



UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

MARCOS FELIPE FERREIRA AMAZONAS DE SOUZA

**OSTARA: UMA ARQUITETURA DE REFERÊNCIA PARA COLETA DE DADOS EM
CAMPO**

Tucuruí

2019



UNIVERSIDADE FEDERAL DO PARÁ
NÚCLEO DE DESENVOLVIMENTO AMAZÔNICO EM ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

MARCOS FELIPE FERREIRA AMAZONAS DE SOUZA

**OSTARA: UMA ARQUITETURA DE REFERÊNCIA PARA COLETA DE DADOS EM
CAMPO**

Dissertação apresentada ao Programa de Pós-graduação em Computação Aplicada do Núcleo de Desenvolvimento Amazônico em Engenharia da Universidade Federal do Pará, como requisito para obtenção do título de Mestre em Computação Aplicada.

Orientadora: Prof^ª. Dr^ª. Viviane Almeida dos Santos

Tucuruí

2019

MARCOS FELIPE FERREIRA AMAZONAS DE SOUZA

OSTARA: UMA ARQUITETURA DE REFERÊNCIA PARA COLETA DE DADOS EM CAMPO

Dissertação apresentada ao Programa de Pós-graduação em Computação Aplicada como pré-requisito para obtenção do título de Mestre em Computação Aplicada pelo Núcleo de Desenvolvimento Amazônico em Engenharia da Universidade Federal do Pará.

Aprovado em: ____/____/____

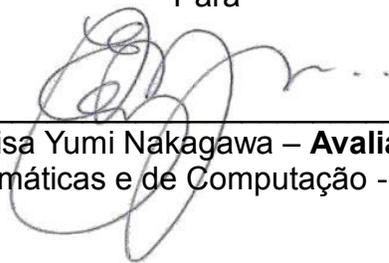
BANCA EXAMINADORA



Prof.ª Dr.ª Viviane Almeida dos Santos – **Orientadora**
Núcleo de Desenvolvimento Amazônico em Engenharia – Universidade Federal do Pará



Prof. Dr. Heleno Fülber – **Avaliador Interno**
Núcleo de Desenvolvimento Amazônico em Engenharia – Universidade do Federal do Pará



Prof.ª Dr.ª Elisa Yumi Nakagawa – **Avaliadora Externa**
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo

Prof. Dr. Bruno Merlin – **Suplente**
Núcleo de Desenvolvimento Amazônico em Engenharia – Universidade Federal do Pará

Dedico esta Dissertação à fonte de minhas virtudes, meus avós, João Batista de Souza, Leticia Amazonas de Souza e Nazaré Brito Ferreira.

AGRADECIMENTOS

Aos meus avós paternos, Leticia Amazonas de Souza e João Batista de Souza, por toda a dedicação. Agradeço pelo cuidado, pelos ensinamentos e pelo incomensurável e incondicional amor. Agradeço pela constante devoção e integral dedicação, em nome da preservação do bem e da prosperidade de nossa família.

À minha avó materna, Nazaré Brito Ferreira, pelas constantes demonstrações de pura força de vontade e resiliência às intempéries da vida. Agradeço aos constantes ensinamentos voltados para a valorização da simplicidade e humildade.

À Universidade Federal do Pará (UFPA), por proporcionar tamanha qualidade de ensino e dedicação aos seus discentes

Ao Núcleo de Desenvolvimento Amazônico em Engenharia por proporcionar crescimento em termos acadêmicos e profissionais.

Ao Programa de Pós-Graduação em Computação Aplicada, por me proporcionar a experiência de um mestrado.

À Agência Eletrobrás Eletronorte, pela disponibilização do time de mantenedores do projeto “Banco de Germoplasma”.

À orientadora Profa. Dra. Viviane Almeida dos Santos, pelos ensinamentos e conselhos durante o desenvolvimento deste trabalho. Obrigado pelo seu empenho.

Aos meus pais, Maria Helina Brito Ferreira e Marcos Fabiano Amazonas de Souza por me conceder o dom da vida. E aos meus irmãos, João Lucas, Roberto, Isabela, Gabriel e Lukas, pelos momentos compartilhados.

A minha esposa Jéssica Antonia Nunes Gomes, uma das mais virtuosas pessoas que pude conhecer. Agradeço as constantes demonstrações de sensibilidade, meiguice e paciência. Agradeço aos ensinamentos de determinação e aos constantes incentivos.

Aos mentores Luiz Sanches, Lucas Mousinho e Daylon Costa pelos constantes ensinamentos.

Aos alunos de iniciação científica Leandro Monte e Kissy Melo, pelo auxílio com o desenvolvimento do projeto.

“Se eu vi mais do que os outros, é por estar sobre os ombros dos gigantes.”

Isaac Newton

RESUMO

Coleta de dados é um importante e recorrente processo para a tomada de decisão, o qual se faz presente em diferentes áreas. De maneira geral, há diversas soluções elaboradas para o auxílio de atividades de coleta de dados em campo. Entretanto, tais soluções têm como foco a resolução de problemas específicos, causando uma falta de uniformidade em relação a características básicas pertencentes ao domínio de coleta de dados. Para suprir a necessidade de um repositório de conhecimento de domínio este trabalho teve como objetivo desenvolver uma arquitetura de referência (AR) que sirva como modelo para construção de sistemas pertencentes ao modelo de domínio de coleta de dados e o subdomínio de coleta de dados móveis. Para elaborar a arquitetura de referência proposta foi utilizado o processo ProSA-RA, este processo visa o desenvolvimento iterativo e incremental de arquiteturas de referência. Este processo é representado em quatro passos, que têm o objetivo de estruturar o desenvolvimento de arquiteturas de referência. Após a elaboração da AR utilizando o processo ProSA-RA, buscou-se instanciar um protótipo arquitetural para poder avaliar qualidades de tempo de execução pertencentes a AR proposta. O protótipo arquitetural foi modelado baseado nas necessidades de negócio do projeto “Banco de Germoplasma” mantido pela Agência Eletrobras Eletronorte. Este projeto busca manter informações referentes a espécies da fauna e flora da região de Tucuruí – PA, de forma que a porção arquitetural instanciada e implementada constitui uma ferramenta para monitoramento fenológico das espécies botânicas do projeto. A avaliação do protótipo arquitetural realizada com aos especialistas de domínio demonstrou qualidades referentes ao tempo de execução como performance, segurança, funcionalidade e usabilidade. Contudo as realizações de mais avaliações para obtenção de métricas de qualidade fazem-se necessárias como, por exemplo, a avaliação de elementos de qualidade de desenvolvimento como modificabilidade, portabilidade, integrabilidade e interoperabilidade.

Palavras Chave: Arquitetura de referência, Coleta de dados, Coleta de dados móveis.

ABSTRACT

Data collection is an important and recurring process for decision making, which is present in different areas. In general, there are several solutions developed to aid field data collection activities. However, such solutions focus on solving specific problems, causing a lack of uniformity in relation to basic characteristics belonging to the domain of data collection. To supply the need for a domain knowledge repository this work proposed the development of a reference architecture (RA) that serves as a model for the construction of systems that belong to the data collection domain model and the subdomain of mobile data collection. In order to elaborate the proposed reference architecture, the ProSA-RA process was used, this process helps the iterative and incremental development of reference architectures. This process is represented in four steps, which aim to structure the development of reference architectures. After the elaboration of RA using the ProSA-RA process, we tried to instantiate an architectural prototype to be able to evaluate execution-time qualities belonging to the proposed RA. The architectural prototype was modeled based on the business requirements of the "Banco de Germoplasma" project maintained by the Eletrobras Eletronorte Agency. This project seeks to maintain information regarding animal and botanic species of the Tucuruí - PA region. The instantiated and implemented architectural portion constitutes a tool for phenological monitoring of the botanical species of the project. The evaluation of the architectural prototype performed with the domain experts demonstrated runtime-related qualities such as performance, security, functionality and usability. However, the achievements of further evaluations to obtain quality metrics are necessary, such as the evaluation of development quality elements such as modifiability, portability, integrability and interoperability.

Key-words: Reference architecture, Data collecting, Mobile data collecting.

SUMÁRIO

1. INTRODUÇÃO	14
1.1. PROBLEMA.....	14
1.2. JUSTIFICATIVA	17
1.3. OBJETIVOS.....	18
1.3.1. Objetivo geral	18
1.3.2. Objetivos específicos	18
1.4. ESTRUTURA DO TRABALHO	18
2. FUNDAMENTAÇÃO TEÓRICA	20
2.1. ARQUITETURA DE SOFTWARE.....	20
2.2. ARQUITETURA DE REFERÊNCIA.....	23
2.2.1. Tipos de arquitetura de referência	26
2.3. O META MODELO RAMODEL.....	28
2.4. O PROCESSO PROSA-RA	31
2.5. DOMÍNIO DE APLICAÇÃO.....	37
2.5.1. Sistemas de coleta de dados	38
2.5.1.1. Mecanismos de coleta de dados.....	39
2.5.2. Sistemas de Coleta de Dados Móveis.....	39
2.5.2.1. Computação móvel.....	40
2.5.2.1.1. Arquirteturas para o domínio móvel	43
2.5.2.1.1.1. <i>Cliente/Servidor</i>	43
2.5.2.1.1.2. <i>Cliente/Agente/Servidor (C/A/S).....</i>	43
2.5.2.1.1.3. <i>Cliente/Interceptador/Servidor (C/I/S).....</i>	44
2.5.2.1.1.4. <i>Ponto a ponto.....</i>	45
2.5.2.1.1.5. <i>Agentes móveis</i>	46
2.5.2.2. Modelos de falha para o domínio móvel	47
2.5.2.3. Banco de dados para o domínio móvel	49
2.5.2.4. Sincronização	52

3.	TRABALHOS CORRELATOS	58
3.1.	SISTEMAS DE COLETA DE DADOS	58
3.1.1.	Sistema repositório de informações agro meteorológicas	58
3.1.2.	Sistema de aquisição e sincronização de dados sísmicos.....	61
3.1.3.	Sistema de coleta de dados sincronizáveis para áreas de mineração.....	62
3.1.4.	Arquitetura Alfredul.....	64
3.2.	SISTEMAS DE COLETA DE DADOS MÓVEIS	67
3.2.1.	Arquitetura genérica para coleta de dados geriátricos	67
3.2.2.	Ferramenta móvel para levantamento epidemiológico em saúde bucal....	69
3.2.3.	<i>Field data collecting synchronously system</i>	<i>71</i>
3.2.4.	Projeto maritaca	74
3.2.5.	Arquitetura de coleta de dados para ambientes heterogêneos	78
3.2.6.	Linha de produto de software para coleta de dados utilizando Android ...	82
3.2.7.	Privacy Protector Framework.....	95
3.2.8.	IOT based data aquisicion security	90
3.3.	CONCLUSÃO.....	93
4.	METODOLOGIA	94
4.1.	ELABORAÇÃO DA ARQUITETURA DE REFERÊNCIA	94
4.1.1.	Definição de modelo de domínio	94
4.1.2.	Passo RA1 - investigação das fontes de informação	95
4.1.2.1.	Pessoas.....	95
4.1.2.2.	Sistemas.....	95
4.1.2.3.	Publicações.....	95
4.1.3.	Passo RA2 - estabelecimento dos requisitos arquiteturais.....	97
4.1.3.1.	Identificação dos requisitos de sistema	97
4.1.3.2.	Identificação dos requisitos arquiteturais.....	97
4.1.3.3.	Identificação de conceitos de domínio	98
4.1.4.	Passo RA3 - projeto arquitetural	98
4.1.4.1.	Ponto de vista transversal	98
4.1.4.1.1.	<i>Visão conceitual.....</i>	<i>98</i>
4.1.4.1.2.	<i>Visão de funcionalidade</i>	<i>99</i>

4.1.4.1.3.	Visão de serviços de sistema	99
4.1.4.1.4.	Visão de variabilidade	100
4.1.4.2.	Ponto de vista de tempo de execução	104
4.1.4.2.1.	Visão de componentes colaborativos	104
4.1.4.2.2.	Visão de processo	107
4.1.4.2.3.	Visão dados compartilhados	116
4.1.4.3.	Ponto de vista de implantação	118
4.1.4.3.1.	Visão de implantação	118
4.1.4.3.2.	Visão de desenvolvimento	118
4.1.4.3.3.	Visão de parte física	118
4.1.4.3.4.	Visão tecnológica	119
4.1.4.4.	Ponto de vista de código fonte	120
4.1.4.1.	<i>Visão lógica</i>	120
4.1.4.1.	<i>Visão de módulo</i>	121
4.1.4.1.	<i>Visão estrutural</i>	121
4.1.5.	Passo RA4 - avaliação arquitetural	121
4.1.5.1.	<i>Avaliação de protótipo arquitetural</i>	121
4.1.5.1.1.	Detalhamento do problema	122
4.1.5.1.2.	Instância arquitetural	125
4.1.5.1.3.	Implementação	125
4.1.5.1.4.	Mecanismo avaliativo	126
4.1.5.1.5.	Participantes	127
4.1.5.1.6.	Ambiente	127
4.1.5.1.7.	Procedimento	128
5.	RESULTADOS	131
5.1.	A ARQUITETURA PROPOSTA	130
5.1.1.	Módulo de coleta	132
5.1.1.1.	Camadas	133
5.1.1.2.	Módulos internos.....	133
5.1.1.3.	Componentes Internos	136
5.1.2.	Módulo de serviços	139

5.1.2.1.	Camadas.....	139
5.1.2.2.	Módulos internos.....	141
5.1.2.3.	Componentes.....	142
5.1.3.	Módulo de sincronização.....	146
5.1.3.1.	Camadas.....	147
5.1.3.2.	Módulos.....	147
5.1.3.3.	Componentes.....	148
5.2.	AVALIAÇÃO DE PROTÓTIPO ARQUITETURAL.....	152
5.2.1.	Análise quantitativa.....	152
5.2.2.	Análise qualitativa.....	153
5.3.	DISCUSSÃO.....	153
6.	CONCLUSÃO.....	153
6.1.	TRABALHOS FUTUROS.....	153
	REFERÊNCIAS.....	153
	APÊNDICE A - ORGANIZAÇÃO DE FUNCIONALIDADES.....	153
	APÊNDICE B – MAPEAMENTO DOS REQUISITOS ARQUITETURAIS.....	153
	APÊNDICE C - MAPEAMENTO DOS CONCEITOS DE DOMÍNIO.....	153
	APÊNDICE D – VISÃO CONCEITUAL.....	153
	APÊNDICE E – VISÃO DE FUNCIONALIDADE.....	153
	APÊNDICE F – VISÃO DE SERVIÇOS DE SISTEMA.....	153
	APÊNDICE G – VISÃO VARIABILIDADE.....	192
	APÊNDICE H – VISÃO COMPONENTES COLABORATIVOS.....	193
	APÊNDICE I – VISÃO PROCESSO.....	195
	APÊNDICE J – VISÃO IMPLANTAÇÃO.....	197
	APÊNDICE K – VISÃO DESENVOLVIMENTO.....	199
	APÊNDICE L – ARQUITETURA DE REFERÊNCIA OSTARA.....	200
	APÊNDICE M – INSTÂNCIA DA AR OSTARA - DIAGRAMA DE PACOTES.....	203
	APÊNDICE N – INSTÂNCIA DA AR OSTARA - DIAGRAMA DE COMPONENTES.....	205
	APÊNDICE O – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO.....	206
	APÊNDICE P – FORMULÁRIO AVALIATIVO.....	207

1. INTRODUÇÃO

Com o advento de tecnologias de captação, armazenamento e disponibilização de dados, juntamente com a constante melhoria de mecanismos de telecomunicações, é possível utilizar-se de soluções de tecnologia de informação em localizações de difícil acesso.

A utilização de soluções computacionais em localidades desprovidas de infraestrutura tecnológica e de difícil acesso propicia uma maior integração informacional, ou seja, a realização de processos que comumente são impedidos por limitações físicas de tempo e/ou espaço.

Existem diversos tipos de aplicações computacionais capazes de realizar a integração de processos a ambientes ermos e com pouca infraestrutura. Sant'ana *et al.*, (2014) demonstra a utilização de elementos de hardware e software para prover diversos tipos de monitoramento do clima do território brasileiro, como por exemplo, a previsão do clima espacial e o monitoramento de dados de cintilação. Os elementos de hardware utilizados para a realização de tal integração são redes de sensores e softwares de processamento de dados, que têm respectivamente o objetivo de coletar dados dos locais que se deseja observar e realizar os passos para a disponibilização e utilização de tais dados.

Outra possível solução para uma maior integração informacional é a utilização de dispositivos computacionais móveis, como por exemplo, *palmtops*, PDAs e *smartphones*. É afirmado por Santos *et al.*, (2013) que, a computação móvel permite a realização de processos que envolvam reunir dados em campo, armazená-los temporariamente e realizar o processo de submissão dos dados para um ou vários centros computacionais de armazenamento, como por exemplo a realização de pesquisas em campo.

Faz-se importante ressaltar, que a utilização de recursos de TI que provêm vantagens ao integrar processos, não é exclusiva somente a locais ermos e sem infraestrutura. Como afirmado por Ito, Ferreira e Sant'ana, (2003), tal integração pode ocorrer de forma sistematizada e controlada, em ambientes onde seja necessário atuar pontualmente quando necessário, como por exemplo, sistemas que necessitem gerenciar aspectos da rotina de vendedores que trabalhem em campo. Tem-se neste cenário a utilização de recursos computacionais móveis para a aquisição de dados em diversos

tipos de ambientes, fornecendo possibilidades de comunicação com um repositório central de dados quando necessário.

Dentre os vários contextos para a utilização de recursos computacionais que proporcionam uma maior integração de processos necessários para a operacionalização de aplicações tanto acadêmicas quanto mercadológicas, verificam-se recorrentes soluções que buscam proporcionar ou aperfeiçoar atividades de coleta de dados (VASILESCU *et al.*, 2005; BONI *et al.*, 2006; BREGA *et al.*, 2008; TIAN, GAO e ZHOU, 2008; SILVA, ANDRADE e MARIN, 2008; JI *et al.*, 2010; ZHAO, ZHANG e MENG, 2012; SANTOS *et al.*, 2013; FREITAS FILHO, 2014; SANT'ANA *et al.*, 2014; WAKU *et al.*, 2015).

É definido por Evans (2004), que domínio de software é denotado pela aplicação de um produto de software, em uma determinada área de atuação, por intermédio do usuário que utiliza o software com a finalidade de resolução ou otimização da resolução de uma determinada atividade. Percebe-se então que os interesses e atividades advindas de processos executados por usuários estão diretamente relacionadas com os produtos de software que buscam suprir as necessidades do usuário.

Devido o crescente número de soluções voltadas para a realização de coleta de dados, torna-se perceptível a utilização de sistemas computacionais que apesar de serem fundamentalmente distintos em seus objetivos finais, buscam realizar ou aperfeiçoar a solução de problemas que fazem parte de uma mesma área de atuação, o que pela definição de Evans (2004), permite categorizar a coleta de dados como o domínio de tais aplicações.

Em virtude da vasta gama de projetos de software pertencentes ao domínio de coleta de dados, é possível verificar uma quantidade similar de propostas de abstrações concretas que buscam resolver problemas de casos específicos.

Corroborando com o íterim exposto, dentro do escopo de arquiteturas de software existe um elemento arquitetural que se popularizou, devido à possibilidade do mesmo servir como agregador de conhecimentos de domínio, a este elemento é dado o nome de arquitetura de referência, ou AR. Este elemento tem o objetivo de ser um repositório de conhecimentos referentes a abstrações e implementações destinadas a resolução de problemas pertencentes a um mesmo domínio, o que por consequência, acaba por

promover o reuso de boas práticas de design arquitetural e a fundamentação de entendimentos de domínio (NAKAGAWA *et al.*, 2011).

1.1. PROBLEMA

Boni *et.al.*, (2006) e Freitas Filho (2014) discorrem sobre o contínuo vislumbre da possibilidade de aquisição de informações realizada através de processos que ocorram independentemente e não são impedidos por limitações de tempo e/ou espaço. Uma vez eliminadas as fronteiras geográficas e conseqüentemente as barreiras físicas, estes processos seriam capazes causar uma mudança de paradigma em como são elaborados processos de aquisição de informações pertencentes a diferentes áreas.

Waku *et.al.*, (2015) estabelece que a coleta de dados pode ser definida como um procedimento sistemático que visa coletar dados para poder determinar informações relativas a variáveis relevantes, proporcionando desta forma a resposta de questões relevantes para pesquisa, o teste de hipóteses e o ponderamento em relação aos resultados obtidos.

Algumas soluções podem ser utilizadas para que a crescente necessidade de coleta de informações sem qualquer tipo de barreira física possa ocorrer, infraestruturas de redes sem fio devem ser disponibilizadas, assim como várias interfaces de comunicação com tais infraestruturas. Tais infraestruturas já existem e estão disponíveis, sendo estas a rede mundial de computadores, redes via satélites, redes via rádio, etc. Da mesma forma as interfaces de comunicação com tais redes também já existem como por exemplo os diferentes protocolos suportados pela internet, além dos diferentes tipos de sinal utilizados para comunicação sem fio, como por exemplo o *WIFI*, *bluetooth*, entre outros.

Os dispositivos móveis são um exemplo de plataforma que compreende o conjunto de requisitos de comunicação móvel citados anteriormente. Segundo Boni *et.al.*, (2006) a utilização de dispositivos como PDAs e *smartphones* como ferramentas de suporte a processos de obtenção e carregamento de vastas quantidades de dados é completamente possível devido à significativa diminuição do tamanho físico dos computadores somada à quantidade de estudos empregados para o progresso de tecnologias de comunicação.

A utilização de tais ferramentas também se torna possível devido ao barateamento do valor de aquisição de dispositivos móveis, o que torna possível a utilização de diversos dispositivos dentro do contexto de uma única aplicação. Outros fatores cruciais para a utilização de dispositivos móveis são a mobilidade e praticidade, características inerentes para aplicações que são pertencentes ao domínio de captação de informações ou captação de dados (BREGA *et.al.*, 2008).

O cerne da utilização de mecanismos de obtenção de informações é prover subsídios que corroborem para a construção de respostas que buscam responder alguma questão pertinente. Os subsídios comumente são evidências de qualidade que permitem a traduções de análises de dados. Por consequência da capacidade de agrupar dados para a futura análise esta metodologia é utilizada tanto em soluções acadêmicas quanto em soluções disponíveis no mercado, o que a torna uma abordagem comum a todos os campos de estudo (WAKU *et.al.*, 2015).

No que se refere ao âmbito prático de soluções que obtenham vantagem da inexistência de barreiras físicas impostas pela tecnologia em relação ao negócio, pode-se citar os cenários de negócio vivenciados por vendedores de campo, tal cenário necessita de uma considerável capacidade de mobilidade, a qual é inerente ao negócio. Na situação de vendas de campo, cada vendedor deve ser capaz de executar diversas tarefas referentes ao negócio de forma remota, como por exemplo, consulta de produtos em estoque, consulta de produtos vendidos, consulta de preços, consulta de comissões, informação de vendas efetivadas, manutenção cadastros de clientes, etc. (ITO, FERREIRA, SANT'ANA, 2003).

Dentro do contexto de aquisição de informações apresentado, outro exemplo que pode ser citado é o de necessidade de acesso de informações remotas, este cenário é muito presente no fluxo de trabalho de pesquisadores que regularmente viajam para efetivamente realizarem suas respectivas pesquisas, necessitando estabelecer comunicações periódicas para a realização da atualização de dados de pesquisa. Estas operações tornam-se possíveis devido à utilização de recursos móveis, como por exemplo, *smartphones* e aparelhos *handhelds* (ITO, FERREIRA, SANT'ANA, 2003).

De acordo com Brega *et.al.* (2008) a área médica também se utiliza destas ferramentas, aonde a certificação digital de imagens médicas também pode ser tida como

um exemplo de aplicação voltada para esta área, demonstrando o domínio de coleta de dados que utiliza-se de dispositivos móveis para a execução de seus processos (BREGA *et.al.*, 2008).

Outra solução para realizar a obtenção de dados de forma automatizada é a utilização de redes de sensores. Esta alternativa é muito utilizada para prover dados climáticos. Como exemplo desta aplicação pode-se mencionar o Programa EMBRACE que tem como objetivo a previsão de fenômenos de clima espacial. O programa capta os dados através de diversas redes de sensores que buscam captar diferentes tipos de informações referentes ao conteúdo eletrônico total localizado sobre a América do Sul até mesmo dados referentes a cintilação. Após a coleta de dados é necessário muito processamento para a realização da análise, o correto armazenamento e a disponibilização dos dados coletados (SANT'ANNA *et al.*, 2014).

Freitas (2014) define em seu trabalho que o processo de obtenção de dados baseada em questionários de papel mostra-se uma prática que necessita de um demasiado esforço para sua realização. Além disto, este processo não consegue garantir a segurança dos dados, podendo violar características de segurança de informação como a integridade, a qual pode ser facilmente comprometida caso os dados mantidos em papel sejam facilmente alteráveis. Em contraponto os mecanismos que automatizam processos inerentes ao domínio de coleta de dados auxiliam a garantir a segurança de informação e reduzem a complexidade do processo de transformação dos dados para o meio digital. Devido os motivos debatidos o processo de obtenção de dados utilizando papel acaba por tornar-se obsoleto.

O grande volume de soluções documentadas é voltado em sua maioria para a solução do próprio problema que está sendo abordado, seja ele acadêmico ou mercadológico. Isto corrobora para uma vasta quantidade de informações referentes a arquiteturas concretas pertencentes a um mesmo domínio, porém poucas destas tentam estabelecer abstrações genéricas no que tange a compreensão de problemas recorrentes pertencentes ao domínio o qual estão inseridas.

1.2. JUSTIFICATIVA

A necessidade de soluções que realizem a obtenção de informações faz-se presente em diversos contextos, o que por natureza acaba gerando um grande volume de informações correlacionadas pelo domínio a qual se referem.

O problema exposto anteriormente pode ser definido de forma similar a adversidade mencionada no trabalho de Fioravanti, Nakagawa e Barbosa (2010), em que é comprovada uma rica diversidade de sistemas computacionais, desenvolvidos para utilização acadêmica ou de mercado, pertencentes a um mesmo domínio, e a insuficiência de informações no que tange funcionalidades básicas, o que corrobora para a falta de consonância de tais funcionalidades. Diante do problema exposto, é possível verificar a crescente necessidade da criação de um repositório de conhecimentos relevantes para o domínio de coleta de dados.

A necessidade de aglomeração de boas práticas de design arquitetural, estilos arquiteturais, padrões de projeto e experiência de domínio em geral, voltada para a coleta de dados é um tema que foi abordado no trabalho de Waku *et al.*, (2015), no qual é proposta uma arquitetura de linha de produtos de software, ou *software product line* (SPL), voltada para a coleta de dados. O motivo pelo qual a utilização de uma abordagem de linha de produto de software foi realizada foi a possibilidade de reutilização sistemática, ou seja, mesmo em diferentes instâncias da arquitetura, as quais podem estar voltadas para situações de coleta de dados distintas, é possível reutilizar os componentes de software propostos pela arquitetura.

Para que seja possível realizar a construção de um repositório de conhecimento de um determinado domínio, primeiramente é de fundamental importância compreender que tais repositórios precisam ser abstratos, proporcionando a criação diferentes instâncias arquiteturais (WAKU *et al.*, 2015).

A utilização de arquiteturas de software como fontes de conhecimento de domínio é proposta por Guessi, Oliveira e Nakagawa (2015). A função de tais repositórios é a de permitir o reuso de experiência, através do entendimento preciso e bem fundamentado do domínio em questão. Estas bases de conhecimento contêm soluções abstratas, organizadas e estruturadas de problemas do domínio.

Portocarrero *et al.*, (2015), define arquiteturas de referência como repositórios de conhecimento de modelos de domínio, cujo os objetivos são fornecer um guia de de boas práticas para a elaboração de arquiteturas concretas para novos sistemas, para as extensões de sistemas pertencentes a domínios próximos, para a evolução de sistemas derivados ou instanciados a partir da arquitetura de referência e para o melhoramento da padronização e interoperabilidade de diferentes sistemas.

Mediante a constante necessidade de coleta de dados como atividade fundamental para diversos sistemas, a presente falta de uniformidade informacional de boas práticas e aspectos de design arquitetural que permeiam funcionalidades intrínsecas do domínio, faz-se necessário realizar a construção de um repositório de conhecimento de domínio, voltado a coleta de dados. Para realizar a construção de tal repositório, foi escolhida a utilização de arquiteturas de referência ou (AR), em virtude da vasta literatura disponível sobre o tema.

1.3. OBJETIVOS

1.3.1. Objetivo geral

Desenvolver uma arquitetura de referência que sirva como modelo para construção de sistemas pertencentes ao modelo de domínio de coleta de dados e o subdomínio de coleta de dados móveis.

1.3.2. Objetivos específicos

- Coletar informações referentes ao domínio de coleta de dados.
- Compreender modelos e arquiteturas de referência.
- Compreender os processos de elaboração de arquiteturas de referência.
- Coletar requisitos referentes a sistemas de coletas de dados.
- Conceber arquitetura de referência para o domínio proposto.
- Instanciar uma arquitetura baseada na AR proposta e nos requisitos coletados.
- Modelar sistema computacional baseado na arquitetura instanciada.
- Testar o sistema juntamente com os colaboradores participantes do projeto para validar sua eficácia e eficiência.

- Validar a arquitetura de referência através do teste do produto de software construído baseado na arquitetura.

1.4. ESTRUTURA DO TRABALHO

Neste capítulo foi contextualizado o escopo em que este trabalho de mestrado está inserido, o problema, a justificativa para a resolução de tal problema e os objetivos deste trabalho. O conteúdo subsequente do trabalho está organizado da seguinte forma. No Capítulo 2 são apresentados os conceitos referentes a arquiteturas de software, arquiteturas de referência (AR), sobre o domínio de coleta de dados e o subdomínio de coleta de dados móveis. No Capítulo 3 são analisados trabalhos e publicações pertencentes ao domínio de coleta de dados e o subdomínio de coleta de dados móveis. No Capítulo 4 é apresentado o processo de construção da arquitetura de referência proposta através do processo ProSA-RA, além disto, também é definido como será realizada a avaliação da arquitetura. A arquitetura de referência é apresentada no Capítulo 5, também são apresentados os resultados obtidos e a análise crítica entre a arquitetura proposta e os trabalhos correlatos. Por fim o Capítulo 6 apresenta as conclusões em relação ao trabalho e os trabalhos futuros.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. ARQUITETURA DE SOFTWARE

Necessidades e orientações acadêmicas e mercadológicas são determinantes para o comportamento, paradigma de funcionamento, tamanho e abordagens diferentes utilizadas em relação a sistemas de software. Em virtude disto soluções informatizadas vêm comumente expandindo suas proporções, tornando-se cada vez mais robustas, volumosas e demonstrando uma complexidade crescente. Devido esta demanda, abordagens mais sistemáticas e eficientes para estruturar e desenvolver tais aplicações fazem-se necessárias (FEITOSA, 2013).

Existiu uma emergente preocupação em conseguir entender as estruturas nas quais sistemas de software poderiam ser organizados, estes movimentos de estudos buscavam alcançar não somente a compreensão de sistemas computacionais e seus componentes, mas também métodos e descrições formais para tais sistemas, tais movimentos tiveram muita força nos anos noventa e contribuíram para o surgimento do estudo de Arquitetura de Software, que nada mais é do que uma área de pesquisa pertencente a Engenharia de Software (NAKAGAWA, 2006).

A construção de sistemas de software é realizada com o objetivo de contemplar os diversos objetivos pertinentes aos negócios de organizações. Pode-se compreender arquitetura como uma ponte entre tais objetivos de negócio das organizações, que são comumente abstratos, e o sistema resultante do final deste processo que é algo mais palpável, concreto. Enquanto que a definição de um caminho de metas abstratas para sistemas concretos pode ser uma tarefa que tenha um inerente alto nível de complexidade, a boa notícia é que arquiteturas de software podem ser projetadas, analisadas, documentadas e implementadas usando um acervo de técnicas conhecidas que apóiam e facilitam a realização desses objetivos de negócios e missão organizacional. A complexidade pode ser domada e tratada (BASS, CLEMENTS e KAZMAN, 2012).

Em virtude da crescente complexidade lógica e volume envolvendo diferentes aspectos e etapas da criação de sistemas de software, acadêmicos e profissionais da engenharia de software apresentam boas práticas, concepções e fundamentos para o

desenvolvimento de tais projetos de software à comunidade científica. Estes princípios de projeto objetivam conseguir produzir sistemas com maior qualidade e um menor custo. Para que a meta de ter projetos de software mais consistentes compreensíveis e de menor custo possa ser alcançada, o foco no projeto da estrutura central do sistema de software vem sendo desenvolvido, com a preocupação em aplicar os padrões validados pela Arquitetura de Software e garantem o entendimento e organização dos diferentes componentes do projeto (NAKAGAWA, 2006).

Dentre as muitas definições a respeito de arquitetura de software que podem ser pesquisadas e verificadas, é possível definir arquitetura de software como múltiplas estruturas ou um conjunto de estruturas que são necessárias para ponderar a respeito de um sistema, tais estruturas podem compreender desde elementos de software, as relações entre estes elementos de softwares, e as propriedades de ambos (BASS, CLEMENTS e KAZMAN, 2012).

Arquitetura de software pode ser conceituada como uma estrutura bem definida que inclui diversos elementos constituintes, as propriedades externas destes elementos e as conexões entre eles. Esta visão estruturada provê uma abstração para o sistema (FIORAVANTI, NAKAGAWA e BARBOSA, 2010).

Contrastante com várias outras conceituações de arquitetura de software que abrangem os conceitos de decisões "iniciais" ou "importantes" para o projeto do sistema, a definição proposta por Bass, Clements e Kazman (2012) abrange a ideia de que embora seja verdade que muitas das decisões arquiteturais são tomadas nos primeiros estágios do projeto, tais decisões não compreendem o conjunto total das decisões arquiteturais, especialmente se forem levados em conta projetos que utilizem filosofias de desenvolvimento ágil ou em espiral. Na verdade muitas das decisões que são tomadas no início do projeto não são decisões arquiteturais. Além disso, é difícil analisar uma decisão e defini-la como "importante", às vezes isto só será averiguado em outros estágios do projeto ou com o passar do tempo. Abordagens que buscam conceituar arquitetura de software através de decisões acabam por atribuir ao arquiteto de software do sistema a tarefa hercúlea de saber no início do projeto quais decisões a arquitetura necessita. Diferente das decisões, as estruturas são elementos mais facilmente

identificáveis no contexto de software, sendo assim estas são ferramentas poderosas para o design da aplicação.

As implicações técnicas da definição de arquiteturas de software utilizando o conceito de utilizado por Bass, Clements e Kazman (2012) podem ser descritas como arquitetura é um conjunto de estruturas de software; Arquitetura é uma abstração; Todo sistema de software tem uma arquitetura de software; Nem todas as arquiteturas são boas arquiteturas.

Nakagawa (2006) afirma que o objetivo central de uma arquitetura de software é diminuir a distância entre o que o software deve executar e a forma de implementação deste software. O ponto focal da utilização de uma arquitetura é auxiliar a visualização e tomada de decisões em aspectos estruturantes do projeto do sistema como, por exemplo, a organização do conjunto de seus componentes internos, estruturas de controle protocolos de comunicação, composição dos elementos do projeto e as funções dos componentes do projeto.

É comum que no desenvolvimento de uma área de pesquisa surjam terminologias que buscam criar uma base de linguagem comum, tal repositório de termos promove o comum entendimento de diferentes conceitos por pesquisadores da mesma área, neste trabalho é necessário o entendimento dos termos estilo arquitetural, modelo de referência, arquitetura concreta, instancia arquitetural.

Ao tentar elaborar e definir as arquiteturas de software que abordam as formas de abstrações e organizações de sua estrutura interna, baseadas em um conjunto de restrições acaba-se tendo como resultado a agregação de várias arquiteturas segundo suas características. Tal agregação pode ser definida como padrão arquitetural ou estilo arquitetural. Em virtude deste conceito, os sistemas que são desenvolvidos utilizando um determinado padrão de arquitetura podem ter suas características arquitetônicas definidas facilmente (FEITOSA, 2013). Existem representações que também que detêm o conhecimento de um escopo de aplicação que buscam revolver um determinado problema. Estas representações expressam-se através da definição das diferentes funcionalidades separadas da aplicação e as interconexões entre estes componentes representadas por suas comunicações de dados. Estas representações podem ser chamadas de modelos de referência (BASS, CLEMENTS e KAZMAN, 2012).

Zambiasi *et al.*, (2012) afirma que um modelo de referência é uma estrutura para decompor um problema de forma organizada, podendo ser caracterizados como modelos abstratos que aglutinam aspectos, pertencentes a um determinado escopo. Devido, agruparem muitos conhecimentos a respeito de componentes pertencentes ao domínio de aplicação, essas abstrações possuem práticas oriundas de domínios com muita maturidade.

Feitosa (2013) discute brevemente a respeito de arquitetura concreta definindo-a como uma arquitetura de software que já adere a uma série de especificações e padrões. Tais padrões e especificações podem facilitar o desenvolvimento dos diferentes módulos dos sistemas que utilizaram a arquitetura concreta proposta. Também é evidenciado que o termo instância arquitetural refere-se a uma arquitetura concreta dentro de um problema específico ou contexto de aplicação.

2.2. ARQUITETURA DE REFERÊNCIA

Angelov, Grefen e Greefhorst (2009) afirmam que as primeiras aparições de arquiteturas de referência foram como graus mais elevados de abstrações baseadas em arquiteturas concretas específicas que pertenciam a um escopo específico de aplicação, e que a complexidade emergente das aplicações informatizadas faz com que o processos sejam mais eficientes no que tange o design de tais aplicações e níveis de interoperabilidade mais elevados dão um posto de crescente destaque a área de Arquitetura de referência.

As abstrações conhecidas como arquiteturas de referência compreendem de um conjunto de boas práticas arquiteturais, sendo altamente recomendável a adoção de tais práticas de forma sistemática em sistemas que pertençam ao mesmo domínio de aplicação pautado pela arquitetura. O objetivo desta abstração é fornecer uma fundação que possibilite que projetos de mesmo domínio de aplicação possam iniciar seu ciclo de vida, permitindo assim que os arquitetos envolvidos no projeto concentrem seus esforços em problemas arquiteturais específicos, não sendo necessária a criação de soluções genéricas para problemas já resolvidos, uma vez que a arquitetura permite a reutilização de soluções de problemas. Sendo assim a não utilização de uma arquitetura de referência em um projeto não necessariamente garante a falha do projeto do sistema, porém o

esforço realizado por parte da equipe responsável pela arquitetura da aplicação será muito maior no que tange a realizar pesquisas, investigações e refletir sobre as decisões arquiteturais (MACHADO, 2015).

Nakagawa, Antonino e Becker (2011) definem que o termo arquitetura de referência pode ser atribuído a um tipo específico de abstração que tem o objetivo de aglutinar conhecimentos de diferentes perspectivas de construção de sistemas de software e suas arquiteturas que pertencem a um domínio de aplicação específico, suportados por uma terminologia de domínio única, clara, não ambígua e amplamente compreendida. Devido isto, tais abstrações são intrinsecamente genéricas, necessitando explorar aplicações que suportam o desenvolvimento de software para o domínio especificado, estilos arquiteturais, padrões de projeto, regras de negócio, boas práticas de desenvolvimento de software.

De acordo com a perspectiva fornecida (BASS, CLEMENTS e KAZMAN 2003 apud ANGELOV, GREFEN e GREEFHORST 2009) as abstrações conhecidas como arquiteturas de referência podem ser compreendidas como esquemas para mapear modelos de referência de um domínio específico, traduzindo as suas diretrizes e orientações para componentes de software que realizem funcionalidades que possam ser parte integrante do modelo de referência.

O RUP (*Rational Unified Process*) define arquitetura de referência como uma forma arquitetural previamente concebida, que tenha sido instanciada, projetada e comprovada dentro de um escopo técnico particular. Existem também artefatos que permitem o apoio a utilização destas abstrações (PROCESS, 2001).

Galster (2015) conclui que arquiteturas de referência de software são ferramentas muito eficazes em proporcionar uma maior facilidade de padronização e interoperabilidade de sistemas que atuam dentro de um domínio de aplicação, uma vez que sistemas e/ou arquiteturas concretas podem ser desenvolvidos seguindo diretrizes, regras de design, boas práticas, padrões e princípios da arquitetura de referência do mesmo domínio. Em virtude disto estas abstrações ajudam a simplificar o processo de

desenvolvimento de novos sistemas de software e arquiteturas que pertençam a um domínio de aplicação específico.

A importância de arquiteturas de referência é evidenciada por Angelov, Grefen e Greefhorst (2009) ao definir que tais abstrações são uma estimativa de que a reutilização do conhecimento adquirido facilita os trabalhos futuros de uma equipe de projeto ou times de design de um determinado domínio. Isto é possível, pois os conhecimentos de domínio adquiridos são referentes a processos sistematizados que definem terminologias únicas de domínio, não ambíguas e amplamente compreendidas dentro do escopo que abrange diversos sistemas de software.

A importância da existência de arquiteturas de referência para determinados domínios é também reforçada por Grosskurth e Godfrey (2005), ao afirmarem que a agregação de conhecimento dos diferentes aspectos fundamentais, subsistemas e relacionamentos a respeito de aplicações que pertencem a um mesmo domínio promove vantagens. Dentre estas são destacadas a reutilização de conhecimento, que proporciona novas aplicações baseadas em conhecimentos validados em nível de projeto e implementação para construção de novas soluções e/ou arquiteturas concretas pertencentes a um mesmo escopo de aplicação, também se destaca a característica de facilitação do entendimento de aplicações que não podem ter sua própria documentação arquitetural específica.

Em virtude da clara semelhança das definições entre arquitetura de referência, modelo de referência e estilo arquitetural, Feitosa (2013) ressalta a importante diferença entre cada um destes tópicos enfatizando as distinções entre seus conceitos, no qual arquiteturas de referência estabelecem arcabouços que são capazes de suprir definições e caracterizações de funcionalidades para aplicações informatizadas de um mesmo domínio de aplicação, diferentemente disto estilos arquiteturais são abstrações mais genéricas e abrangentes, pois podem ser utilizadas independentemente de seu domínio, por fim os modelos de referência são coleções de funcionalidades de um domínio, porém estas divisões não geram caracterizações, pois não possuem aplicação específica.

2.2.1. Tipos de arquitetura de referência

O problema em identificar categorias ou tipos de arquiteturas de referência se dá devido ao fato desta categorização ser baseada em satisfação e restrição de diversos requisitos arquiteturais, que necessitam ser massivamente abrangentes ao escopo de domínio da AR, além de muitas vezes dependerem de características intrínsecas ao modelo de domínio em que estão contidas. Este problema se intensifica se levarmos em consideração os diferentes interesses envolvidos em sua utilização (FEITOSA, 2013).

Angelov, Grefen e Greefhorst (2009) estabelecem uma estrutura de classificação para arquiteturas de referência à luz de três dimensões chamadas contexto, metas e design. Basicamente se as metas são relevantes para o contexto e o design reflete as metas e contexto uma AR é "congruente". Também se define o conceito de tipo de tipo que são as combinações de valores em cada dimensão capazes de gerar uma AR congruente, que tem mais chances de efetividade. Outro conceito definido é o de sub-dimensão, esta é uma dimensão que aborda um aspecto específico da dimensão principal na qual está contida. As sub-dimensões propostas para a dimensão de contexto são "Onde vai ser usado?", "Quem define?" e "Quando é definido?", a sub-dimensão da dimensão de metas é "Por que é definida", por fim as sub-dimensões da dimensão de design são "O que é descrito?", "O quão detalhado é descrito?", "O quão concreto é descrito?", "Como é representado?".

Tabela 1 - Tipos de arquitetura de referência

Tipo de AR	Descrição do tipo de AR
Tipo 1	São arquiteturas abstratas e semi-formais, projetadas com o intuito de serem utilizadas por várias organizações e apresentarem uma especificação clara que permita que leigos possam compreendê-la. Estas arquiteturas permitem a liberdade de como as empresas irão implementar os elementos arquiteturais que compõem a sua estrutura, contendo a descrição de interfaces e componentes.
Tipo 2	São abstrações concretas ou semi-concretas, semi-formais desenhadas para que sua implementação seja exclusiva a uma única organização, servindo como modelo padrão para a elaboração do conjunto de aplicações de software da empresa, definindo interfaces e componentes arquiteturais. Diferentemente das arquiteturas de tipo 1 estas arquiteturas utilizam em sua estrutura políticas e diretrizes organizacionais com o objetivo de facilitar seu uso dentro da instituição, bem como utilizam tecnologias e padrões que se adéquam a realidade institucional.
Tipo 3	As arquiteturas de referência do Tipo 3 são concretas e semi-formais, sendo assim promovem as tecnologias organizacionais e diretrizes detalhadas, além disto elas devem ser claras e facilmente entendíveis. Estas temo intuito de impulsionar produtos de softwares organizacionais abordando a implementação de suas principais interfaces e componentes, e a participação de várias instituições, para facilitar arquiteturas de referência posteriores.
Tipo 4	Semelhantes às arquiteturas do Tipo 2 as arquiteturas do tipo 4 também são desenvolvidas para serem implementadas em uma única empresa. Utilizadas somente como um guia na concepção e implementação das futuras aplicações da organização.
Tipo 5	São arquiteturas inovadoras, preliminares de facilitação com o intuito de serem implementadas em várias organizações. Tem o objetivo de facilitar a concepção de arquiteturas que serão necessárias no futuro. Faz-se necessário que estas arquiteturas definam os componentes necessários para implementá-la, programas que suportem seus componentes e protocolos que sintetizem as interações entre componentes.
Tipo 6	Arquiteturas preliminares criadas por uma organização, que visam auxiliar o desenvolvimento de arquiteturas de referência para o futuro, este tipo de criação necessita de um grande esforço para ser executado, o que a limita somente a instituições muito grandes.

Fonte: Angelov, Grefen e Greefhorst, 2009.

2.3. O META MODELO RAMODEL

O desenvolvimento de sistemas de software é diretamente impactado pela quantidade e qualidade das informações disponíveis no decorrer de projetos de software, estas informações descrevem os elementos e estruturas que irão compor o produto de software resultante destes projetos. Devido à importância e relevância das investigações e estudos à cerca de arquiteturas de referência têm tomado um corpo muito expressivo no meio acadêmico e na indústria, devido ao fato destas abstrações funcionarem como repositórios de conhecimento de um determinado domínio de aplicação, provendo um entendimento mais refinado e essencial sobre os elementos que compõem arquiteturas de software, como utilizá-las, e evoluí-las corretamente. Em virtude desta crescente relevância e da ausência de modelos de referência que possam suprir a construção de arquiteturas de referência com informações estruturadas, foi desenvolvido um modelo de referência para arquiteturas de referência chamado RAModel (*Reference Architecture Model*). Tal modelo oferece informações adaptáveis a qualquer domínio de aplicação, descrevendo os componentes e as conexões presentes em arquiteturas de referência (NAKAGAWA, OQUENDO e BECKER, 2012).

É definido por Allian, Oliveira e Nakagawa (2015) que o RAModel define um total de quatro grupos chamados domínio, aplicação, infra-estrutura e elementos transversais. Tais grupos podem vir a fazer parte da estrutura de uma arquitetura de referência, cada um destes grupos tem um sub-conjunto de elementos que expressam aspectos arquiteturais relevantes.

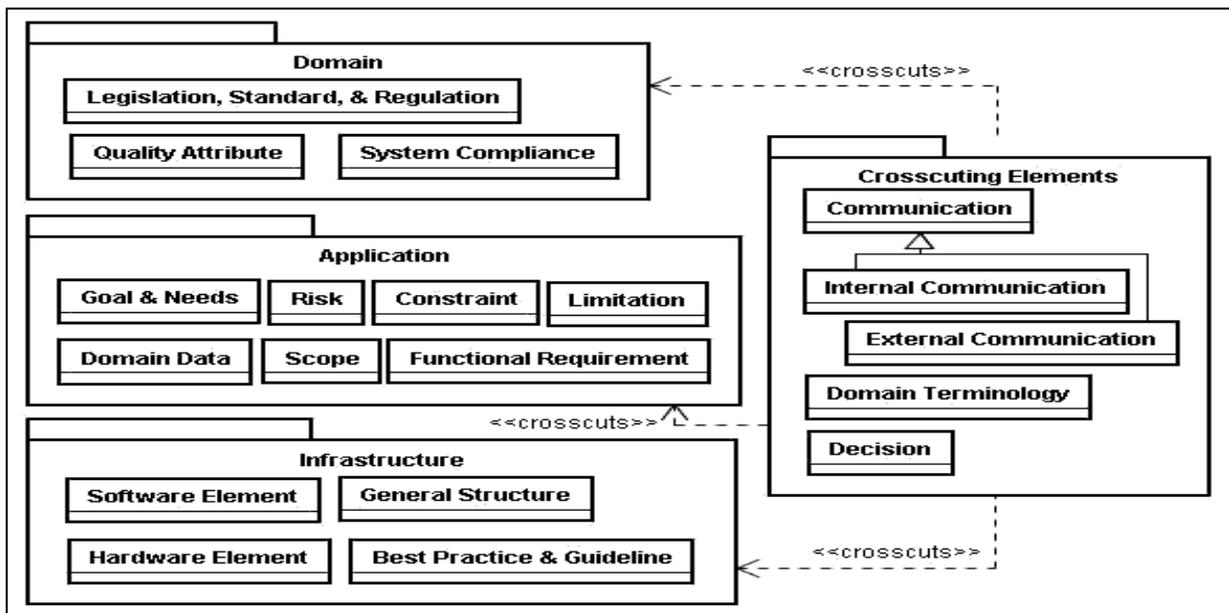


Figura 1 - Estrutura do modelo de referência RAModel. Fonte: NAKAGAWA, OQUENDO e BECKER, 2012.

Tabela 2 - Grupos de elementos definidos pelos RAModel

(Continua)

Elementos do grupo Domínio	
Elemento	Descrição
Legislação, Padrões, Regulamentos	Leis, padrões e regulamentos existentes no domínio que deve estar presente nos sistemas resultantes da arquitetura de referência.
Atributos de qualidade	Atributos de qualidade, por exemplo, manutenibilidade, portabilidade, e escalabilidade, que são desejados em sistemas resultantes da arquitetura de referência.
Conformidade de sistema	Meios para verificar se os sistemas desenvolvidos a partir da arquitetura de referência seguem as legislações, normas e regulamentos existentes.
Elementos do grupo Aplicação	
Elemento	Descrição
Restrições	Restrições apresentadas pela arquitetura de referência e / ou restrições em partes específicas de uma arquitetura de referência.

Tabela 2 - Grupos de elementos definidos pelos RAModel

(Continua)

Elementos do grupo Aplicação	
Elemento	Descrição
Dados de domínio	Dados comuns encontrados em sistemas do domínio. Estes dados são apresentados em um nível mais elevado de abstração, considerando o maior nível de abstração da arquitetura de referência.
Requisitos funcionais	Conjunto de requisitos funcionais comuns em sistemas desenvolvido usando esta arquitetura.
Meta e necessidades	Intenção da arquitetura de referência e necessidades que poderiam ser satisfeitas pela arquitetura de referência.
Limitações	Limitações apresentadas pela arquitetura de referência e / ou limitações em parte (s) específica (s) de uma arquitetura de referência.
Riscos	Riscos no uso da arquitetura de referência e / ou riscos em usar alguma parte dessa arquitetura.
Escopo	Escopo definido pela arquitetura de referência, ou seja, o conjunto de sistemas desenvolvidos com base na arquitetura de referência.
Elementos do grupo Infra-estrutura	
Elemento	Descrição
Melhores práticas e diretrizes	Práticas maduras que visam desenvolver sistemas do domínio. Essas práticas podem ser acompanhadas de diretrizes descrevendo como aplicá-las.
Estrutura geral	Estrutura geral da arquitetura de referência, representada às vezes usando estilos arquiteturais existentes.
Elementos de hardware	Elementos de hardware, como servidores e dispositivos, que hospedam sistemas resultantes da arquitetura de referência.
Elementos de software	Elementos de software presentes na arquitetura de referência, por exemplo, subsistemas e classes, que poderiam ser usados para desenvolver sistemas de software.

Tabela 2 - Grupos de elementos definidos pelos RAModel

(Conclusão)

Elementos do grupo Elementos transversais	
Elemento	Descrição
Decisões	Decisões, incluindo descrição da decisão, opções (alternativas), lógica e <i>tradeoffs</i> , devem ser relatados durante o desenvolvimento da arquitetura de referência.
Terminologia de domínio	Conjunto de termos do domínio que são amplamente aceitos pela comunidade relacionada a esse domínio, sendo assim, aceitos na descrição da arquitetura de referência.
Comunicação externa	Meios pelos quais ocorrem trocas de informações entre os sistemas resultantes da arquitetura de referência e o ambiente externo.
Comunicação interna	Meios pelos quais ocorrem trocas de informações entre partes internas dos sistemas resultantes da arquitetura de referência exposição.

Fonte: Adaptado de Nakagawa, Oquendo e Becker, 2012.

2.4. O PROCESSO PROSA-RA

A existência de um processo que facilite a tarefa de design e uso de arquiteturas de referência é muito importante, devido ao fato de que tais ferramentas configuram-se como um grande facilitador no desenvolvimento do projeto de aplicações informatizadas pertencentes a um domínio de aplicação, uma vez que reúnem uma base de conhecimentos a respeito do domínio, define as restrições, a estrutura do sistema e as funcionalidades necessárias que devem ser contempladas em sistemas de software daquele domínio. Devido a esta crescente necessidade, o estabelecimento de métodos que suportem e facilitem o design e teste de arquiteturas de referência é importante (NAKAGAWA, 2006).

Ao abordar em seu artigo a definição e teste de uma arquitetura de referência para ambientes educacionais, Fioravanti, Nakagawa e Barbosa (2010) conceituam o processo ProSA-RA como uma metodologia que visa o desenvolvimento iterativo e incremental de aplicações, composta em quatro passos, que tem o objetivo de estruturar o desenvolvimento de arquiteturas de referência. Sendo parte do ProSA (*Pro-cess based*

on Software Architecture) que ajuda a estabelecer arquiteturas de referência ARs (ProSA-RA), a especialização das mesmas (ProSA-S) e suas instanciações (ProSA-I).

Allian, Oliveira e Nakagawa (2015) define que o processo ProSA-RA toma como subsídios os conhecimentos levantados pelo modelo de referência RAModel à cerca das características fundamentais que podem estar presentes em arquiteturas de referência, tais como, elementos e estruturas arquiteturais e suas conexões internas.

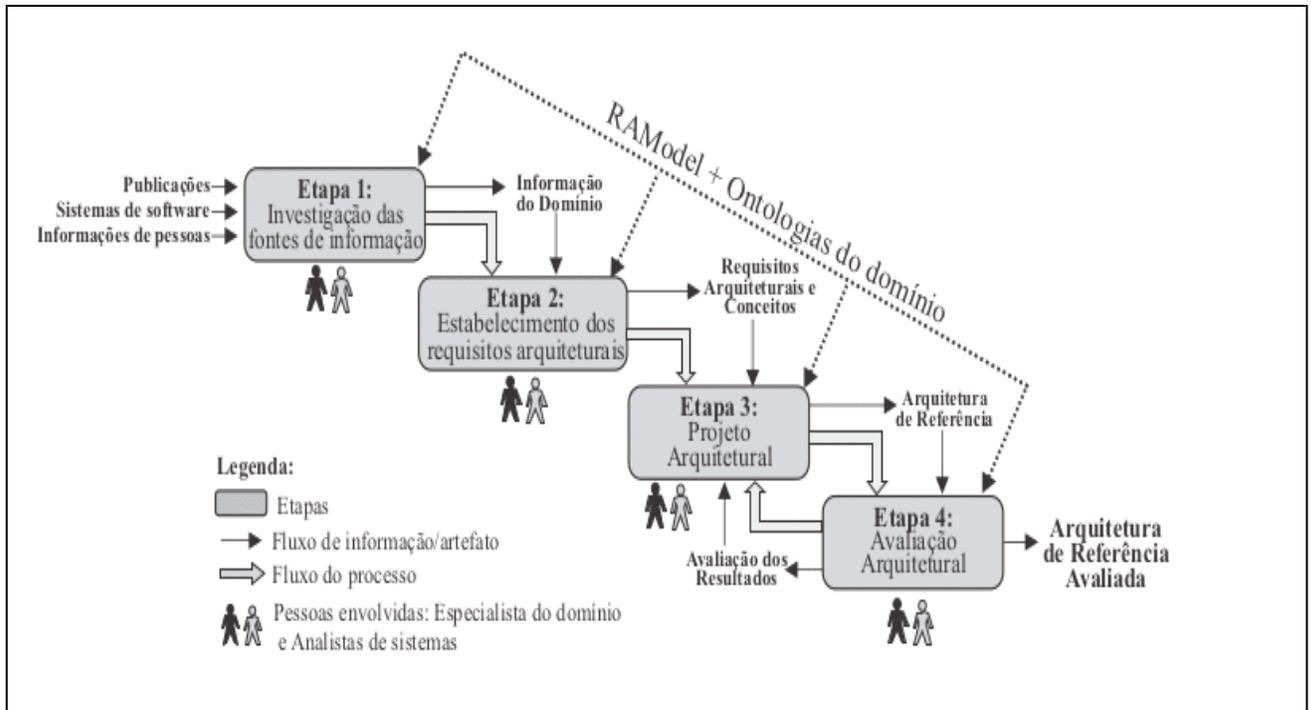


Figura 2 - Etapas do processo ProSA-RA. Fonte: Allian, Oliveira e Nakagawa, 2015.

De acordo com Nakagawa *et al.*, (2014) o processo ProSA-RA é composto por quatro etapas distintas, conforme a Figura 2 são descritas a seguir:

- **Passo RA-1: Investigação das fontes de informação:**

Para que a arquitetura de referência possa gerar arquiteturas concretas e sistemas informatizados eficientes, faz-se necessário um agrupamento do conhecimento acerca do domínio de aplicação ao qual estas arquiteturas concretas e sistemas de software gerados pela arquitetura de referência pertencerão. Desta forma é necessário coletar informações de diferentes fontes para que este conjunto de informações seja reunido. As fontes pesquisadas devem suprir os pesquisadores com dados a respeito das atividades e processos que são utilizados para a coesão arquitetural de sistemas informatizados e

arquiteturas pertencentes ao mesmo domínio de aplicação. O processo ProSA-RA define que as fontes de informação principais são pessoas (usuários, pesquisadores, clientes, desenvolvedores ou pessoas envolvidas com o domínio); sistemas de software pertencentes ao domínio de aplicação, publicações, modelos de referência e/ou outras arquiteturas de referência e ontologias de domínio. Além da sugestão das fontes anteriores, também é sugerido que o metamodelo RAModel também possa ser utilizado neste processo como provedor de informações a respeito dos elementos pertencentes a uma arquitetura de referência e seus relacionamentos.

- **Passo RA-2: Estabelecimento dos requisitos arquiteturais:**

Uma vez que a investigação das fontes de informação já foi concluída a segunda etapa pode ser iniciada. Nesta etapa será realizada a análise arquitetural da arquitetura de referência, para tal tarefa ser realizada faz-se necessário a análise dos dados coletados e posteriormente a identificação de três conjuntos de elementos, a coleção de requisitos de software pertencentes ao domínio de aplicação, baseado nestes requisitos de software a coleção de requisitos de arquitetura de referência é definida, por último a coleção de conceitos que devem estar presentes na arquitetura de referência.

- **Passo RA-3: Projeto arquitetural:**

Esta etapa é utilizada para construir a descrição da estrutura arquitetural da arquitetura de referência. São consideradas arquiteturas concretas, padrões, estilos e suas combinações provenientes da primeira etapa do processo ProSA-RA o passo RA-1. Em virtude desta abordagem, os conceitos identificados anteriormente são organizados tendo como base os padrões e estilos verificados. É importante ressaltar que esta etapa é construída utilizando-se o RAModel como um framework de base para a estrutura da AR.

Esta etapa também lança mão da utilização de visualizações arquiteturais para descrever a arquitetura de referência. Devido a constante evolução empregada processo ProSA-RA devido suas várias experiências de utilização, é sugerido que as visualizações arquiteturais sejam organizadas em grupos perspectivas arquiteturais, que nada mais são do que coleções de uma ou mais visualizações. Informações que abrangem um aspecto informacional pertencente a arquitetura de referência são representadas por visualizações dentro de um mesmo grupo de perspectiva arquitetural. A representação

de tais visualizações com diagramas UML 2.2 é possível independentemente de domínio trabalhado.

O projeto arquitetural utiliza quatro pontos de vista arquiteturais, sendo eles o ponto de vista de elementos transversal, execução, implantação e código fonte. Ao abordar o ponto de vista de elementos transversais, verifica-se que este é utilizado como base para a construção dos pontos de vista posteriores. Este ponto de vista tem como objetivo destacar aspectos gerais da arquitetura de referência, estes aspectos podem ser terminologias, definições relevantes e elementos variáveis. O ponto de vista de tempo de execução exibe em um nível mais alto de abstração o comportamento de execução dos sistemas gerados pela arquitetura de referência. O ponto de vista de implantação descreve os aspectos de rede, hardware, software e a estrutura analítica do projeto inerentes aos sistemas gerados por esta arquitetura. Por fim o ponto de vista de código fonte explana os detalhes da arquitetura de referência oriundos dos módulos e estruturas de software que a compõem.

Tabela 3 – Pontos de vista, visões, descrições e diagramas

(Continua)

Pontos de vista	Visualizações	Descrições	Diagramas
Elementos transversais	Conceitual	Apresenta a estrutura (opcionalmente o conteúdo) da informação gerenciada internamente pela arquitetura de referência. Um glossário pode ser utilizado para descrever cada termo utilizado, podendo ser substituído por uma ontologia de domínio.	Diagrama de definição de bloco
	Funcionalidade	Define, num alto nível de abstração, as principais funcionalidades, seus relacionamentos e os seus possíveis consumidores – Ex., Funções sistemas ou pessoas. Um fluxo de ação pode ser adicionado para detalhar cada funcionalidade.	Diagrama de caso de uso

Tabela 3 – Pontos de vista, visões, descrições e diagramas

(Continua)

Pontos de vista	Visualizações	Descrições	Diagramas
------------------------	----------------------	-------------------	------------------

Elementos transversais	Serviços de sistema	Define, num alto nível de abstração o conjunto de serviços (funções) mínimo que precisam ser suportados por todas as implementações da arquitetura de referência. Esta informação é útil para garantir a interoperabilidade entre diferentes instâncias da arquitetura de referência. Tabelas mostrando o relacionamento entre serviços e requisitos arquiteturais, ou entre serviços e conceitos suportando um serviço em questão também podem ser apresentadas.	Diagrama de definição de bloco
	Variabilidade	Descreve quais são os elementos da arquitetura de referência e como eles podem ser utilizados para construir instâncias da arquitetura de referência. Por exemplo, pode mostrar alternativas (relacionamentos or-exclusivos), elementos opcionais e elementos abstratos.	Diagrama de definição de bloco
Execução	Componentes Colaborativos	Descreve a arquitetura de referência como um conjunto de componentes que interagem usando um conjunto de repositórios de dados compartilhados (fluxo de dados) para executar funções em tempo de execução. Também pode apresentar portas e interfaces fornecidas / necessárias de componentes.	Diagrama de blocos internos
	Processo	Descreve a arquitetura de referência como um conjunto de unidades executadas concorrentemente-abstrações de elementos de software concretos, como tarefas, processos ou threads de execução - e suas interações. Também mostra repositórios de dados acessados por essas unidades, mensagens trocadas e interfaces expostas.	Diagrama de blocos internos

Tabela 3 – Pontos de vista, visões, descrições e diagramas

(Continua)

Pontos de vista	Visualizações	Descrições	Diagramas
------------------------	----------------------	-------------------	------------------

Execução	Dados compartilhados	Descreve como a arquitetura de referência é estruturada em componentes de leitura e gravação de informações em vários repositórios de dados compartilhados por meio de portas e interfaces. Repositórios de dados agem como mediadores da interação entre diferentes módulos acessando seu conteúdo.	Diagrama de blocos internos
Implantação	Implantação	Descreve máquinas, softwares instalados nessas máquinas e conexões de rede usadas por instâncias da arquitetura de referência.	Diagrama de pacote
	Desenvolvimento	Descreve máquinas, softwares instalados nessas máquinas e conexões de rede usadas por instâncias da referência arquitetura.	Diagrama de componente
	Parte física	Descreve o mapeamento de software no hardware. Ele enfatiza os requisitos não funcionais de instâncias da arquitetura de referência, como confiabilidade, desempenho e escalabilidade.	Diagrama de pacote
	Tecnologia	Ele apresenta requisitos para padrões/ferramentas comumente usados em instâncias da arquitetura de referência para orientar a seleção de tecnologias apropriadas ou compatíveis para os propósitos da arquitetura de referência.	Descrição textual
Código fonte	Logical	Descreve a decomposição dos módulos da arquitetura de referência e apresenta os requisitos funcionais do sistema. A arquitetura de referência pode ser representada em termos de subsistemas, pacotes, estruturas, classes e interfaces.	Diagrama de definição de bloco
	Module	Apresenta os principais módulos funcionais, o fluxo de dados entre esses módulos e as principais interfaces.	Diagrama de componente

Tabela 3 – Pontos de vista, visões, descrições e diagramas

(Conclusão)

Pontos de vista	Visualizações	Descrições	Diagramas
-----------------	---------------	------------	-----------

Código fonte	Estrutural	Separa o contexto de implementação (ou seja, partes de hardware e software disponíveis e outros recursos externos) do contexto de aplicativo que será desenvolvido. Mostra a interação com o ambiente. Cada módulo também pode conter uma breve explicação descrevendo seus recursos.	Diagrama de definição de bloco
---------------------	------------	---	--------------------------------

Fonte: Adaptado de Nakagawa *et al.*, 2014.

- **Passo RA-4: Avaliação arquitetural:**

Nesta etapa é importante revisar juntamente com partes interessadas as definições arquiteturais da arquitetura de referência proposta, isto é realizado com o intuito de localizar inconsistências nas definições arquiteturais. Esta sistematização que tem como objetivo a elaboração, representação e avaliação de arquiteturas de referência também possui maturidade. Isto é demonstrado por Nakagawa *et al.*, (2014) ao afirmar ter realizado experimentações do uso do processo ProSA-RA para o desenvolvimento de cinco arquiteturas de referência para o meio acadêmico e uma para a indústria. Visando tornar o processo mais eficaz, estas experimentações foram utilizadas para realizar um processo de realimentação de informações a respeito da estrutura do processo e o refinamento do mesmo. Dentre as melhorias e observações feitas ao processo denota-se que é necessário definir o domínio de aplicação que a arquitetura irá abranger e logo após isso a execução dos quatro passos do processo pode ser realizada.

2.5. DOMÍNIO DE APLICAÇÃO

Com objetivo de tornar o processo ProSA-RA mais efetivo, e por consequência evoluí-lo, foram conduzidos diversos estudos referentes à natureza de arquiteturas de referência (NAKAGAWA *et al.*, 2011; DUARTE, 2012; FEITOSA, 2013; NAKAGAWA *et al.*, 2007; OLIVEIRA e NAKAGAWA, 2011).

Os estudos e experimentações realizados corroboram com a proposição de que o estabelecimento do domínio de aplicação da arquitetura de referência faz-se fundamentalmente necessário como primeira tarefa pertencente ao processo de elaboração e design da mesma. Uma abordagem comumente utilizada para determinar o escopo/modelo de domínio ao qual a arquitetura irá abranger é a realização de uma

análise de um conjunto de sistemas que devem ser produzidos utilizando-se a arquitetura de referência (NAKAGAWA *et al.*, 2014).

2.5.1. Sistemas de coleta de dados

Sob a luz da definição de domínio de aplicação citada anteriormente, este trabalho estipula como modelo de domínio, sistemas que efetuem coletas de dados.

Waku *et al.*, (2015) estabelece o conceito de coleta de dados como sendo uma metodologia que busca agregar e aferir informações sobre aspectos relevantes a um determinado contexto, permitindo o estabelecimento de conclusões acerca de questões de pesquisa que tangem os resultados obtidos.

Boni *et al.*, (2006) sugere a ideia de que a utilização de coleta de dados faz-se presente em diferentes em diferentes contextos, discutindo especificamente o contexto da saúde pública e afirmando que este apresenta abundante utilização deste recurso. Também é argumentado que grande parte das coletas de dados realizadas nesta área deparam-se com a necessidade do deslocamento de materiais e ferramentas até o local onde a coleta será realizada, culminando na necessidade de uma metodologia que consiga abranger o seguro armazenamento das informações coletadas e sua disponibilização em formato digital, para a realização de análises futuras.

Comuns em diversas áreas de pesquisa as aplicações que implementam esta metodologia tem como seu propósito fundamental a obtenção de indicadores de qualidade. Estas amostras representam a análise de dados que foi aplicada, auxiliando os pesquisadores no processo de composição de respostas que tenham um nível aceitável de confiabilidade. Devido a sua necessidade em diversos aspectos de pesquisa este tipo de domínio acabou se adaptando para os meios informatizados, como sensores, PDA's e *smartphones* (WAKU *et al.*, 2015).

2.5.1.1. Mecanismos de coleta de dados

É possível perceber o impacto drástico que recursos tecnológicos mais novos e computacionalmente mais poderosos conseguem causar dentro do escopo de coleta de dados. A otimização de processos, redução de custos, o aumento das características de confidencialidade integridade, disponibilidade, autenticidade das informações e a não

restrição a ambientes de controlados são algumas das vantagens oriundas da utilização de ferramentas informatizadas mais avançadas (BONI *et al.*, 2006).

Efetivamente a coleta de dados pode ser executada utilizando-se tecnologias munidas de capacidades de computação móvel, ou tecnologias que forneçam funções de monitoramento remoto. Sendo assim é notória a divisão entre o domínio de coleta de dados e seu subdomínio de coleta de dados móveis. O efeito prático desta divisão é que comumente, sistemas de coletas de dados que são utilizados para monitoramento remoto são implementados com sensores e/ou redes de sensores, apresentando pouca ou nenhuma característica de computação móvel (VASILESCU *et al.*, 2005; TIAN, GAO e ZHOU, 2008; SILVA, ANDRADE e MARIN, 2008; JI *et al.*, 2010; ZHAO, ZHANG e MENG, 2012; SANT'ANNA, 2014).

Em função da utilização da técnica de coleta de dados em várias áreas do conhecimento é comum se deparar com abordagens mais rudimentares desta metodologia, sendo a utilização de questionários de papel uma das mais comuns. Todavia este tipo de implementação simplista de coleta de dados acaba por se tornar obsoleta quando colocados em perspectiva aspectos como custo de processo e processamento do montante de dados coletados. Para contrabalançar os altos custos oriundos da coleta e processamento de dados realizados com questionários de papel, surgem os sistemas de coleta de dados móveis que têm como meta realizar ao mesmo tempo a coleta e processamento dos dados evitando a manipulação física dos mesmos, além de apresentarem custos e esforços operacionais reduzidos (FREITAS FILHO, 2014).

2.5.2. Sistemas de Coleta de Dados Móveis

Santos *et al.*, (2013) elaboram sobre o processo de rápida evolução do mercado de comunicação pessoal, elicitando como causas para tal evolução a queda dos preços dos aparelhos, a alta capacidade dos dispositivos móveis disponíveis, por fim o advento de novas tecnologias para a confecção de aplicações para tais dispositivos. Como consequência deste processo, surge a coleta de dados móvel.

É possível ter uma definição um pouco mais consolidada para sistemas de coletas de dados móveis, Freitas Filho (2014) explica que ao realizar a união de elementos bem

definidos dentro do contexto de coleta de dados é possível se chegar ao conceito de sistemas de coletas de dados móveis. Os elementos definidos são interfaces coletoras de dados que funcionem em dispositivos mobile, infraestrutura de redes sem fio, servidores de banco de dados que coletam e transferem os dados para os repositórios de dados centrais e os repositórios de armazenamento de dados. Este conjunto de elementos conferem alguns desafios significativos para serem levados em conta durante o processo de design e implementação da arquitetura de tais soluções.

Santos *et al.*, (2013) afirmam que os procedimentos de manipulação de informações são simplificados uma vez que estas já estejam armazenadas em meio digital. Sendo assim aplicações que realizam coletas de dados móveis são particularmente interessantes, uma vez que simplificam e aceleram o processo de coleta de dados ao eliminar o processo de digitalização das informações e poderem efetuar a disponibilização dos indicadores desejados em uma série de interfaces relevantes ao negócio logo após a coleta de dados ter sido realizada.

2.5.2.1. Computação móvel

Conforme mencionado anteriormente o processo de coleta de dados é beneficiado de diversas maneiras ao se utilizar os recursos de computação móvel. Este tipo de abordagem ocasiona uma especialização do domínio de aplicação pertinente à arquitetura de referência que será proposta por este trabalho. O domínio de aplicação de coleta de dados passa então a ter como seu subdomínio a coleta de dados móveis. Em virtude de características particulares deste domínio provenientes do contexto computação móvel faz-se necessário abordar brevemente estes aspectos.

Como pode se observar, o despontar de dispositivos móveis como ferramentas informatizadas providas de significativo poder de processamento, baixo custo e versatilidade de negócio, acaba por despertar o interesse de instituições acadêmicas e empresas, fazendo-as ponderar a respeito da relevância destas ferramentas, uma vez que sua utilização pode auxiliar no desenvolvimento de novas tecnologias, aperfeiçoamento de processos, diminuição de custos, refinamento de sistemas, agregando mais contribuições acadêmicas e/ou aumentando o valor de negócio.

O paradigma denominado computação móvel emerge como uma nova abordagem em relação aos estudos e a utilização de aplicações distribuídas, tendo como foco a capacidade de deslocamento de elementos pertencentes a tais sistemas e o desenvolvimento de novos métodos de interconexão *wireless*. Este modelo é motivado pela vasta quantidade de dispositivos portáteis disponíveis (BARBOSA *et al.*, 2007).

Freitas Filho (2014) elege mobilidade, *handoff*, adaptação e desconexão como sendo as características quintessências da computação móvel. Estas características existem em função do ecossistema wireless disponível para o seu funcionamento, o qual corrobora para a qualidade reduzida, menor largura de banda, frequentes desconexões e probabilidade maior de erros, aumentando a latência de retransmissores, atraso nas retransmissões, etc.

Tabela 4 - Características da computação móvel

(Continua)

Característica	Descrição
Mobilidade	Princípio primordial que serve como orientador para o processo de desenvolvimento de novas tecnologias e métodos para o âmbito de computação móvel. Por definição compreende a habilidade de deslocamento espacial em uma rede, o que garante a possibilidade de movimentação aos dispositivos.
Handoff	Processo transparente ao usuário que busca manter a conectividade de uma rede fixa com dispositivos móveis que estão executando algum processo transacional, independentemente de seus deslocamentos pela rede. Estes deslocamentos podem ocorrer dentro da célula de cobertura de uma estação-base, ou entre duas células de cobertura vizinhas pertencentes a estações-base próximas.

Tabela 4 - Características da computação móvel

(Conclusão)

Característica	Descrição
-----------------------	------------------

Adaptação	<p>Garantir a entrega de resultados de sistemas esperados, mesmo que as condições de funcionamento não sejam as ideais. É possível dividir as políticas de adaptação em alguns tipos, a adaptação de unidade, adaptação de sistema e adaptação ciente da aplicação. Estas classificações referem-se ao grau de responsabilidade de adaptação conferido as partes integrantes do sistema. No primeiro tipo há o completo repasse de responsabilidade adaptativa para os sistemas móveis individuais. Já no segundo tipo responsabilidade de adaptação é voltada para todo ecossistema computacional. Por fim, as adaptações cientes da aplicação estabelecem um meio termo entre o primeiro e o segundo tipo, em que as unidades móveis podem gerenciar sua adaptação, porém o sistema como um todo supervisiona e gerencia os recursos presentes em sua estrutura.</p>
Desconexão	<p>Impossibilidade de comunicação dos elementos móveis de um sistema caso haja falha em sua rede. Caracterizadas como voluntária se as capacidades de comunicação sem fio do dispositivo são propositalmente desativadas por algum motivo, e forçada se a comunicação é interrompida espontaneamente, sem que haja intenção de tal interrupção, ela recebe o nome de desconexão. Existem três estados que podem ocorrer durante as desconexões, o estado de <i>hoarding</i>, que é a replicação ou repasse dos dados para um banco local. O estado de operações locais é a execução de operações com os dados existentes no aparelho durante desconexão, ou a criação de uma fila de execuções futuras para o momento de re-conexão, criada mediante a ausência de dados no dispositivo. Por último a reintegração diz respeito a sincronização dos dados após a re-conexão.</p>

Fonte: Adaptado de Freitas Filho, 2014.

2.5.2.1.1. Arquiteturas para o domínio móvel

2.5.2.1.1.1. Cliente/Servidor

Arquitetura composta de sistemas que solicitam serviços para um ou mais sistemas computacionais distintos. Os sistemas solicitantes de serviço são intitulados clientes, enquanto que os sistemas provedores de serviço são chamados de servidores. Em virtude de sua estrutura, esta arquitetura pode fornecer disponibilidade e escalabilidade, uma vez que a porção servidora pode ser dividida em sub-servidores comunicáveis entre si, que desempenham funções distintas para manter a execução do sistema, como por exemplo, suporte a serviços, processamento de requisições e replicação de dados. No que tange o domínio móvel, a comunicação desta arquitetura é feita de forma semi-cabeada, onde o cliente se comunica através de sinal de rádio com um servidor, que por sua vez, realiza seus processos através de rede cabeada fixa. Dentro deste contexto, na grande maioria das vezes, os sistemas fixos compõem a porção servidora, enquanto que dispositivos móveis são a porção cliente, podendo emular algumas funcionalidades servidoras para continuar seu funcionamento mediante uma desconexão (CUNHA *et al.*, 2003).

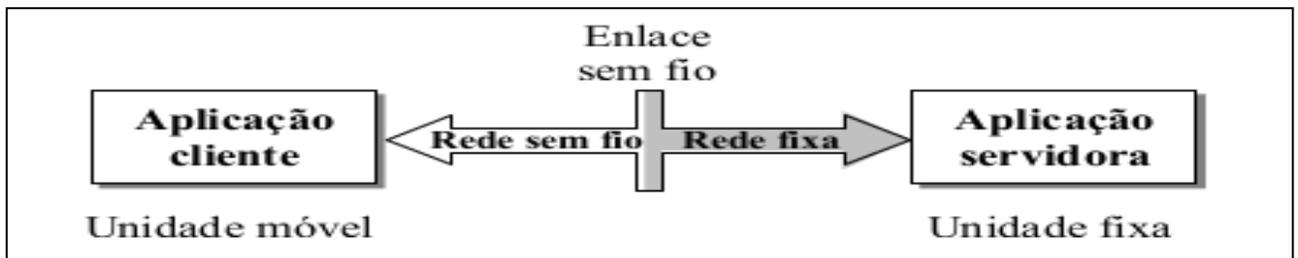


Figura 03 - Arquitetura cliente/servidor. Fonte: Cunha *et al.*, 2003.

2.5.2.1.1.2. Cliente/Agente/Servidor (C/A/S)

Questões como baixa velocidade de comunicação, falta de confiabilidade e desconexão são inerentes da utilização da arquitetura cliente/servidor em ambientes móveis sem nenhuma adaptação. Para que esta abordagem possa ser efetiva para o domínio móvel é preciso desenvolver estratégias de tratamento de desconexões e comunicações não confiáveis e não estáveis entre servidores e clientes. Objetivamente, esta arquitetura é uma extensão da arquitetura cliente/servidor que suporta momentos

de desconexões, de forma que ao se realizarem conexões entre o cliente e o servidor ambos realizam atualizações (ITO, 2001).

Cunha *et al.*, (2003) define que em uma arquitetura (C/A/S) os agentes são mecanismos unitários de conexão que residem na rede fixa ou na rede móvel, desempenhando ações de aplicação ou comunicação, necessárias para o funcionamento do sistema. Devido ao acesso de canais de comunicação de alta velocidade e funcionalidades extra, os agentes permitem otimização dos processos de clientes e servidores, diminuindo o tempo de resposta de operações remotas e assumindo algumas responsabilidades do servidor. É importante ressaltar que devido à limitação operacional dos agentes esta arquitetura é voltada para clientes que dispõem de meios computacionais limitados e não demandam muitas operações.

Agentes são aplicações de *middleware* que proporcionam maneiras cômodas de alocar informações que serão alvo de permuta com o cliente, podendo desempenhar tarefas, efetuar consultas e reaver respostas voltadas para um cliente. Em virtude destas características a abordagem cliente/agente/servidor é considerada muito eficiente para a utilização de sistemas do domínio móvel (ITO, 2001).

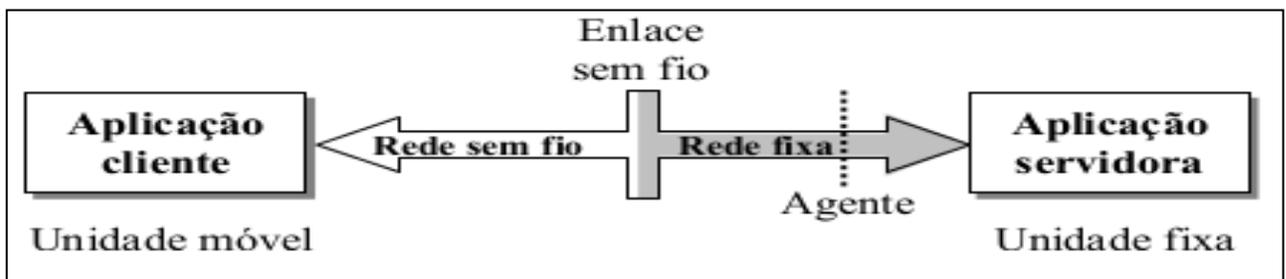


Figura 04 - Arquitetura cliente/agente/servidor (C/A/S). Fonte: Cunha *et al.*, 2003.

2.5.2.1.1.3. Cliente/Interceptor/Servidor (C/I/S)

Em razão da restrição da capacidade operacional das aplicações cliente apresentada pela arquitetura cliente/agente/servidor, fez-se necessário o desenvolvimento de mais uma adaptação da arquitetura cliente/servidor. Esta adaptação tem como intuito fornecer uma estratégia para sistemas que exigem aplicações móveis com grande poder computacional, armazenamento e quantidade de bateria (ITO, 2001).

Para que os subsistemas móveis possam ter maior poder de processamento, armazenamento e autonomia de energia é adotada a política de dois agentes para uma comunicação, de forma que um agente atua do lado do cliente e um atua no lado do servidor, trabalhando em conjunto para orquestrar a comunicação entre os subsistemas móveis e os subsistemas fixos. A utilização da política de divisão do agente proporciona uma melhor comunicação, um melhor tratamento de desconexões e um refinamento do processo de compressão dos dados que serão transmitidos. Apesar das vantagens proporcionadas por esta arquitetura ela também oferece algumas desvantagens, das quais a mais significativa é a necessidade de processamento dos clientes e servidores para cada aplicação. (FREITAS FILHO, 2014).

Cunha *et al.*, (2003) sustenta que devido aos fatos dos agentes controlarem o fluxo de entrada e saída das informações compartilhadas pelas aplicações clientes e servidoras, desempenharem aprimoramentos no processo de comunicação de dados e funcionarem como um portão para as aplicações clientes e servidoras, a sua alcunha é adaptada para melhor representar seu papel, sendo alterada para interceptador.

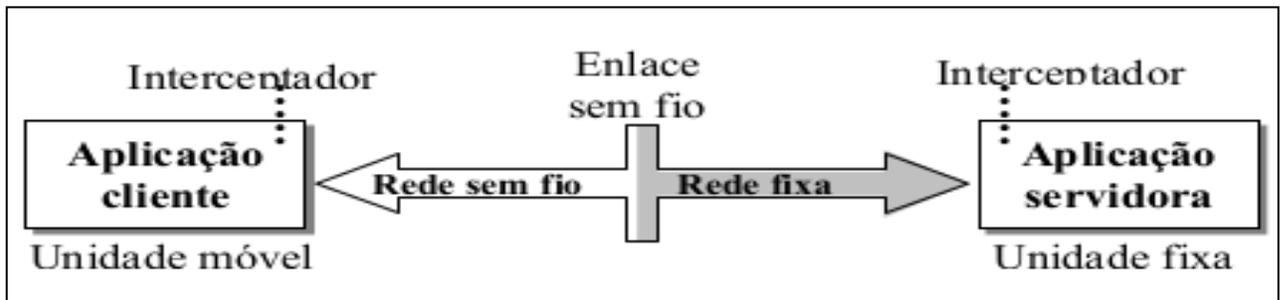


Figura 05 - Arquitetura cliente/interceptador/servidor (C/I/S). Fonte: Cunha *et al.*, 2003.

2.5.2.1.1.4. Ponto a ponto

Diferentemente da arquitetura cliente/servidor e suas duas adaptações, a arquitetura ponto a ponto propõe uma estrutura em que subsistemas clientes podem assumir o papel de um subsistema servidor, o que permite que comunicações possam ser realizadas entre duas ou mais aplicações móveis, caso o servidor principal esteja incomunicável (ITO, 2001).

Cunha *et al.*, (2003) ratifica que é possível notar o comportamento de elementos clientes emulando aspectos de elementos servidores e vice-versa através da transposição da arquitetura ponto a ponto de redes fixas para o domínio móvel, o que acaba por causar uma duplicidade de papéis nos subsistemas de um sistema que utilize a arquitetura. Esta arquitetura necessita um meio de comunicação entre seus componentes, uma vez que para que ela precisa um forte acoplamento das aplicações clientes e quase nenhum acoplamento com as unidades fixas.

Ito (2001) também ressalta a aplicação desta arquitetura afirmando que o seu contexto é em sistemas que possuam um grupo de aplicações móveis clientes muito acopladas e que possuem uma conexão limitada ou nula com o servidor. Destacando também os altos custos oriundos da falta de comunicação entre as aplicações móveis, ou caso uma aplicação queira se conectar com o servidor que está distante.

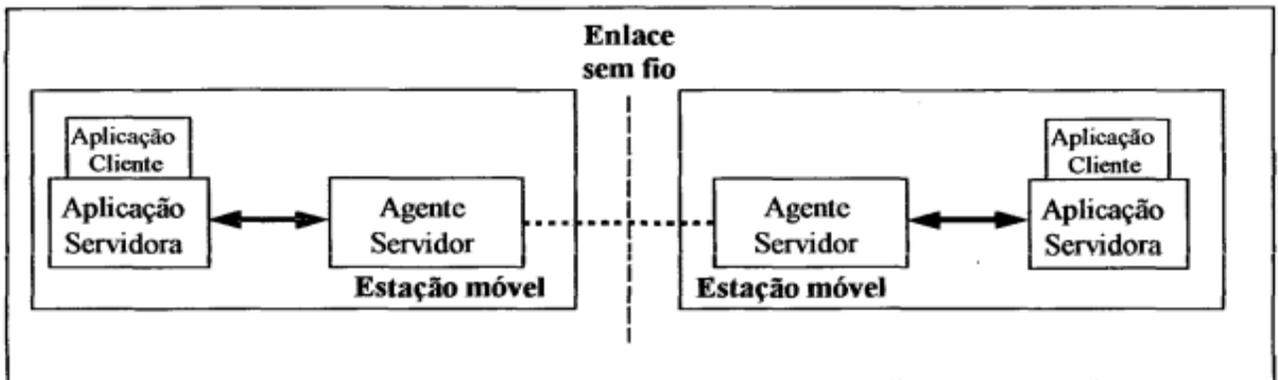


Figura 06 - Arquitetura ponto a ponto. Fonte: Ito, 2001.

2.5.2.1.1.5. Agentes Móveis

Existem duas estruturas utilizadas para prover comunicação dentro do contexto da arquitetura cliente/servidor, são estas *Remote Procedure Calls* (RPC) e troca de mensagens. Para que a arquitetura cliente/servidor possa ser implementada no domínio móvel é necessário adaptar o funcionamento das *Remote Procedure Calls*, tornando-as assíncronas, o que permite tolerâncias as recorrentes desconexões. Tomando como base a utilização de RPC, a arquitetura de agentes móveis une a capacidade de envio de requisições cliente e servidor com estruturas que executam tarefas remotamente

chamadas agentes. Estas estruturas são compostas por um estado, dados e instruções (CUNHA *et al.*, 2003).

Freitas Filho (2014) sustenta que dentro do escopo de agentes móveis, agentes são programas que podem ser enviados de uma unidade emissora para uma unidade receptora, para que possam executar tarefas na unidade receptora. Estas estruturas são autônomas a partir do momento em que são enviadas entre computadores, podendo transportar-se para outras unidades, interagir ou gerar novos agentes. Ao atingir a unidade receptora a estrutura entra em uma fila de execução de agentes, informando a unidade sua execução para as unidades relevantes.

Tabela 5 - Características da arquitetura de agentes móveis

Característica	Descrição
Objetos passantes	Transferência dos objetos compostos de estado, dados, instruções e caminho para o servidor.
Assincronismo	Refere-se ao <i>thread</i> de execução do agente móvel.
Interação local	Interação com objetos não móveis.
Desconexão	Capacidade de executar tarefas desconectado, aguardando uma reconexão caso seja necessário enviar um agente.
Execução paralela	Capacidade enviar agentes para várias unidades diferentes afim de processar tarefas em paralelo.

Fonte: Adaptado de Ito, 2001

2.5.2.2. Modelos de falha para o domínio móvel

Para que se possa discutir e compreender aspectos relativos a falhas específicas para aplicações pertencentes ao domínio de aplicação móvel, faz-se necessário lançar mão de alguns conceitos prévios, porém talvez os mais importantes sejam os conceitos de falha e tolerância à falhas.

Pullum (2001) define que erros são elementos que compõem o estado do sistema, sendo eles derivados de falhas (*faults*), ou seja, uma falha (*fault*) pode ser compreendida como o motivo ou circunstância pelo qual o erro ocorreu. Para que uma falha (*fault*) gere um erro ela deve ser ativa. Podendo ser identificadas ou hipotéticas, as falhas (*faults*)

podem ser interpretadas como repercussões da má disponibilização de serviços de outros sistemas. Também é estabelecido que erros podem ocasionar falhas (*failures*), estas ocorrem quando a especificação dos serviços do sistema comparada em relação ao serviço disponibilizado começa a demonstrar disparidades.

Scott et al., (1984) auxilia a instituir a definição clássica de tolerância a falhas como a localização e a recuperação de falhas (*faults*) advindas de software através da implementação de um conjunto de técnicas de software.

Waku *et al.*, (2015) propõem que ao ser possível garantir a disponibilidade dos serviços de uma aplicação que seguem as de especificações de serviços estabelecidas no projeto independentemente da ocorrência de falhas pode-se dizer que a aplicação é tolerante a falhas. Para que um sistema de software possa alcançar esta capacidade é necessário fazer uso de diversas técnicas que têm como meta possibilitar que a aplicação possa suportar falhas pertencentes ao sistema, inerentes ao negócio, aos elementos arquiteturais, ou às tecnologias empregadas no projeto.

Pullum (2001) chega a conclusão que ao implementar tolerância de falhas (*faults*) de software, almeja-se impedir ou impossibilitar conjunturas de falhas (*failures*), permitindo que os sistemas de software sejam capazes de serem submetidos à falhas (*faults*) cujas ocorrências são conhecidas quando erros são identificados.

Waku *et al.*, (2015) detalha algumas técnicas de tolerância a falhas se do estas:

- **Redundância:** Cópias de contingência dos recursos principais, as quais geralmente são ativadas mediante a falha de um recurso principal. Vários tipos de redundância também são evidenciados como as redundâncias de hardware e de dados, que são respectivamente a utilização de hardwares extra em caso de falha do equipamento e o armazenamento duplicado do mesmo dado. É importante ressaltar os tipos de redundâncias de software, sendo estas redundâncias de armazenamento de dados e em estrutura de dados que são respectivamente o armazenamento replicado em locais diferentes e o uso de diferentes formatos de dados.
- **Manejo de exceções:** Perícia que confere capacidade para identificar erros, e após sua identificação, tratá-los e retornar ao fluxo de operações costumeiro.

Waku *et al.*, (2015) conceitua o agrupamento das falhas (*faults*) mais relevantes e preponderantes a um domínio de aplicação específico, organizadas dentro de uma

estrutura de disposição hierárquica de classes de falha (*fault*) como modelo de falha para modelo de domínio, dividindo-as em três classes:

- **Nível de bateria baixo:** Compromete a disponibilidade do sistema. No contexto de coleta de dados apresenta as falhas (*faults*) relativas coletas de dados, sincronização, importação e exportação indisponíveis.
- **Falha de serviço de coleta de dados:** Erros que congelam o sistema ou processam dados de forma incorreta comprometendo respectivamente a disponibilidade e confiabilidade do sistema. Falhas (*faults*) relativas a esta classe são a parada de execução do sistema devido a uma falha, o sistema não conseguir acessar os dados, a geração de bancos de dados inválidos, o sistema processa dados de forma errada.
- **Perda de dados:** Suscetível a ocorrer em aparelhos que não foram projetados para serem usados em condições adversas. Falhas (*faults*) relativas a esta classe são a quebra do dispositivo e o corrompimento do banco de dados.

2.5.2.3. Banco de dados para o domínio móvel

Ao compreender a computação móvel como parte integrante de uma nova abordagem do processo de coleta de dados, também deve-se atentar para as novas particularidades e adversidades arquiteturais provenientes desta agregação. A utilização de unidades de coleta de dados móveis que dispõem de bancos de dados próprios e exportam, importam e sincronizam informações com um servidor central a fim de disponibilizar informações, é uma das aplicações mais comuns de coletas de dados móveis. Pode-se constatar que no que tange o projeto da base de dados de tais aplicações tem-se como objetivo principal a garantia da consistência de informação entre os diferentes nodos de coleta de informação e a base de dados matriz (FREITAS FILHO, 2014).

Sistemas que utilizam bancos de dados móveis são a escolha perfeita para modelos de negócio que necessitem de constante deslocamento, pois estas soluções permitem a utilização de recursos informatizados em diversos cenários e circunstâncias. Devido ao grande poder computacional de dispositivos móveis disponíveis no mercado, sua vasta disponibilidade e a redução do preço de tais aparelhos, é comum verificar um aumento de soluções voltadas para disponibilização de dados em tempo real, que

necessitam de uma alta mobilidade e que acessem remotamente alguma base de dados (ITO, 2001).

Para que sistemas de software se utilizem de preceitos de computação móvel em favor do negócio, empregando estratégias de computação móvel a sua estrutura de base de dados, são necessárias alterações nas diretrizes de gerência e nos instrumentos que proporcionam a consistência dos dados. Limitações como largura de banda, as incongruências de conexão frequentes e o tempo de bateria limitado são só algumas das considerações que necessitam ser levadas em consideração (FREITAS FILHO, 2014).

Ito, Ferreira e Sant'Ana (2003) afirmam que existe uma vasta gama de arquiteturas possíveis para este determinado escopo, porém, determina uma arquitetura básica para sistemas de bancos móveis composta por três elementos fundamentais, sendo estes:

- **Plataformas de aplicação:** Dispositivos móveis dispoendo de poder computacional e capacidade de comportar um banco de dados interno, o qual contém uma parcela do banco de dados principal. Estes dispositivos atuam como as principais interfaces entre o sistema e o mundo real garantindo o acesso dos colaboradores às informações pertinentes ao negócio. Tais instrumentos possuem diversas camadas de gerência de software, sendo algumas nativas, ou seja, camadas que coordenam o funcionamento operacional do aparelho como comunicação, nível de sinal e de bateria, topologia dinâmica, até camadas de aplicação sendo estas abstrações inerentes ao modelo de negócio o qual se está trabalhando, manejo da base de dados e comunicação com o servidor.
- **Host servidor:** Computador poderoso que serve como estabelecimento principal para o banco de dados. Sua função é fornecer uma opção estável e segura para agregar as informações a fim de disponibilizá-las para que diferentes sistemas de software, incluindo o próprio sistema, possam utilizá-las.
- **Comunicações entre host e plataformas de aplicação:** As plataformas de aplicação efetuam tarefas de consultas, importação, exportação e sincronização de dados com o host servidor para mantê-lo atualizado. Devido ao domínio móvel a troca de informações do servidor com as plataformas de aplicação ocorrem em sua maioria através de em redes distintas, em períodos de tempo irregulares e curtos, com diversas interrupções.

É estabelecido por Freitas Filho (2014) que, a utilização de bancos de dados móveis torna-se um desafio para o estabelecimento de uma arquitetura específica para o modelo de domínio de coleta de dados móveis, devido a herança de características que são intrínsecas a computação móvel. Estas características são *caching*, difusão de dados, transações e recuperação de falhas.

As políticas de *caching* e de Difusão de dados são estratégias específicas para melhoria de desempenho de acesso de dados móveis, tais políticas almejam a diminuição do tráfego de informações na rede para o acesso de informações contidas no servidor. A estratégia de *caching* busca elevar a chances de disponibilidade de dados possivelmente reutilizáveis em cache para as entidades móveis, o que reduz o tempo de respostas às consultas. Já a técnica de difusão de dados constitui-se de propagação estruturada de mensagens de uma plataforma móvel para todas as outras plataformas móveis dentro da área de alcance de *broadcasting* (ITO, FERREIRA e SANT'ANA, 2003).

Outro importante ponto para o desenvolvimento de sistemas que utilizem bancos de dados móveis é a gestão de transações móveis, esta área tem se tornado alvo de estudos devido aos diversos desafios provenientes da sua implementação. Alguns dos fatores pertinentes a este contexto são a mobilidade entre regiões diferentes que os dispositivos móveis possuem, a largura de banda reduzida, as desconexões frequentes e a natureza de transações de longa duração (*long-lived transaction*). Todos estes fatores acarretam um aumento de complexidade no que tange o processamento de transações móveis, tornando-as mais suscetíveis a erros (LI e WANG, 2009).

Devido às constantes desconexões e operações parciais realizadas por aplicações móveis, as transações utilizadas neste contexto devem considerar estas características, evitando o escopo atômico, isolado, durável e consistente. Um dos aspectos a serem levados em conta é o deslocamento do dispositivo móvel, uma vez que este tem controle sobre a realização da transação, dividindo-a e executando-a sequencialmente. Além disto, como consequência do contexto móvel sem fio, transações passam a consultar bases de informação heterogêneas mais comumente, tendo seu tempo de duração estendido, e com maiores chances de imprecisão. Alguns conceitos norteiam as abordagens necessárias para o trabalho com bases de dados móveis, em via de regra, cada transação pode ser abstraída como um bloco de transações parciais, além disto, se

caso uma desconexão ocorra durante uma transação, a aplicação deve ser capaz de aplicar ao banco as consequências da transação não concluída no momento de reconexão (ITO, FERREIRA e SANT'ANA, 2003).

Aplicações que utilizam bancos de dados móveis necessitam de uma atenção especial em relação as políticas que corroboram para a proteção de integridade da base de dados, evitando a perda de informação, uma vez que a conjuntura móvel é suscetível a erros provenientes do mal funcionamento dos componentes de hardware e de software durante a execução de transações. Uma estratégia para manter a coesão da base de dados apesar da ocorrência de falhas é a utilização de checkpoints, esta estratégia salva o último estado do banco, o qual pode ser retomado caso alguma falha no sistema ocorra. Outra política para auxiliar as transações móveis é a utilização de um local fixo enquanto o processamento das transações está ocorrendo. Estas políticas de recuperação também devem ser capazes de identificar de forma célere o momento de ocorrências dos erros, além de prover uma descrição adequada dos erros (FREITAS FILHO, 2014).

2.5.2.4. Sincronização

Analisando do ponto de vista da computação móvel, o processo de sincronização de dados refere-se ao envio de mudanças realizadas em um banco de dados de um de um dispositivo móvel para um banco de dados principal em servidor remoto fixo. Este cenário é composto, na grande maioria das vezes, por aplicações móveis que obtém de um servidor principal as informações necessárias para o seu funcionamento *off-line*. Após a realização das tarefas *off-line* é esperado que existam disparidades entre as bases de dados das unidades móveis e do servidor fixo, sendo necessária resolução de conflitos e a manutenção da equivalência de dados entre as bases móveis e as bases principais, a fim de evitar inconsistências de dados (REN e SONG, 2002).

A utilização de sistemas que fazem uso de bases de dados distribuídas e heterogêneas vem se tornando estratégia comum para diferentes organizações, isto se deve muito pelo refinamento de tecnologias e o aumento de negócios corporativos. Neste contexto também se fazem presentes alguns desafios como transparência na experiência do usuário, manutenção da consistência dos dados e prevenção de perda de dados. Tais

metas podem ser atingidas com a utilização de estratégias e tecnologias de sincronização (ZHANG e LU, 2010).

Um sistema distribuído deve possuir algumas características que permitam a sua operação plena, provendo transparência aos usuários que o utilizam. Algumas características cruciais para o correto funcionamento de aplicações distribuídas são a disponibilidade e velocidade de acesso aos dados. Tais atributos podem ser alcançados de algumas formas diferentes, como por exemplo, a cópia de um servidor central em diversos servidores diferentes auxilia o acesso aos dados, de forma que uma consulta feita de um dispositivo móvel a um servidor com problemas pode ser redirecionada para algum outro servidor estável que contenha a informação alvo da estação móvel. Também se pode trabalhar com a ideia de cópias controladas, inseridas nas estações móveis com o intuito de evitar ao máximo a quantidade de conexões desnecessárias e reduzir o número de requisições feitas para o servidor pelas aplicações móveis, o que conseqüentemente aumenta a velocidade do acesso a informação e o tempo de resposta dos sistemas móveis. Esta estratégia de cópia de dados recebe o nome de replicação (CUNHA *et al.*, 2003).

A operação de réplica segundo Zhang e Lu (2010) tem o objetivo de igualar operações de adição, modificação e deleção de dados de uma aplicação para outra, através de consecutivas alterações incrementais dos dados sincronizáveis. A utilização desta abordagem provê melhor eficiência e menor tráfego na rede.

Ito (2001) reforça que a propagação assíncrona das transações executadas em uma base de dados para outra base de dados de forma serial é o processo que caracteriza a replicação de dados. A utilização de um método assíncrono resulta na possibilidade de armazenamento e envio, armazenando as informações replicáveis em um banco de dados até o momento da replicação. Em relação a ordem das transações, estas são executadas de forma serial, com o objetivo de evitar inconsistências entre o servidor e as unidades móveis. Além disto, também se pode utilizar a técnica de replicação de dados total, que é superior à estratégia de *caching* uma vez que confere melhor desempenho da unidade móvel devido à utilização plena dos dados do dispositivo e aperfeiçoa a confortabilidade, aumentando o número de opções para solicitação de dados.

Cunha et al., (2003) afirma que as réplicas podem ser repassadas ao servidor com uma série de primitivas de sincronização, e que estes mecanismos suportam aspectos inerentes do domínio móvel como desconexões frequentes, conexões esporádicas e de pouca confiança, por melhorarem a disponibilidade dos dados.

Ren e Song (2002) ressalta a importância da existência de protocolos de sincronização que possam servir como interface para uma vasta gama de aparelhos em diferentes tipos de redes, devido o número crescente de dispositivos computacionais e os diferentes tipos de informação e formas de armazenamento que estes dispositivos utilizam.

Devido à natureza complexa do ecossistema de computação móvel, tem-se a necessidade de uma boa gestão de dados entre as unidades móveis pertencentes ao sistema e o servidor central de informações. Um dos aspectos importantes pertencentes a gerência de funcionamento coordenado de sistemas distribuídos é a sincronização dos dados existentes nos dispositivos móveis e no servidor central. Este aspecto é capaz de manter a consistência de dados de maneira eficiente. Uma vez que as unidades móveis funcionam de modo independente, o estado de seus bancos de dados será muitas vezes diferente do estado da base de dados principal, ocasionando conflitos ao se realizar uma conexão com a base de dados principal. Sendo assim pode-se definir como ponto focal dos estudos de sincronização a investigação de modos de identificar a consistência dos dados e corrigir inconsistências, dividindo-se em duas áreas: protocolos e tecnologias (LI e LI, 2012).

Sun e Liu (1996) afirmam que os protocolos de sincronização são os responsáveis por estruturar como as tarefas e sub tarefas são executadas, de modo que sejam respeitadas definições de precedência, além de garantir que possa ser realizada a análise do agendamento de tais processos.

Para Agarwal, Starobinski e Trachtenberg (2002) protocolos de sincronização de dados são mecanismos capazes de verificar rapidamente as mudanças realizadas em um banco de dados pertencente a uma unidade móvel, resolver conflitos e repassar as alterações realizadas na base de dados móvel para os outros dispositivos integrantes do sistema que necessitam da sincronia dos dados. Estes mecanismos são particularmente necessários em sistemas que trabalham com dados distribuídos, ou seja, sistemas que

necessitem que tarefas feitas durante condições não ideais de execução dos sistemas móveis possam ser corretamente repassadas para os servidores principais.

O ecossistema de dispositivos computacionais móveis evolui a cada dia, este rápido crescimento disponibiliza uma diversidade de sistemas móveis que funcionam em plataformas distintas como celulares, PDA's, computadores de bordo, *lap-tops*, entre outros. Estas plataformas contribuem para melhorias significativas na execução de tarefas diárias desempenhadas por seus usuários, porém a utilização de tecnologias específicas para uma plataforma ou ecossistema de software pode ocasionar a impossibilidade de comunicação. Deve-se levar em conta que as diferentes plataformas de software para dispositivos móveis pertencem ao domínio móvel, em virtude disto é possível se deparar com cenários em que seja necessário que sistemas pertencentes a ecossistemas de software diferentes necessitem trabalhar em conjunto afim de gerar valor de negócio. Mediante ao cenário proposto um ambiente móvel que não tem mecanismos que permitam a comunicação de aplicações díspares pode ocasionar dificuldade ou impossibilidade de acesso a informações e conseqüentemente, perda de valor para o negócio ou usuário (REN e SONG, 2002).

Mecanismos que permitem a comunicação de aplicações pertencentes a ecossistemas isolados umas com as outras são um ponto determinante para a integração de serviços que tem a capacidade de gerar valor ao serem utilizados em conjunto. Esta integração permite que dados possam ser acessados e manipulados em diversos dispositivos diferentes. Estes mecanismos precisam considerar a operação em rede sem fio, os diferentes protocolos de transporte, tolerância de dados arbitrários de rede, acesso a aplicações distintas, limitações de recursos do domínio móvel e integrar com tecnologias de comunicação já existentes. O maior desafio enfrentado por estes mecanismos é a utilização de diferentes redes pertencentes ao domínio móvel, uma vez que estas têm propriedades que dificultam esta tarefa. É necessário considerar a latência de rede, o limite da largura de banda, custo alto dos pacotes e confortabilidade de dados e conectividade baixa (FREITAS FILHO, 2014).

Para suprir a necessidade de um protocolo que realize a comunicação de tecnologias distintas, foi desenvolvido no ano de 2001 o protocolo chamado SyncML *Synchronous Mark Up Language*. Este protocolo baseia-se na utilização de XML

eXtended Markup Language e utiliza esta tecnologia para representar suas mensagens, que são registadas como dois tipos específicos, *application/vnd.syncml+xml* e *application/vnd.syncml+wxml*, estas são respetivamente utilizadas para identificação das mensagens do protocolo e identificação de representação binária WBXML (REN e SONG, 2002).

O protocolo SyncML permite a sincronização dos dados de dispositivos em diversos contextos de redes como por exemplo redes fixas, infravermelho cabo e *bluetooth*. Isto é possível pois o protocolo é utilizado como uma linguagem padrão de sincronização que é capaz de minimizar o uso de largura de banda, alta latência, a capacidade de operar com desconexões frequentes (LEE *et al.*, 2002).

Ren e Song (2002) abordam o protocolo SyncML sistematicamente dividindo-o em três partes sendo estas baseadas na sintaxe SyncML :

- **Protocolo de sincronização SyncML:** Determina o tipo de sincronização e o fluxo de mensagens entre o cliente e o servidor SyncML durante uma sessão de sincronização de dados. Resumidamente este módulo coordena as sessões de sincronização.
- **Protocolo de representação SyncML:** Busca estruturar os dados da sincronização.
- **Banda de transporte de mensagens SyncML:** Corresponde a capacidade do protocolo de comunicar-se de forma efetiva e célere através de diversos tipos redes sem fio. Para se comunicar o protocolo provê o transporte de mensagens em redes sem fio através dos protocolos HTTP, WSP e OBEX.

É estabelecido por Lee *et al.*, (2002) que o protocolo SyncML possui tipos de diferentes de sincronização que podem ser realizados em sua estrutura, estes são definidos de acordo com a ordem de conexão entre cliente e servidor.

Tabela 6 - Características da arquitetura de agentes móveis

Tipo de sincronização	Descrição
<i>Two-way Synchronization</i>	Processo referente a troca de informação dos dados que foram modificados. Nesta abordagem cliente e servidor são incumbidos deste processo.

<i>Slow Synchronization</i>	Processo de comparação onde itens de um banco de dados ou mais são comparados uns com os outros campo por campo. Nesta abordagem o cliente repassa toda a base o servidor. Após a análise o servidor envia o resultado ao cliente.
<i>One-way Synchronization from Client Only</i>	Processo de envio das modificações do cliente para o servidor, sendo que o servidor não retorna as alterações para o cliente.
<i>Refresh Synchronization from Client Only</i>	Processo de envio de todo o banco do cliente para o servidor, onde posteriormente é realizada a substituição de toda a base de dados servidora pela base de dados recebida do cliente.
<i>One-way Synchronization from Server Only</i>	Processo de envio das modificações do servidor para o cliente, sendo que o cliente não retorna as alterações para o servidor.
<i>Refresh Synchronization from Server Only</i>	Processo de envio de todo o banco do servidor para o cliente, onde posteriormente é realizada a substituição de toda a base de dados cliente pela base de dados recebida do servidor.
<i>ServerAlerted Synchronization</i>	Processo no qual o servidor sinaliza para o cliente realizar uma sincronização, além de especificar o tipo de sincronização que será necessária.

Fonte: Adaptado de Lee *et al.*, 2002.

3. TRABALHOS CORRELATOS

Para construir uma base de conhecimentos aplicados referentes a um domínio é necessário coletar tais conhecimentos, mais especificamente esta é a primeira etapa proposta pelo processo ProSA-RA, chamada de Passo RA1 - Investigação das fontes de informação. Nesta etapa são verificados diversos provedores de conhecimento relevante para a arquitetura de referência que se propõe construir. Configuram-se como fontes de informações pessoas que lidam com o domínio pesquisado, sistemas de software pertencentes ao domínio de aplicação, publicações, modelos de referência e/ou outras arquiteturas de referência e ontologias de domínio (NAKAGAWA, 2014).

Para poder fundamentar os requisitos sistêmicos e arquiteturais abordados na construção da AR proposta por este trabalho foi realizada a análise de requisitos e elementos arquiteturais de trabalhos pertencentes ao domínio de coleta de dados, esta seção busca comentar sobre este processo de análise de tais aspectos.

3.1. SISTEMAS DE COLETA DE DADOS

Para melhor compreender o contexto em que os trabalhos pesquisados se encontram, fez-se necessário definir a diferença entre sistemas pertencentes ao domínio de coleta de dados e sistemas pertencentes ao subdomínio de coleta de dados móveis. Em virtude desta necessidade este trabalho propôs como escopo do domínio de coleta de dados, sistemas não que utilizem recursos de computação móvel, os quais em sua maioria trabalham com sensores e/ou rede de sensores como interfaces de coleta. Segue-se a análise destes trabalhos.

3.1.1. Sistema repositório de informações agro meteorológicas

O primeiro trabalho que foi analisado, levando em consideração os aspectos arquiteturais apresentados na solução discorrida no mesmo foi o trabalho de Silva, Andrade e Marin (2008), tal publicação tem o objetivo de discorrer de forma abrangente sobre a necessidade, a concepção e o impacto de um repositório de reunião informações agro meteorológicas. Para que tal meta pudesse ser atingida foi necessário conceber uma arquitetura coesa que suprisse os requisitos de coleta de dados exigidos pelo projeto.

Tabela 7 - Requisitos arquiteturais do trabalho de Silva, Andrade e Marin (2008)

Requisito Arquitetural	Descrição
RA - 01	Permitir a disponibilidade do sistema.
RA - 02	Permitir chamadas do servidor controlador para as estações de coleta.
RA - 03	Permitir recuperação de informações do servidor controlador nas estações de coleta.
RA - 04	Permitir conexões das estações através de redes de telefonia.
RA - 05	Permitir a criação de arquivos e diretórios que obedecem a uma estrutura de diretórios pré-definida.
RA - 06	Permitir a escrita dos dados coletados em um arquivo que obedeça a uma estrutura de diretórios.
RA - 07	Permitir leitura automática e regular de estrutura de arquivos criados pelas estações de coleta de dados.
RA - 08	Garantir controle de concorrência.
RA - 09	Garantir serviços de rede robustos independentemente das cargas de rede.
RA - 10	Garantir o controle transacional para o banco de dados.

Fonte: Adaptado de Silva, Andrade e Marin, 2008.

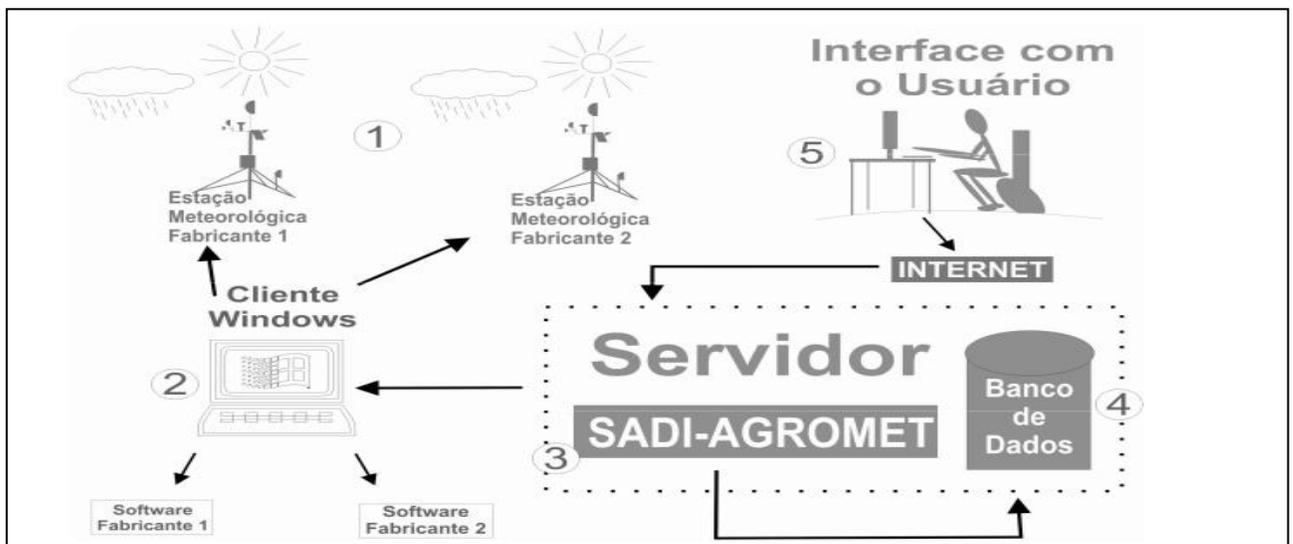


Figura 07 - Arquitetura apresentada por Silva, Andrade e Marin (2008). Fonte: Silva, Andrade e Marin, 2008.

Silva, Andrade e Marin (2008) dividem a sua solução em seis componentes, os detalhes dos componentes da arquitetura são listados abaixo:

- **Componente nº1:** Estações de coleta automática de dados agro-meteorológicos.
- **Componente nº2:** Software desenvolvido pelos fabricantes do sensor de coleta de dados instalados em um terminal Windows, juntamente com um sistema de envio e recebimento de dados baseado em tecnologia Web.
- **Componente nº3:** Servidor central do sistema proposto, o qual é responsável pela recuperação dos dados no banco de dados para a exibição na interface.
- **Componente nº4:** Banco de dados principal que aglomera os dados coletados em campo.
- **Componente nº5:** Interface de disponibilização dos dados coletados para os usuários finais.
- **Componente nº6:** O sexto elemento da arquitetura é a linha pontilhada presente na representação gráfica. Esta demarcação demonstra a divisão entre o sistema web e os elementos externos.

3.1.2. Sistema de aquisição e sincronização de dados sísmicos

Após a análise do trabalho proposto por Silva, Andrade e Marin (2008) foi realizado o mesmo processo de observação de requisitos arquiteturais e dos elementos que compõem a arquitetura do trabalho de Tian, Gao, Zhou (2008). Em sua publicação, Tian (2008) aborda um sistema computacional para coleta de dados sísmicos através da utilização de uma rede de sensores sem fio multicanal.

Tabela 8 - Requisitos arquiteturais do trabalho de Tian, Gao, Zhou (2008)

(Continua)

Requisito Arquitetural	Descrição
RA - 01	Garantir a transmissão síncrona de dados.
RA - 02	Permitir distribuição de dados.
RA - 03	Permitir coleta de dados.
RA - 04	Permitir conversão de dados.
RA - 05	Permitir transformação de dados.
RA - 06	Permitir armazenamento local de dados.

Tabela 8 - Requisitos arquiteturais do trabalho de Tian, Gao, Zhou (2008)

(Conclusão)

Requisito Arquitetural	Descrição
RA - 07	Permitir utilização de dados multicanal.
RA - 08	Permitir gerencia de nível de bateria do módulo de coleta.
RA - 09	Permitir gerencia de localização GPS do módulo de coleta.
RA - 10	Permitir gerencia de conversão de dados no módulo de coleta.

Fonte: Adaptado de Tian, Gao, Zhou, 2008.

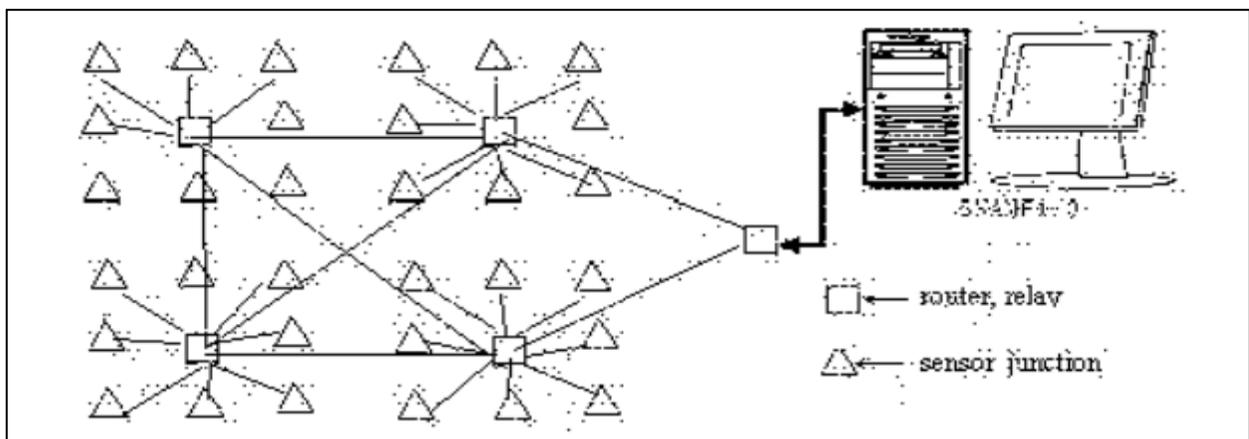


Figura 08 - Arquitetura apresentada por Tian, Gao, Zhou (2008). Fonte: Tian, Gao, Zhou, 2008.

Tian, Gao, Zhou (2008) detalham brevemente os componentes da arquitetura do sistema desenvolvido em seu trabalho dividindo-os da seguinte forma:

- **Servidor:** Componente responsável pelo processamento e armazenamento dos dados coletados.
- **Relay router:** Componente utilizado para direcionar a informação de forma multicanal.
- **Sensor junction:** Componente responsável para coletar e transmitir os dados para o *relay router* para futura sincronização dos dados.

3.1.3. Sistema de coleta de dados sincronizáveis para áreas de mineração

Outra publicação analisada foi realizada por Zhao, Zhang e Meng (2012), esta busca demonstrar a elaboração e o design de um sistema de coleta de dados voltados para gasodutos e oleodutos. É evidenciado na publicação que devido as grandes distâncias que são percorridas pelos oleodutos e gasodutos chineses, ocorre uma complicação no processo de observação e manutenção da infra-estrutura de tais sistemas responsáveis pelo deslocamento e distribuição de fontes de energia, uma vez que sua rota comumente percorre locais ermos, com pouca habitação e provavelmente de difícil acesso. Dentro deste contexto foi proposta a utilização de um sistema de coleta de dados que seja capaz de suprir as informações necessárias, evitando o deslocamento até o local.

Tabela 9 - Requisitos arquiteturais do trabalho de Zhao, Zhang e Meng (2012)

Requisito Arquitetural	Descrição
RA - 01	Permitir a disponibilidade do sistema.
RA - 02	Permitir a disponibilidade das estações de coleta de dados.
RA - 03	Permitir a transmissão dos dados coletados para o servidor através de GPRS.
RA - 04	Garantir controle de processamento de múltiplas requisições.
RA - 05	Garantir o controle transacional para o banco de dados.
RA - 06	Permitir a disponibilização dos dados coletados através de uma interface.
RA - 07	Permitir a autonomia do sensor de coleta de dados em relação ao carregamento.
RA - 08	Permitir a autonomia do sensor de coleta de dados em relação a verificação de sinal.
RA - 09	Permitir o monitoramento das condições do ambiente dos oleodutos e gasodutos.
RA - 10	Prever desastres geológicos.

Fonte: Adaptado de Zhao, Zhang e Meng, 2012.

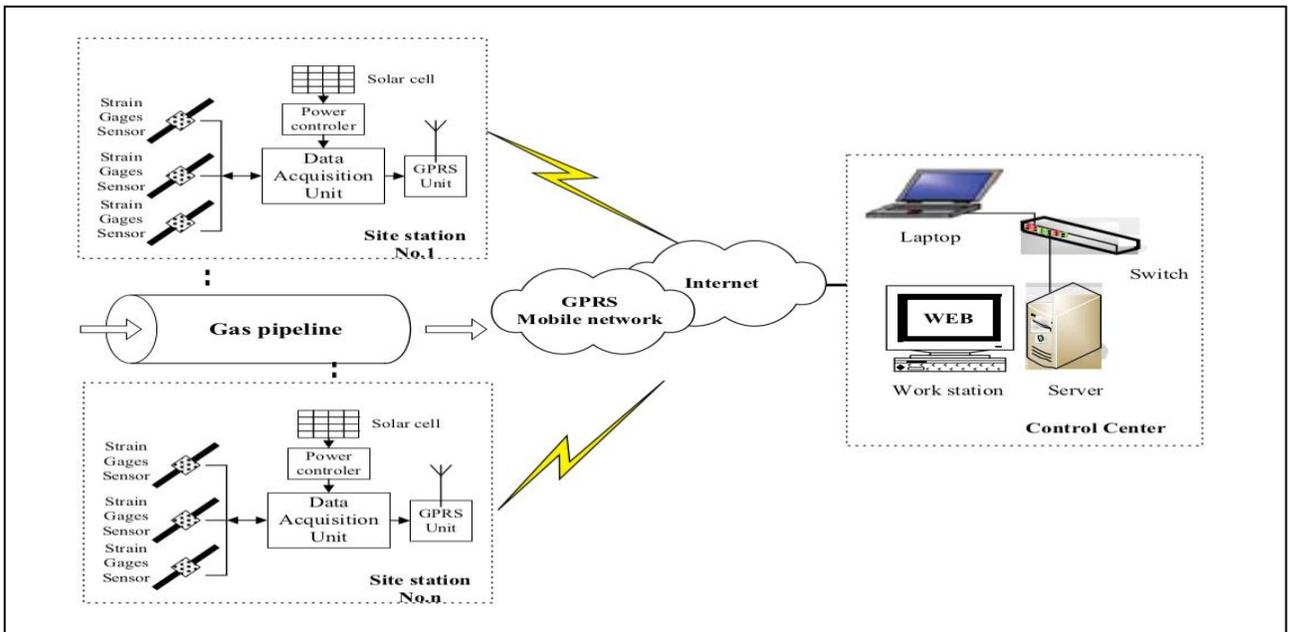


Figura 09 - Arquitetura apresentada por Zhao, Zhang e Meng (2012). Fonte: Zhao, Zhang e Meng, 2012.

A arquitetura do sistema computacional proposto é constituída de duas partes, a primeira são as estações remotas ou *Site stations*, a outra parte é o centro de análise de dados localizado em uma sala de controle. Estes dois elementos integrantes da arquitetura têm respectivamente o papel de realizar a coleta/transmissão dos dados de campo e agrupar todos os dados que são coletados e enviados de diferentes estações remotas.

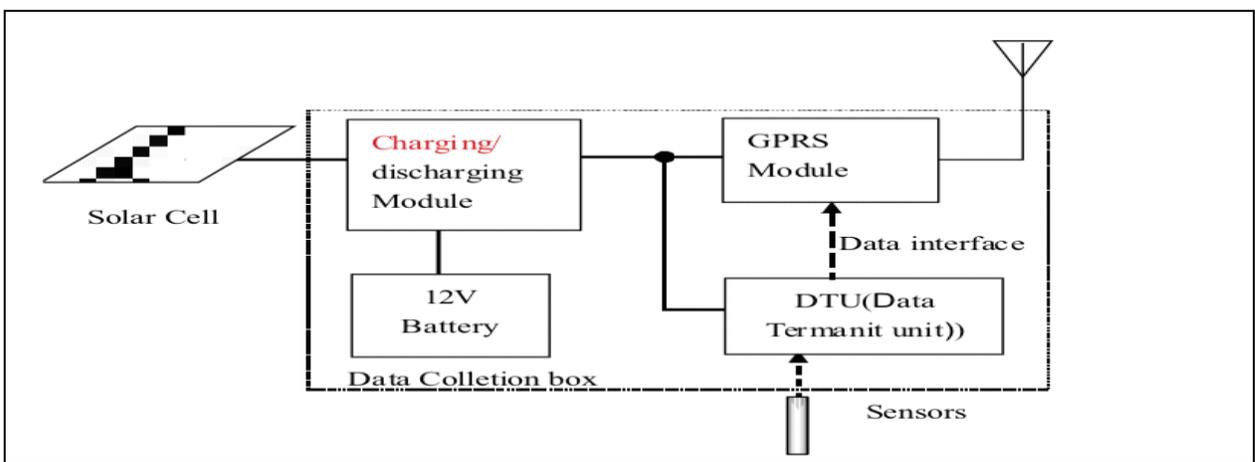


Figura 10 - Representação do sensor utilizado por Zhao, Zhang e Meng (2012). Fonte: Zhao, Zhang e Meng, 2012.

3.1.4. Arquitetura Alfrodul

O próximo trabalho analisado compreendendo o domínio de coleta de dados foi a publicação realizada por Sant'ana et al., (2014), esta publicação trata da elaboração de uma arquitetura de coleta, processamento e visualização de dados de clima espacial, de forma que tais dados possam ser analisados e previsões do clima espacial possam ser realizadas com base nestas análises. A publicação também demonstra a construção de um sistema baseado nesta arquitetura.

A solução proposta chama-se Alfrodul e tem o objetivo de atender uma série de requisitos impostos pelo processo de coleta de dados geograficamente distribuídas, através da utilização de sensores.

Tabela 10 - Requisitos arquiteturais do trabalho de Sant'anna et al., (2014)

(Continua)

Requisito Arquitetural	Descrição
RA - 01	Permitir a inspeção do diretório previamente definido para identificar a criação de novos arquivos.
RA - 02	Permitir abertura e pré-processamento de um arquivo, transformando-o em um ou mais arquivos aptos a serem enviados. Adicionalmente deve ser possível compactar os arquivos para otimizar o tempo de envio.
RA - 03	Permitir o envio de informações do sensor gerador de um arquivo específico para um local centralizado para permitir o processamento dos seus dados. No caso da existência de diversos arquivos, deve ser feito um controle de quais já foram enviados com sucesso, com reenvio em caso de falha.
RA - 04	Permitir o que o sistema receba os dados enviados de cada sensor, sabendo organizar e identificar sua origem. No caso dos arquivos de dados que foram quebrados em diversas partes, deve ser possível receber os mesmos em qualquer ordem e realizar processamento logo após sua chegada.
RA - 05	Possuir capacidade de lidar com uma grande demanda de processamento quando diversos dados forem recebidos de forma simultânea e cadenciar o processamento dos mesmos para não prejudicar o seu desempenho.
RA - 06	Persistir em uma base de dados os arquivos recebidos.

Tabela 10 - Requisitos arquiteturais do trabalho de Sant'anna et al., (2014)

(Continua)

Requisito Arquitetural	Descrição
RA - 01	Permitir a inspeção do diretório previamente definido para identificar a criação de novos arquivos.
RA - 02	Permitir a abertura e pré-processamento de um arquivo, transformando-o em um ou mais arquivos aptos a serem enviados. Adicionalmente deve ser possível compactar os arquivos para otimizar o tempo de envio.
RA - 03	Permitir o envio de informações do sensor gerador de um arquivo específico para um local centralizado para permitir o processamento dos seus dados. No caso da existência de diversos arquivos, deve ser feito um controle de quais já foram enviados com sucesso, com reenvio em caso de falha.
RA - 04	Permitir o que o sistema receba os dados enviados de cada sensor, sabendo organizar e identificar sua origem. No caso dos arquivos de dados que foram quebrados em diversas partes, deve ser possível receber os mesmo em qualquer ordem e realizar processamento logo após sua chegada.
RA - 05	Possuir capacidade de lidar com uma grande demanda de processamento quando diversos dados forem recebidos de forma simultânea e cadenciar o processamento dos mesmos para não prejudicar o seu desempenho.
RA - 06	Persistir em uma base de dados os arquivos recebidos.
RA - 07	Permitir exibição dos dados armazenados por meio de gráficos, imagens, vídeos, índices, tabelas e mapas de visualização geográfica. Uma vez que a geração desses formatos de saída demanda o acesso a uma grande quantidade de dados e são resultado de um processamento intenso, eles devem ser gerados previamente a requisição do usuário. Isso se agrava com a necessidade de atender múltiplos acessos simultâneos.
RA - 08	Devido ao recebimento contínuo de dados de instrumentos do programa de Clima Espacial e a importância da disponibilização dessas informações para a comunidade científica em tempo hábil para viabilizar análises e previsões, é necessário que os sistemas de recebimento, processamento e visualização estejam sempre disponíveis. A falha na disponibilidade pode ferir a credibilidade do sistema e causar uma impossibilidade de recuperar as informações no momento em que são necessárias.

Tabela 10 - Requisitos arquiteturais do trabalho de Sant'anna et al., (2014)

(Conclusão)

Requisito Arquitetural	Descrição
RA - 09	Capacidade de continuar disponível mesmo em caso de falha de software ou hardware na infra-estrutura utilizada. Adicionalmente, não pode haver perdas de dados no caso de falhas na comunicação, devendo o sistema retomar o envio de informações assim que as comunicações forem retomadas.
RA - 10	Capacidade de fornecer aos interessados as visualizações e análises fornecidas em tempo hábil. Para isso deve receber, processar, e disponibilizar esses dados na mesma cadência em que os dados são recebidos.
RA - 11	Capacidade de armazenar uma grande quantidade de dados sem comprometer as aplicações de visualização e os algoritmos de processamento.
RA - 12	Os módulos da arquitetura devem ser totalmente separados e isolados para o funcionamento de forma desacoplada, evitando a forte dependência entre as mesmas.
RA - 13	Capacidade de comunicação com outros sistemas através de novas tecnologias de interoperabilidade como exemplo, o uso de <i>web services</i> .

Fonte: Adaptado de Sant'anna et al., 2014.

Baseado nos requisitos arquiteturais vistos acima, foi analisada e sintetizada a arquitetura da solução Alfredul. A solução final foi proposta com a intenção de ser modular, abaixo seguem as especificações de cada módulo.

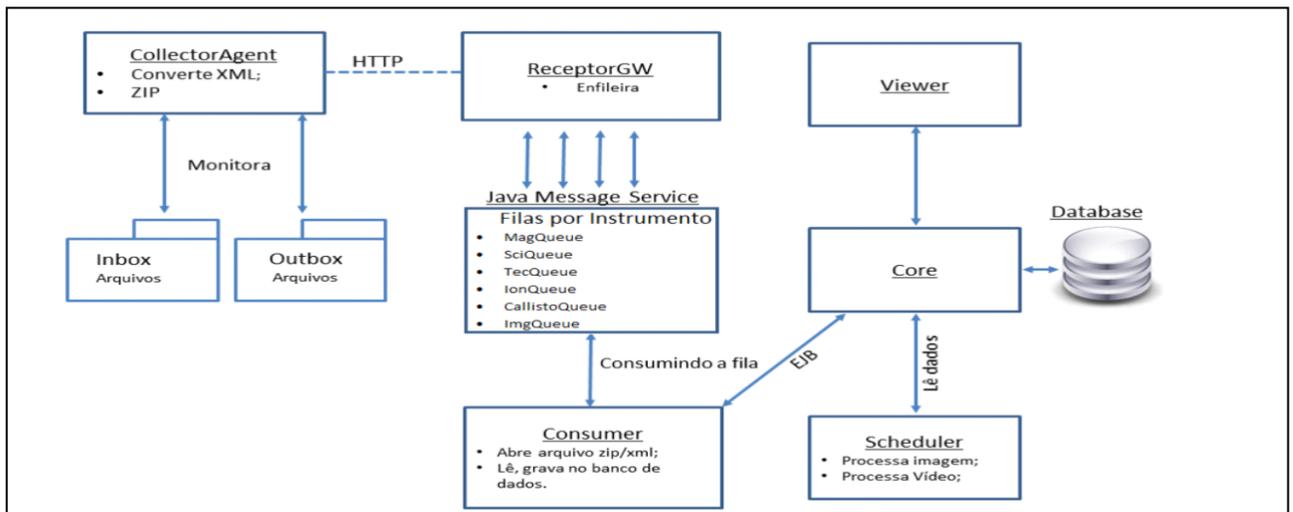


Figura 11 - Arquitetura Alfredul. Fonte: Sant'ana et al., 2014.

Santánna et al. (2014) compõem a arquitetura em oito componentes, os detalhes dos componentes da arquitetura são listados abaixo:

- **CollectorAgent:** Módulo que realiza o pré-processamento dos arquivos antes do envio para o servidor. O pré-processamento é constituído de leitura, reconhecimento dos campos, interpretação dos registros e a transformação para um arquivo XML adequado. Este módulo está localizado nas máquinas que contém o sistema de armazenamento dos arquivos gerados pelo sistema.
- **ReceptorGW:** Módulo que recebe e enfileira os elementos pré-processados pelo módulo *CollectorAgent*.
- **Consumer:** Módulo responsável pela captura e persistência dos arquivos enfileirados.
- **Processing Queue:** Módulo que proporciona a comunicação entre os diferentes módulos integrantes da arquitetura, este módulo padroniza a criação, a leitura, o recebimento e o envio das diferentes mensagens utilizadas pela aplicação.
- **Core:** Módulo que gerencia a comunicação e disponibiliza a base de dados.
- **Scheduler:** Módulo responsável pelo monitoramento regular da base de dados, verificando novos dados para o processamento e repasse para a interface de visualização.
- **Viewer:** Módulo que cria as representações visuais dos dados coletados, disponibilizando-as para as diferentes aplicações que necessitam exibir índices, mapas tabelas e gráficos.
- **Database:** Módulo responsável pelo armazenamento dos dados, abrangendo aquisições, dados processados e metadados.

3.2. SISTEMAS DE COLETA DE DADOS MÓVEIS

Como abordado anteriormente este trabalho define como escopo de sistemas de coleta de dados aplicações que não utilizem de recursos de computação móvel. Em contrapartida, este trabalho define que sistemas que façam uso de recursos de computação móvel compõem o subdomínio de coleta de dados móveis. Segue-se a análise destes trabalhos.

3.2.1. Arquitetura genérica para coleta de dados geriátricos

O primeiro trabalho analisado referente ao subdomínio de coleta de dados móveis foi o trabalho realizado por Boni *et al.*, (2006), este trabalho tem como meta o desenvolvimento de uma plataforma computacional genérica, que auxilie em pesquisas de coleta de dados em campo. Para que a meta do trabalho fosse alcançada, tornando a solução proposta viável e aplicável a diversos contextos de pesquisa, fez-se o uso de tecnologias de computação móvel.

O modelo de domínio do trabalho realizado por Boni *et al.*, (2006) foi voltado a coleta de dados móveis, porém sua aplicação foi no contexto de saúde pública, sendo focado na coleta de informações de pesquisas de avaliação geriátrica.

Tabela 11 - Requisitos arquiteturais do trabalho de Boni *et al.*, (2006)

Requisito Arquitetural	Descrição
RA - 01	Utilizar dispositivos móveis com boa capacidade de armazenamento.
RA - 02	Utilizar dispositivos móveis com bons recursos de entrada de dados
RA - 03	Utilizar dispositivos móveis com bom tempode duração da bateria.
RA - 04	Controle dos recursos do dispositivo móvel.
RA - 05	Permitir gerenciamento do nível de bateria.
RA - 06	Permitir gerenciamento do nível de memória.
RA - 07	Permitir coleta de dados de texto.
RA - 08	Permitir a coleta de dados de áudio.
RA - 09	Permitir validação dos dados coletados.
RA - 10	Permitir manutenção dos dados coletados.
RA - 11	Permitir o armazenamento de dados estruturados.
RA - 12	Permitir o armazenamento de dados não estruturados.
RA - 13	Permitir a sincronização dos dados coletados com o servidor principal, através de <i>Webservices</i> .
RA - 14	Garantir a correta replicação dos dados dos dispositivos mobile para o servidor.
RA - 15	Garantir a consistência das bases de dados móveis com o servidor através de políticas de transação.
RA - 16	Disponibilizar em uma interface web o conteúdo coletado para os usuários que necessitam consultá-lo.

Fonte: Adaptado de Boni *et al.*, 2006.

A aplicação desenvolvida no trabalho foi decomposta em camadas distintas, totalizando três camadas a imagem e a contextualização de cada uma das camadas que compõem a arquitetura seguem abaixo.

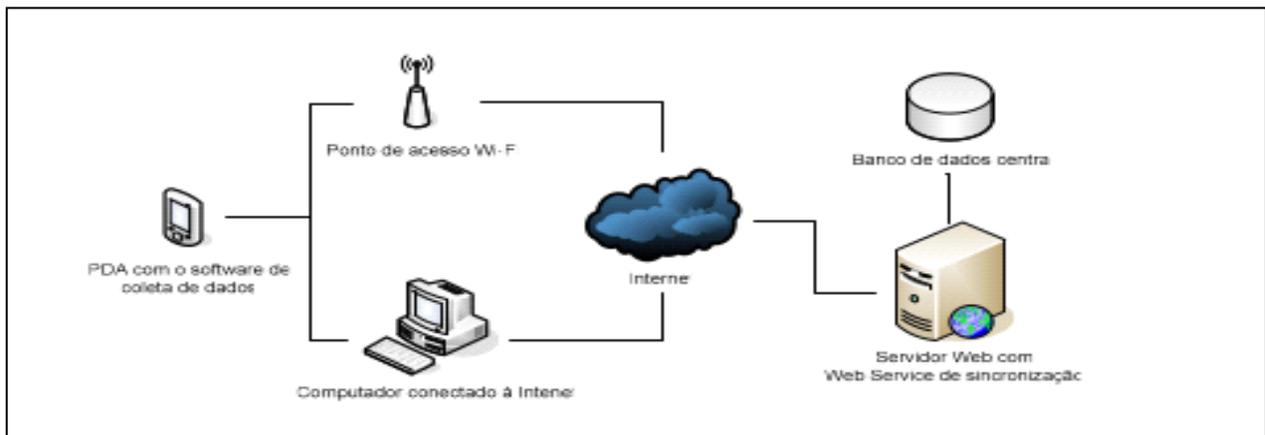


Figura 12 - Arquitetura genérica para coleta de dados geriátricos. Fonte: Boni *et al.*, 2006.

Boni *et al.*, (2006) compõem a arquitetura em três componentes básico, sendo:

- **Camada de acesso de dados:** Conjunto de classes que compõem a lógica utilizada no negócio, juntamente com as estruturas SQL (*Structured Query Language*) utilizadas para realizar consultas na base de dados.
- **Camada de interface:** Esta porção da arquitetura provê o meio de comunicação entre o usuário e o sistema, sendo composta por uma série de formulários dinâmicos e controle do usuário.
- **Camada de gerenciamento de recursos do dispositivo móvel:** Conjunto de abstrações que garantem um maior controle do status e do funcionamento de recursos presentes no dispositivo utilizado para a coleta móvel, como por exemplo nível de bateria, gerenciamento de memória e controle do microfone.

Devido a necessidade da recuperação e exportação de dados para o uso efetivo dos PDA's utilizados pela aplicação desenvolvida no trabalho, surge a necessidade da implementação da porção de gerência de sincronização de dados. Comumente esta sincronização é realizada através de um meio físico (cabo), sendo apoiada por softwares que podem ser proprietários ou não. Por motivos de flexibilidade e agilidade na utilização

dos dispositivos móveis, optou-se pela utilização de tecnologias de comunicação *wireless* baseada em *Web Services* para a realização do processo de sincronização. A escolha da utilização de *Web Services* para a realização do processo de sincronização também deu-se devido a capacidade de processamento de transações e a serialização dos dados. A abstração baseada em *Web Service* é uma aplicação hospedada em um servidor, com o objetivo de proporcionar uma camada intermediária entre os dispositivos móveis e o banco de dados.

3.2.2. Ferramenta móvel para levantamento epidemiológico em saúde bucal

O próximo trabalho que passou pelo processo de análise crítica observando suas características e requisitos arquiteturais foi o trabalho de Brega *et al.* (2008). Esta publicação tem como objetivo o desenvolvimento de um sistema computacional para sistemas embarcados que suporte, agilize e aprimore a confiabilidade dos dados do processo de coleta de dados em levantamentos epidemiológicos em saúde bucal. Tais processos de coleta são regimentados pelo padrão estipulado pela Organização Mundial da Saúde (OMS), sendo assim a solução proposta teve de ser modelada com base nos requerimentos deste padrão.

Tabela 12 - Requisitos arquiteturais do trabalho de Brega *et al.*, (2008)

Requisito Arquitetural	Descrição
RA - 01	Permitir que a coleta de dados possa ser realizada remotamente.
RA - 02	Garantir que o padrão de coleta siga o regimento da Organização Mundial da Saúde.
RA - 03	Garantir que a interface de coleta de dados seja simples e intuitiva para os seus realizadores.
RA - 04	Permitir que os dados possam ser guardados no dispositivo móvel.
RA - 05	Permitir que os dados possam ser verificados no dispositivo móvel.
RA - 06	Garantir a comunicação dos dispositivos móveis com o o servidor de aplicação de forma <i>wireless</i> .
RA - 07	Garantir a correta transmissão dos dados para o servidor de aplicação de forma <i>wireless</i> .
RA - 08	Garantir o correto processamento das requisições realizadas pelos dispositivos móveis, permitindo resultado adequado dos softwares estatísticos utilizados.
RA - 09	Permitir a análise dos dados estatísticos e produção de relatórios.

Fonte: Adaptado de Brega *et al.*, 2008.



Figura 13 - Representação do processo de coleta de dados sem a solução proposta.

Fonte: Brega *et al.*, 2008.

É evidenciada no trabalho a desaceleração do processo de coleta de dados devido a dois motivos, o primeiro motivo é a necessidade da digitalização dos dados para que o restante do processamento ocorra, o segundo motivo é a análise dos dados, esta é realizada com o auxílio de softwares de análise estatística que necessitam de conhecimentos específicos para serem utilizados.

A solução proposta para a resolução do problema apresentado na publicação foi uma aplicação para dispositivos móveis, especificamente PDAs, que funciona em conjunto com softwares estatísticos *opensource*. Esta abordagem permite que os processos que necessitam de mais tempo e interação humana sejam otimizados e possam prover os requisitos almejados pelo trabalho.

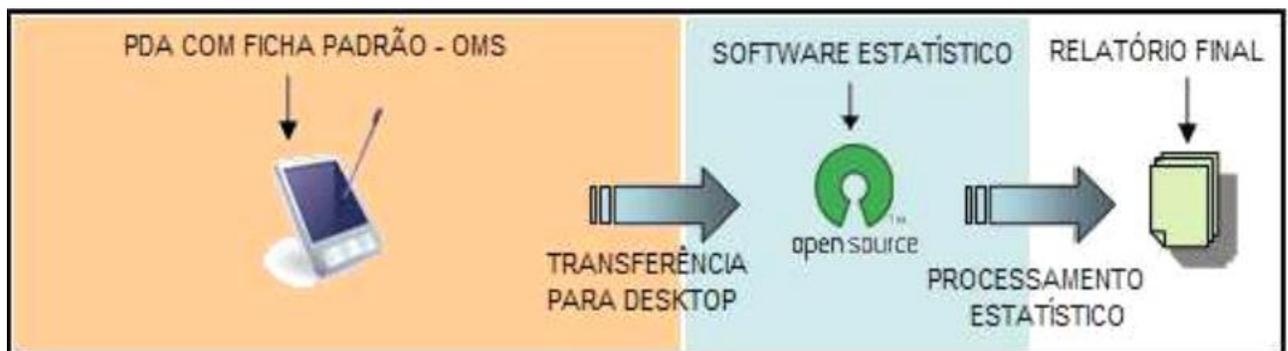


Figura 14 - Representação do processo de coleta de dados com a solução proposta.

Fonte: Brega *et al.*, 2008.

As estruturas de dados para envio e recebimento de dados foram padronizadas com técnicas de modelagem de objetos de negócio na linguagem XML (Extensible

Markup Language). A utilização de XML garante uma maior interoperabilidade para o sistema, podendo migrar facilmente os registros para qualquer base de dados, apenas seguindo seus XML Schemas propostos.

Brega *et al.*, (2008) descreve brevemente o processo de construção do sistema proposto, sintetizando as etapas em (i) análise das planilhas OMS utilizadas na coleta de dados e as possíveis formas de implementação, (ii) modelagem de dados obtidos e desenvolvimento dos *Schemas* XML para armazenamento apropriado, (iii) análise e definição de requisitos, (iv) desenvolvimento da aplicação de coleta de dados, (v) desenvolvimento dos métodos de transferência, (vi) desenvolvimento de *middleware* para transformação dos dados iniciais em dados estatísticos.

3.2.3. *Field Data Collecting Synchronously System*

A publicação realizada por Ji *et al.*, (2010) foi outro trabalho examinado no que tange a requisitos e estrutura arquitetural. Neste trabalho é definida uma solução computacional que busca prover assistência no processo de compreensão de modificação, colapso e afundamento do solo de áreas de mineração, oriundos da própria atividade de exploração de minérios subterrâneos. Além disto o sistema proposto busca melhorar a eficiência da coleta de dados de campo. A solução foi batizada de *Field Data Collecting Synchronously System* (FDCSS).

Tabela 13 - Requisitos arquiteturais do trabalho de Ji *et al.*, (2010)

(Continua)

Requisito Arquitetural	Descrição
RA - 01	Garantir a disponibilidade do sistema.
RA - 02	Garantir a confiabilidade dos dados do sistema.
RA - 03	Permitir a realização da coleta de dados (dados espaciais e propriedades) em campo.
RA - 04	Permitir o armazenamento dos dados coletados no banco de dados dos dispositivos móveis.
RA - 05	Permitir o armazenamento dos dados coletados no banco de principal.
RA - 06	Permitir a manutenção dos dados coletados remotamente no banco de dados remoto.
RA - 07	Permitir a manutenção dos dados coletados remotamente no banco de dados principal.

Tabela 13 - Requisitos arquiteturais do trabalho de Ji *et al.*, (2010)

(Conclusão)

Requisito Arquitetural	Descrição
RA - 08	Garantir a correta representação dos dados de mapeamento de áreas tridimensionais nos dispositivos móveis.
RA - 09	Garantir a correta transformação dos dados de mapeamento de áreas tridimensionais nos dispositivos móveis.
RA - 10	Garantir a correta representação dos dados de mapeamento de áreas tridimensionais no banco de dados principal.
RA - 11	Garantir a correta transformação dos dados de mapeamento de áreas tridimensionais no banco de dados principal.
RA - 12	Permitir a utilização do recurso de GPS nativo do dispositivo móvel.
RA - 13	Permitir a comunicação entre os dispositivos móveis e o servidor de aplicações.
RA - 14	Permitir a sincronização das bases de dados móveis com a base de dados principal.
RA - 15	Permitir a exportação dos dados contidos na base de dados remota através de conexões cabeadas, quando o dispositivo móvel estiver conectado com um computador.
RA - 16	Permitir a exportação dos dados contidos na base de dados remota através de conexões <i>wireless</i> , com a utilização de <i>Web Services</i> .
RA - 17	Permitir a visualização dos mapas no dispositivo móvel.

Fonte: Ji *et al.*, (2010).

O sistema proposto busca coletar dados referentes a deformação tridimensional causada nas áreas estudadas. A estrutura arquitetural do sistema como um todo foi organizada em três subsistemas, estes são listados abaixo:

- **Server side data management:** Subsistema que proporciona o armazenamento e gerência dos dados espaciais coletados utilizando respectivamente uma base de dados *Oracle* e o subsistema *ArcSDE*. Este subsistema também pode fornecer acesso para listagem ou edição de dados através de *Web Map Services*.

De acordo com Ji *et al.* (2010) *Web Map Service* é um protocolo que busca fornecer imagens de mapas georreferenciados. Este recurso funciona como um protocolo padrão disponibilizando estes tipos de dados na internet.

- **Client side data collection:** Subsistema remoto que coleta informações em campo, armazenando-as localmente e sincronizando-as de forma *wireless* através de *Web Services* ou através de conexão física com uma máquina.
- **Intermediate communication links: Interconexões** estabelecidas de forma sem fio e cabeada, necessária para a comunicação dos diferentes subsistemas que compõem o *Field Data Collecting Synchronously System (FDCSS)*.

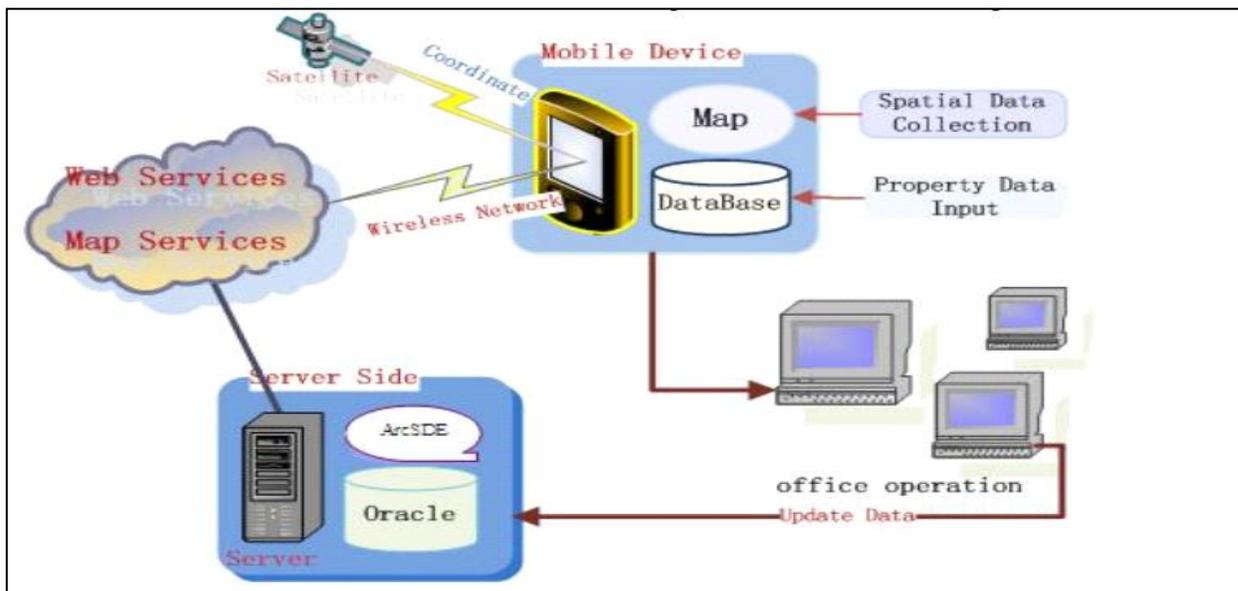


Figura 15 - Arquitetura do *Field Data Collecting Synchronously System (FDCSS)*. Fonte: JI *et al.*, (2010).

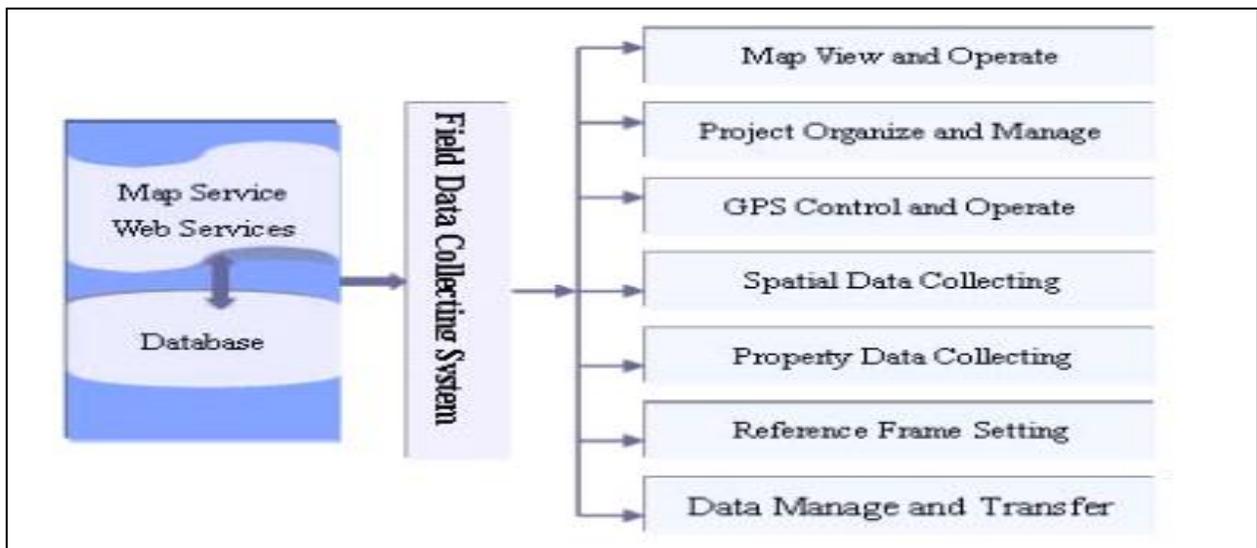


Figura 16 - Funções do subsistema móvel. Fonte: JI *et al.*, (2010).

3.2.4. Projeto maritaca

Outro projeto analisado foi o projeto Maritaca, que permite a construção de aplicações de coletas de dados. Objetivamente o projeto permite que o usuário final elabore formulários de pesquisa utilizando uma interface Web, definindo papéis e seus níveis de permissão em relação aos formulários criados. Posteriormente, estes formulários podem ser importados para os dispositivos móveis de coleta de dados para que a coleta seja realizada e futuramente sincronizada com o servidor. É importante ressaltar que não é obrigatória a conexão com a internet para que a coleta seja realizada.

Devido a possibilidade de criação de relatórios específicos para cada coleta, juntamente com a utilização de uma aplicação *mobile* que possibilita a visualização de tais formulários e a posterior exportação dos dados coletados para um servidor em nuvem, a aplicação possibilita a utilização de suas funcionalidades como serviço, tendo caráter genérico.

Para proporcionar um sistema mais manutenível e organizado, foi adotada a estratégia de modularização do sistema, dividindo as responsabilidades por entre os elementos integrantes do sistema. A arquitetura é composta por um servidor de aplicação, um editor de formulários, a aplicação móvel, um servidor de dados, e um sistema de arquivos hadoop. Abaixo segue a imagem ilustrativa da arquitetura do sistema Maritaca e o detalhamento de cada um dos seus módulos.

Tabela 14 - Requisitos arquiteturais do trabalho Santos *et al.*, 2013

(Continua)

Requisito Arquitetural	Descrição
RA - 01	Garantir disponibilidade do sistema.
RA - 02	Permitir que o usuário possa autenticar-se através do navegador.
RA - 03	Permitir que o usuário possa autenticar-se através do dispositivo móvel.
RA - 04	Garantir que o usuário se autentique para utilizar o sistema.
RA - 05	Garantir que sistema suporte diversas permissões para cada questionário.
RA - 06	Garantir que o sistema aceite diversos tipos de perguntas.
RA - 07	Garantir que o sistema aceite dados estruturados.
RA - 08	Garantir que o sistema aceite dados não estruturados.
RA - 09	Permitir que o usuário possa manter questionários customizados.
RA - 10	Garantir que o sistema aceite diversos tipos de perguntas.

Tabela 14 - Requisitos arquiteturais do trabalho Santos *et al.*, 2013

(Conclusão)

Requisito Arquitetural	Descrição
RA - 11	Garantir que a transformação do questionário em um arquivo APK seja uma tarefa que rode em <i>background</i> , maximizando a disponibilidade.
RA - 12	Garantir que o armazenamento do APK questionário no <i>clusterhadoop</i> seja uma tarefa que rode em <i>background</i> , maximizando a disponibilidade.
RA - 13	Permitir a replicação do formulário para um dispositivo móvel <i>android</i> .
RA - 14	Permitir que o dispositivo móvel possa realizar coletas de dados sem necessitar de acesso a internet.
RA - 15	Permitir que o dispositivo móvel possa realizar o armazenamento local dos questionários respondidos.
RA - 16	Permitir que o formulário possa validar as informações antes de salvá-las localmente.
RA - 17	Permitir que o dispositivo móvel possa enviar todos os dados coletados para o servidor de aplicação.
RA - 18	Permitir que o dispositivo móvel possa sincronizar os dados mais atuais com o servidor.

Fonte: Adaptado de Santos *et al.*, 2013.

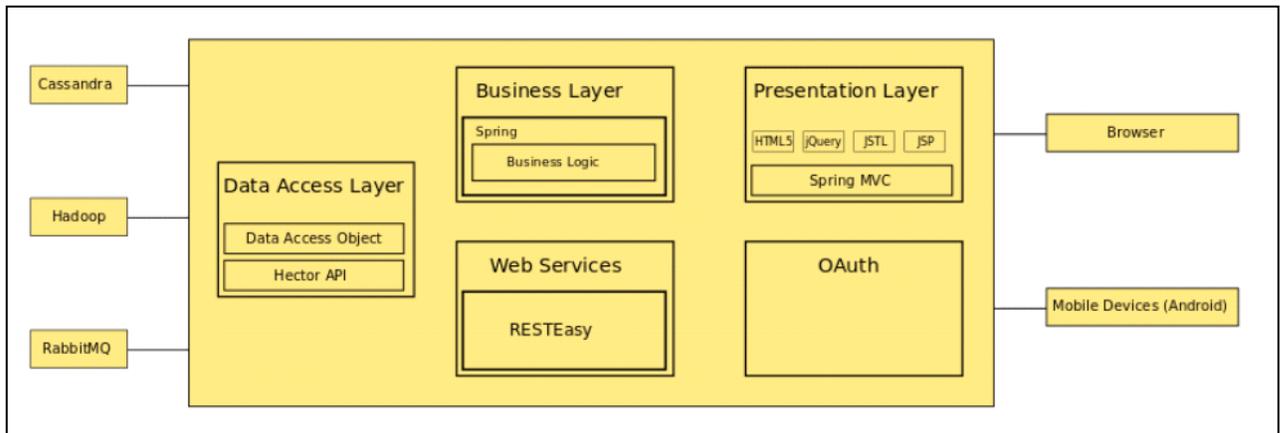


Figura 17 - Componentes da plataforma Maritaca. Fonte: Santos *et al.*, 2013.

- **Módulo servidor:** Local onde residem os serviços Web da aplicação como a camada de manutenção dos dados como o armazenamento de dados multimídia e APPs, a camada de lógica de negócio, os serviços de comunicação REST, a camada de apresentação e a camada de segurança.

A comunicação entre a plataforma e os usuários que não estão utilizando o módulo móvel é feita através da camada de apresentação composta pela pilha do *framework Spring*. O ambiente web é o local onde são desenvolvidos os questionários de coleta de dados através de um navegador Web, após o processo de criação os questionários podem ser utilizados nos dispositivos móveis e posteriormente serem sincronizados com o servidor.

- **Módulo móvel:** Subsistema móvel que possui um motor de tradução, proporcionando a correta representação do questionário criado no ambiente web para o dispositivo móvel. A interação entre este módulo e o módulo servidor é realizada através dos serviços REST expostos pelo módulo servidor.

Antes que qualquer operação de sincronização ou de coleta de dados com questionários possa ser realizada na aplicação é necessária a autenticação do usuário com o módulo servidor, o que verifica se o usuário pode acessar o sistema. O processo de autenticação utiliza a técnica chamada OAuth.

O processo de autenticação implementado espera o seguinte conjunto de passos entra a unidade móvel e o servidor:

- Solicitação de autorização e redirecionamento para página de autenticação;

- Ao se autenticar com sucesso é gerado um código de autenticação;
- A aplicação gera um *access token* com o código de autenticação;
- Ao obter o token o usuário pode realizar o processo de manutenção e sincronização dos questionários.



Figura 18 - Processo de autenticação OAuth do projeto Maritaca. Fonte: Santos *et al.*, 2013.

Para que o questionário seja disponibilizado no dispositivo móvel primeiro é realizada uma consulta ao módulo servidor, esta consulta retorna arquivos XML que sofrem um processo de tradução inspirada no padrão de projeto *Interpreter*. Este processo ocorre através da tradução de uma parte do arquivo do questionário em XML chamada descritor, este descritor traduz uma hierarquia de objetos necessária para representar o questionário e validar seu preenchimento na aplicação móvel.

Algumas observações adicionais devem ser feitas em relação a arquitetura proposta pelo trabalho de Santos *et al.*, (2013), primeiramente a integração entre os componentes é mantida através do arquivo XML do questionário, este é gerado pelo editor de formulários com *tags* especiais que proporcionam estruturas que auxiliam a traduzir a representação visual e comportamental do formulário, esta solução também suporta a coleta de dados não estruturados como áudio, fotos e vídeo.

Por fim é importante explicar como o projeto Maritaca busca resolver a questão de software como serviço, uma vez que se pode criar vários questionários independentemente do escopo da pesquisa. Isto é resolvido com a criação de uma

aplicação individual em formato APK para cada questionário ou grupo de questionários, estes arquivos APK são salvos no sistema de arquivos Hadoop. Esta estratégia foi utilizada devido a problemas que surgiam em relação ao processo de divulgação e manutenção do versionamento.

3.2.5. Arquitetura de coleta de dados para ambientes heterogêneos

Outro trabalho pesquisado foi o trabalho de Freitas Filho (2014) que tem como meta a especificação e implementação de uma arquitetura de coleta de dados para pesquisas de campo em ambientes computacionais heterogêneos.

Tabela 15 - Requisitos arquiteturais do trabalho de Freitas Filho (2014)

(Continua)

Requisito Arquitetural	Descrição
RA- 01	A arquitetura deve ser interoperável, ou seja, não depender de tecnologia específica.

Tabela 15 - Requisitos arquiteturais do trabalho de Freitas Filho (2014)

(Conclusão)

Requisito Arquitetural	Descrição
RA- 02	A arquitetura deve ser flexível para atender as diversas dimensões que uma coleta de dados em campo pode ter, envolvendo uma ou mais instituições.
RA- 03	A arquitetura necessita ser flexível para atender as mudanças necessárias que a arquitetura precisa passar para se adequar aos contextos de coleta de dados diferentes.
RA- 04	Permitir que o usuário possa autenticar-se através do navegador.
RA- 05	Permitir que o usuário possa autenticar-se através do dispositivo móvel.
RA- 06	Garantir que o usuário se autentique para utilizar o sistema.
RA- 07	Garantir que o sistema aceite dados estruturados.
RA- 08	Garantir que o sistema aceite dados não estruturados.
RA- 09	Permitir que o dispositivo móvel possa realizar coletas de dados sem necessitar de acesso a internet.
RA- 10	Permitir que o dispositivo móvel possa realizar o armazenamento local dos questionários respondidos.

RA- 11	Permitir que o formulário possa validar as informações antes de salvá-las localmente.
RA- 12	Permitir que o dispositivo móvel possa enviar todos os dados coletados para o servidor de aplicação.
RA- 13	Permitir sincronização dos dados mais atuais com o servidor e o dispositivo móvel.
RA- 14	Garantir a integridade durante a sincronização dos dados coletados.
RA- 15	Garantir o reconhecimento de erros de requisição durante o processo de sincronização dos dados.

Fonte: Adaptado de Freitas Filho, 2014.

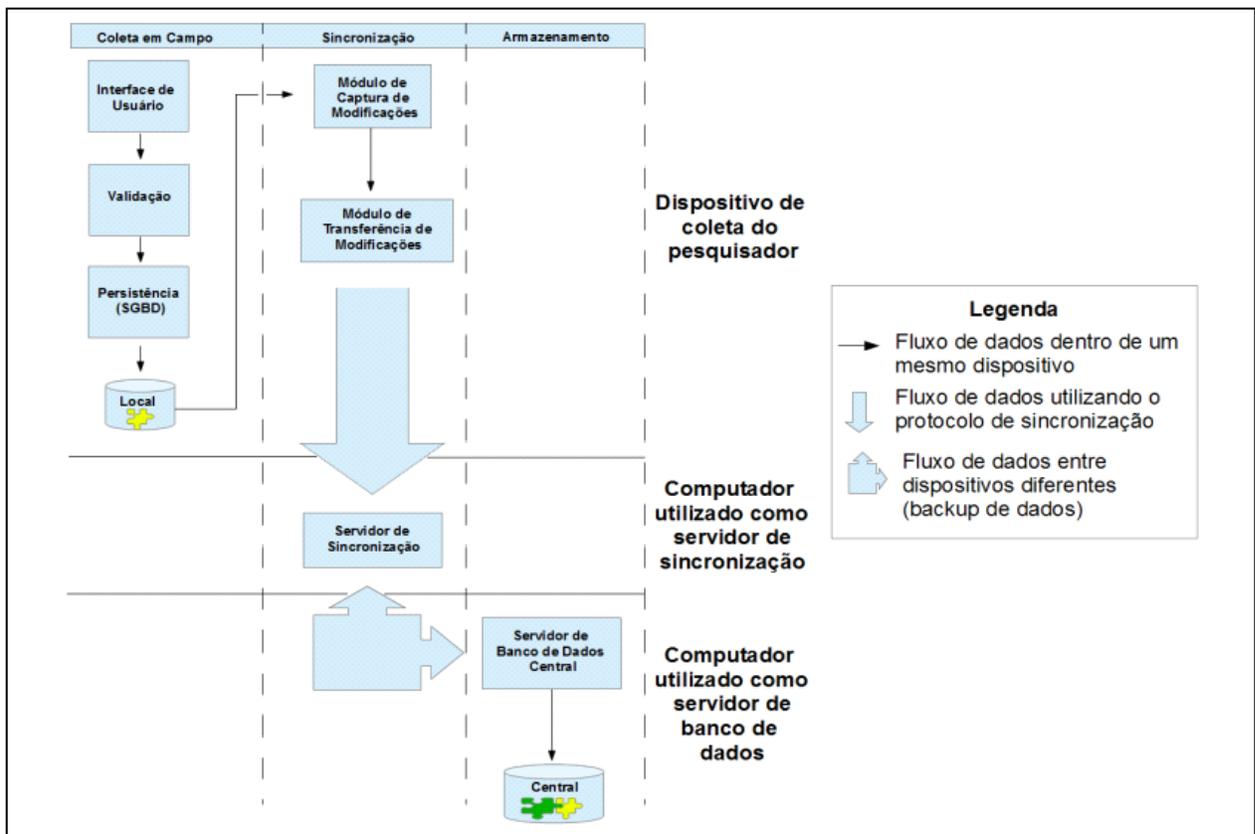


Figura 19 - Modelo de recursos parcial para a arquitetura. Fonte: Freitas Filho, 2014.

Freitas Filho (2014) define que a organização da arquitetura é estabelecida em três etapas que compreendem as funcionalidades chave da arquitetura, sendo estas a coleta de campo, a sincronização e o armazenamento. Para que os requisitos de interoperabilidade e flexibilidade sejam alcançados é definido a necessidade de escolher um protocolo de sincronização livre, uma vez que este estabelecerá tais aspectos arquiteturais.

- **Coleta de campo:** Processo suportado pelo sistema em dispositivo móvel. Tal sistema deve poder validar os dados e possuir uma forma de armazenamento.
- **Armazenamento:** Este processo pode ser realizado em um único servidor ou pode-se utilizar a estratégia distribuída.
- **Sincronização:** Processo de criação de replicas totais dos dados coletados para os servidores de aplicação, que são posteriormente repassam a informação para os servidores de banco de dados. O processo de sincronização explorado pela arquitetura envolve a utilização de dois módulos diferentes.

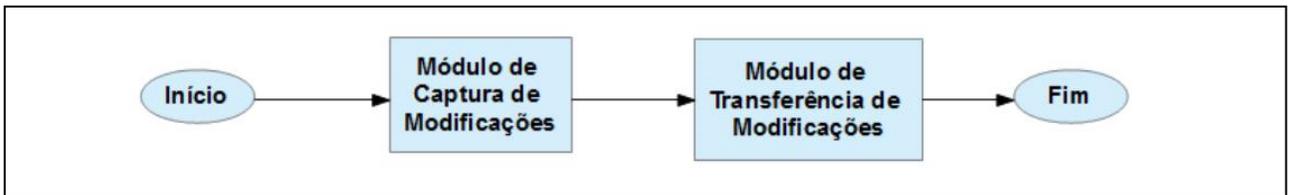


Figura 20 - Processo de sincronização proposto por Freitas Filho (2014). Fonte: Freitas Filho, 2014.

- **Módulo de Captura de Modificações:** Este módulo é ativado após o pesquisador escolher a opção de sincronização. Basicamente é realizada a captura dos dados que possuem uma *flag* de com o valor 1. Este mecanismo permite que o sistema possa demarcar o que já foi sincronizado. A captura dos dados pode ser efetuada em uma tabela que contenha as modificações do bando ou em todas as tabelas existentes na aplicação.
- **Módulo de Transferência de Modificações:** Processo realizado após a realização da captura de modificações. A transferência é realizada para o servidor de sincronização, este processo acontece de acordo com o protocolo de sincronização escolhido.

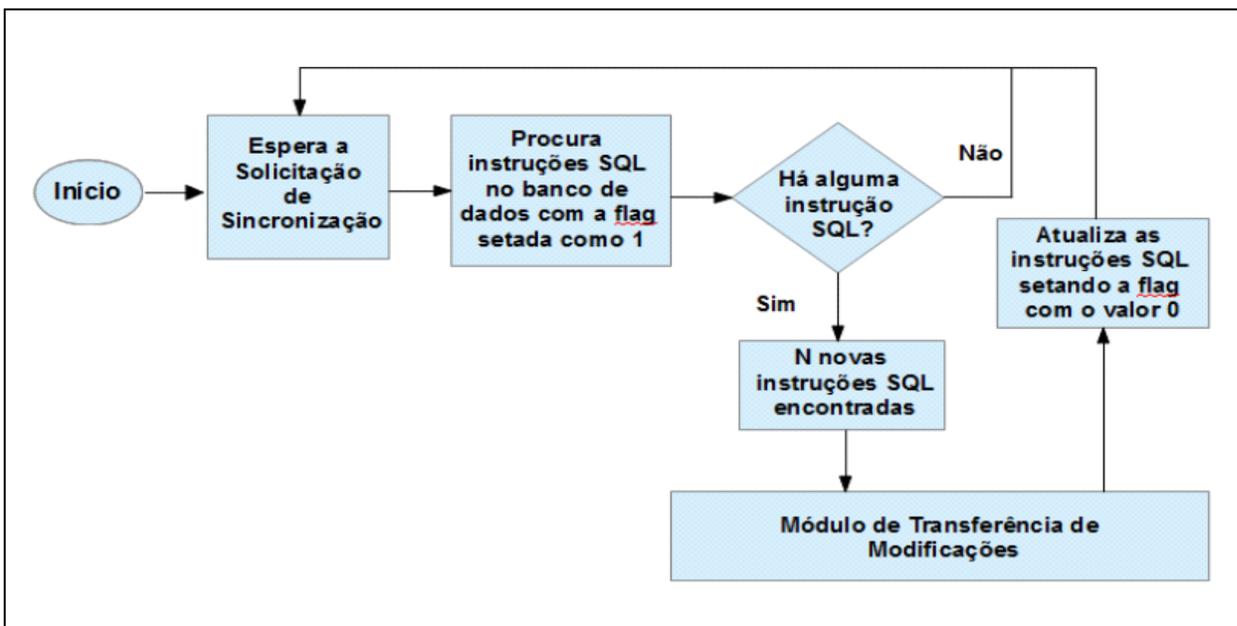


Figura 21 - Diagrama de fluxo do módulo de captura de instruções SQL. Fonte: Freitas Filho, 2014.

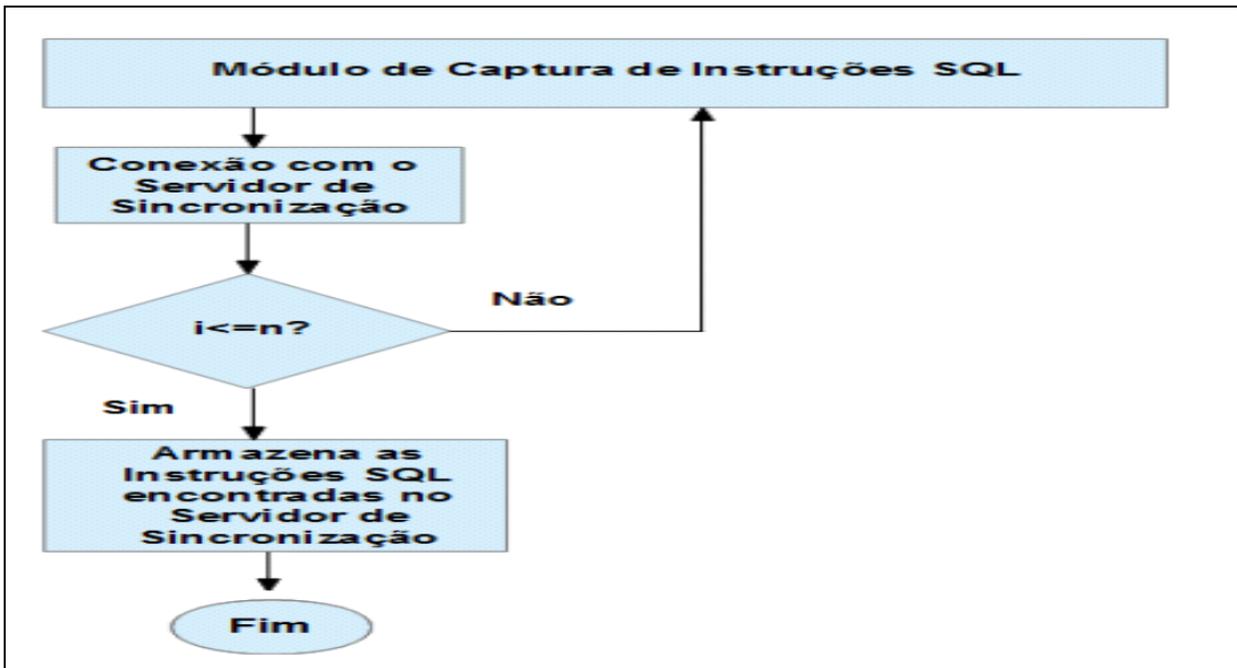


Figura 22 - Diagrama de fluxo do módulo de transferência de modificações. Fonte: Freitas Filho, 2014.

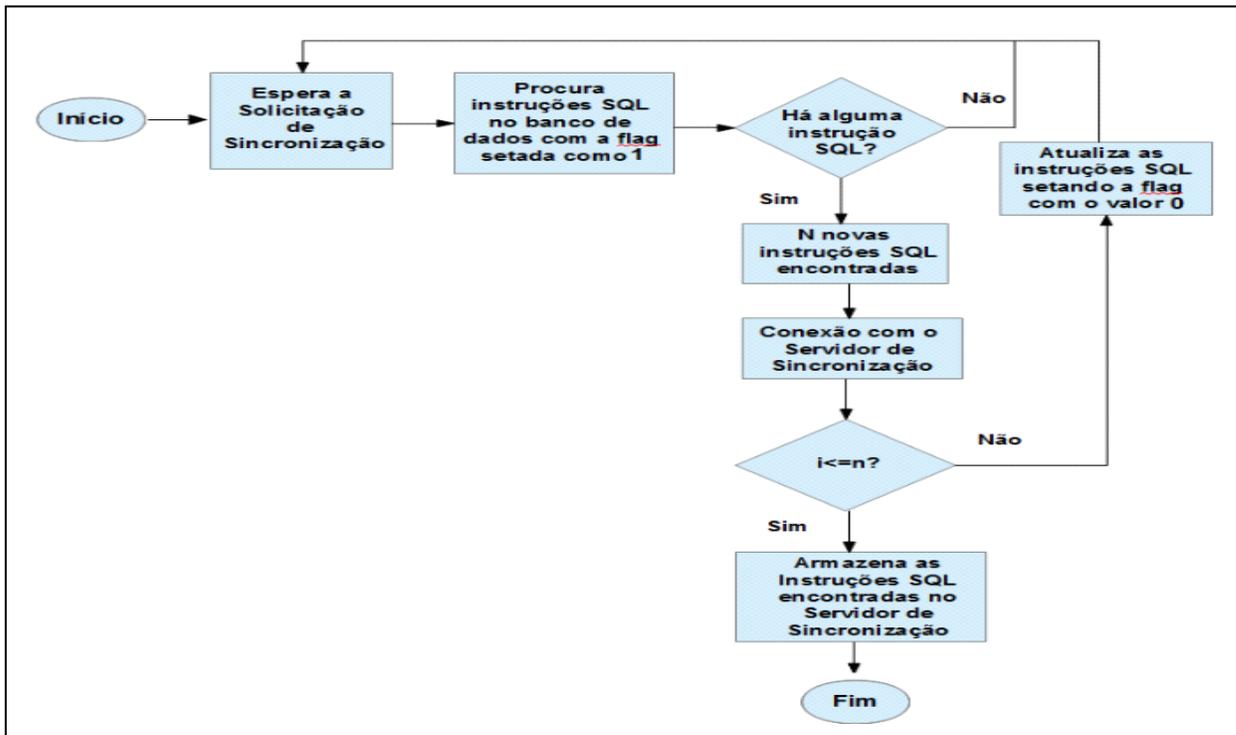


Figura 23 - Diagrama de fluxo dos módulos de captura e de transferência de modificações. Fonte: Freitas Filho, 2014.

Caso algum erro de conexão ocorra enquanto uma transferência de modificação esteja em andamento será atribuído a variável n o valor de todas as transferências que falharam, para que estas possam ser transmitidas corretamente quando a conexão for restabelecida.

3.2.6. Linha de produto de software para coleta de dados utilizando Android

O sexto trabalho analisado foi a publicação realizada por Waku *et al* (2015) que tem como objetivo a proposta de uma arquitetura de linha de produto de software (LPS) ou *Software Product Line* (SPL) que seja robusta e genérica chamada R-SPL-DC, voltada para o domínio de coleta de dados móveis. Como é explicado por Waku *et al* (2015) a características de robustez de um sistema é basicamente a capacidade de entrega de um serviço, dentro de um ambiente repleto de condições adversas.

A arquitetura proposta apresenta estratégias para prover requisitos como disponibilidade, confiabilidade, performance e integridade dos dados. Dentre os seus requisitos a R-SPL-DC propõe os seguintes:

Tabela 16 - Requisitos arquiteturais do trabalho de Waku *et al.*, (2015)

(Continua)

Requisito Arquitetural	Descrição
RA- 01	O uso de múltiplos dispositivos para lidar com coletas de dados de longa duração.
RA- 02	Sincronização de dados para aproveitar as informações entre dispositivos.
RA- 03	Redundância de armazenamento para evitar a perda de dados.
RA- 04	Diferentes tipos de formato de armazenamento para compartilhar com aplicativos de terceiros o mesmo banco de dados.
RA- 05	Monitoramento e bloqueio de operações básicas como importação e exportação de dados do servidor, utilizando técnicas de desenvolvimento de software orientadas a aspecto, para evita perda de dados.
RA- 06	O sistema deve armazenar dados coletados no campo.
RA- 07	A localização do GPS deve ser usada para mostrar o local onde ocorreu a coleta de dados.
RA- 08	O sistema deve identificar o usuário.
RA- 09	O sistema deve registrar a data e a hora da coleta de dados.
RA- 10	O sistema deve permitir a possibilidade de carregar informações sobre quais dados seriam coletados.
RA- 11	O sistema deve persistir a informação

Tabela 16 - Requisitos arquiteturais do trabalho de Waku *et al.*, (2015)

(Conclusão)

Requisito Arquitetural	Descrição
RA- 12	O sistema deve exportar dados para outros sistemas ou criar uma representação de banco de dados externa, como planilhas.
RA- 13	O sistema deve permitir a validação dos dados coletados.
RA- 14	O sistema deve permitir visualização do mapa atual da área de coleta de dados
RA- 15	O sistema precisa prevenir a perda de dados e falha do serviço de coleta de dados, porque se a base de dados da aplicação foi perdida ou corrompida, o principal objetivo do sistema será afetado, e o sistema não irá garantir a integridade e disponibilidade dos dados.
RA- 16	O sistema pode não ser afetado pelo baixo nível de bateria do dispositivo móvel, porque a coleta de dados pode levar várias horas, e enquanto o nível da bateria estiver baixo, o sistema não estará disponível.
RA- 17	O sistema deve garantir que falha nos serviços de coleta de dados não ocorreram porque se o sistema não puder armazenar e recuperar os dados corretamente, o sistema não será confiável e, se o serviço não estiver sendo executado, o sistema não estará disponível
RA- 18	O sistema deve garantir que operações básicas de coleta de dados tais como armazenar e recuperar informações são executadas em menos de um segundo, porque se isso não ocorrer, os usuários podem não coletar dados mais rapidamente e o desempenho geral seria comprometida.

Fonte: Waku *et al.*, 2015.

Utilizando o desenvolvimento orientado a aspecto o trabalho pretende dividir sua solução em duas partes um modelo de recursos juntamente com uma view de aspecto-recurso e a elaboração de uma arquitetura de linha de produto de software utilizando o método AO-FarM, abaixo segue a imagem do modelo de recursos e a descrição dos recursos.

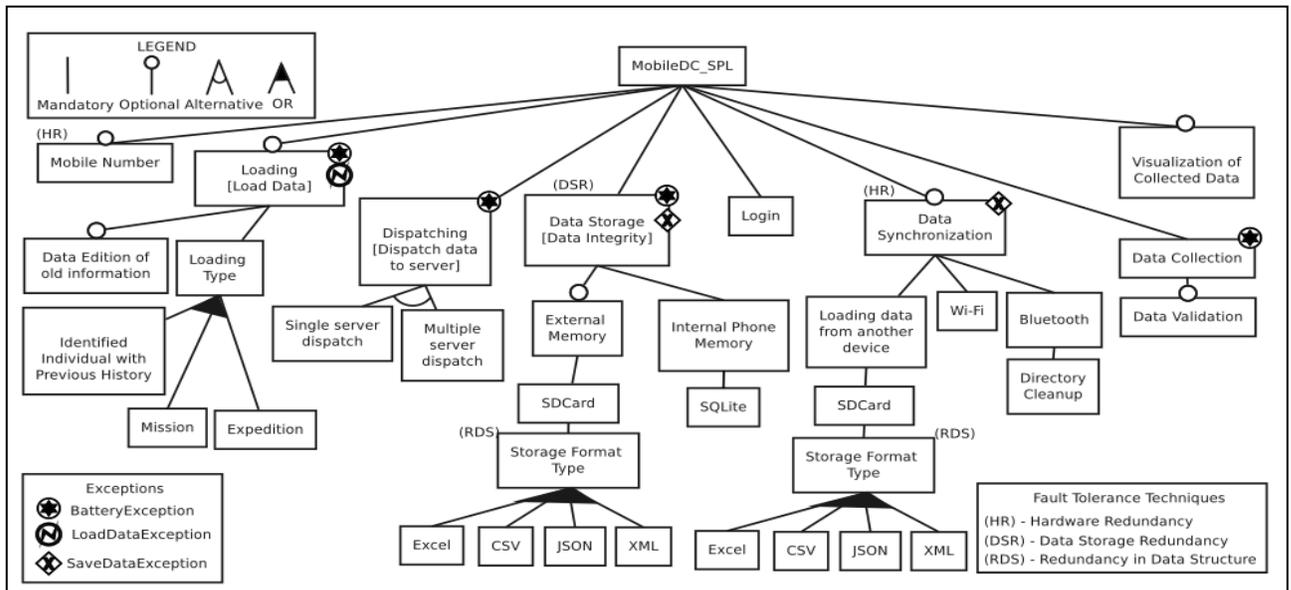


Figura 24 - Modelo de recursos parcial para a arquitetura. Fonte: Waku *et al.*, 2015.

- **Mobile number:** Quantidade de dispositivos necessários para a realização da coleta móveis levados para o campo. De acordo com o número de horas e a configuração do aparelho, pode haver a necessidade de utilização de mais de um dispositivo.
- **Loading:** Carregador de informações de coletas anteriores. Existem três tipos de carregamento de dados Indivíduo com História Prévia, Missão (uma coleção de indivíduos identificados) ou Expedição (coleção de missões).
- **Data edition:** Recurso opcional de edição dos dados.
- **Dispatching:** Recurso que cuida do despacho de dados para o servidor, podendo ser executado de duas formas, para múltiplos servidores ou um servidor único.
- **Data Storage:** Recurso que agrupa o conjunto de diretrizes para registro de dados. Pode gravar dados com SQLite na memória interna e com o recurso Storage Format Type na memória externa. A utilização da memória externa é opcional, gravando os dados em diversos formatos, por exemplo, Excel, CSV, JSON ou XML).
- **Login:** Recurso que autentica o usuário.
- **Data Synchronization:** Recurso que compreende a sincronização de dados realizada em campo entre os dispositivos móveis que estão em campo. O que possibilita que o usuário possa repassar os dados de um dispositivo para outro, caso haja algum

problema com o dispositivo móvel em uso, para que a coleta possa continuar. O processo pode ocorrer utilizando Wi-Fi, Bluetooth, ou SDCARD.

- **Data Collection:** Recurso que simboliza a aquisição de dados. Este recurso utiliza do recurso de Armazenamento de Dados *Data Storage*, podendo guardar nas memórias interna ou externa. Este recurso também pode usar o recurso de Validação, *validation feature*.
- **Data Validation:** Recurso que de acordo com as regras de negócio, verifica a consistência dos dados.
- **Visualization of Collected Data:** Recurso que implementa diferentes formas de cisualizar os dados. Este recurso é opcional.

Waku *et al.*, (2015) também discorre sobre o conceito de recursos transversais (*Crosscutting Features*), estes são características do sistema presentes e influentes em relação a outros recursos.

Os recursos transversais identificados dentro do contexto do domínio de coleta de dados foram os recursos de bloqueio de função (*Block Usage*) e monitor de bateria (*Battery Monitor*) que são responsáveis respectivamente por bloqueio de funções, como carregar, despachar e coletar dados para o servidor e o gerenciamento e estima da quantidade de energia disponível na bateria do dispositivo móvel, para que seja possível verificar se as atividades desempenhadas pela aplicação poderão ser completamente executadas sem erros oriundos do baixo nível de energia da bateria.

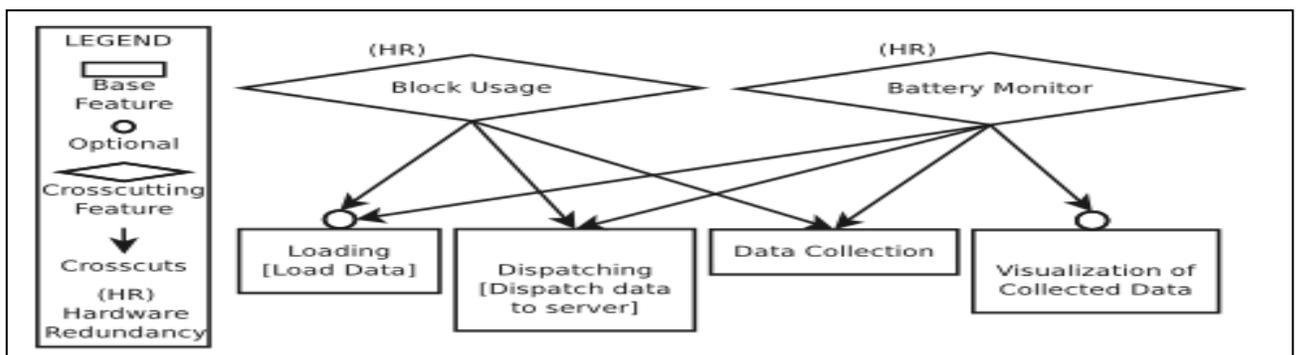


Figura 25 - View de aspecto-recurso exibindo os recursos transversais. Fonte: Waku *et al.*, 2015.

A figura abaixo demonstra a arquitetura proposta por Waku *et al.* (2015), sendo esta uma representação em componentes das *features* ou recursos modelados no

modelo de recursos. É importante ressaltar que apesar do diagrama apresentar uma forte relação entre seus componentes e os recursos (*features*) do modelo de recursos, alguns recursos não necessitam serem representados como componentes pois são representações abstratas de um componente da arquitetura que não é nem hardware e nem software, como por exemplo, a *feature Mobile Number* necessita ser explicitamente definida na arquitetura, porém representa o número de dispositivos móveis que serão utilizados no processo de coleta de dados móveis.

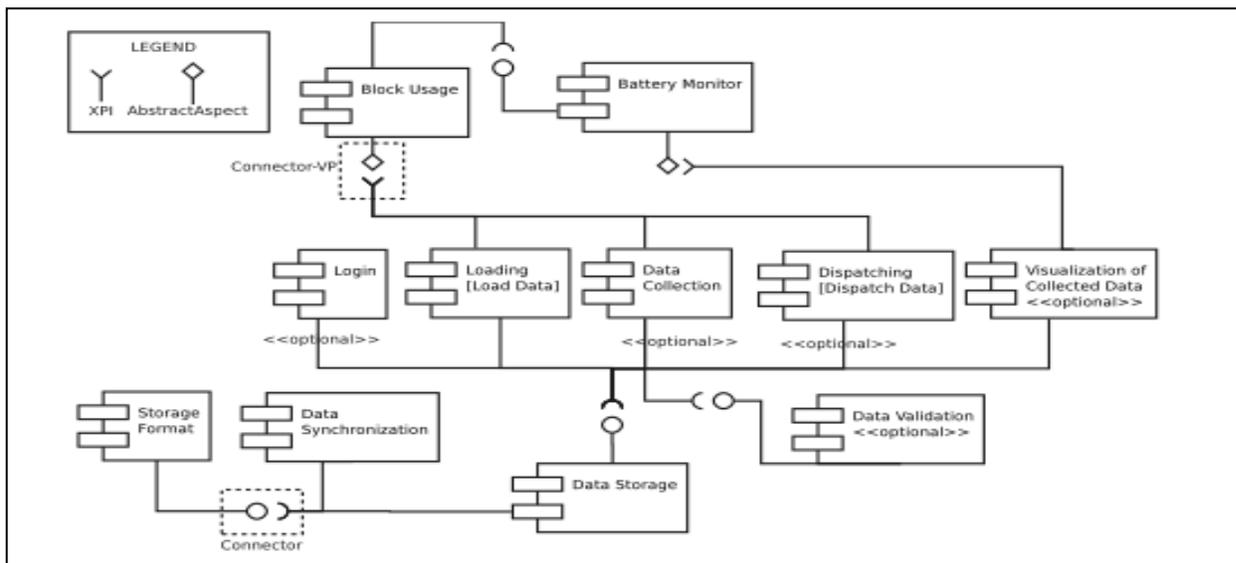


Figura 26 - Arquitetura para o domínio de coleta de dados. Fonte: Waku *et al.*, 2015.

- **Data Collection component:** Mapeamento do recurso de coleta de dados. O recurso de validação de dados (*Data Validation*) é utilizado pelo recurso de coleta de dados (*Data Collection*).
- **Data Validation component:** Mapeamento do recurso Validação de Dados (*Data Validation*). Esse componente é opcional pois o recurso também é opcional.
- **Data Storage component:** Mapeamento dos recursos armazenamento de dados (*Data Storage*), Memória interna do telefone (*Internal Phone Memory*), *SQLite*, Redundância de armazenamento (*Storage Redundancy*) e *SDCard*. Os componentes mapeados utilizam o componente formato de armazenamento (*Storage Format*).
- **Dispatching Data component:** Mapeamento do recurso de *Dispatching* além de seus sub-recursos.

- **Data Synchronization component:** Mapeamento dos recursos sincronização de dados (*Data Synchronization*), redundância de armazenamento (*Storage Redundancy*), o número do celular (*Mobile Number*), WI-FI e *Bluetooth*
- **Loading Data component:** Mapeamento dos recursos *Loading Data* e seus sub recursos.
- **Visualization of Collected Data component:** Mapeamento do recurso (*Visualization of Collected Data*).
- **Login component:** Mapeamento do recurso de Login.
- **Block Usage component:** Mapeamento do recurso transversal (*Block Usage*).
- **Battery Monitor component:** Mapeamento do recurso transversal Battery Monitor.
- **Data Storage component:** Componente que oferece duas APIs, uma com o recurso de redundância de dados e outra sem este recurso. Dando a escolha de tolerância a falhas para o cliente.

3.2.7. Privacy Protector Framework

O próximo trabalho analisado foi o trabalho de Luo *et al* (2018), o qual propõe um *framework* de coleta de dados voltado para a segurança de dados de usuários de sistemas de saúde baseados em IOT (*Internet of things*). Este *framework* é chamado de PrivacyProtector e junta a aplicação da técnica *Slepian-Wolf-coding-based secret sharing* (SW-SSS) combinando princípios de *secret sharing* e *share repairing* para manter a privacidade dos dados.

Tabela 17:

(Continua)

Requisito Arquitetural	Descrição
RA - 01	Coletar os dados através da interação dos dispositivos IOT com o corpo dos pacientes.
RA - 02	Enviar de forma segura os dados confidenciais dos pacientes para múltiplos servidores.
RA - 03	Persistir os dados de forma segura em uma nuvem de servidores distribuídos.

RA - 04	Prover interface de consulta segura para os dados privados.
RA - 05	Integração de múltiplos dispositivos para lidar com coletas de dados.
RA - 06	Redundância de armazenamento para prover segurança.
RA - 07	Evitar vazamento de dados dos usuários.

Tabela 17:

(Conclusão)

Requisito Arquitetural	Descrição
RA - 08	Evitar destruição de dados dos usuários.
RA - 09	Permitir divisão de pacotes privados utilizando SW-SSS.
RA - 10	Permitir reconstrução de pacotes privados utilizando SW-SSS.
RA - 11	Permitir reparo de pacotes privados utilizando SW-SSS.

Fonte: Luo *et al.*, (2018).

A abordagem arquitetural proposta pelo *framework* PrivacyProtector é composta por quatro componentes sendo estes:

Rede IOT: Rede composta de elementos de rede juntamente com vários sensores utilizados para coletar informações médica a respeito do paciente.

Provedor de serviço de comunicação: Meio pelo qual os dados dos pacientes serão enviados aos servidores distribuídos. É importante ressaltar que este elemento também é responsável pela preparação e distribuição dos pacotes de dados dos pacientes.

Sistema de storage distribuído: Guarda os dados dos pacientes que foram coletados através da rede IOT. Este sistema também oferece serviços de pesquisa de informações referentes aos pacientes.

O sistema de storage consiste basicamente de um banco de dados distribuído constituído de uma nuvem de servidores. Esta abordagem garante que as informações

dos pacientes ainda estejam protegidas, mesmo que somente um dos servidores esteja operante.

Sistema de controle de acesso de dados: Sistema utilizado para acessar os dados dos usuários localizados na base de dados distribuída.

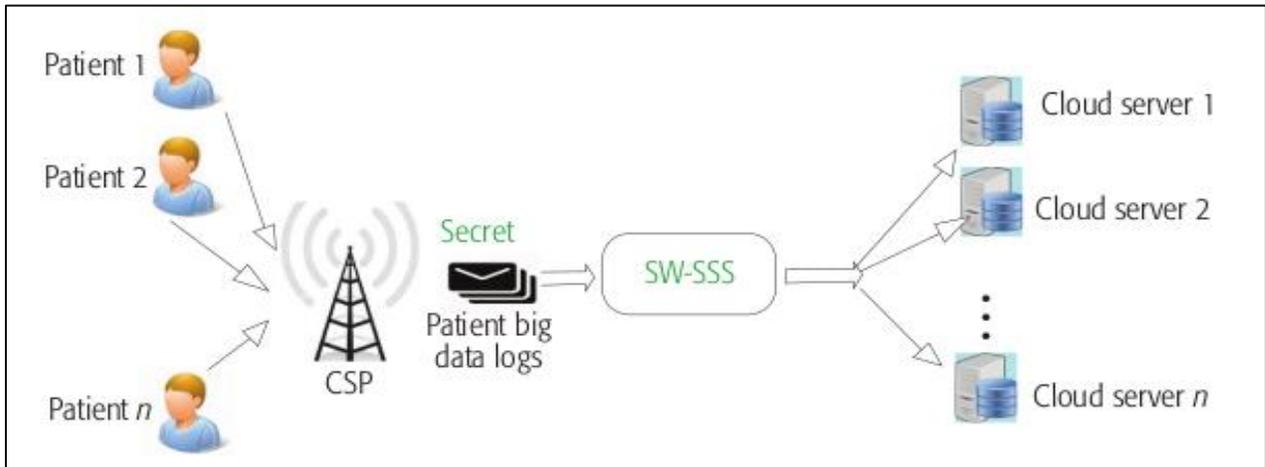


Figura x - Modelo para a arquitetura. Fonte: Luo et al., 2018.

Além dos elementos arquiteturais propostos, também é sugerida uma abordagem para fornecimento de informação dos pacientes através da colaboração dos múltiplos servidores que compõem o sistema de *storage* distribuído. Esta abordagem permite que os dados sejam disponibilizados sem que seu conteúdo seja revelado.

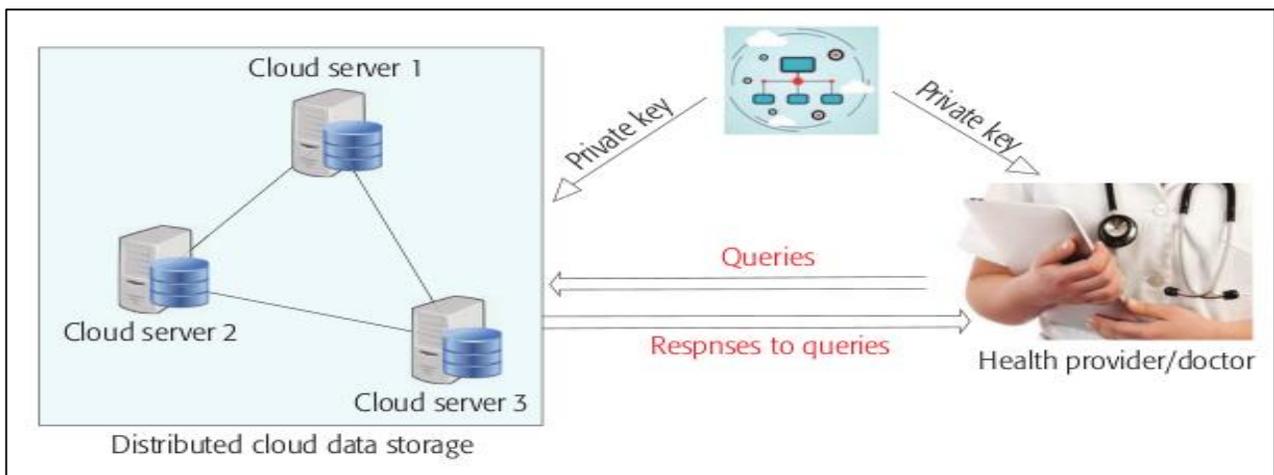


Figura x – Modelo de pesquisa realizada na arquitetura. Fonte: Luo et al., 2018.

3.2.8. IOT based data acquisition security

O próximo trabalho analisado foi o trabalho de Wang, Xu e Yang (2018), o qual demonstra uma abordagem de gerenciamento de dados baseada e IOT auxiliada com tecnologias em nuvem, tal método proporciona manter a confidencialidade de dados durante os processos de coleta, armazenamento e consulta dos mesmos.

O trabalho analisado é voltado para o eixo de segurança computacional, contudo abrange os principais requisitos e elementos arquiteturais de uma arquitetura voltada para o processo de coleta de dados de campo.

Tabela 18:

Requisito Arquitetural	Descrição
RA – 01	Coletar os dados através da interação dos dispositivos IOT com diversos agentes.
RA – 02	Enviar os dados coletados para múltiplos servidores.
RA – 03	Persistir os dados de forma distribuída.
RA – 04	Prover interface de consulta segura para os dados persistidos.
RA – 05	Prover integração de diferentes tipos de interface com os servidores.
RA – 06	Prover diferentes tipos de comunicação entre os servidores.
RA – 07	Prover métodos eficientes de armazenamento distribuído.

Fonte: Wang, Xu e Yang (2018).

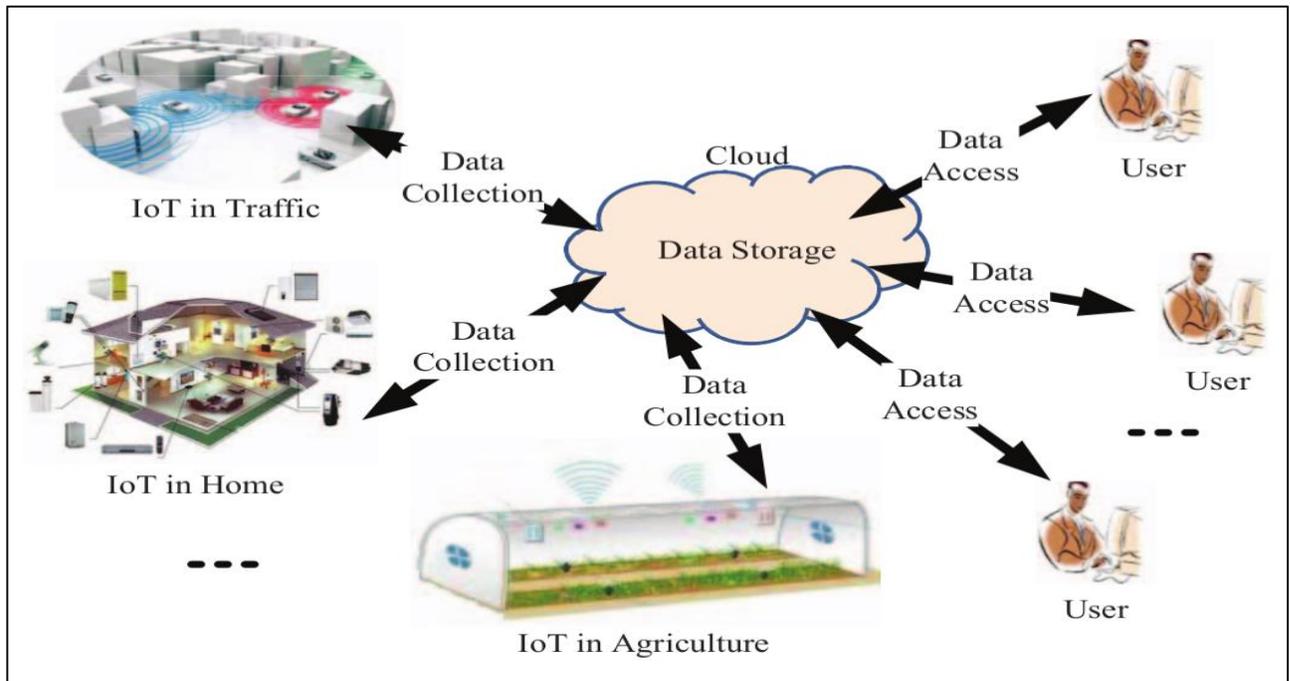


Figura x – Arquitetura para coleta de dados com IOT realizada. Fonte: WANG, XU e YANG 2018.

Ambiente IOT: Diferentes redes IOT utilizadas para coletar informações relevantes a seus respectivos domínios.

Ambiente de comunicação: Meio pelo qual os dados são enviados aos ambientes de storage.

Ambientes de storage: Guardam os dados coletados através das redes IOT.

Sistemas de acesso de dados: Sistemas utilizado para acessar os dados dos usuários localizados nas bases de dados.

3.3. CONCLUSÃO

Após a análise dos trabalhos foi possível obter algumas conclusões interessantes referentes ao domínio de coleta de dados. Percebeu-se que o domínio e o subdomínio pesquisados possuem uma grande quantidade de pesquisas e sistemas. Também foi verificado que embora o grande volume de soluções, publicações e trabalhos voltados para a coleta de dados, existe uma falta de uniformidade em relação a conceitos básicos que permeiam o domínio de coleta de dados e o subdomínio de coleta de dados móveis, de forma que dentre os doze trabalhos pesquisados somente o trabalho de Waku *et al.*, (2015) tem como meta o estabelecimento de uma estrutura padrão de instâncias voltadas para a coleta de dados móveis, a qual é definida através de uma linha de produto de software baseada em conhecimentos de domínio, especificamente o subdomínio de coleta de dados móveis.

Em virtude do exposto verifica-se que a elaboração de uma arquitetura de referência voltada o domínio analisado pode prover grandes melhorias no desenvolvimento de novas instâncias arquiteturais pertencentes ao mesmo domínio, uma vez que conhecimentos relevantes que permeiam o domínio de coleta de dados e o subdomínio de coleta de dados móveis estejam disponíveis em uma fonte de informação uniforme.

4. METODOLOGIA

Neste capítulo é abordado o processo de elaboração da arquitetura de referência proposta por este trabalho. Além disto este capítulo apresentará brevemente uma ferramenta construída a partir da AR proposta. O intuito do desenvolvimento da ferramenta foi proporcionar um protótipo arquitetural testável que pudesse validar atributos de qualidade da AR proposta.

4.1. ELABORAÇÃO DA ARQUITETURA DE REFERÊNCIA

Para realizar a construção desta abstração optou-se pela utilização do processo ProSA-RA, que é um método que estrutura o desenvolvimento, a representação e o teste de arquiteturas de referência, através da utilização de um modelo interativo e incremental de desenvolvimento composto de quatro passos sendo estes ra1-investigação das fontes de informação, ra2-estabelecimento dos requisitos arquiteturais, ra3-projeto arquitetural, ra4-avaliação arquitetural.

A escolha da utilização da metodologia ProSA-RA ocorreu devido a alguns fatores, primeiramente a vasta literatura disponível em relação a metodologia auxilia no processo de utilização e adaptação desta ferramenta para o contexto deste trabalho, em segundo lugar o processo ProSA-RA possui um nível de maturidade bastante interessante, uma vez que segundo Nakagawa *et al.*, (2014), já foi utilizado para o design representação e teste de arquiteturas de referência, sendo que seis destas AR foram o cerne de produções acadêmicas. Por fim o processo ProSA-RA dispõe de flexibilidade em relação a técnicas de desenvolvimento, representação e testes das arquiteturas com ele construídas.

4.1.1. Definição de modelo de domínio

A própria natureza das arquiteturas de referência necessita da definição de um domínio de aplicação, uma vez que o objetivo de tais abstrações é fornecer um repositório de conhecimentos e boas práticas arquiteturais para uma classe de aplicações. Este trabalho define como modelo de domínio sistemas de coletas de dados, sendo que tal modelo de domínio possui como sub-modelo de domínio sistemas de coletas de dados móveis.

4.1.2. Passo RA1 - investigação das fontes de informação

Para a execução desta etapa do processo optou-se por utilizar como fontes de informação pessoas, sistemas de informação, trabalhos e publicações envolvidos ou pertencentes ao modelo de domínio.

4.1.2.1. Pessoas

As pessoas consultadas nesta etapa foram o Prof. Dr. Eduardo Martins Guerra. Também foi realizada a coleta de informações com a equipe de engenharia de software da Empresa de Tecnologia da Informação e Comunicação do Estado do Pará (PRODEPA) sobre requisitos de sistemas pertencentes ao do modelo de domínio escolhido.

Por fim foi realizada a coleta de informações com os colaboradores Agência Eletrobras Eletronorte que são participantes do projeto de monitoramento fenológico da ilha de Germoplasma e que utilizam o sistema voltado para o auxílio do projeto.

4.1.2.2. Sistemas

Para a análise de investigação de fontes de informações foram coletados requisitos de sistemas e elementos arquiteturais referentes a quatro soluções. A primeira solução é utilizada pela Agência Eletrobras Eletronorte, mais especificamente a unidade localizada na cidade de Tucuruí-PA, com o objetivo de monitorar fenologicamente espécies da flora local, mantidas em ambiente controlado.

Também foi realizada a análise de três sistemas pertencentes ao domínio de coleta de dados, tais aplicações foram desenvolvidas por uma instituição pública confidencial.

4.1.2.3. Publicações

Para poder obter informações a respeito de aspectos arquiteturais de sistemas pertencentes ao domínio de aplicação optou-se pela realização de uma revisão da literatura.

Esta estratégia auxilia na busca de fontes de informação capazes de definir aspectos comuns aos sistemas que desempenhem coletas de dados. Alguns dos aspectos são estilos arquiteturais, padrões de projeto, requisitos funcionais e não funcionais, requisitos de sistema e requisitos de arquitetura. Fioravanti, Nakagawa e Barbosa (2010) definem que uma revisão sistemática de literatura que tenha um processo rigoroso, confiável e auditável é uma avaliação válida do tópico pesquisado. Isto se deve, pois, este processo já possui como meta a identificação, avaliação e interpretação de materiais disponíveis á respeito de um determinado tema de pesquisa ou questões que circundam o tema.

Tabela 19 - Revisão sistemática de sistemas de coletas de dados

Nome	Descrição
Questão de pesquisa	Quais são as funcionalidades, os requisitos e os aspectos arquiteturais comuns a sistemas que desempenham coletas de dados.
Palavras-chave	<i>data collection,</i> <i>data collecting architecture,</i> <i>data synchronization,</i> <i>synchronization architecture.</i> <i>reference architecture</i> <i>software architecture</i> <i>software reference architecture</i>
Critérios de seleção	Motores de busca de internet.
Idiomas	Inglês, Português.
Fontes	Google acadêmico, IEEE e Research gate.
String de busca	<i>“data collecting architecture”</i> OR <i>“data collecting software architecture”</i> OR <i>“data collecting software reference architecture”</i> OR <i>“data synchronization architecture”</i> OR <i>“coleta de dados móveis”</i> OR <i>“arquiteturacoleta de dados móveis”</i>

Fonte: Própria do autor

Após a realização da pesquisa nos motores de pesquisa propostos, utilizando as strings de busca descritas foram retornados 40 trabalhos e 18 livros nas plataformas Google acadêmico, IEEE e Research gate. Logo após esta etapa foi selecionado um subconjunto dos resultados obtidos, esta inclusão foi guiada pelo seguinte questionamento: *“O trabalho demonstra funcionalidades ou requisitos ou aspectos*

arquiteturais relevantes para o domínio proposto?”. Posteriormente a esta etapa obteve-se um conjunto de doze trabalhos, estes foram alvo de uma leitura mais refinada com o objetivo localizar quais funcionalidades, requisitos e aspectos arquiteturais são mais adequadas para compor a arquitetura de referência proposta por este trabalho.

4.1.3. Passo RA2 - estabelecimento dos requisitos arquiteturais

Nesta etapa do processo buscou-se agrupar os requisitos de sistema referentes ao modelo de domínio que foi proposto, posteriormente mapeando-os para requisitos arquiteturais, culminando na elaboração dos conceitos de domínio.

Baseado nas análises referentes aos requisitos e componentes arquiteturais realizados no capítulo três, foram reunidas funcionalidades relevantes para sistemas pertencentes ao domínio e ao subdomínio que este trabalho buscou pesquisar. Além disto, a análise realizada proporcionou ponderamentos referentes a outros elementos arquiteturais e sistêmicos que são pertinentes a construção de soluções voltadas para o domínio de coleta de dados.

4.1.3.1. Identificação dos requisitos de sistema

Sob a luz do trabalho de Fioravanti, Nakagawa e Barbosa (2010), o processo de organização das funcionalidades coletadas foi realizado por meio de divisão por categorias (Apêndice A). As categorias elaboradas para a organização das funcionalidades são: (1) Armazenamento, (2) Coleta, (3) Sincronização, (4) Erros, (5) Segurança, (6) Administração.

4.1.3.2. Identificação dos requisitos arquiteturais

Uma vez identificados os requisitos sistêmicos, foi necessário realizar uma análise cuidadosa para efetivamente mapear os requisitos listados para requisitos arquiteturais ap. Os artefatos conhecidos como requisitos arquiteturais serviram de insumos para o desenvolvimento da arquitetura de referência (Apêndice B).

4.1.3.3. Identificação de conceitos de domínio

Após o mapeamento dos requisitos de sistema para requisitos arquiteturais, realizou-se o mapeamento dos requisitos arquiteturais definidos para conceitos de domínio (Apêndice C). Os conceitos de domínio são necessários para a execução do passo seguinte RA3 - Projeto arquitetural.

4.1.4. Passo RA3 - projeto arquitetural

Neste passo buscou-se construir uma representação da estrutura arquitetural da arquitetura de referência, sob diferentes pontos de vista. O processo ProSA-RA sugere a utilização de quatro abordagens distintas, estas abordagens são chamadas de pontos de vista, que são a compreensão dos diferentes aspectos da arquitetura sob uma determinada abstração.

Os pontos de vista propostos pelo processo ProSA-RA são os pontos de vista transversal, execução, de implantação e código fonte. Também é importante salientar que estes são subdivididos em especializações chamadas de visualizações.

4.1.4.1. Ponto de vista transversal

Neste ponto de vista foram definidos aspectos gerais da AR, como por exemplo terminologias, definições relevantes e elementos variáveis. Este ponto de vista é subdividido em diferentes visões, sendo estas a visão conceitual, funcionalidade, serviços de sistema, variabilidade.

4.1.4.1.1. *Visão conceitual*

Nesta visão buscou-se descrever conceitos identificados e seus relacionamentos. Para descrição dos conceitos foi utilizado um glossário e para demonstrar os relacionamentos entre os conceitos foi utilizado um diagrama de definição de bloco (Apêndice D).

4.1.4.1.2. Visão de funcionalidade

Esta visão foi utilizada para descrever as principais funcionalidades e os seus possíveis consumidores. Para realizar esta descrição foi utilizado um diagrama de caso de uso (Apêndice E).

4.1.4.1.3. Visão de serviços de sistema

Definiu-se um conjunto base de funções que necessitam ser suportados em qualquer implementação da AR. Demonstrou-se tal conjunto de serviços através de um diagrama de definição de bloco (Apêndice F).

Os principais elementos do diagrama proposto são os elementos de manutenção de dados, comunicação, sincronização e gerência de erros. Tais elementos são detalhados abaixo:

- **Manutenção de dados:** Este bloco encapsula as funcionalidades de criação, consulta, edição e deleção que serão disponíveis nos módulos de coleta de dados.
- **Comunicação:** Este bloco representa os aspectos característicos referentes à comunicação, por exemplo, a escolha do formato que os dados serão enviados, a fragmentação e reestruturação dos pacotes de dados durante o envio, o tipo de envio, o tratamento da comunicação entre dois módulos pertencentes à arquitetura e o gerenciamento de operações de acordo com o sinal.
- **Sincronização:** Representa os aspectos referentes à importação, exportação e replicação necessários para o correto funcionamento dos módulos pertencentes a arquitetura de referência, como por exemplo o tipo de sincronização. Além disto, este bloco busca contemplar o gerenciamento da fila de elementos que serão sincronizados entre os módulos.
- **Gerência de erros:** Representa o gerenciamento de erros proposto para cada módulo pertencente à arquitetura de referência. Este bloco será transposto em uma série de componentes presentes em cada módulo da arquitetura de referência, tais componentes serão responsáveis pelo gerenciamento de erros oriundos de outros módulos e componentes internos da arquitetura de referência.

4.1.4.1.4. Visão de variabilidade

Nesta etapa foram descritos elementos da arquitetura utilizando o diagrama de definição de bloco (Apêndice G). Esta visão foi utilizada para definir algumas das diferentes formas de instanciação dos elementos arquiteturais.

Algo fundamental para a arquitetura de referência proposta neste trabalho é que a mesma possa ser instanciada de formas diferentes de acordo com as necessidades de projeto, além disso, o modelo de domínio de coleta de dados pode utilizar diversas estratégias de coleta, comunicação, sincronização e armazenamento de dados. Estes fatores corroboram com a estratégia de divisão da AR em módulos que compõem sua estrutura, sendo esta uma abordagem fundamental para garantir sua adaptabilidade a cada caso.

O diagrama de bloco proposto para a representação da visão de variabilidade é composto por seis blocos e dez conexões. Os blocos representam os três módulos principais que compõem a arquitetura de referência, juntamente com os mecanismos de armazenamento de dados que cada módulo possui. Já as conexões representam respectivamente o fluxo de informações entre cada módulo e a possibilidade de arranjo dos entre os módulos. A variabilidade referente a cada bloco está descrita abaixo.

- **Módulo de armazenamento remoto:** Este módulo representa a estrutura de armazenamento que será utilizada pelo módulo de coleta móvel. Em relação as diferentes formas de instância possíveis para este módulo, podem ser utilizadas bases de dados que buscam estruturar os dados de uma forma específica, como por exemplo bases de dados relacionais, bases de dados orientadas a objeto, bases de dados NoSql, ou até mesmo estruturas de armazenamento customizadas, como por exemplo, uma estrutura customizada de diretórios, juntamente a arquivos onde as informações são armazenadas e regras de navegação, criação, alteração leitura e deleção de tais diretórios e arquivos.

Este módulo, assim como os outros módulos de armazenamento presentes nesta visão, pode requerer alterações e transformações dos dados que foram coletados pelo módulo de coleta, tais alterações se fazem necessárias quando o tipo de dado necessita ser armazenado em um formato específico, geralmente requerido pela

ferramenta de armazenamento específica, como demonstrado na publicação de Santáanna (2014).

- **Módulo de armazenamento principal:** Este módulo representa a estrutura de armazenamento que será utilizada pelo módulo de servidor principal. Muito similar ao módulo de armazenamento remoto, as bases de dados utilizadas para o armazenamento e disponibilização final dos dados buscam estruturar os dados de uma forma específica, também sendo necessário em alguns casos o processo de transformação dos dados para o seu correto armazenamento.

Este módulo se diferencia do módulo de armazenamento remoto a parti do momento que permite que a mesma base seja replicada de forma estruturada para outras bases, o que aumenta a disponibilidade dos dados. Ou seja este módulo pode ser constituído de diversos servidores de armazenamento de dados operando como um só.

- **Módulo de armazenamento de sincronia:** Este módulo representa a estrutura de armazenamento que será utilizada pelo módulo de sincronização opcional. De forma similar aos módulos de armazenamento remoto e principal, as bases de dados utilizadas para o armazenamento e disponibilização final dos dados buscam estruturar os dados de uma forma específica, também sendo necessário em alguns casos o processo de transformação dos dados para o seu correto armazenamento. As variações para este módulo são intrínsecas ao tipo de informação que será compartilhada entre os módulos e o volume de compartilhamento.

Sendo assim, se a necessidade de projeto for um fluxo constante de informações advindas de fontes heterogêneas devido a mobilidade podem vir a coletar os mesmos dados, este módulo então não fará o tratamento de registros únicos, pois o seu intuito é fornecer um fluxo contínuo e rápido para os módulos de coleta.

- **Módulo de coleta de dados:** Este módulo representa o sistema pelo qual a coleta de dados será realizada. A variabilidade deste módulo é intrínseca às necessidades de projeto, ou seja, é possível que o mecanismo de coleta utilizado em projetos diferentes possa variar, desta forma, algumas instâncias da arquitetura podem compreender os mecanismos de coleta como dispositivos móveis, tais como celulares, *palmtops* e PDAs, enquanto que outras instâncias arquiteturais podem ser desenvolvidas para

utilizarem um sensor ou uma rede de sensores para atuarem como dispositivos de coleta.

Pode-se perceber que diferentemente dos módulos de servidor principal e sincronização opcional, o módulo de coleta apresenta uma grande possibilidade de variabilidade, no sentido de que as diferentes camadas que compõem o módulo de coleta podem ser instanciadas e implementadas de formas completamente diferentes. Desta forma, apesar dos módulos de servidor principal e de sincronização possam ser concebidos como aplicações que se diferem por seus estilos arquiteturais, padrões de projeto, linguagens de programação e design de solução em geral, estes são muito mais consistentes em relação à sua natureza, ou seja, em diferentes instâncias da arquitetura estes módulos continuarão sendo aplicações que expõem seus serviços de acordo com algumas estratégias, tendo como meta agrupar e disponibilizar os dados coletados.

Diferente do módulo servidor principal e o módulo de sincronização opcional, o módulo de coleta de dados pode variar completamente de acordo com as diferentes instâncias da arquitetura de referência. A interface de coleta pode ser composta de um sensor, ou uma rede de sensores, ou um dispositivo móvel ou uma rede de dispositivos móveis, o módulo de armazenamento remoto e suas políticas de acesso, que são partes integrantes do módulo de coleta de dados, também podem ser drasticamente diferentes de acordo com a instância da AR.

As estruturas internas que compõem este módulo podem ser agrupadas em uma única máquina, responsável pelo correto processamento e armazenamento de todos dados coletados por todas as interfaces de coleta, denotando uma clara divisão entre os módulos de coleta de dados e de armazenamento remoto, sendo este cenário muito comum em sistemas que utilizam sensores para a realização da coleta de dados, como pode ser visto nos trabalhos de Silva, Andrade e Marin (2008), Tian, Gao e Zhou (2008), Zhao, Zhang e MENG (2012), Sant'anna (2014). Por outro lado, se analisarmos sob a perspectiva de coleta de dados móveis, os módulos de coleta de dados e módulo de armazenamento remoto podem compor uma única unidade de coleta, este cenário é caracterizado pela utilização de dispositivos que disponibilizem a interface para coleta, as regras internas da camada de aplicação e as bases de dados em uma única unidade, ou seja, um dispositivo móvel, como demonstrados nos trabalhos de Boni *et al.*, (2006),

Brega *et al.* (2008), Ji *et al.*, (2010), Santos *et al.*, (2013), Freitas Filho (2014) e Waku *et al.*, (2015).

- **Módulo servidor principal:** Este módulo representa o sistema responsável por receber os dados fornecidos pelo(s) módulo(s) de coleta, validá-los, enfileirá-los, processá-los e armazená-los, para que os mesmos possam ser futuramente disponibilizados.

Como exemplificado anteriormente, a variabilidade deste módulo encontra-se na possibilidade deste ser instanciado utilizando-se de diversas abordagens de design, estilos arquiteturais, padrões de projeto, ferramentas de implementação e linguagens de programação. Por exemplo, as instâncias deste módulo podem adotar a divisão de camadas para cada módulo interno que compõem a camada de aplicação (Apêndice H). Além disto, podem ser adotados diferentes tipos de implementação para controlar o fluxo de informação interno da aplicação, para a exposição dos serviços dispostos, para a comunicação de dados e para a comunicação com as bases de dados.

As estratégias de instância para os módulos internos e componentes pertencentes à aplicação estão parcialmente atreladas às escolhas de ferramentas utilizadas para a implementação da arquitetura instanciada.

É importante ressaltar que dependendo da necessidade do projeto, o servidor principal será o responsável pelo enfileiramento e preparação para a validação em termos de sincronização.

- **Módulo de sincronização opcional:** Este módulo representa o sistema responsável por receber os dados fornecidos pelo(s) módulo(s) de coleta, armazená-los, enfileirá-los, processá-los e repassá-los para o servidor principal.

Sob a perspectiva de variações este módulo pode pertencer ao servidor principal, pode ser um servidor individual disponibilizado separadamente do módulo servidor principal, ou pode nem mesmo ser necessário. Estas possibilidades estão diretamente relacionadas com as necessidades requeridas pela instância da arquitetura de referência.

De forma similar ao módulo servidor principal, a variabilidade deste módulo encontra-se na possibilidade deste ser instanciado utilizando-se de diversas abordagens de design e ferramentas para sua futura implementação.

4.1.4.2. Ponto de vista de tempo de execução

Neste ponto de vista foram abordados componentes, relacionamentos e fluxos de informação da AR que são observados uma vez que a arquitetura esteja instanciada e funcionando. As visões presentes neste ponto de vista são as visões componentes colaborativos, processo, dados compartilhados e implantação.

4.1.4.2.1. Visão de componentes colaborativos

Nesta visão foi realizada uma descrição da arquitetura de referência de acordo com a interação de seus componentes e repositórios de dados. Optou-se pela utilização do diagrama de componentes para realizar esta representação (Apêndice H).

Nesta visão os módulos da aplicação foram divididos em três módulos que compõem a arquitetura de referência, sendo estes um módulo de coleta de dados, um módulo servidor principal e por fim um módulo de sincronização opcional.

Cada módulo da aplicação foi subdividido em três camadas, o que proporciona maior organização para a estrutura de cada módulo. A definição de cada componente segue-se na Tabela 18.

Tabela 20 - Descrição dos componentes do módulo de coleta

(Continua)

Camada	Nome	Descrição
Apresentação	Aquisição de dados	Representa a interface com o usuário, ou a interface de coleta de sensores.
Aplicação	Gerenciamento de recursos	Componente responsável por disponibilizar os recursos necessários para a realização da coleta e sincronização de dados, mediante as condições ideais, como por exemplo, quantidade adequada de bateria e nível de sinal adequado.
	Gerenciamento de bateria	Componente que proverá informações referentes ao nível de energia do aparelho de coleta. Dependendo da instância da arquitetura este módulo também é responsável por recarregar a bateria interna do módulo de coleta. Este componente é frequentemente consultado pelo módulo de Gerencia de recursos.

Tabela 20 - Descrição dos componentes do módulo de coleta

(Conclusão)

Camada	Nome	Descrição
--------	------	-----------

	Gerenciamento de sinal	Componente que proverá informações referentes ao nível de energia do aparelho de coleta. Este módulo é frequentemente ser consultado pelo módulo de Gerenciamento de recursos. Este componente trabalha em conjunto com o módulo de sincronização, uma vez que as decisões do tamanho dos pacotes e da estratégia de sincronização com o módulo servidor principal são extremamente influenciadas pelo tipo e qualidade do sinal disponível.
	Gerenciamento de erros	Componente que proverá informações referentes aos erros que ocorreram em uma determinada execução. Este componente interage constantemente com todos os outros módulos, monitorando, gerenciando e provendo informações referentes aos erros ocorridos.
	Segurança	Componente que proverá a segurança ao comunicar-se com o módulo servidor principal, abrangendo as operações de autenticação de interface, autenticação de usuário e autorização de usuário.
	Comunicação	Componente que proverá a possibilidade de escolha de formatos de comunicação, gerenciamento do tamanho de pacotes e mapeamento de códigos de resposta do servidor.
	Sincronização	Componente capaz de verificar a existência de dados sincronizáveis, gerenciar a importação, exportação e sincronização dos dados entre o módulo de coleta e o módulo servidor.
	Gerenciamento de dados	Componente responsável por gerenciar a consistência dos dados, a validação interna dos dados, a aplicação das regras de negócio, e a transformação dos dados para o correto armazenamento e recuperação dos dados do banco de dados.
Persistência	Base de dados interna	Componente responsável pela principal forma de armazenamento dos dados.
	Base de dados externa	Componente responsável pelo backup da base de dados.
	Base de dados customizada	Componente responsável pela principal forma de armazenamento dos dados dependendo da instancia. Pode ser instanciado através de uma estrutura de diretórios e arquivos que obedecem a padrões e políticas de criação, consulta, edição e deleção.

Fonte: Própria do autor

Tabela 21 - Descrição dos componentes do módulo servidor principal

Camada	Nome do componente	Descrição
--------	--------------------	-----------

Apresentação	Gateway de serviço	Componente responsável por direcionar o fluxo de requisição para o serviço correto.
	Gerenciamento de recursos	Componente responsável por verificar anormalidades nos recursos do servidor principal e executar ações para manter a disponibilidade da aplicação.
	Gerenciamento de comunicação	Componente responsável por gerir as respostas provenientes do servidor principal.
Aplicação	Gerenciamento de serviços	Componente responsável pelo correto fluxo de informação dentro da aplicação.
	Gerenciamento de erros	Componente que proverá informações referentes aos erros que ocorreram em uma determinada execução. Este componente interage constantemente com todos os outros módulos, monitorando, gerenciando e provendo informações referentes aos erros ocorridos.
	Segurança	Componente que proverá a segurança ao comunicar-se com o módulo servidor principal, abrangendo as operações de autenticação de interface, autenticação de usuário e autorização de usuário.
	Sincronização	Componente responsável por verificar inconsistências oriundas da sincronização de novos dados, providos pelos módulos de coleta móvel e manter a coesão dos dados mantidos pelo servidor principal.
	Gerenciamento de dados	Componente responsável por gerenciar a consistência dos dados, a validação interna dos dados, a aplicação das regras de negócio, e a transformação dos dados para o correto armazenamento e recuperação dos dados do banco de dados.
Persistência	Base de dados estruturada	Componente responsável por manter todos os tipos de dados que podem ser considerados como estruturados.
	Base de dados não estruturada	Componente responsável por manter todos os tipos de dados que podem ser considerados como não estruturados.

Fonte: Própria do autor

Tabela 22 - Descrição dos componentes do módulo de sincronização opcional

Camada	Nome do componente	Descrição
Apresentação	Gateway de serviço	Componente responsável por direcionar o fluxo de requisição para o serviço correto.
	Gerenciamento de comunicação	Componente responsável por gerir as respostas provenientes do servidor principal.
Aplicação	Gerenciamento de recursos	Componente responsável por verificar anormalidades nos recursos do servidor principal e performar ações para manter a disponibilidade da aplicação.
	Gerenciamento de serviços	Componente responsável por gerenciar o fluxo de informação dentro da aplicação.
	Gerenciamento de fila	Componente responsável pela ordenação dos dados que podem ser sincronizados.
	Gerenciamento de erros	Componente que proverá informações referentes aos erros que ocorreram em uma determinada execução. Este componente interage constantemente com todos os outros módulos, monitorando, gerenciando e provendo informações referentes aos erros ocorridos.
	Gerenciamento de dados	Componente responsável por gerenciar a consistência dos dados, a validação interna dos dados, a aplicação das regras de negócio, e a transformação dos dados para o correto armazenamento e recuperação dos dados do banco de dados.
Persistência	Base de dados estruturada	Componente responsável por manter todos os tipos de dados que podem ser considerados como estruturados.
	Base de dados não estruturada	Componente responsável por manter todos os tipos de dados que podem ser considerados como não estruturados.

Fonte: Própria do autor

4.1.4.2.2. Visão de processo

Nesta visão foi realizada uma descrição da arquitetura de referência como um conjunto de unidades executadas concorrentemente, sendo representada como diagrama de bloco interno (Apêndice I).

Assim como na visão de componentes colaborativos, a abordagem que foi utilizada para a construção da visão de processos foi a divisão da arquitetura de referência em três módulos principais, sendo estes o módulo de coleta, o módulo servidor principal e o módulo de sincronização opcional.

Nesta visão cada um dos três módulos é composto por componentes e módulos internos, sendo estes respectivamente uma unidade especializada na execução de uma tarefa e um agrupamento de unidades especializadas que executam tarefas diferentes pertencentes a um mesmo contexto. Em virtude desta abordagem foram propostos para o módulo de coleta de dados quatro módulos internos e vinte e cinco componentes. Para o módulo servidor principal foram propostos quatro módulos internos e vinte componentes. Por fim para o módulo de sincronização opcional foram propostos quatro módulos e dezesseis componentes.

Tabela 23 - Definição dos módulos internos

(Continua)

Módulo	Nome	Descrição
Coleta	Gerenciamento de fluxo	Compreende processos referentes a coleta de informação e acesso dos recursos disponibilizados pelo módulo.
	Gerenciamento de dados	Compreende processos referentes a validação e regras de negócio, transformação de dados e gerenciamento dos dados da base de dados.
	Comunicação	Compreende processos referentes ao gerenciamento de tipo de requisições, o volume transmitido a cada requisição, o tratamento e o gerenciamento de cada tipo de resposta.
	Sincronização	Compreende processos referentes a gerência de dados sincronizáveis, a escolha do tipo de sincronizações, o estado da unidade de coleta, a realização de operações de importação exportação e replicação e a notificação da execução de tais processos.
Servidor principal	Gerência de serviços	Compreende processos referentes a exposição dos serviços fornecidos pelo servidor principal.
	Gerenciamento de dados	Compreende processos referentes a validação e regras de negócio, transformação de dados e gerenciamento dos dados da base de dados.
	Comunicação	Compreende processos referentes ao gerenciamento de tipo de requisições, o volume transmitido a cada requisição, o tratamento, o gerenciamento de cada tipo de resposta e o envio correto de cada tipo de resposta.

Tabela 23 - Definição dos módulos internos

(Conclusão)

Módulo	Nome	Descrição
Servidor principal	Sincronização	Compreende processos referentes a organização dos dados que serão sincronizados, a realização de operações de importação exportação e replicação e a garantia da não repetição de registros.
Sincronização	Gerenciador de serviços	Compreende processos referentes a exposição dos serviços fornecidos pelo servidor de sincronização opcional.
	Gerência de dados	Compreende processos referentes a validação e regras de negócio, transformação de dados e gerenciamento dos dados da base de dados.
	Comunicação	Compreende processos referentes ao gerenciamento de tipo de requisições, o volume transmitido a cada requisição, o tratamento, o gerenciamento de cada tipo de resposta e o envio correto de cada tipo de resposta.
	Sincronização	Compreende processos referentes a organização dos dados que serão sincronizados, a realização de operações de importação exportação e replicação e a garantia da não repetição de registros.

Fonte: Própria do autor

Tabela 24 - Definição dos componentes

(Continua)

Módulo	Nome	Descrição
Coleta	Gerência de erros	Bloco de processo que é utilizado para o tratamento de mensagens de comunicação em relação à ocorrência de erros.
	Gerência de recursos	Bloco de processo responsável pela disponibilização e bloqueio de recursos do módulo de coleta. Os processos de bloqueio e disponibilização são realizados de acordo com a consulta dos componentes de Gerência de bateria e Gerência de sinal consulta do nível de bateria e qualidade de sinal.
	Gerência de sinal	Bloco de processo responsável pela verificação da existência, do tipo e da qualidade do sinal sem fio que o módulo de coleta possui. Este processo influencia diretamente os processos de Gerência de erros, Gerência de volume de requisição, Gerência de importação, Gerência de exportação e Gerência de sincronização,

Tabela 24 - Definição dos componentes

(Continua)

Módulo	Nome	Descrição
---------------	-------------	------------------

Coleta	Gerência de bateria	<p>Bloco de processo responsável pela verificação do nível de energia restante na bateria do módulo de coleta.</p> <p>Dependendo da implementação escolhida este bloco de processo pode conter diversos componentes, incluindo um gerenciador de nível de bateria e um módulo de recarga de bateria.</p> <p>Este processo influencia diretamente todos os processos pertencentes ao módulo de coleta, uma vez que sem energia torna-se impossível utilizar qualquer serviço do módulo de coleta móvel. É importante ressaltar que este aspecto é relevante somente para as implementações que necessitem de baterias para a realização da coleta de dados.</p>
	Interface de coleta	Bloco de processo que representa a coleta de informações através de uma interface com o mundo.
	Validação externa	Bloco de processo que representa a análise de corretude inicial aplicado as informações coletadas no bloco Interface de coleta.
	Validação interna	Bloco de processo que representa a análise de corretude final aplicado as informações analisadas no bloco Validação externa.
	Regras de negócio	Bloco de processo que representa a análise de corretude das informações analisadas no bloco Validação interna de acordo com regras específicas do negócio proposto.
	Transformação de dados	Bloco de processo que representa a transformação dos dados para seu armazenamento e para a manipulação dentro da aplicação.
	Gerência de base de dados interna	Bloco de processo que representa o controle da integridade da(s) base(s) de dados disposta(s) no módulo de coleta. Compreendendo controle transacional, mapeamento de entidades e padrões que facilitam o acesso aos dados.
	Gerência de base de dados externa	Bloco de processo que representa o controle de acesso da(s) base(s) de dados disposta(s) no módulo de coleta como forma de armazenamento extra, ou armazenamento de dados específicos.

Tabela 24 - Definição dos componentes

(Continua)

Módulo	Nome	Descrição
--------	------	-----------

Coleta	Gerência de dados sincronizáveis	Bloco de processo que representa a localização de dados que podem sofrer o processo de exportação para posterior realização de pareamento com o servidor principal.
	Gerência de status da unidade	Bloco de processo que representa a verificação do estado inicial para que o módulo de coleta de dados possa ser utilizado.
	Gerência do tipo de sincronização	Bloco de processo que representa a gerência de qual método de sincronização será utilizado, uma sincronização cabeada ou uma sincronização sem fio.
	Gerência de importação	Bloco de processo que representa as tarefas necessárias para a realização da importação de dados para a base de coleta de dados. Este bloco comunica-se principalmente com os blocos de processo pertencentes ao módulo de comunicação.
	Gerência de exportação	Bloco de processo que representa as tarefas necessárias para a realização da exportação de dados para o módulo de sincronização opcional, ou para o módulo servidor principal. Este bloco comunica-se principalmente com os blocos de processo pertencentes ao módulo de comunicação.
	Gerência de sincronização	Bloco de processo que representa as tarefas necessárias para a realização da sincronização ou replicação de dados entre base de coleta de dados e o módulo servidor principal. Este bloco comunica-se principalmente com os blocos de processo pertencentes ao módulo de comunicação.
	Notificação de operação	Bloco de processo que representa as tarefas necessárias para a comunicação para os diferentes blocos de processo da aplicação, em especial blocos pertencentes ao módulo interno Gerenciador de dados.
	Gerência de tipo de requisição	Bloco de processo que representa a seleção do tipo de comunicação que será utilizada.

Tabela 24 - Definição dos componentes

(Continua)

Módulo	Nome	Descrição
Coleta	Gerência de volume de requisição	Bloco de processo que representa a seleção do tamanho do pacote de dados que será enviado no processo de comunicação.

		Este bloco de processo está diretamente relacionado com o bloco de Gerência de sinal o qual é importantíssimo para a determinação do tamanho do pacote a ser transmitido.
	Gerência de tipo de resposta	Bloco de processo que representa a primeira etapa em relação ao tratamento das respostas. Este processo pode conter a avaliação de códigos de resposta padrão, utilizados entre os três módulos que compõem a arquitetura de referência.
	Tratamento de respostas	Bloco de processo que representa a segunda etapa em relação ao tratamento das respostas. Este processo agrupa as comunicações com os processos de Gerência de dados para possíveis inserções na base de dados, Notificação de operação, Gerência de importação, Gerência de exportação e Gerência de sincronização para a continuação destes processos.
	Base de dados interna	Este bloco de processo representa a base de dados principal utilizada em dispositivos móveis.
	Base de dados externa	Este bloco de processo representa a base de dados que pode armazenar cópias da base de dados principal e dados não estruturados como imagens, arquivos de áudio e vídeo.
	Base de dados customizada	Este bloco de processo representa uma base de dados utilizada para o armazenamento dos dados coletados em campo. Pode-se citar como exemplo estruturas customizadas de diretórios, arquivos, políticas de criação, leitura, edição e deleção.
Servidor principal	Gerência de erros	Bloco de processo responsável pelo gerenciamento de eventos que possam colocar em risco a disponibilidade, integridade e confiabilidade dos serviços disponibilizados pelo servidor principal.
	Gerência de recursos	Bloco de processo responsável pela disponibilização de recursos de redundância mediante à ocorrência de erros que possam comprometer a disponibilidade dos serviços do servidor principal.

Tabela 24 - Definição dos componentes

(Continua)

Módulo	Nome	Descrição
	Autenticação de interface	Bloco de processo responsável pela verificação de autenticidade da interface que está realizado a

Servidor principal		requisição, seja esta o módulo de coleta ou o módulo de sincronização opcional. Este processo é importantíssimo para evitar o processamento de requisições de fontes suspeitas.
	Autenticação de usuário	Bloco de processo responsável pela verificação de autenticidade do usuário que está realizado a requisição. Este processo depende da implementação escolhida para a arquitetura de referência.
	Autorização de usuário	Bloco de processo responsável pela verificação das permissões de um usuário autenticado em relação aos serviços que foram requisitados.
	Gerência de serviços	Bloco de processo que agrupa as tarefas de distribuição de requisições para as referentes áreas que disponibilizam os serviços requisitados.
	Validação interna	Bloco de processo que representa a análise de corretude das informações recebidas na requisição.
	Regras de negócio	Bloco de processo que representa a análise de corretude das informações analisadas no bloco Validação interna de acordo com regras específicas do negócio proposto.
	Transformação de dados	Bloco de processo que representa a transformação dos dados para seu armazenamento e para a manipulação dentro da aplicação.
	Gerência de base de dados estruturada	Bloco de processo que representa o controle da integridade da(s) base(s) de dados disposta(s) no módulo de coleta. Compreendendo controle transacional, mapeamento de entidades e padrões que facilitam o acesso aos dados.
	Gerência de base de dados não estruturada	Bloco de processo que representa o controle de acesso da(s) base(s) de dados disposta(s) no módulo de coleta como forma de armazenamento extra, ou armazenamento de dados específicos.
	Gerência de fila	Bloco de processo que organiza a estrutura de requisições que objetivam o transporte de informações coletadas nos dispositivos de coleta.
Gerência de consistência	Bloco de processos que representa a verificação do registro proposto em relação a aplicação.	

Tabela 24 - Definição dos componentes

(Continua)

Módulo	Nome	Descrição
	Gerência de repetição	Bloco de processos que representa a verificação da duplicidade de registros na base de dados.

Servidor principal	Gerência do tipo de sincronização	Bloco de processo que representa a gerência de qual método de sincronização será utilizado, uma sincronização cabeada ou uma sincronização sem fio.
	Gerência do tipo de requisição	Bloco de processo que representa a seleção do tipo comunicação que será utilizada.
	Gerência do tipo de resposta	Bloco de processo que representa a primeira etapa em relação ao tratamento das respostas. Este processo pode conter a avaliação de códigos de resposta padrão, utilizados entre os três módulos que compõem a arquitetura de referência.
	Tratamento de resposta	Bloco de processo que representa a segunda etapa em relação ao tratamento das respostas. Este bloco de processo tem o intuito de agrupar processos refinados de processamento de resposta baseado no conteúdo de cada resposta.
	Base de dados estruturada	Bloco de processo que representa a base de dados utilizada para realização da persistência de dados estruturados.
	Base de dados não estruturada	Bloco de processo que representa a base de dados utilizada para realização da persistência de dados não estruturados, como arquivos de imagens, vídeo e áudio.
Sincronização	Gerência de erros	Bloco de processo responsável pelo gerenciamento de eventos que possam colocar em risco a disponibilidade, integridade e confiabilidade dos serviços disponibilizados pelo módulo de sincronização opcional.
	Gerência de recursos	Bloco de processo responsável pela disponibilização recursos de redundância mediante à ocorrência erros que possam comprometer a disponibilidade dos serviços do servidor principal.
	Autenticação de interface	Bloco de processo responsável pela verificação de autenticidade da interface que está realizado a requisição, a interface é o módulo de coleta de dados. Este processo é importantíssimo para evitar o processamento de requisições de fontes suspeitas.

Tabela 24 - Definição dos componentes

(Continua)

Módulo	Nome	Descrição
---------------	-------------	------------------

Sincronização	Gerência de serviços	Bloco de processo que agrupa as tarefas de distribuição de requisições para as referentes áreas que disponibilizam os serviços requisitados.
	Transformação de dados	Bloco de processo que representa a transformação dos dados para seu armazenamento e para a manipulação dentro da aplicação.
	Gerência de base de dados estruturada	Bloco de processo que representa o controle da integridade da(s) base(s) de dados disposta(s) no módulo de coleta. Compreendendo controle transacional, mapeamento de entidades e padrões que facilitam o acesso aos dados.
	Gerência de base de dados não estruturada	Bloco de processo que representa o controle de acesso da(s) base(s) de dados disposta(s) no módulo de coleta como forma de armazenamento extra, ou armazenamento de dados específicos.
	Gerência de fila	Bloco de processo que organiza a estrutura de requisições que objetivam o transporte de informações coletadas nos dispositivos de coleta.
	Gerência de consistência	Bloco de processos que representa a verificação do registro proposto em relação a aplicação.
	Gerência de repetição	Bloco de processos que representa a verificação da duplicidade de registros na base de dados. Este bloco de processo depende da implementação escolhida para a arquitetura de referência.
	Gerência do tipo de sincronização	Bloco de processo que representa a gerência de qual método de sincronização será utilizado, uma sincronização cabeada ou uma sincronização sem fio.
	Gerência do tipo de requisição	Bloco de processo que representa a seleção do tipo comunicação que será utilizada.
	Gerência do tipo de resposta	Bloco de processo que representa a primeira etapa em relação ao tratamento das respostas. Este processo pode conter a avaliação de códigos de resposta padrão utilizados entre os três módulos que compõem a arquitetura de referência.
Tratamento de resposta	Bloco de processo que representa a segunda etapa em relação ao tratamento das respostas. Este bloco de processo tem o intuito de agrupar processos refinados de processamento de resposta baseado no conteúdo de cada resposta.	

Tabela 24 - Definição dos componentes

(Conclusão)

Módulo	Nome	Descrição
--------	------	-----------

Sincronização	Base de dados estruturada	Bloco de processo que representa o controle da integridade da(s) base(s) de dados disposta(s) no módulo de coleta. Compreendendo controle transacional, mapeamento de entidades e padrões que facilitam o acesso aos dados.
	Base de dados não estruturada	Bloco de processo que representa o controle de acesso da(s) base(s) de dados disposta(s) no módulo de coleta como forma de armazenamento extra, ou armazenamento de dados específicos.

Fonte: Própria do autor

4.1.4.2.3. *Visão dados compartilhados*

Esta visão foi utilizada para entender a estrutura da AR sob a ótica de elementos de leitura e gravação de informações em vários repositórios de dados compartilhados (Apêndice I).

O diagrama apresentado para a representação da visão de processos é capaz de suprir a abstração necessária para a visão de dados compartilhados. Como já foi mencionado anteriormente a arquitetura de referência deve proporcionar diversas formas de instanciar arquiteturas concretas, além de implementações de projetos oriundos de tais arquiteturas concretas. Em virtude deste motivo é possível perceber que os módulos que compõem a arquitetura podem ser arranjados de forma a se adequar da melhor forma possível as necessidades de cada projeto, ou seja dependendo do projeto é possível ter arranjos compostos por um a vários módulos de coleta de dados, de nenhum a vários módulos de sincronização e de um a vários módulos servidores principais.

Em virtude das possibilidades de arranjo dos módulos propostos pode-se assumir quatro possibilidades de compartilhamento de dados, sendo estas:

- **Compartilhamento de dados unidirecional:** O compartilhamento de dados unidirecional refere-se ao processo de exportação de dados coletados pelo módulo de coleta de dados para o módulo servidor principal. Neste processo o módulo de coleta não necessita recuperar nenhum tipo de informação previamente sincronizada para desempenhar a coleta de dados.
- **Compartilhamento de dados bidirecional:** O compartilhamento de dados bidirecional é similar ao processo de compartilhamento de dados unidirecional. Sua diferença encontra-se no processo de sincronização de dados entre o módulo de

coleta de dados e o módulo servidor principal, este processo é realizado após o processo de exportação de dados do módulo de coleta de dados para o módulo servidor principal.

- **Compartilhamento de dados unidirecional assistido:** Este tipo de compartilhamento envolve os três módulos propostos para a arquitetura de referência, de modo que o módulo de coleta de dados envia os dados para o módulo de sincronização, que por sua vez insere os dados sem restrição de repetições, os enfileira, e os envia para o servidor principal que aplica as validações e regras de negócio necessárias.

A utilização deste compartilhamento é recomendada para arquiteturas que necessitem que o(s) módulo(s) de coleta de dados enviem um grande volume de dados de forma consecutiva, sem que seja necessária a espera de validações, regras de negócio, confirmações e possíveis tratamentos de erro por parte do servidor que irá armazenar os dados coletados. Além disto este tipo de compartilhamento de dados necessita que os módulos de coleta de dados não necessitem realizar importações de dados para prosseguir com suas tarefas.

- **Compartilhamento de dados bidirecional assistido:** Este tipo de compartilhamento de é uma junção adaptativa do compartilhamento de dados bidirecional com o compartilhamento de dados unidirecional assistido. Nesta abordagem o módulo de coleta de dados envia os dados para o módulo de sincronização, que por sua vez insere os dados sem restrição de repetições, os enfileira, e os envia para o servidor principal que aplica as validações e regras de negócio necessárias. Após um período de tempo pré-determinado o(s) módulo(s) de coleta de dados realizam uma operação de importação de dados presentes no servidor principal, o que acarreta na sincronização do(s) módulo(s) de coleta com o servidor. É importante ressaltar que durante o período de sincronização estipulado o servidor de sincronização deve ter terminados as suas atividades.

Um exemplo para a aplicação desta abordagem de compartilhamento de dados é a utilização de sistemas de registro de vendas, em que um vendedor pode exportar diversos tipos de informações para um servidor, porém ao final do dia seja necessário realizar um relatório de vendas gerais realizadas.

4.1.4.3. Ponto de vista de implantação

Neste ponto de vista foram abordados aspectos que podem existir mediante a implantação da AR. As visões presentes nesta perspectiva são as visões implantação, desenvolvimento, parte física e tecnologia.

4.1.4.3.1. Visão de implantação

Visão utilizada para descrever máquinas, softwares instalados e conexões de rede utilizadas pela arquitetura de referência. Para representar esta visão foi utilizado o diagrama de pacotes representando os módulos de coleta remota e móvel, o servidor de sincronização opcional e o servidor principal (Apêndice J).

4.1.4.3.2. Visão de desenvolvimento

Visão utilizada para descrever máquinas, softwares instalados nessas máquinas e conexões de rede usadas por instâncias da referência. Para representar esta visão foi utilizado o diagrama de componentes representando os módulos de coleta remota e móvel, o servidor de sincronização opcional e o servidor principal (Apêndice K).

4.1.4.3.3. Visão de parte física

Visão utilizada para mapear software para hardware, enfatizando requisitos não funcionais. Para representar esta visão foi utilizado o diagrama de componentes representando os módulos de coleta remota e móvel, o servidor de sincronização opcional e o servidor principal (Apêndice J).

4.1.4.3.4. Visão tecnológica

Esta visão foi utilizada para apresentar requisitos, padrões/ferramentas comumente usados em instâncias da arquitetura de referência. O objetivo desta etapa é proporcionar uma orientação em relação a seleção de tecnologias apropriadas para a arquitetura de referência.

Para a realização desta etapa optou-se por uma descrição textual, esta decisão foi tomada devido ao fato de que esta visão é um conjunto de boas práticas que podem impactar a implementação das instancias da arquitetura de referência. As boas práticas

foram subdivididas em algumas seções que representam um contexto de boas práticas. É importante ressaltar que é aconselhável a adoção de tais práticas no processo de desenvolvimento dos diferentes módulos que compõem a AR.

- **Paradigma:** Esta guia representa a escolha do paradigma de implementação correto para poder modelar e implementar os elementos de software que irão compor cada um dos módulos da arquitetura.

A escolha de um paradigma foi fundamental, pois, é reflexo da natureza e escopo do problema abordado, por exemplo, a utilização de um paradigma orientado a objetos em uma aplicação que não necessite de tais abstrações, mostra-se como uma desnecessária camada adicional de complexidade a implementação do projeto.

Sendo assim é necessário realizar uma rigorosa análise do problema proposto para compreender qual paradigma melhor se adéqua para o desenvolvimento de uma solução, por exemplo, se a natureza do problema exhibe um ambiente altamente distribuído e que necessite de um bom gerenciamento de concorrência e paralelismo, faz-se então necessária a escolha por um paradigma que proporcione conceitos relevantes para os aspectos requeridos, como por exemplo, imutabilidade, neste caso o paradigma funcional seria uma boa opção, uma vez que a imutabilidade é parte fundamental de sua filosofia.

- **Linguagem:** A escolha da linguagem adequada está relacionada a quatro fatores, como a natureza do problema, ao escopo da aplicação, o paradigma escolhido e a disponibilidade de bibliotecas. Em virtude destas quatro características, a análise cuidadosa de que linguagem adotar para realizar a codificação dos módulos da arquitetura é fundamental. É importante ressaltar que existem outros fatores que podem influenciar na escolha de uma linguagem, tais como a existência de boa documentação da linguagem e suas bibliotecas, a maturidade da linguagem, a existência de *frameworks* para o auxílio da construção de diferentes soluções, porém estes não são tão críticos quando colocados em perspectiva juntamente aos quatro fatores iniciais.
- **Organização:** A escolha de uma estrutura que organize o código gerado, desta forma a compreensão do projeto será aprimorada, sendo isto determinante para a manutenção e criação de novas funcionalidades. Alguns *frameworks* proporcionam

uma estruturação do código desde a criação de um projeto até sua manutenção e o desenvolvimento de novas *features*.

- **Padrões:** A utilização de padrões é fundamental para a arquitetura de referência que foi desenvolvida em específico a utilização do padrão *observer* é fundamental para a comunicação e execução de tarefas assíncronas nos módulos de coleta.
- **Arquitetura:** A utilização de estilos arquiteturais também foi fundamental para o desenvolvimento da arquitetura proposta. O estilo arquitetural adotado para a arquitetura foi arquitetura cliente-servidor. É importante ressaltar que a utilização de estilos arquiteturais também pode ser aplicada em cada módulo da arquitetura, por exemplo, o módulo de coleta móvel e o módulo de serviços foram desenvolvidos utilizando a arquitetura em três camadas.

4.1.4.4. Ponto de vista de código fonte

Neste ponto de vista foram abordadas as diferentes abstrações referentes ao contexto de implementação da AR. As visões presentes neste ponto de vista são as visões lógica, módulo, parte física e estrutural.

4.1.4.4.1. Visão lógica

Descreve a decomposição dos módulos da arquitetura de referência e apresenta os requisitos funcionais do sistema. Foi utilizada a representação de diagrama de bloco interno para representar esta visão (Apêndice I).

4.1.4.4.2. Visão de módulo

Apresenta os principais módulos funcionais, o fluxo de dados entre esses módulos e as principais interfaces. Foi utilizada a representação de diagrama de bloco interno para representar esta visão (Apêndice I).

4.1.4.4.3. Visão estrutural

Separa a perspectiva de implementação da perspectiva de aplicativo que será desenvolvido exibindo a interação com o ambiente. Foi utilizada a representação de diagrama de bloco interno para representar esta visão (Apêndice I).

4.1.5. Passo RA4 - avaliação arquitetural

Após a conclusão do passo RA3 - Projeto arquitetural, a arquitetura de referência Ostara foi definida, desta forma proporcionando a realização da última etapa do processo ProSA-RA, o passo RA4 - Avaliação arquitetural (Apêndice L).

4.1.5.1. Avaliação de protótipo arquitetural

Foi proposto verificar a eficácia da arquitetura de referência Ostara através do processo de análise de protótipo arquitetural. Optou-se pela utilização desta metodologia para a realização do processo de validação, pois Avgeriou (2003) afirma que, é possível instanciar e utilizar partes da arquitetura de referência para a promoção de avaliações.

Corroborando com o exposto, para realizar a avaliação da porção arquitetural instanciada foi escolhida a utilização do método de *UserStudies*. Este método foi escolhido, pois segundo Nielsen (1994) *UserStudies* permitem gerar ponderações referentes a como ou por qual razão o objeto de estudo é efetivo, seja este uma técnica ou uma aplicação. Este método pode até mesmo ser utilizada para o aprimoramento de tais objetos de estudo.

Desta forma, para realizar o processo de avaliação da arquitetura de referência proposta, primeiramente foi necessário definir quais elementos pertencentes à arquitetura de referência Ostara deveriam ser instanciados, pois estes componentes contemplaram as principais funcionalidades de domínio descritas no passo RA2 - Estabelecimento dos requisitos arquiteturais.

Posteriormente foi necessário verificar a possibilidade de aplicação prática para a instância arquitetural que viria a ser avaliada. Realizou-se então uma parceria com a Agência Eletrobrás Eletronorte, esta parceria surge devido às necessidades pelo time mantenedor do projeto “Banco de Germoplasma”.

4.1.5.1.1. Detalhamento do problema

Criado e ainda mantido pela empresa Agencia Eletrobras Eletronorte o projeto “Banco de Germoplasma” coleta, reúne e preserva dados de espécies vegetais e animais que são preservadas pela organização, com o objetivo de criar uma base de dados que

possa auxiliar a extração, análise e uso de informações sobre os ambientes mantidos pelo projeto.

Para que a organização possa cumprir com êxito o objetivo proposto pelo projeto, faz-se imprescindível que a coleta dos dados necessários para a realização do projeto resulte em uma amálgama de dados mais transparentes possíveis e fidedignos a um ambiente naturalmente preservado. Tendo em vista a obrigatoriedade de tais resultados para o cumprimento do projeto a Agência Eletrobras Eletronorte mantém nodos de preservação ambiental, estes ambientes, conhecidos internamente pela organização como bases de coleta, podem ser classificados como espaços controlados de pesquisa que propiciam condições ideais para o crescimento, desenvolvimento e preservação da biodiversidade vegetal e animal local. Contendo espécies selecionadas para os fins descritos acima as bases de preservação e coleta de informações da biodiversidade local são a Ilha de Germoplasma Eletronorte UHE Tucuruí, a base de coleta 03 e a base de coleta 04.

O projeto também conta com uma operação mantida por diversos setores organizacionais internos, que em função do escopo do projeto tem uma escala, complexidade e custo elevados. A operação do projeto tem o objetivo de realizar a coleta dos dados das espécies preservadas nas bases de coleta.

O processo operacional necessário para auxiliar a realização do objetivo do projeto inicia-se com a visita de uma equipe responsável pela categorização e coleta de dados aos locais de preservação. É importante ressaltar que devido ao escopo da operação, este trabalho manteve seu foco no processo operacional de uma das três bases de coleta, mais especificamente a Ilha de Germoplasma Eletronorte UHE Tucuruí, além de resumir-se ao escopo de espécies vegetais mantidas pelo projeto, sendo assim os processos descritos abaixo foram mapeados através das práticas desenvolvidas pela equipe responsável na base em questão.

A incursão ao nodo de preservação inicia-se com a ida da equipe responsável à base de coleta, em seguida tem-se o início do processo com a execução de sua primeira etapa, que se constitui da definição das espécies presentes, da quantidade de espécimes de cada espécie e do mapeamento de cada um destes elementos na ilha. Este levantamento proporciona uma fundamentação inicial de dados, crucial para a tomada

de decisão da próxima etapa do processo, que será a seleção de matrizes. Este levantamento inicial é muito importante devido a forma como a base de coleta foi dividida e organizada, a Ilha de Germoplasma Eletronorte UHE Tucuruí, foi subdividida em quadrantes, estes podem ser chamados de quadras e/ou parcelas. Obter a informação inicial de onde cada espécime se encontra dentro do mapeamento da ilha, confere mais velocidade, assertividade e produtividade para equipe responsável.

Após a execução da primeira etapa a equipe especializada deve eleger uma matriz correspondente para cada espécie vegetal que é mantida na base de coleta pelo projeto. Dentro no que tange o contexto citado acima podemos caracterizar uma matriz como um espécime vegetal mantido na base de coleta, que representará uma espécie vegetal encontrada na mesma base. Esta etapa do processo é realizada pois os dados serão mantidos pelo projeto, serão obtidos através da análise fenológica, ou fenologia, destas matrizes.

O processo então segue para a etapa seguinte, cujo objetivo é realizar a análise fenológica das matrizes eleitas nas bases de coleta. A equipe responsável realiza a coleta sistemática de informações, realizando a fenologia das matrizes e anotando os resultados em um caderno de papel que serve como base de dados local. É importante ressaltar que existem dois tipos de processos de coleta de dados fenológicos, são estes a coleta anual e a coleta quinzenal, estes dois tipos de coleta tem semelhanças, afinal coletam dados sobre as mudanças de um espécime devido estímulos do ambiente em que ele se encontra, porém quando mapeados para objetos possuem um número e tipos de atributos diferentes.

Em seguida a equipe responsável retorna para a base organizacional da empresa com as informações coletadas, e devidamente descritas na base de dados utilizada em campo. A base de dados de campo então é entregue a um colaborador da organização, para que este realiza o processo de persistência dos dados obtidos em campo em uma base de dados digital, garantindo desta forma a perenidade desses dados, para utilização futura, de modo a contribuir com o cenário de negócio da empresa.

Ao realizar uma análise da operação descrita acima, é possível detectar certos aspectos que não são tão eficientes dentro do contexto de negócio da empresa, são estes:

- Coleta de dados manual *In Loco* utilizando lápis e papel;
- Tempo elevado de organização de dados coletados *In Loco*;
- Persistência manual do montante de dados coletado em campo no banco de dados permanente;
- Tempo elevado de operação;
- Custo elevado na operação.

Os pontos citados acima são características inerentes do processo de coleta utilizado na operação montada pela empresa, e apesar de que estes não impossibilitem a realização da operação, é notório que se pode utilizar elementos e técnicas de tecnologia de informação de maneira a auxiliar as etapas do processo da operação de coleta de dados.

Deve-se ressaltar que a operação proposta consegue ser desenvolvida sem o auxílio de recursos de tecnologia de informação, porém ao optar pela não utilização de tais recursos, a operação da organização terá um baixo desempenho em relação ao tempo de coleta de dados *In Loco*.

É necessário evidenciar que também será muito presente o risco de perda de dados colhidos na base de coleta, uma vez que estes estarão a mercê de quaisquer tipos de acidentes e adversidades, devido sua forma de persistência em papel, o que poderá causar retrabalho, aumento no custo e tempo de coleta.

A estratégia de persistência dos dados de forma manual na base de dados principal também é um problema muito grande para a operação, pois conta com o fator humano para sua realização de tal tarefa, e assim como a persistência de dados em papel, pode resultar em perda de informação. Pode-se destacar também que o tempo de inserção das informações coletadas *In Loco* é maior, devido à necessidade da revisão do que foi obtido na base de coleta, da ordenação dessas informações e do cadastro singular e enfileirado de cada um dos registros de coleta. Esta estratégia também corrobora para a demasiada demora na disponibilização das informações coletadas para os administradores e gerentes da operação, além de não reunir os dados e metadados do projeto em uma base integrada de informação para utilização dos mesmos em benefício do negócio da organização.

4.1.5.1.2. Instância arquitetural

A instância arquitetural proposta foi construída com o intuito de suprir as necessidades de coleta de dados, voltada para a realização de estudos fenológicos, vivida pelos responsáveis do projeto “Banco de Germoplasma”, mantido na Agência Eletrobras Eletronorte.

Devido à possibilidade de instanciar parcialmente a arquitetura de referência Ostara para a realização dos *UserStudies*, foi instanciada uma arquitetura de software composta pelo módulo de coleta e o módulo de serviços, sendo que nem todos os componentes e módulos internos da arquitetura Ostara foram utilizados.

A proposta para cada um destes módulos foi que as implementações dos mesmos fossem respectivamente, um aplicativo capaz de coletar dados *off-line* e uma API Restful capaz de receber e prover os dados coletados. A definição da instância da arquitetura de referência Ostara foi realizada utilizando-se de duas ferramentas pertencentes à UML, o diagrama de pacotes e o diagrama de componentes (Apêndice M; Apêndice N).

4.1.5.1.3. Implementação

Baseada nas necessidades operacionais apresentadas pelo projeto da Agência Eletrobras Eletronorte, foi desenvolvido um sistema informatizado de coleta de dados baseado na instância arquitetural proposta, tal processo de implementação totalizou um período de quatorze meses de desenvolvimento e refinamento.

Devido ao escopo definido para este trabalho, fez-se imprescindível a construção de um produto de software que tivesse a capacidade de ser manutenível e que possa evoluir, no sentido de que após o término deste trabalho, caso seja necessário, novas funcionalidades podem ser desenvolvidas para o sistema. Em virtude disto, ao planejar o desenvolvimento do sistema, optou-se pelo desenvolvimento de componentes modulares em API (*Application Programming Interface*).

O desenvolvimento baseado em API permite o desacoplamento do *back-end* e do *front-end* da aplicação, conferindo flexibilidade em relação as escolhas de implementação de tais componentes.

O sistema que foi desenvolvido é composto de três partes principais, sendo estas (i) uma API (*Application Programming Interface*) *restful* que será responsável pela

persistência e disponibilidade das dos dados, (ii) um aplicativo *móvel* que servirá como mecanismo de coleta e exportação de dados em campo, (iii) uma interface web, que terá o objetivo de prover um ambiente administrativo de interação com as informações das bases de coleta.

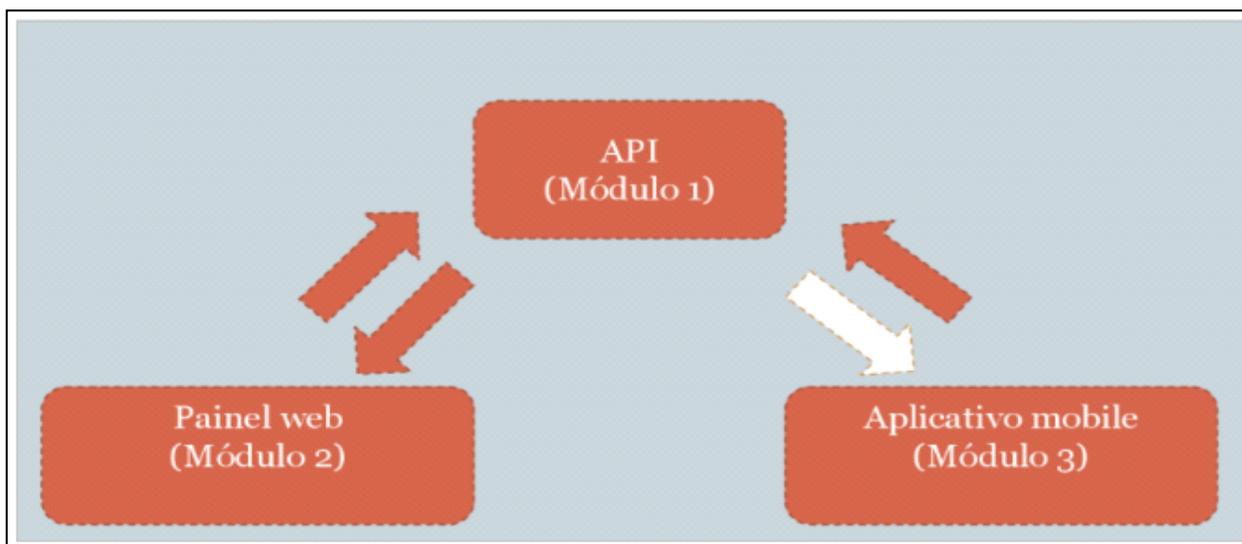


Figura 27 - Organização dos módulos que compõem o sistema e as suas relações.

Fonte: Própria do autor.

4.1.5.1.4. *Mecanismo avaliativo*

Para realizar a análise de protótipo arquitetural juntamente com a metodologia de *User Studies*, foi desenvolvido um questionário avaliativo. No desenvolvimento das perguntas presentes no questionário foram levados em consideração aspectos que pudessem refletir qualidades de tempo de execução, como performance, segurança, funcionalidade e usabilidade. É importante ressaltar que o desenvolvimento das perguntas também foi guiado pelo contexto ao qual a implementação da instância arquitetural pertencia, mais especificamente, a coleta de dados móveis voltada para estudos fenológicos de espécies da flora nativa da região de Tucuruí PA (Apêndice O).

Além disto, para a composição avaliativa dos *User Studies*, não se levou em consideração o Módulo 2, o qual detinha a função de ser um painel web administrativo. Sendo assim a análise considerou a utilização e funcionamento da API juntamente com o aplicativo, uma vez que estes dois módulos representam requisitos chave para o

problema proposto, e os requisitos comuns a sistemas pertencentes ao domínio de coleta de dados.

4.1.5.1.5. Participantes

O grupo de participantes do processo de *User Studies* foi constituído por especialistas de domínio, os quais são colaboradores da Agencia Eletrobras Eletronorte que participam do projeto “Banco de Germoplasma”. O processo foi conduzido constando com seis participantes, os quais são de fundamental importância, uma vez que trabalham diretamente com modelo de domínio de coleta de dados.

4.1.5.1.6. Ambiente

Os testes de conceito foram realizados nas instalações da Agencia Eletrobras Eletronorte localizada em Tucuruí – PA. O ambiente contava com duas salas, uma sala para a realização da exibição do funcionamento da ferramenta e realização dos testes, e a outra para a espera dos participantes. A sala de realização dos testes estava equipada com um equipamento projetor, o que proporcionou a demonstração do funcionamento da aplicação.

Foi disponibilizado um *smartphone* aparelhado com o sistema operacional Android. Este dispositivo tinha como função ser utilizado como instrumento de contato entre os colaboradores e o módulo de coleta de dados que foi desenvolvido para a organização.

A instância do módulo principal de serviços foi disponibilizada na plataforma de serviços remota Heroku. A instância do módulo de serviços principal é representada pela API que foi desenvolvida.

Também foi utilizado um laptop equipado com o sistema Ubuntu 18.04, este recurso foi utilizado como forma de contingência, caso os dispositivos móveis utilizados no teste, ou a conexão remota com o módulo servidor principal apresentasse problemas. Esta contingência era possível pois o equipamento utilizado poderia simular o módulo servidor principal e o módulo de coleta de dados móveis.

4.1.5.1.7. Procedimento

Para que o processo de aplicação do estudo pude-se ser mais controlado em termos de variabilidade dentro do estudo, optou-se por utilizar uma estrutura pré definida para a realização do processo. Esta estrutura foi composta de duas etapas, sendo respectivamente uma exibição da ferramenta e a segunda a realização dos testes.

A primeira etapa resumiu-se a uma explicação referente a como a ferramenta mapeou as necessidades apresentadas pela organização, e como as operações necessárias para a realização das tarefas organizacionais podiam ser realizadas. Este processo foi voltado para os colaboradores participantes do projeto, conferindo-lhes o conhecimento necessário para operar a solução proposta na etapa de testes.

A segunda etapa realizada foi composta pela realização dos testes, estes foram compostos por um número de quatorze passos, tal estrutura está descrita a seguir.

Tabela 25 - Descrição dos passos constituintes da etapa de testes

(Continua)

Passo	Descrição
nº 01	É entregue a cada colaborador um formulário de consentimento, juntamente com um formulário de informações sobre o estudo.
nº 02	É pedido para que os colaboradores preencham o formulário de consentimento.

Tabela 25 - Descrição dos passos constituintes da etapa de testes

(Conclusão)

Passo	Descrição
nº 03	Os colaboradores que participam dos testes ficam em uma sala de espera.
nº 04	O mediador dos testes chama os colaboradores separadamente para a execução individual do uso da ferramenta.
nº 05	O colaborador testa o processo de <i>login</i> e <i>logout</i> .
nº 06	O colaborador testa o processo de manutenção de matrizes, composto pelos processos de criação, substituição, exibição, edição, deleção.
nº 07	O colaborador testa o processo de manutenção de fenologias quinzenais, composto pelos processos de criação, exibição, edição e deleção.
nº 08	O colaborador testa o processo de manutenção de fenologias anuais, composto pelos processos de criação, exibição, edição e deleção.
nº 09	O colaborador testa o processo de sincronização dos dados coletados através do módulo de coleta móvel.
nº 10	O mediador dos testes pede para que o colaborador que terminou o teste de aplicação se retire da sala e aguarde na sala de espera.

nº 11	O mediador dos testes verifica se todos os colaboradores já realizaram o teste com a ferramenta, ou se por acaso, ainda existe um colaborador que não tenha realizado o teste.
nº 12	Após a execução de todos os testes todos os colaboradores que estão na sala de espera são instruídos a retornarem para a sala de testes simultaneamente.
nº 13	É pedido para que os colaboradores preencham o formulário de referente ao estudo. O mediador pode fornecer esclarecimentos ao questionário.
nº 14	Uma vez que os colaboradores tenham terminado de preencher os formulários e não tenham mais perguntas eles podem se retirar da sala de testes.

Fonte: Própria do autor

5. RESULTADOS

5.1. A ARQUITETURA PROPOSTA

A arquitetura de referência desenvolvida neste trabalho foi nomeada como Ostara, tal nome é uma modificação da palavra Eostre, advinda da mitologia anglo-saxã. Este nome é utilizado para referenciar a deusa do renascimento, fertilidade e amor, a qual teve seu significado correlacionado com o primeiro dia de primavera, tempo que era celebrado devido a possibilidade do início de um novo ciclo de colheita (CUSACK, 2007). De forma semelhante à arquitetura de referência proposta neste trabalho busca-se auxiliar o desenvolvimento arquiteturas concretas e implementações que permitam a realização da tarefa de coletar dados.

Após a realização dos três primeiros passos do processo ProSA-RA foram desenvolvidas estruturas necessárias para atender aos requisitos arquiteturais que foram identificados durante a execução do processo. Estas estruturas necessitavam ser organizadas de acordo com um modelo coeso, que representasse as diferentes etapas do processo de coleta de dados. Tomando como referência o modelo de pipeline proposto por Sant'anna (2014), as estruturas foram divididas em módulos que representassem de forma adequada o fluxo de dados necessário para sistemas de coleta.

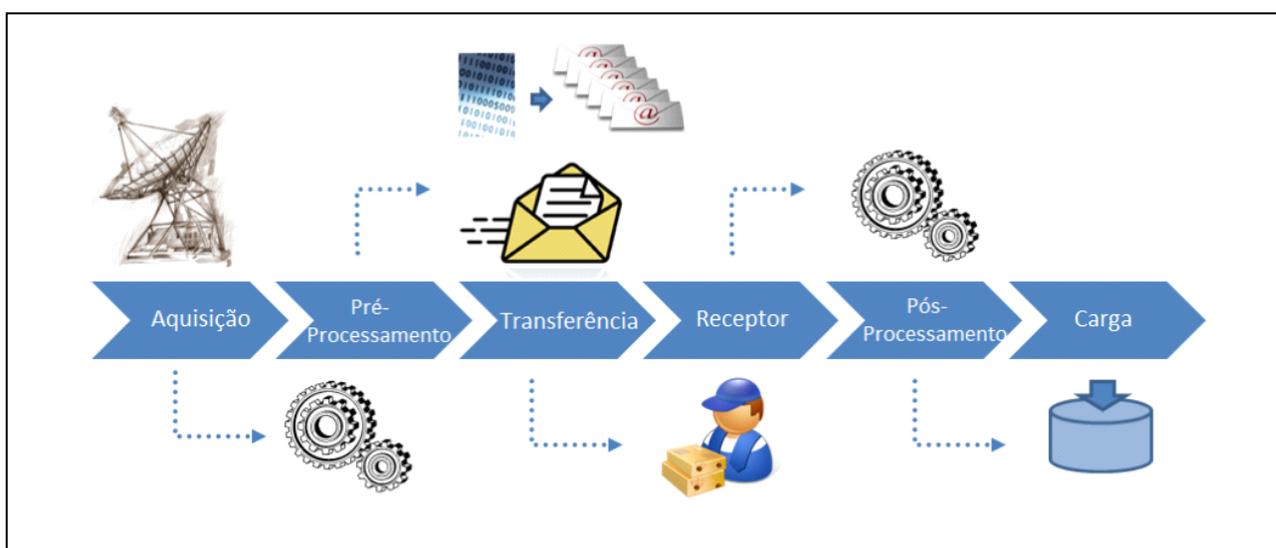


Figura 28 - Estrutura de pipeline de carga de dados. Fonte: Sant'anna, 2014.

De acordo com os estudos realizados a respeito de sistemas especializados em coletas de dados, foi possível notar a divisão modular do fluxo de operações necessárias

para a realização de tal processo. Desta forma, a organização do fluxo de dados sob o ponto de vista de abstração genérico e modular dispõe-se segundo a Figura 29.

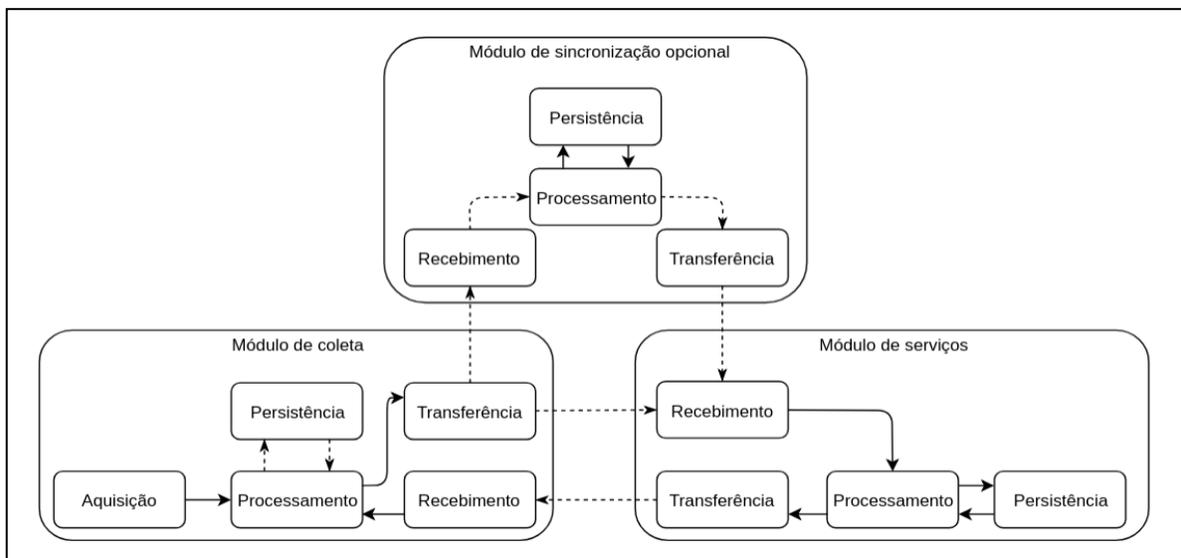


Figura 29 - Estrutura de fluxo de dados da arquitetura de referência Ostara. Fonte: Própria do autor.

A disposição de fluxo de dados proposta acima foi subdividida em módulos, esta estratégia de representação foi escolhida devido ao fato desta ser capaz de demonstrar de forma organizada a estrutura da arquitetura de referência. Esta representação busca demonstrar as possibilidades de fluxo de informações para diferentes tipos de sistemas de coleta, subdividindo a estrutura da AR em módulos, conferindo versatilidade e adaptabilidade a mesma, de modo a permitir o processo de instância de arquiteturas concretas que melhor atendam os escopos de projeto ao qual permeiam.

Uma vez definida a estrutura de fluxo de dados genérica e modular, pode-se organizar as estruturas internas de cada módulo, responsáveis pela execução dos processos de aquisição, processamento, persistência, transferência e recebimento de dados em cada módulo que compõem a arquitetura. As estruturas internas que compõem os três módulos principais da arquitetura de referência Ostara são chamadas de componentes e módulos internos que são respectivamente, e uma unidade especializada na realização de uma tarefa fundamental e uma aglomeração de componentes que executem funções pertencentes ao mesmo contexto.

Faz-se importante ressaltar, que a organização arquitetural dos módulos principais representados acima, ou seja, o módulo de coleta de dados, o módulo de serviços e o

módulo de sincronização opcional, foi elaborada de acordo a arquitetura cliente-servidor. Enquanto que a organização arquitetural interna destes módulos principais foi desenhada de acordo com o estilo de arquitetura em camadas.

Utilizando-se da organização baseada em componentes e módulos, foi possível subdividir e organizar exatamente os elementos necessários para a execução das tarefas requeridas durante o processo de coleta de dados, compondo a estrutura da arquitetura de referência Ostara (Apêndice L). A definição dos módulos principais que compõem a arquitetura de referência Ostara são descritos abaixo.

5.1.1. Módulo de coleta

Para promover a organização interna dos principais módulos arquiteturais foi proposta a divisão e estruturação de cada um dos elementos exibidos acima. A primeira porção a ser elaborada foi a porção de coleta de dados, pois esta parte da arquitetura detinha grande complexidade, devido a necessidade de englobar e representar os requisitos arquiteturais dos mais diferentes tipos de sistemas de coletas de dados e de coleta de dados móveis. A imagem abaixo exhibe a organização da porção de coleta de dados.

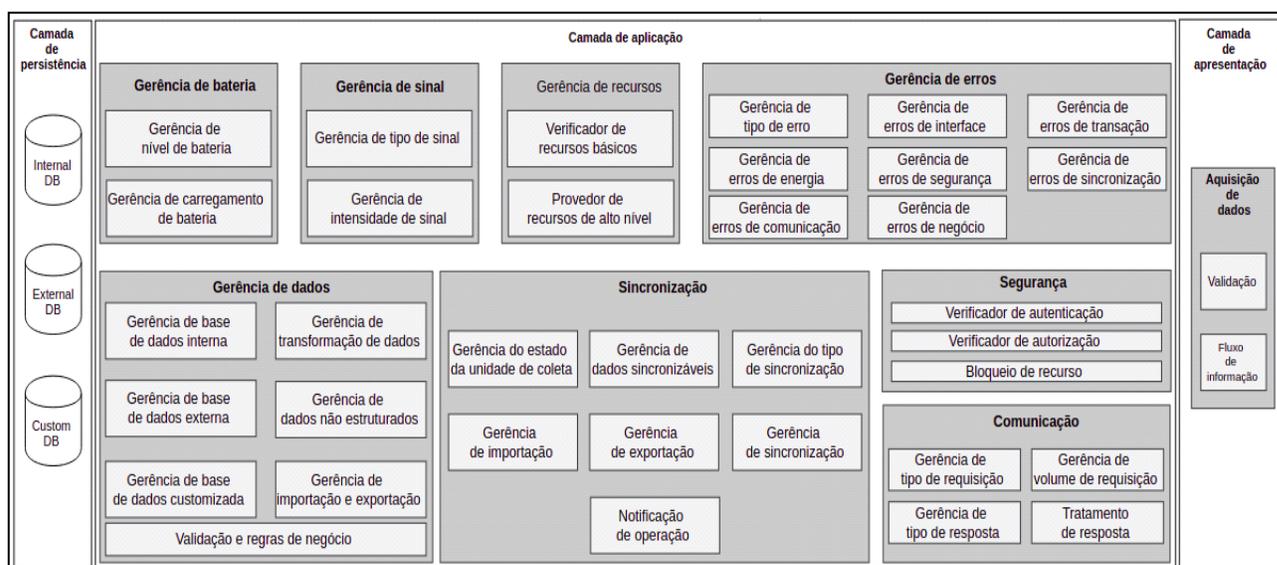


Figura 30: Porção de coleta de dados. Fonte: Própria do autor.

5.1.1.1. Camadas

A organização em três camadas foi escolhida para prover uma estruturação coesa ao elemento em questão, dispondo os seus módulos e componentes de forma estruturada. As camadas pertencentes a este módulo são:

- **Persistência:** Camada responsável por armazenar os dados coletados em capo. Os elementos presentes nesta camada são as possíveis bases de dados e gerenciadores utilizados em sistemas de coleta de dados.
- **Aplicação:** Camada responsável pelo pré-processamento de dados interno e operações de sincronização da aplicação. Os elementos pertencentes a esta camada são os módulos de Gerência de bateria, Gerência de sinal, Gerência de recursos, Gerência de erros, Gerência de dados, Comunicação, Sincronização e Segurança.
- **Apresentação:** Camada responsável pela coleta e exibição de dados. O elemento presente nesta camada é o módulo de Aquisição de dados.

5.1.1.2. Módulos internos

A organização modular das funcionalidades da desta porção da arquitetura foi escolhida para prover uma divisão coesa e bem definida. Os módulos que compõem esta porção da arquitetura são:

- **Aquisição de dados:** Módulo responsável por cuidar da entrada e/ou saída de dados com o usuário final, ou com o ambiente de coleta. Os componentes presentes neste módulo são os componentes (1) Validação, (2) Fluxo de informação.
- **Gerência de bateria:** Módulo transversal responsável pela observação do nível de energia presente na bateria do dispositivo de coleta. Dependendo da implementação da arquitetura este módulo também pode prover mecanismos de gerenciamento do processo de recarga da bateria do dispositivo de coleta. Os componentes pertencentes a este módulo são os componentes (1) Gerência de nível de bateria e (2) Gerência de carregamento de bateria.
- **Gerência de sinal:** Módulo transversal responsável pela observação do nível de sinal de comunicação sem fio presente na bateria do dispositivo de coleta. Os componentes

presentes neste módulo são os componentes (1) Gerência de tipo de sinal (2) Gerência de intensidade de sinal.

- **Gerência de recursos:** Módulo transversal responsável pela disponibilização e restrição dos recursos de coleta e sincronização de dados, funcionando em conjunto com os módulos de Gerência de bateria e Gerência de sinal. Este módulo tem o intuito de evitar erros oriundos da utilização de recursos de alto nível durante a indisponibilidade de recursos de baixo nível, fundamentais para o funcionamento do dispositivo de coleta. Pode-se citar como exemplo a restrição de funções de coleta ou de sincronização durante períodos de baixo nível de energia da bateria do dispositivo de coleta, ou períodos em que o sinal sem fio apresenta má qualidade. Os componentes presentes neste módulo são os componentes (1) Verificador de recursos básicos e (2) Provedor de recursos de alto nível.
- **Gerência de erros:** Módulo transversal responsável pelo tratamento e disponibilização de comunicação de erros da aplicação. Os componentes presentes neste módulo são os componentes (1) Gerência de tipo de erro, (2) Gerência de erros de interface, (3) Gerência de erros de negócio, (4) Gerência de erros de transação, (5) Gerência de erros de comunicação, (6) Gerência de erros de sincronização, (7) Gerência de erros de segurança, (8) Gerência de erros de energia.
- **Gerência de dados:** Módulo que atua como forma de acesso aos dados contidos na base de dados. Os componentes presentes neste módulo são os componentes (1) Gerência de base de dados interna, (2) Gerência de base de dados externa, (3) Gerência de base de dados customizada, (4) Gerência de transformação de dados, (5) Gerência de dados não estruturados, (6) Gerência de importação e exportação, (7) Validação e Regras de negócio.
- **Comunicação:** Módulo que é responsável por gerenciar a comunicação da porção de coleta de dados com os diferentes servidores necessários para o funcionamento de todo o sistema de coleta de dados. Os componentes presentes neste módulo são os componentes (1) Gerência de tipo de requisição, (2) Gerência de volume de requisição, (3) Gerência de tipo de resposta, (4) Tratamento de resposta.
- **Sincronização:** Módulo que é responsável por gerenciar a importação, exportação e sincronização dos dados coletados para o servidor de sincronização ou o servidor

principal. Os componentes presentes neste módulo são os componentes (1) Gerência do estado da unidade de coleta, (2) Gerência de dados sincronizáveis, (3) Gerência do tipo de sincronização, (4) Gerência de importação, (5) Gerência de exportação, (6) Gerência de sincronização, (7) Notificação de operação.

- **Segurança:** Módulo responsável por cuidar da proteção de serviços disponibilizados pela porção de coleta de dados. Os componentes presentes neste módulo são os componentes (1) Verificador de autenticação, (2) Verificador de autorização, (3) Bloqueio de recurso.

5.1.1.3. Componentes Internos

- **Validação:** Componente responsável pela validação da entrada de dados da porção de coleta de dados.
- **Fluxo de informação:** Componente responsável pela decisão de qual módulo irá ser invocado para processar os dados coletados.
- **Gerência de nível de bateria:** Componente responsável por verificar e informar o nível da bateria.
- **Gerência de carregamento de bateria:** Componente responsável ativar e desativar o carregamento do dispositivo de coleta de dados. Este processo é baseado na verificação do nível de energia realizada pelo componente de Gerência de nível de bateria.
- **Gerência de tipo de sinal:** Componente responsável por verificar e informar o tipo de sinal que está sendo utilizado pela porção de coleta de dados.
- **Gerência de intensidade de sinal:** Componente responsável por verificar a intensidade e qualidade do sinal que está sendo utilizado pela porção de coleta de dados.
- **Verificador de recursos básicos:** Componente responsável por verificar a qualidade e disponibilidade e estabilidade dos recursos básicos necessários para o funcionamento da porção de coleta de dados. Objetivamente este componente utiliza os componentes de Gerência de nível de bateria, Gerência de tipo de sinal, Gerência de intensidade de sinal, para averiguar a possibilidade da execução de recursos de alto nível;

- **Provedor de recursos de alto nível:** Componente responsável pela efetiva disponibilização de recursos de alto nível como recursos de coleta, importação, exportação e sincronização.
- **Gerência de tipo de erro:** Componente responsável pela definição do tipo de erro ocorrido baseado na operação que estava ocorrendo.
- **Gerência de erros de interface:** Componente responsável pela exibição de mensagens de erros ocorridos na interface de captura de dados.
- **Gerência de erros de negócio:** Componente responsável pela exibição de mensagens de erros ocorridos devido a regras de negócio.
- **Gerência de erros de transação:** Componente responsável pela exibição de mensagens de erros ocorridos devido a problemas de transação.
- **Gerência de erros de comunicação:** Componente responsável pelo tratamento e exibição de mensagens de erros ocorridos devido a problemas de transação.
- **Gerência de erros de sincronização:** Componente responsável pelo tratamento e exibição de mensagens de erros ocorridos devido a problemas de importação, exportação e sincronização de informações.
- **Gerência de erros de segurança:** Componente responsável pelo tratamento e exibição de mensagens de erros ocorridos devido a problemas de autenticação e autorização.
- **Gerência de erros de energia:** Componente responsável pelo tratamento e exibição de mensagens de erros ocorridos devido a problemas níveis baixos de energia da bateria.
- **Gerência de base de dados interna:** Componente responsável pela manutenção dos dados em relação a base de dados interna do dispositivo de coleta.
- **Gerência de base de dados externa:** Componente responsável pela manutenção dos dados em relação a base de dados externa do dispositivo de coleta.
- **Gerência de base de dados customizada:** Componente responsável pela manutenção dos dados em relação à base de dados customizada. Pode-se utilizar como exemplo a utilização de uma estrutura customizada de diretórios, arquivos e organização de dados interna dos arquivos.

- **Gerência de transformação de dados:** Componente responsável pelo processo de transformação dos dados para serem armazenados na base de dados do dispositivo de coleta e transmissão para os servidores externos.
- **Gerência de dados não estruturados:** Componente responsável pela manutenção dos dados que não são mantidos em bases estruturadas, como por exemplo, imagens, arquivos de áudio, vídeo, etc.
- **Gerência de importação e exportação:** Componente responsável pela importação e exportação do banco baseada em arquivos salvos na base de dados externa dos dispositivos de coleta.
- **Validação e Regras de negócio:** Componente responsável pelas aplicações das validações de *backend* e das aplicações das regras de negócio.
- **Gerência de tipo de requisição:** Componente responsável pela definição do tipo de comunicação que será realizada, podendo englobar diferentes tecnologias de comunicação como, por exemplo, a comunicação através de arquivos XML, JSON, etc.
- **Gerência de volume de requisição:** Componente responsável pela definição do volume do pacote que será enviado na requisição, corroborando com a capacidade de exportação múltipla de dados. Esta definição baseia-se na disponibilidade, qualidade e estabilidade da conexão sem fio utilizada pelo dispositivo de coleta, sendo assim este componente está diretamente ligado com os componentes Gerência de tipo de sinal e Gerência de intensidade de sinal.
- **Gerência de tipo de resposta:** Componente responsável pela correta interpretação dos diferentes tipos de resposta enviados pela porção de sincronização ou pelo servidor principal. Este componente também é responsável pela ativação do componente de tratamento de resposta baseado no tipo de resposta processada.
- **Tratamento de resposta:** Componente responsável pelo tratamento da resposta recebida pelo servidor. Este componente comumente trabalha em conjunto com processos de importação exportação e sincronização.
- **Gerência do estado da unidade de coleta:** Componente responsável pela verificação do status de coleta da porção de coleta, ou seja, se a unidade necessita realizar

importação, exportação ou sincronização com o servidor para que as coletas de dados possam ser desempenhadas.

- **Gerência de dados sincronizáveis:** Componente responsável pela recuperação dos dados que podem ser exportados para o servidor.
- **Gerência do tipo de sincronização:** Componente responsável pela definição do tipo de sincronização que será realizada.
- **Gerência de importação:** Componente responsável pela execução do processo de importação de dados do servidor.
- **Gerência de exportação:** Componente responsável pela execução do processo de exportação de dados do servidor.
- **Gerência de sincronização:** Componente responsável pela execução do processo de sincronização de dados da base de dados do servidor com a base de dados do dispositivo de coleta.
- **Notificação de operação:** Componente responsável pela notificação de finalização da execução de processos assíncronos, como por exemplo, requisições para o servidor de sincronização ou o servidor principal.
- **Verificador de autenticação:** Componente responsável pelo processo de comunicação com o componente de Gerência de base de dados interna para a realização da autenticação do usuário.
- **Verificador de autorização:** Componente responsável pelo processo de comunicação com o componente de Gerência de base de dados interna para a realização da autorização do usuário.
- **Bloqueio de recurso:** Permissão ou negação de acesso de determinadas funcionalidades da porção de coleta de dados.

5.1.2. Módulo de serviços

A segunda porção a ser elaborada foi a porção de serviços, pois esta parte da arquitetura concentrava a responsabilidade de receber os dados coletados, processá-los, validá-los, persisti-los e disponibilizá-los. Devido isto era necessária a realização cuidadosa do design, composto de elementos que podem ser instanciados e

implementados em diversos cenários de projeto. A imagem abaixo exibe a organização da porção de coleta de dados.

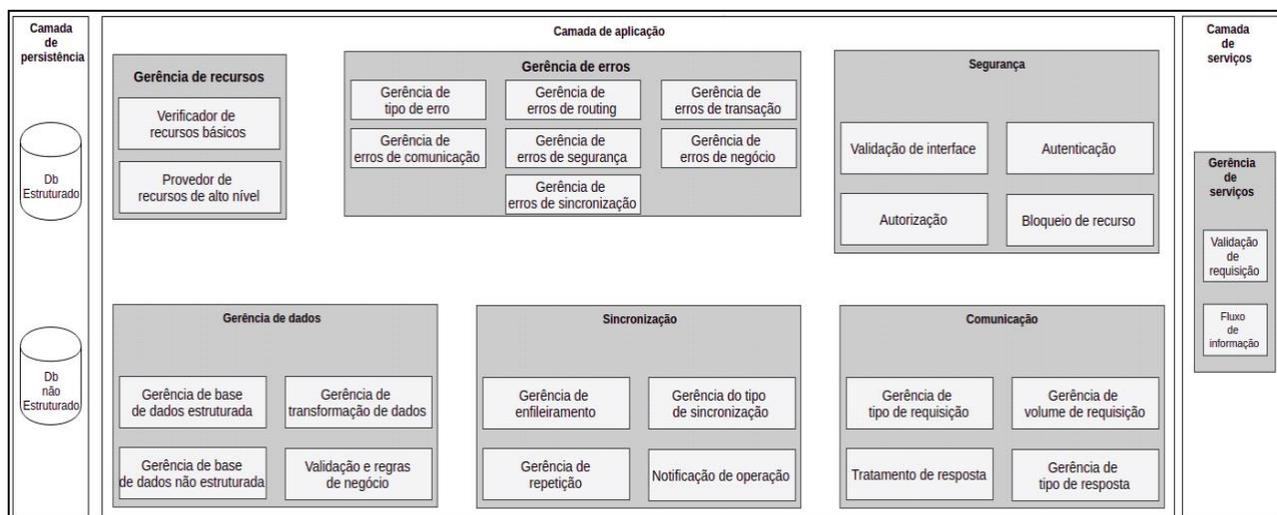


Figura 31: Porção de serviços. Fonte: Própria do autor.

5.1.2.1. Camadas

A organização em três camadas também foi escolhida como estrutura de organização dos diferentes componentes e módulos que compõe a arquitetura. As camadas pertencentes a este módulo são:

- **Persistência:** Camada responsável por armazenar os dados que foram transmitidos pelo módulo de coleta ou pelo módulo de sincronização opcional. Os elementos presentes nesta camada são as possíveis bases de dados estruturadas e não estruturadas.
- **Aplicação:** Camada responsável pelo pós-processamento de dados interno e operações de sincronização da aplicação. Os elementos pertencentes a esta camada são os módulos de Gerência de recursos, Gerência de erros, Gerência de dados, Segurança, Sincronização e Comunicação.
- **Serviços:** Camada responsável pela pelo processo de garantia de fornecimento dos serviços corretos. O elemento presente nesta camada é o módulo de Gerência de serviços.

5.1.2.2. Módulos internos

- **Gerência de serviços:** Módulo responsável pela exposição dos serviços disponibilizados pelo módulo de serviço, além da organização do fluxo de informação para a realização dos serviços. Este módulo contém os componentes de (1)Validação de requisição de recurso e (2)Fluxo de informação.
- **Gerência de recursos:** Módulo responsável pela verificação da disponibilidade e estabilidade dos recursos básicos disponibilizados pelo servidor, além da capacidade de manutenção da disponibilidade, redirecionando a requisição mediante a instabilidades. Este módulo é composto pelos componentes (1)Verificador de recursos básicos e (2)Provedor de recursos de alto nível.
- **Gerência de erros:** Módulo responsável pelo tratamento dos erros ocorridos, além disso este módulo também provê as respostas para os erros que necessitam informar algum tipo de resposta. Este módulo é composto pelos componentes (1)Gerência de tipo de erro, (2)Gerência de erros de routing, (3)Gerência de erros de transação, (4)Gerência de erros de comunicação, (5)Gerência de erros de segurança, (6)Gerência de erros de negócio, (7)Gerência de erros de sincronização.
- **Segurança:** Módulo responsável pela garantia da segurança dos serviços e dos dados disponíveis no módulo de serviços, provendo o acesso de interfaces e usuários autorizados. Este módulo é composto pelos componentes (1)Validação de interface, (2)Autenticação, (3)Autorização e (4)Bloqueio de recurso.
- **Gerência de dados:** Módulo responsável pelo gerenciamento dos dados dentro do módulo de serviços, o gerenciamento é composto da validação dos dados, regras de negócio transformação de dados e as interfaces de gerenciamento de dados estruturados e não estruturados. Este módulo é composto pelos componentes (1)Validação e regras de negócio, (2)Gerência de transformação de dados, (3)Gerência de base de dados estruturada, (4)Gerência de base de dados não estruturada.
- **Sincronização:** Módulo responsável pelo gerenciamento de sincronização de dados, compreendendo o enfileiramento, a utilização do protocolo de sincronização adequado, da verificação da repetição dos registros obedecendo as regras de negócio

estipuladas para o projeto, além da notificação da realização das operações caso seja necessário realizar o processamento de sincronização em *background*. Este módulo é composto pelos componentes. (1)Gerência de enfileiramento, (2)Gerência de repetição, (3)Gerência do tipo de sincronização e (4)Notificação de operação.

- **Comunicação:** Módulo responsável pelo gerenciamento da comunicação do módulo de serviços, este módulo é capaz de determinar o tipo de requisições que serão realizadas, o volume das requisições, o tratamento de respostas e a gerência dos diferentes tipos de respostas. Este módulo é composto pelos componentes (1)Gerência de tipo de requisição, (2)Gerência de volume de requisição, (3)Gerência de volume de resposta, (4)Tratamento de resposta.

5.1.2.3. Componentes

- **Validação de requisição de recurso:** Este componente tem como objetivo validar a rota do serviço requisitado, se a rota existir o componente de Fluxo de informação e o módulo de segurança serão ativados para continuar o processamento da requisição. Se por ventura a rota para o serviço não existir os módulos de Gerência de erros e de Comunicação serão ativados para processar o erro e informar sua ocorrência.
- **Fluxo de informação:** Este componente tem como meta gerenciar o fluxo informacional dentro da aplicação, definindo quais módulos e componentes serão ativados enquanto um serviço é executado, uma implementação para este componente pode ser uma classe controladora.
- **Validação de interface:** Este componente é responsável por comparar chaves de validação existentes nas requisições realizadas por todas as interfaces válidas que se comunicar com o módulo de serviços. Esta validação garante um grau inicial de segurança onde somente interfaces válidas terão acesso aos serviços disponibilizados.
- **Autenticação:** Este componente verifica se o usuário ou *endpoint* que está realizando as requisições dos serviços disponibilizados está autenticado, ou seja, se o usuário ou *endpoint* são entidades registradas.

- **Autorização:** Este componente verifica se o usuário ou *endpoint* autenticado que está realizando as requisições dos serviços disponibilizados está autorizado a acessar os recursos disponibilizados, ou seja, se o perfil atrelado ao usuário ou endpoint garante o acesso ao recurso.
- **Bloqueio de recurso:** Este componente é responsável por evitar que o recurso possa ser utilizado mediante a falha na validação de interface, de autenticação e de autorização. A ativação dos módulos de Gerência de erros e Comunicação são necessários para informar a ocorrência do erro de segurança.
- **Verificador de recursos básicos:** Este componente gerencia a disponibilidade e estabilidade do servidor em que a aplicação está disponibilizada, de forma que mediante situações que possam comprometer a disponibilidade e a estabilidade dos serviços propostos, este módulo pode ativar o módulo Provedor de recursos de alto nível, com o objetivo de redirecionar a requisição para outro servidor saudável.
- **Provedor de recursos de alto nível:** Este componente tem a meta de redirecionar as requisições existentes para outros servidores saudáveis.
- **Gerência de tipo de erro:** Este componente tem o papel de gerenciar a ativação dos componentes de gerência de erro corretos para cada tipo de erro previsto pela aplicação.
- **Gerência de erros de routing:** Este componente tem o papel de gerenciar os diferentes tipos de erro de routing possíveis dentro da aplicação, respondendo com um código correto e uma mensagem, é importante salientar que o tipo de comunicação realizada depende do componente Gerência do tipo de requisição presente no módulo de Comunicação.
- **Gerência de erros de transação:** Este componente encarrega-se da gerência de erros oriundos das transações ocorridas durante a execução dos serviços disponibilizados na aplicação. Caso as operações impactem diretamente na não execução de um serviço, a aplicação deverá finalizar o fluxo de serviço respondendo com um código correto e uma mensagem, é importante salientar que o tipo de comunicação realizada depende do componente Gerência do tipo de requisição presente no módulo de Comunicação.

- **Gerência de erros de comunicação:** Este componente deverá ser responsável por gerenciar erros ocorridos devido complicações referentes a comunicação com outros *endpoints*, ou com o módulo de coleta de dados. Caso necessário deve-se ativar o módulo de comunicação e fornecer a resposta apropriada informando o erro de comunicação.
- **Gerência de erros de segurança:** Este componente gerencia os erros oriundos de problemas de identificação de requisições vindas de interfaces válidas, entidades não autenticadas e entidades não autorizadas. Caso necessário deve-se ativar o módulo de comunicação e fornecer a resposta apropriada informando o erro de comunicação.
- **Gerência de erros de negócio:** Este componente é responsável por gerenciar a comunicação de erros advindos do não cumprimento de validações requeridas pela aplicação. Caso necessário deve-se ativar o módulo de comunicação e fornecer a resposta apropriada informando o erro de comunicação.
- **Gerência de erros de sincronização:** Este componente é responsável por gerenciar erros advindos de problemas em relação a sincronização. O erro mais comum que pode ser verificado por este componente é o erro de repetição de registro, onde a criação de um registro não pode ser realizada pois o registro já existe na base de dados principal. Além deste erro, este componente pode verificar a utilização de protocolos de sincronização não suportados pela aplicação.
- **Validação e regras de negócio:** Este componente é encarregado de verificar o cumprimento das validações específicas do projeto. Em virtude disto este módulo é íntimo da instância do módulo de serviços, uma vez que as validações necessárias irão variar de acordo com o projeto.
- **Gerência de transformação de dados:** Este componente é responsável por realizar a transformação dos dados caso seja necessária. Este processo pode ocorrer antes da persistência dos dados nas bases de dados, com o objetivo de prover o tipo de dado correto que deve ser armazenado. Além disto o processo de transformação também pode ocorrer depois da consulta a base de dados e armazenamento dos dados em alguma estrutura de dados, para a realização do manejo de informações dentro da aplicação.

- **Gerência de base de dados estruturada:** Este componente permite o processo de leitura, inserção, alteração e deleção de dados estruturados.
- **Gerência de base de dados não estruturada:** Este componente permite o processo de leitura, inserção, alteração e deleção de dados não estruturados, como por exemplo, fotos arquivos de áudio, vídeos e arquivos APK.
- **Gerência de enfileiramento:** Este componente é responsável pelo enfileiramento de operações de sincronização, este componente deve ser capaz de iniciar tarefas que rodem em *background*, reduzindo desta forma o impacto na porção de serviços de disponibilização de informações.
- **Gerência de repetição:** Este componente é responsável pela verificação de registro duplos em relação a operações que tangem aspectos de sincronização.
- **Gerência do tipo de sincronização:** Este componente gerencia de os diferentes protocolos de sincronização utilizados durante as requisições fornecidas pelo módulo de coleta de dados, ou o módulo de sincronização opcional.
- **Notificação de operação:** Este componente é responsável por observar as operações que são realizadas em *background* referentes à sincronização, com o objetivo de informar os componentes e módulos que possam ser influenciados pela execução de tais operações.
- **Gerência de tipo de requisição:** Este componente é responsável pela verificação do tipo de requisição interceptada, ou do tipo de requisição que será necessário realizar. Podendo controlar se a requisição será do tipo de requisição única ou requisição em bloco.
- **Gerência de volume de requisição:** Este componente gerencia a estratégia de divisão do bloco de informações que será enviado.
- **Gerência de volume de resposta:** Este componente é responsável pela montagem de blocos de informação vindos em requisições oriundas dos módulos de coleta de dados e sincronização opcional.
- **Tratamento de resposta:** Este componente é responsável por realizar o tratamento das diferentes respostas advindas da interação do módulo de serviços com outros *endpoints*.

5.1.3. Módulo de sincronização

O terceiro módulo a ser elaborado foi o módulo de sincronização, também chamado de módulo de sincronização opcional. A elaboração deste módulo foi inspirada nas necessidades de gerência do processo de sincronização definidos por Sant'anna *et al.*, (2014), além das necessidades demonstradas pelo grupo de usuários que foi avaliado por meio de *User studies*, com o objetivo de avaliar as qualidades de tempo de execução do protótipo arquitetural elaborado.

O módulo de sincronização tem o objetivo de fornecer uma abstração arquitetural coesa para de sistema de coleta de dados que em que (i) Os dispositivos de coleta necessitam enviar dados de forma rápida para o servidor, (ii) Os dispositivos de coleta não podem resolver conflitos e/ou erros de registros com o servidor e (iii) os dispositivos de coleta apresentam mobilidade, e mediante esta, seja possível que um dispositivo móvel colete dados que já foram coletados por outro dispositivo de coleta.

Este módulo também tem sua estrutura organizada em camadas, as quais apresentam módulos internos e componentes.

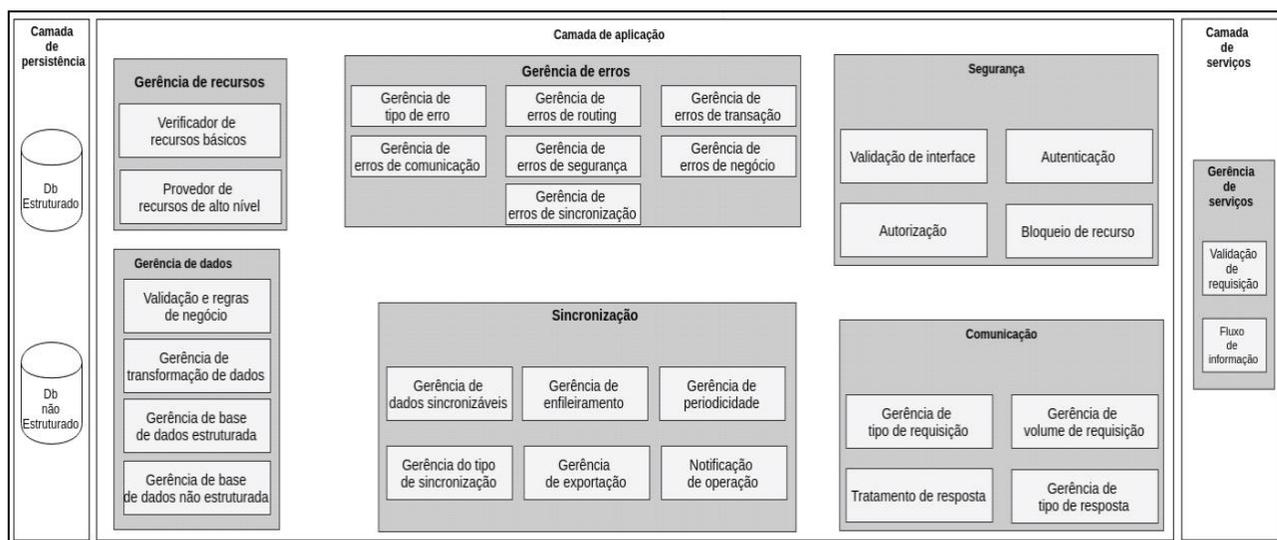


Figura 32: Porção de serviços. Fonte: Própria do autor

5.1.3.1. Camadas

- **Persistência:** Camada responsável por armazenar os dados que foram transmitidos pelo módulo de coleta. Os elementos presentes nesta camada são as possíveis bases de dados estruturadas e não estruturadas. É importante ressaltar que devido o escopo de atuação deste módulo, as suas bases de dados poderão permitir a repetição de registros, uma vez que a correção de tais repetições será fruto da a interação deste módulo com o módulo de serviços.
- **Aplicação:** Camada responsável pelo processamento de dados e operações de sincronização da aplicação. Os elementos pertencentes a esta camada são os módulos de Gerência de recursos, Gerência de erros, Gerência de dados, Segurança, Sincronização e Comunicação.
- **Serviços:** Camada responsável pela pelo processo de garantia de fornecimento dos serviços corretos. O elemento presente nesta camada é o módulo de Gerência de serviços.

5.1.3.2. MÓDULOS

- **Gerência de serviços:** Módulo responsável pela exposição dos serviços disponibilizados pelo módulo de sincronização, além da organização do fluxo de informação para a realização dos serviços. Este módulo é contém os componentes de (1)Validação de requisição de recurso e (2)Fluxo de informação.
- **Gerência de recursos:** Módulo responsável pela verificação da disponibilidade e estabilidade dos recursos básicos disponibilizados pelo servidor, além da capacidade de manutenção da disponibilidade, redirecionando a requisição mediante a instabilidades. Este módulo é composto pelos componentes (1)Verificador de recursos básicos e (2)Provedor de recursos de alto nível.
- **Gerência de erros:** Módulo responsável pelo tratamento dos erros ocorridos, além disso este módulo também provê as respostas para os erros que necessitam informar algum tipo de resposta. Este módulo é composto pelos componentes (1)Gerência de tipo de erro, (2)Gerência de erros de routing, (3)Gerência de erros de transação,

(4)Gerência de erros de comunicação, (5)Gerência de erros de segurança, (6)Gerência de erros de negócio, (7)Gerência de erros de sincronização.

- **Segurança:** Módulo responsável pela garantia da segurança dos serviços e dos dados disponíveis no módulo de sincronização, provendo o acesso de interfaces e usuários autorizados. Este módulo é composto pelos componentes (1)Validação de interface, (2)Autenticação, (3)Autorização e (4)Bloqueio de recurso.
- **Gerência de dados:** Módulo responsável pelo gerenciamento dos dados dentro do módulo de sincronização, o gerenciamento é composto da validação dos dados, regras de negócio transformação de dados e as interfaces de gerenciamento de dados estruturados e não estruturados. Este módulo é composto pelos componentes (1)Validação e regras de negócio, (2)Gerência de transformação de dados, (3)Gerência de base de dados estruturada, (4)Gerência de base de dados não estruturada.
- **Sincronização:** Módulo responsável pelo gerenciamento de sincronização de dados, compreendendo o enfileiramento, a utilização do protocolo de sincronização adequado, da verificação da repetição dos registros obedecendo as regras de negócio estipuladas para o projeto, além da notificação da realização das operações caso seja necessário realizar o processamento de sincronização em *background*. Este módulo é composto pelos componentes (1)Gerência de dados sincronizáveis, (2)Gerência de enfileiramento, (3)Gerência de periodicidade, (4)Gerência do tipo de sincronização, (5)Gerência de exportação e (6)Notificação de operação.
- **Comunicação:** Módulo responsável pelo gerenciamento da comunicação do módulo de sincronização, este módulo é capaz de determinar o tipo de requisições que serão realizadas, o volume das requisições, o tratamento de respostas e a gerência dos diferentes tipos de respostas. Este módulo é composto pelos componentes (1)Gerência de tipo de requisição, (2)Gerência de volume de requisição, (3)Gerência de volume de resposta, (4)Tratamento de resposta.

5.1.3.3. COMPONENTES

- **Validação de requisição de recurso:** Este componente tem como objetivo validar a rota do serviço requisitado, se a rota existir o componente de Fluxo de informação e o

módulo de segurança serão ativados para continuar o processamento da requisição. Se por ventura a rota para o serviço não existir os módulos de Gerência de erros e de Comunicação serão ativados para processar o erro e informar sua ocorrência.

- **Fluxo de informação:** Este componente tem como meta gerenciar o fluxo informacional dentro da aplicação, definindo quais módulos e componentes serão ativados enquanto um serviço é executado, uma implementação para este componente pode ser uma classe controladora.
- **Verificador de recursos básicos:** Este componente gerencia a disponibilidade e estabilidade do servidor em que a aplicação está disponibilizada, de forma que mediante a situações que possam comprometer a disponibilidade e a estabilidade dos serviços propostos, este módulo pode ativar o módulo Provedor de recursos de alto nível, com o objetivo de redirecionar a requisição para outro servidor saudável.
- **Provedor de recursos de alto nível:** Este componente tem a meta de redirecionar as requisições existentes para outros servidores saudáveis.
- **Gerência de tipo de erro:** Este componente tem o papel de gerenciar a ativação dos componentes de gerência de erro corretos para cada tipo de erro previsto pela aplicação.
- **Gerência de erros de routing:** Este componente tem o papel de gerenciar os diferentes tipos de erro de routing possíveis dentro da aplicação, respondendo com um código correto e uma mensagem. É importante salientar que o tipo de comunicação realizada depende do componente Gerência do tipo de requisição presente no módulo de Comunicação.
- **Gerência de erros de transação:** Este componente encarrega-se da gerência de erros oriundos das transações ocorridas durante a execução dos serviços disponibilizados na aplicação. Caso as operações impactem diretamente na não execução de um serviço, a aplicação deverá finalizar o fluxo de serviço respondendo com um código correto e uma mensagem. É importante salientar que o tipo de comunicação realizada depende do componente Gerência do tipo de requisição presente no módulo de Comunicação.
- **Gerência de erros de comunicação:** Este componente deverá ser responsável por gerenciar erros ocorridos devido complicações referentes a comunicação com outros

endpoints, com o módulo de coleta de dados e o módulo de serviços. Caso necessário deve-se ativar o módulo de comunicação e fornecer a resposta apropriada informando o erro de comunicação.

- **Gerência de erros de segurança:** Este componente gerencia os erros oriundos de problemas de identificação de requisições vindas de interfaces válidas, entidades não autenticadas e entidades não autorizadas. Caso necessário deve-se ativar o módulo de comunicação e fornecer a resposta apropriada informando o erro de comunicação.
- **Gerência de erros de negócio:** Este componente é responsável por gerenciar a comunicação de erros advindos do não cumprimento de validações requeridas pela aplicação. Caso necessário deve-se ativar o módulo de comunicação e fornecer a resposta apropriada informando o erro de comunicação.
- **Gerência de erros de sincronização:** Este componente é responsável por gerenciar erros advindos de problemas em relação a sincronização, o que no caso do módulo de sincronização é representado por problemas referentes a exportação dos dados já existentes nas bases do módulo.
- **Validação de interface:** Este componente é responsável por comparar chaves de validação existentes nas requisições realizadas por todas as interfaces válidas que se comunicar com o módulo de serviços. Esta validação garante um grau inicial de segurança onde somente interfaces válidas terão acesso aos serviços disponibilizados.
- **Autenticação:** Este componente verifica se o usuário ou *endpoint* que está realizando as requisições dos serviços disponibilizados está autenticado, ou seja, se o usuário ou *endpoint* são entidades registradas.
- **Autorização:** Este componente verifica se o usuário ou *endpoint* autenticado que está realizando as requisições dos serviços disponibilizados está autorizado a acessar os recursos disponibilizados, ou seja se o perfil atrelado ao usuário ou *endpoint* garante o acesso ao recurso.
- **Bloqueio de recurso:** Este componente é responsável por evitar que o recurso possa ser utilizado mediante a falha na validação de interface, de autenticação e de autorização. A ativação dos módulos de Gerência de erros e Comunicação são necessários para informar a ocorrência do erro de segurança.

- **Validação e regras de negócio:** Este componente é encarregado de verificar o cumprimento das validações específicas do projeto. Em virtude disto este módulo é íntimo da instância do módulo de serviços, uma vez que as validações necessárias irão variar de acordo com a instância arquitetural que está sendo utilizada.
- **Gerência de transformação de dados:** Este componente é responsável por realizar a transformação dos dados caso seja necessária. Este processo pode ocorrer antes da persistência dos dados nas bases de dados, com o objetivo de armazenar os dados de forma correta. Além disto o processo de transformação também pode ocorrer depois da consulta a base de dados e armazenamento dos dados em alguma estrutura de dados, para a realização do manejo de informações dentro da aplicação.
- **Gerência de base de dados estruturada:** Este componente permite o processo de leitura, inserção, alteração e deleção de dados estruturados.
- **Gerência de base de dados não estruturada:** Este componente permite o processo de leitura, inserção, alteração e deleção de dados não estruturados, como por exemplo, fotos arquivos de áudio, vídeos e arquivos APK.
- **Gerência de dados sincronizáveis:** Este componente é responsável pela verificação de dados que podem sofrer o processo de enfileiramento para a exportação, exportação e deleção posterior que ocorrer o fim do processo de exportação de tais dados para o módulo de serviços.
- **Gerência de enfileiramento:** Este componente é responsável pelo enfileiramento de operações de exportação, este componente deve ser capaz de iniciar tarefas que rodem em *background*, reduzindo desta forma o impacto na porção de serviços de disponibilização de informações.
- **Gerência de periodicidade:** Este componente é responsável pela determinação da periodicidade com que serão realizadas atividades de exportação para o(s) módulo(s) de serviço(s).
- **Gerência do tipo de sincronização:** Este componente gerencia de os diferentes protocolos de sincronização utilizados durante as requisições fornecidas pelo módulo de coleta de dados, além disto, a escolha de protocolo também pode ser realizada, para a efetivação do processo de exportação de dados para o módulo de serviço, o qual deve estar devidamente preparado para tratar o tipo de protocolo utilizado.

- **Gerência de exportação:** Este componente é responsável pelo processo composto pelas atividades de preparação para o processo de exportação, a utilização do módulo de Comunicação, e a possível utilização dos módulos de Gerência de erros ou Gerência de comunicação, além de informar a outras atividades e processos que dependem da execução do processo de exportação de dados para o módulo de serviço.
- **Notificação de operação:** Este componente é responsável por observar as operações que são realizadas em *background* referentes a exportação, com o objetivo de informar os componentes e módulos que possam ser influenciados pela execução de tais operações.
- **Gerência de tipo de requisição:** Este componente é responsável pela verificação do tipo de requisição interceptada, ou do tipo de requisição que será necessário realizar. Podendo controlar se a requisição será do tipo de requisição única ou requisição em bloco.
- **Gerência de volume de requisição:** Este componente gerencia a estratégia de divisão do bloco de informações que será enviado.
- **Gerência de volume de resposta:** Este componente é responsável pela montagem de blocos de informação vindos em requisições oriundas dos módulos de coleta de dados e sincronização opcional.
- **Tratamento de resposta:** Este componente é responsável por realizar o tratamento das diferentes respostas advindas da interação do módulo de sincronização com outros *endpoints*.

5.2. AVALIAÇÃO DE PROTÓTIPO ARQUITETURAL

Para analisar o protótipo arquitetural juntamente com a metodologia de *User Studies*, foi desenvolvido um questionário avaliativo. A elaboração do mecanismo avaliativo considerou elementos que pudessem refletir aspectos de qualidade em tempo de execução, como performance, segurança, funcionalidade e usabilidade. Além disto, também foram considerados aspectos inerentes ao contexto em que a implementação da instância arquitetural pertencia.

Como descrito na seção de metodologia, aplicou-se o questionário de avaliação para um total de seis participantes, os quais são colaboradores da Agência Eletrobras Eletronorte e mantenedores do projeto “Banco de Germoplasma”, o que confere aos usuários avaliados uma grande experiência referente ao domínio de coleta de dados. O questionário era constituído de uma breve análise de perfil dos futuros usuários, bem como um total de oito perguntas objetivas e duas perguntas discursivas, o que permitia que os futuros usuários analisassem a aplicação.

5.2.1. Análise quantitativa

A avaliação foi realizada com 6 funcionários da Agência Eletrobrás Eletronorte, os quais são integrantes do projeto “Banco de Germoplasma”. Quanto ao perfil desses integrantes, a maioria (50%) apresentava idade maior que 40 anos, pertencia ao sexo masculino (83%), tem como ocupação Fiscal de meio ambiente (67%), utiliza dados móveis e faz a coleta de dados ocasionalmente ou frequentemente, ambos com 50% (Tabela 26).

Tabela 26 - Perfil dos usuários

	(N)	(%)
Idade		
20-30	2	33
30-40	1	17
>40	3	50
Total	6	100
Sexo		
Feminino	1	17
Masculino	5	83
Total	6	
Ocupação		
Fiscal de meio ambiente	4	67
Técnico de produção	2	33
Total	6	100
Uso de D. Móveis		
Nunca	0	0
Ocasionalmente	3	50
Frequentemente	3	50
Total	6	100
Coleta de dados		
Nunca	0	0

Ocasionalmente	3	50
Frequentemente	3	50
Total	6	100

Fonte: Própria do autor

Na avaliação dos integrantes quanto ao sistema que foi desenvolvido com base na instância da arquitetura de referência Ostara, as perguntas objetivas tinham como resposta uma escala de um a sete, onde o valor um significava “Discordo totalmente” e o valor sete “Concordo totalmente” e para melhor analisar as repostas, as mesmas foram agrupadas.

Com isso, 83% dos participantes afirmaram que concordam totalmente que fora capaz de realizar o *login* na aplicação (Gráfico 01) e realizar a sincronização dos dados criados no aparelho móvel (Gráfico 02). E 100% dos participantes concordaram totalmente serem capazes de realizar manutenção das matrizes, fenologias quinzenais e fenologias anuais; que o sistema reflete o processo de coleta vivenciado na organização simplificaria a realização das tarefas internas e recorrentes na empresa, traria vantagens para a operação mantida pela organização e que poderia ser adaptado para outros contextos de coleta de dados mantidos pela organização.

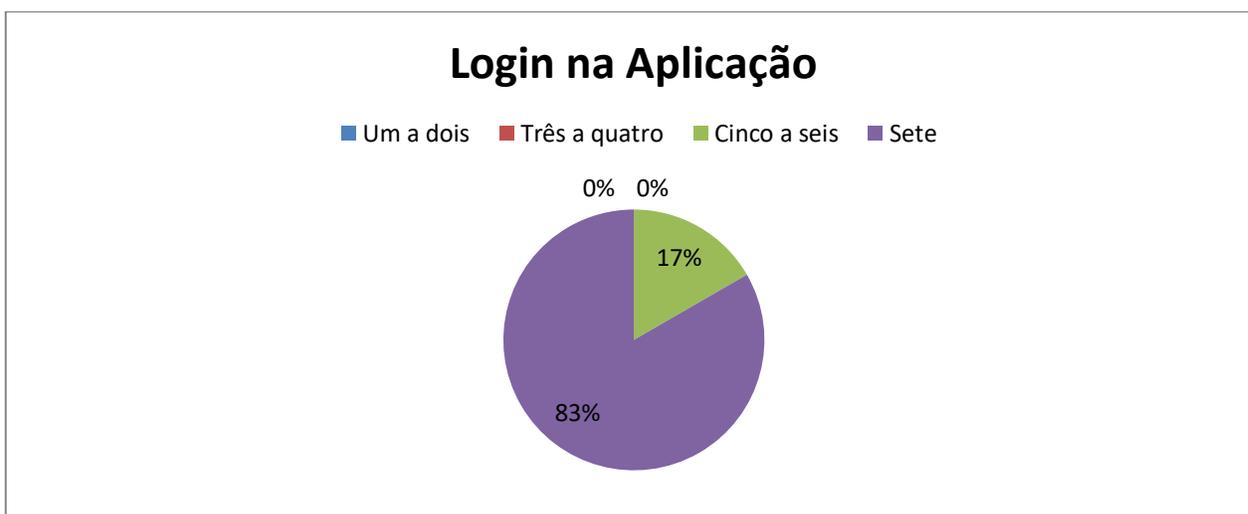


Gráfico 1 - Gráfico da questão sobre se o participante foi capaz de fazer o *login*.Fonte:

Própria do autor



Gráfico 2 - Gráfico da questão sobre se o participante foi capaz de realizar a sincronização dos dados criados no aparelho móvel. Fonte: Própria do autor

5.2.2. Análise qualitativa

Para analisar os dados coletados através das questões descritivas presentes no formulário avaliativo optou-se pelo desdobramento e agrupamento analógico dos textos coletados em categorias. Este método avaliativo foi escolhido devido sua capacidade de avaliar opiniões demonstradas de forma qualitativa.

As categorias foram utilizadas para agrupar de forma estruturada as respostas coletadas nas perguntas descritivas presentes no formulário avaliativo. A criação das categorias foi guiada através do conteúdo das respostas dos usuários contrapostos a atributos de qualidade de tempo de execução como performance, funcionalidade, usabilidade e segurança.

Após a realização do processo de análise categorial, os dados foram interpretados de acordo com o método de análise de conteúdo. Posteriormente os resultados obtidos foram quantificados. Desta forma espera-se obter resultados expressivos em relação as perguntas descritivas.

O número de entrevistados foi de 6 pessoas, onde 5 tiveram mais de uma resposta para a pergunta “*Em que sentido você acha que este sistema de coleta de dados melhora ou piora a execução de suas tarefas?*”, sendo assim, o número de respostas é de 14

(N=14). Com isso, percebe-se que a maioria dos entrevistados revela que a aplicação traz Agilidade no processo de disponibilização final dos dados (35,7%) (Tabela 27).

Tabela 27 - Pergunta “*Em que sentido você acha que este sistema de coleta de dados melhora ou piora a execução de suas tarefas?*”

(Continua)

Respostas	N	%
Agilidade no processo de coleta de dados <i>in loco</i>	3	21,4
Agilidade no processo de disponibilização final dos dados	5	35,7
Facilidade do processo de coletar de dados <i>in loco</i>	2	14,3

Tabela 27 - Pergunta “*Em que sentido você acha que este sistema de coleta de dados melhora ou piora a execução de suas tarefas?*”

(Conclusão)

Respostas	N	%
Facilidade do processo de disponibilização final dos dados	1	7,1
Segurança das informações, devido o uso de recursos informatizados	1	7,1
Diminuição de custos	1	7,1
Outros	1	7,1
Total	14	100

Fonte: Própria do autor

Para a pergunta “*Você tem algum outro comentário relacionado ao sistema, ou algo relacionado?*”, dos 6 entrevistados, todos responderam somente uma das categorias, portanto o número de respostas foi igual a 6 (N=6). Sendo assim, a categoria Adição de funcionalidade revelou com 33,3% que existem funcionalidades que poderiam ser

adicionadas ao protótipo para proporcionar melhores aspectos de qualidades referentes a agilidade facilidade e diversidade de dados monitorados.

Tabela 28 - Pergunta 2

Respostas	N	%
Utilização da instância para outro contexto de coleta de dados mantido pela organização.	1	16,7
Correção de conceitos referentes ao negócio	1	16,7
Adição de funcionalidade	2	33,3
Outros	2	33,3
Total	6	100

Fonte: Própria do autor

5.3. DISCUSSÃO

A arquitetura de referência Ostara foi desenvolvida para auxiliar o desenvolvimento de sistemas que tenham o processo de coleta de dados em campo como parte integrante de seu escopo e/ou suas atividades.

Os resultados obtidos através da análise do protótipo arquitetural proposto demonstram a presença de qualidades de tempo de execução, o que permite verificar o cumprimento das necessidades de negócio requeridas pelos especialistas de domínio participantes do estudo conduzido.

Contudo é importante ressaltar que a avaliação realizada se mostrou incapaz em avaliar aspectos de qualidade de desenvolvimento, como por exemplo, manutenibilidade, modificabilidade, portabilidade, integrabilidade e interoperabilidade, portanto não demonstra a total validação da arquitetura Ostara.

Por este motivo, é necessária a análise crítica entre a arquitetura de referência Ostara e os trabalhos correlatos que serviram como base para o seu desenvolvimento, com o intuito de realizar questionamentos e ponderamentos referentes a sua possível utilização.

Essencialmente a instância arquitetural elaborada buscou representar os requisitos básicos necessários para a definição de um sistema de coleta e dados, o qual

pudesse utilizar recursos de computação móvel. Além disto, a arquitetura instanciada também foi elaborada com o objetivo de suprir as necessidades específicas do projeto “Banco de Germoplasma”, ou seja, a instância arquitetural foi elaborada para prover um sistema de coleta de dados referentes aos eventos de ciclo de vida periódico de espécies botânicas da flora local de Tucuruí - PA, com suporte a computação móvel.

Desta forma faz-se lógico analisar a arquitetura de referência Ostara e sua instância juntamente com o trabalho de Waku *et al.*, (2015), pois é possível perceber o mesmo escopo de aplicação, no sentido da utilização de softwares que desempenham coleta de dados, tendo como meta a realização de acompanhamento fenológico.

Ao realizar uma análise crítica do trabalho de Waku *et al.*, (2015) averigua-se que a estratégia de validação utilizada é composta de uma sucessão de testes de campo, realizados em protótipos de software. Tais testes buscavam verificar as implementações geradas, melhorando-as incrementalmente, buscando responder perguntas em relação à tolerância à falhas. As perguntas propostas foram respondidas, de modo que a tolerância a falhas pode ser mantida mediante a existência de propriedades de redundância definidas pelo software *product line*. Em virtude disto percebe-se que a arquitetura de referência Ostara poderia ser utilizada para prover um sistema com os requisitos de tolerância a falhas necessários, devido ao componente de Gerência de importação e exportação, o qual provê mecanismos de backup mediante uma necessidade programática. Entretanto a possibilidade de utilização de vários dispositivos móveis que compartilham uma mesma base de dados, como alternativa para melhorar a disponibilidade de sistemas móveis que são limitados a quantidade de bateria, não está bem definida na especificação da (AR).

Ao realizar a análise crítica dos trabalhos de Freitas Filho (2014), Santos *et al.*, (2013), Ji *et al.*, (2010) e Boni *et al.*, (2006) é possível perceber que os módulos internos e os componentes da arquitetura de referência Ostara podem fornecer uma base para o desenvolvimento de instâncias arquiteturais dos dois trabalhos.

Referente à parte específica de sincronização dos dados, é importante ressaltar que o trabalho de Freitas Filho (2014) apesar de definir um breve esquema de como a sincronização de dados deve ser executada, não comenta sobre as notificações referentes às execuções efetuadas com sucesso nas sincronizações, as quais são

necessárias para que o módulo de coleta de dados não fique sem resposta e possa continuar a execução de suas funções.

A avaliação da publicação realizada por Ji *et al.*, (2010), demonstra que o maior desafio seria a representação correta dos dados de mapeamento tridimensional, os quais são aspectos referentes às regras de negócio inerentes as implementações. Estas características poderiam ser desenvolvidas utilizando-se de instâncias dos componentes de Gerência de transformação dos dados, Gerência de base de dados interna e Validação e regras de negócio.

A publicação de Brega *et al.*, (2008) apresenta um desafio que é suportado parcialmente pela arquitetura de referência Ostara, sendo este a comunicação com outros softwares e *endpoints*. No formato atual a arquitetura Ostara poderia utilizar-se da instância dos módulos internos de Comunicação e Gerência de dados, pertencentes ao módulo de serviços. A utilização destes módulos internos e seus componentes pode suprir as necessidades de comunicação com diferentes *endpoints* e o processamento de suas respostas. Todavia mediante a necessidade de comunicação e tratamento de respostas de diversos *endpoints* para a disponibilização de um determinado serviço, acaba por denotar a necessidade de desenvolvimento de um módulo interno específico pela extensibilidade dos serviços disponíveis no módulo de serviços.

Ao analisar a arquitetura de referência Ostara juntamente com trabalhos de Silva, Andrade e Marin (2008); Tian, Gao, Zhou (2008); Zhao, Zhang e Meng (2012) e Sant'ana *et al.*, (2014), é possível ver que apesar de que os referentes trabalhos pertençam fundamentalmente ao domínio de coleta de dados de campo, os mecanismos de coleta de dados utilizados por eles não possuem características de computação móvel. De modo que uma vez que a avaliação realizada neste trabalho pode avaliar parcialmente uma instância arquitetural voltada para o subdomínio de coleta de dados móveis, não se faz auspicioso inferir que a utilização da arquitetura de referência Ostara poderia prover todos os requisitos arquiteturais necessários para a elaboração de sistemas semelhantes aos expostos nos trabalhos de Silva, Andrade e Marin (2008); Tian, Gao, Zhou (2008); Zhao, Zhang e Meng (2012) e Sant'ana *et al.*, (2014).

Mediante as considerações realizadas em relação aos resultados provenientes da avaliação, conclui-se que a arquitetura de referência proposta neste trabalho faz-se útil

para sistemas que realizem tarefas de coleta de dados móveis, e que possuam o escopo arquitetural semelhante ao escopo da instância arquitetural avaliada.

6. CONCLUSÃO

O presente trabalho teve por objetivo, elaborar uma arquitetura de referência de software voltada para domínio de coleta de dados, e por consequência o subdomínio de coleta de dados móveis, abrangendo sistemas que de alguma forma desempenham tarefas coleta de dados, com o objetivo de concluir total ou parcialmente um determinado processo específico.

Este trabalho iniciou-se com o estudo da literatura disponível sobre arquiteturas de referência, tal estudo teve como objetivo compreender a definição e conceitos de características primordiais de arquiteturas de referência de software. Durante a referida pesquisa, foi possível definir conceitos de domínio de aplicação ao qual a arquitetura de referência iria pertencer, além da escolha de um processo de elaboração e construção de arquiteturas de referência consolidado e maduro.

Após a definição do domínio, buscou-se reunir conhecimentos que pudessem auxiliar no processo de compreensão da definição do domínio de coleta de dados. Também buscou-se reunir conhecimentos sobre o subdomínio de coleta de dados móveis, uma vez que, a arquitetura elaborada também busca abranger sistemas que desempenham coleta de dados utilizando-se de dispositivos móveis.

A pesquisa para definir o domínio e o subdomínio os quais a arquitetura de referência é voltada utilizou-se de uma revisão de literatura, da análise de três sistemas organizacionais pertencentes a uma instituição pública confidencial e a análise do projeto “Banco de Germoplasma”, o qual é mantido pela Agência Eletrobras Eletronorte. A pesquisa buscava encontrar correlações entre os requisitos arquiteturais de sistemas que realizassem coleta de dados, com o objetivo final de definir quais requisitos são comuns e primordiais em sistemas de coletas de dados.

Depois de realizar o estudo para compreender a definição e conceitos de características primordiais do domínio que a arquitetura de referência iria pertencer, deu-se início ao processo de elaboração da arquitetura proposta. O processo que foi utilizado é denominado ProSA-RA, basicamente este processo composto de quatro etapas é uma forma estruturada de elaboração e validação de arquiteturas de referência.

Para a realização das etapas do processo ProSA-RA foram utilizadas as seguintes abordagens:

Passo RA1 - Investigação das fontes de informação: Para a obtenção de informações referentes a conceitos domínio relevantes foram utilizadas pessoas, mais especificamente, especialistas de domínio que trabalham desempenhando coleta de dados, a análise de três sistemas de coleta de dados pertencentes a uma instituição pública confidencial e uma revisão sistemática da literatura.

Passo RA2 - Estabelecimento dos requisitos arquiteturais: Foram identificados os requisitos de sistemas de cada um dos sistemas e trabalhos analisados, os quais foram mapeados para requisitos arquiteturais. Por fim os requisitos arquiteturais listados foram mapeados para conceitos de domínio.

Passo RA3 - Projeto arquitetural: Nesta etapa, foram construídos quatro pontos de vista da arquitetura de referência, sendo estes a (i) ponto de vista de elementos transversais (ii) ponto de vista de tempo de execução, (iii) ponto de vista de implantação, (iv) ponto de vista de código fonte. Cada ponto de vista buscou refletir características específicas da arquitetura proposta.

Os pontos de vista desenvolvidos foram compostos de visões, de forma que, cada visão busca exemplificar um ponto de vista por uma determinada ótica. As visões desenvolvidas neste processo foram (i)visão conceitual, (ii)visão funcionalidade, (iii)visão serviços de sistema, (iv)visão variabilidade, (v)visão componentes colaborativos, (vi)visão processo, (vii)visão dados compartilhados, (viii)visão implantação, (ix)visão desenvolvimento, (x)visão da parte física, (xi)visão de tecnologia, (xii)visão lógica, (xiii)visão de módulo, (xiv)visão estrutural.

Após o término do processo do passo RA3 - Projeto arquitetural, nomeou-se a arquitetura de referência proposta como Ostara. O nome proposto é uma referência a deusa anglo-saxã da colheita.

Passo RA4 - Avaliação arquitetural: Nesta etapa, foi proposta a avaliação de protótipo arquitetural, com o intuito de validar qualidades de tempo de execução. Para executar tal objetivo necessitou-se realizar uma instância arquitetural baseada na arquitetura de referência proposta, esta instância foi modelada tomando como base as necessidades do time de mantenedores do projeto Banco de Germoplasma, mantido pela Agência Eletrobras Eletronorte.

Uma vez modelada, a instância arquitetural foi implementada e logo após isto foi realizado o teste do protótipo arquitetural. O teste foi desenvolvido utilizando a metodologia de *UserStudies*, a qual foi escolhida devido à característica de permitir avaliar se uma técnica ou aplicação são eficazes e como isto ocorre.

O estudo foi executado através da utilização do protótipo arquitetural desenvolvido, seguido do preenchimento de um questionário o qual contava com uma breve descrição dos usuários, oito perguntas objetivas e duas perguntas descritivas.

Após o término dos testes, os formulários preenchidos foram analisados quantitativamente para prover uma análise descritiva do grupo de especialistas de domínio que foi avaliado, além de descrever as respostas fornecidas às perguntas objetivas. Posteriormente foi realizada uma análise qualitativa voltada para as perguntas descritivas preenchidas pelos especialistas de domínio. A análise realizada foi a análise de conteúdo, que teve como meta verificar de forma metódica o que cada usuário buscou expressar em suas respostas.

Após o término do processo ProSA-RA, foram avaliados os resultados obtidos através da análise de protótipo arquitetural. Os resultados denotam que a instância arquitetural avaliada demonstra aspectos referentes a qualidade de tempo de execução, tais como performance, segurança, funcionalidade e usabilidade. Desta forma, verifica-se que a arquitetura de referência proposta proporcionou uma instância arquitetural que se mostrou muito relevante para o contexto operacional da Agência Eletrobras Eletronorte, demonstrando o claro auxílio em relação ao processo interno do projeto “Banco de Germoplasma”. Entretanto, o método avaliativo que foi utilizado mostrou-se incapaz de avaliar aspectos de qualidade referentes ao desenvolvimento, como por exemplo, modificabilidade, portabilidade, integrabilidade e interoperabilidade, os quais são importantes para proporcionar indicadores de efetividade da arquitetura de referência, no que se refere à criação de diferentes instâncias arquiteturais para o domínio e subdomínio propostos.

6.1. TRABALHOS FUTUROS

Devido ao fato de que a arquitetura de referência ter demonstrado uma eficiência parcial para o domínio de coleta de dados e o subdomínio de coleta de dados móveis,

denotada pela verificação de eficiência específica para sistemas que se assemelhem com o protótipo arquitetural desenvolvido para o contexto organizacional da Agência Eletrobras Eletronorte, faz-se necessária a realização de novas avaliações dos artefatos arquiteturais gerados por este trabalho.

As avaliações subsequentes terão como objetivo de proporcionar mais indicadores de qualidade, capazes de atestar a efetividade da arquitetura de referência proposta para o domínio de coleta de dados.

Em virtude da necessidade de mais indicadores de qualidade propõe-se a avaliação da arquitetura de referência Ostara com o auxílio do *framework* FERA (*Framework for Evaluation of Reference Architectures*), o qual demonstra ser uma ferramenta madura. Através da avaliação espera-se obter *insights* cruciais para evoluir a arquitetura de referência Ostara (SANTOS *et al.*, 2013; NAKAGAWA *et al.*, 2014).

REFERÊNCIAS

ALLIAN, Ana Paula; OLIVEIRA JR, Edson; NAKAGAWA, Elisa. VMTTools-RA-Uma Proposta de Arquitetura de Referência para Ferramentas de Gerenciamento de Variabilidades. 2015.

ANGELOV, Samuil; GREFEN, Paul; GREEFHORST, Danny. A classification of software reference architectures: Analyzing their success and effectiveness. In: **2009 Joint Working IEEE/IFIP Conference on Software Architecture (WICSA) & 3rd European Conference on Software Architecture (ECSA)**. IEEE, 2009. p. 141-150.

AGARWAL, Sachin; STAROBINSKI, David; TRACHTENBERG, Ari. On the scalability of data synchronization protocols for PDAs and mobile devices. **IEEE network**, v. 16, n. 4, p. 22-28, 2002.

AVGERIOU, Paris. Describing, instantiating and evaluating a reference architecture: A case study. **Enterprise Architecture Journal**, 2003.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. **Software architecture in practice**. Addison-Wesley Professional, 2003.

BASS, Len; CLEMENTS, Paul; KAZMAN, Rick. **Software architecture in practice**. Addison-Wesley Professional, 2012.

BARBOSA, Jorge et al. Computação móvel e ubíqua no contexto de uma graduação de referência. **Brazilian Journal of Computers in Education**, v. 15, n. 3, 2007.

BREGA¹, José Remo Ferreira et al. Levantamento Epidemiológico em Saúde Bucal Utilizando Ferramentas Móveis. 2008.

BONI, Gabriel N. et al. Desenvolvimento de um Software de Coleta de Dados para Pesquisas de Campo Através de Dispositivos Móveis. In: **X Congresso Brasileiro de Informática em Saúde**. 2006.

CUNHA, Daniel Pezzi da et al. Um estudo das estratégias de replicação e reconciliação de banco de dados móveis em um ambiente wireless. 2003.

CUSACK, Carole. The Goddess Eostre: Bede's Text and Contemporary Pagan Tradition (s). **Pomegranate**, v. 9, n. 1, 2007.

DOS SANTOS, Bruno G.; MAMANI-ALIAGA, Alvaro H.; SÁNCHEZ, Jimmy V. Projeto maritaca: Arquitetura e infraestrutura para coleta móvel de dados usando smartphones. In: **31 Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)**. 2013. p. 1076-1083.

DUARTE, L. S. Establishment of a reference architecture for digital television applications. 2012.

EVANS, Eric. **Domain-driven design: tackling complexity in the heart of software**. Addison-Wesley Professional, 2004.

FEITOSA, Daniel. **Simus-uma arquitetura de referência para sistemas multirrobóticos de serviço**. 2013. Tese de Doutorado. Universidade de São Paulo.

FIORAVANTI, Maria Lydia; NAKAGAWA, Elisa Yumi; BARBOSA, Ellen Francine. EDUCAR: Uma Arquitetura de Referência para Ambientes Educacionais. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2010.

FREITAS FILHO, Henrique Pereira de. Arquitetura de coleta de dados para pesquisas de campo em ambientes computacionais heterogêneos. 2014.

GALSTER, Matthias. Software reference architectures: related architectural concepts and challenges. In: **Exploring Component-based Techniques for Constructing Reference Architectures (CobRA), 2015 1st International Workshop on**. IEEE, 2015. p. 1-4.

GUESSI, Milena; DE OLIVEIRA, Lucas Bueno Ruas; NAKAGAWA, Elisa Yumi. Representation of Reference Architectures: A Systematic Review. In: **SEKE**. 2011. p. 782-785.

GROSSKURTH, Alan; GODFREY, Michael W. A reference architecture for web browsers. In: **null**. IEEE, 2005. p. 661-664.

ITO, G. Bancos de Dados Móveis: Uma Análise de e Soluções Propostas para Gerenciamento de Dados. Dissertação de Mestrado em Ciência da Computação, UFSC, 2001.

ITO, Giani Carla; FERREIRA, Maurício; SANT'ANNA, Nilson. Computação móvel: Aspectos de gerenciamento de dados. **INPE-Instituto Nacional de Pesquisas espaciais**, v. 10, p. 17-18, 2003.

JI, Min et al. Research and development of field data collecting synchronously system of mining area. In: **2010 IEEE International Geoscience and Remote Sensing Symposium**. IEEE, 2010. p. 3948-3951.

L. L. Pullum, Software fault tolerance techniques and implementation. Artech House, 2001.

LEE, Byung-Yun et al. Data synchronization protocol in mobile computing environment using yncML. In: **5th IEEE International Conference on High Speed Networks and Multimedia Communication (Cat. No. 02EX612)**. IEEE, 2002. p. 133-137.

LI, Jiao-Long; LI, Jian-Ping. Data synchronization protocol in mobile computing environment using SyncML and Huffman coding. In: **2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP)**. IEEE, 2012. p. 260-262.

LI, Jing; WANG, Jianhua. A new architecture model of mobile database based on agent. In: **2009 First International Workshop on Database Technology and Applications**. IEEE, 2009. p. 341-344.

LUO, Entao et al. Privacyprotector: Privacy-protected patient data collection in IoT-based healthcare systems. **IEEE Communications Magazine**, v. 56, n. 2, p. 163-168, 2018.

MACHADO, Carlos Alberto Nunes. Uma linguagem de descrição arquitetural baseada em uma arquitetura de referência para sistemas ubíquos. 2015.

NAKAGAWA, Elisa Yumi. **Uma contribuição ao projeto arquitetural de ambientes de engenharia de software**. 2006. Tese de Doutorado. Universidade de São Paulo.

NAKAGAWA, Elisa Yumi et al. Towards a Reference Architecture for Software Testing Tools. In: **SEKE**. 2007. p. 1-6.

NAKAGAWA, Elisa Y. et al. An aspect-oriented reference architecture for Software Engineering Environments. **Journal of Systems and Software**, v. 84, n. 10, p. 1670-1684, 2011.

NAKAGAWA, Elisa Yumi; ANTONINO, Pablo Oliveira; BECKER, Martin. Reference architecture and product line architecture: A subtle but critical difference. In: **European Conference on Software Architecture**. Springer, Berlin, Heidelberg, 2011. p. 207-211.

NAKAGAWA, Elisa Yumi; OQUENDO, Flavio; BECKER, Martin. Ramodel: A reference model for reference architectures. In: **Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on**. IEEE, 2012. p. 297-301.

NAKAGAWA, Elisa Y. et al. Consolidating a process for the design, representation, and evaluation of reference architectures. In: **Software Architecture (WICSA), 2014 IEEE/IFIP Conference on**. IEEE, 2014. p. 143-152.

NIELSEN, Jakob. **Usability engineering**. Elsevier, 1994.

OLIVEIRA, Lucas Bueno Ruas; NAKAGAWA, Elisa Yumi. A service-oriented reference architecture for software testing tools. In: **European Conference on Software Architecture**. Springer, Berlin, Heidelberg, 2011. p. 405-421.

PORTOCARRERO, Jesús MT et al. Self-adaptive middleware for wireless sensor networks: a reference architecture. In: **Proceedings of the 2015 European Conference on Software Architecture Workshops**. ACM, 2015. p. 12.

PROCESS, Rational Unified. Best practices for software development teams. **A Rational Software Corporation White Paper**. Recuperado de: [https://www. ibm.](https://www.ibm.com)

com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf, 2001.

REN, Ligang; SONG, Junde. Data synchronization in the mobile Internet. In: **The 7th International Conference on Computer Supported Cooperative Work in Design**. IEEE, 2002. p. 95-98.

SANTOS, José Filipe Marreiros et al. A Checklist for Evaluation of Reference Architectures of Embedded Systems (S). In: **SEKE**. 2013. p. 1-4.

SANT'ANNA, Nilson et al. Modelo Arquitetural para Coleta, Processamento e Visualização de Informações de Clima Espacial. **Simpósio Brasileiro de Sistemas de Informação (2014)**, p. 125-136, 2014.

SCOTT, R. K. et al. Investigating version dependence in fault-tolerant software. **AGARD Design for Tactical Avionics Maintainability 11 p(SEE N 85-16731 08-01)**, 1984.

SILVA, Aldo Jean Soares; ANDRADE, Aderson Soares; MARIN, Fábio Ricardo. Arquitetura para plataforma de coleta e disseminação de dados climáticos no estado do Piauí. **Revista Tecnologia**, v. 29, n. 2, 2008.

SUN, Jun; LIU, Jane. Synchronization protocols in distributed real-time systems. In: **Proceedings of 16th International Conference on Distributed Computing Systems**. IEEE, 1996. p. 38-45.

TIAN, Jingwen; GAO, Meijuan; ZHOU, Hao. Multi-channel seismic data synchronizing acquisition system based on wireless sensor network. In: **2008 IEEE International Conference on Networking, Sensing and Control**. IEEE, 2008. p. 1269-1272.

VASILESCU, Iuliu et al. Data collection, storage, and retrieval with an underwater sensor network. In: **Proceedings of the 3rd international conference on Embedded networked sensor systems**. ACM, 2005. p. 154-165.

WAKU, Gustavo M. et al. A robust software product line architecture for data collection in android platform. In: **2015 IX Brazilian Symposium on Components, Architectures and Reuse Software**. IEEE, 2015. p. 31-39.

WANG, Wei; XU, Peng; YANG, Laurence T. Secure data collection, storage and access in cloud-assisted IoT. **IEEE Cloud Computing**, v. 5, n. 4, p. 77-88, 2018.

ZAMBIASI, Saulo Popov et al. Uma arquitetura de referência para softwares assistentes pessoais baseada na arquitetura orientada a serviços. 2012.

ZHANG, Zhenyou; LU, Cuifang. Research and implementation for data synchronization of heterogeneous databases. In: **2010 International Conference on Computer and Communication Technologies in Agriculture Engineering**. IEEE, 2010. p. 464-466.

ZHAO, Xiaoyan; ZHANG, Zhaohui; MENG, Wei. GPRS based remote data acquisition and forecasting system for long-distance natural Gas pipeline monitoring. In: **2012 International Conference on Computational Problem-Solving (ICCP)**. IEEE, 2012. p. 504-507.

APÊNDICE A - ORGANIZAÇÃO DE FUNCIONALIDADES

(Continua)

Categoria	Subcategoria	Nº	Funcionalidades	
Armazenamento	Módulo servidor principal	01	Criar registros na base de dados estruturada.	
		02	Editar registros na base de dados estruturada.	
		03	Deletar registros na base de dados estruturada.	
		04	Consultar registros na base de dados estruturada.	
		05	Criar registros na base de dados não estruturada.	
		06	Editar registros na base de dados não estruturada.	
		07	Deletar registros na base de dados não estruturada.	
		08	Consultar registros na base de dados não estruturada.	
		09	Validar dados coletados.	
		10	Aplicar regras de negócio.	
		11	Permitir controle transacional.	
		12	Transformar dados para a recuperação e exibição.	
		13	Transformar dados para a recuperação e transporte.	
		14	Transformar dados para o armazenamento apropriado.	
		15	Permitir armazenamento distribuído	
		Módulo de coleta de dados	16	Identificação da estrutura de armazenamento de dados.
			17	Navegação na estrutura de armazenamento de dados.
			18	Criação de arquivos de registro de dados.
			19	Edição de arquivos de registro de dados.
			20	Deleção de arquivos de registro de dados.
			21	Consulta de arquivos de registro de dados.
			22	Leitura de arquivos de registro de dados.
			23	Criação de registros em arquivos na base de dados interna.
			24	Edição de registros em arquivos na base de dados interna.
			25	Deleção de registros em arquivos na base de dados interna.
			26	Consulta de registros em arquivos na base de dados interna.
			27	Permitir validação de dados coletados.
			28	Transformação dos dados para a recuperação e exibição.
			29	Transformação dos dados para a recuperação e transporte.
			30	Transformação dos dados para o armazenamento apropriado.
			31	Permitir criação de backup dos dados coletados.
			32	Backup automático anterior aos processos de sincronização.

(Conclusão)

Categoria	Subcategoria	Nº	Funcionalidades
Armazenamento	Módulo de coleta de dados	33	Deleção automática do backup depois da realização dos processos de sincronização.
	Módulo de coleta de dados móveis	34	Criação automática da base de dados interna dos dispositivos.
		35	Criação de registros na base de dados interna.
		36	Edição de registros na base de dados interna.
		37	Deleção de registros na base de dados interna.
		38	Desabilitar registros na base de dados interna.
		39	Consulta de registros na base de dados interna.
		40	Criação de multi registros.
		41	Deleção de multi registros.
		42	Permitir validação dos dados coletados.
		43	Permitir controle transacional.
		44	Criação automática da base de dados externa dos dispositivos.
		45	Criação de registros na base de dados móvel externa.
		46	Edição de registros na base de dados móvel externa.
		47	Edição de registros na base de dados móvel externa.
		48	Consulta de registros na base de dados móvel externa.
		49	Transformação dos dados para a recuperação e exibição.
		50	Transformação dos dados para a recuperação e transporte.
		51	Transformação dos dados para o armazenamento apropriado.
		52	Replicação automática da base de dados interna para a base de dados externa antes da realização dos processos de sincronização.
		53	Deleção automática da base de dados interna para a base de dados externa depois da realização dos processos de sincronização.
		54	Criação e preenchimento automáticos da base de dados interna dos dispositivos através de backup de base de dados externa.

(Continua)

Categoria	Subcategoria	Nº	Funcionalidades
Coleta	Módulo servidor principal	01	Validação de interface.
		02	Validação de autenticação.
		03	Validação de autorização.
		04	Controlar fluxo para busca de dados.
		05	Controlar fluxo para listagem de dados.
		06	Controlar fluxo para exibição de dados.
		07	Controlar fluxo para criação de dados.
		08	Controlar fluxo para edição de dados.
		09	Controlar fluxo para deleção de dados.
		10	Controlar fluxo para desabilitar dados.
		11	Validação dos dados de criação.
		12	Validação dos dados de edição.
		13	Validação de registro existente.
		14	Gerenciamento de recursos em background.
		15	Redirecionamento de para outro servidor disponível.
	Módulo de coleta de dados	16	Identificação da estrutura de armazenamento de dados.
		17	Permitir captura dos dados
		18	Permitir validação dos dados
		19	Permitir Transformação dos dados
		20	Permitir verificação do nível de energia da bateria.
		21	Permitir carregamento da bateria do módulo de coleta.
		22	Permitir gerenciamento de funções de coleta e sincronização de acordo com o nível de energia da bateria.
		23	Permitir gerencia de carregamento da bateria do módulo de coleta de acordo com o nível de energia da bateria.
	Módulo de coleta de dados móveis	24	Disponibilizar interfaces para listagem dedados.
		25	Disponibilizar interfaces para exibição dedados.
		26	Disponibilizar interfaces para consulta dedados.
		27	Disponibilizar interfaces para criação dedados.
		28	Disponibilizar interfaces para edição dedados.
		29	Disponibilizar interfaces para deleção dedados.
		30	Disponibilizar interfaces para desabilitar dados.
		31	Disponibilizar interfaces para reabilitação dedados.
		32	Disponibilizar interfaces para importação dos dados.
		33	Disponibilizar interfaces para exportação dos dados.
		34	Disponibilizar interfaces para sincronização dos dados.
		35	Permitir navegação entre as diferentes interfaces.
		36	Controlar fluxo para busca de dados.
		37	Controlar fluxo para listagem de dados.

Categoria	Subcategoria	Nº	Funcionalidades
Coleta	Módulo de coleta de dados móveis	38	Controlar fluxo para exibição de dados.
		39	Controlar fluxo para criação de dados.
		40	Controlar fluxo para edição de dados.
		41	Controlar fluxo para deleção de dados.
		42	Controlar fluxo para desabilitar dados.
		43	Validação dos dados de criação coletados.
		44	Validação dos dados de edição coletados.
		45	Confirmação da deleção de dados coletados.
		46	Confirmação para desabilitar dados coletados.
		47	Permitir interpretação arquivos de formulários externos para objetos.
		48	Permitir gerenciamento de sinal.
		49	Permitir verificação do nível de energia da bateria.
		50	Permitir gerencia de funções de acordo com o nível de energia da bateria.

(Continua)

Categoria	Subcategoria	Nº	Funcionalidades	
Sincronização	Módulo servidor principal	01	Verificar existência e duplicidade do registro.	
		02	Replicar para os outros servidores disponíveis.	
		03	Enviar resposta de importação.	
		04	Enviar resposta de exportação.	
		05	Enviar resposta de sincronização.	
	Módulo de coleta de dados	de de	06	Verificar existência dedados sincronizáveis.
			07	Permitir exportar dados individuais.
			08	Permitir exportar dados em bloco.
			09	Permitir requisição em individual.
			10	Permitir requisição em bloco.
			11	Permitir ajuste do bloco de acordo com o sinal.
			12	Tratar respostas retornadas pelo servidor.
			13	Marcar registros que foram sincronizados.
			14	Reenviar dados que não puderam ser sincronizados.
			15	Permitir escolha de comunicação síncrona
			16	Permitir escolha de comunicação assíncrona
	Módulo de coleta de dados móveis	de de	17	Verificar dados sincronizáveis.
			18	Verificar necessidade de sincronização.
			19	Permitir importar dados em bloco.

(Conclusão)

Categoria	Subcategoria	Nº	Funcionalidades
Sincronização	Módulo de coleta de dados	20	Permitir importar dados individuais.
		21	Permitir exportar dados em bloco.
		22	Permitir exportar dados individuais.
		23	Permitir sincronizar dados do servidor e estação móvel.
		24	Permitir requisição em individual.
		25	Permitir requisição em bloco.
		26	Permitir ajuste do bloco de acordo com o sinal.
		27	Tratar respostas retornadas pelo servidor.
		28	Reenviar os dados que não puderam ser sincronizados.
		29	Permitir escolha de comunicação síncrona
		30	Permitir escolha de comunicação assíncrona
		31	Garantir a correta replicação dos dados para manter a integridade de dados entre o módulo de coleta e o módulo servidor.
		32	Garantir a velocidade de comunicação com o servidor.
33	Recriar e popular a base após sincronização de dados.		

(Continua)

Categoria	Subcategoria	Nº	Funcionalidades
Erros	Módulo servidor principal	01	Garantir customização de respostas de erros.
		02	Garantir resposta com o tipo de erro no processo de consulta.
		03	Garantir resposta com o tipo de erro no processo de listagem.
		04	Garantir resposta com o tipo de erro no processo de exibição.
		05	Garantir resposta com o tipo de erro no processo de criação.
		06	Garantir resposta com o tipo de erro no processo de edição.
		07	Garantir resposta com o tipo de erro no processo de importação.
		08	Garantir resposta com o tipo de erro no processo de exportação.
		09	Garantir resposta com o tipo de erro no processo de sincronização.
		10	Garantir resposta com o tipo de erro no processo de autenticação de interface.

(Continua)

Categoria	Subcategoria	Nº	Funcionalidades	
Erros	Módulo servidor principal	11	Garantir resposta com o tipo de erro no processo de autenticação de usuário.	
		12	Garantir resposta com o tipo de erro no processo de autorização de usuário.	
		13	Garantir processo de <i>rollback</i> caso algum erro de inserção múltipla ocorra.	
		14	Garantir processo de <i>rollback</i> caso algum erro de inserção individual ocorra.	
		15	Garantir processo de <i>rollback</i> caso algum erro de edição múltipla ocorra.	
		16	Garantir processo de <i>rollback</i> caso algum erro de edição individual ocorra.	
		17	Garantir proteção de serviços em caso de erro de autenticação de interface.	
		18	Garantir proteção de serviços em caso de erro de autenticação de usuário.	
		19	Garantir proteção de serviços em caso de erro de autorização de usuário.	
		20	Garantir o redirecionamento para outro servidor em funcionamento caso o servidor principal pare de funcionar.	
	Módulo de coleta de dados	de de	21	Garantir a realização da coleta de dados novamente caso algum dado tenha sido captado de forma incorreta.
			22	Garantir a realização do reenvio de dados novamente caso algum dado tenha sido enviado de forma incorreta.
			23	Garantir a identificação de erro.
			24	Garantir o processo correto a ser tomado de acordo com a identificação de erro.
			25	Garantir a integridade do banco local caso algum erro de sincronização ocorra.
			26	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de inserção.
			27	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de exportação.
	Módulo de coleta de dados móveis	de de	28	Permitir exibição de erros na interface de criação de dados.
			29	Permitir exibição de erros na interface de edição de dados.
			30	Permitir a exibição dos erros na interface de importação de dados.
			31	Permitir a exibição dos erros na interface de exportação de dados.

(Conclusão)

Categoria	Subcategoria	Nº	Funcionalidades
------------------	---------------------	-----------	------------------------

Erros	Módulo de coleta de dados móveis	32	Permitir a exibição dos erros na interface de sincronização de dados.
		33	Permitir a exibição dos erros na interface de autenticação caso o usuário não exista no banco de dados remoto.
		34	Permitir a exibição dos erros na interface de autenticação caso o usuário não exista no banco de dados local.
		35	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro criação de registro.
		36	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro edição de registro.
		37	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro importação de registro.
		38	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro exportação de registro.
		39	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro sincronização de registro.
		40	Garantir processo de <i>rollback</i> caso algum erro de inserção múltipla ocorra.
		41	Garantir processo de <i>rollback</i> caso algum erro de inserção individual ocorra.
		42	Garantir processo de <i>rollback</i> caso algum erro de edição múltipla ocorra.
		43	Garantir processo de <i>rollback</i> caso algum erro de edição individual ocorra.
		44	Garantir a integridade do banco local caso algum erro de importação ocorra.
		45	Garantir a integridade do banco local caso algum erro de exportação ocorra.
		46	Garantir a integridade do banco local caso algum erro de sincronização ocorra.
		47	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de inserção.
		48	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de edição.
		49	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de importação.
		50	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de exportação.
		51	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de sincronização.
Categoria	Subcategoria	Nº	Funcionalidades
Segurança	Módulo servidor principal	01	Permitir identificação de interfaces de coleta de dados.
		02	Permitir identificação de interfaces de exibição de dados.
		03	Permitir autenticação de usuários para acesso de serviços.

		04	Permitir autorização de usuários para acesso de serviços.
		05	Permitir autenticação de <i>endpoints</i> para acesso de serviços.
		06	Permitir autorização de <i>endpoints</i> para acesso de serviços.
		07	Permitir configuração para o acesso de usuários para acesso de serviços.
		08	Permitir configuração para o acesso de <i>endpoints</i> para acesso de serviços.
		09	Garantir que o usuário possa importar somente dados através de serviços permitidos.
		10	Garantir que o usuário possa sincronizar somente dados através de serviços permitidos.
	Módulo de coleta de dados	11	Garantir que os dados sejam acessados somente por um servidor específico.
	Módulo de coleta de dados móveis	12	Permitir autorização local de usuários para acesso de serviços.
		13	Permitir autenticação remota de usuários para acesso de serviços.
		14	Permitir a verificação do prazo da autenticação remota antes de importar dados.
		15	Permitir a verificação do prazo da autenticação remota antes de exportar dados.
		16	Permitir a verificação do prazo da autenticação remota antes de sincronizar dados.

Categoria	Subcategoria	Nº	Funcionalidades
Administração	Módulo de visualização de dados	01	Listar sincronizações.
		02	Detalhar sincronizações.
		03	Avaliar sincronizações.
		04	Organizar sincronizações de diferentes formas.

APÊNDICE B – MAPEAMENTO DOS REQUISITOS ARQUITETURAIS

(Continua)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
01	Criar registros na base de dados estruturada.	Manter dados no servidor.
02	Editar registros na base de dados estruturada.	Manter dados no servidor.
03	Deletar registros na base de dados estruturada.	Manter dados no servidor.
04	Consultar registros na base de dados estruturada.	Disponibilizar dados no servidor.
05	Criar registros na base de dados não estruturada.	Manter dados no servidor.
06	Editar registros na base de dados não estruturada.	Manter dados no servidor.
07	Deletar registros na base de dados não estruturada.	Manter dados no servidor.
08	Consultar registros na base de dados não estruturada.	Disponibilizar dados no servidor.
09	Validar dados coletados.	Manter dados no servidor.
10	Aplicar regras de negócio.	Manter dados no servidor.
11	Permitir controle transacional.	Segurança de dados do servidor.
12	Transformar dados para a recuperação e exibição.	Permitir interoperabilidade.
13	Transformar dados para a recuperação e transporte.	Permitir interoperabilidade.
14	Transformar dados para o armazenamento apropriado.	Manter dados no servidor.
15	Permitir armazenamento distribuído	Disponibilizar dados no servidor.
16	Identificação da estrutura de armazenamento de dados.	Manter dados remotos.
17	Navegação na estrutura de armazenamento de dados.	Manter dados remotos.
18	Criação de arquivos de registro de dados.	Manter dados remotos.
19	Edição de arquivos de registro de dados.	Manter dados remotos.
20	Deleção de arquivos de registro de dados.	Manter dados remotos.
21	Consulta de arquivos de registro de dados.	Manter dados remotos.
22	Leitura de arquivos de registro de dados.	Manter dados remotos.
23	Criação de registros em arquivos na base de dados interna.	Manter dados remotos.
24	Edição de registros em arquivos na base de dados interna.	Manter dados remotos.
25	Deleção de registros em arquivos na base de dados interna.	Manter dados remotos.
26	Consulta de registros em arquivos na base de dados interna.	Manter dados remotos.
27	Permitir validação de dados coletados.	Manter dados remotos.
28	Transformação dos dados para a recuperação e exibição.	Permitir interoperabilidade.
29	Transformação dos dados para a recuperação e transporte.	Permitir interoperabilidade.

(Conclusão)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
30	Transformação dos dados para o armazenamento apropriado.	Manter dados remotos.
31	Permitir criação de backup dos dados coletados.	Segurança de dados remotos
32	Backup automático anterior aos processos de sincronização.	Segurança de dados remotos
33	Deleção automática do backup depois da realização dos processos de sincronização.	Segurança de dados remotos
34	Criação automática da base de dados interna dos dispositivos.	Manter dados móveis.
35	Criação de registros na base de dados interna.	Manter dados móveis.
36	Edição de registros na base de dados interna.	Manter dados móveis.
37	Deleção de registros na base de dados interna.	Manter dados móveis.
38	Desabilitar registros na base de dados interna.	Manter dados móveis.
39	Consulta de registros na base de dados interna.	Manter dados móveis.
40	Criação de multi registros.	Manter dados móveis.
41	Deleção de multi registros.	Manter dados móveis.
42	Permitir validação dos dados coletados.	Manter dados móveis.
43	Permitir controle transacional.	Manter dados móveis.
44	Criação automática da base de dados externa dos dispositivos.	Manter dados móveis.
45	Criação de registros na base de dados móvel externa.	Manter dados móveis.
46	Edição de registros na base de dados móvel externa.	Manter dados móveis.
47	Edição de registros na base de dados móvel externa.	Manter dados móveis.
48	Consulta de registros na base de dados móvel externa.	Manter dados móveis.
49	Transformação dos dados para a recuperação e exibição.	Disponibilizar dados móveis.
50	Transformação dos dados para a recuperação e transporte.	Disponibilizar dados móveis.
51	Transformação dos dados para o armazenamento apropriado.	Manter dados móveis.
52	Replicação automática da base de dados interna para a base de dados externa antes da realização dos processos de sincronização.	Segurança de dados móveis.
53	Deleção automática da base de dados interna para a base de dados externa depois da realização dos processos de sincronização.	Segurança de dados móveis.
54	Criação e preenchimento automático da base de dados interna dos dispositivos através de backup de base de dados externa.	Segurança de dados móveis.

(Continua)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
01	Validação de interface.	Segurança do servidor.
02	Validação de autenticação.	Segurança do servidor.
03	Validação de autorização.	Segurança do servidor.
04	Controlar fluxo para busca de dados.	Gerenciar serviços do servidor.
05	Controlar fluxo para listagem de dados.	Gerenciar serviços do servidor.
06	Controlar fluxo para exibição de dados.	Gerenciar serviços do servidor.
07	Controlar fluxo para criação de dados.	Gerenciar serviços do servidor.
08	Controlar fluxo para edição de dados.	Gerenciar serviços do servidor.
09	Controlar fluxo para deleção de dados.	Gerenciar serviços do servidor.
10	Controlar fluxo para desabilitar dados.	Gerenciar serviços do servidor.
11	Validação dos dados de criação.	Manter dados no servidor.
12	Validação dos dados de edição.	Manter dados no servidor.
13	Validação de registro existente.	Manter dados no servidor.
14	Gerenciamento de recursos em background.	Gerenciar recursos do servidor.
15	Redirecionamento de para outro servidor disponível.	Gerenciar recursos do servidor.
16	Identificação da estrutura de armazenamento de dados.	Manter dados remotos.
17	Permitir captura dos dados	Manter dados remotos.
18	Permitir validação dos dados	Manter dados remotos.
19	Permitir Transformação dos dados	Manter dados remotos / Permitir interoperabilidade.
20	Permitir verificação do nível de energia da bateria.	Gerenciar recursos remotos.
21	Permitir carregamento da bateria do módulo de coleta.	Gerenciar recursos remotos.
22	Permitir gerenciamento de funções de coleta e sincronização de acordo com o nível de energia da bateria.	Gerenciar recursos remotos.
23	Permitir gerencia de carregamento da bateria do módulo de coleta de acordo com o nível de energia da bateria.	Gerenciar recursos remotos.
24	Disponibilizar interfaces para listagem dedados.	Manter dados móveis.

(Conclusão)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
25	Disponibilizar interfaces para exibição dedados.	Manter dados móveis.
26	Disponibilizar interfaces para consulta dedados.	Manter dados móveis.
27	Disponibilizar interfaces para criação dedados.	Manter dados móveis.
28	Disponibilizar interfaces para edição dedados.	Manter dados móveis.
29	Disponibilizar interfaces para deleção dedados.	Manter dados móveis.
30	Disponibilizar interfaces para desabilitar dados.	Manter dados móveis.
31	Disponibilizar interfaces para reabilitação dedados.	Manter dados móveis.
32	Disponibilizar interfaces para importação dos dados.	Manter dados móveis.
33	Disponibilizar interfaces para exportação dos dados.	Manter dados móveis.
34	Disponibilizar interfaces para sincronização dos dados.	Manter dados móveis.
35	Permitir navegação entre as diferentes interfaces.	Gerenciar serviços móveis.
36	Controlar fluxo para busca de dados.	Gerenciar serviços móveis.
37	Controlar fluxo para listagem de dados.	Gerenciar serviços móveis.
38	Controlar fluxo para exibição de dados.	Gerenciar serviços móveis.
39	Controlar fluxo para criação de dados.	Gerenciar serviços móveis.
40	Controlar fluxo para edição de dados.	Gerenciar serviços móveis.
41	Controlar fluxo para deleção de dados.	Gerenciar serviços móveis.
42	Controlar fluxo para desabilitar dados.	Gerenciar serviços móveis.
43	Validação dos dados de criação coletados.	Manter dados móveis.
44	Validação dos dados de edição coletados.	Manter dados móveis.
45	Confirmação da deleção de dados coletados.	Manter dados móveis.
46	Confirmação da desabilitar dados coletados.	Manter dados móveis.
47	Permitir interpretação arquivos de formulários externos para objetos.	Permitir interoperabilidade.
48	Permitir gerenciamento de sinal.	Gerenciar recursos móveis.
49	Permitir verificação do nível de energia da bateria.	Gerenciar recursos móveis.
50	Permitir gerencia de funções de acordo com o nível de energia da bateria.	Gerenciar serviços móveis.

(Continua)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
01	Verificar existência e duplicidade do registro.	Integridade de dados no servidor Segurança de dados do servidor.
02	Replicar para os outros servidores disponíveis.	Disponibilizar dados no servidor Segurança de dados do servidor.
03	Enviar resposta de importação.	Comunicação.

(Conclusão)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
04	Enviar resposta de exportação.	Comunicação.
05	Enviar resposta de sincronização.	Comunicação.
06	Verificar existência de dados sincronizáveis.	Sincronização.
07	Permitir exportar dados individuais.	Exportação.
08	Permitir exportar dados em bloco.	Exportação.
09	Permitir requisição em individual.	Comunicação.
10	Permitir requisição em bloco.	Comunicação.
11	Permitir ajuste do bloco de acordo com o sinal.	Gerenciar serviços remotos.
12	Tratar respostas retornadas pelo servidor.	Comunicação.
13	Marcar registros que foram sincronizados.	Sincronização.
14	Reenviar dados que não puderam ser sincronizados.	Sincronização.
15	Permitir escolha de comunicação síncrona	Comunicação síncrona
16	Permitir escolha de comunicação assíncrona	Comunicação assíncrona
17	Verificar dados sincronizáveis.	Sincronização.
18	Verificar necessidade de sincronização.	Sincronização.
19	Permitir importar dados em bloco.	Importação.
20	Permitir importar dados individuais.	Importação.
21	Permitir exportar dados em bloco.	Exportação.
22	Permitir exportar dados individuais.	Exportação.
23	Permitir sincronizar dados do servidor e estação móvel.	Sincronização.
24	Permitir requisição em individual.	Comunicação.
25	Permitir requisição em bloco.	Comunicação.
26	Permitir ajuste do bloco de acordo com o sinal.	Gerenciar serviços móveis.
27	Tratar respostas retornadas pelo servidor.	Comunicação.
28	Reenviar dados que não puderam ser sincronizados.	Sincronização
29	Permitir escolha de comunicação síncrona	Comunicação síncrona
30	Permitir escolha de comunicação assíncrona	Comunicação assíncrona
31	Garantir a correta replicação dos dados para manter a integridade de dados entre o módulo de coleta e o módulo servidor.	Sincronização
32	Garantir a velocidade de comunicação com o servidor.	Sincronização
33	Recriar e popular a base após sincronização de dados.	Sincronização

(Continua)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
01	Garantir customização de respostas de erros.	Comunicação.

(Continua)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
02	Garantir resposta com o tipo de erro no processo de consulta.	Comunicação.
03	Garantir resposta com o tipo de erro no processo de listagem.	Comunicação.
04	Garantir resposta com o tipo de erro no processo de exibição.	Comunicação.
05	Garantir resposta com o tipo de erro no processo de criação.	Comunicação.
06	Garantir resposta com o tipo de erro no processo de edição.	Comunicação.
07	Garantir resposta com o tipo de erro no processo de importação.	Comunicação.
08	Garantir resposta com o tipo de erro no processo de exportação.	Comunicação.
09	Garantir resposta com o tipo de erro no processo de sincronização.	Comunicação.
10	Garantir resposta com o tipo de erro no processo de autenticação de interface.	Comunicação.
11	Garantir resposta com o tipo de erro no processo de autenticação de usuário.	Comunicação.
12	Garantir resposta com o tipo de erro no processo de autorização de usuário.	Comunicação.
13	Garantir processo de <i>rollback</i> caso algum erro de inserção múltipla ocorra.	Segurança de dados do servidor.
14	Garantir processo de <i>rollback</i> caso algum erro de inserção individual ocorra.	Segurança de dados do servidor.
15	Garantir processo de <i>rollback</i> caso algum erro de edição múltipla ocorra.	Segurança de dados do servidor.
16	Garantir processo de <i>rollback</i> caso algum erro de edição individual ocorra.	Segurança de dados do servidor.
17	Garantir proteção de serviços em caso de erro de autenticação de interface.	Segurança do servidor.
18	Garantir proteção de serviços em caso de erro de autenticação de usuário.	Segurança do servidor.
19	Garantir proteção de serviços em caso de erro de autorização de usuário.	Segurança do servidor.
20	Garantir o redirecionamento para outro servidor em funcionamento caso o servidor principal pare de funcionar.	Gerenciar recursos do servidor. Disponibilizar dados no servidor.

(Continua)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
21	Garantir a realização da coleta de dados novamente caso algum dado tenha sido captado de forma incorreta.	Manter dados remotos.
22	Garantir a realização do reenvio de dados novamente caso algum dado tenha sido enviado de forma incorreta.	Sincronização
23	Garantir a identificação de erro.	Gerenciar serviços remotos.
24	Garantir o processo correto a ser tomado de acordo com a identificação de erro.	Gerenciar serviços remotos.
25	Garantir a integridade do banco local caso algum erro de sincronização ocorra.	Segurança de dados remotos
26	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de inserção.	Segurança de dados remotos.
27	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de exportação.	Segurança de dados remotos.
28	Permitir exibição de erros na interface de criação de dados.	Comunicação.
29	Permitir exibição de erros na interface de edição de dados.	Comunicação.
30	Permitir a exibição dos erros na interface de importação de dados.	Comunicação.
31	Permitir a exibição dos erros na interface de exportação de dados.	Comunicação.
32	Permitir a exibição dos erros na interface de sincronização de dados.	Comunicação.
33	Permitir a exibição dos erros na interface de autenticação caso o usuário não exista no banco de dados remoto.	Comunicação.
34	Permitir a exibição dos erros na interface de autenticação caso o usuário não exista no banco de dados local.	Comunicação.
35	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro criação de registro.	Comunicação.
36	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro edição de registro.	Comunicação.
37	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro importação de registro.	Comunicação.
38	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro exportação de registro.	Comunicação.

(Conclusão)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
-----------	----------------------------------	------------------------------------

39	Permitir o detalhamento das entidades ou modelos que tiveram algum tipo de erro sincronização de registro.	Comunicação.
40	Garantir processo de <i>rollback</i> caso algum erro de inserção múltipla ocorra.	Segurança de dados móveis.
41	Garantir processo de <i>rollback</i> caso algum erro de inserção individual ocorra.	Segurança de dados móveis.
42	Garantir processo de <i>rollback</i> caso algum erro de edição múltipla ocorra.	Segurança de dados móveis.
43	Garantir processo de <i>rollback</i> caso algum erro de edição individual ocorra.	Segurança de dados móveis.
44	Garantira integridade do banco local caso algum erro de importação ocorra.	Segurança de dados móveis.
45	Garantir a integridade do banco local caso algum erro de exportação ocorra.	Segurança de dados móveis.
46	Garantir a integridade do banco local caso algum erro de sincronização ocorra.	Segurança de dados móveis.
47	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de inserção.	Segurança de dados móveis.
48	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de edição.	Segurança de dados móveis.
49	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de importação.	Segurança de dados móveis.
50	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de exportação.	Segurança de dados móveis.
51	Garantir a integridade do banco local caso a bateria do dispositivo termine durante o processo de sincronização.	Segurança de dados móveis.

(Continua)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
01	Permitir identificação de interfaces de coleta de dados.	Segurança do servidor.
02	Permitir identificação de interfaces de exibição de dados.	Segurança do servidor.
03	Permitir autenticação de usuários para acesso de serviços.	Segurança do servidor.
04	Permitir autorização de usuários para acesso de serviços.	Segurança do servidor.
05	Permitir autenticação de <i>endpoints</i> para acesso de serviços.	Segurança do servidor.

(Conclusão)

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
06	Permitir autorização de <i>endpoints</i> para acesso de serviços.	Segurança do servidor.

07	Permitir configuração para o acesso de usuários para acesso de serviços.	Gerenciar recursos do servidor.
08	Permitir configuração para o acesso de <i>endpoints</i> para acesso de serviços.	Gerenciar recursos do servidor.
09	Garantir que o usuário possa importar somente dados através de serviços permitidos.	Segurança do servidor.
10	Garantir que o usuário possa sincronizar somente dados através de serviços permitidos.	Segurança do servidor.
11	Garantir que os dados sejam acessados somente por um servidor específico.	Segurança remota.
12	Permitir autorização local de usuários para acesso de serviços.	Segurança móvel.
13	Permitir autenticação remota de usuários para acesso de serviços.	Segurança móvel.
14	Permitir a verificação do prazo da autenticação remota antes de importar dados.	Segurança móvel.
15	Permitir a verificação do prazo da autenticação remota antes de exportar dados.	Segurança móvel.
16	Permitir a verificação do prazo da autenticação remota antes de sincronizar dados.	Segurança móvel.

Nº	Requisito de sistema (RS)	Requisito arquitetural (RA)
01	Listar sincronizações.	Consumo dos dados.
02	Detalhar sincronizações.	Gerência dos dados.
03	Avaliar sincronizações.	Gerência dos dados.
04	Organizar sincronizações de diferentes formas.	Gerência dos dados.

APÊNDICE C - MAPEAMENTO DOS CONCEITOS DE DOMÍNIO

Requisito arquitetural (RA)	Conceitos de domínio (CD)
Segurança do servidor.	Segurança.
Segurança de dados do servidor.	Integridade.
Manter dados no servidor.	Manutenção de dados.
Disponibilizar dados do servidor.	Disponibilização.
Gerenciar recursos do servidor.	Gerencia do ambiente.
Gerenciar serviços do servidor.	Gerencia da aplicação.
Comunicação.	Comunicação.
Permitir interoperabilidade.	Comunicação.
Integridade de dados no servidor.	Integridade.
Segurança remota.	Segurança.
Segurança de dados remotos.	Integridade.
Manter dados remotos.	Manutenção de dados.
Permitir interoperabilidade.	Comunicação.
Gerenciar recursos remotos.	Gerencia do ambiente.
Gerenciar serviços remotos.	Gerencia da aplicação.
Comunicação.	Comunicação.
Comunicação síncrona	Comunicação.
Comunicação assíncrona	Comunicação.
Sincronização.	Sincronização.
Exportação.	Sincronização.
Segurança móvel.	Segurança.
Segurança de dados móveis.	Integridade.
Manter dados móveis.	Manutenção de dados.
Disponibilizar dados móveis.	Disponibilização.
Gerenciar recursos móveis.	Gerencia do ambiente.
Gerenciar serviços móveis.	Gerencia da aplicação.
Permitir interoperabilidade.	Comunicação.
Integridade de dados móveis.	Integridade.
Comunicação.	Comunicação.
Comunicação síncrona	Comunicação.
Comunicação assíncrona	Comunicação.
Sincronização.	Sincronização.
Importação.	Sincronização.
Exportação.	Sincronização.
Consumo dos dados.	Utilização.
Gerência dos dados.	Gerencia da aplicação.

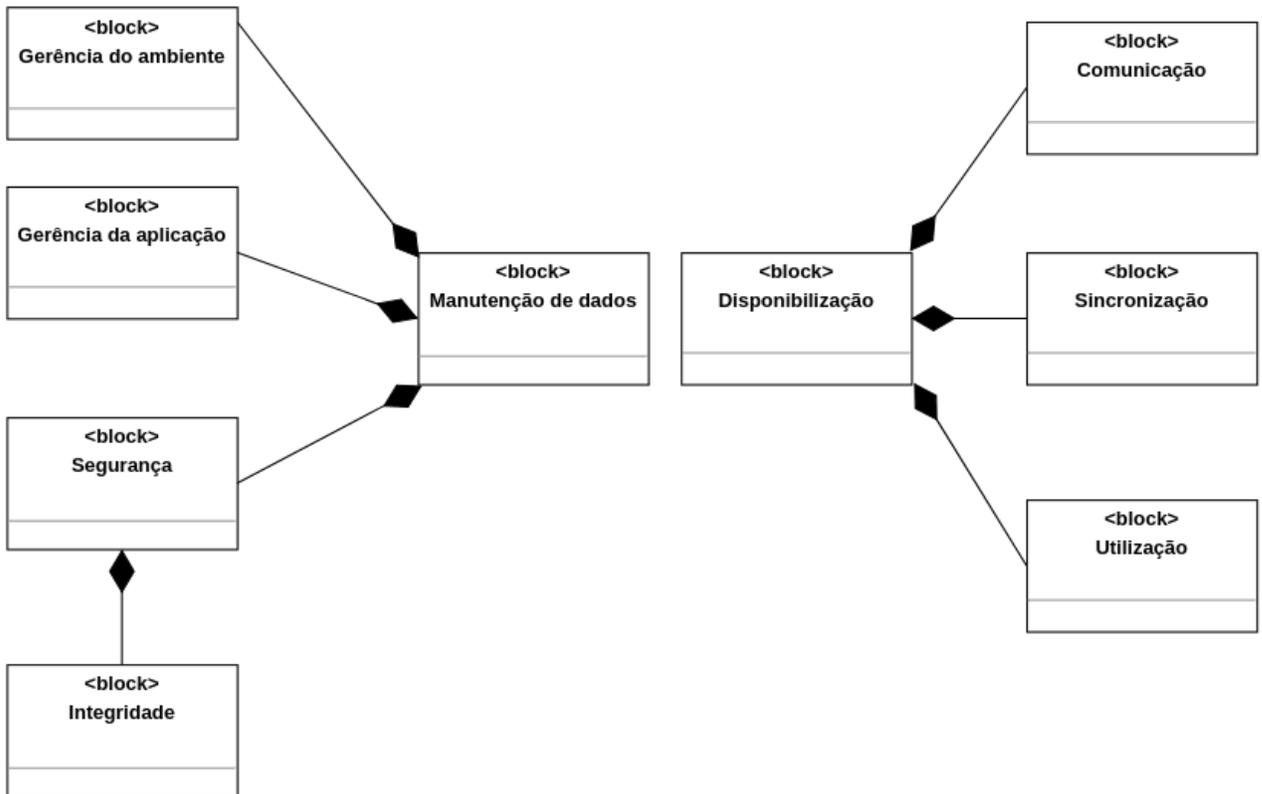
APÊNDICE D – VISÃO CONCEITUAL

(Continua)

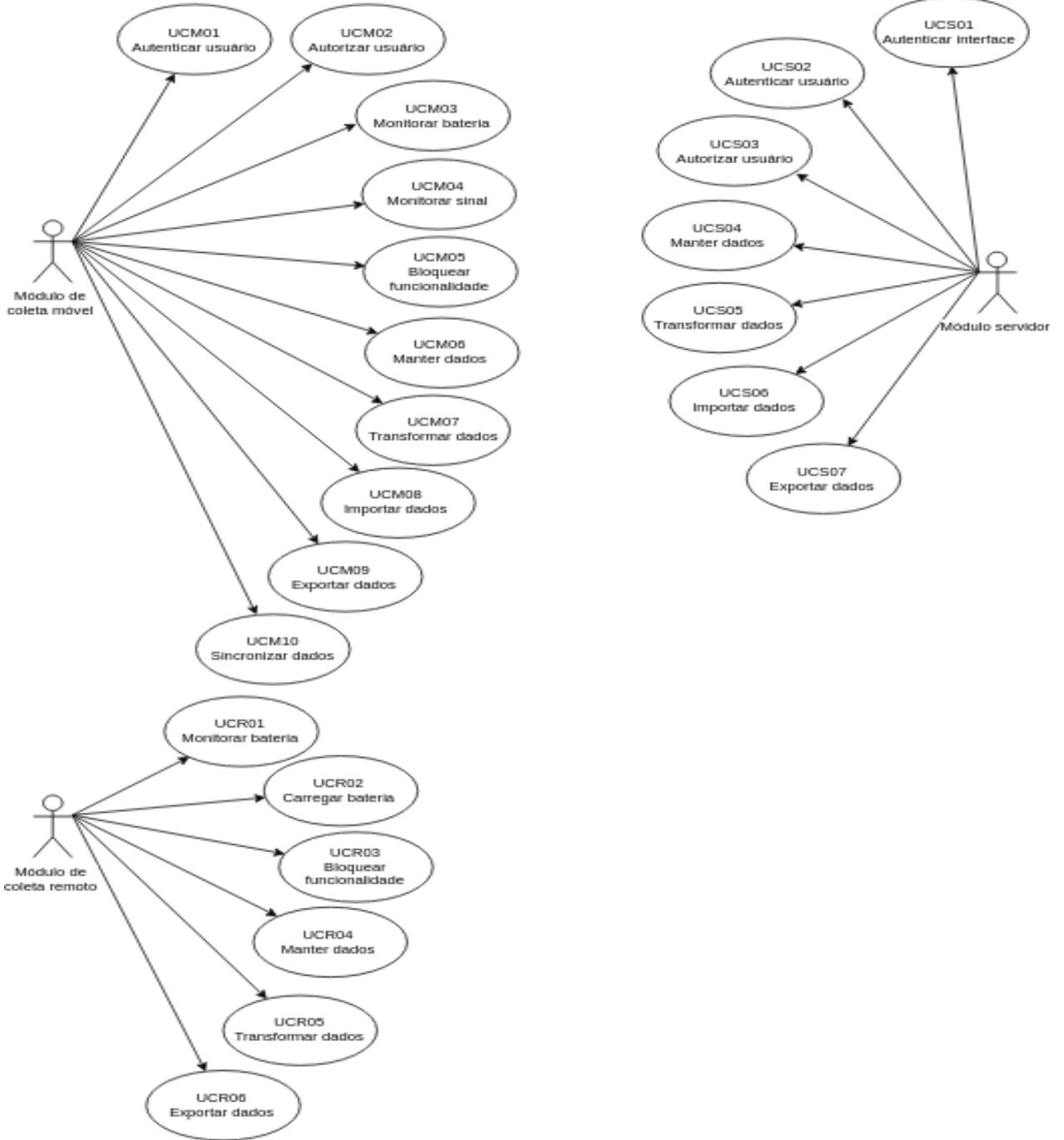
Conceitos de domínio (CD)	Significado
Segurança	<p>Conceito relacionado com o correto fornecimento de serviços oferecidos pelos subsistemas que compõem o sistema de coleta de dados para usuários e sistemas com a correta permissão para utilizá-los.</p> <p>Este conceito também se relaciona com a proteção dos dados que transitam pelo sistema de coleta de dados no que tange a manutenção, comunicação e sincronização.</p>
Integridade	<p>Conceito que visa proporcionar a estabilidade dos dados coletados sob a perspectiva do correto armazenamento edição deleção e desabilitar dados.</p>
Manutenção de dados	<p>Conceito que significa a realização dos serviços de criação, consulta, edição e deleção.</p>
Disponibilização	<p>Conceito que relacionado com a capacidade de acessar e obter as informações e/ou serviços independentemente dos problemas e/ou erros que possam acontecer.</p> <p>Este conceito também engloba a capacidade de fornecer as informações nos formatos corretos de acordo com sua necessidade.</p> <p>Além disto, este conceito também pode ser interpretado como a manifestação de características que não se configuram como requisitos sistêmicos ou arquiteturas. Por exemplo, o número de dispositivos móveis levados para coleta em campo.</p>
Gerencia do ambiente	<p>Capacidade de organizar, gerenciar e utilizar recursos e funções disponíveis nos ambientes que provêm o sistema, proporcionando o seu correto funcionamento.</p>
Gerencia da aplicação	<p>Capacidade de organizar, gerenciar e utilizar serviços disponíveis nos subsistemas, proporcionando o seu correto funcionamento.</p> <p>Além disto, este conceito que engloba o caráter administrativo do sistema. Voltado para a análise dos dados coletados.</p>
Comunicação	<p>Conceito que engloba os aspectos exibição de mensagens, seleção de meios, formas e padrões de comunicação, e detalhamento e tratamento de erros de comunicação.</p>

(Conclusão)

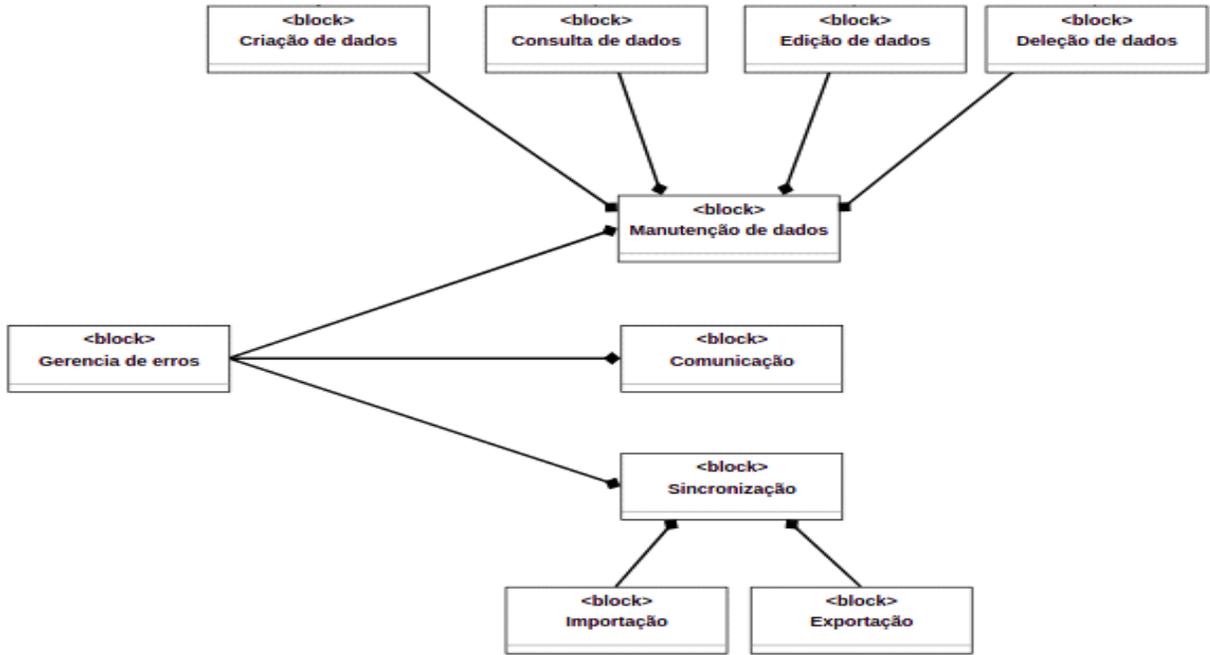
Sincronização	<p>Conceito que engloba aspectos necessários para o correto funcionamento entre os sistemas coletores de informação e o servidor principal.</p> <p>Este conceito engloba funções como a importação, exportação e replicação dos dados. Além disto, este conceito também é responsável pelo gerenciamento do tipo e processo de sincronização.</p>
Utilização	<p>Conceito que se refere ao consumo dos dados que foram coletados.</p>



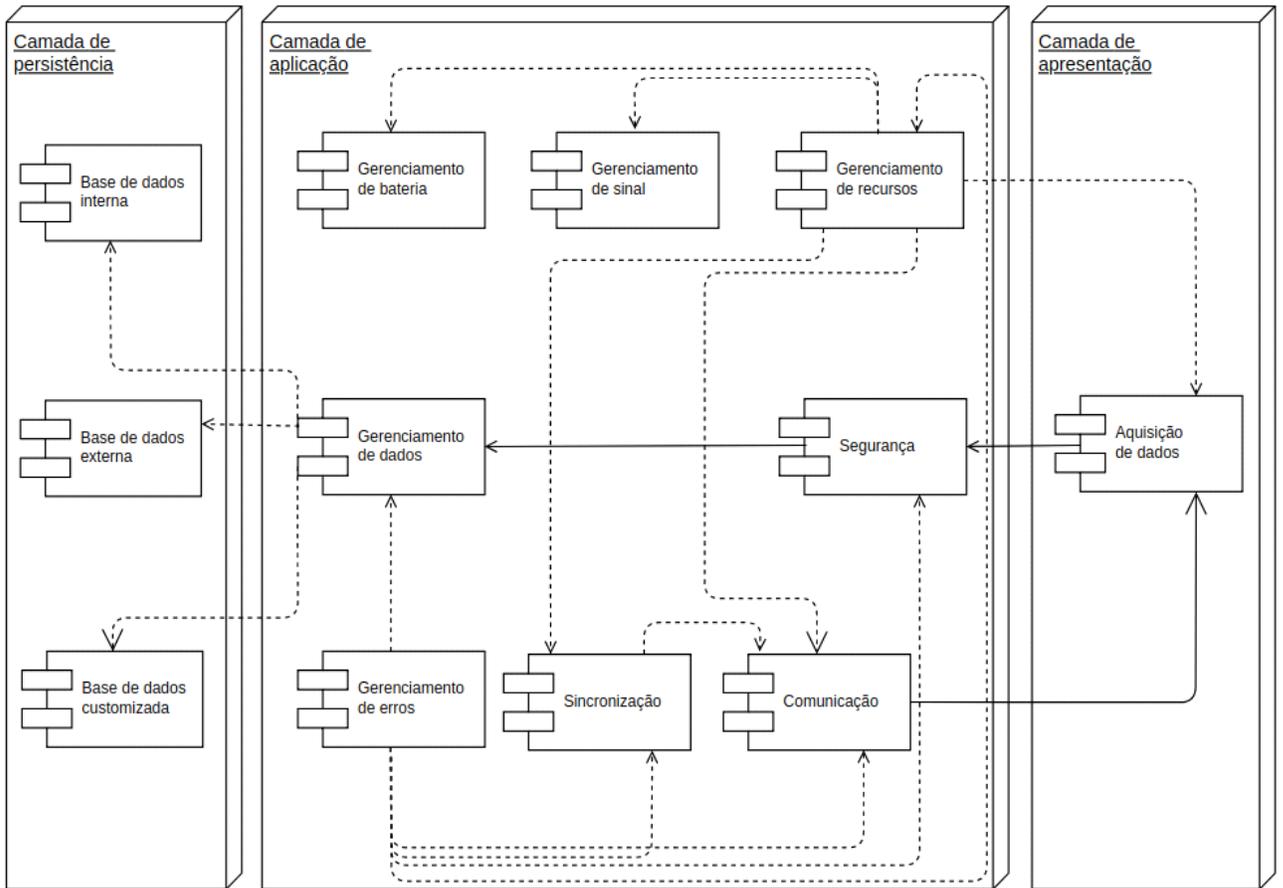
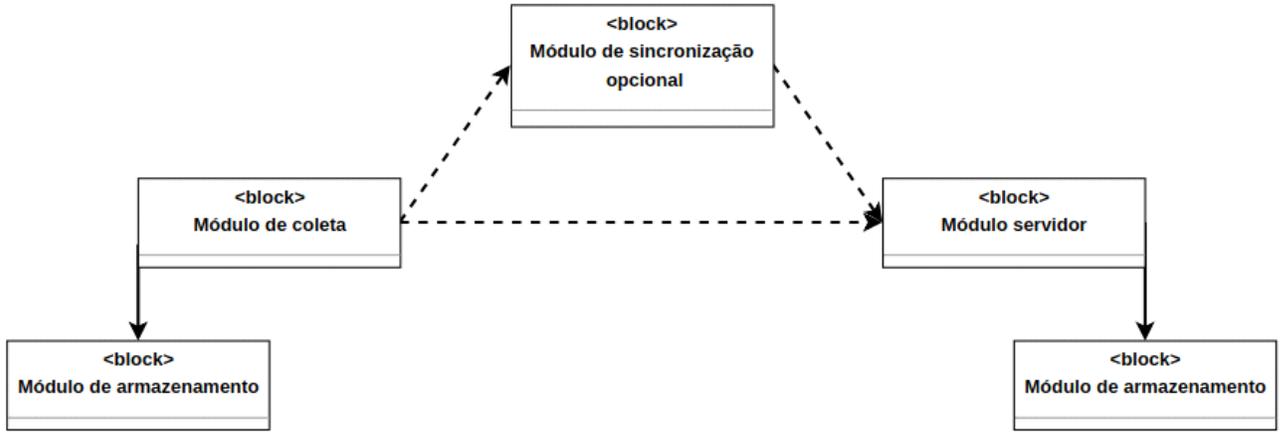
APÊNDICE E – VISÃO DE FUNCIONALIDADE



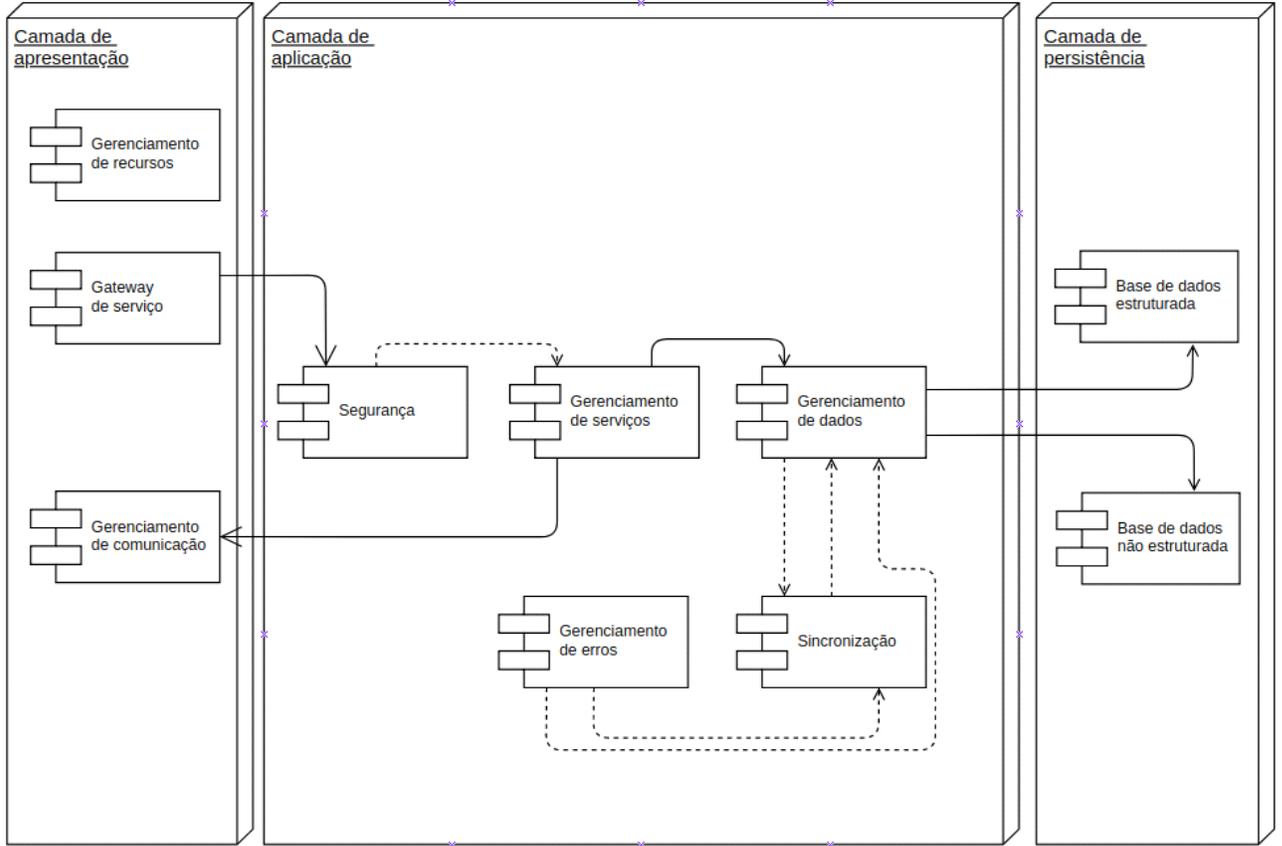
APÊNDICE F – VISÃO DE SERVIÇOS DE SISTEMA

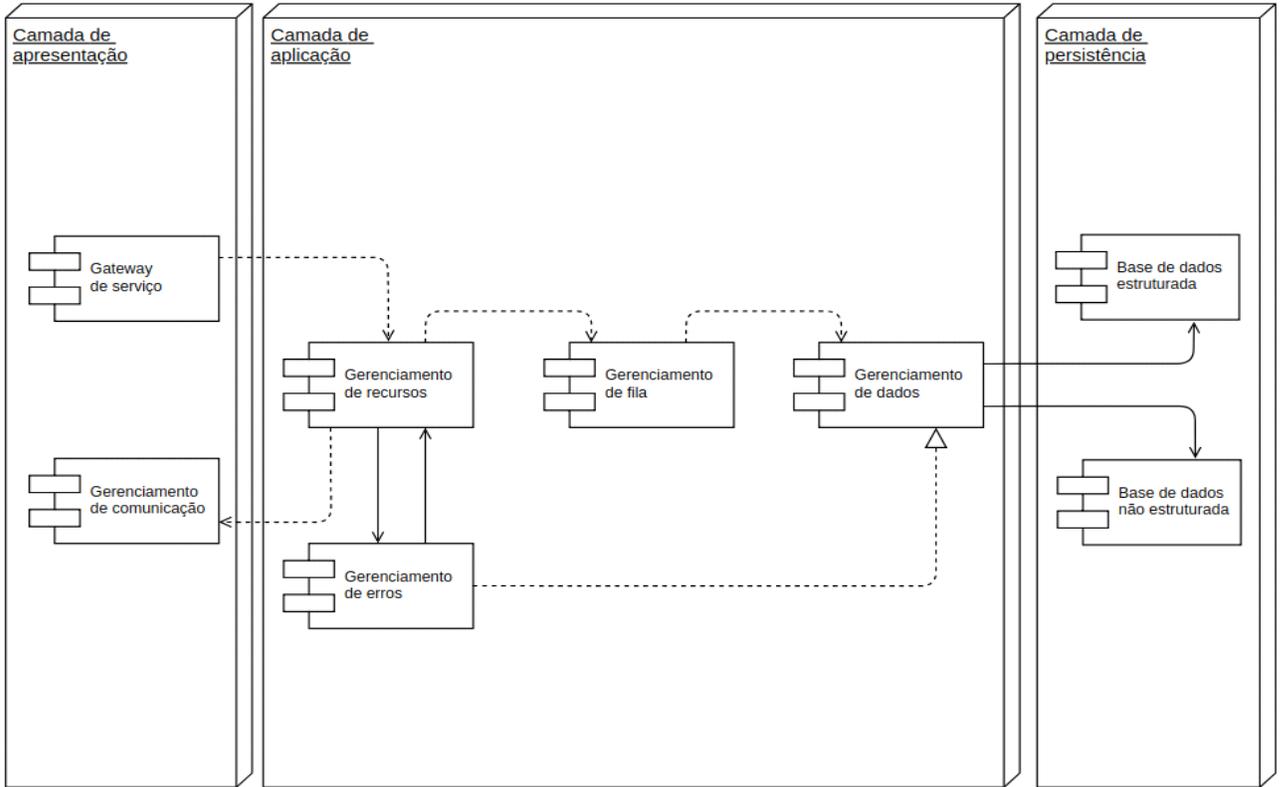


APÊNDICE G – VISÃO VARIABILIDADE

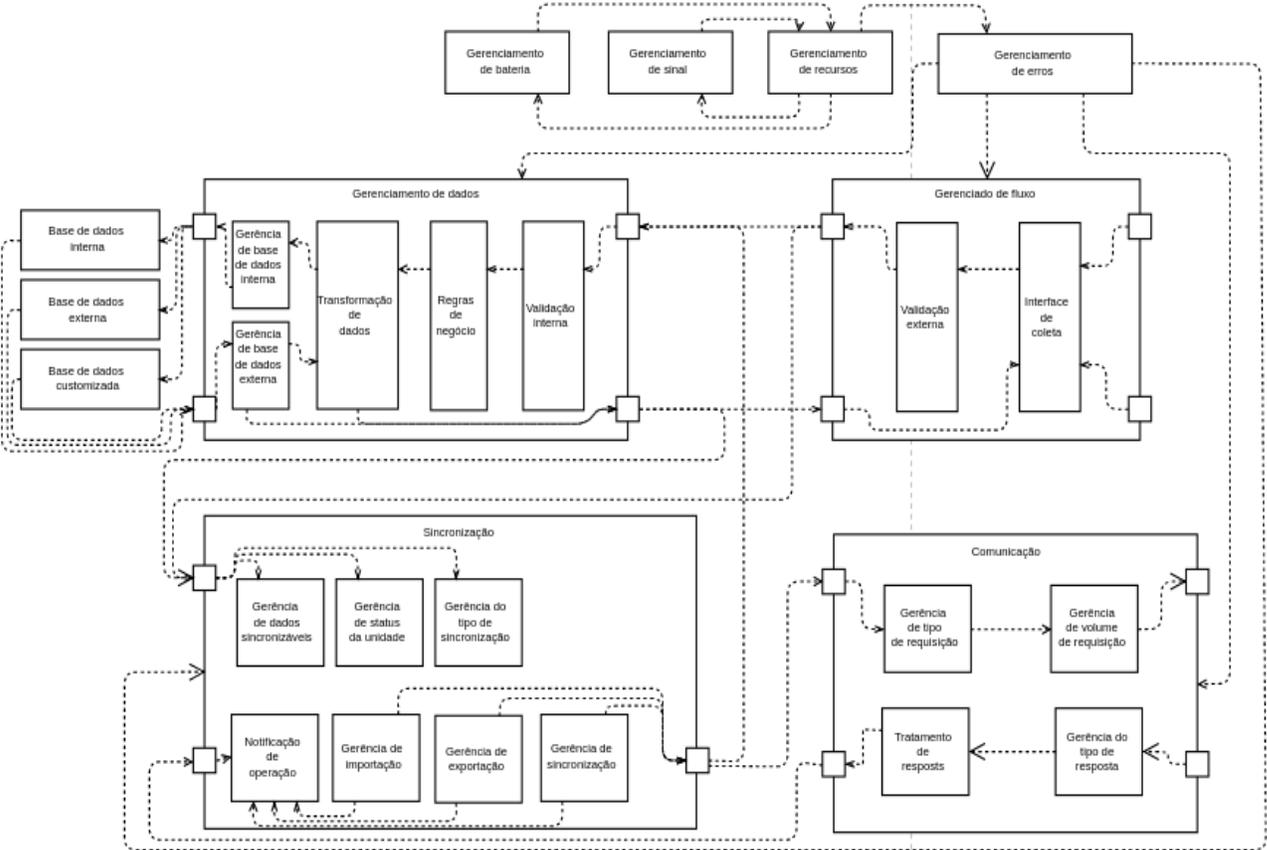


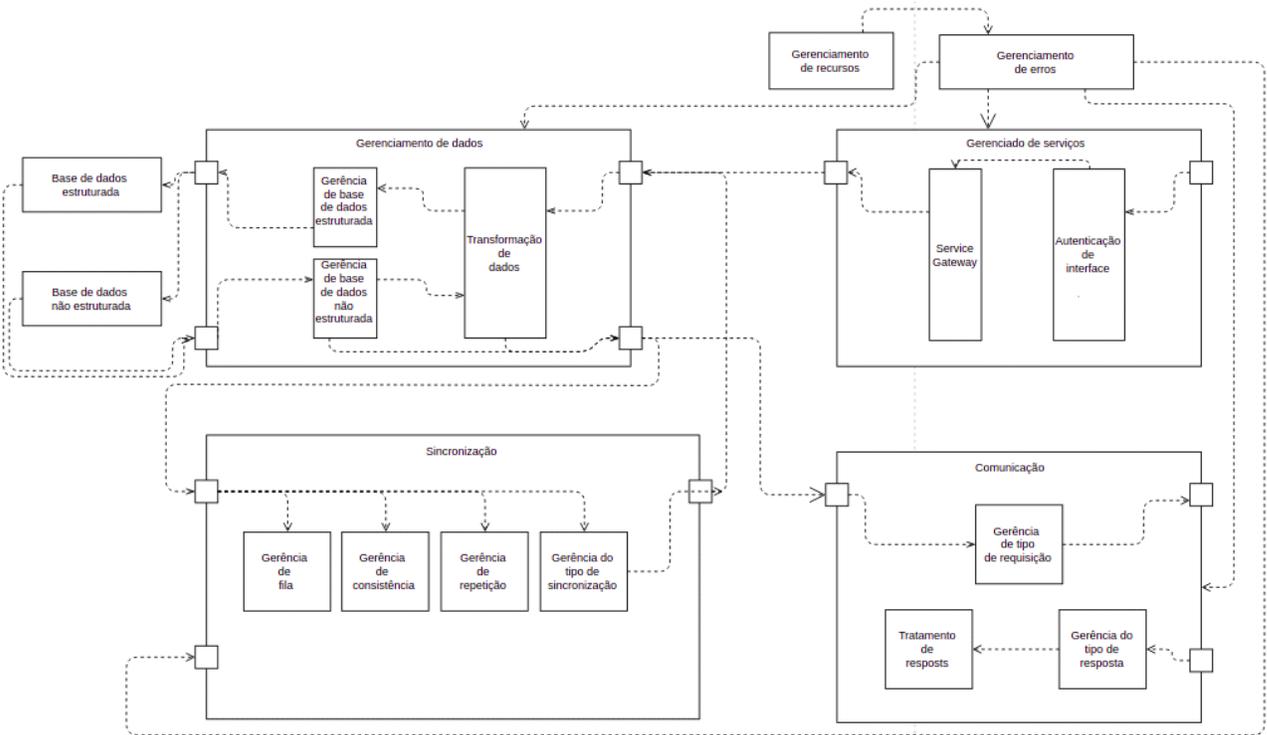
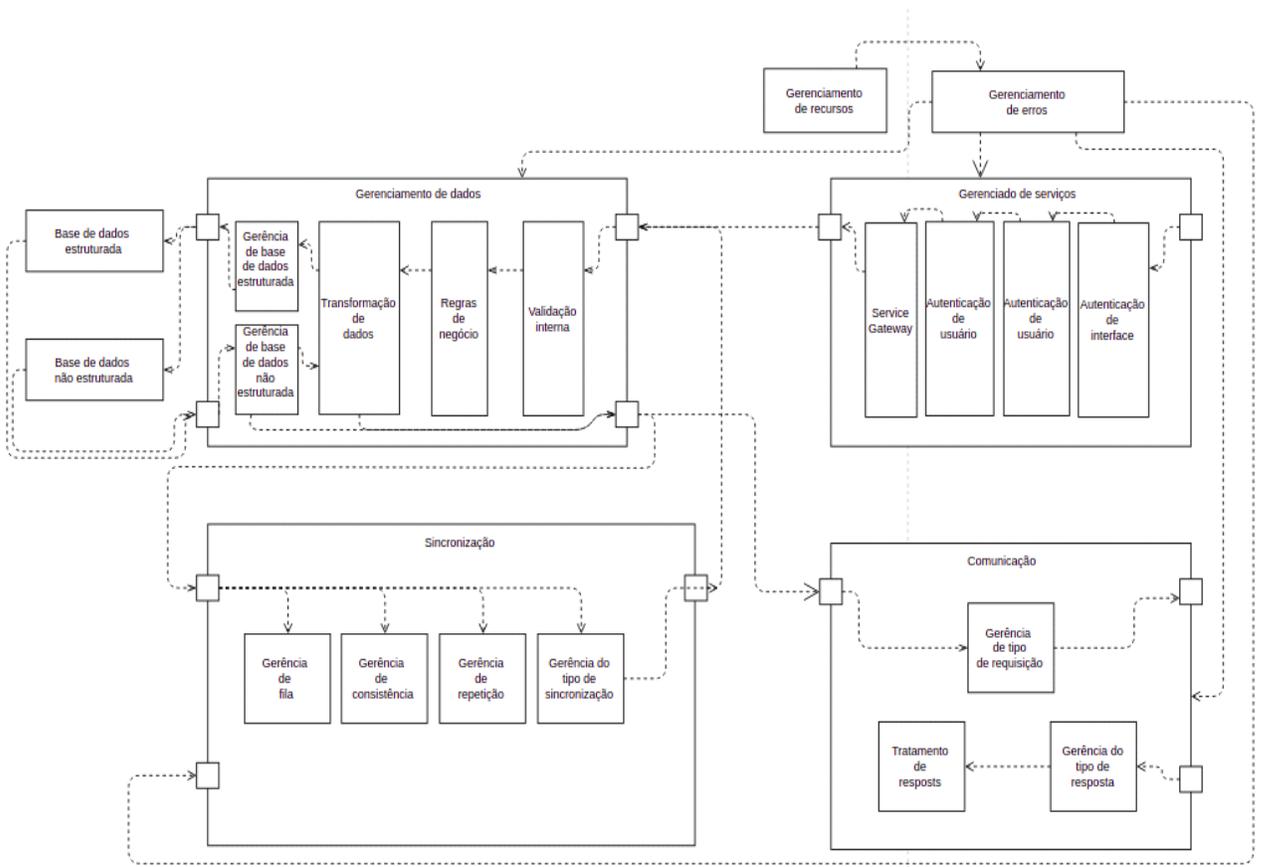
APÊNDICE H – VISÃO COMPONENTES COLABORATIVOS



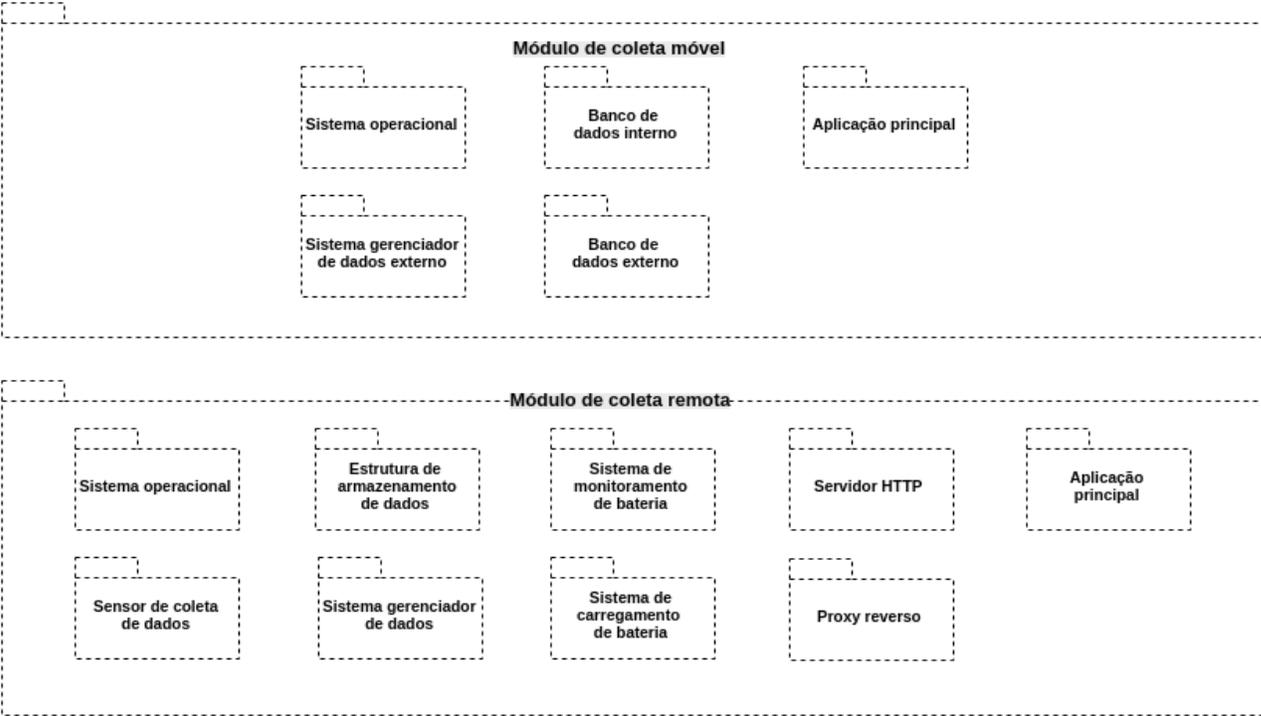


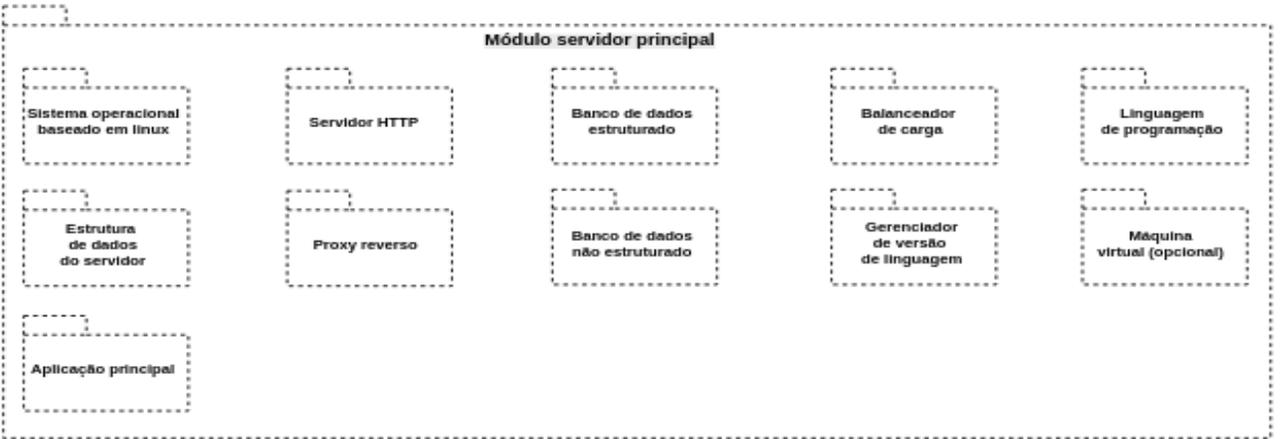
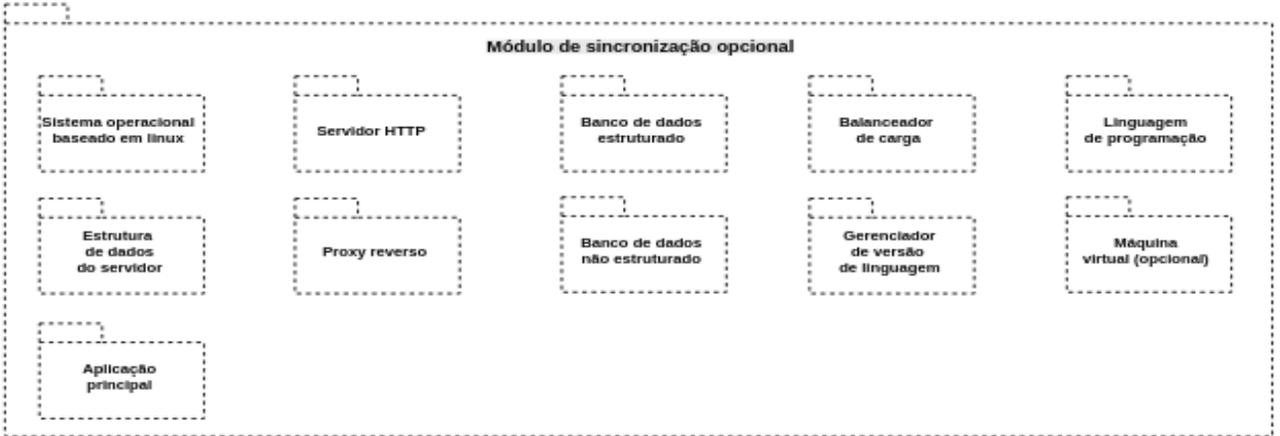
APÊNDICE I – VISÃO PROCESSO



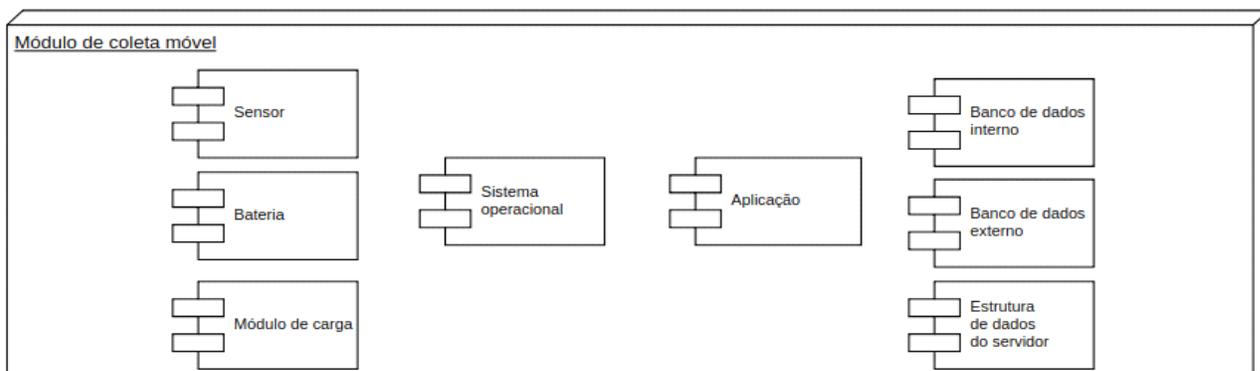
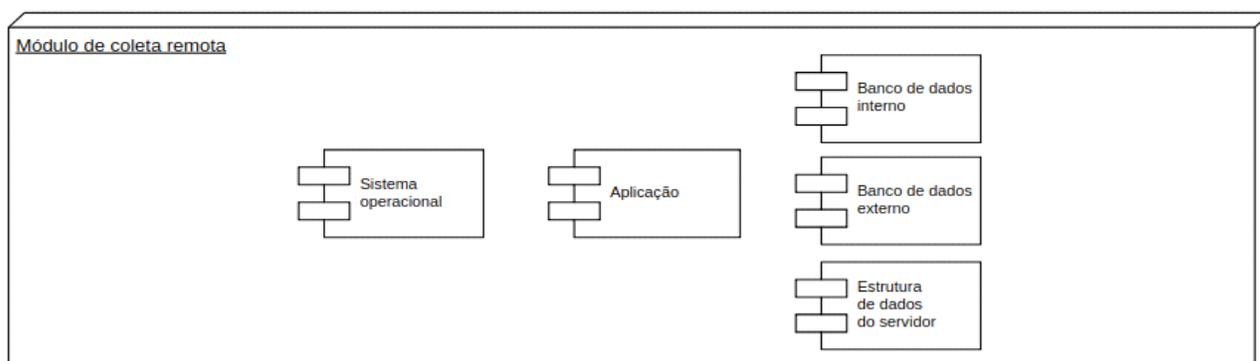


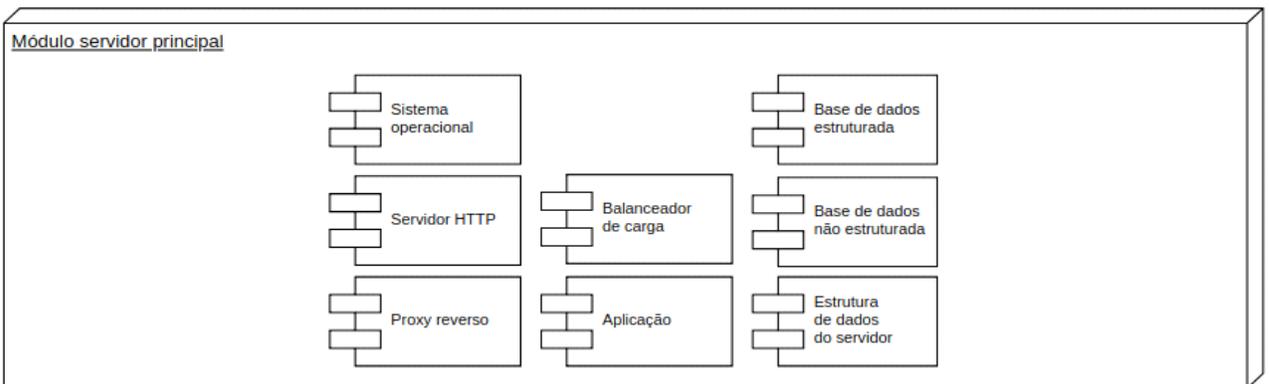
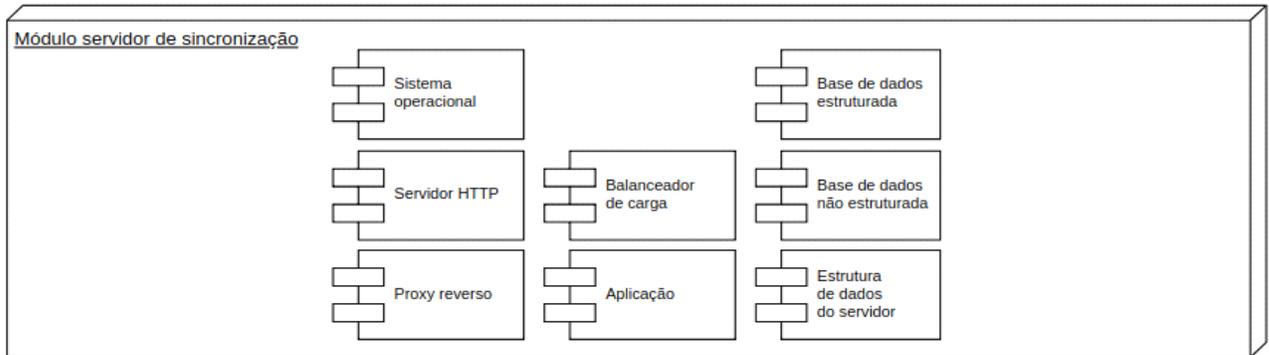
APÊNDICE J – VISÃO IMPLANTAÇÃO



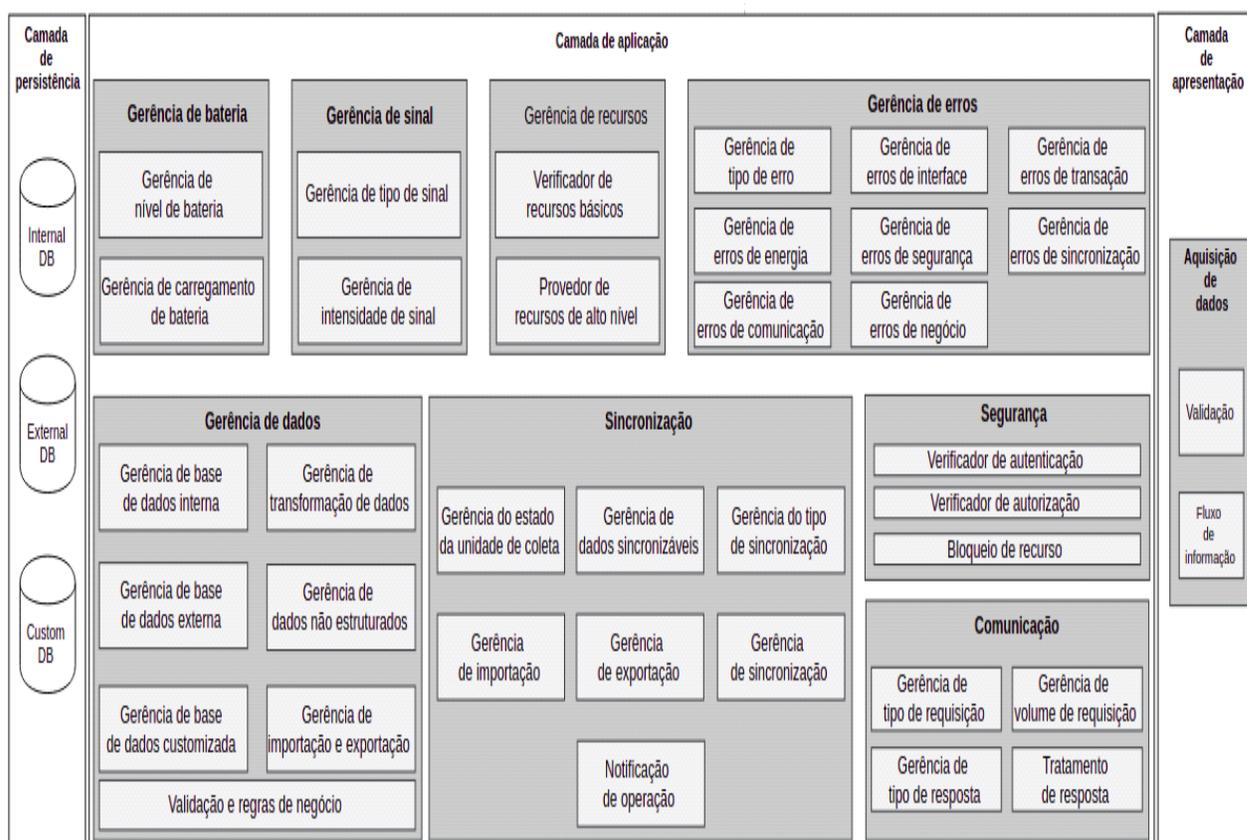


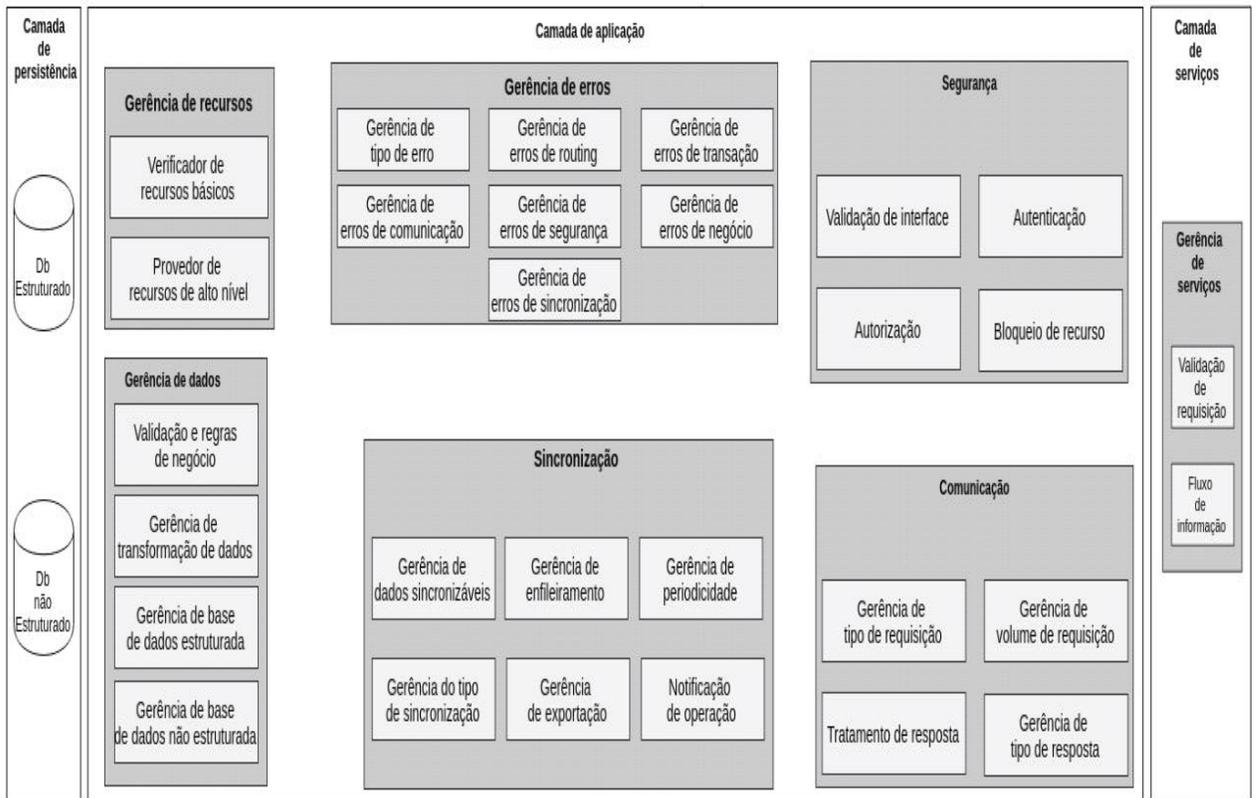
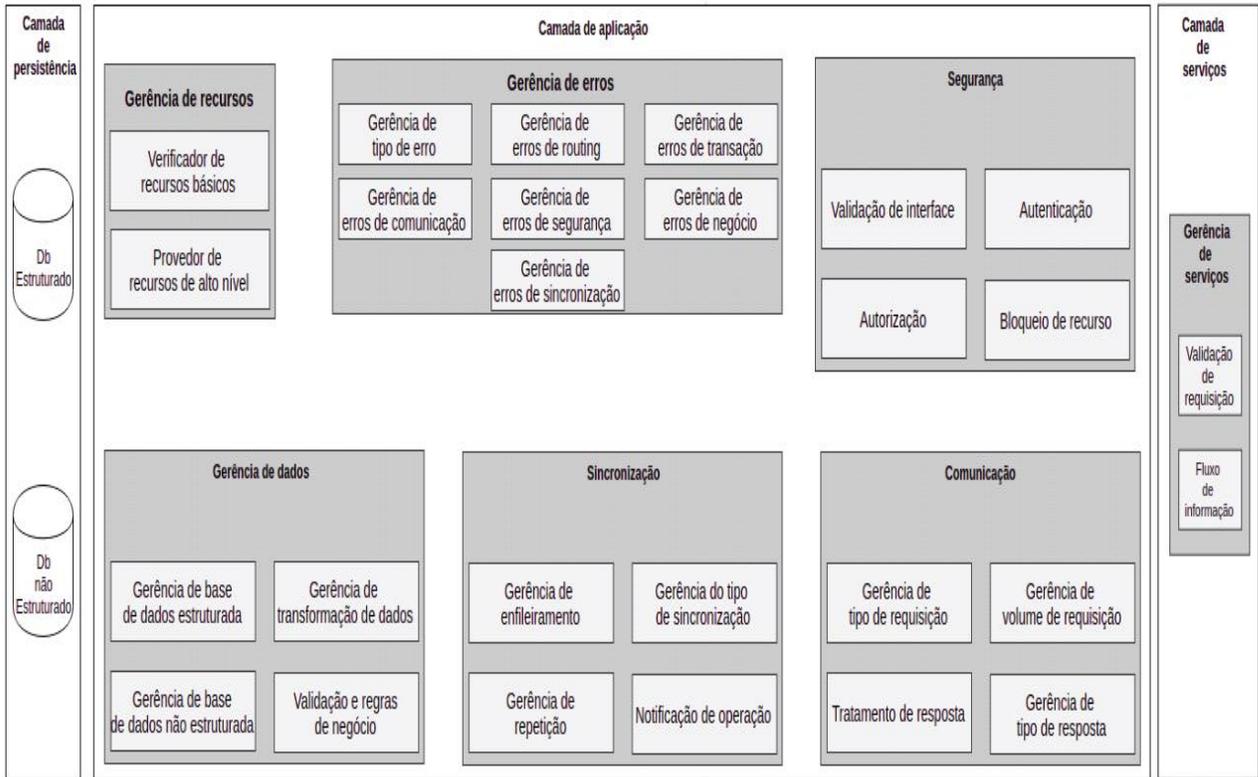
APÊNDICE K – VISÃO DESENVOLVIMENTO



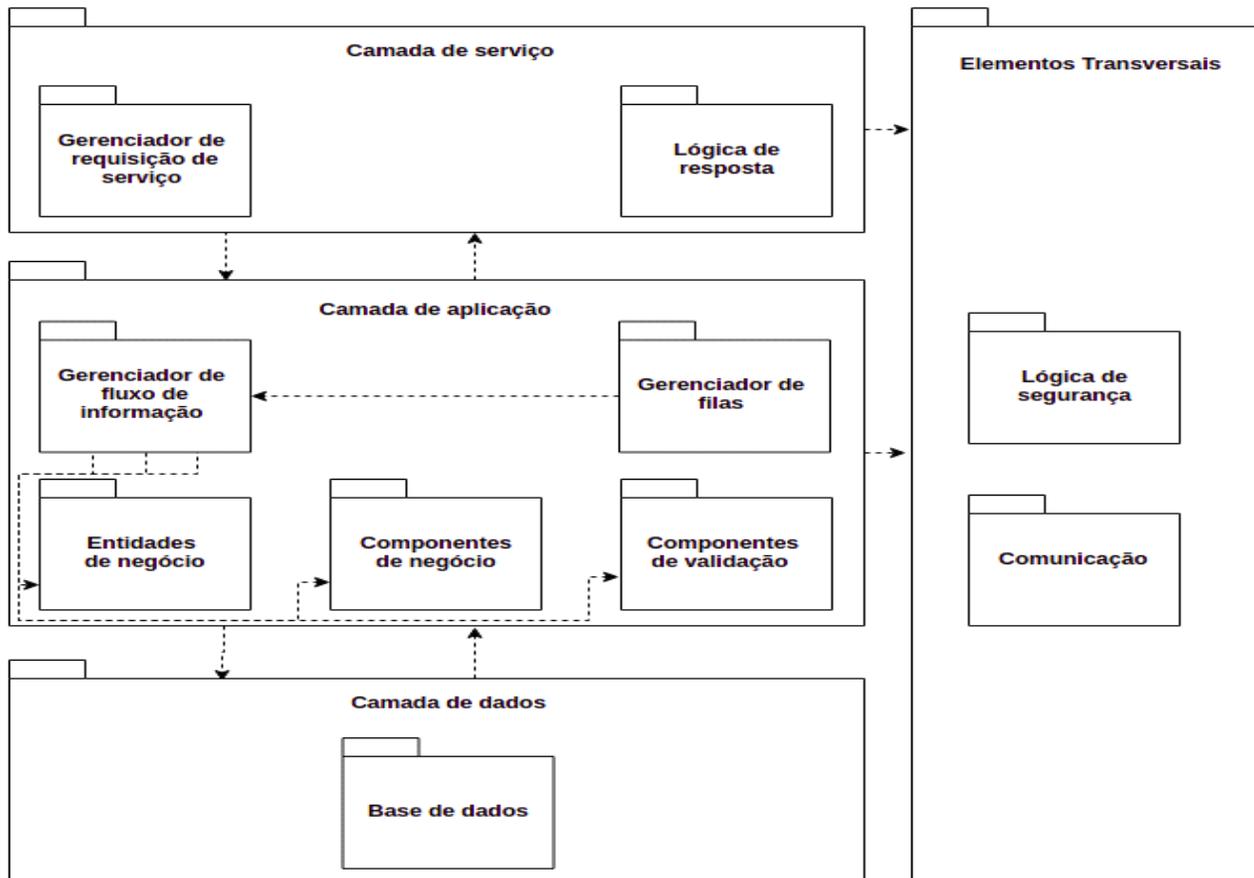


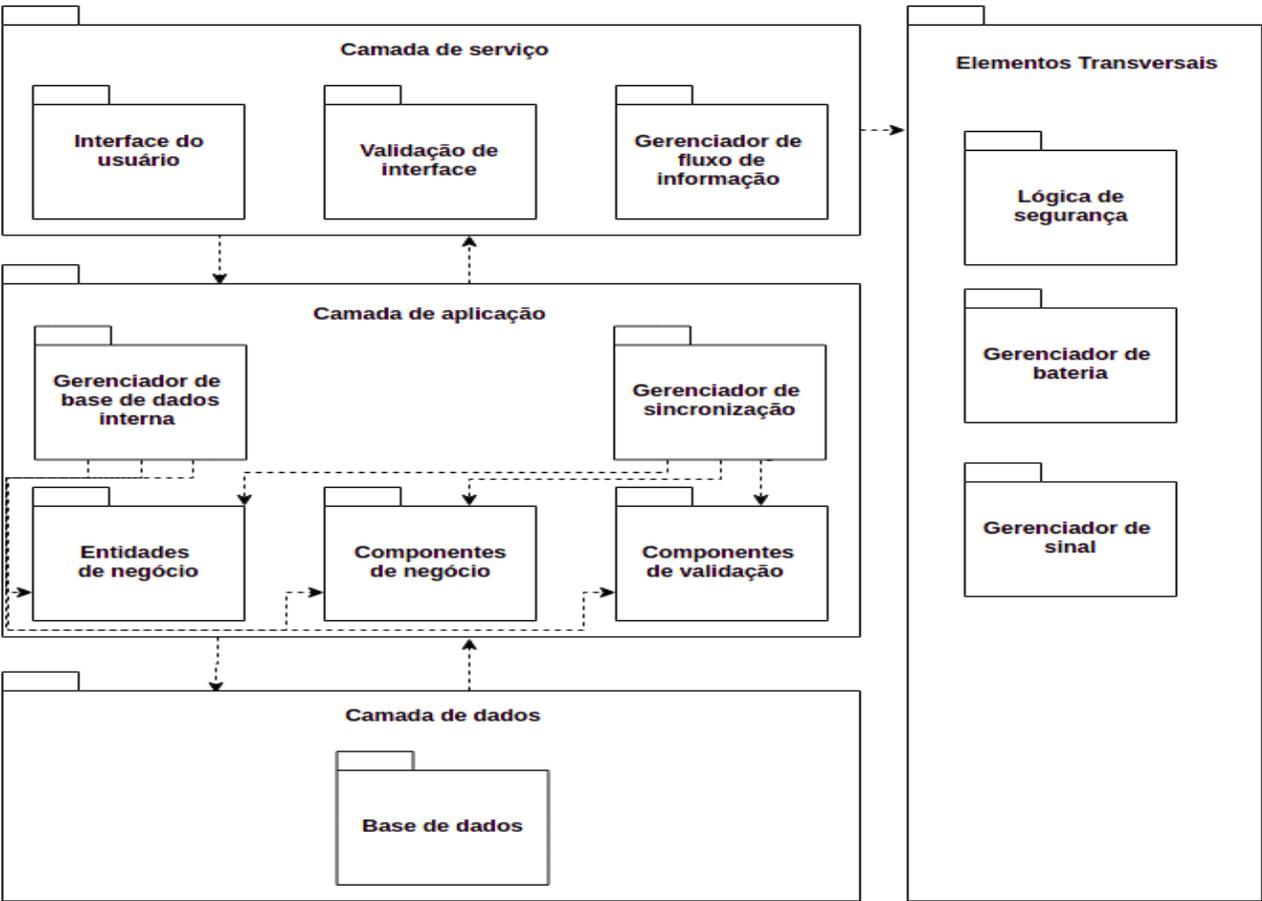
APÊNDICE L – ARQUITETURA DE REFERÊNCIA OSTARA



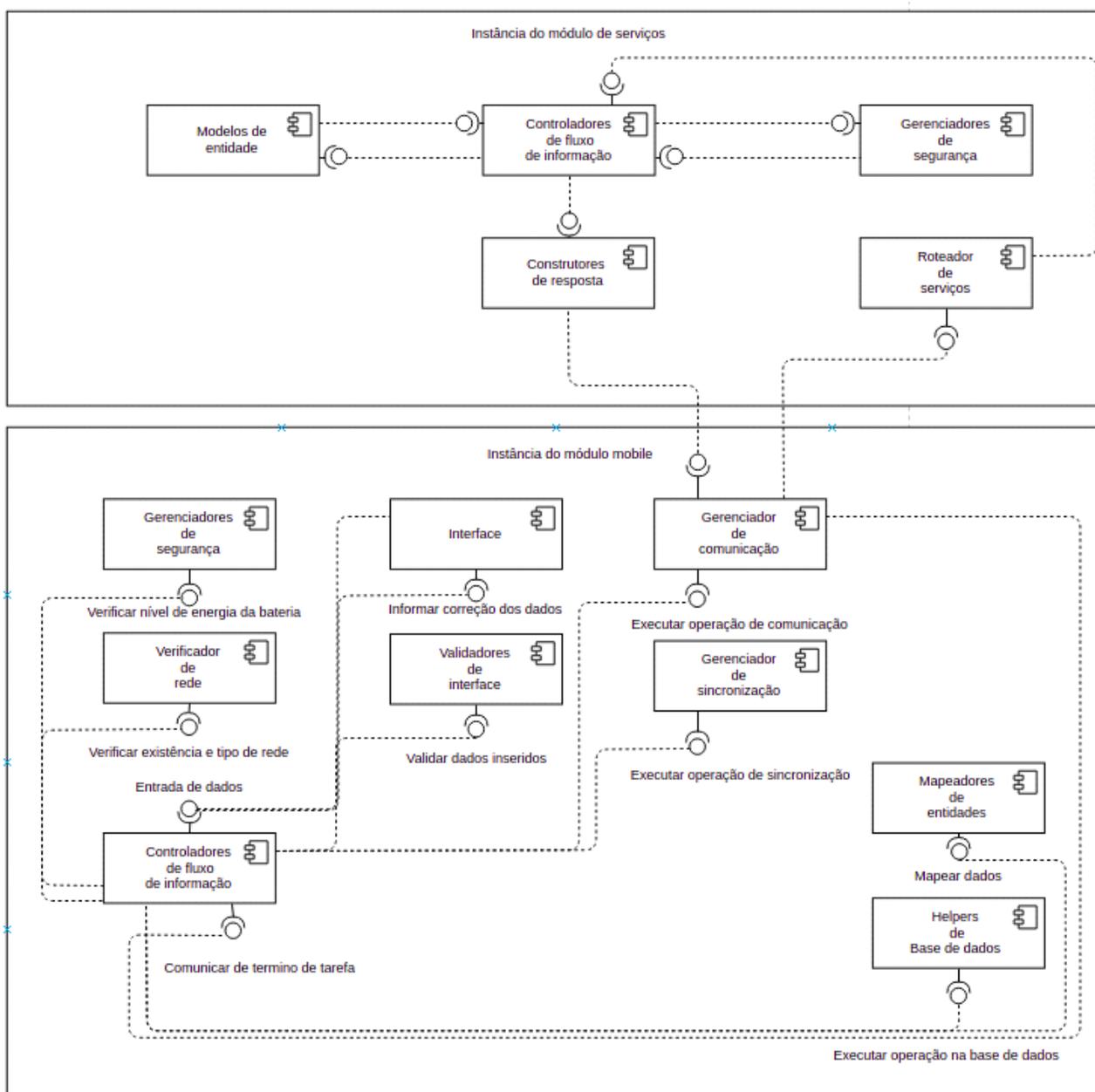


APÊNDICE M – INSTÂNCIA DA AR OSTARA - DIAGRAMA DE PACOTES





APÊNDICE N – INSTÂNCIA DA AR OSTARA - DIAGRAMA DE COMPONENTES



APÊNDICE O – TERMO DE LIVRE CONSENTIMENTO LIVRE E ESCLARECIDO**Consentimento da participação da pessoa como sujeito da pesquisa**

Eu, _____, detentor do RG de nº _____, concordo em ser participante voluntário do estudo de avaliação do sistema de coleta de dados proposto para utilização organizacional interna, fruto da pesquisa de arquiteturas de referências que tangem o modelo de domínio de coleta de dados, conduzida pelo aluno de mestrado Marcos Felipe Ferreira Amazonas de Souza, sob orientação da Prof.^a Dra. Viviane Almeida dos Santos.

Sob a responsabilidade do pesquisador _____, informo que fui devidamente informado e esclarecido sobre a pesquisa, os procedimentos nela envolvidos, assim como os possíveis riscos e benefícios decorrentes de minha participação.

Foi-me garantido que posso retirar meu consentimento a qualquer momento, sem que isto leve à qualquer penalidade.

Pesquisador:

Nome: _____ Assinatura: _____

Participante:

Nome: _____ Assinatura: _____

Testemunhas não ligadas à equipe de pesquisadores:

Nome: _____ Assinatura: _____

Nome: _____ Assinatura: _____

Tucuruí, __/__/2019.

APÊNDICE P - FORMULÁRIO AVALIATIVO

Questionário

Obrigado por disponibilizar uma porção de seu tempo para preencher este questionário.

Por favor, responda de forma sincera e crítica.

1 Informações Pessoais

Nome:

Idade:

Sexo: Masculino Feminino

Qual a sua ocupação:

Com que frequência você utiliza dispositivos móveis?

Nunca Ocasionalmente Frequentemente (Pelo menos uma vez por dia)

Você participa da operação de coleta de dados da organização?

Nunca Ocasionalmente Frequentemente (Pelo menos uma vez por dia)

2 Perguntas objetivas:

Sendo o Número 1 **Discordo totalmente**, e o número 7 significa **Concordo totalmente** responda as questões abaixo.

Fui capaz de realizar o login na aplicação.

1 2 3 4 5 6 7

Fui capaz de realizar a manutenção de matrizes.

1 2 3 4 5 6 7

Fui capaz de realizar a manutenção de fenologias anuais.

1 2 3 4 5 6 7

Fui capaz de realizar a manutenção de fenologias quinzenais.

1 2 3 4 5 6 7

Fui capaz de realizar a sincronização dos dados criados no aparelho móvel.

1 2 3 4 5 6 7

O sistema reflete o processo de coleta vivenciado na organização.

1 2 3 4 5 6 7

O sistema simplificaria a realização das tarefas internas e recorrentes na empresa.

1 2 3 4 5 6 7

O sistema aceleraria a realização das tarefas internas e recorrentes na empresa.

1 2 3 4 5 6 7

O sistema traria vantagens para a operação mantida pela organização.

1 2 3 4 5 6 7

O sistema poderia ser adaptado para outros contextos de coleta de dados mantidos pela organização.

1 2 3 4 5 6 7

3 Perguntas subjetivas

1. Em que sentido você acha que este sistema de coleta de dados melhora ou piora a execução de suas tarefas?

2. Você tem algum outro comentário relacionado ao sistema, ou algo relacionado?