário força a verificação de dados entre as tabelas. Caso haja discrepância ele faz a atualização das partes de maneira a permanecer a mesma informação.

Capítulo 5 Aplicação

Este capítulo é referente à aplicação do software cliente e do *SyncClient*. Neleé exposta a operação do sistema e uma possível inserção de conteúdo assim como a sua edição. Como os métodos e as classes já foram devidamente explicados no capítulo anterior, este capítulo irá tratar apenas da funcionalidade de maneira macro. Com isso, não haverá alusão à nomenclatura dos mecanismos, evitando digressões à micro subsistemas e com isso a perdado objetivo central.

5.1 Autenticação Inicial

Toda vez que um usuário quiser fazer qualquer tipo de conexão com obanco de dados, ele terá que ser autenticado. Nas especificações do sistema, foram retratados vários recursos de autenticação interna. Porém, o usuário não deve estar ciente das quantidades de validações feitas pelo sistema internamente. Ele faz uma autenticação manual uma única vez no cliente (*software* Java e celular) e permanecerá"logado" no sistema.

No software Sincronia esta autenticação é feita na página inicial do sistema (ver figura 7, pg 40). Após a autenticação do usuário, não haverá necessidadede outras autenticações.

Para o *SyncClient* o procedimento é o mesmo, mas, como não existe nenhum tipo conexão interna de controle com o programa de sincronização desenvolvido neste projeto (Sincronia), será necessário uma autenticação nova. A grande vantagem do sistema proposto pela Funambol é que, uma vez escrito o usuário e senha, ele é automaticamente gravado.

5.2 Iniciar Software

Uma vez dentro do sistema (Sincronia) inicia-se a navegação.O sistema é alimentado com todas as *flags* de controle do usuário. Funciona como o sistema "*session*" aplicado na *web*.

A tela de contatos não possui nenhum tipo de informação inicial. Caso o usuário queira carregar a biblioteca principal, esta opção deverá ser acionada (ver figura 9). Uma vez assinalada, o programa carregaráe listará todos os contatos existentes.

O mesmo sistema ocorre na tela Tarefas. Inicialmente esta tela também não conterá dados. Por isso deve-se escolher um mês e carregar as opções.

Existe em particular um botão de Sincronização. Este botão é responsável por atualizar todas as informações ente os diversos bancos do sistema, de maneira a equalizar toda a informação.

5.3 Inserir Conteúdo

Caso o usuário deseje adicionar um novo contato, é necessário ir a tela Contato. Os campos “Nome do Contato” e “Telefone” são obrigatórios, todos os outros são opcionais. O sistema indicará condição deerro, caso estes campos não sejam preenchidos. Depois de inserido, volta-se à tela principal. É necessário atualizar a biblioteca para recarregar os dados (novos) na tela. Para isso basta clicar no botão “Biblioteca Central”.

Para editar um contato antigo, o usuário deverá marcar o nome desejado e depois clicar no botão Alterar. A tela Alterar Contato irá conter todos os dados existentes do contato. O usuário pode, assim, alterar a informação desejada, ou inserir um novo dado.

Caso o usuário deseje inserir uma nova tarefa (ver figura 11) ele deveráutilizar a tela Tarefas. Apóso *upload* de informações, pode-se verificaro que há no sistema.

Para inserir uma nova tarefa clica-se no botão Nova Tarefa. Este remeterá a uma nova interface,tela Nova Tarefa (ver figura 12).

Na tela Nova Tarefa, o usuário deverá informar três datas, ou seja, início, previsão e término da tarefa. Além disso, deverá fornecer o nome. Voltando àtela tarefa, bastará apertar Atualizar para ver a nova tarefa presente no sistema. É sempre importante lembrar que o sistema deverá ser atualizado através do botão *Sincronização* na tela Contatos. Toda vez que o aplicativo for encerrado, os bancos serão automaticamente atualizados.

5.4Iniciando o SyncClient

O cliente precisa estar“*logado”* em uma conta junto ao servidor. O *SyncClient* Funambol pede para informar três dados: usuário, *password*, URL do servidor. Para o projeto Sincronia estes dados são: usuário igual à“sincronia”, senha igual à“sincronia” e a URL igual a “sincronia.no-ip.org:8080.syncml/sync.php”. A figura abaixo mostra o *SyncClient* já com os campos preenchidos.

Figura 14 – SyncClient.
Fonte: RESEARCH IN MOTION BLACKBERRY, 2010.

5.5 Iniciar Sincronização

Uma vez “*logado*” no sistema, pode-se escolher qual opção deseja ser sincronizada. Como especificado, temos os campos contatos, calendário, e tarefas. O *SyncClient* da Funambol ainda permite a inserção de foto, assim como *vcard*. Na figura 19, tem-se a tela do Blackberry 8520 após a autenticação do usuário. Nesta tela é possível iniciar uma sincronização dos contatos com o servidor.

Figura 15- Tela de sincronização.
Fonte: RESEARCH IN MOTION BLACKBERRY, 2010.

Capítulo 6 – Resultados e Avaliação

Este capítulo se prepõe a mostrar a comunicação XML entre o *SyncCliente* o *SyncServer* e os resultados desta transmissão. Foram utilizados três servidores diferentes para análise do protocolo através do analisador de pacotes (*Wireshark*), sendo realizada transferência de *vcards, vnotes, vcalendars*.

O protocolo utiliza os preceitos antes explicados e definidos pela OMA (próxima seção). A partir da estrutura inicial, cada servidor é individualizado e tema sua comunicação apresentada.

6.2 Estruturação do Código XML

O código XML, nada mais é do que um envio de dados do dispositivo ativo, confirmandodados (autenticação, versão do protocolo) etendo no final uma nova forma de informação destinada ao componente passivo. Este código é dividido em duas partes: *SyncHeader* (cabeçalho) e*SyncBody* (corpo).

Neste projeto, o sincronização sempre será iniciado pelo *SyncClient*(BlackBerry Curve 8520 - Funambol). Este emissor envia um *post*(Apêndice) ao servidorcontendo as suas estruturas.

O *SyncHeader* do *post* inicial deve conter a versão do dispositivo, a versão do protocolo, o *SessionID*, além dos endereçosde origem e de destino. O cabeçalho funciona para o servidor identificar o remetente e testar a compatibilidade entre as versões. Muitas vezes ainda há a necessidade de fazer a autenticação do usuário criptografada, fazendo-se o uso do campo de Cred (Credentials). Nesse caso, podem-se utilizar vários métodos MD5, BASE 64, entre outros. No projeto foi utilizada a criptação base 64.

```
<Cred>
<Meta >
     <Type xmlns="syncml:metinf">syncml:auth-basic</ Type>
     <Format xmlns="syncml:metinf">b64</Format>
</Meta>
<Data>c2luY3JvbmlhOnNpbmNyb25pYQ==</Data>
</Cred>
```

Código em Anexo – Apêndice.

O *SyncBody*, por sua vez é onde se escreve a comunicação propriamente. Ele é responsável por iniciar a sincronização. Após a estruturação do *SyncHeader,* o cliente deverá preparar o envio de suas âncoras de sincronização*Last* e *Next*, assim como os endereços de origem e de destino das âncoras (todos englobados dentro da estrutura ALERT).

```
<Alert>
              <CmdID>1</CmdID>
              <Data>200</Data>
          <Item>
              <Target>
                   <LocURI>configuration</LocURI>.
              </Target>
          <Source><LocURI>config</LocURI></Source>
          <Meta>
          <Anchor xmlns="syncml:metinf">
              <Last>1288497949375 </Last>
              <Next>1289048219062</Next>
          </Anchor>
          </Meta>
          </Item>
          </Alert>
```

Outro comando que compõe o *SyncBody* é o *put*. Apesar de ser mais extenso, estecomando trás informações menos importantes. Geralmente informações do fabricante e do dispositivo. A estrutura é composta por vários campos, sendo que o campo *Item* é o pai da nova sub-árvore construída para informar ao servidor sobre os requisitos do cliente. O campo *SyncCap e SyncType* contém as possibilidades que o software cliente suporta para a sincronização. No caso do *SyncClient* abaixo temos 1, 2 e 7 correspondendo a *Two-Way Sync, Slow Sync*e *Server Alerted Sync*.

```
<Put>
          <Cmd ID>2</CmdID>
          <Meta>
          <Type xmlns='syncml:metinf'> application/vnd.syncml-
devinf+xml</Type>
          </Meta>
          <Item>
          <Source><LocURI>./devinf 12</LocURI></Source>
          <Data>
              <DevInfxmlns='syncml:devinf'>
              <VerD TD>1.2</VerDTD>
              <Man>Research In  Motion</Man>
              <Mod>8520</Mod>
```

```
                <OEM></OEM>
                <FwV>< /FwV>
                <SwV>8.7.1 </SwV>
                <HwV>2.13.0.140</HwV>
                <DevID>fbb-553648138</DevID>
                <DevTyp>phone</DevTyp>
                <UTC/>
                <SupportNumberOfChanges/>
          <DataStore><SourceRef>config</SourceRef>
          <Rx-Pref>
                <CTType>application/*</CTType>
                <VerCT></VerCT>
          </Rx-Pref>
          <Tx- Pref>
                <CTType>application/*</CTType>
                <VerCT></VerCT>
          </Tx-Pref>
          <SyncCap>
                <SyncType>1</SyncType>
                <SyncType>2</SyncType>
                <SyncType>7</SyncType>
          </SyncCap>
          </Data Store>
          </DevInf>
          </Data>
          </Item>
</Put>
```

O Comando *Get* é responsável por enviar o *Content-Type* para o servidor. Ele indica o tipo de conteúdo da mensagem. No servidor Nicolas Bougues se essa informação vier corrompida o servidor não será acionado.

```
<Get>
          <CmdID>3</CmdID>
          <Meta><Type
          xmlns='syncml:metinf'>application/vnd.syncml-
          devinf+xml</Type></Meta>
          <Item>
                <Target>< LocURI>./devinf12</LocURI></Target>
          </Item>
</Get >
```

Uma vez que o *post* tenha sido criado, ele será enviado para o servidor. Comoresposta o servidor deverá enviar um pacote HTTP OK. Comoas informações dos servidores são distintas, cada servidor será analisado separadamente.

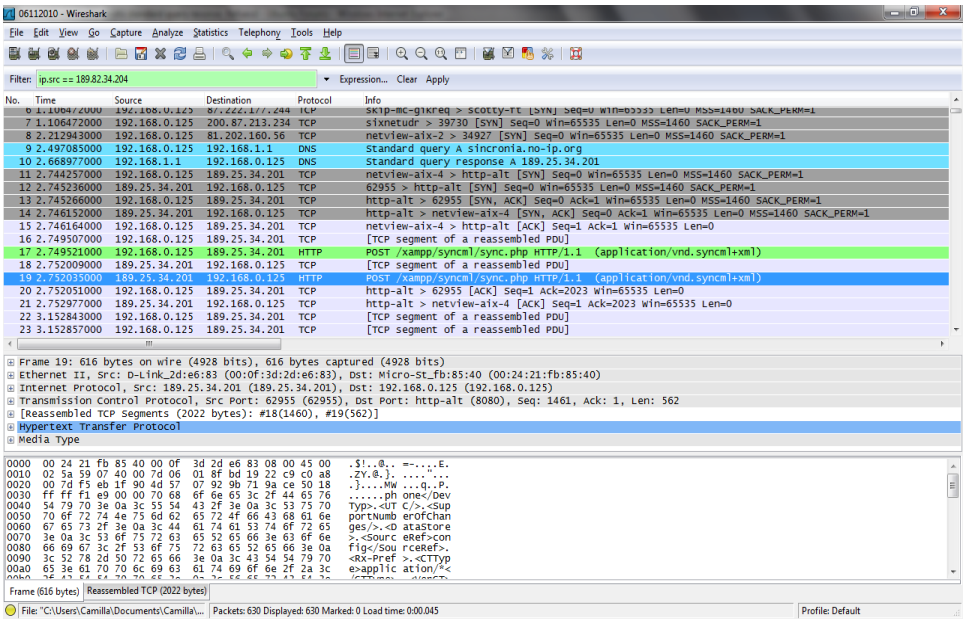


Figura 16 - Wireshark-Analisador de Protocolos.
Fonte: Wireshark, 12/2010

6.3 Servidor Memotoo

Todos os servidores possuem, basicamente, a mesma estrutura de resposta, compequenas modificações na ordenação da listagem de seus dispositivos e versões (validação de compatibilidade).

A primeira análise foi feita a partir do servidor da França chamado Memotoo. Através do *Wireshark* temos a resposta do servidor. Podemos identificar o mecanismo de conversação. A resposta do servidor está em anexo sobre o título *HTTP OK –Servidor Memotoo*.

Pode-se destacar que a resposta do servidor é dividia em duas partes. Ele valida as informações do cliente, retransmitindo os comandos solicitados o quefunciona como uma espécie de *acknowledge*. Para cada comando referido na seção anterior, o servidor deverá responder em sinal de entendimento, incluíndo as âncoras de sincronização. A segunda parte do documento diz respeito à capacidade de sincronização através do *SyncCap e SyncType*. Isto, no servidor Memotoo corresponde a todas as opções de sincronização: *Two-Way Sync*, *Slow Sync*, *One-Way Sync,Two-Way Sync* e *Server Sync Alerted*.Uma vez estabelecida a primeira fase, pode-se analisar o sincronização de dados.

O Memotoo utiliza várias portas para realizar a sincronização no estilo *Two Way Sync*. A primeira delas é o *post* na porta prizma. O *prizma* post faz um requerimento para novas trocas de âncoras referentesao tipo de arquivo (contatos, tarefas ou calendário pré-selecionado no celular - Cliente (ver figura 14).

O servidor responde ao prizma validando a informação e requerendo novas informações para qual tipo sincronização será tratado. Supondo que seja requerido um sincronização de contatos, um *post* será enviado, desta vez pela porta Lam pedindo umsincronização*Two Way*e informando sua âncora *Next*. O servidor responderáao *post* validando a informação e informando sua âncoras (*Next e Last*). A troca de informação ocorre dentro da estrutura *DataStore* nos objetos XML não exposto no código.

Uma vez acionado o sincronização de tarefas,um novo *post*seráemitido pelo cliente, utilizando a porta Isis (ver código apêndice). Neste caso, as âncoras de sincronização não precisaram ser validadas, pois pertecem a uma mesma sessão. Um comando *sync* é disparado, mas o *TargetRef* e o *SourceRef* serão diferentes.

Como a programação do servidor do Memotoo é atualizada, não há necessidade de se mostrar os *ctCaps*. Eles ocupam bastante espaço devidoà sua numerosidade, além de tornar o sistema frágil caso exista uma pessoa mal intencionada na rede querendo roubar os contatos ou agenda de outro usuário.

Todo o código de comunicação poderá ser verificado detalhadamente no apêndice. Existem dois datagramas tratados na parte teórica, que não estão coberto nesta avaliação e não foram incluídos no apêndice. São eles o *Data Update Status* e o *Map acknowwlegdement*. Eles podem ser vistos na análise do *Wireshark*, todavia, estão em *"Tokenized form"* (hexadecimal). Para o entendimento seria necessário uma conversão dos dados.

6.4 Servidor Funambol

O servidor Funambol é o servidor, entre os três estudados, que mais diverge em sua resposta inicial. Isto é atribuído às imensas opções de sincronização que a desenvolvedora dispõe.

O *SyncClient* envia ao servidor um requerimento para a sincronização de dados. Uma vez recebido o pedido,o *SyncServer* envia as respostas de todos os seus *DataStores*. Neste ponto é que a programação diverge,os dados serão: *briefcase, notes, e-mail, card, scard, snote, event* e *configuratio*n *scal*.

Dessas opções (*DataStore*) foram analisadas:*card, note, calendars, task e configuration* para poder ser feita a comparação. A comunicação completa em XML está exposta em anexo no Apêndice.

Apesar de todos os *DataStores* serem comunicados ao cliente einstanciados no servidor, a configuração inicial é a mesma feita no servidor Memotoo. Isso quer dizer que somente o *DataStore configuration* é populado no *SyncServer*. As âncoras de sincronização são transmitidas somente ao servidor pelo cliente, o inverso não ocorre. A resposta do servidor permite ao cliente acessar a segunda interface no *SyncClient* (figura 14). Pode–se ressaltar que a inicialização neste caso é realizada por dois *posts*.

Uma vez inicializado o protocolo SyncML resta a análise dasincronização. Para isso, novamente utiliza-se o *Wireshark*.

Mesmo depois de estar habilitado para asincronização, o sistema ainda não saberá qual será o seu conteúdo. Por isso quando se escolhe, por exemplo, o tipo calendário, é necessário haver troca das âncoras como se pode ratificar na comunicação XML no apêndice. Somente depois desta segunda fase é que os comandos *sync* serão utilizados.

Uma vez com as âncoras atualizadas asincronizaçãodo calendário será realizada pelo *DataStore* event do tipo de dado *xcalendar*. O *SyncClient* enviará suas modificações e ficará esperando pelas modificações do *SyncServer*. O servidor enviará a reposta através de um HTTP OK.

Para o caso de sincronização das tarefas primeiramente ocorrre a transmissão das âncoras de sincronização (padrão). O *SyncClient* informará sua âncora *Next* e recebe as âncoras *Next* e *Last* do servidor. Passada essa etapa, o *SyncClient* enviaráum novo *post* com *DataStore Task* sendo o seu *Cttype* do tipo *text/x-calendar*. Dessa maneira a informação será encaminhada dentro do campo *Data* e aparecerá na transmissão da seguinte maneira:

```
<Source><LocURI>330021558</LocURI></Source
    <Data>BEGIN:VCALENDAR..
        VERSION:1.0.
        BEGIN:VTODO..
        SUMMARY:TesteTarefa..
        DUE:20110119T235900..
        DESCRIPTION:..
        PERCENT-COMPLETE:0..
        STATUS:NEEDSACTION..
        UID:330021558..
        PRIORITY:5..
        RRULE :..
        AALARM:;;0;..
        END:VTODO..
        END:VCALENDAR..
    </Data >
```

O servidor receberá as informações e alimentará seu banco de dados com os novos dados. Depois de terminar o processamento de dados, ele enviará uma resposta validando a transferência.

No caso dasincronização dos contatos, o mesmo princípio será adotado. Assim como a transmissão do calendário, a mensagem de texto será passada através do *vcard* em forma de texto logo no primeiro *post* (após a inicialização).

Os campos do *vcard* são:

VERSION:2.1..N:; Ariel Soares;;;..
TEL;VOICE;HOME: ..
TEL;VOICE;HOME:..
TEL;VOICE;WOR K:..
TEL;VOICE;WO RK:..
TEL;PAGER:..
TEL;FAX;WORK:..
TEL;CELL:23467589..
TEL;VOICE:..
A DR;HOME:;;;;;;..
ADR;WORK:;;;;;;. .
ADR:;;;;;;..
BDA Y:..
EMAIL;
INTERN ET:..
EMAIL;INTER NET;HOME:..
EMAIL ;
INTERNET;WORK:. .
TITLE:..URL:..O RG:..NOTE:..
PHOTO:..CATEGORIES:..
END:VCARD

6.5 Servidor Nicolas Bougues

Por fim, iremos analisar o servidor Nicolas Bougues. Este servidor foi programado segundo a versão do Protocolo SyncML 1.0 e foi o escolhido para a implementação deste projeto. A estrutura XML é um pouco diferente. Por exemplo, todos os *ctCaps* são expostos na incialização do sistema. Isto acaba tornando o código um pouco maior.

Primeiramente a incialização éfeita normalmente inicializando os *DataStores* e o arquivos de armazenamento do celular no servidor. As âncoras de sincronização também sãotrocadas, o que permite que a primeira parte da transaçãoseja feita sempre problemas. A

primeira diferença significativa do servidor Nicolas Bougues surge com os comandos *SyncCap*. Pode-se observar no trecho retirado abaixo, que a transmissão só existe com dois *SyncTypes* de números 1 e 2.

```
<SyncCap>
        <SyncType>01</SyncType>
        <SyncType>02</SyncType>
        </SyncCap>
        </DevInf>
        </Data>
        </Item>
        </Results>
        <Alert>
        <CmdID>5</CmdID>
        <Data>201</Data><!-- 201 = TWO_WAY_ALERT -->
        <Item>
                …
        <Final/>
</SyncBody>
</SyncML>
```

Isto significa que asincronização só poderá ser realizada no estilo *Two-Way Sync* e no estilo S*low Sync*. Como visto anteriormente, o estilo S*low Sync*nada mais é que uma derivação do *Two-Way Sync* apenas de maneira mais segura. O *Server Alert* (*SyncType*) não é permitido, então não se tem como sincronizar os contatos a partir do desktop.Neste caso obrigatoriamente o celular, ou PDA deverá fazer a requisição.

Uma vez concluídaa inicialização,a troca de dados poderá ser analisada. Esta funciona com o mesmo princípio dosservidores descritos anteriormente. Por consequinte, somente as principais divergências serão analisadas.

Quando ocorre uma requisição por parte do *SyncClient*, o servidor Nicolas devolve a resposta do comando *Sync* com dois comandos.São eles o *Add* e o *Replace*. O servidor analisará em seus dados, através do LocURI, se a presente informação existe ou não em seu domínio. Se a informação disponível no servidor for mais atual do que a recebida do cliente, uma estrutura <*Replace*>será utilizada para informar sobre as modificações ao cliente. Caso contrário, a informação permanecerá na estrutura <*Add*>. Toda resposta de sincronização do contato terá essas duas estrututras, mesmo que não haja informação a ser preenchida. Neste caso a estrutura permanecerá vazia (Ex:<*Data*></*Data*>).

Outra diferença do servidor do Nicolas é que não há a existência do conteúdo Tarefas (*Tasks*) somente Note e Calendar.

Para a sincronização do *Note*, primeiro ocorre um envio de trocas das últimas âncoras de sincronização para balisar o sistema. Somente depois o cliente enviará um *post* informando sobre suas modificações. Caso não haja nenhuma *note* anterior, será utilizado por definição o elemento zero nos *flags* internos *local_last_anchor* e *remote_last_anchor*. Este procedimento é válido para todos os outros conteúdos (contatos e calendário). No primeiro teste os *flags* não estavam setados corretamente.

O conteúdo *note* foi adaptado para o projeto, visto que o código do Nicolas Bougues não possuia o conteúdo tarefa.

Por fim temos a sincronização do conteúdo calendário. Novamente o conteúdo é do tipo *text* soba forma de *vcalendar*. O campo *Data* é o responsável por armazenar os dados do *vcalendar*.

A cada nova informação no calendário, ou seja, a cada novo registro inserido, obrigatoriamente existirá uma nova estrutura como o código mostrado anteriormentenas modificações do cliente. Esta estrutura será comparada aos dados existentes no *SyncServer*. Caso não ocorra modificações o servidor enviará uma resposta repetindo todos os dados. E fica a espera de *acknowledge* para finalmente transmisitir uma nova mensagem de dados (PDU – *Protocol Data Unit*) com o mapeamento final.

Capítulo 7 Falhas Detectadas

7.1 Incompatibilidade das Versões do SuncML

As versões SyncML 1.0 e a versão SyncML 1.2 não possuem grande divergência no que diz respeito ao conteúdo em si. A maior modificação de conteúdo está associada aos *ctCaps* anteriomente explicados. Porém, no que tange a estrutura da comunicação, o código de Nicolas Bougues não continha o *handle* para o *DataStore* de referência interna *configuration*. Foi necessário inserir algumas linhas para a adaptaçãodesta função.

Outra dificuldade está na ausência do conteúdo tarefas no código de Nicolas Bougues. Como o celular usado para a criação do sistema era um celular Erickson, do ano de 2000, a aplicação estava restrita a *vnotes vcards e calendars*. Para contonar a situação, as tarefas inseridas no programa *standalone* deverão ser consideradas no servidor como notes possuindo os campos nome da tarefa, data de emissão, data prevista para a conclusão e data de término.

7.2 Problemas de Retransmissão de Pacotes

Os problemas de retransmissão de pacotes tornam a leitura da comunicação bastante árdua, além de tornar a transmissão mais demorada. Existe uma sequência de pacotes trocadosentre o cliente e o servidor, uma espécie de pré-comunicação, a fim de estabelecer uma conexão para a efetiva troca de dados.

A maior perda de pacotes na comunicação foi no servidor Memotoo. Nos pacotes iniciais não cabiam todo o conteúdo utilizado pelo servidor. Para contornar este problema o servidor envia primeiramente um datagrama (TCP segment of reassembled PDU) sob o protocolo TCP com a informação incial.Logo depois, envia um datagrama sob o protocolo HTTP com o restante da informação. A falha se devido à demora entre os pacotes da comunicação entre si (HTTP e TCP). A partir disso o sistema de pré-comunicação ficou desordenado e as mensagens perderam a sequencia. Quando isso ocorre é necessário reiniciar toda a comunicaçõ.

Este tipo de comunicação também é utilizado pelo servidor de Nicolas Bougues, mas a quantidade de retransmissões de informação foi bem menor.

O sistema Funambol, evitando este problema, utilizou um sistema de codificação para o conteúdo. Este servidor possui o maior código de inicialização dentre os três servidores e não possui erros na transmissão, o que torna a conexão e a transmissão de dadosmelhor. Basicamente o *SyncServer* utiliza um datagrama para passar toda a informação.

Capítulo 8 – Conclusões Finais

O presente projeto teve por desiderato atingir os seguintes objetivos: fazer um programa que possua uma ligação com um banco de dados capaz de sincronizar uma conta para cada usuário e armazenar os dados deste usuário; sincronizar os dados armazenados em um celular com os dados armazenados em computador a partir da *Internet*; e, por fim, propiciar a análise da transmissão de dados XML de vários servidores diferentes a fim de comparar os vários resultados.

A programação do *software standalone* foi realizada em linguagem JAVA. Ela obteve êxito ao guardar as informações do usuário, além de possibilitar a sua autenticação. Uma vez realizada a autenticação do usuário, o sistema distribui, entre as diversas interfaces, a validação da conta. Isso evita a necessidade de novas autenticações, além de excluir um possível período de expiração. Contudo, o usuário só tem acesso ao programa depois do administrador o ter registrado, prática comumente empregada nos softwares. Esta última ressalva tornou o campo endereço eletrônico (*e-mail*) obrigatório.

A conexão com os bancos de dados também foi executada de maneira eficiente. Na realidade, esta conexão é a chave para a comunicação entre os dois diferentes sistemas. Isso significou integrar a programação do servidor PHP com a programação Java através do livre acesso às tabelas. Para que tudo ocorresse de maneira eficiente, as tabelas foram armazenadas em um mesmo *Database*. Mesmo assim, ainda foinecessário criar a classe conexão de dados visando manter o sistema atualizado.

Para alcançar o objetivo da sincronização do celular com o computador, foi feita uma forte pesquisa de ferramentas auxiliares. A *priori*, inúmeros testes foram realizados com os simuladores de *smartphone* da Nokia, aproveitando que esta era uma das fundadoras do consórcio OMA. Porém, mediante a complicações entre a biblioteca winsock e o sistema operacional, o simulador escolhido foi o da Blackberry com o *SyncClient* provido pela Funambol.

Estabelecido o cliente alterou-se o servidor de Nicolas Bougues de maneira a atender todos os requisitos. A sincronização foi feita para o conteúdo *vcards, vnotes, vcalendar* como tinha sido proposto. Como a programação do servidor de Nicolas Bougues não possuía a sincronização de tarefas, o conteúdo *vnotes* passou a ser utilizado com o nome da tarefa e as datas de previsão, início e término. Uma vez realizada a sincronização, pode-se fazer a comparação entre o três servidores.

Para finalizar a totalidade das metas previstas no projeto, passou-se a fazer testes em torno do protocolo SyncML. A grande quantidade de resultados dos três servidores forneceu grande material para a análise da transmissão. Todas as transmissões foram captadas pelo analisador de pacotese incluídasno apêndice. O *SyncClient* sempre iniciará a requisição do pedido enviando um *post* para servidor. Este *post*, em todos os casos, servirá para a autenticação do usuário. O servidor, por sua vez, enviará a sua resposta inicializando todos os seus recursos e esperando que o cliente informe qual o conteúdo que deverá ser sincronizado. A partir desta primeira etapa qualquer conteúdo poderá ser escolhido para a sincronização.

Esta primeira etapa dasincronização se mostrou semelhante em todos os servidores. Apenas o servidor da Funambol teve uma diferença com relação aos outros. Ele continha sua informação XML compactada dentro do datagrama. A resposta inicial foi enviada em apenas um pacote e continha o maior código dentre todos. Os outros servidores, Memotoo e Nicolas Bougues, sempre utilizaram dois pacotes para a resposta ser completa, sendo o primeiro utilizandoo protocolo TCP (TCP segment of reassembled PDU) e o segundo utilizando o protocolo HTTP. Esta diferença não ressaltou aos olhos à primeira vista, porém, à posteriori, foi identificada como a grande bem feitora por não haver falhas de comunicação e necessidade de retransmissão de informação no servidor Funambol. Este servidor mostrou-se o mais eficiente dentre os três servidores graças à solução de codificar a informação.

Uma vez definida e validada a conexão do protocolo SyncML o estudo foi direcionado àsincronização dos conteúdos. Assim sendo, foi registrado que a troca de âncoras é um processo estável e simples como proferido pela desenvolvedora. Isto torna o processo viável, rápido e confiável. O método *Slow Sync*, descrito para ser utilizado uma vez que ocorra ruptura durante a conexão do T*wo Way Sync*, nunca foi acionado.

Desta maneira o protocolo atendeu a todas as expectativas e se mostrou uma poderosa ferramenta para os programadores. Uma vez confirmada e validada a sua integridade, todos os objetivos do projeto foram atendidos.

Este trabalho se predispôs a iniciar e esclarecer uma das novas tecnologias que prometem despontar no mercado e estar nocotidiano das pessoas. Novos projetos poderão ser realizados a partir deste ou apenas utilizá-lo como base.

A realização desse trabalho propiciou o desenvolvimento dos conhecimentos nas linguagens de programação JAVA e PHP, assim como a descobertade funcionalidades e características do protocolo SyncML. Outra grande ferramenta que se pôde trabalhar foi na modelagem do problema e necessidades de atender a todas as especificações.

Outros trabalhos poderão ser desenvolvidos a fim de completar esta linha de pesquisa ou de aprimorar o projeto para ser igualado aos outros aqui estudados

Referências Bibliográficas

ALTN TECHNOLOGY COMPANY. **SyncML Setup for BlackBerry** Disponível em: <http://www.altn.com/Products/MDaemon-Email-Server-Windows/BlackBerry-SyncML/>. Acesso em: 17 set. 2010.

ARISTÓTELES, E. S. **The Organon**. Califórnia: Harvard University Press, 1958. v.3.

BOUGUES, Nicolas. **SyncM Tools**. 2004. Disponível em: <http://nicolas.bougues.net/syncml/>. Acesso em: 13 set. 2010.

BROOKS, David R. **Introduction to PHP for scientists and engineers**: beyond JavaScript Springer. [S.l.: s.n.], 2008. 141p.

CHAPPELL, Laura. **Wireshark network**: The official Wireshark Certified Network Analyst study Guide. Laura Chappell Univeersity, Local: editora, 2010.

COMBS, Gerald. **Documentation and Resource.**Disponível em: <Wireshark: <http://www.wireshark.org/>. Acesso em: 13 set. 2010.

DEITEL, Harvey M. **XML**: como programar. São Paulo: Bookman, 2003. 972p.

DYER, Russel J. T. **MySql in a Nutshell**. [S.l.]: O'Reilly Media, 2008. 545p.

FLANAGAN, David. **Java in a Nutshell**: A Desktop Referenc. 5.ed. [S.l.]: O´Reilly, 2005.1147p.

FULTUS CORPORATION. **Apache Sofware Foundation**: Apache HTTP Server 2.2 Server Administration. [S.l.: s.n.], 2010. v.1.

HARVEY, M. et al. **Como Programar**. São Paulo: Bookman, 2003. 972p.

LEE. T. Berners. **Projeto Worl Wilde Web CERN**. Geneva Suiça, 1994. Disponível em: http://tools.ietf.org/html/rfc1630)>. Acesso em: 17 out. 2010.

NO-IP DNS SERVICE. **Download**. Disponível em: <http://www.no-ip.com/downloads.php>. Acesso em: 24 ago. 2010.

_____. **Home**. Disponível em: <http://www.no-ip.com/downloads.php>. Acesso em: 24 ago. 2010.

OMA DATA SYNCHRONIZATION WORKING GROUP. **File Format Info**. Disponível em: <http://www.fileformat.info/info/mimetype/application/vnd.syncml.+xml/index.html>. Acesso em: 17 out. 2010.

OPEN MOBILE ALLIANCE - OMA. **Protocolo SyncMl Device Information versão 1.1**. [S.l.: s.n.], 2001a. Disponível em: <http://www.openmobilealliance.org/tech/affiliates/syncml/syncmlindex.html#V101>. Acesso em: 17 out. 2010.

_____. **Protocolo SyncMl versão 1.0.1**. [S.l.: s.n.], 2001b. Disponível em: <http://www.openmobilealliance.org/tech/affiliates/syncml/syncmlindex.html#V101>. Acesso em: 17 out. 2010.

ORACLE COMPANY. **Java 2 Plataform Std Edv1.4.2**. 2003. Disponível em: <http://download.oracle.com/javase/1.4.2/ docs/api/java/sql/ ResultSet.html. Acesso em: 10 nov. 2010.

_____. **Technology Database**: the Java Database. Technology Disponível em: <http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>. Acesso em: 10 nov. 2010.

RESEARCH IN MOTION BLACKBERRY. **BlackBerry Java Development Environment Version: 4.7.0**:Fundamentals Research In Motion Limited, 295 Phillip Street,Waterloo. Ontário, Canada, 2008. Disponível em: <http://us.blackberry.com/developers /javaappdev/javadevenv.jsp>. Acesso em: 10 nov. 2010.

_____. **Guia do desenvolvedor**: versão 1.0.1. Ontário, Canada, 2010. Disponível em: <http://docs.blackberry.com/pt-br/developers/deliverables/16728/BlackBerry_Smartphone_Simulator--1001926-0618115637-012-5.0-PT.pdf>. Acesso em: 10 nov. 2010.

RESEARCH IN MOTION LIMITED. **Guia do Desenvolver**. Canada, 2010. Disponível em: <http://docs.blackberry.com/pt-br/developers/deliverables/16728/BlackBerry_Smartphone_Simulator--1001926-0618115637-012-5.0-PT.pdf>. Acesso em: 17 out. 2010.

RIEHLE, Dirk. **Framework Design:** A Role Modeling Approach. Tese Doutorado em Computação e Frameworks- Zürich,Switzerland. ETH Zürich, 2000.
ROSS, Júlio. **Rede de Computadores.** [S.l.]: Antena Edições técnicas, 2005. 148p.

SHEKHAR, Viajay. **Red HAT Linux**: the complete bible. 2.ed. [S.l.]: Laxmi Publications, 2008. 1184p.

SANTOS, Ciro M.. **Programação orientada a objeto**. 2009. Disponível em: http://www.tga-online.com.br/index.php?option=com_content&view=article&id=78&Itemid=56). Acesso em: 17 set. 2010.

SILVA, Carlos Eduardo; SANTOS, César Frederico dos. **Classification in Object**: Oriented System (Resenha). Autor do artigo Peter Wegner. [S.l.]: UFSCAR, 1999.

ULMANN, Larry. **Visual QuickStartGuide**. 2.ed. [S.l.]: PeachPit Press, 2006. 464p.

WORKBENCH COMPANY. **MySQL Workbench. Disponível em:**<http://wb.mysql.com/>. Acesso em: 10 jul. 2010.

XAVIER, Beatriz Rego. As categorias de Aristóteles e o conhecimento científico. **Revista da OAB,** Ceará-Fortaleza, v. 13, n. 1, p. 57-64, jan./jun. 2008.

APÊNDICE

Post SyncClient

```
POST /xampp/syncml/sync.php HTTP/1 .1..
Host: sincronia.no-ip.org:8080..
Connection:  Close..
Content-Type: application/vnd.syncml+xml..
Content-Length:  1714..
User-Agent: Funambol BlackBerry Plug-in 8 .7.1..
Device-Agent: BlackBerry85 20/5.0.0.681 MID P-2.1 CLDC-1.1..
Accept-Encoding:  gzip..
Content-Language: en-GB..
..
<SyncML>
     <SyncHdr>
          <VerDTD>1.2</VerDTD>
          <VerProto>SyncML/1.2</ VerProto>
          <SessionID>1289048219062</SessionID>
          <MsgID>1</MsgID>
          <Target><LocURI>http://sincronia.no-
ip.org:8080/xampp/syncml/sync.php</LocURI></ Target>
          <Source>
          <LocURI>fbb-
          553648138</LocURI><LocName>sincronia</LocName></Sourc
          e>
     <Cred>
          <Meta >
          <Type xmlns="sy ncml:metinf">syncml:auth-basic</Type>
          <Format xmlns="syncml:metinf">b64</Format>
          </Meta>
               <Data>c2luY3JvbmlhOnNpbmNyb25pYQ==</Data></Cred>
          <Meta>
          <MaxMsgSize>65536</MaxMsgSize></ Meta>
     </SyncHdr>
     <SyncBody>
          <Alert>
               <CmdID>1</CmdID>
               <Data>200</Data>
          <Item>
               <Target>
                    <LocURI>configuration</LocURI>.
               </Target>
          <Source><LocURI>config</LocURI></Source>
          <Meta>
          <Anchor xmlns="syncml:metinf">
               <Last>1288497949375 </Last>
               <Next>1289048219062</Next>
          </Anchor>
          </Meta>
          </Item>
          </Alert>
```

```
     <Put>
          <Cmd ID>2</CmdID>
          <Meta>
          <Type xmlns= 'syncml:metinf'> application/vnd.syncml-
devinf+xml</Type>
          </Meta>
          <Item>
          <Source><LocURI>./devinf 12</LocURI></Source>
          <Data>
               <DevInfxmlns='syncml:devinf'>
               <VerD TD>1.2</VerDTD>
               <Man>Research In  Motion</Man>
               <Mod>8520</Mod>
               <OEM></OEM>
               <FwV>< /FwV>
               <SwV>8.7.1 </SwV>
               <HwV>2.13.0.140</HwV>
               <DevID>fbb-553648138</DevID>
               <DevTyp>phone</DevTyp>
               <UTC/>
               <SupportNumberOfChanges/>
          <DataStore><SourceRef>config</SourceRef>
          <Rx-Pref>
               <CTType>application/*</CTType>
               <VerCT></VerCT>
               </Rx-Pref>
               <Tx- Pref>
               <CTType>application/*</CTType>
               <VerCT></VerCT>
               </Tx-Pref>
          <SyncCap>
               <SyncType>1</SyncType>
               <SyncType>2</SyncType>
               <SyncType>7</SyncType>
          </SyncCap>
          </Data Store>
          </DevInf>
          </Data>
          </Item>
     </Put>
     <Get>
          <CmdID>3</CmdID>
          <Meta><Type
          xmlns='syncml:metinf'>application/vnd.syncml-
          devinf+xml</Type></Meta>
          <Item>
               <Target>< LocURI>./devinf12</LocURI></Target>
          </Item>
     </Get >
     <Final/>
     </Sync Body>
     </SyncML>.
```

Servidor Memotoo
Resposta da Inicialização

```
HTTP/1.1 200 OK..
Date: Sat, 06 Nov 2010 22:39:34 G MT..
Server: Apache..
Set-Cookie:  PHPSESSID=3e41b540e867475099c30a5aab795379; path
=/..
Content-Leng th: 2511..
Connection: close..
Content-Type: application/vnd.syncml+xml..
..
<SyncML >
     <SyncHdr>
          <Ver DTD>1.2</VerDTD>
          <VerProto>SyncML/1.2</VerProto>
          <SessionID>1289083174140</SessionID>
          <MsgID>1</ MsgID>
          <Target><LocURI>fbb-553648138</LocURI>
               <LocName>sincronia</LocName>
          </Target>
               <Source><
LocURI>http://sync.memotoo.com/syncml</LocURI>.</Source>
               <RespURI>http://sync.memotoo.com/syncml ?
sid=3e41b540e86 7475099c30a5aab7 95379</RespURI>.
     <Meta>.<MaxMsgSi ze xmlns="syncml :metinf">20000</
MaxMsgSize>.</Me ta>
     </SyncHdr>
          <SyncBody>
          <Statu s>
          <CmdID>1</Cmd ID>
          <MsgRef>1</M sgRef>
          <CmdRef>0 </CmdRef>
          <Cmd>SyncHdr</Cmd>

     <TargetRef>http://sync.memotoo.com/syncml</TargetRef>
               <SourceRef>fbb-553648138</SourceRef>
               <Data>212</Data>
               </Status>
          <Status>
               <CmdID>2</CmdID>
               <Ms gRef>1</MsgRef>
               <CmdRef>1</CmdRe f>
               <Cmd>Alert</Cmd>
               <TargetRef>configuration</TargetRef>
               <SourceRef>config</SourceRef>
               <Data>200 </Data>
               <Item>
                    < Data><Anchor xm lns="syncml:meti nf">
                    <Next>1289083174140</Next>
```

```
                    </Anchor>
                    </Data >
                    </Item>
                    </Status>
               <Status>
               <Cm dID>3</CmdID>
               <MsgRef>1</MsgRef>
               <CmdRef>2</CmdRef>
               <Cmd>Put</Cmd>
               <SourceRef>./devinf12</Source Ref>
               <Data>200</ Data>
               </Status>
          <Status>
               <CmdID> 4</CmdID>
               <MsgRe f>1</MsgRef>.<CmdRef>3</CmdRef>
               <Cmd>Get</Cmd>
               <TargetRef>./devinf12</TargetRef>
               <Data>200</Data >
          </Status>
          <Results>
               <CmdID>5</CmdID>
               <MRef>1</MsgR ef>1</MsgRef>
               <CmdRef>3</CmdRef>
               <Meta>.<Type xm lns='syncml:meti
nf'>application/ vnd.syncml-devinf+xml</Type>.</M eta>

     <Item>.<Source>.<LocURI>./devinf12</LocURI></Source>
               <Data >
                    <DevInf xmlns= 'syncml:devinf'>

                    <VerDTD>1.2</Ve rDTD>
                    <Man>Memotoo.com</Man>
                    <Mod>SyncML</Mod>
                    < OEM>-</OEM>.<FwV >-</FwV>.<SwV>2.
17</SwV>
                    <HwV>-</HwV>
                    <DevID>memotoo</DevID>
                    <DevTyp>server</DevTyp>
                    <UTC/>.<SupportLargeObjs/>
                    <SupportNumberOf Changes/>
                    <DataStore>
                         <SourceRef>picture</SourceRef>
                         <DisplayName>Pictures</DisplayName>
                         <Rx-Pref>
               <CTType>application/vnd.omads-file+xml</CTType>
                         </Rx-Pref>
                         <T x-Pref>
                         <CTType> application/vnd.comands-
file+xml</ CTType>
                         </Tx-Pref>
                              <SyncCap>
                                   <SyncType>1</SyncType>
                                   <SyncType>2</SyncType>
```

```
                                        <SyncType>3</SyncType>
                                        <SyncType>4</SyncType>
                                        <SyncType>5</SyncType>
                                        <SyncType>6</SyncType>
                                        <SyncType >7</SyncType>
                              </SyncCap>
                         </DataStore>
                         <Ext>
                         <XNam>X-funambol-smartslow</XNam>
                         </Ext>
                         <Ext>.<XNam>X-funambol-media-http-upload</X
Nam></Ext>
                         </DevInf>.</Data>
                         </Item>
                         </Results>
                         <Alert><CmdID>6</CmdID>
                         <Data> 200</Data>
                         <Item >

     <Target>.<LocURI>config</LocURI>.</Target>

     <Source>.<LocURI>configuration</LocURI>.</Source>
                              < Meta>.<Anchor xmlns='syncml:meti
nf'>
                                   <Last>1289077023000</Last>
                                   <Next>1289079574000</Next>
                              </Anchor>
                              </Meta>
                              </Item>
                              </Alert>
                         <Final/>
                         </SyncBody>
          </SyncML>
```

Post Prizma

```
POST /syncml HTTP/ 1.1..
Host: sync. memotoo.com..Con nection:Close..
Content-Type: application/vnd.syncml+xml..
Conten t-Length: 1511..
User-Agent: Funambol BlackBerry  Plug-in 8.7.1..
Device-Agent: BlackBerry8520/5.0.0.509 MIDP-2.1 C LDC-1.1..
Accept- Encoding: gzip..
Content-Language: en-GB..
..
<SyncML>
     <SyncHdr>
          <VerDTD>1.2</VerDT D>
          <VerProto>Syn cML/1.2</VerProto>
          <SessionID>12 86482723593</Ses sionID>
          <MsgID>1 </MsgID>
```

```
          <Target ><LocURI>http://sync.memotoo.com
/syncml</LocURI></Target>
          <Source><LocURI>fbb-55
3648138</LocURI><LocName>sincronia</LocName></Source>
          <Cred>
               <Meta><Type xmlns=" syncml:metinf">syncml:auth-
basic </Type>
                    <Format xmlns="syncml:me tinf">b64</Format>
               </Meta>
               <Data >c2luY3JvbmlhOnN pbmNyb25pYQ==</Data>
          </Cred>
          <Meta><MaxMsgSize>65 536</MaxMsgSize></Meta>
     </SyncHdr>
     <SyncBody>
     <Alert>
     <CmdID>1</ CmdID>
     <Data>200 </Data>
     <Item>
     < Target><LocURI>card</LocURI></Target>
     <Source>< LocURI>contact</ LocURI></Source>
     <Meta>
          <Anchor  xmlns="syncml:metinf">
               <Last>128 6482641703</Last >
               <Next>12864827 23593</Next>
          </Anchor>
     </Meta>
     </Item>
     </Alert>
     <Put>
     <CmdID>2</CmdID>
     <Meta><Type xmlns='syncml:metinf'>application/vnd.syncml-
devinf+xml</Type>.</Meta>
     <Item>
          <Source><LocURI>./devinf12</LocURI></Source><Data>
          <DevInf xmlns='syncml:devi nf'>
          <VerDTD>1.2 </VerDTD>
          <Man>Research In Motio n</Man>
          <Mod>852 0</Mod>
          <OEM></OEM>
          <FwV></FwV>
          <SwV>8.7.1</SwV>
          <HwV>2.13.0.97</HwV>
          <DevID>fbb -553648138</DevID>
          <DevTyp>phone</DevTyp>.<UTC/>
          <SupportNumberOfChanges/>
          <DataStore>
               <SourceRef>contact</S ourceRef>
               <Rx-Pref>
               <CTType>text /x-vcard</CTType >
               <VerCT></VerCT >.</Rx-Pref>
               <Tx -Pref>
                    <CTType>text/x-vcard</CTT ype>
                    <VerCT></VerCT>
```

```
                    </Tx-Pref>
                    <SyncCap>
                    <SyncType>1</SyncType>
                    <SyncType>2</Sy ncType>
                    <SyncType>7</SyncType>
                    </SyncCap>
          </Data Store>
          </DevInf>
          </Data>
     </Item>
     </Put>
     <Final/>
     </SyncBody>
     </SyncML>
```

Reposta Post Prizma

```
HTTP/1.1 200 OK..
Date: Thu, 07 Oct  2010 2018:38 G MT..
Server: Apache..
Set-Cookie: PHPSESSID=b7c52e62c85f9a16a4d8cdf781cc464e; path
=/..
Content-Leng th: 1346..
Connection: close..
Con tent-Type: application/vnd.syncm l+xml..
..
<SyncML >
     <SyncHdr>
          <VerDTD>1.2</VerDTD>
          <VerProto>SyncML/1.2</VerProto>
          <SessionID>1286482723593</SessionID>
          <MsgID>1</ MsgID>
          <Target><LocURI>fbb-553648138</LocURI>
               < LocName>sincronia</LocName>
          </Target>
          <Source><
LocURI>http://sync.memotoo.com/syncml</LocURI></Source>
          <RespURI>http://sync.me motoo.com/syncml?
sid=b7c52e62c85f9a16a4d8cdf781cc464e</RespURI>
          <Meta>
               <MaxMsgSi ze xmlns="syncml:metinf">20000</
MaxMsgSize>
          </Meta>
     </SyncHdr>
     < SyncBody>
     <Status>
          <CmdID>1</Cmd ID>
          <MsgRef>1</M sgRef>
          <CmdRef>0</CmdRef>
          <Cmd>SyncHdr</Cmd>
          <TargetRef>http://sync.memotoo.com/syncml</TargetRef>
          <SourceRef>fbb-553648138</SourceRef>
```

```
          <Data>21 2</Data>
     </Statu s>
     <Status>
          <CmdID>2</CmdID>
          <MsgRef>1</MsgRef>
          <CmdRef></CmdRe f>
          <Cmd>Alert</Cmd>.
          TargetRef>card</TargetRef>
          <SourceRef>contact</SourceRef>
          < Data>200</Data>
          <Item><Data>
                    <Anchor xmlns="syncml:metinf">
                    <Next>1286482723593 </Next>
                    </Anchor >
               </Data>
          </Item >
     </Status>
     <Status>
          <CmdID>3</CmdID>
          <MsgRef>1< /MsgRef>
          <CmdRef >2</CmdRef>
          <Cmd >Put</Cmd>
          <SourceRef>./devinf12 </SourceRef>
          <Data>200</Data>
     </Status>
     <Alert>
     <CmdID>4</CmdID>
          <Data>200</Data>
          <Item>
          <Target>.<LocURI>contact</LocURI>.</Target>
          <Source>.<LocURI>card</LocURI>.</Source>.
          <Meta>.<Anchor xmlns='syncml:metinf'>
          <Last>1286479036000</Last>
          <Next>1286479118000</Next>
</Anchor>
</Meta>
</Item>
</Alert>
<Final/>
</SyncBody>
</SyncML>
```

Post Lam

```
POST /syncml?sid=b7c52e62c85f9a16a4d8cdf781cc464e
HTTP/1.1
Host:sync.memotoo.com
Connection: Close
Content-Type: application/v nd.syncml+xml
Content-Length: 9 75
User-Agent:  FunambolBlackBerry Plug-in 8.7. 1
Device-Agent:  BlackBerry8520/ 5.0.0.509 MIDP-2 .1 CLDC-1.1
```

```
Accept-Encoding: gz ip
Content-Lang uage: en-GB

< SyncML>
     <SyncHdr>
          <VerDTD>1.2</VerDTD>
          <VerProto >SyncML/1.2</Ver Proto>
          <SessionID>1286482723593</SessionID>
          <MsgID>2</MsgID>
          <Target><LocURI>http://sync.memotoo.com/syncml?
sid=b7c52e62c85f9a16a4d8cdf781cc464e </LocURI></Target>
          <Source><LocURI>fbb-
553648138</LocURI><LocName>sincronia</Loc Name></Source>
          <Meta><MaxMsgSize >65536</MaxMsgSi ze></Meta>
     </SyncHdr>
<SyncBody>
<Status>
     <CmdID>1</CmdID>
     <MsgRef>1</MsgRef>
     <CmdRef>0</CmdRef>
     <Cmd>SyncHdr</Cmd>
     <TargetRef>fbb-553648138</Ta rgetRef>
     <Source Ref>http://sync.memotoo.com/syncml</SourceRef>
     < Data>200</Data>
</Status>
<Status>
     <CmdID>2</Cmd ID>
     <MsgRef>1</M sgRef>
     <CmdRef>4< /CmdRef>
     <Cmd>Ale rt</Cmd>
     <Target Ref>card</Target Ref>
     <SourceRef>contact</SourceR ef>
     <Data>200</Data>
     <Item><Data><Anchor xmlns ="syncml:metinf" >
     <Next>1286482723593</Next></Anchor></Data></Item>
</Status>
< Sync>
     <CmdID>3</ CmdID>
     <Target><LocURI>card</LocURI></Target>
     <Source><LocURI>contact</LocURI></ Source>
</Sync>
<Final/>
</SyncBody>
</SyncML>.
```

Post Isis (SyncClient)

```
POST/syncml?sid=b7c52e62c85f9a16a4d8cdf781cc464e  HTTP/1.1..
Host:  sync.memotoo.com ..
Connection: Close..
Content-Type: application/vnd.syncml+xml..
```

```
Content-Length: 9 53..
User-Agent:  Funambol
BlackBerry Plug-in 8.7. 1..
Device-Agent:  BlackBerry8520/ 5.0.0.509 MIDP-2 .1 CLDC-1.1..
Accept-Encoding: gz ip..
Content-Language: en-GB..
..
< SyncML>
     <SyncHdr>
          <VerDTD>1.2</VerDTD>
          <VerProto >SyncML/1.2</Ver Proto>
          <SessionID>1286482723593</SessionID>
          <Msg ID>4</MsgID>
          <Target>
               <LocURI>http://sync.memotoo.com/syncml?
sid=b7c52e62c85f9a16a4d8cdf781cc464e</LocURI>
          </Target>
          <Source><LocURI>fbb-553648138 </LocURI>
          <LocName>sincronia</Loc Name>
          </Source>
     </SyncHdr>
     <SyncBody>
     <Status>
          <CmdID>1</CmdID>
          <MsgRef>3</MsgRef>
          <CmdRef>0</Cmd Ref>
          <Cmd></Cmd>
          <TargetRef>fbb-553648138</Targe tRef>
          <SourceRef
>http://sync.memotoo.com/syncml</SourceRef>
          <Data>200</Data>
     </Status>
     <Status>
          <CmdID>2</CmdID>
          <MsgRef>3</MsgR ef>
          <CmdRef>0</C mdRef>
          <Cmd>Sync Hdr</Cmd>
          <TargetRef>fbb-553648138</TargetRef>
          <SourceRef>http://sync.memotoo.com/syncml?
sid=b7c52e62c85f9a16a4d8cdf781cc464e</SourceRef>
          <Data>200</Data>
     </Status>
     <Status><CmdID>3</CmdID>
     <MsgRef>3</MsgRef>
     <CmdRef>2</CmdR ef>
     <Cmd>Sync</Cmd>
     <SourceRef>card</SourceRef>
     <TargetRef>task</TargetRef>
     <Data>200</Data>
     </Status>
     <Final />
     </SyncBody>
</ SyncML>.
```

Servidor FUNAMBOL
Post de Inicialização (SyncClient)

POST/syncHTTP/1. 1..
Host:my.funambol.com
..
Connec tion: Close..

```
Content-Type: application/vnd.syncm l+xml..
Content-L ength: 1712..Use r-Agent:
Funambol BlackBerry Plu g-in 9.0.1..
Device-Agent: BlackBerry8520/5.0.0.5 09 MIDP-2.1 CLDC -1.1..
Accept-Enc oding: gzip..Con tent-Language: en-GB..
..
<SyncML>
     <SyncHdr>
     <VerD TD>1.2</VerDTD>
     <VerProto>SyncML /1.2</VerProto>
     <SessionID>12953 07389165</Sessio nID>
     <MsgID>1</M sgID>
     <Target>
     <LocURI><![CDATA[http://my.funambol.com/sync]]></LocURI>
     </Target>
     <Source>
          <LocURI> fbb-553648138</L ocURI>
          <LocName>camilla.gueiros</ LocName>
     </Source >
     <Cred>
          <Meta>< Type xmlns="sync ml:metinf">syncm l:auth-
basic</Ty pe>.           <Format xmln s="syncml:metinf
">b64</Format>
          < /Meta>
     <Data>Y2FtaWxsYS5ndWVpcm9 zOnNpbmNyb25pYQ= =</Data>
     </Cred>
     <Meta><MaxMsgSize>65536< /MaxMsgSize></Meta>
     </SyncHdr>
     <SyncBody >
     <Alert>
     <CmdID >1</CmdID>
     <Data >200</Data>]
     <Item>
          <Target><LocURI>configuration </LocURI>.</Target>
          <Source><LocURI>config</LocU RI></Source>
          <Meta>.<Anchor xmln s="syncml:metinf ">
               <Last>1295307253041</Last>
               <Next>129530738916 5</Next>
               </Anchor>
          </Meta>
     </Ite m>
     </Alert>
     <Put >
          <CmdID>2</CmdID>
```

```
          <Meta>
          <Type  xmlns='syncml:metinf'>application/vnd.syncml-
dev inf+xml</Type>.      < /Meta>
     <Item>
     <Source><LocURI>./ devinf12</LocURI ></Source>
     <Data >
          <DevInf xmlns= 'syncml:devinf'> .<VerDTD>1.2</Ve
rDTD>
          <Man>Resea rch In Motion</Man>
          <Mod>8520</Mod>
          <OEM></OEM>.
          <FwV></FwV>
          <SwV >9.0.1</SwV>
          <Hw V>2.13.0.97</HwV >
          <DevID>fbb-553 648138</DevID>
          < DevTyp>phone</De vTyp>.<UTC/>
          <SupportNumberOfCha nges/>.<DataStor e>
          <SourceRef>config</SourceRef>
          <Rx-Pref>
               <CTType>application/* </CTType>
          <VerCT ></VerCT>]
          </Rx-P ref>
          <Tx-Pref>
               < CTType>application/*</CTType>.<VerCT></VerCT>
          </ Tx-Pref>]
          <SyncCap>
               <SyncType>1</ SyncType>
               <SyncT ype>2</SyncType>
               <SyncType>7</Sy ncType>
          </SyncCa p>
          </DataStore>
          </DevInf>
     </Data>
     </Item>
     </Put>
     <Get>
          <CmdID>3</ CmdID>
          <Meta>
          <Type xmlns='syncml :metinf'>application/vnd.syncml-
devinf+xml</Type >       </Meta>
          <Item>
          < Target>
          <LocURI>. /devinf12</LocUR I>
          </Target>
          </Item>
     </Get>
     <Final />
</SyncBody>]
</SyncML>.
```

Resposta do Servidor (SyncServer)

```
HT TP/1.1 200 OK..
Server: Apache-Coyote/1.1..
Set-Cookie: ..X-funambol-ds-server: DS  Server CarEd v.
9.0.0..
Accept-Encoding: gzip,def late
..
Uncompressed-Content-Length: 8512..
Content -Encoding: gzip..
Content-Type: application/vnd.syncml+xml..
Content-Length: 1239..
Date: Mon, 17 J an 2011  23:36:30  GMT..
Connection : close.
...
<?xml ve rsion="1.0" enco ding="UTF-8"?>
< SyncML>
<SyncHdr >
<VerDT D>1.2</V erDTD>.<VerProto >SyncML/1.2</Ver Proto>
<SessionI D>1295307389165< /SessionID>.<Msg ID>1</MsgID>
<Target>
     <LocURI>fb b-553648138</Loc
URI>.<LocName>camilla.gueiros</LocName>.</Target >
<Source>.<LocU RI>http://my.fun ambol.com/sync</
LocURI>.</Source >
<RespURI>http: //67.202.18.157/ sync;jsessionid=
E579E225DA4942EB 6E8AF9C2A739B853 </RespURI>
</Syn cHdr>
<SyncBody>
<Status>.
     <CmdID >1</CmdI>.<MsgR ef>1</MsgRef>.<CmdRef>0</CmdRef>
     <Cmd>SyncHdr</Cmd>
     <TargetRef>http://my.funambo l.com/sync</Targ etRef>
     <SourceRe f>fbb-553648138< /SourceRef>.<Data>212</Data>
</ Status>
<Status>
     <CmdID>2</CmdID> .<MsgRef>1</MsgR ef>.<CmdRef>1</C mdRef>
     <Cmd>Aler t</Cmd>
     <TargetR ef>configuration </TargetRef>
     <SourceRef>config</ SourceRef>
     <Data >200</Data>.
     <Item>
     <Data>.<Ancho rxmlns="syncml: metinf">
     <Next>1 295307389165</Next>.</Anchor>.
     </ Data>
     </Item>
</ Status>
<Status>
     <CmdID>3</CmdID >.<MsgRef>1</Msg Ref>.<CmdRef>2</ CmdRef>
     <Cmd>Put </Cmd>
     <SourceRef>./devinf12</SourceRef>
     <Data>2 00</Data>
</Stat us>
```

```
<Status>
     <CmdID>4</CmdID>.<M sgRef>1</MsgRef> .<CmdRef>3</CmdR ef>
     <Cmd>Get</Cm d>
     <TargetRef>./ devinf12</Target Ref>.<Data>200</
Data>.</Status>
     <Results>
     <CmdID >5</CmdID>.<MsgR ef>1</MsgRef>.<C mdRef>3< /CmdRef>
     <Meta>
          <Type xm lns='syncml:meti nf'>application/
vnd.syncml-devin f+xml</Type>
     </Meta>
     <Item>
     <Source>.<LocURI>./devinf12</LocURI> .</Source>
     <Data><DevInfxmlns= "syncml:devinf"> .<VerDTD>1.2</VerDTD>
     <Man>Funambol</Man>
     <Mod>DSServerCarEd</ Mod>
     <OEM>-</OEM >.<FwV>-</FwV>.<SwV>9.0.</SwV>. <HwV>-</HwV>
     <DevID>funambol</DevID>
     <DevTyp>ser ver</DevTyp>
     <UTC/>.<SupportLarg eObjs/>
     <Support NumberOfChanges/ >
<DataStore>
     <SourceRef>briefcase</SourceRef>
     < DisplayName>briefcase</DisplayNa me>.<MaxGUIDSize
>32</MaxGUIDSize >. <Rx-Pref>
          <CTT ype>application/ *</CTType>
          <VerC T>1.0</VerCT>
     </ Rx-Pref>
     <Tx-Pre f>.<CTType>application/*</CTType >
     <VerCT>1.0</Ve rCT>
     </Tx-Pref>
     <SyncCap>
          <SyncT ype>1</SyncType>
          <SyncType>2</Sy ncType>
          <SyncTyp e>3</SyncType>
          < SyncType>4</Sync Type>
          <SyncType> 5</SyncType>
          <Sy ncType>6</SyncTy pe>
          <SyncType>7</SyncType>
     </Syn cCap>
</DataStore>
<DataStore>
     < SourceRef>cal</SourceRef>
     <Displ ayName>cal</Disp layName><MaxGUI DSize>32</MaxGUI
DSize>
     <Rx-Pref>
          <CTType>text/x- vcalendar</CTTyp e>
          <VerCT>1.0</V erCT>
     </Rx-Pref>
     <Rx>.<CTType>text/calendar</CTT ype>
          <VerCT>2.0< /VerCT>
     </Rx>
     <T x-Pref>.<CTType> text/x-vcalendar </CTType>
```

```
          <VerCT >1.0</VerCT>
     </Tx-Pref>.<Tx>
     <CTType>text/calendar</CTType>.<Ver CT>2.0</VerCT>.< /Tx>.
     <SyncCap>
     < SyncType>1</Sync Type>
     <SyncType> 2</SyncType>
     <SyncType>3</SyncTy pe>
     <SyncType>4< /SyncType>
     <Sync Type>5</SyncType >
     <SyncType>6</SyncType>
     <SyncTy pe>7</SyncType>
     </SyncCap>
</Dat aStore>
<DataStore>
     <SourceRef>card</SourceRef>
     <DisplayName>card</DisplayName>. <MaxGUIDSize>32<
/MaxGUIDSize>
     <Rx-Pref>
     <CTType> text/x-vcard</CT Type>
     <VerCT>2.1 </VerCT></Rx-Pref>
     <Tx-Pref>.<CTType>text/x-vcard</CTType>.<Ver
CT>2.1</VerCT>.< /Tx-Pref>
     <SyncC ap>
     <SyncType>1< /SyncType>
     <Sync Type>2</SyncType >
     <SyncType>3</S yncType>
     <SyncTy pe>4</SyncType>.
     <SyncType>5</Syn cType>
     <SyncType >6</Sync Type>
     <S yncType> 7</SyncT ype>
     </SyncCap>.
</DataStore>
<DataStore>
     <Source Ref>configuration</SourceRef>
     <DisplayName>configuration</Displa yName>.
     <MaxGUIDS ize>32</ MaxGUID ize>
     <Rx-Pref>
          < CTType>text/plain</CTType>.<VerC T>1.0</VerCT>
     </ Rx-Pref>
     <Tx-Pre f>
          <CTType>text/ plain</CTType>
          < VerCT>1.0</VerCT >
     </Tx-Pref>
     <SyncCap>
     <SyncType >1</SyncType>
     <SyncType>2</SyncT ype>
     <SyncType>3 </SyncType>
     <Syn cType>4< /SyncTyp e>
     <SyncType>5</ SyncType>
     <SyncT ype>6</SyncType>
     <SyncType>7</SyncType>
     </SyncCap>
</DataStore>
```

```
<DataStore>
     <SourceRef>event</SourceRef>
     <DisplayName>event</DisplayName>.<MaxGU IDSize>32</MaxGU
IDSize>
     <Rx-Pref >.<CTType>text/x -vcalendar</CTType>.<VerCT>1.0</
VerCT>
     </Rx-Pref >
     <Rx>
          <CTType>t ext/calendar</CT Type>.<VerCT>2.0 </VerCT>
     </Rx>
     < Tx-Pref>
          <CTType>text/x-vcalenda r</CTType>.<VerC
T>1.0</VerCT>
     </Tx-Pref>.<Tx>
          <CTType>text/calendar</CTType><Ve rCT>2.0</VerCT>.
     </Tx>
     <SyncCap>
     <SyncType>1</Syn cType>
     <SyncType>2</SyncType>
     <SyncType>3</SyncT ype>
     <SyncType>4</SyncType>
     <SyncType>5</SyncType>
     <SyncType>6</SyncType>
     <SyncType>7</SyncType>
     </SyncCap>
</DataStore>
<DataStore>
     <SourceRef>mail</SourceRef>
     <DisplayName>mail</DisplayName><MaxGUIDSize>32
</MaxGUIDSize>
     < Rx-Pref>
          <CTType >application/vnd.omads-email+xml </CTType>
          <VerCT >1.2</VerCT>
     </Rx-Pref>
     <Rx>
     <CTType>application /vnd.omads-folde r+xml</CTType>
     < VerCT>1.2</VerCT >
     </Rx>
     <Tx-Pref >
          <CTType>application/vnd.omads- email+xml</CTTyp e>
          <VerCT>1.2</V erCT>
     </Tx-Pref>
     <Tx>
          <CTType>application/vnd.omads-folder+xml</ CTType>
          <VerCT>1 .2</VerCT>
     </Tx>
     <SupportHierarchicalSync/>
     <SyncCap>
          <SyncType> 1</SyncType>
          <SyncType>2</SyncTy pe>
          <SyncType>3< /SyncType>
          <Sync Type>4</SyncType >
          <SyncType>5</S yncType>
```

```
          <SyncTy pe>6</SyncType>
          <SyncType>7</Syn cType>
     </SyncCap >
</DataStore>
< DataStore>
     <SourceRef>note</SourceRef>
     <DisplayN ame>note</Displa yName>
     <MaxGUIDS ize>32</MaxGUIDS ize>
     <Rx-Pref>
          < CTType>text/plain</CTType>
          <VerC T>1.0</VerCT>
     </ Rx-Pref>
     <Tx-Pref>
          <CTType>text/plain</CTType>.< VerCT>1.0</VerCT >
     </Tx-Pref>
     <SyncCap>
     <SyncType >1</SyncType>
     <SyncType>2</SyncT ype>
     <SyncType>3 </SyncType>
     <Syn cType>4</SyncType>
     <SyncType>5</ SyncType>
     <SyncT ype>6</SyncType>
     <SyncType>7</SyncType>
     </SyncCa p>
</DataStore>
<DataStore>
     <SourceRef>picture</ SourceRef>
     <DisplayName>picture< /DisplayName>.<M
axGUIDSize>32</MaxGUIDSize>
     <Rx- Pref>
          <CTType>application/vnd.omads-file+xml</CTType>
          <VerCT>1.0 </VerCT>
     </Rx-Pref>
     <Tx-Pref>
          <CTType>application/vnd.omads-file +xml</CTType>
          <VerCT>1.0</VerCT>
</Tx-Pref>
<SyncCap>
     <SyncType> 1</SyncType>
     <SyncType>2</SyncTy pe>
     <SyncType>3< /SyncType>
     <Sync Type>4</SyncType >
     <SyncType>5</S yncType>
     <SyncTy pe>6</SyncType>
     <SyncType>7</SyncType>
     </SyncCap >
</DataStore>
< DataStore>
     <SourceRef>scal</Sour ceRef>
     <DisplayName>scal</DisplayName>
     <MaxGUIDSize>32</MaxGUIDS ize>
     <Rx-Pref>
          < CTType>text/x-s4 j-sife</CTType>
```

```
          <VerCT>1.0</VerC T>
     </Rx-Pref>
     <Tx-Pref>
          <CTType> text/x-s4j-sife< /CTType>
          <VerCT> 1.0</VerCT>
     </Tx -Pref>
     <SyncCap>
     <SyncType>1</Sy ncType>
     <SyncTyp e>2</SyncType>
     < SyncType>3</Sync Type>
     <SyncType> 4</SyncType>
     <SyncType>5</SyncTy pe>
     <SyncType>6< /SyncType>
     <Sync Type>7</SyncType >
     </SyncCap>.</D ataStore>
<DataS tore>
     <SourceRef >scard</SourceRe f>
     <DisplayName> scard</DisplayName>
     <MaxGUIDSize >32</MaxGUIDSize >
     <Rx-Pref>.<CTT ype>text/x-s4j-s ifc</CTType>.<Ve
rCT>1.0</VerCT>. </Rx-Pref>. <Tx-Pref>.<CTType>tex t/x-s4j-
sifc</CT Type>.<VerCT>1.0 </VerCT>.</Tx-Pr ef>.  <SyncCap>
          <S yncType>1</SyncT ype>
          <SyncType>2 </SyncType>
          <Syn cType>3</SyncTyp e>
          <SyncType>4</ SyncType>
          <SyncType>5</SyncType>
          <SyncType>6</SyncType>
          <SyncTyp e>7</SyncType>
     < /SyncCap>
</Data Store>
<DataStor e>
     <SourceRef>snote</SourceRef>
     <DisplayName>snote</DisplayName>
     <MaxGUIDSize>32 </MaxGUIDSize>
     <Rx-Pref>
          <CTType >text/x-s4j-sifn </CTType>.<VerCT
>1.0</VerCT>
     </R x-Pref>
     <Tx-Pref >
          <CTType>text/x -s4j-sifn</CTTyp e>.<VerCT>1.0</V
erCT>
     </Tx-Pref>
     <SyncCap>
     <Sync Type>1</SyncType >
     <SyncType>2</S yncType>
     <SyncTy pe>3</SyncType>
     <SyncType>4</Syn cType>
     <SyncType >5</SyncType>
     <SyncType>6</SyncT ype>
     <SyncType>7 </SyncType>
     </SyncCap>
</DataStore>
```

```
<DataStore>
     <SourceRef>stask </SourceRef>.<DisplayName>stask<
/DisplayName>
     <M axGUIDSize>32</M axGUIDSize>
     <Rx- Pref>.<CTType>te xt/x-s4j-sift</C TType>.<VerCT>1.
0</VerCT>.</Rx-P ref>.   <Tx-Pref>.< CTType>text/x-s4 j-
sift</CTType>. <VerCT>1.0</VerC T>.</Tx-Pref>
     <S yncCap>
     <SyncTyp e>1</SyncType>
     < SyncType>2</Sync Type>
     <SyncType> 3</SyncType>
     <SyncType>4</SyncType>
     <SyncType>5< /SyncType>
     <Sync Type>6</SyncType >
     <SyncType>7</S yncType>
     </SyncC ap>
</DataStore>
<DataStore>

<SourceRef>task</SourceRef>.<DisplayName>task</DisplayName>
     <MaxGUI DSize>32 </MaxGUI DSize>
     <Rx-Pref><CTType>text/x- vcalendar</CTTyp e>.<VerCT>1.0</V
erCT>.</Rx-Pref> .  <Rx>
          <CTType>text/calendar</CTT ype>
          <VerCT>2.0< /VerCT>
     </Rx>
     <T x-Pref>.<CTType> text/x-vcalendar </CTType>.<VerCT
>1.0</VerCT>.</T x-Pref>.     <Tx>
          <CTType>tex/calendar</CTType>.<Ver CT>2.0</VerCT>
     < /Tx>
     <SyncCap>
          < SyncType>1</Sync Type>
          <SyncType> 2</SyncType>
          <SyncType>3</SyncTy pe>
          <SyncType>4< /SyncType>
          <SyncType>5</SyncType >
          <SyncType>6</S yncType>
          <SyncType>7</SyncType>
     </SyncCap>
</DataStore>
<Ext>.<X Nam>X-funambol-s martslow</XNam></Ext>
<Ext>.<XN am>X-funambol-me dia-http-upload< /XNam></Ext>
<E xt>.<XNam>X-funambol-msu</XNam>. </Ext>
</DevInf>
</Data>
</Item>
</Results>
<Alert>
     <CmdID>6</Cm dID>
     <Data>200</ Data>
     <Item>
     <Target>.<LocURI>config</LocURI>.</Target>
     <Source> .<LocURI>configuration</LocURI></Source>
```

```
     <Meta> .<Anchorxmlns=' syncml:metinf'>
          <Last>1295307254 959</Last>.
          <Next >1295307390936</ Next>
     </Anchor>
     </Meta>
     </Item>
</Alert>
<Final/ >
</SyncBody>
</SyncML>.
```

POST Sincronia Inicial (SyncClient)

```
POST /sync jsessio nid=E579E225DA49 42EB6E8AF9C2A739B853
HTTP/1.1..
Host: 67.202.18.1 57..
Connection:  Close..
Content-T ype: application /vnd.syncml+xml..
Content-Length:  1151..
User-Agent: Funambol BlackBerry Plug-in 9 .0.1..
Device-Agent: BlackBerry85 20/5.0.0.509 MID P-2.1 CLDC-1.1..
Accept-Encoding:  gzip..
Content-L anguage:en-GB..
Cookie: JSESSION ID=E579E225DA494 2EB6E8AF9C2A739B 853..
..
<SyncML>
<SyncHdr>
     <VerDT D>1.2</VerDTD>.< VerProto>SyncML/ 1.2</VerProto>
     < SessionID>1295307389165</Session ID>.<MsgID>2</Ms gID>
     <Target>
     <LocURI><![CDATA[ht tp://67.202.18.1
57/sync;jsessionid=E579E225DA494       2EB6E8AF9C2A739B
853]]></LocURI>
     < /Target>
     <Source >
     <LocURI>fbb-553 648138</LocURI>< LocName>camilla.
gueiros</LocName >  </Source>
     <Meta ><MaxMsgSize>655 36</MaxMsgSize>< /Meta>
</SyncHdr >
<SyncBody>
     <St atus>
     <CmdID>1</ CmdID>.<MsgRef>1 </MsgRef>.<CmdRe f>0</CmdRef>
     <Cmd>SyncHdr</Cmd>
     <TargetRef>fbb-553648138</Target Ref>
     <SourceRef> http:
//my.funambol.com/sync</SourceRef>.Data>200</Data>
     </Status>
<Status>
     <Cmd ID>2</CmdID><MsgRef>1</MsgRef><CmdRef>6</CmdRef >
     <Cmd>Alert</Cmd >.<TargetRef>configuration</TargetRef>
```

```
     <SourceRef>config</Source Ref><Data>200</ Data>
     <Item>
     <Data><Anchor xmln s="syncml:metinf ">
     <Next>12953073 90936</Next>
     </Anchor>
     </Data>
     </ Item>
     </Status>
<Status>
     <CmdID>3 </CmdID>.<MsgRef>1</MsgRef>.<CmdRef>5</CmdRef>
     <Cmd>Results</Cmd >
     <SourceRef>./devinf12</SourceR
ef>.<Data>200</Data>.</Status>
     < Sync>
     <CmdID>4</CmdID>
     <Target>< LocURI>configura tion</LocURI></T arget>
     <Source>< LocURI>config</L ocURI></Source>
</Sync>
<Final/>
</SyncBody>
</SyncML>.
```

Resposta Servidor HTTP - OK

```
POST /sync;
jsessio nid=E579E225DA49 42EB6E8AF9C2A739B853
 HTTP/1.1..H ost: 67.202.18.1 57..
Connection:  Close..
Content-T ype: application /vnd.syncml+xml. .
Content-Length:  980..
User-Agent : Funambol Black Berry Plug-in 9. 0.1..
Device-Agent: BlackBerry852 0/5.0.0.509 MIDP -2.1 CLDC-1.1..
Accept-Encoding:  gzip..
Content-La nguage:
en-GB..
Cookie: JSESSIONID=E579E225DA4942 EB6E8AF9C2A739B8 53..
..
<SyncML>
< SyncHdr>
     <VerDTD >1.2</VerDTD>.<V erProto>SyncML/1 .2</VerProto>
     <SessionID>1295307 389165</SessionI D>
     <MsgID>4</Msg ID>
     <Target>
     <LocURI>
     <![CDATA[htt p://67.202.18.15 7/sync;jsessioni
d=E579E225DA4942EB  6E8AF9C2A739B8 53]]>
     </LocURI>
     </ Target>
     <Source>
     <LocURI>fbb-5536 48138</LocURI>
     <LocName>camilla.gueiros</LocName></Source>
```

```
</SyncHdr>
     <SyncBody>
     <Status>.<CmdID> 1</CmdID>.<MsgRe f>3</MsgRef>
     <Cm dRef>0</CmdRef>. <Cmd></Cmd>
     <TargetRef>fbb-55364 8138</TargetRef>
     <SourceRef>http://my.funambol.com/sync</SourceRef>
     <Data>200</Data>
     </Status>
     < Status>
     <CmdID>2 </CmdID><MsgRef >3</MsgRef>.<Cmd Ref>0</CmdRef>
     < Cmd>SyncHdr</Cmd >
     <TargetRef>fbb -553648138</TargetRef>
     <SourceRe f>http://67.202. 18.157/sync;jses
sionid=E579E225D    A4942EB6E8AF9C2A 739B853</SourceRef>
     <Data>200</Data>
     </Status>
     < Status>
     <CmdID>3< /CmdID>.<MsgRef> 3</MsgRef>.<CmdRef>2</CmdRef>
     <Cmd>Sync</Cmd>
     <SourceRef>configuration</SourceRe f>
     <TargetRef>config</TargetRef>
     <Data>200</Data >.</Status>
<Final/>
</SyncBody>
</SyncML>.
```

POST Calendar (SyncClient)

```
HTTP/1. 1..Host:my.funa mbol.com..
Connec tion: Close..
Content-Type: appli cation/vnd.syncm l+xml..
Content-L ength: 1544..
User-Agent:Funambol BlackBerry Plu g-in 9.0.1..
Device-Agent: BlackBerry8520/5.0.0.5 09 MIDP-2.1 CLDC -1.1..
Accept-Enc oding: gzip..
Content-Language: e n-GB..

<SyncML>
     <SyncHdr>
          <VerD TD>1.2</VerDTD>
          <VerProto>SyncML /1.2</VerProto>
          <SessionID>12953 17166034</SessionID>
          <MsgID>1</MsgID>
          <Target>
          <LocURI><![CDATA[h ttp://my.funambo l.com/sync]]></L
ocURI>
          </Target>
          <Source>
          <LocURI> fbb-553648138</L ocURI>
          <LocName>c amilla.gueiros</ LocName>
          </Source >
```

```
          <Cred>
          <Meta>< Type xmlns="sync ml:metinf">syncm l:auth-
basic</Type>
          <Format xmln s="syncml:metinf ">b64</Format>.< /Meta>
          <Data>Y2F taWxsYS5ndWVpcm9 zOnNpbmNyb25pYQ= =</Data>
          </Cred>
          <Meta><MaxMsgSiz e>65536</MaxMsgS ize></Meta>
     </SyncHdr>
     <SyncBody >
     <Alert>
          <CmdID >1</CmdID>.<Data >200</Data>.<Item>
          <Target><LocURI>event</LocURI >.</Target>
          <Source><LocURI>Calendar</LocURI></Source>
          <Meta><Anchor xmlns="syn cml:metinf">
          <Last>1295317132021 </Last>
          <Next>12 95317166034</Nex t>
          </Anchor>
          </M eta>
          </Item>
     </Alert>
     <Put>
     <Cmd ID>2</CmdID>
     <Me ta>
     <Type xmlns= 'syncml:metinf'> application/vnd. syncml-
devinf+xm l</Type>. </Meta>
     <Item>
          <Source>
          <LocURI>./devinf 12</LocURI>
          </Sou rce>
          <Data>
          <Dev Inf xmlns='syncm l:devinf'>.<VerD
TD>1.2</VerDTD>
          <Man>Research In  Motion</Man>.<M od>8520</Mod>.<O
EM></OEM>.          <FwV>< /FwV>.<SwV>9.0.1 </SwV>.<HwV>2.13 .
0.97</HwV>
          <Dev ID>fbb-553648138 </DevID><DevTyp
>phone</DevTyp>. <UTC/>.          <SupportNumberOfChanges/>
          <DataStore>
          <SourceRef>Calendar </SourceRef>
          <Rx -Pref>
          <CTType>t ext/x-vcalendar< /CTType>.<VerCT></VerCT>
          </Rx-Pr ef>
          <Tx-Pref>
               <CTType>text/x-vca lendar</CTType>
               <VerCT></VerCT>
          </Tx-Pref>
          <Sync Cap>
          <SyncType>1 </SyncType>
          <Syn cType>2</SyncType>
          <SyncType>7</ SyncType>
          </SyncCap>
          </DataStore >
          </DevInf>
```

```
          </Dat a>
     </Item>
</Put >
<Final />
</Sync Body>
</SyncML>.
```

Resposta Calendário

```
......HT TP/1.1 200 OK..
Server: Apache-Coyote/1.1.
X-funambol-ds-server:  DS Server CarEd  v.9.0.0..
Set-Coo kie:
Accept-En coding:
gzip,def late..
Uncompress ed-Content-Lengt h: 738..
Content- Encoding: gzip..
Content-Type: application/vnd.syncml+xml..
Conten t-Length: 349..
Date: Tue 18 Jan  2011 02:19:28 G MT..
Connection:  close...
                                                    .

<?xml version="1.0" enco ding="UTF-8"?>
< SyncML>
<SyncHdr >
     <VerDTD>1.2</V erDTD>.<VerProto >SyncML/1.2</Ver Proto>
     <SessionID>1295317166034< /SessionID><Msg ID>4</MsgID>
     <Target>
     <LocURI>fbb-553648138</LocURI>
     <LocName>camilla.gueiros</L ocName>
     </Target >
     <Source>
     <LocURI>http://67.202 .18.157/sync;jse ssionid=473FC577
4B42F032DF173400    12A4B7A6</LocURI >
     </Source>
     <RespURI>http://67.2 02.18.157/sync;j sessionid=473FC5
774B42F032DF1734    0012A4B7A6</Resp URI>
</SyncHdr>
<SyncBody>
     <Status>.<CmdID >1</CmdID>
     <MsgRef>1</MsgRef>.<CmdRef>0< /CmdRef>
     <Cmd>SyncHdr</C md>
     <TargetRef>h ttp://my.funambo l.com/sync</Targ etRef>
     <SourceRef>fbb-553648138< /SourceRef>
     <Data>212</Data>
     </Status>
     <Status>.
     <CmdID>2</CmdID>
     <MsgRef>1</MsgR ef>
     <CmdRef>1</CmdRef>
     <Cmd>Aler t</Cmd>
```

```
      <TargetRef>event</Target Ref>
      <SourceRef> Calendar</Source Ref>
      <Data>200</ Data>
      <Item>
      <Da ta><Anchor xmln s="syncml:metinf ">
      <Next>1295317 166034</Next>
      </ Anchor>
      </Data>
      </Item>
      </Status >
      <Status>
      <CmdI D>3</CmID>
      <Msg Ref>1</MsgRef>
      < CmdRef>2</CmdRef >
      <Cmd>Put</Cmd>
      <SourceRef>./de vinf12</SourceRe f>
      <Data>200</D ta>.</Status>
      <Alert>
      <CmdID>4</ CmdID>
      <Data>200 </Data>
      <Item>
      < Target>
      <LocURI> Calendar</LocURI >.</Target>
      <Source>.<LocURI>event</LocURI>.</So urce>
      <Meta>.<Anchor xmlns='sync ml:metinf'>
      <Last>1295317133693< /Last>
      <Next>129 5317167725</Next >
      </Anchor>
      </Me ta>
      </Item>
      </Alert>
      <Final/>
</ SyncBody>
</SyncML>.
```

Post Tarefas

Devi ce-Agent: BlackBerry8520/5.0.0.5 09 MIDP-2.1 CLDC -1.1..
Accept-Enc oding: gzip..
Content-Language: e n-GB..
..

```
<SyncML>
<SyncHdr>
      <VerD TD>1.2</VerDTD>. <VerProto>SyncML /1.2</VerProto>
      <SessionID>1295320512515</Sessio nID>
      <MsgID>1</M sgID>
      <Target><L ocURI><![CDATA[h ttp://my.funambo l.com/sync]]></L ocURI>
      </Target>
      <Source>
      <LocURI> fbb-553648138</L ocURI>
```

<LocName>c amilla.gueiros</ LocName>
</Source >
<Cred>
<Meta>
< Type xmlns="sync ml:metinf">syncm l:auth-basic</Type>
<Format xmln s="syncml:metinf ">b64</Format>
< /Meta>
<Data>Y2FtaWxsYS5 ndWVpcm9zOnNpbmNyb25pYQ= =</Data></Cred>
<Meta><MaxMsgSiz e>65536</MaxMsgS ize></Meta>
</SyncHdr>
<SyncBody >
<Alert>
<CmdID >1</CmdID>.<Data >200</Data>
<Item>
<Target>
<LocURI>task</LocURI>
</Target>
<Source>
<LocURI>Tasks </LocURI>
</Sourc e>
<Meta>
<Ancho rxmlns="syncml: metinf">.<Next>1 295320512515</Next></Anchor>
</ Meta>
</Item>
</ Alert>
<Put>
<Cm dID>2</CmdID>
<Meta>
<Type xmlns ='syncml:metinf' >application/vnd .syncml-devinf+x ml</Type>
</Meta >
<Item>
<Source ><LocURi>./devin f12</LocURI></Source>
<Data>
<De vInf xmlns='syncml:devinf'>.<Ver DTD>1.2</VerDTD>
<Man>Research I n Motion</Man>
< Mod>8520</Mod>.< OEM></OEM>.<FwV></FwV>.<SwV>9.0. 1</SwV>
<HwV>2.1 3.0.97</HwV>
<DevID>fbb-553648138</DevID>
<DevTy p>phone</DevTyp>
<UTC/>.<Support NumberOfChanges/ >
<DataStore>
<S ourceRef>Tasks</ SourceRef>
<Rx-P ref>
<CTType>tex t/x-vcalendar</C TType>
<VerCT></ VerCT>
</Rx-Pref >
<Tx-Pref>
<CTT ype>text/x-vcale ndar</CTType>
<V erCT></VerCT>

</ Tx-Pref>
<SyncCa p>
<SyncType>1</ SyncType>.<SyncT ype>2</SyncType> .<SyncType>7</Sy ncType>.
</SyncCa p>
</DataStore>.
</DevInf>
</Data>
</Item>
</Put>
<Final/>
</SyncBo dy>
</SyncML>.

Http OK Tarefas

```
Server: Apache-Co yote/1.1..
Set-Cookie: .. X-funamb ol-ds-server: DS  Server CarEd v.
9.0.0.
Accept-En coding: gzip,def late..
Uncompressed-Content-Lengt h: 1316..
Content -Encoding: gzip. .
Content-Type: application/vnd.syncml+xml..
Content-Length: 528..
Date: Tue, 18 Ja n 2011 03:15:17
GMT..Connection:  close..

<?xml ve rsion="1.0" enco ding="UTF-8"?>
< SyncML>
     <SyncHdr >
     <VerDTD>1.2</V erDTD>
     <VerProto >SyncML/1.2</VerProto>
     <SessionI D>1295320512515< /SessionID>
     <MsgID>1</MsgID>
     <Target>
     <LocURI>fbb-553648138</LocURI>
     <LocName>camilla.gueiros</LocName>
     </Target >
     <Source>.<LocURI>http://my.fun ambol.com/sync</
LocURI>.</Source >. <RespURI>
     http: //67.202.18.157/ sync;jsessionid=
5C10B1C366DA357CE006E0E404E7CB2C  </RespURI>
     </Syn cHdr>
     <SyncBody>
     <Status>
          <CmdID >1</CmdID>
          <MsgR ef>1</MsgRef>
          <CmdRef>0</CmdRef>
          <Cmd>SyncHdr</C md>
          <TargetRef>h ttp://my.funambo l.com/sync</TargetRef>
          <SourceRe f>fbb-553648138< /SourceRef>
          <Data>212</Data>
```

```
     </Status>
     <Status>
          <CmdID>2</CmdID>
          <MsgRef>1</MsgR ef>
          <CmdRef>1</C mdRef>
          <Cmd>Alert</Cmd>
          <TargetRef>task</TargetRef>
          <SourceRef>Tasks</SourceRef>
          <Data>508</Data >
          <Item>.<Data>.
          <Anchor xmlns="s yncml:metinf">
               < Next>12953205125 15</Next>
          </Anch or>
          </Data>
          </Item>
     </Status>
     <Status>
          <CmdID>3< /CmdID>
          <MsgRef> 1</MsgRef>
          <CmdR ef>2</CmdRef>
          <Cmd>Put</Cmd>
          <SourceRef>./devinf 12</SourceRef>
          < Data>200</Data>
     </Status>
     <Alert >
     <CmdID>4</CmdI D>
     <Data>201</Da ta>
     <Item>
     <Target>.<LocURI>Tasks</LocURI>.</Target>
     <Source>.<LocURI>task</LocURI>.</Source>
     <Meta>.<Anchor xml ns='syncml:metin f'>
     <Last>129532 0517182</Last>.
     < Next>12953205171 82</Next>
     </Anchor>
     </Meta>
     </It em>
     </Alert>
     <Fi nal/>
     </SyncBody >
</SyncML>.
```

Segundo Post

```
POST/sync;jsessionid=5C10B1C366DA357CE006E0E404E7CB2C
HTTP/1.1..
Host: 67.202.18.157..
Connection:  Close..
Content-Type: application /vnd.syncml+xml..
Content-Length:  1380..
User-Agent: Funambol BlackBerry Plug-in 9 .0.1..
Device-Agent: BlackBerry8520/5.0.0.509 MID P-2.1 CLDC-1.1..
Accept-Encoding:  gzip..
```

```
Content-Language:en-GB..
Cookie: JSESSION ID=5C10B1C366DA357CE006E0E404E7CB2C..
..
<SyncML>
     <SyncHdr>
     <VerDT D>1.2</VerDTD>.< VerProto>SyncML/ 1.2</VerProto>
     < SessionID>129532 0512515</Session ID>
     <MsgID>2</MsgID>
     <Target>
     <LocURI><![CDATA[http://67.202.18.157/sync;jsessionid=
     5C10B1C366DA357CE006E0E404E7CB2C]]></LocURI>
     </Target>
     <Source><LocURI>fbb-553 648138</LocURI>
     <LocName>camilla.gueiros</LocName></Source>.
     <Meta ><MaxMsgSize>65536</MaxMsgSize>< /Meta>
     </SyncHdr>
     <SyncBody>
     <Status>
     <CmdID>1</CmdID>.<MsgRef>1 </MsgRef>.<CmdRe f>0</CmdRef>
     <Cmd>SyncHdr</Cmd><TargetRef>fbb-553648138</TargetRef>

<SourceRef>http://my.funambol.com/sync</SourceRef>.<Data>200</
Data>.    </Status>
     <Status>.<CmdID>2</CmdID>.<Ms gRef>1</MsgRef><
CmdRef>4</CmdRef >
     <Cmd>Alert</Cmd>
     <TargetRef>task</TargetRef>
     <SourceRef>Tasks</SourceRef>
     <Data>200</Data>
     <Item>

<Data>.<Anchorxmlns="syncml:metinf"><Next>1295320517182</Next>
     </Anchor>]
     </Data>
     </Item>
     </Status>
     <Sync>
     <CmdID>3</CmdID>
     <Target><LocURI>task</LocURI></Target>
     <Source><LocURI>Tasks</LocURI></Source>
     <Replace>
     <CmdID>4</CmdID>
     <Item>
     <Meta><Type xmlns="syncml:metinf ">text/x-
vcalendar</Type></Meta>
     <Source><LocURI>330021558</LocURI></Source
     <Data>BEGIN:VCALENDAR..
          VERSION:1.0.
          BEGIN:VTODO..
          SUMMARY:TesteTarefa..
          DUE:20110119T235900..
          DESCRIPTION:..
          PERCENT-COMPLETE:0..
```

```
          STATUS:NEEDSACTION..
          UID:330021558..
          PRIORITY:5..
          RRULE :..
          AALARM:;  ;0;..
          END:VTODO..
          END:VCALENDAR..
     </Data >
     </Item>
     </Replace>
     </Sync>
     <Fi nal/>
</SyncBody>
</SyncML>.
```

Http OK

```
HTTP/1.1 00 OK.
.Server:  Apache-Coyote/1 .1..
X-funambol-d s-server: DS Ser ver CarE d v.9.0. 0..
Set-Cookie: .
.Accept-Encoding : gzip,deflate..
Uncompressed-Con tent-Length: 103 6..
Content-Encod ing: gzip..
Conte nt-Type:applica tion/vndsyncml+ xml..
Content-Len gth: 396..
Date:  Tue, 18 Jan 2011  03:15:19 GMT..
Connection: close

<?xml ve rsion="1.0" enco ding="UTF-8"?>
< SyncML>
<SyncHdr >
     <VerDTD>1.2</V erDTD>.<VerProto >SyncML/1.2</Ver Proto>
     <SessionID>1295320512515< /SessionID>.<MsgID>2</MsgID>
     <Target>.<LocURI>fbb-553648138</LocURI>
     <LocName>camilla.gueiros</L ocName>.</Target >
     <Source>
     <LocURI>http://67.202 .18.157/sync;jse ssionid=5C10B1C3
     66DA357CE006E0E404E7CB2C</LocURI >
     </Source>
     <RespURI>http://67.2 02.18.157/sync;j sessionid=5C10B1
     C366DA357CE006E0E404E7CB2C</RespURI>
</SyncHdr>
     <SyncBody>
     <Status>.<CmdID>1</CmdID>
     <MsgRef>2</ MsgRef>.<CmdRef> 0</CmdRef>.<Cmd>
SyncHdr</Cmd>
     <TargetRef>http:// 67.202.18.157/sy nc;jsessionid=5C
     10B1C366DA357CE0    06E0E404E7CB2C</ TargetRef>
     <SourceRef>fbb-553648138</SourceRef>
     <Data>200</Data> .</Status>
     <Status>.<CmdID>2</CmdID>
```

```
     <MsgRef>2</ MsgRef>.<CmdRef> 3</CmdRef>
     <Cmd> Sync</Cmd>
     <TargetRef>task</TargetRef>
     <SourceRef>Tasks</SourceR ef>.<Data>200</Data>
     </Status>
     < Status>
     <CmdID>3 </CmdID>
     <MsgRef >2</MsgRef>
     <Cmd Ref>4</CmdRef>
     < Cmd>Replace</Cmd >
     <SourceRef>330021558</SourceRe f>
     <Data>201</Data>
     </Status>
</ SyncBody>
</Sync ML>.
```

Post Contatos

```
...{..PO ST /sync;jsessio nid=A2F452931B7E B7263FA22742FF35
7CA3
HTTP/1.1..
Host: 174.129.171 .70..
Connection:  Close..
Content- Type: applicatio n/vnd.syncml+xml..
Content-Length : 1985..
User-Agent: Funambol Bla ckBerry
Plug-in  9.0.1..
Device-Agent: BlackBerry8 520/5.0.0.509 MI DP-2.1 CLDC-1.1. .
AcceptEncoding : gzip..
Content- Language: en-GB. .
Cookie:JSESSIO NID=A2F452931B7E B7263FA22742FF35 7CA3..
..
<SyncML>
     <SyncHdr>
     <VerD TD>1.2</VerDTD>. <VerProto>SyncML /1.2</VerProto>
     <SessionID>12955 00100565</Sessio nID>.<MsgID>2</M sgID>
     <Target><L ocURI><![CDATA[h ttp://174.129.17
1.70/sync;jsessi onid=A2F452931B7 EB7263FA22742FF3
57CA3]]></LocURI ></Target>
     <Sour ce<LocURI>fbb-5 53648138</LocURI >
     <LocName>camill a.gueiros</LocNa me></Source>.
     <Meta><MaxMsgSize>6 5536</MaxMsgSize ></Meta>
     </SyncHdr>
<SyncBody>
     < Status>
     <CmdID>1 </CmdID>.<MsgRef >1</MsgRef>.<Cmd Ref>0</CmdRef>
     < Cmd>SyncHdr</Cmd >
     <TargetRef>fbb -553648138</TargetRef>
     <SourceRe f>http://my.funa mbol.com/sync</S ourceRef>
     <Data> 200</Data>
     </Sta tus>
     <Status>
```

```
      <C mdID>2</CmdID>.< MsgRef>1</MsgRef ><CmdRef>4</CmdR ef>
      <Cmd>Alert</C md>
      <TargetRef>card</TargetRef>
      <SourceRef>Contacts</SourceRef>
      <Data>200</Data>
      <Item>
           <Data>
           < Anchor xmlns="sy ncml:metinf">
           <Ne xt>1295500105360 </Next></Anchor>
           </Data>
      </Item>
      </Status>
      <Sync >
      <CmdID>3</CmdI D>
      <Target><LocURI>card</LocURI></Target>
      <Sourc e><LocURI>Contacts</LocURI>
      </Source>
      <Add>
      <CmdI D>4</CmdID>
      <Item>
      <Meta>.<Type  xmlns="syncml:me tinf">text/x-
vcard</Type></Meta>
      <Source><LocURI >122159960</LocURI></Source>
      <Data>
      BEGIN:VCARD..
      VERSION:2.1..N:; Ariel Soares;;;..
      TEL;VOICE;HOME: ..
      TEL;VOICE;HOME:..
      TEL;VOICE;WOR K:..
      TEL;VOICE;WO RK:..
      TEL;PAGER:..
      TEL;FAX;WORK:..
      TEL;CELL:23467589..
      TEL;VOICE:..
      A DR;HOME:;;;;;;..
      ADR;WORK:;;;;;;. .
      ADR:;;;;;;..
      BDA Y:..
      EMAIL;
      INTERN ET:..
      EMAIL;INTER NET;HOME:..
      EMAIL ;
      INTERNET;WORK:. .
      TITLE:..URL:..O RG:..NOTE:..
      PHOTO:..CATEGORIES:..
      END:VCARD..
      </Data>
      </Item>
      <Item>
      <Meta>
      <Type  xmlns="syncml:me tinf">text/x-vcard</Type></Meta>
      <Source><LocURI >122159962</LocU RI></Source>
      <Data>
```

```
     BEGIN:VCARD..
     VERSION:2.1..N:;
     Joao Santana;;;. .
     TEL;VOICE;HOME: ..
     TEL;VOICE;HOME :..
     TEL;VOICE;WOR K:..
     TEL;VOICE;WORK:..
     TEL;PAGER:. .
     TEL;FAX;WORK:..
     TEL;CELL:8765342 1..
     TEL;VOICE:..
     A DR;HOME:;;;;;;..
      ADR;WORK:;;;;;;. .
     ADR:;;;;;;..BDA Y:..
     EMAIL;INTERN ET:..
     EMAIL;
     INTER NET;HOME:..
     EMAIL ;INTERNET;WORK:. .
     TITLE:..URL:..O RG:..NOTE:..PHOTO:..
     CATEGORIES:. .
     END:VCARD..
     </Da ta>.</Item>
     </Add>
     </Sync>
     <Final/>
</SyncBody>
< /SyncML>
```

RespostaPost Contatos

```
<?xml version="1.0" enco ding="UTF-8"?>
< SyncML>
     <SyncHdr >
     <VerDTD>1.2</V erDTD>.<VerProto >SyncML/1.2</Ver Proto>
     <SessionID>1295500100565< /SessionID>
     <Msg ID>2</MsgID>
     <Target>.<LocURI>fbb-553648138</Loc URI>
     <LocName>camilla.gueiros</L ocName>.</Target >
     <Source>
     <LocURI>http://174.12 9.171.70/sync;jsessionid=A2F4529
31B7EB7263FA2274    2FF357CA3</LocURI>
     </Source>
     <RespURI>http://174 .129.171.70/sync ;jsessionid=A2F4
52931B7EB7263FA2    2742FF357CA3</RespURI>
     </SyncHdr >
<SyncBody>
     <Status>
     <CmdID>1</ CmdID>.<MsgRef>2 </MsgRef>.<CmdRe f>0</CmdRef>
     <Cmd>SyncHdr</Cmd>
     <TargetRef>http: //174.129.171.70 /sync;jsessionid
=A2F452931B7EB72    63FA22742FF357CA 3</TargetRef>
     <SourceRef>fbb-553648138</SourceRe f>
```

```
     <Data>200</Data>
     </Status>
     <Status>
     <CmdID>2< /CmdID>.<MsgRef> 2</MsgRef>.<CmdRef>3</CmdRef>
     <C md>Sync</Cmd>
     <T argetRef>card</T argetRef>
     <Sourc eRef>Contacts</S ourceRef>
     <Data> 200</Data>.</Status>
     <Status>.<C mdID>3</CmdID>
     <MsgRef>2</MsgRef ><CmdRef>4</Cmd Ref>
     <Cmd>Add</Cmd>
     <Data>201</D ata>
     <Item>.<Source>.<LocURI>122 159960</LocURI>
     </Source>
     </Item >
     <Item>
     <Source >.<LocURI>122159 962</LocURI>
     </Source>
     </Item>
     < /Status>
     </SyncB ody>
</SyncML>.
```

Servidor Nicolas Bougues

HTTP/1.1 200 OK.. Post Inicial

```
Date: Sat, 21 Jan  2011 21:33:33 G MT..
Server: Apache/2.2.9(Win32)  DAV/2 mod_ssl/2 .2.9
OpenSSL/0.9 .8i mod_autoinde x_color PHP/5.2. 6..
X-Powered-By:  PHP/5.2.6..Set- Cookie: PHPSESSI
D=3ff3cf2399729d7874ccaa117bbf72 08; path=/..
Expires: Thu, 19 Nov  1981 08:52:00 G MT..
Pragm: no-ca che..
Accept-Char set: UTF-8..
Connection: close..
T ransfer-Encoding : chunked..
Conte nt-Type: applica tion/vnd.syncml+ xml....201c..
<? xml version="1.0 " encoding="UTF- 8">
<SyncML  xmlns='SYNCML:S YNCML1.0'>
     <Sy ncHdr>
     <VerD TD>1.2</VerDTD><VerProto>SyncML/1.2</VerPro to>
     <Session ID>1289079212859 </SessionID>
     <MsgID>1</MsgID >
     <Target><LocURI>fbb -553648138</LocURI></Target >
     <Source>
     <LocURI>htt p://sincronia.no
-iporg:8080/xampp/syncml/sync.php</LocURI>.      </Source>
     </SyncHdr>
     <SyncBo dy>
     <Status>
```

```
     <CmdID>1< /CmdID>
     <M sgRef>1< /MsgRef>
     <CmdRef>0 </CmdRef>
     <Cmd>SyncHdr</Cmd>
     <TargetRef>http://sincronia.no-ip.org:8
080/xampp/syncml /sync.php</Targe tRef>
     <Sou rceRef>fbb-55364 8138</SourceRef>
     <Data>212 </Data>
     </St atus>
     <Statu s>
     <CmdID> 2</CmdID>
     <MsgRef>1</MsgRe f>
     <CmdRef >1</CmdRef>
     <Cmd>Alert</Cmd>
     <Target Ref>configuration</TargetRef>
     <SourceRef>c onfig</SourceRef >
     <Data>200</Data>
     </Status>
     <Stat us>
     <CmdID >3</CmdID>
     <MsgRef >1</MsgRef>
     <CmdRef>2< /CmdRef>
< Cmd>Put</Cmd>
<SourceRef>. /devinf12</Sourc eRef><Dat a>200</Data>
</Status>
< Status>
     <C mdID>4</CmdID>
     <MsgRef>1</ MsgRef>
     <C mdRef>3</CmdRef>
     <Cmd>Get< /Cmd>
     <Tar getRef>./devinf1 2</TargetRef>
     <Dat a>200</D ata>
</Statu s>
     < Results>
     <CmdID>5< /CmdID>
     <M sgRef>1</MsgRef>
     <CmdRef>3 </CmdRef>
     <Meta>
< Type xmlns='sync ml:metinf'>application/vnd.syncm l-
devinf+xml</Ty pe>.     </Meta >
     <Item>
     <Source>
     <LocUR I>./devinf12</LocURI>
     </ Source>
     <Data><DevInf xmlns=' syncml:devinf'>
     <Ver DTD>1.2<  /VerDTD>
     <Man>Nicolas Bougue s</Man>
     <Mod >SyncML/ PHP serv er</Mod>
     <DevID>12345</DevID >
     <DevTyp>server</De vTyp>
     <DataStore>
     <Sou rceRef>Notes</SourceRef>
<MaxGUIDS ize>4</MaxGUIDSi ze>.
```

```
<Rx-Pref>
     <CTT ype>text /x-vnote </CTType   >
<VerCT >1.1</VerCT>
     </Rx- Pref>
     <Tx-   Pref>
     <C TType>text/x-vnote</CTType>
     <Ver CT>1.1</  VerCT>
</T x-Pref>
<SyncCap>
<SyncType>1</Syn cType>
<SyncType >2</Sync   Type>
</SyncCap>
</DataStore>
     <CTCap>
     <CTTyp e>text/x -vnote</   CTType>
<PropName>   BEGIN</P ropName>
     <ValEnum>VCARD</ValEnum>
     <PropName>END</PropName>
     <ValEnum>VCARD</ValEnum>
     <PropName>VERSION</PropName>
     <ValEnum>2.1</ValEnum>
     <PropName>N</PropName>
     <PropName>TEL</PropName>
     <ParamName>VOICE</ParamName>
     <ParamName>CELL</ParamName>
     <CTType>text/vcard</CTType>
     <PropName>BEGIN</PropName>
     <ValEnum>VCARD</ValEnum>
     <PropName>END</PropName>
     <ValEnum>VCARD</ValEnum>
     <PropName>VERSION</PropName>
     <ValEnum>3.0</ValEnum>
     <PropName>N</PropName>
     <PropName>TEL</PropName>
     <ParamName>VOICE</ParamName>
     <ParamName>FAX</ParamName>
     <ParamName>CELL</ParamName>
     <CTType>text/vcard</CTType>
     <PropName>BEGIN</PropName>
     <ValEnum>VCARD</ValEnum>
     <PropName>END</PropName>
     <ValEnum>VCARD</ValEnum>
     <PropName>VERSION</PropName>
     <ValEnum>3.0</ValEnum>
     <PropName>N</PropName>
     <PropName>TEL</PropName>
     <ParamName>VOICE</ParamName>
     <ParamName>FAX</ParamName>
     <ParamName>CELL</ParamName>
     </CTCap>
     <SyncCap>
     <SyncType>01</SyncType>
     <SyncType>02</SyncType>
```

```
     </SyncCap>
     </DevInf>
     </Data>
     </Item>
     </Results>
     <Alert>
     <CmdID>5</CmdID>
     <Data>201</Data><!-- 201 = TWO_WAY_ALERT -->
     <Item>
     <Target><LocURI>./dev-contacts</LocURI></Target>
     <Source><LocURI>./contacts</LocURI></Source>
     <Meta>
     <Anchor xmlns='syncml:metinf'>
     <Last>20110010217081812 </Last>
     <Next>2001101022909322 </Next>
     </Anchor>
     </Meta>
     </Item>
     </Alert>
     <Final/>
</SyncBody>
</SyncML>
```

HTTP OK - Contatos

```
Server: Apache/2.2.9(Win32)  DAV/2 mod_ssl/2 .2.9
OpenSSL/0.9 .8i mod_autoinde x_color PHP/5.2. 6..
X-Powered-By:  PHP/5.2.6..Set- Cookie: PHPSESSI
D=3ff3cf2399729d7874ccaa117bbf72 08; path=/..
Expires: Thu, 19 Nov  1981 08:52:00 G MT..
Pragm: no-ca che..
Accept-Char set: UTF-8..
Connection: close..
Content-Type: application/vnd.syncml+xml....201c..

<? xml version="1.0 " encoding="UTF- 8">
<SyncML  xmlns='SYNCML:S YNCML1.0'>
     <Sy ncHdr>
     <VerD TD>1.2</VerDTD><VerProto>SyncML/1.2</VerPro to>
     <Session ID>1289079212859 </SessionID>
     <MsgID>1</MsgID >
     <Target><LocURI>fbb -553648138</LocURI></Target >
     <Source>
     <LocURI>http://sincronia.no
-ip.org:8080/xampp/syncml/sync.php</LocURI>.     </Source>
     </SyncHdr>
<SyncBo dy>
     <CmdID>1< /CmdID>
     <MsgRef>1< /MsgRef>
     <CmdRef>0 </CmdRef>
     <Cmd>SyncHdr</Cmd>
```

```
     <TargetRef>http://sincronia.no-ip.org:8
080/xampp/syncml /sync.php</Targe tRef>
     <SourceRef>fbb-55364 8138</SourceRef>
     <Data>212 </Data>
     <Status>
     <CmdID> 2 </CmdID>
     <MsgRef>2</MsgRef><CmdRef>3</CmdRef><Cmd>Sync</Cmd>
     <TargetRef>card</TargetRef>
     <SourceRef>Contacts</SourceRef>
     <Data>200</Data>
     </Status>
     <Status>
     <CmdID>3</CmdID>
     <MsgRef>3<MdgRef><CmdRef>4<CmdRef><Cmd>Replace</Cmd>
     <Data>200</Data>
     <Sync>
     <Target><LocURI>./dev-contacts</LocURI></Target>
     <Source><LocURI>./contacts</LocURI></Source>
     <Replace>
     <CmdID>5</CmdID>
     <Meta><Type xmlns='syncml:metinf'>text/x-
vcard</type></Meta>
     <Item>
     <Target>>card</Target>
     </Item>
     </Status>
     <Sync>
<CmdID>4</CmdID>
<Target><LocURI>./dev-contacts</LocURI></Target>
<Source><LocURI>./contacts</LocURI></Source>
<Replace>
<CmdID>5</CmdID>
<Meta><Type xmlns='syncml:metinf'>text/x-vcard</type></Meta>
<Item>
<Target>card</Target>
<Data></Data>
</Item>
</Replace>
<Add>
<CmdID>6</CmdID>
<Meta><Type xmlns='syncml:metinf'>text/x-vcard</type></Meta>
<Item>
<Source><LocURI>10536681</LocURI></Source>
<Data>
VERSION:2.1..N:; CamillaGueiros;;;..
     TEL;VOICE;HOME: ..
     TEL;VOICE;HOME:..
     TEL;VOICE;WOR K:..
     TEL;VOICE;WO RK:..
     TEL;PAGER:..
     TEL;FAX;WORK:..
     TEL;CELL:32414298..
     TEL;VOICE:..
```

```
     A DR;HOME:;;;;;;..
     ADR;WORK:;;;;;;. .
     ADR:;;;;;;..
     BDA Y:..
     EMAIL;
     INTERN ET:..
     EMAIL;INTER NET;HOME:..
     EMAIL ;
     INTERNET;WORK:. .
     TITLE:..URL:..O RG:..NOTE:..
     PHOTO:..CATEGORIES:..
     END:VCARD
</Data>
</Status>
</Item>
</Add>
</Sync>
<Final/>
</SyncBody>
</SyncML>
```

Http Ok Vnotes

```
Server: Apache/2.2.9(Win32)  DAV/2 mod_ssl/2 .2.9
OpenSSL/0.9 .8i mod_autoinde x_color PHP/5.2. 6..
X-Powered-By:  PHP/5.2.6..Set- Cookie: PHPSESSI
D=3ff3cf2399729d7874ccaa117bbf72 08; path=/..
Expires: Thu, 19 Nov  1981 08:52:00 G MT..
Pragm: no-ca che..
Accept-Char set: UTF-8..
Connection: close..
T ransfer-Encoding : chunked..
Conte nt-Type: applica tion/vnd.syncml+ xml....201c..
<? xml version="1.0 " encoding="UTF- 8">


<SyncML  xmlns='SYNCML:S YNCML1.0'>
     <Sy ncHdr>
     <VerD TD>1.2</VerDTD><VerProto>SyncML/1.2</VerPro to>
     <Session ID>1289079212859 </SessionID>
     <MsgID>1</MsgID >
     <Target><LocURI>fbb -553648138</LocURI></Target >
     <Source>
     <LocURI>http://sincronia.no
-ip.org:8080/xampp/syncml/sync.php</LocURI>.     </Source>
     </SyncHdr>
<SyncBo dy>
     <Status>
     <CmdID>1</CmdID>.<M sgRef>1</MsgRef>
     <CmdRef>0</CmdR ef>
     <Cmd>SyncHdr </Cmd>
     <TargetRe f>fbb-553648138< /TargetRef>
```

```
     <SourceRef>http://sincronia.no-
ip.org:8080/syncml/sync</SourceRef>
     < Data>200</Data>
     </Status>
     <Status><CmdID>2</Cmd ID>.<MsgRef>1</MsgRef><CmdRef>4<
/CmdRef>
     <Cmd>Alert</Cmd>
     <TargetRef>vnote</Targe tRef>
     <SourceRef >notes</SourceRe f>
     <Data>200</Da ta>
     <Item>.<Data >.<Anchorxmlns= "syncml:metinf">
     <Next>1295589096379</Next>
     </Anchor>
     </Data>
     </It em>
     </Status>
     <Sync>
     <CmdID>3</C mdID>
     <Target><L ocURI>notes</Loc URI></Target>
     <Source><LocURI>vNotes</LocURI></Source>
     <Replace>. <CmdID>4</CmdID>
     <Item>.<Meta>.< Type xmlns="sync
ml:metinf">text/vnote</Type > <Formatxmlns=' syncml:metinf'>b
64</Format>.</Me ta>
     <Source><LocURI>1044048115</ LocURI>
     </Source>
     <Data>PG5vdGU+C jxTdWJqZWN0PlNpbmNyb25pYU5vdGU8L
     1N1YmplY3Q+CjxCb    2R5PkVzdGUgZSB1bWEgQW5vdGFhbyBUZ
     XN0ZSEKRGF0YSBJbljaW8KRGF0YSBQcmV2aXNhbwpEYXRhI
     FRlcm1pbm88L0JvZ Hk+CjxDYXRlZ29yaWVzPjwvQ2F0ZWdvc
     mllcz4KPENvbG9yPjwvQ29sb3I+CjxIZWlnaHQ+PC9IZWlna
     HQ+CjxXaWR0aD48L    1dpZHRoPgo8TGVmdD48L0xlZnQ+CjxUb
     3A+PC9Ub3A+Cjwvb m90ZT4K</Data>
     < /Item>
     </Replace >
     </Sync>
     <Final />
     </SyncBody>
</ SyncML>.
```

Post SyncClient (Modificações)

```
Server: Apache/2.2.9(Win32)  DAV/2 mod_ssl/2 .2.9
OpenSSL/0.9 .8i mod_autoinde x_color PHP/5.2. 6..
X-Powered-By:  PHP/5.2.6..Set- Cookie:
PHPSESSID=3ff3cf2399729d7874ccaa117bbf72 08; path=/..
Expires: Thu, 19 Nov  1981 08:52:00 G MT..
Pragm: no-ca che..
Accept-Char set: UTF-8..
Connection: close..
T ransfer-Encoding : chunked..
Conte nt-Type: applica tion/vnd.syncml+ xml....201c..
```

```
<? xml version="1.0 " encoding="UTF- 8">


<SyncML>
     <SyncHdr> .<VerDTD>1.2</Ve rDTD>.<VerProto>
SyncML/1.2</VerProto>
     <SessionID >1295658119855</ SessionID>.<MsgI
D>2</MsgID>
     <Target><LocURI><![CDATA[http://sync
.memotoo.com/sync?
sid=c593cc8cc4b9e7dcff102c9ccfdcfba7]]></LocURI></Target>
     <Source><LocURI>fbb- 553648138</LocUR
I><LocName>sincronia</LocName></ Source>
     <Meta><MaxMsgSize>65536< /MaxMsgSize>
     </Meta>
</SyncHdr>
     < SyncBody>
     <Statu s>
     <CmdID>1</CmdID>.<MsgRef>1</MsgRef>
     <CmdRef>0 </CmdRef>
     <Cmd>SyncHdr</Cmd>
     <TargetRef>fbb-5536 48138</TargetRef>
     <SourceRef>http://sincronia.no-
ip.org:8080/sync</SourceRef>
     <Data>200< /Data>.</Status> .<Status>.<CmdID
>2</CmdID>
     <MsgRef>1</MsgRef><CmdRef>4</CmdRef>
     < Cmd>Alert</Cmd>
     <TargetRef> calendars </TargetRef>
     <SourceRef>Calendar </SourceRef>.<Data>200</Data>
     <Item>
     <Data>
     <Anchor xmlns="syncm l:metinf">
     <Next> 1295654526000</Next>
     </Anchor>
     </ Data>
     </Item>
     </ Status>
     <Sync>
     < CmdID>3</CmdID>
     <Target><LocURI> event</LocURI>
     </ Target>.<Source>
     <LocURI>Calendar </LocURI></Source>
     <Replace>
     <CmdID>4</CmdID>.<Item>.<Meta>
     <Type xmlns="syncml: metinf">text/x-v
calendar</Type>< /Meta>
     <Source>< LocURI>11484354304573</LocURI></Source>
     <Data>
     BEGIN:VCALENDAR..
     VERSION:1.0. .
     BEGIN:VEVENT..SUMMARY:Teste Sin cronia..LOCATION :..
     DTSTART:20110 122T110000Z..
```

```
     DTE ND:20110122T1200 00Z..
     DESCRIPTION :..
     UID:114843543 0457370490..
     AALA RM:20110122T1045 00Z;;0;..
     RRULE:. .X-FUNAMBOL-ALLD AY:0..
     END:VEVENT ..
     END:VCALENDAR. .
     </Data>
     </Item>
     <Item>
     <Meta>
     < Type xmlns="sync ml:metinf">text/ x-vcalendar</Typ
e>   </Meta>.
     <Sourc
e><LocURI>114843543O457370493</LocURI></Source>
     <Data>BEGIN:VCAL ENDAR..VERSION:1 .0..BEGIN:VEVENT ..
     SUMMARY:Teste  Sincronia..
     LOCATION:..
     DTSTART:20 110122T140000Z..
     DTEND:20110122T150000Z..
     DESCRIPT ION:..
     UID:114843 5430457370493..
     AALARM:20110122T134500Z;;0;..
     RRUL E:..X-FUNAMBOL-A LLDAY:0..
     END:VEV ENT..
     END:VCALEND AR..
     </Data>
     </Item>
     <Item>
     <Meta >
     <Type xmlns="s yncml:metinf">text/x-vcalendar</ Type>
     </Meta>
     <Source><LocURI>114843543O457370487</LocURI></Source>
     <Data>BEGIN:V CALENDAR..
     VERSIO N:1.0..
     BEGIN:VEV ENT..
     SUMMARY:Teste Sincronia..
     LOCATION:..
     DTSTART :20110122T100000 Z..
     DTEND:2011012 2T110000Z..
     DESCR IPTION:..
     UID:114 8435430457370487 ..
     AALARM:2011012 2T094500;;0;..
     R RULE:..X-FUNAMBOL-ALLDAY:0..
     END:
     VEVENT..
     END:VCALENDAR..
     </Data>
     < /Item>
     </Replace >
     </Sync>
     <Final />
     </SyncBody>
```

</ SyncML>.