CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MIDDLEWARE PARA FUSÃO DE SENSORES: ESTUDO DE CASO COM GPS

OTÁVIO GUILHERME ARRUDA DOS SANTOS

MARÍLIA

2013

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MIDDLEWARE PARA FUSÃO DE SENSORES: ESTUDO DE CASO COM GPS

Monografia apresentada ao Centro Universitário Eurípides de Marília como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador:

Prof. Dr. Fábio Dacêncio Pereira

MARÍLIA

2013

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Otavio Guilherme Arruda dos Santos

Middleware de Integração de Sensores para Sistemas Ciberfísicos

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 10 (Dez)

Orientador: Fábio Dacêncio Pereira

1°. Examinador: Paulo Augusto Nardi

2°. Examinador: Leonardo Castro Botega

Marília, 04 de dezembro de 2013.

ERRATA

DOCUMENTO DE APROVAÇÃO

Onde se lê:

MIDDLEWARE DE INTEGRAÇÃO DE SENSORES PARA SISTEMAS CIBERFÍSICOS

Deve-se Ler:

MIDDLEWARE PARA FUSÃO DE SENSORES: ESTUDO DE CASO COM GPS

AGRADECIMENTOS

Primeiramente agradeço a Deus, por todas as bênçãos e por ser o responsável por todas as portas que se abriram na minha vida.

Agradeço à toda minha família. Meu pai Flávio e minha mãe Eunice que me de uma maneira única! Grandemente me apoiaram e me incentivaram nos estudos, pelos concelhos e que tiveram grande paciência e compreensão nos dias que fiquei nos laboratórios e q ia pra casa praticamente pra comer e dormir (tá... isso aconteceu praticamente todos os dias!!). Mãe eu amo a senhora, pai eu amo o senhor!

À minha vó Guiomar que também me criou com seu amor e dedicação e com tanto carinho. Também agradeço à minha vó Cida.

Às minhas tias Genny e Marilene que sempre estiveram presentes na minha vida, me ensinaram a ler e a escrever, participaram de toda minha formação e criação. Tia Marilene em especial, que deixou saudades em meu coração, mas que sua risada, sua alegria, seus concelhos e seu amor permanecerá eternamente!

Agradeço a minha irmã Lígia que me atura todos esses anos. Suporta minhas brincadeiras e sempre me ajuda.

À minha namorada Suellen, meu grande amor, presente de Deus na minha vida que esteve ao meu lado em todos os momentos, me ajudando e me auxiliando em tudo! É o Sol que aquece minha vida, que me preenche de felicidade e cor.

Ao Éttore meu grande amigo que me socorreu nas dúvidas, pelas conversas e grande companheirismo.

Ao Rafael (Pudim) pela amizade nestes quatro anos.

Ao meu orientador, Prof. Dr. Fábio Dacêncio Pereira, pelo incentivo e paciência.

Ao oráculo, Prof. Rodolfo.

Ao mestre dos mestres, Prof. Bugatti.

Agradeço ao inventor do café, líquido precioso!

Às tias que fazem o café para os laboratórios.

À todos os funcionários do UNIVEM.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

S.C.F ..... SISTEMAS CIBERFÍSICOS

GPS ..... GLOBAL POSITIONING SYSTEM

A.B.S..... ANTI-LOCK BRAKING SYSTEM

XML..... EXTENSIBLE MARKUP LANGUAGE

NMEA..... NATIONAL MARINE ELECTRONICS ASSOCIATION

COMPSI .. COMPUTING INFORMATION SYSTEMS RESEARCH LAB

ASCII ..... AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE

RESUMO

Com a popularização de computadores e dispositivos eletrônicos, a computação está cada vez mais presente e tomando grande importância para realização de tarefas. Os Sistemas Cíber-Físico (S.C.F.) surgiram como um meio de integrar sistemas computacionais com os processos físicos, integrando e interagindo o mundo real com o mundo virtual através da fusão de conceitos como computação, comunicação e controle. Neste cenário, os processos físicos exercem grande influência nos S.C.F. exigindo que este tipo de sistema tenha uma rede de diversos tipos de sensores para a percepção do meio físico. A integração de diversos sensores e o desenvolvimento de rotinas para facilitar a aquisição de informações e atuação no meio físico são alguns dos desafios encontrados neste cenário, uma vez que existem diversos tipos de sensores, fabricantes e modos de configuração conforme propósito. Visando mitigar estas dificuldades, propõem-se neste trabalho a modelagem e desenvolvimento de uma camada de abstração (*middleware*) para integração de sensores.

Palavras-Chave: Fusão de Dados de Sensores; Sistemas Ciberfísicos; Rede de Sensores; *Middleware* de sensores.

ABSTRACT

With the popularization of computers and electronic devices, computing is increasingly present and having great importance for the completion of tasks. Cyber Physical Systems (CFS) have emerged as a means to integrate computer systems with physical processes, integrating and interacting the real world with the virtual world by merging concepts such as computing, communication and control. In this scenario, the physical processes that have a major influence on CFS requiring this type of system has a network of different types of sensors for the perception of the physical environment. The integration of various sensors, the development of routines to facilitate the information acquisition and the actuation in the physical environment are some of the challenges in this scenario, since there are various kinds of sensors, manufacturers and setting modes according to the purpose. To mitigate these problems, is proposed in this project the modeling and development of an abstraction layer (*middleware*) to integrate sensors and offer support to data *Fusion*.

Keywords: Sensor *Fusion*; Cyber Physical Systems; Sensor Networks; Sensor *Middleware*

INTRODUÇÃO

Os Sistemas Cíberfísicos (SCF) surgiram como um meio de integrar sistemas computacionais com os processos físicos, integrando e interagindo o mundo real com o mundo virtual através da fusão de conceitos como computação, comunicação e controle. Usualmente compostos de um conjunto de agentes unidos por uma rede de trabalho, incluindo sensores e atuadores, unidades de controle de processamento e dispositivos de comunicação (Cardenas, 2008).

Os processos físicos atuam de forma intensa nos S.C.F., portanto são compostos por diversos sensores, responsáveis por fazer a interação direta com o meio físico e os dados obtidos pelos sensores, são utilizados como forte auxílio à tomadas de decisões (Liggins, 2008).

A fim de tornar as informações dos sensores mais precisas e completas, tolerantes a falhas e prover uma alta integração entre os mesmos, pode-se organizá-los de modo que se consiga aplicar algoritmos de fusão de sensores (Faceli, 2001).

A fusão de sensores consiste na aplicação de algoritmos nos dados provenientes dos sensores, a fim de se obter dados mais precisos e maior percepção do ambiente (Khaleghi. 2011) (Faceli, 2001).

Essa organização depende do objetivo a ser alcançado pela aplicação que utiliza a fusão, já que os sensores podem ser organizados de forma competitiva, complementar, cooperativa e independente e, conforme o propósito, os sensores podem ser utilizados diferentes níveis de fusão como: nível de sinal, nível de *pixel*, nível de característica, nível de símbolo (Faceli, 2001) (Salustiano, 2006) (Wellington, 2002).

Os sistemas Cíberfísicos são compostos por sensores podendo fazer o uso de vários sensores para formar uma rede que pode atuar em diferentes propósitos. Porém em algum momento estes dados são fundidos agregando valor e transformando-se em informações para tomada de decisão. Neste cenário um dos problemas a ser enfrentado é a integração de diferentes tipos de sensores. Por haver diversos fabricantes, diferentes configurações destes dispositivos (que podem variar conforme o propósito da aplicação) e por existir diversos tipos de sensores (que vão desde sensores simples como infravermelho até sensores de *laser* para mapeamento tridimensional), o desenvolvimento de aplicações que envolvam estes

dispositivos se torna complexa, pois conceitos como heterogeneidade dos sensores tanto no modo de atuação como na estrutura do dado disponibilizado devem ser considerados no desenvolvimento (Sá, 2010) (Garay *et al*, S. T., 2013).

Este projeto tem como objetivo a pesquisa e desenvolvimento de um *middleware* apoiado por facilitadores de integração para diferentes sensores, criando uma camada de abstração dos sensores, para que assim, os detalhes de funcionamento destes dispositivos fiquem mais transparentes ao desenvolvedor.

Para isso pode-se definir as seguintes fases:

- Revisão bibliográfica e levantamento de trabalhos correlatos;
- Pesquisa e caracterização dos dados dos sensores;
- Criar relações de generalização das características dos sensores;
- Estudo de algoritmos utilizados em fusão de sensores;
- Definir e projetar o *middleware*;
- Desenvolver o modelo;
- Testes e validações através de estudo de caso;

Um *middleware* é uma solução que contem um ambiente que suporta e gerencia várias aplicações. Ele padroniza e abstrai rotinas inerentes aos sensores como capturar dados, integrá-los e gerenciá-los (HADIM *et al*, 2006).

O projeto é dividido em etapas que contemplam o levantamento de trabalhos correlatos, estudo e classificação dos sensores, estudo de algoritmos utilizados em fusão de sensores, definição e projeto do *middleware*, desenvolvimento do mesmo e testes mediante estudo de caso.

Uma arquitetura para o *middleware* foi desenvolvida para abstração dos processos de captura dos dados dos dispositivos e também prover integração de diversos sensores. Além disso, é possível aplicar algoritmos de fusão de sensores nos dados obtidos. Como estudo de caso, uma simulação de três GPS's marcando as coordenadas durante um percurso em uma rua. O *middleware* desenvolvido foi utilizado para integrar os dados dos sensores e realuzar a fusão dos dados dos GPS's (coordenadas).

A monografia está dividida da seguinte forma. O capítulo 1 apresenta a fundamentação teórica do projeto, aborda conceitos sobre sistemas cíberfísicos e fusão de sensores, bem como suas respectivas aplicações. O capítulo 2 apresenta os trabalhos correlatos relacionados ao desenvolvimento de *middleware*s com foco em sensores. O

capítulo 3 apresenta as etapas de concepção do projeto como modelagem e implementação. No capítulo 4 é apresentado os estudos de caso. E por fim no capítulo 5 são apresentadas as conclusões e os trabalhos futuros.

CAPÍTULO 1 – FUNDAMENTAÇÃO TEÓRICA

Com a popularização de computadores e dispositivos eletrônicos, a computação está cada vez mais presente na rotina diária dos seres humanos, se tomando assim um instrumento de grande importância para realização de tarefas. Os Sistemas Cíberfísicos (S.C.F.) surgiram como um meio de integrar sistemas computacionais com processos físicos, integrando e interagindo o mundo real com o mundo virtual através da fusão de vários conceitos como computação, comunicação e controle. A integração com o meio físico se dá por sensores que exercem o papel de fornecer ao sistema as condições e características do ambiente, e para prover dados mais completos e precisos há uma agregação de informações de diversos sensores podendo ser utilizada uma técnica chamada fusão de sensores.

Nesta seção são apresentados os conceitos importantes acerca de Sistemas Cíberfísicos e fusão de sensores para o entendimento da elaboração deste trabalho.

1.1 - SISTEMAS CIBERFÍSICOS

A Computação passou por várias evoluções e transformações até chegar no que há disponível hoje. Estas transformações englobam muitos conceitos que vão desde a capacidade de processamento, até paradigmas de programação e *interface* com usuário.

O período entre as décadas de 60 e 70 foi marcado pelo uso dos *mainframes*, que eram grandes computadores feitos para aplicações que envolviam processamento de grande volume de dados e se encontravam concentrados nas grandes empresas e universidades. Já as décadas de 80 e 90 foram caracterizadas pela popularização dos computadores através dos computadores pessoais, e pelo o surgimento da *Internet*. Aproximadamente dez anos após essa popularização o conceito de computação ubíqua surgiu com a definição de que dispositivos computacionais são partes “invisíveis” do ambiente, ou seja, a comutação ubíqua tem como objetivo tornar a interação homem computador transparente transformando seu uso em ações e comportamentos naturais, fazendo parte do cotidiano dos usuários. Por fim os sistemas cíberfísicos (S.C.F.) são uma nova revolução da computação, onde se há integração entre sistemas de informação e processos físicos (Lee, 2009).

Segundo (Rajkumar, 2010) da mesma forma em que a *Internet* transformou a maneira das pessoas interagirem e se comunicarem uma com as outras, os S.C.F. irão transformar a maneira de interação e controle do mundo físico, pois, os S.C.F. integram computação e comunicação com monitoramento e controle no mundo físico e seus processos são coordenados, monitorados, controlados e integrados por um núcleo computacional.

Como características destes sistemas, se destacam a heterogeneidade e escalabilidade de componentes bem como questões relacionadas a controle, sensoriamento e aplicações, fazendo com que a pesquisa e desenvolvimento de um *middleware* seja motivada para lidar com estas características que são essenciais à estas aplicações (Garay, 2013). Isto faz com que os sistemas ciberfísicos sejam uma potencial área para aplicação do *middleware* proposto neste trabalho.

Deste modo pode-se considerar que os S.C.F. emergiram mediante a junção de tecnologias como sistemas embarcados, sistemas de controle, sistema de informação e sensoriamento distribuído (Rajkumar, 2010).

Em sistemas cíberfísicos há uma grande interação entre o mundo virtual e mundo real, pois o sistema ciberfísico depende da percepção do ambiente como auxílio nas tomadas de decisões. Estes sistemas são eficientes e confiáveis e geralmente possuem a capacidade de integrar sistemas que operam isoladamente a fim de formar um sistema mais complexo, com novas capacidades (Lee, 2009).

Tem-se como exemplo de áreas de aplicação de sistemas cíberfísicos:

- **Dispositivos e sistemas médicos:** auxiliam na concepção de marca-passos, dispositivos de suporte a vida como, por exemplo, injeção automática de medicamentos, aparelhos médicos como ressonância magnética (Broy, 2010) (Penteado, 2010);
- **Sistemas aeroespaciais e de Defesa:** sistemas de controle de tráfego aéreo, controle de satélites, estações espaciais que requerem controles precisos, alta segurança nas operações e grande poder computacional (Penteado, 2010);
- **Robótica, Automação de processos e máquinas:** robôs de linha de produção, controle de manufatura, dispositivos robóticos utilizados para substituir humanos em situações perigosas (Penteado, 2010);

- **Controle de tráfego e rodovias inteligentes:** Sistema de integração e sincronia de semáforos, controle de desvio de rotas (Broy, 2010);
- **Veículos inteligentes e autônomos:** Sistema de auxílio ao motorista, sistema de injeção eletrônica, *air bag*, sistemas de freio ABS (Penteado, 2010);
- **Engenharia e construção civil:** utilização de sensores para auxiliar em construções de pontes, prédios podendo servir de base para prevenção de acidentes naturais, catástrofes ou apenas para comodidade, como por exemplo, casas inteligentes, controle de distribuição de energia, gerenciamento de recursos hídricos (Penteado, 2010).
- **Monitoramento de ambientes:** Sistema em que há uma distribuição de sensores em locais geograficamente variados e que possuem grande área de cobertura como florestas, rios, montanhas. Devem operar sem intervenção humana por longos períodos com baixo consumo de energia (Sanislav e Miclea, 2012).

De acordo com (Rajkumar, 2010) e (Sanislav e Miclea, 2012) os Sistemas cíberfísicos devem operar em tempo real de modo a prover confiança, segurança e eficiência. Questões relacionadas à tolerância a falhas, segurança tanto em questão de rede ou mesmo a *software* e a precisão dos dados devem ser tratadas, pois como características estes sistemas requerem controles precisos e confiáveis, uma alta segurança nas operações e geralmente exigem novas tecnologias de análise, síntese e integração de subsistemas bem como formas de prover com precisão dados e redução do consumo de energia, tais questões são alguns dos desafios encontrados e enfrentados no desenvolvimento destes sistemas cíberfísicos.

Segundo (Sanislav e Miclea, 2012) sistemas cíberfísicos estão sendo atualmente impulsionados por várias tecnologias pelos seguintes motivos:

- Proliferação de sensores de baixo custo;
- Disponibilidade de pequenos dispositivos computacionais a um baixo custo;
- Consolidação da comunicação sem fio e o aumento de dispositivos que possuem a capacidade de se comunicar desta forma;
- Aumento da largura de banda;
- Constantes melhorias em economia de energia, uso de fontes alternativas de energia bem como a pesquisa por novas fontes.

Porém mesmo com o impulso gerado por diversas tecnologias ainda há grande necessidade de uma base tecnológica consolidada para que se possam desenvolver sistemas cíberfísicos em larga escala e de maneira acessível.

Atualmente são utilizados diversos sensores como, por exemplo, infravermelho, sonar, GPS, câmera, temperatura, pressão; que podem compor nodos (ou nós) homogêneos ou heterogêneos e estes, por sua vez, podem estar unidos por uma rede pelo qual desempenham a tarefa de perceber o meio físico e com isso, transmitir as condições do ambiente para o sistema. Os dados gerados pelos sensores, são utilizados como forte auxílio à tomadas de decisões, já que os processos físicos atuam de forma intensa nos S.C.F. (Loureiro, 2003) (Garay, 2013).

Os nós são de baixo custo, baixo consumo de energia e quando combinados possuem características como escalabilidade, alta precisão dos dados obtidos e mecanismos para adaptação. Eles podem operar de forma isolada ou também podem ser organizados de maneira em que se forme uma rede para que as informações obtidas durante observações de algum fenômeno possam ser combinadas (Hadim, 2006).

As redes de sensores bem como seus nós podem executar a fusão de dados agregando os dados dos sensores mediante uma métrica definida. A fusão de dados é apontada como uma alternativa para pré-processar os dados obtidos por meio destes dispositivos, porem é importante destacar que todos os sensores disponíveis estão sujeitos a erros de suas medições, podendo ser acarretadas por diversos fatores, como por exemplo, ruídos de campo eletromagnético, radiações, interferências, variações bruscas de temperatura, das condições do ambiente e etc., e mesmo se o sensor estiver sob condições perfeitas do ambiente, os dados coletados ainda podem estar sujeito erros cometidos pela própria precisão do dispositivo ou interrupção do funcionamento de algum componente do nó (Loureiro, 2003).

Nestes casos a fusão pode ser utilizada em situações onde se necessita de dados mais precisos, tornando a rede mais robusta e tolerante a falhas. (Loureiro, 2003).

Se tratando de rede de sensores, a agregação de dados tem grande importância, pois a eliminação de dados redundantes tem como resultado final a redução do volume de dados, fazendo com que o número de trocas de mensagens entre os nós seja reduzido, impactando na economia de energia (Loureiro, 2003).

Técnicas de fusão também permitem complementar informações, onde dados de

sensores diferentes podem ser combinados obtendo assim, novas informações que não seriam possíveis se estivesse utilizando um único sensor, ou seja, que não estivessem usando técnicas de fusão para a obtenção de dados (Loureiro, 2003).

Explicação detalhada sobre fusão de dados como tipos de fusão, técnicas, algoritmos de fusão e a forma de se aplicar estes algoritmos em função da organização dos sensores, será descrita no próximo capítulo.

1.2 - FUSÃO DE SENSORES

Os sensores são dispositivos que tem como função transmitir para o mundo computacional, as características e condições do ambiente em que ele está inserido. Ele recebe estímulos físicos do ambiente e os transformam em sinais elétricos que posteriormente são processados em sinais digitais.

As grandes maiorias dos sensores no mercado já possuem capacidade de processar estes sinais e transformá-los em dados inteligíveis (dados brutos) como medidas de temperatura, distância, posicionamento, pressão, etc., estes dispositivos que estão inseridos no meio físico, estão sujeitos a erros de suas medições provocadas, por exemplo, por ruídos eletromagnéticos, falhas de componentes e interferências.

A utilização de diversos sensores trabalhando em conjunto observando um ambiente e a aplicação de fusão de dados nestes dispositivos é uma alternativa de se conseguir maior tolerância a falhas, maior precisão e também derivar informações que não são fornecidas pela utilização individual destes dispositivos (Khaleghi, 2011) (Liggins, 2008).

Os termos Fusão de Dados, Fusão de Sensores, Integração de Múltiplos Sensores, Fusão de Dados de Múltiplos Sensores e Fusão da Informação, são algumas das denominações de técnicas e tecnologias, que visam à combinação de dados de múltiplas fontes (Salustiano, 2006).

A fusão de dados é o processo que trata detecção, associação, correlação, estimação e combinação de dados de diversas fontes. O termo também é empregado como definição de um conjunto de técnicas e metodologias e ferramentas que visam à combinação de dados de baixa qualidade com o objetivo de obter dados de melhor qualidade (Salustiano, 2006).

A fusão de dados de múltiplos sensores tem como objetivo combinar dados provenientes de sensores a fim de se obter inferências sobre o meio físico (Salustiano, 2006).

Fusão da informação é o termo que engloba a fusão de qualquer tipo de informação. Está relacionado com a redução do volume das informações e a integração delas, cobrindo todos os aspectos da fusão. A fusão da informação envolve aspectos relacionados às informações provenientes de diversas fontes, tais como sensores, banco de dados, informações fornecida pelo homem (Salustiano, 2006).

A integração de múltiplos sensores refere-se à utilização sinérgica de múltiplos sensores a fim de que mediante a integração da informação destes dispositivos suas informações possam ajudar na realização de tarefas em um sistema. Os dados dos sensores são utilizados diretamente no controle da aplicação. O foco da integração de sensores está em características como, modularização, escalonamento, robustez, comunicação de dados (Salustiano, 2006) (Elmenreich, 2002).

A fusão de sensores e fusão de múltiplos sensores refere-se à combinação de dados sensoriados ou dados derivados de dados sensoriados a fim de que o dado resultante desta combinação seja de alguma maneira melhor do que seria obtido se estivesse utilizando estes dispositivos individualmente. O principal objetivo deste processo de fusão é fazer uma representação ou modelo do ambiente dentro dos ambientes computacionais (Salustiano, 2006).

De acordo com (Liggins, 2008) técnicas de fusão de dados refere-se à combinação de múltiplos sensores de tipos iguais ou diferentes. Porém pode-se utilizar um conjunto de medições de um único sensor para que se possa combinar estes dados posteriormente.

Os sensores utilizados podem ser de naturezas iguais como, por exemplo, vários sensores de temperatura ou podem ser também de naturezas diferentes como, por exemplo, sensor de temperatura, pressão, GPS, etc.

O termo que melhor se encaixa neste trabalho é o termo de fusão de sensores, onde as informações destes dispositivos, podendo ser de naturezas iguais ou diferentes são combinadas com o objetivo de prover informações mais refinadas, precisas e completas sobre o meio observado, resultando assim uma melhor percepção do ambiente.

A utilização da fusão de sensores faz com que a obtenção de novas informações seja realizada a partir dos dados dos sensores utilizados, sem a necessidade de sensores específicos que possivelmente possam acarretar custos altos (Salustiano, 2006).

1.2.1 - VANTAGENS DA FUSÃO DE SENSORES

Sistemas que fazem a utilização da fusão de sensores oferecem vantagens sobre sistemas que utilizam sensores individualmente. Sistemas que fazem o uso de sensores individuais, podem geralmente gerar problemas como falhas dos sensores, baixa faixa de cobertura, atrasos de leituras, imprecisão e incertezas (Elmenreich, 2002). Logo abaixo, detalhes sobre cada um destes casos:

- **Falha de sensor:** a falha de algum sensor causa a perda da percepção do objeto desejado. Esta perda pode ser tanto de uma característica como do objeto como um todo (Elmenreich, 2002).
- **Faixa de cobertura restrita:** geralmente um sensor cobre apenas uma parte da região observada. Por exemplo, um sensor de temperatura em uma caldeira cobre apenas uma região próxima de si e pode não indicar corretamente a temperatura média da (Elmenreich, 2002).
- **Atrasos em leituras:** alguns sensores possuem atraso (*delay*) entre uma leitura e outra. Estes atrasos acabam gerando "janelas" em que não há nenhuma informação sobre este período, limitando a frequência de medições (Elmenreich, 2002).
- **Imprecisão:** as medições feitas por sensores individuais são limitadas à precisão do sensor empregado (Elmenreich, 2002).
- **Imperfeição:** as imperfeições não dependem somente da imprecisão. Os dados dos sensores são geralmente afetados por algum nível de imprecisão que possa haver nos dispositivos ou erros provocados também por ruídos.

Uma solução para estes problemas é a fusão de sensores, na qual tem a proposta de fornecer informações úteis sobre características do ambiente, bem como permitir sua visão global. Uma das vantagens da fusão é que informações mais refinadas e precisas do ambiente podem ser obtidas, em um menor intervalo de tempo e custo (Salustiano, 2006). Isso é possível através das seguintes vantagens:

- **Tolerância a falhas:** O uso de múltiplos sensores aumenta a tolerância a falhas de um sistema. Caso algum sensor falhar, os demais sensores que não falharam continuam a fornecer informações para o sistema (Elmenreich, 2002);
- **Maior cobertura no domínio espaço-tempo:** Um sensor pode cobrir uma área (espaço) em que outros não cobrem, e em contrapartida, outros podem realizar uma medição enquanto outros não podem. Um sensor pode fazer uma medição enquanto outro está dentro de um *delay*, desta forma, fornecendo dados sobre um período em que não seria possível se estivesse utilizando um sensor isoladamente (Elmenreich, 2002);
- **Redução de ambiguidade:** A união de informações reduz a ambiguidade da interpretação dos valores observados. Valores discrepantes do conjunto observado podem ser descartados mediante critério definido (Elmenreich, 2002);
- **Confiança:** A medição de um sensor é confirmada pelas medições de outros sensores cobrindo o mesmo domínio (Elmenreich, 2002);
- **Robustez em relação a interferências:** Com o aumento da dimensão do espaço medido utilizando sensores diferentes para o mesmo fim, como sensores ópticos juntamente com sonares, torna sistema menos suscetível a interferências (Elmenreich, 2002). Um sensor óptico, por exemplo, pode sofrer interferência da iluminação do ambiente, dependendo da intensidade. Neste caso, o sonar opera em conjunto para suprir possíveis defasagens ocorridas por causa da interferência.
- **Aumento da resolução:** Quando os dados de vários sensores são utilizados e fundidos, a resolução do valor resultante é melhor do que um valor obtido por um sensor individualmente (Elmenreich, 2002);

Em um sistema onde os dados são processados por métodos de fusão, a entrada para o controle da aplicação é padronizada independentemente dos sensores utilizados, facilitando a implementação da aplicação e também a possibilidade da alteração da camada de sensores quanto ao número e tipos empregados sem a necessidade de modificações no *Software* da aplicação (Elmenreich, 2002).

1.2.2 - APLICAÇÃO DA FUSÃO DE SENSORES

A aplicação da fusão de sensores se dá nas mais diversas áreas. A princípio, foi utilizada em aplicações militares para reconhecimento de alvos inimigos, monitoramento de armas, veículos autônomos, sensoriamento remoto, monitoramento dos campos de batalha. As aplicações militares, não somente são utilizadas em guerras, mas também no sistema de defesa, monitoramento de saúde dos soldados e no auxílio de operações logísticas. Em aplicações não militares, pode-se incluir o monitoramento de processos manufaturados, robótica, monitoramento de ambientes e aplicações médicas (Liggins, 2008).

De maneira sintética, as tabelas 1 e 2 mostram, respectivamente, as aplicações militares e não militares e suas características:

Tabela 1 - Aplicações de fusão no âmbito militar. Fonte: Liggins, 2008

Aplicações	Objetivo da Fusão	Dados (Sinais) Observados	Raio de Alcance da Aplicação	Sensores embarcados
Monitoramento do Oceano e costas marítimas	Detecção, rastreamento e identificação de alvos ou eventos	Sinais de Ultra-som, radares, radiação de ondas eletromagnéticas	Centenas de milhas náuticas	Navios, aviões, submarinos, bases terrestres e marítimas de monitoramento
Defesa Aérea	Detecção, rastreamento e identificação de aviões	Ondas eletromagnéticas	Dezenas e centenas de quilômetros, para aplicações estratégicas ou táticas respectivamente	Bases no solo e aviões
Inteligência em combates, monitoramento, identificação e detecção de alvos	Detecção, rastreamento e identificação de alvos terrestres	Ondas eletromagnéticas, infravermelho	Dezenas de centenas de quilômetros	Bases terrestres, aviões, veículos terrestres
Centros estratégicos e defesa	Detecção de ações inimigas de misseis	Ondas eletromagnéticas	Global	Satélites, aviões

Tabela 2 - Aplicações de fusão no âmbito não militar. Fonte: Liggins, 2008

Aplicações	Objetivo da Fusão	Dados (Sinais) Observados	Raio de Alcance da Aplicação	Sensores embarcados
Manutenção baseada em condição	Detecção, caracterização de falas, recomendação de manutenção	Sinais de ultra-som, radiação de ondas eletromagnéticas, raios X, temperatura, vibrações	Microscópico a centenas de quilômetros	Navios, aeronaves, construções
Robótica	Localização de objetos, parâmetros para locomoção e desvio de obstáculos	Sonares, radiação e ondas eletromagnéticas, imagens, raios X	Microscópico a dezenas de metros	Robôs
Diagnósticos médicos	Localização e identificação de tumores, doenças, etc.	Raios X, temperatura, inspeção visual, imagens, dados, químicos e biológicos	Corpo humano	Dispositivos como equipamento de raio X, ressonância magnética
Monitoramento de ambientes	Detecção e identificação de fenômenos naturais, como terremotos, furacões, etc.	Sinais de radares, satélites, sensores sísmicos, dados químicos e biológicos	Quilômetros, centenas de quilômetros	Satélites, aviões, dispositivos subterrâneo

Como exemplo da utilização de sensores, podemos citar um sistema de manutenção baseada em condição, onde vários sensores como, por exemplo, sensores de temperatura, ultrassom, monitores de impurezas em fluídos lubrificantes, acelerômetros, trabalham em conjunto para monitorar uma determinada estrutura, podendo ser aeronaves, veículos terrestres, construções, estruturas prediais etc., a fim de monitorar e identificar a evolução da condição desta estrutura ao longo do tempo e estima-se com base nestas informações sua vida útil. Isso faz com que a manutenção de estruturas ocorra de maneira preventiva, pois possíveis falhas podem ser detectadas com antecedência.

Em aplicações médicas, principalmente a fusão de imagens, pode ser utilizada para identificação de doenças, tumores, etc. Para prevenção de desastres naturais, sensores sísmicos, satélites, radares, são utilizados em conjunto para se ter dados mais precisos sobre as condições do clima.

Sistemas de defesa naval, como por exemplo, monitoramento de zonas costeiras, podem se utilizar de sensores embarcados em diversos locais. Sonares e radares submarinos

ou alocados em aviões, juntamente com informações provenientes de satélites, podem ser utilizados para identificar submarinos inimigos ou embarcações estranhas (Liggins, 2008).

1.2.3 - CONFIGURAÇÃO E ORGANIZAÇÃO DOS SENSORES

Apesar de cada sensor prover informações em diferentes aspectos do ambiente, a utilização de vários sensores abre a possibilidade de combinar as informações destes dispositivos. A construção de uma rede de sensores envolvem sensores de diversos tipos, tanto de naturezas iguais ou diferentes, medindo partes iguais ou diferentes do ambiente. A maneira em que estes sensores são organizados define a forma de como a fusão deve ocorrer sendo que ela pode ocorrer na forma complementar, competitiva, cooperativa ou independente (Faceli, 2001).

A fusão complementar trata informações incompletas ou segmentadas obtidas a princípio a partir de sensores diferentes. Neste tipo de configuração, os sensores são organizados de maneira que se possam medir partes diferentes do mesmo fenômeno observado. A informação obtida como resultado do processo de fusão possui uma área de cobertura maior em comparação a utilização de um único sensor (Faceli, 2001).

Pode-se citar como exemplos deste tipo de fusão, diversos sensores de distância localizados em pontos distintos (apontados para direções diferentes) em um veículo. Isso faz com que maiores informações sobre os objetos e possíveis obstáculos ao redor do veículo sejam obtidos. Velocímetro e odômetro, por exemplo, podem ser usados, ambos medindo informações diferentes, mas contribuindo para a navegação do veículo (Faceli, 2001).

Na fusão competitiva são utilizados sensores que são organizados de maneira a realizar medições da mesma parte do ambiente ou aspecto de um fenômeno. Desta forma é feita a fusão de informações redundantes provenientes destes dispositivos, tornando o sistema mais confiável e tolerante a falhas. Neste caso, os sensores podem ser idênticos ou utilizar tecnologias diferentes para medir o ambiente. Como exemplo da fusão competitiva, pode-se ter em um robô um conjunto de sensores como, por exemplo, sensor de distância a *laser* e um sonar apontados para o mesmo obstáculo provendo a distancia deste obstáculo onde radares idênticos irão cobrir a mesma área fazendo com que o sistema continue funcionando mesmo que haja falha de um deles (Faceli, 2001).

A fusão cooperativa tem como objetivo obter novas informações a partir da fusão de

sensores independentes. As informações obtidas por este processo não seriam possíveis de se derivar se fossem usados sensores individualmente. Podem-se citar como exemplo, vários sensores de temperatura dispostos em linha, efetuando além da medição da temperatura, informações sobre a taxa de variação de temperatura (Faceli, 2001).

Por fim, a fusão independente utiliza sensores nas quais suas informações a princípio não estão relacionadas. Neste caso, essas informações que não possuem relação podem estar armazenadas na mesma estrutura de dados. Para exemplo deste tipo de fusão podemos citar informações provenientes de um radar e em sensor de temperatura obtidas em um mesmo local e que armazenadas na mesma estrutura são capazes de obter a informação da temperatura do local medido (Faceli, 2001).

Na Figura 1 ilustra a organização dos sensores e os dados resultantes obtidos após o processo de fusão.

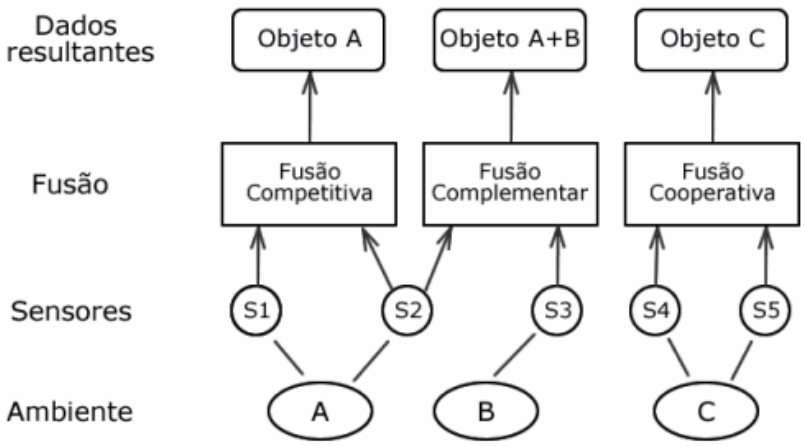

Figura 1 - Organização dos sensores em relação ao tipo de fusão. Fonte: Salustiano, 2006.

Os sensores S1 e S2 realizam medições da porção A do ambiente, ou seja, estes sensores estão organizados de forma a realizar medições da mesma parte do ambiente. Portanto o processo de fusão utilizado é a fusão competitiva que utiliza os dados fornecidos por estes sensores, resultando por fim, no Objeto A, representando para o sistema, informações mais confiáveis sobre o ambiente.

Os sensores S2 e S3 estão medindo partes diferentes do ambiente. O sensor S2 realiza medições na porção A enquanto o sensor S3 realiza medições na porção B. Na aplicação da fusão complementar nestes sensores, as informações obtidas por cada um são utilizadas para se obter uma área de cobertura maior do ambiente, resultando no Objeto A+B.

Os sensores S4 e S5 realizam medições da porção C do ambiente. Em seus dados é aplicado a fusão cooperativa, onde são derivadas novas informações a partir dos dados dos sensores, resultando assim no Objeto C.

1.2.4 - NÍVEIS DE FUSÃO

Um conceito importante abordado na literatura é o nível em que ocorre a fusão. A fusão pode ocorrer nos níveis definidos como *low level fusion* (dados brutos dos sensores), *medium level fusion* (extração de características) e *high level fusion* (tomada de decisão, nível semântico). Estes níveis dividem os processos fusão de acordo com a natureza do dado, além de classificar os tipos de fusão que podem ocorrer nos dados (Wellington, 2002) (Nakamura, 2007).

Além da classificação e de considerar o tipo, ela também leva em conta a maneira em que a informação sensorial está modelada, métodos utilizados para fusão (Salustiano, 2006). Este modelo trata a relação de cada tipo de dado com seu respectivo processo de fusão. A Tabela 4 apresenta uma breve comparação entre os níveis de fusão.

1.2.4.1 - FUSÃO NO NÍVEL DE SENSORES

O termo fusão de sensores também pode ser encontrado na literatura como *low level fusion* que refere-se à combinação de sinais provenientes de um grupo de sensores de mesma natureza como temperatura, pressão, coordenadas geográficas, luminosidade; ou também sinais naturezas diferentes, mas que possuem correlação entre estas variáveis como, por exemplo, posição e velocidade. Estes dados são utilizados como entrada do processo de fusão e como resultado do processo, se obtém dados mais confiáveis e precisos (Wellington, 2002).

A fusão neste nível gera como resultado um dado que geralmente é da mesma natureza dos sensores, mas com uma representação mais fiel do meio físico. A melhoria na qualidade do dado vai estar relacionada ao algoritmo utilizado para realizar a fusão (Salustiano, 2006). Neste nível de fusão, os dados obtidos devem estar sincronizados, para que o processo de fusão ocorra de maneira eficiente. Caso não haja sincronia dos dados, pode-se fazer uso de registros do instante da obtenção dos dados juntamente com os dados obtidos com o intuito de efetuar a combinação dos dados de outros sensores neste mesmo instante. Em alguns casos podem ser usadas técnicas utilizadas em sistemas distribuídos como relógios

globais para fazer a sincronia das medições dos sensores (Luo, 1990).

Um dos processos utilizados para efetuar a fusão de sensores da mesma natureza é o processo de se aplicar médias aritméticas aos dados. Uma forma de fazer o refinamento deste processo é aplicar média ponderada, sendo utilizado como parâmetro para ponderamento o inverso do valor da precisão de cada dispositivo. Isso faz com que os sensores com maior precisão tenham maior peso (Salustiano, 2006).

Outra maneira de efetuar a fusão neste nível é através do Filtro de Kalman. O Filtro de Kalman é um modelo matemático estimador para problemas lineares quadráticos, que consiste em estimar e filtrar o estado/sinais de situações que possam ser descritas em um sistema dinâmico linear (Grewal, 2008).

O trajeto de um veículo por uma estrada, órbita de satélites ou fluxo de um rio, são sistemas dinâmicos e que podem ser descritos em equações lineares. Em sistemas dinâmicos nem sempre é possível ou desejável efetuar medições de todas as variáveis envolvidas, então o Filtro de Kalman provê meios de se fazer inferências sobre estas medições perdidas ou ruidosas. Ele também é utilizado para predições de estados futuros (Grewal, 2008).

Na prática, este método utiliza uma série de equações, geralmente utilizando vetores e matrizes, para calcular os processos de predição (*Time Update*) e correção (*Measurement Update*). O processo de predição é responsável por projetar o estado atual das variáveis à frente no tempo e o processo de correção ajusta esta medida projetada por uma medição atual.

Uma das vantagens deste filtro é que seu uso não se restringe somente ao uso de um tipo de dado (somente temperatura, ou aceleração, ou velocidade). No Filtro de Kalman é possível utilizar dados que possuem relação entre si, como por exemplo velocidade a aceleração.

O Critério de Chauvenet é um método que, dado um conjunto de medidas de um objeto, pode eliminar os valores discrepantes deste conjunto. Nele é definido que se a probabilidade de desvio em relação a média de um valor medido for inferior que 1/2n (onde n é a quantidade de valores medidos), este valor pode ser excluído (Magalhães, 2011).

A princípio é calculado a média e o desvio padrão de um conjunto de medições. A próxima etapa consiste em calcular a razão entre o desvio médio de cada ponto e o desvio padrão. O resultado desta razão é comparado com valores de referência conforme a tabela 3 a seguir. Esta tabela apresenta a relação entre o valor máximo permitido e a quantidade de medições realizadas. Se o valor do cálculo for maior que o valor tabelado, este valor pode ser

excluído do conjunto. Se um valor for excluído do grupo de medições, uma nova média e desvio padrão devem ser calculados para o novo conjunto de dados obtido (Magalhães, 2011).

Tabela 3 - Tabela dos valores de referência utilizados no Critério de Chauvenet Fonte: Magalhães, 2011.

Numero de Leituras	Razão entre o máximo desvio aceitável e o desvio padrão
3	1,38
4	1,54
5	1,65
6	1,76
7	1,80
10	1,96
15	2,13
25	2,33
50	2,57
100	2,81
300	3,14
500	3,29

Desta forma é possível eliminar valores divergentes dos demais valores registrados. Isso faz com que sejam eliminados os dados dos sensores que possivelmente possam apresentar erros em relação aos outros sensores.

Se tratando de sinais provenientes de sensores ópticos, este nível de fusão também é responsável por melhorar a informação de uma imagem. Estas imagens podem ser de naturezas diferentes e podem ser obtidas por meio de sensores ópticos como câmeras, que podem registrar diversas cores do espectro visível ou invisível. A partir da combinação destas imagens é possível gerar novas imagens como resultado dessa combinação e assim extrair suas características. Além da combinação de imagens, os algoritmos de fusão deste nível podem ser aplicados em imagens de um único sensor, com o objetivo de se melhorar o contraste, luminosidade, detecção de movimento (Salustiano, 2006).

Um dos modos de se fazer a combinação de imagens é através da aplicação de filtros lógicos aos *pixel*s de uma imagem. Operações lógicas como *AND, OR, NOT* e *XOR* são aplicadas aos *pixel*s de forma a realizar a fusão das imagens. Na aplicação deste filtro, é

possível obter uma nova imagem combinada de sensores que registram diferentes faixas do espectro luminoso, aumentar o tamanho da imagem através da junção de imagens complementares, fazer comparação entre imagens capturadas pelo mesmo sensor (Luo, 1990).

A morfologia matemática é outro método utilizado no processo de fusão no nível de *pixel* que aplica às imagens operações da teoria dos conjuntos com o objetivo de detectar bordas, utilizado, por exemplo, na detecção dos limites de rodovias (Luo, 1990).

O recozimento simulado é a técnica que realiza o processamento de imagens analisando cada *pixel* e seus vizinhos. O objetivo é estabelecer relações entre os *pixel*s de uma imagem ou *pixel*s de varias imagens. Utilizando técnicas de recozimento simulado é possível realizar a fusão de duas imagens de câmeras distintas para estimação de profundidade ou para compensar parcialmente regiões ocluídas em imagens (Luo, 1990).

1.2.4.2 - FUSÃO NO NÍVEL DE CARACTERÍSTICA

Neste nível, também conhecido como *medium level fusion*, o processo de fusão tem como característica a extração de padrões, aspectos ou similaridades dos dados dos sensores. Geralmente é utilizado em imagens, mas pode ser feito em outros tipos de dados, a fim de se obter sentido semântico às medições dos sensores. Pode ser utilizado tanto para aumentar a probabilidade que uma característica extraída da informação fornecida por um sensor, corresponde efetivamente a um aspecto importante do ambiente, tanto como um meio de criação de características compostas adicionais para uso pelo sistema (Luo, 1990).

O objetivo deste nível é incrementar as informações de um sensor a partir da análise dos dados dos dispositivos vizinhos. A partir disso, é possível fazer o reconhecimento de padrões nos dados e também eliminar possíveis informações duvidosas. Quando vários sensores provêm informações similares de um determinado ambiente, a probabilidade de que esta característica realmente existe aumenta e a precisão pode ser melhorada também (Luo, 1990).

1.2.4.3 - FUSÃO NO NÍVEL SEMÂNTICO

Fusão no nível semântico, podendo ser chamado de *high level fusion*, permite que informações possam ser utilizadas em um alto nível de abstração. Neste nível é possível

realizar a fusão de dados de sensores que não possuem nenhuma relação entre si ou de informações de diferentes regiões do ambiente. A semântica utilizada neste nível de fusão pode ser obtida tanto pelos sensores, quanto por informações previamente conhecidas ou de fontes externas do sistema. O resultado da fusão é utilizado para tomada de decisão. Esta decisão é feita geralmente por características derivadas de informações dos sensores correspondentes a um modelo. Os símbolos utilizados geralmente tem uma medida do grau em que a informação sensorial coincide com o modelo (Salustiano, 2006) e (Luo e Kay, 1995).

Estimativa Bayesiana, Lógica Booleana e Logica de Fuzzy, são alguns dos métodos utilizados para realizar a fusão neste nível. Estes métodos fazem inferências para indicar ao sistema o grau de veracidade ou crença no caso da estimativa Bayesiana das informações e assim desencadeando a tomada de decisão do sistema (Salustiano, 2006) (Luo e Kay, 1995).

Tabela 4 - Tabela comparativa entre os níveis de fusão. Fonte: Luo, 1990.

	Nível de Sensores	**Nível de Característica**	**Nível de Semântica**
Tipo de Dado	Nível de Sensores	Nível de Característica	Nível de Semântica
Nível de Representação	Dados Brutos dos sensores e imagens	Características extraídas dos dados e imagens	Combina informações de múltiplas fontes de dados
Método de Fusão	Baixo Nível	Nível Médio	Alto Nível
Melhorias providas pela Fusão	Estimação, combinação de dados sensoriados, técnicas de processamento de imagens	Transformações Geométricas, combinação de características.	Estimativa Bayesiana, Lógica Fuzzy, métodos que fazem inferências na estimação da veracidade da informação

CAPÍTULO 2 – TRABALHOS CORRELATOS

O estudo de trabalhos relacionados é um ponto importante para elaborar uma abordagem viável, competitiva e com diferenciais. Neste capítulo são apresentados trabalhos em que são definidas abordagens de processos de desenvolvimento e modelagem de *middlewares* para sensores, considerando também pontos relacionados a fusão de dados e Sistemas Cíberfísicos.

No trabalho desenvolvido por Garay, J. R. B., Oliveira, A. M. e Kofuji, S. T. foi elaborada uma proposta de um *middleware* para rede de sensores para sistemas ciberfísicos. O trabalho aborda características e especificações que devem ser abordadas em um *middleware* com foco em SCF como heterogeneidade de sensores e outros dispositivos, topologia de rede, protocolos e padrões de comunicação, segurança da aplicação e gerenciamento de dados e eventos. Neste projeto, o *middleware* foi dividido em camadas (Ver figura 2) que são responsáveis de prover abstração necessária conforme explicado abaixo:

- **Camada de comunicação e Dados:** Nesta camada são abordados a heterogeneidade, gerenciamento de consumo energia e protocolos. Para atender esses pontos, o uso de topologias e protocolos devem ser padronizadas para aplicações suportadas pelo *middleware.*
- **Módulo de Integração:** Realiza integração do conjunto de sensores e atuadores com o *middleware.* É responsável por realizar operações de controle da rede, inserir ou remover módulos de hardware, abstração de protocolos e *interface*s com dispositivos.
- **Camada de Serviço:** Nesta camada se faz o uso de padrões de projeto e tecnologias para gerenciamento de serviços e dados, podendo ser localmente ou remotamente a fim de fazer abstração das rotinas inerentes a estas tarefas fornecendo recursos para desenvolvimento de aplicações.
- **Camadas de *Interface* e Domínio de Serviço:** Os Parâmetros de *Software* que automatizam ou apoiam processos utilizados por usuários na camada de aplicação são agrupados nesta camada. Os serviços são baseados em um modelo que pode ser dividido em três níveis, sendo estes: Nível de rede que trata comunicação entre o *middleware* e servidor e também entre dispositivos da

camada de sensores; *Hardware* de infraestrutura, trata a heterogeneidade de componentes e aplicações que fazem parte do *middleware* como servidores de banco de dados e *Web Services*; Nível de *Software*, onde é descrito um conjunto de aplicações para abstração de processos.

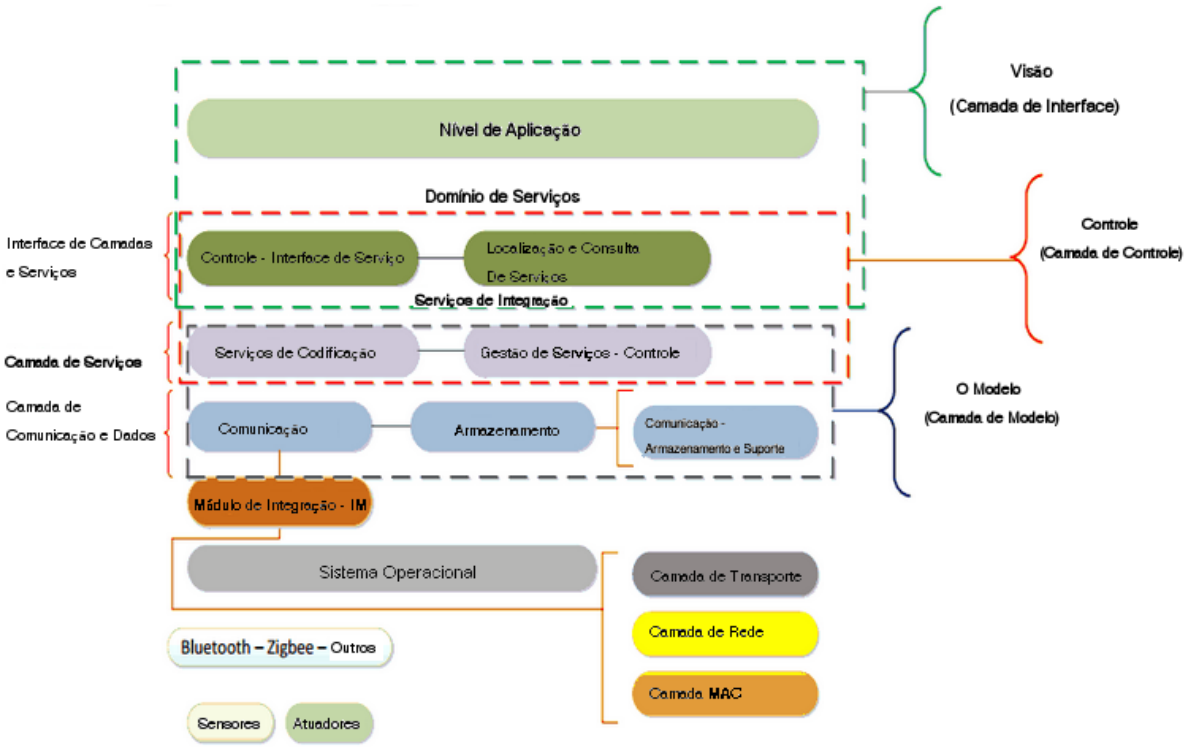

Figura 2 - Arquitetura do *middleware* para rede de sensores. Fonte: Garay e Kofuji, 2013

No trabalho desenvolvido por (Salustiano, 2006) o principal objetivo aplicar técnicas de fusão de dados em um sistema de monitoramento de ambientes. Tal sistema possui elementos denominados monitores no qual possuem um ou mais sensores para diversos fins.

O monitor é responsável por interagir com os sensores, integrar suas informações por meio de XML (*Extensible Markup Language*) e enviá-las para um servidor através da *Internet*. Cada monitor pode haver um ou mais canais de comunicação, sendo cada canal responsável pela comunicação com um sensor sendo que para utilizá-los, é necessário um cadastro prévio especificando a quantidade de sensores (canais) de cada nó de monitoramento (monitor) e a natureza do dado enviado por estes canais.

Depois é cadastrado no servidor uma ou mais tabelas para cada monitor, fornecendo ao sistema parâmetros de como os dados recebidos destes canais serão combinados. Após esta

etapa de configuração são aplicadas funções utilizando os dados coletados como: lógica Fuzzy, processamento das imagens coletadas, médias aritméticas e demais algoritmos conforme necessário.

A plataforma *ConBus* desenvolvida por (Sá, 2010), é um *middleware* com foco em aplicações móveis sensíveis ao contexto que visa oferecer facilidades para o desenvolvimento de aplicações que envolvam sensores que atuam como fornecedores de informação de contexto. A plataforma como um todo é capaz de gerenciar o ciclo de vida dos módulos de sensoriamento, oferecer um *framework* de integração de novos módulos, e prover às aplicações móveis sensíveis de contexto uma *interface* que padroniza o acesso aos dados providos por diferentes sensores através dos módulos de sensoriamento. Para a concepção do projeto, foram feitas as seguintes divisões em camadas (Ver figura 3):

- **Camada de Aplicação:** São as aplicações sensíveis à contexto que utilizam o *ConBus* como meio de obtenção das informações contextuais. As aplicações podem utilizar de uma API fornecida pelo próprio *middleware*. A API disponibiliza comandos dos quais as informações podem ser requisitadas.
- **Camada de *Middleware ConBus*:** Gerencia informações coletadas pelos sensores que são utilizadas em outro momento pelas aplicações sensíveis de contexto. O *ConBus* faz uso de componentes como *ConUnits*, Gerenciador de Contexto e Controlador de Comunicações. O *ConUnit* intermediam a comunicação entre os sensores e o restante da arquitetura para que posteriormente as informações possam ser utilizadas pelas aplicações; Gerenciador de Contexto atua intermediando as aplicações sensíveis de contexto e os *ConUnits* gerenciando o ciclo de vida das unidades que consistem em instalação, ativação, desativação e desinstalação; O Controlador de Comunicações recebe requisições das aplicações, as processa e envia os dados necessários ao Gerenciador de Contexto para que, após o processamento do gerenciador, a resposta seja enviada à aplicação.
- **Camadas de Sensores:** os módulos de sensoriamento são acoplados às unidades *ConUnits*.

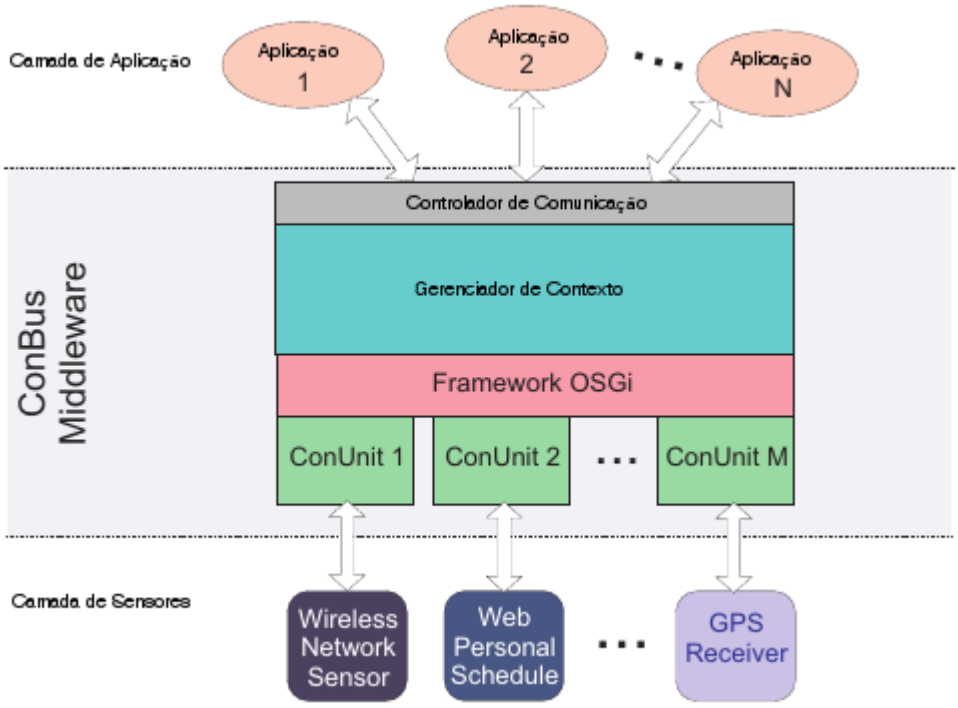

Figura 3 - Arquitetura da plataforma *ConBus*. Fonte: Sá 2010.

2.1 CONSIDERAÇÕES FINAIS

Neste capítulo foi demonstrado alguns dos trabalhos relacionado com o desenvolvimento de *middlewares*.

O primeiro trabalho tem foco abordar a heterogeneidade de dispositivos encontrados nos Sistemas cíberfísicos e na integração e abstração de operações inerentes à utilização de rede de sensores. Essa integração é concentrada na camada de comunicação do *middleware* que é composto pelo modelo de integração onde contempla a comunicação com a rede de sensores, inserção e remoção de módulos e também características de topologia são tratadas neste módulo.

O segundo trabalho aborda técnicas de fusão de sensores como parte de um sistema de monitoramento de ambientes. A integração dos sensores se dá previamente nos monitores onde posteriormente os dados são enviados pela *Internet* para o servidor onde serão aplicados os algoritmos de fusão. O funcionamento deste sistema é dependente da *Internet*, e por isso pode sofrer atrasos na comunicação. Deste modo é recomendado pelo próprio autor o uso em monitoramentos que não necessite de respostas em tempo real.

Por fim, o *middleware ConBus* descrito no último trabalho, é possível realizar

integração de dados de diversas fontes, não somente dos sensores, porém se existir mais de um sensor ou módulo acoplado que coleta o mesmo tipo de informação associada, o Gerenciador de Contexto irá utilizar o primeiro que encontrar, não possibilitando, por exemplo, o uso de coordenadas geográficas juntamente com indicações de localidade como nome da rua. O objetivo plataforma é prover abstração de rotinas inerentes à captura de dados dos sensores para aplicações sensíveis ao contexto.

Como diferencial dos demais, com o *middleware* proposto neste trabalho é possível utilizar os dados obtidos pelos sensores individualmente além da possibilidade de aplicar algoritmos de fusão de sensores. Outro ponto importante é que a arquitetura desenvolvida possui meios de se desenvolver e aplicar novos algoritmos de fusão conforme necessidade do desenvolvedor bem como facilitadores para inclusão de novos dispositivos. Todo o processo de aquisição e processamento dos dados é realizado localmente podendo ser em tempo real (dependendo da aplicação). O acesso à fonte de dados não se restringe apenas aos sensores, ou seja, a arquitetura do *middleware* permite o desenvolvimento de rotinas de acesso a diferentes fontes de dados, como por exemplo, arquivos de texto, banco de dados.

A abordagem do *middleware* desenvolvido será descrita no capítulo 3.

CAPÍTULO 3 – ABORDAGEM DA PROPOSTA

No desenvolvimento de aplicações que fazem o uso de sensores, um dos problemas a ser enfrentado é a integração de diferentes tipos de sensores. Por haver diversos fabricantes, diferentes configurações destes dispositivos e por existir diversos tipos de sensores, o desenvolvimento de aplicações que envolvam estes dispositivos se torna complexa, pois conceitos como heterogeneidade dos sensores tanto no modo de atuação como na estrutura do dado disponibilizado devem ser considerados no desenvolvimento (Sá, 2010) (Garay *et al*, 2013).

Um *middleware* é uma solução provê uma camada de abstração das rotinas de acesso aos dispositivos e, no caso deste trabalho, também oferece recursos para aplicar algoritmos de fusão de sensores bem como meios para inserção de novos dispositivos e algoritmos.

Este capítulo apresenta como o *middleware* foi desenvolvido, desde a concepção da arquitetura à execução dos experimentos e como os testes foram realizados.

3.1 - LEVANTAMENTO DOS SENSORES

Para a concepção do projeto, foi necessário realizar uma pesquisa e consequentemente um levantamento dos sensores a serem utilizados neste trabalho. Esta pesquisa foi realizada com base nas especificações (*datasheets*) dos sensores disponíveis no COMPSI - Computing and Information Systems Research Lab.

3.1.1 GPS EM-408

Este sensor é um módulo GPS (*Global Positioning System*) que opera seguindo os padrões do protocolo NMEA 0183 desenvolvido pela Associação Nacional de Eletrônicos Marinhos (*National Marine Electronics Association* - *NMEA*) que é uma associação que padroniza dispositivos eletrônicos utilizados em veículos náuticos.

Figura 4 - GPS EM-408 SiRF III *Receiver*

O padrão NMEA 0183 define protocolos e sinais na comunicação entre dispositivos. Este padrão suporta somente uma via de comunicação, havendo somente um transmissor, mas podem haver vários receptores. Para a transmissão dos dados, são utilizados caracteres ASCII incluindo informações como posição, profundidade, etc. As mensagens transmitidas seguem normas estabelecidas pela NMEA, sendo elas descritas abaixo:

- O caractere inicial da mensagem inicia com cifrão "$";
- Após o cifrão, duas letras indicam o dispositivo que está enviando a mensagem e logo após mais três caracteres indicando o protocolo utilizado;
- Os valores dos dados são separados por vírgula;
- Após o último dado deve haver um asterisco "*" seguido pelo *checksum* composto por 2 dígitos hexadecimais obtidos através da aplicação da lógica XOR na mensagem, do cifrão até o asterisco.

Como exemplo de uma mensagem do padrão NMEA 0183, pode-se citar a mensagem do protocolo GGA (*Global Positioning System Fixed Data)* também utilizada no GPS EM-408 conforme pode ser visualizada na figura 5:

$GPGGA,161229.487,3723.2475,N,12158.3416,W,1,07,1.0,9.0,M,,,,0000*18

Figura 5 - Mensagem padrão NMEA 0183. Protocolo GGA - Global Positioning System Fixed Data

Abaixo, segue tabela com explicação de cada campo contido na mensagem na ordem

em que aparecem.

Tabela 5 - Tabela com os campos e valores da mensagem do protocolo GGA

Nome do Campo	Valor
ID da Mensagem	$GPGGA
Horário UTC	161229.487
Latitude	3723.2475
Indicador Norte/Sul	N
Longitude	12158.3416
Indicador Leste/Oeste (E/W)	W
Indicador de Ajuste de Posição	1
Satélites Utilizados	07
Diluição da Precisão Horizontal	1.0
Altitude Acima do Nível do Mar	9.0
Unidade	M
Separação Geoidal	
Unidade	M
Tempo da Última Atualização	
ID da Estação de Referência	0000
Checksum	*18

Este protocolo foi utilizado na implementação do estudo de caso para simulação de um GPS.

3.1.2 - SONAR LV-EZ4

O sonar LV-EZ4 é um sensor que provê a detecção de objetos e medição de distância. Ele é capaz de detectar um objeto na distância entre 0 a 6,45 metros. A saída de seus sinais pode ser dos seguintes formatos: largura de pulso, saída analógica de voltagem e saída digital serial.

Figura 6 - Sonar LV-EZ4

Utilizando a comunicação serial do dispositivo, a saída obtido será o caractere ASCII “R” seguido de três dígitos representando a distância em polegadas, com valores que vão de 0 até 255, e em seguida o caractere “*carriage return*” (ASCII 13).

3.1.3 - ACELEROMETRO LIS331HH

Este sensor é capaz de detectar a variação de aceleração de um objeto nos eixos X, Y, Z.

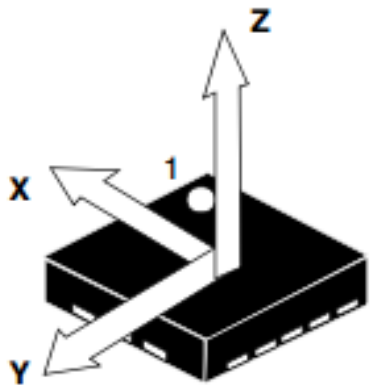

Figura 7 - Eixos de detecção da aceleração

Ele é capaz de disponibilizar a aceleração (dada em G) por meio de leitura dos valores dos bits lidos conforme tabela:

Tabela 6 - Relação dos valores de aceleração em função do valor dos bits

Bit 1	Bit 0	Aceleração
0	0	±6g
0	1	±12g
1	1	±24g

3.2 - ARQUITETURA

Com base no estudo dos sensores e no levantamento dos correlatos, foi elaborada a seguinte arquitetura ilustrada pela Figura 8. A arquitetura foi dividida em camadas, sendo que cada camada oferece um nível de abstração, esta figura ilustra o *middleware* dividido entre as camadas *Integração* e *Fusão*.

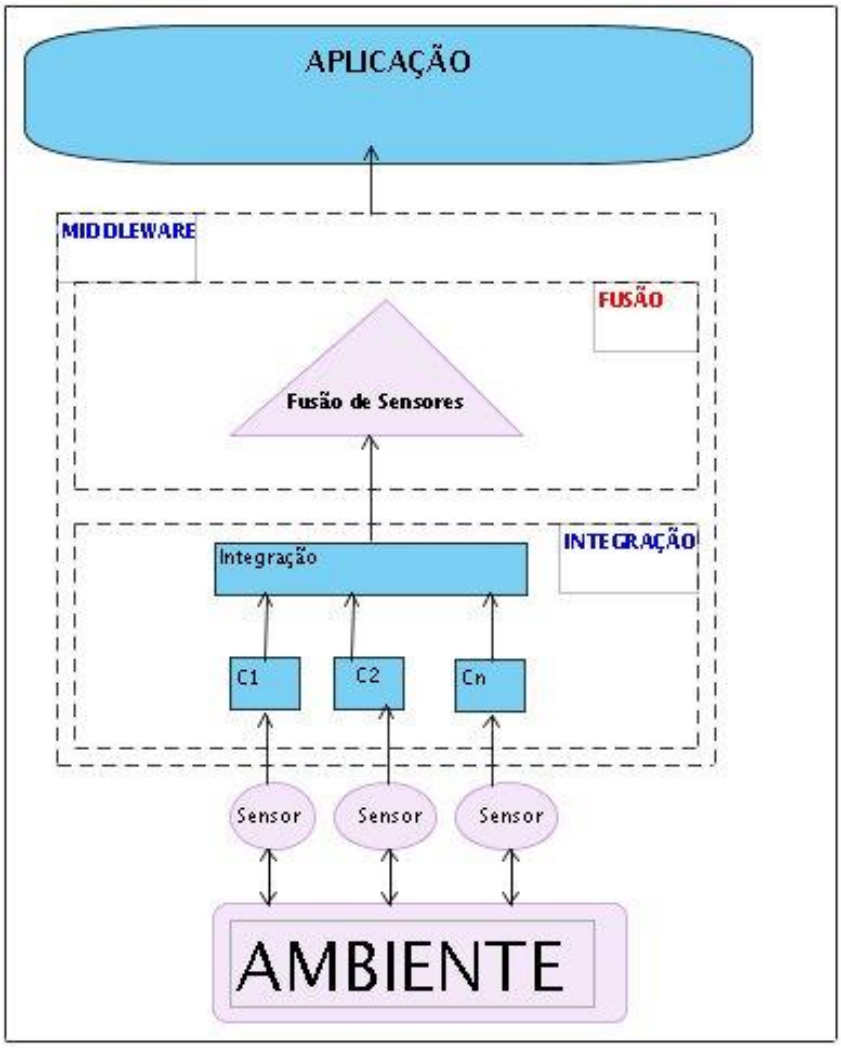

Figura 8 - Arquitetura Proposta

A camada *Integração* contém componentes que são responsáveis pela atuação com os sensores (C1, C2, Cn), abstraindo rotinas inerentes a captura, decodificação e estruturação dos dados dos sensores conforme mostrado na figura. Estes componentes são denominados controladores.

Na captura dos dados, o controlador irá efetuar a leitura dos dados dos sensores. Essa leitura não necessitam ser um dispositivo físico, podendo ser também qualquer outra fonte de dados que forneça as informações destes dispositivos como arquivos, banco de dados, outros *softwares*, etc. A captura de informações pelo controlador, se dá através do componente *SensorReader* conforme Figura 10. O *SensorReader* é uma camada que abstrai a rotina de capturar efetivamente os dados, como leitura de arquivos, comunicação serial, acesso ao banco de dados.

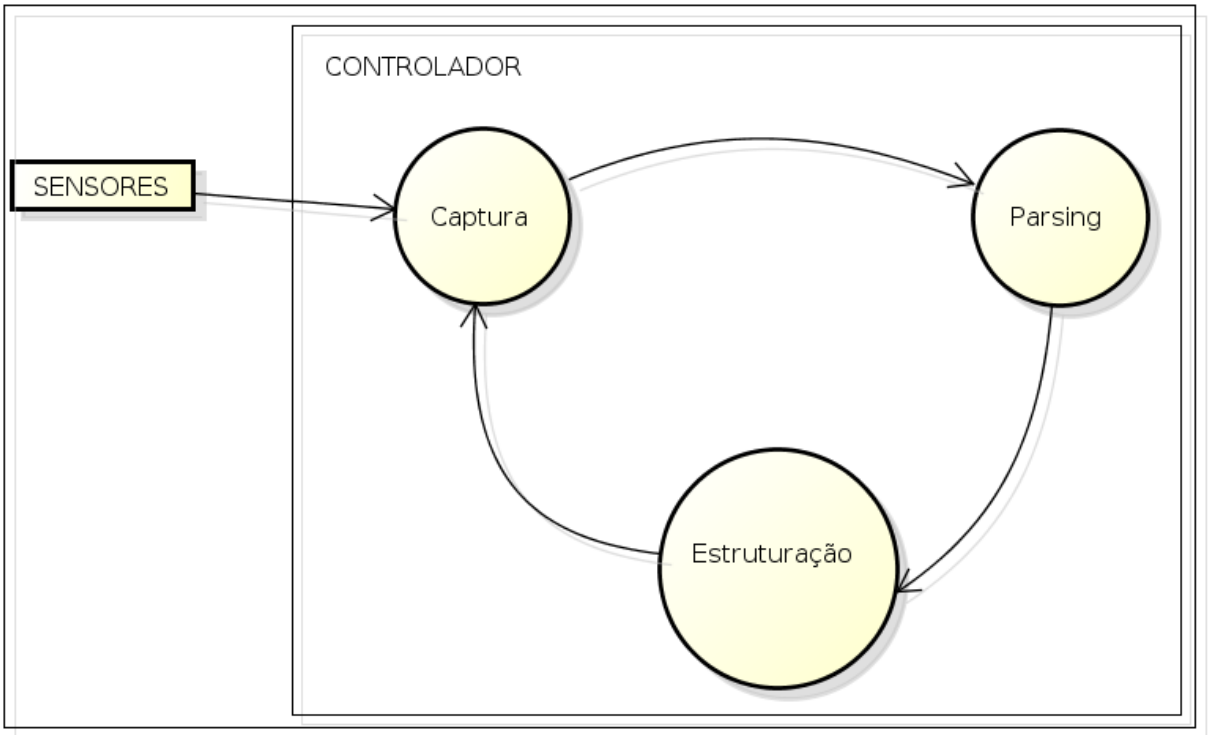

Figura 9 - Fluxo seguido pelo controlador.

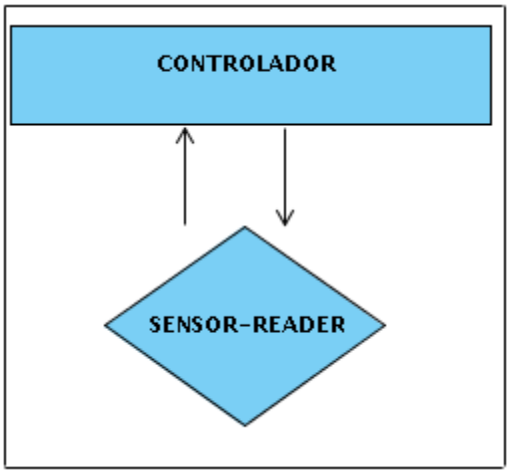

Figura 10 - Interação entre Controlador e *SensorReader*.

Alguns sensores implementam protocolos de comunicação e de estruturação de seus dados (como o GPS), por isso nas etapas de *parsing* e estruturação o controlador irá decodificar a *string* contendo os dados do sensor e estruturá-los seguindo o modelo de dados. Cada controlador é responsável por somente um sensor.

No componente Integração, como o próprio nome já diz, é onde é feita a integração dos controladores e por consequência dos dados de cada controlador. Também é por meio deste elemento que se faz acesso ao dado já estruturado pelo controlador. É possível acessar os dados de um único controlador individualmente, obtendo um conjunto de medições realizadas por ele.

Além da integração dos controladores, essa subcamada é o que faz a ligação entre a camada de *Middleware* e a camada de Fusão, fornecendo a entrada para aplicação de algoritmos de fusão. Este componente não segue um fluxo como o controlador, pois toda ação no componente é realizada pelo usuário do *middleware*.

Na camada de Fusão, é realizada a extração dos dados dos sensores que se deseja aplicar os algoritmos de fusão e também a aplicação da fusão. No modelo do sensor, podem haver mais dados do que se deseja para aplicar a fusão, um exemplo disso são os GPS's, conforme demonstrado na tabela 5, eles disponibilizam várias informações além das coordenadas. Então se o objetivo da fusão for aplicar algoritmos nas coordenadas dos GPS's, na camada de fusão somente as coordenadas (latitude e longitude) serão extraídas dos controladores para que após isso, seja aplicado o algoritmo de fusão. Por fim, após a aplicação de fusão os dados são estruturados em um modelo específico para o resultado fornecido pela fusão e disponibilizados para a aplicação.

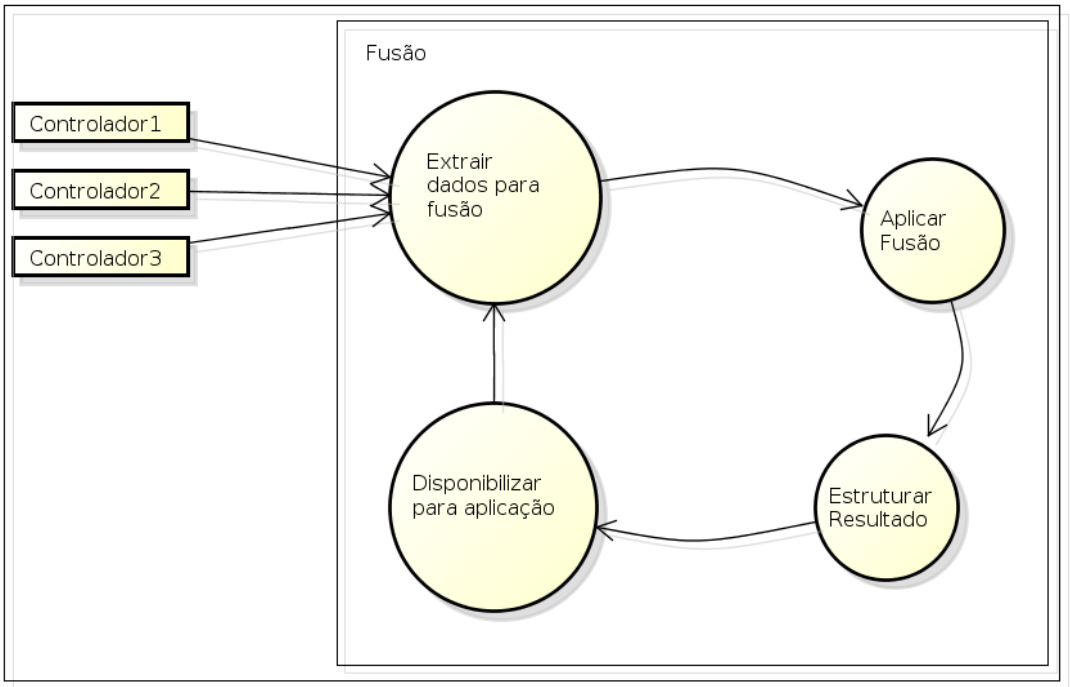

Figura 11 - Fluxo seguido na camada de fusão.

Cada instância de um *Middleware* é capaz de operar somente com sensores do mesmo tipo (somente GPS, somente acelerômetro), porém, é possível trabalhar com sensores de diferentes fabricantes. Para realizar operações com sensores de diferentes tipos, é necessário inserir um módulo denominado *TopMiddleware* conforme Figura 13.

O módulo *TopMiddleware* é responsável por efetuar a integração de várias instâncias da camada *Middleware* e realizar operações de fusão nos dados dos sensores, desta vez, provenientes de sensores de diferentes tipos.

O modo de operação da camada de fusão do *TopMiddleware* é similar à camada de fusão do *Middleware*, a diferença é que no momento da extração dos dados é possível utilizar tanto os dados estruturados pelo controlador quanto os dados já processados pela camada de fusão de cada *middleware*.

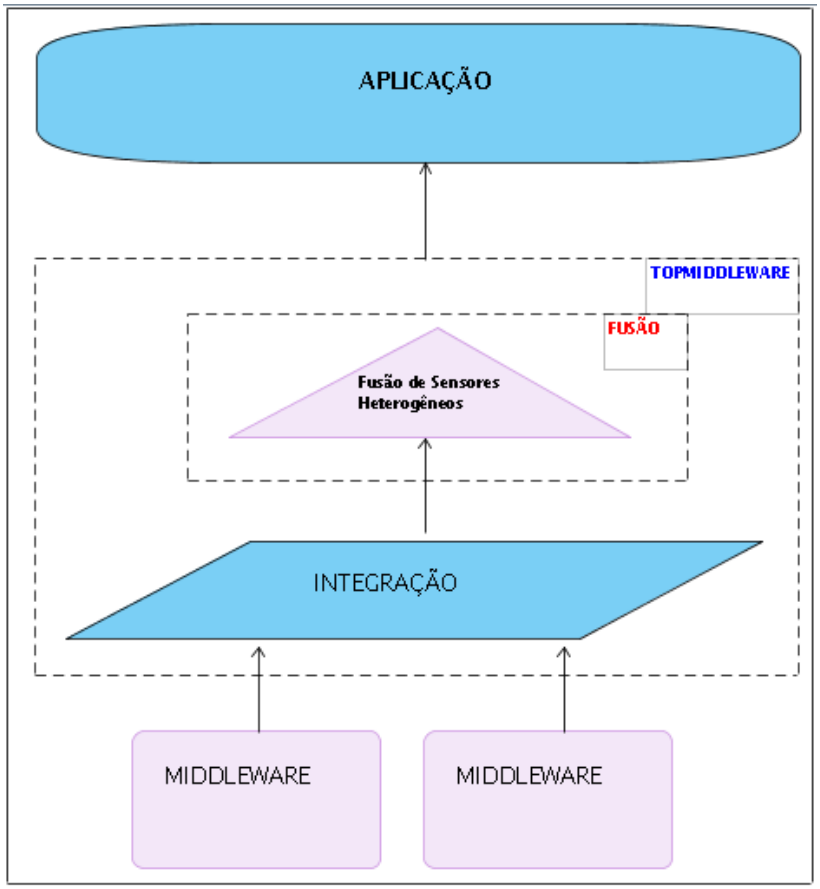

Figura 12 - Arquitetura *TopMiddleware*

Na Figura 13 e 14 a seguir, pode-se visualizar a arquitetura sintética do *Middleware* e

TopMiddleware.

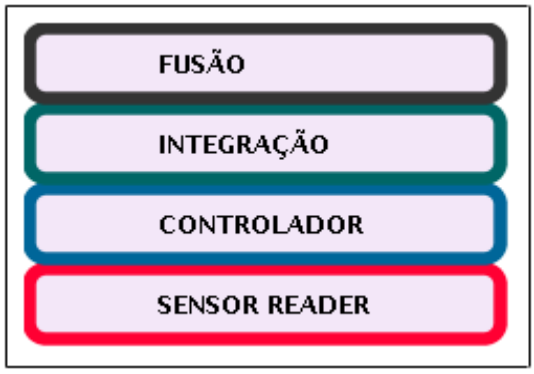

Figura 13 - Arquitetura *Middleware*

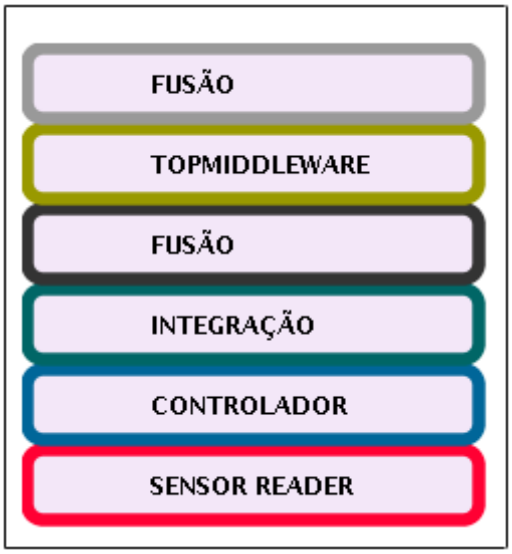

Figura 14 - Arquitetura com *TopMiddleware.*

3.3 - IMPLEMENTAÇÃO

O desenvolvimento de aplicações q fazem a utilização de sensores e a tarefa de integrá-los

Nesta seção é abordado o processo de implementação do *middleware* conforme arquitetura descrita na seção anterior. Portanto, será abordada a estrutura da arquitetura do *Software* com o diagrama de classes, organização dos pacotes e funcionamento.

O *middleware* foi desenvolvido utilizando linguagem de programação Java, JRE e JDK versão 1.7.0_25.

3.3.1 - ESTRUTURA DOS PACOTES

Para melhor organização do projeto, as classes implementadas no desenvolvimento do *middleware* foram organizadas em pacotes conforme Figura 15 e, em seguida, serão mostrados os pacotes com suas respectivas classes desenvolvidos.

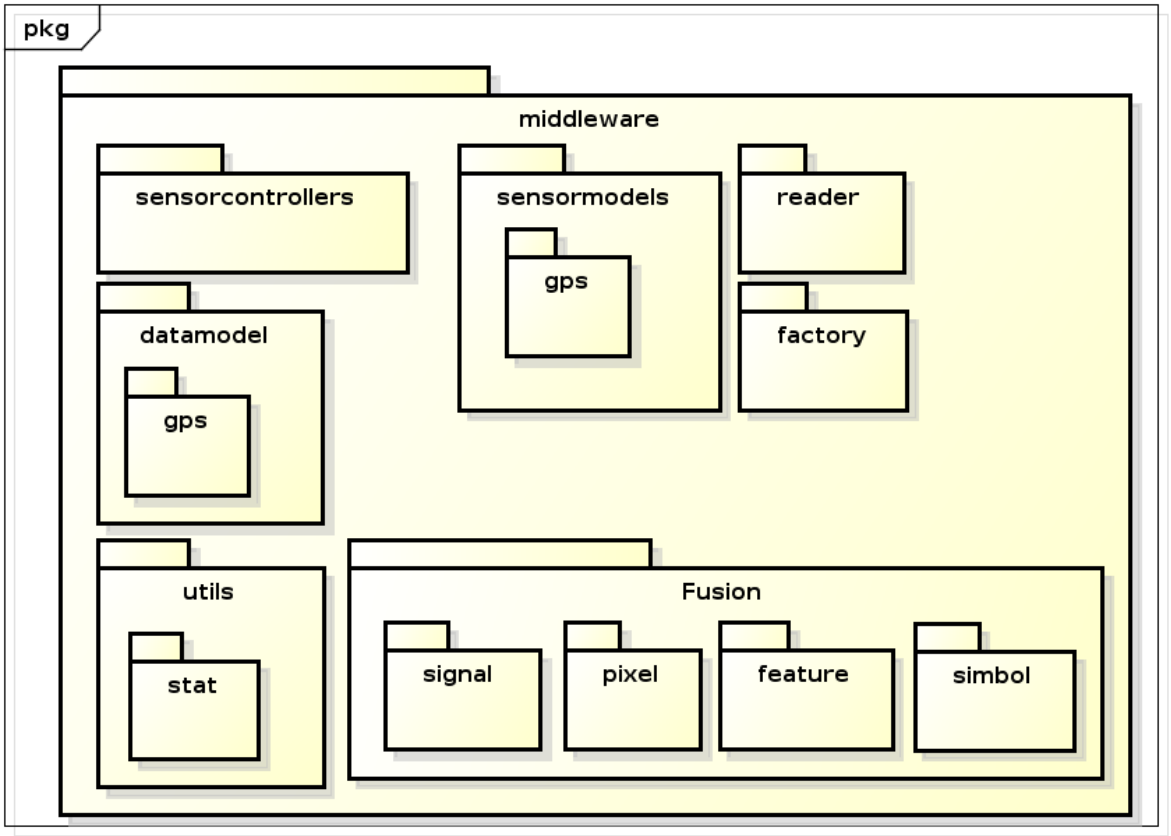

Figura 15 - Organização dos pacotes

O pacote *middleware* é o pacote principal onde está contido todo o projeto, e também as classes *Middleware* e Top*Middleware* conforme Figura 16. A classe *Middleware* é responsável por iniciar a captura de dados, integrar as classes controladoras, acessar as medições dos dispositivos e também é nela onde é definido o modelo para os dados da fusão bem como o algoritmo de fusão utilizado. A classe Top*Middleware* tem a função de integrar as classes *Middleware* e é definido o modelo para os dados da fusão bem como o algoritmo de fusão utilizado.

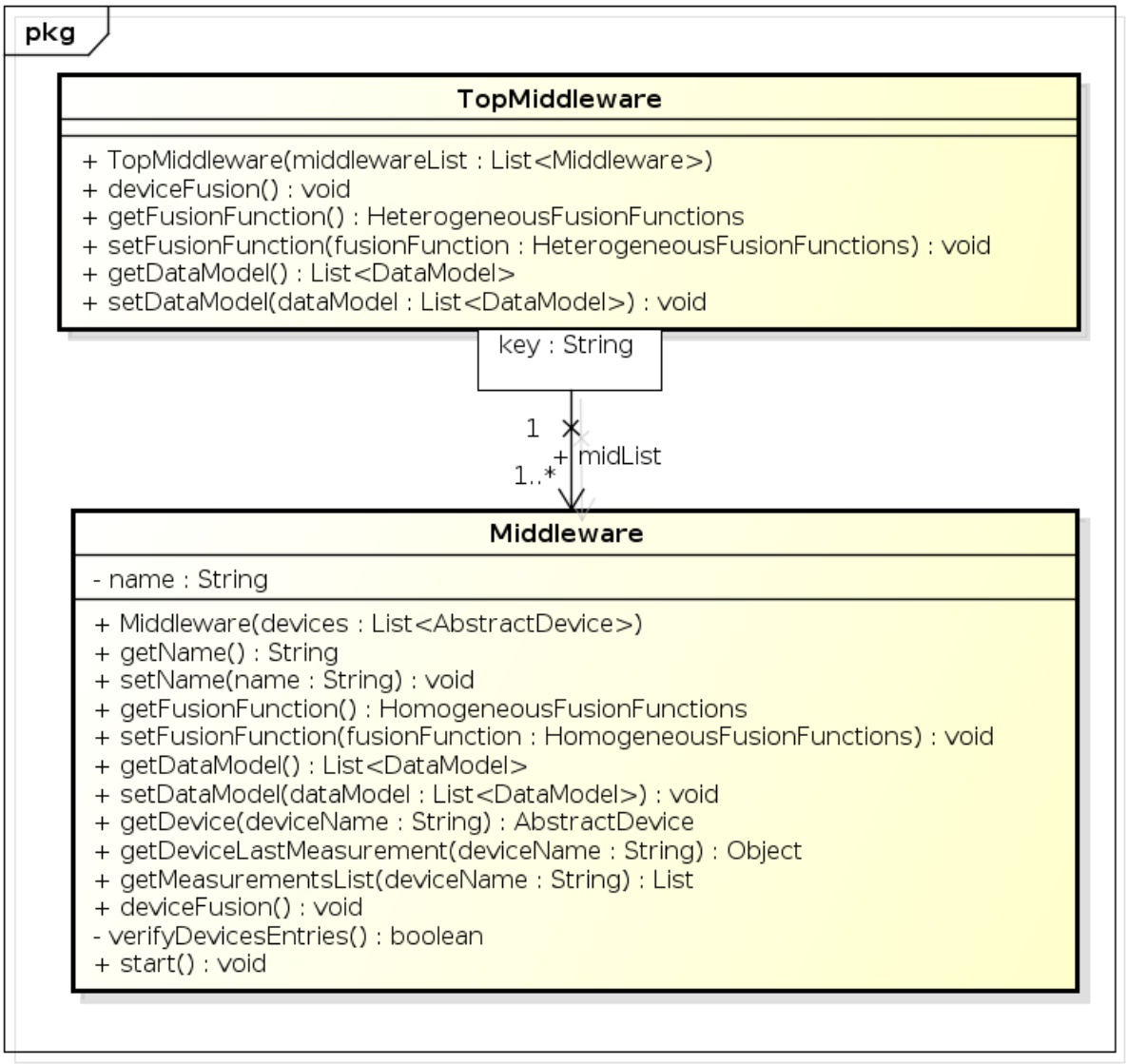

Figura 16 - Classes do pacote *middleware*

O pacote *sensorcontrollers* é responsavel por conter as classes que farão o papel do controlador dos dispositivos, que efetuarão os processos de decodificação e estruturação do dado no modelo do sensor. Toda classe controladora deve extender da classe abstrata *AbstractDevice* que implementam métodos e possui atributos necessários para o controle dos dispositivos. A classe *GPGGA* é um exemplo de classe controladora (ver Figura 17). Esta classe é responsável pela decodificação e estruturação dos dados provenientes de GPS's que implementam o protocolo GGA do padrão NMEA 0183.

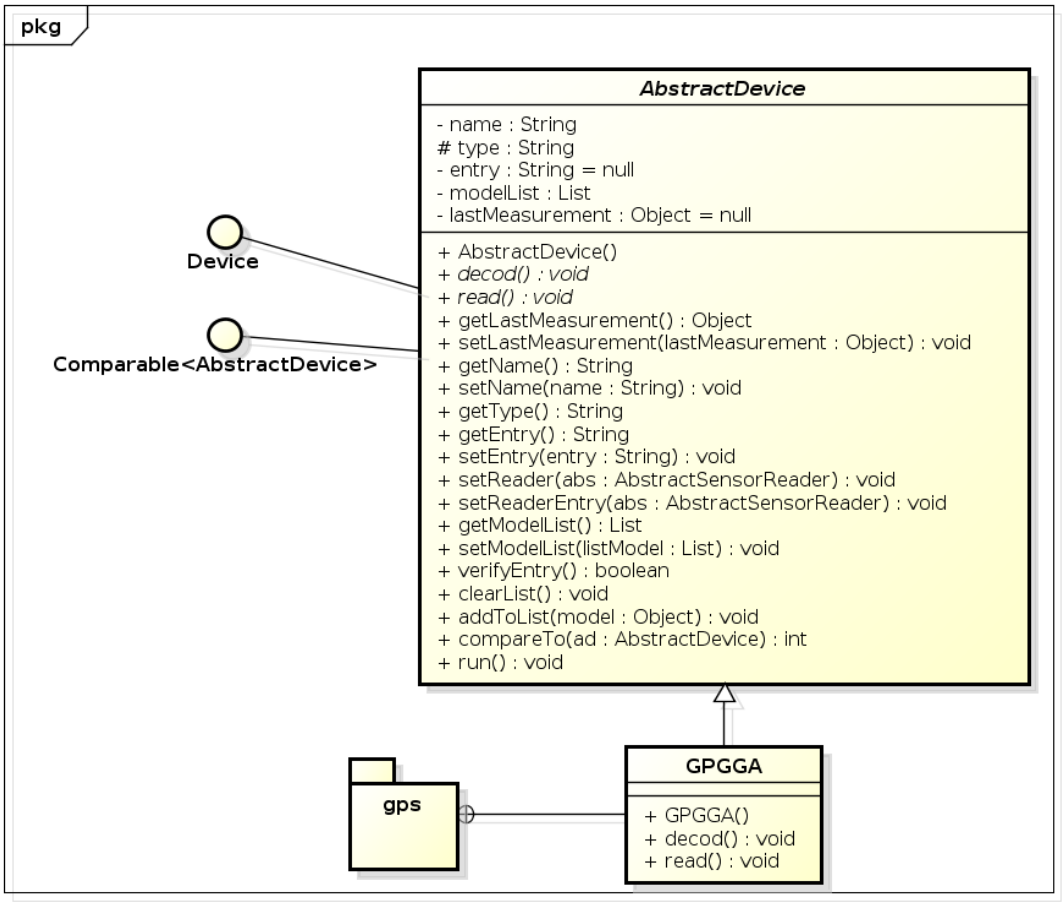

Figura 17 - Classes e *interface* do pacote *sensorcontrollers*

O pacote *sensormodels* é o pacote onde estão as classes modelo de cada sensor. A classe modelo de um sensor é construída com base nas informações que os dispositivos fornecem. Para cada sensor diferente implementado, é recomendado criar um pacote específico dentro de *sensormodels*. Um exemplo disso é o pacote *middleware.sensormodels.gps* onde ficam as classes referente aos GPS. Na Figura 16 estão as classes referentes ao modelo de um GPS. A classe *GenericGpS* persiste apenas latitude e longitude, que são informações providas pela grande maioria dos GPS. Portanto a classe *GPGGAModel* extende de *GenericGpS* e persiste as informações do protocolo GGA do padrão NMEA 0183. É recomendado que as demais classes que persistam informações de GPS's, extendam de *GenericGpS*.

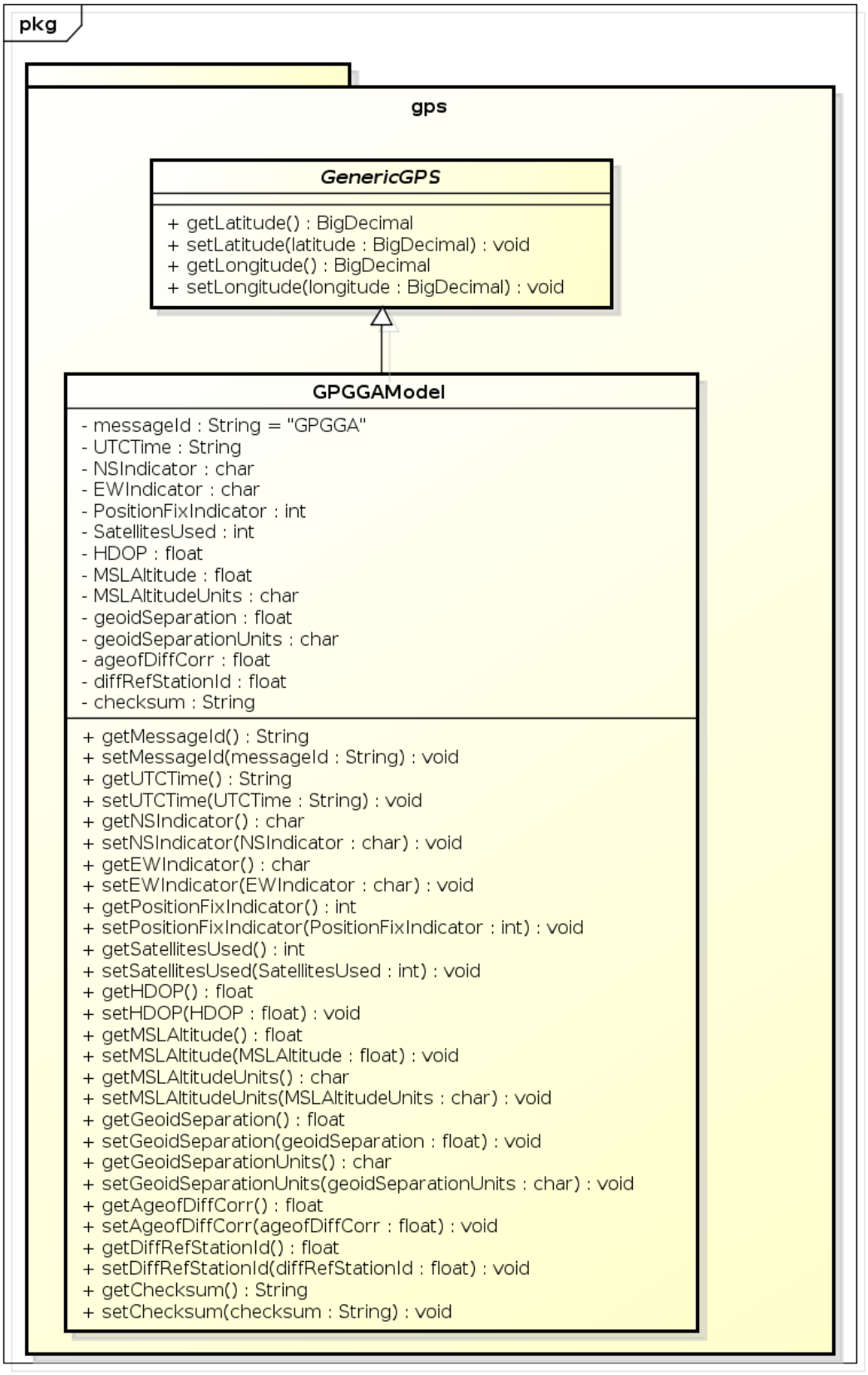

Figura 18 - Classes do pacote *sensormodels*.

O pacote *reader*, contém classes responsáveis por realizar a captura dos dados dos dispositivos. A fonte dos dados não presicam ser necessariamente o dispositivo físico, mas pode ser também arquivos, banco de dados, etc. As classes deste pacote devem extender da classe abstrata *AbstractSensorReader*. Na Figura 19, a classe *FileSensorReader* faz leitura de dados dos sensores que estejam armazenadas em arquivos de texto.

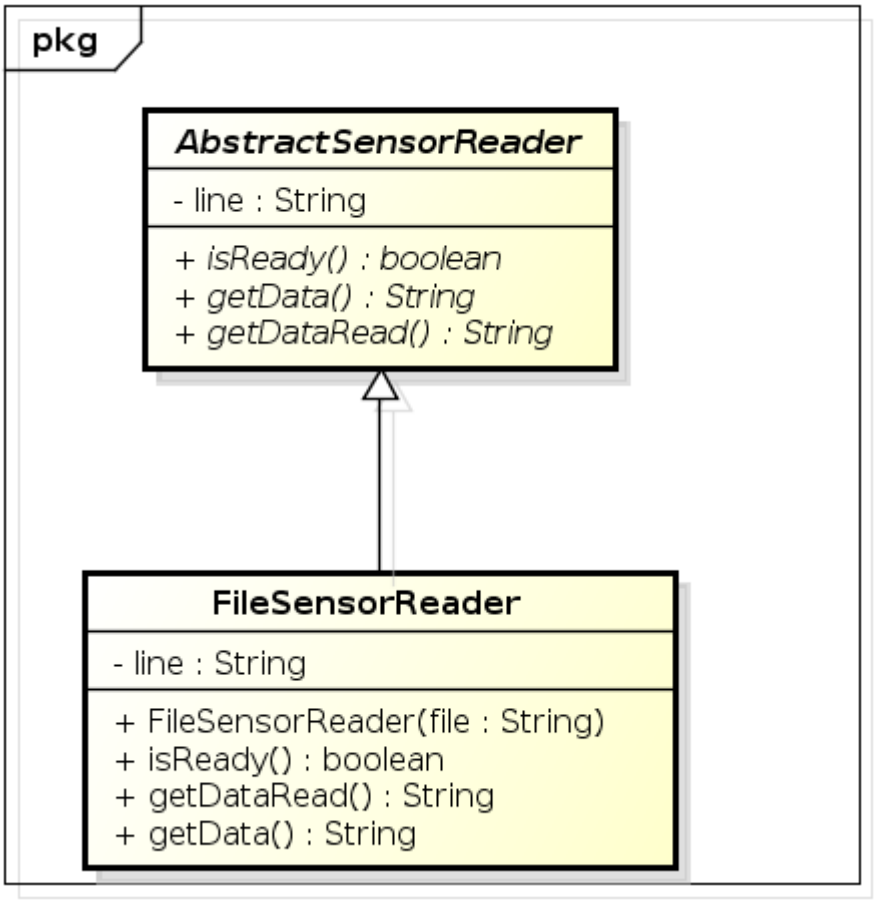

Figura 19 - Classes do pacote *reader*.

O pacote *factory* é composto somente de uma classe que implementa o padrão de projeto *Factory*, ele é utilizado para facilitar a instância de novos controladores. Apenas passando o nome do controlador, a classe já retorna uma instância do controlador requisitado.

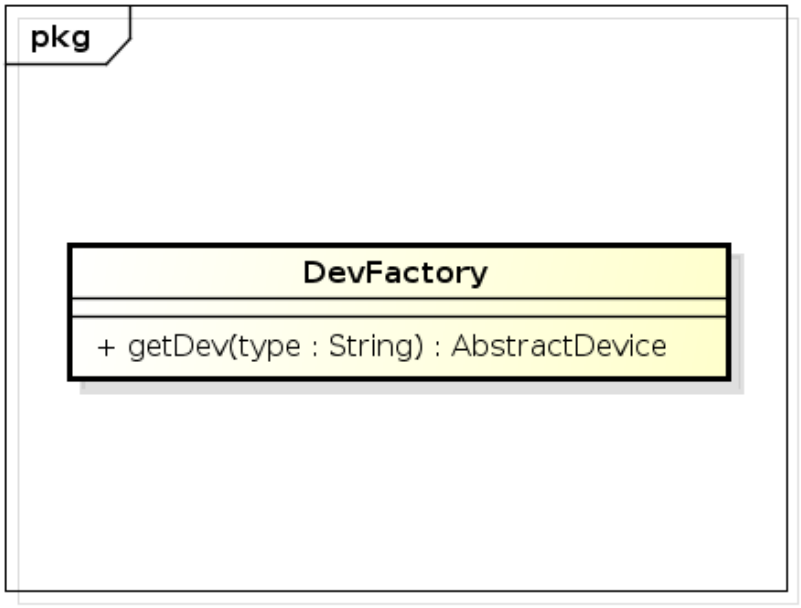

Figura 20 - Classe do pacote *factory*.

As classes de persistência (modelo) dos dados resultantes dos processos de fusão, ficam no pacote *DataModel*. Para cada dispositivo, podem haver vários tipos de modelagem para os dados resultantes da fusão, já que dependendo do algoritmo, o dado resultante pode ser diferente. Por isso, é recomendada a criação de um pacote para cada dispositivo. Como exemplo, no pacote *middleware.datamodel*.gps se encontram os modelos referentes aos GPS's, como exemplo, a classe *GPSCoordsDataModel* persiste os dados de um processo de fusão que tem como resultado latitude e longitude. Cada modelo deve implementar a *interface DataModel*.

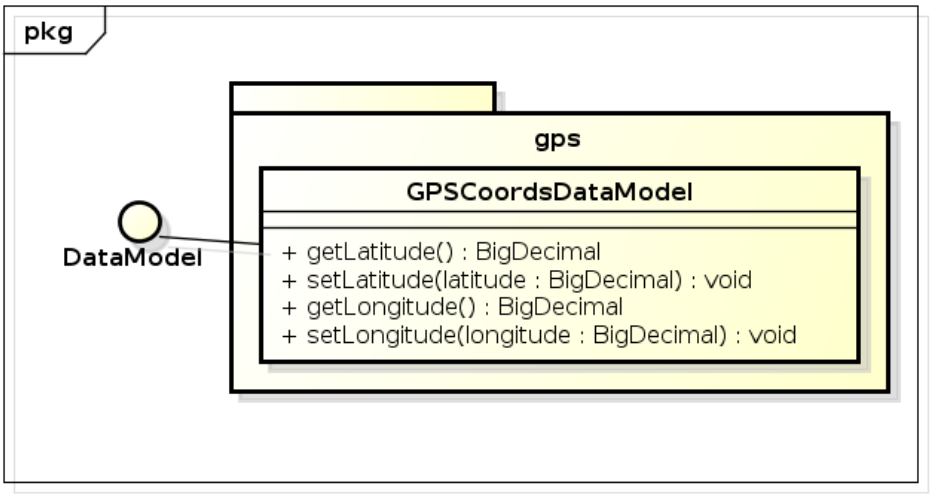

Figura 21 - Classe do pacote *DataModel* e *DataModel*.gps

Os algoritmos de fusão de fusão de sensores estão no pacote *Fusion* e estão divididos dentre os pacotes *Fusion.signal*, *Fusion.pixel*, *Fusion.feature* e *Fusion.simbol* que representam os níveis de fusão de dados. Dentro de cada sub-pacote esta contido os *enums* onde relacionam os processos de extração dos dados do modelo dos sensores e execução dos algoritmos de fusão. O *enum HeterogeneousFusionFunctions* contém algoritmos para fusão de sensores heterogêneos (tipos diferentes de sensores) e o *enum HomogeneousFusionFunctions* algoritmos para fusão de sensores homogêneos (tipos iguais de sensores). Na Figura 21 os *enums HomogeneousFusionFunctions HeterogeneousFusionFunctions* implementam cada um as *interfaces* Run*MiddlewareFusion* e RunTop*MiddlewareFusion* respectivamente. Estas *interfaces* têm o método *run* que é chamado para iniciar o processo de fusão. Os algoritmos se encontram no pacote *algorithm* dentro de seus respectivos pacotes de cada nível de fusão.

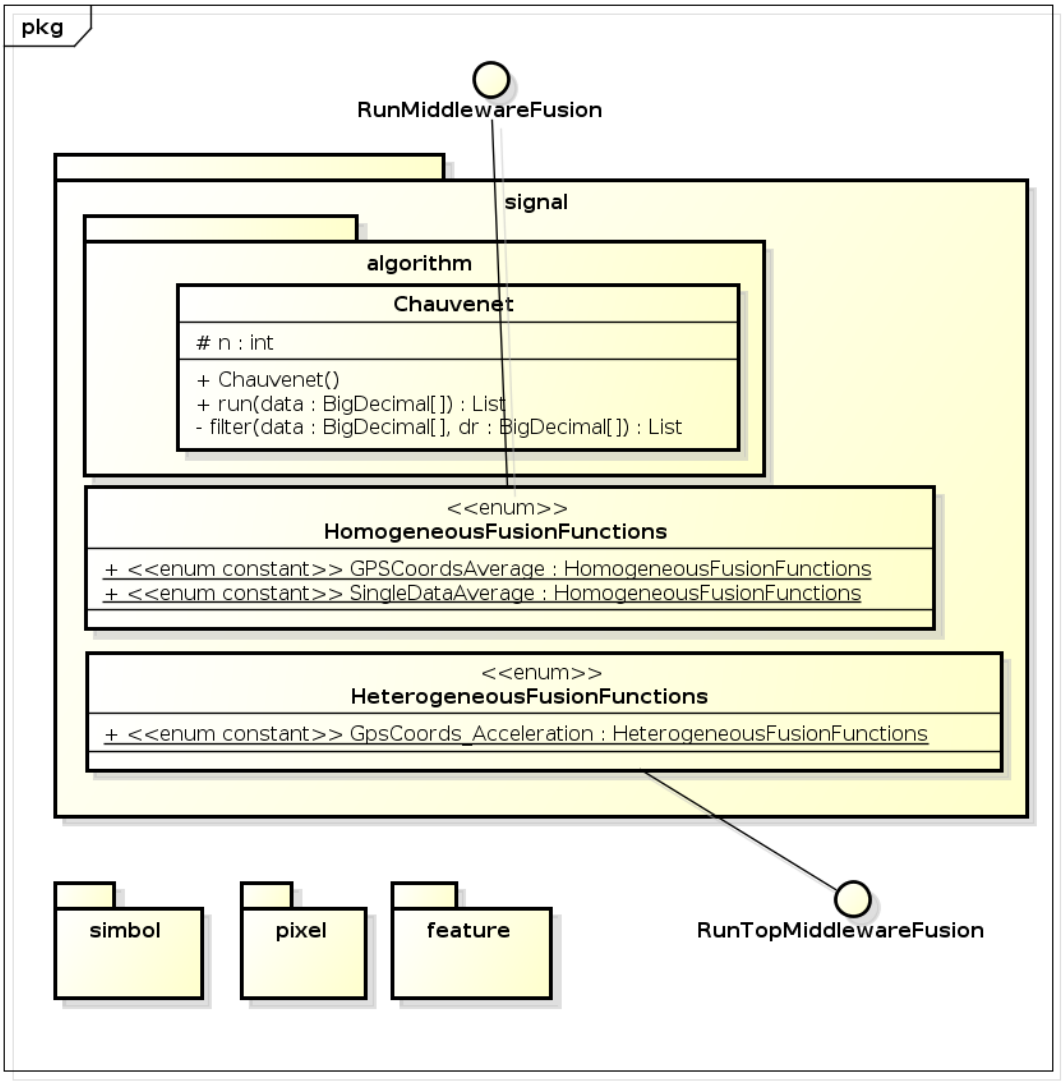

Figura 22 - Classes do pacote *Fusion*.

O pacote *utils* é o pacote destinado às classes auxiliares para cálculos. Por exemplo, o pacote *middleware.utils.stat* se refere ao pacote onde contém classes para operações de cálculos estatísticos como média, desvio padrão, variância. A classe *BigDecimalUtils* foi implementada a fim de efetuar cálculos com o tipo *BigDecimal* do java (pacote java.math). A utilização deste tipo foi necessário para substituir o tipo *double* das variáveis utilizadas nos cálculos. O tipo *double* apresentou vários erros de arredondamento dos valores decimais e a solução encontrada foi a utilização do tipo *BigDecimal*.

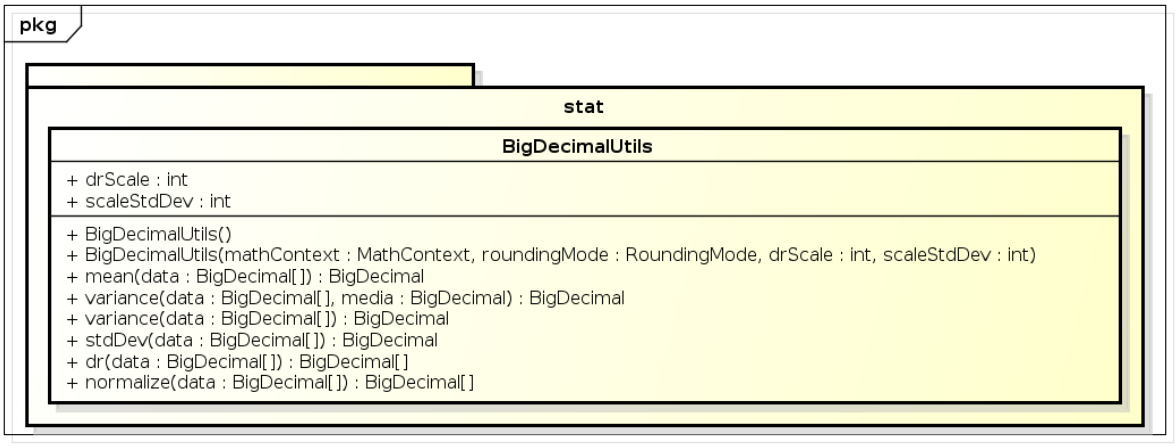

Figura 23 - Pacote *utils* e classe do pacote *middleware.utils* .stat.*BigDecimalUtils*

Um exemplo típico do erro acometido pelo uso de variáveis *double*, pode ser demonstrado na soma de dois valores decimais. Considerando duas variáveis do tipo *double*, n1 e n2, e sendo seus valores 0,1 e 0,2 respectivamente. Ao realizar a soma destas duas variáveis, o resultado desta operação será 0,30000000000000004 e não o desejável 0,3. Ao executar diversas operações aritméticas utilizando double, este erro que a princípio pode parecer pequeno, irá apresentar erros significativos por causa dos arredondamentos.

3.4 - FUNCIONAMENTO

Nesta seção serão descritos o modo de funcionamento do *middleware* proposto. Serão detalhados os processos de funcionamento, configuração, inserção de novos sensores e novos algoritmos de fusão.

3.4.1 - CONFIGURAÇÃO

Para utilização do *middleware* desenvolvido, é necessário realizar sua configuração previamente. As etapas de configuração são as seguintes:

1. Instância dos controladores;
2. Definição de um identificador para cada controlador;
3. Definir a entrada para leitura dos dados para cada controlador;
4. Definir classe responsável pela captura dos dados para cada controlador;
5. Instanciar *middleware* e adicionar controladores no mesmo*;*
6. Definir algoritmo de fusão;
7. Iniciar captura dos dados.

Os exemplos de códigos das configurações serão baseados na configuração do *middleware* para trabalhar com GPS.

3.4.1.1 - INSTÂNCIA DOS CONTROLADORES

A instância dos controladores define quais dispositivos serão utilizados. A instância pode ser feita através da utilização da classe Dev*Factory*, que implementa o padrão de projeto *Factory*, ou pela instanciação normal da classe controladora (ver Figura 24 e 25). Lembrando que para realizar este processo atraves da classe Dev*Factory*, antes deve-se instanciá-la.

```
DevFactory devF = new DevFactory();

GPGGA dev1 = (GPGGA) devF.getDev("GPGGA");
```

Figura 24 - Instância utilizando padrão de projeto *Factory*.

```
GPGGA dev1 = new GPGGA();
```

Figura 25 - Instância normal da classe controladora.

Como visto na Figura 24, no momento da instancia do controlador pela classe Dev*Factory* é feito um *casting* para o tipo GPGGA. Isso ocorre pois a classe Dev*Factory* trabalha com diversos tipos de dispositivos, então seu retorno uma instância mais genérica abstraindo a classe instanciada (GPGGA) para sua superclasse (classe *AbstractDevice*).

3.4.1.2 - DEFINIÇÃO DO IDENTIFICADOR

Um controlador é identificado no *middleware* atraves de seu nome e, é a partir deste identificador que é possível fazer o acesso aos dados do controlador (Figura 26). Para nomear o controlador, basta fazer uma chamada ao metodo *setName* passando como parâmetro o nome a ser dado.

```
dev1.setName("GpsRed");
```

Figura 26 - Definindo o nome “GpsRed” para o controlador.

3.4.1.3 - DEFINIR ENTRADA E CLASSE LEITORA DE DADOS

A classe leitora é responsável por efetuar a captura de dados que estão contidos no local definido pela entrada de dados. Portanto deve-se definir o local onde o dado será lido e instanciar a classe leitora. Na Figura 27, é demonstrado a instância de uma classe que fará a leitura dos dados contidos em arquivo de texto.

```
dev1.setEntry("/home/Devices/gpgga");
FileSensorReader sensorReader1 = new FileSensorReader(dev1.getEntry());
dev1.setReaderEntry(sensorReader1);
```

Figura 27 - Definindo caminho para arquivo e instância da classe leitora

A princípio é definido a fonte dos dados para o controlador que também é passado no momento da instância da classe *FileSensorReader*. Após isso, o leitor é adicionado no controlador.

3.4.1.4 - INSTANCIAR *MIDDLEWARE* E ADICIONAR CONTROLADORES

Para o *middleware* reconhecer os controladores, é necessário que estes sejam adicionados. Para isso, é necessário instanciar o *middleware* e adicionar os controladores conforme Figura 28 e 29.

```
LinkedList<AbstractDevice> devList = new LinkedList<>();
devList.add(dev1);
/*lista dos dispositivos sendo passada no
 *construtor para inicializar o middleware
 */
Middleware mid = new Middleware(devList);
```

Figura 28 - Instância do *middleware* e adição do controlador

```
LinkedList<AbstractDevice> devList = new LinkedList<>();
devList.add(dev1);
devList.add(dev2);
devList.add(dev3);

/*lista dos dispositivos sendo passada no
 *construtor para inicializar o middleware
 */
Middleware mid = new Middleware(devList);
```

Figura 29 - Instância do *middleware* e adição dos controladores

No caso do exemplo da Figura 28, existem três controladores a serem adicionaodos no *middleware*. Tanto no caso de existir somente um controlador, quanto no caso de existirem vários, deve-se adicionar em uma lista para posteriormente passar a lista no construtor da classe *Middleware*

3.4.1.5 - DEFINIR ALGORITMO DE FUSÃO

Para definir o algoritmo de fusão, deve-se utilizar dos algoritmos listados no *enum* *HomogeneousFusionFunctions* ou *HeterogeneousFusionFunctions*.

```
mid.setFusionFunction(middleware.Fusion.signal.
                      HomogeneousFusionFunctions.GPSCoordsAverage);
```

Figura 30 - Definindo algoritmo de fusão.

No exemplo da Figura 30, está sendo definido o algoritmo *GPSCoordsAverage* como algoritmo do processo de fusão de dados.

3.4.1.6 - DEFININDO *TOPMIDDLEWARE*

Esta etapa não é obrigatória, sendo somente necessária se houver utilização de sensores de tipos diferentes.

Após configuração dos *middlewares* eles devem ser adicionados em uma lista e esta deve ser passada como parâmetro no construtor do *TopMiddleware*. Também deve ser definido um identificador para cada *Middleware*.

```
/*
 * Definindo identificador para os middlewares
 */
Middleware mid = new Middleware(devList);
mid.setName("mid");
Middleware mid1 = new Middleware(devList1);
mid1.setName("mid1");
Middleware mid2 = new Middleware(devList2);
mid2.setName("mid2");

LinkedList<Middleware> midList = new LinkedList<>();
midList.add(mid);
midList.add(mid1);
midList.add(mid2);
/*
 * Vinculando middleware ao TopMiddleware
 */
TopMiddleware tpMid = new TopMiddleware(midList);
```

Figura 31 - Configuração do Top*Middleware*

Após vinculação, deve-se definir o algoritmo de fusão ao Top*Middleware* (Figura 31).

```
tpMid.setFusionFunction(middleware.Fusion.signal.
                        HeterogeneousFusionFunctions.GpsCoords_Acceleration);
```

Figura 32 - Definindo algoritmo de fusão no Top*Middleware*.

3.4.1.7 - INICIANDO *MIDDLEWARE*

Após toda configuração o *middleware* pode iniciar os controladores a fezerem a captura dos dados. A captura dos dados foi desenvolvido utilizando *threads* então, no momento de início do *middleware*, as threads são iniciadas (Figura 33).

```
try{
    mid.start();
}catch(Exception ex){
    System.out.println(ex.getMessage());
    System.exit(1);
}
```

Figura 33 - Iniciando *middleware*.

3.4.2 - UTILIZAÇÃO DOS RECURSOS DISPONÍVEIS

O *middleware* oferece meios facilitadores para captura das informações dos sensores bem como para inserção de novos componentes e algoritmos de fusão. Esta seção está dividida nos seguintes assuntos:

- Etapa para definição de nova classe leitora de dados;
- Etapas para definição de novos sensores;
- Etapas para definição de novos algoritmos de fusão;
- Etapas para obter as informações dos sensores;
- Executar a fusão dos dados.

Os exemplos de código exibidos, serão baseados no sensor GPS utilizando protocolo GGA do padrão NMEA 0183.

3.4.2.1 - DEFINIÇÃO DE NOVA CLASSE LEITORA

A captura dos dados dos sensores é responsabilidade das classes leitoras. São nas operações destas classes que estão definidas as operações de acesso à fonte de dados, podendo ser o próprio sensor, arquivos de texto, banco de dados, etc. Nesta seção será descrito como exemplo, a definição de uma classe cujo nome será *FileSensorReader*, que fará a captura dos dados dos sensores que se encontram em arquivos de texto.

Como descrito anteriormente, toda classe leitora deve extender de *AbstractSensorReader* e sobrescrever os métodos abstratos.

Para manipulação dos arquivos, será utilizada duas classes disponíveis pelo Java, *FileReader* e *BufferedReader*.

1. **Construtor da Classe:** No construtor da classe será definido os processos iniciais da classe, como configuração e instância das classes que farão a manipulação do arquivo conforme figura 34. Como parâmetro, o construtor receberá o caminho para acesso do arquivo.

```
public FileSensorReader(String file){
    try {
       in = new BufferedReader(new FileReader(file));

    } catch (FileNotFoundException ex) {
        System.out.println(ex.getMessage());
    }
}
```

Figura 34 - Construtor da classe *FileSensorReader*.

2. **Método *isReady*:** Indica se a fonte de dados está pronta para leitura. No caso dos arquivos, indica se o *buffer* não está vazio ou se a próxima linha já pode ser lida.

```
@Override
public boolean isReady(){
    try{
        return in.ready();
    }catch(IOException ioEx){
        System.out.println(ioEx.getMessage());
        return false;
    }
}
```

Figura 35 - Sobrescrita do método *isReady*

3. **Método *getData*:** Este método irá efetuar a leitura do dado no dispositivo (fonte de dados). No caso dos arquivos, este método irá retornar a linha lida no arquivo e será persistida no atribudo *line* da classe.

```
@Override
public String getData() {
    try {
       if(isReady()){
           this.line = in.readLine();
       }

    } catch (FileNotFoundException ex) {
        System.out.println(ex.getMessage());
    }catch(IOException ioEx){
        System.out.println(ioEx.getMessage());
    }
    return line;
}
```

Figura 36 - Sobrescrita do método *getData*.

4. **Método *getDataRead*:** Este método irá retornar o dado atual lido para o controlador.

```
@Override
public String getDataRead() {
    return line;
}
```

Figura 37 - Sobrescrita do método *getDataRead*

3.4.2.2 - DEFINIÇÃO DE NOVOS SENSORES

Na arquitetura desenvolvida, caso o desenvolvedor não encontrar o sensor desejado para sua aplicação, é permitido a criação de toda estrutura de classes necessárias para operar com o novo dispositivo. As etapas incluem os processos de criação da classe modelo do dispositivo e classe controladora.

A classe modelo deve possuir atributos referentes aos dados fornecidos pelo sensor. Com base na tabela 5 a classe *GenericGPS* e *GPGGAModel* foi implementada. A classe *GenericGPS* possui atributos utilizados pela maioria dos GPS's e protocolos e a classe *GPGGAModel* extende dela (ilustração dos métodos *getters* e *setters* foram omitidos da Figura 39).

```java
public abstract class GenericGPS{
    private BigDecimal  latitude;
    private BigDecimal  longitude;
    public BigDecimal getLatitude() {
        return latitude;
    }

    public void setLatitude(BigDecimal latitude) {
        this.latitude = latitude;
    }

    public BigDecimal getLongitude() {
        return longitude;
    }

    public void setLongitude(BigDecimal longitude) {
        this.longitude = longitude;
    }

}
```

Figura 38 - Implementação da classe *GenericGPS*.

```java
public class GPGGAModel extends GenericGPS {
    private String  messageId = "GPGGA";
    private String  UTCTime;

    private char    NSIndicator;
    private char    EWIndicator;
    private int     PositionFixIndicator;
    private int     SatellitesUsed;
    private float   HDOP;
    private float   MSLAltitude;
    private char    MSLAltitudeUnits;
    private float   geoidSeparation;
    private char    geoidSeparationUnits;
    private float   ageofDiffCorr;
    private float   diffRefStationId;
    private String  checksum;

    .
    .
    .
}
```

Figura 39 - Implementação da classe *GPGGAModel* (atributos).

Para a definição de uma nova classe controladora, deve-se extender de *AbstractDevice* e implementar os métodos *decod* e *read* e sobrescrever os métodos se necessário.

O primeiro passo é implementar o método construtor e read. No método construtor deve-se chamar o construtor da superclasse, definir o atributo type e o atributo *accuracy* que

define a precisão do dispositivo. Em aplicações que fazem o uso de dispositivos com precisão diferente, o atributo *accuracy* pode ser utilizado nos algoritmos de fusão para estabelecer um peso maior nos dispositivos de maior precisão. Já o método read fará chamadas à classe leitora para obter uma nova medição e após isso irá executar o método *decod*.

```
public  class GPGGA extends AbstractDevice{
    public GPGGA(){
        super();
        type = "GPGGA";
        accuracy = new BigDecimal("10");
    }
```

Figura 40 - Implementação do construtor.

```
@Override
public void read() {
    sr.getData();
    decod();
}
```

Figura 41 - Implementação do método *read*.

O próximo passo é implementar o método *decod* que é responsável por fazer o *parsing* da *string* vinda do sensor. Neste método deve-se obter a medição do sensor, efetuar o *parsing* para estruturar os dados na classe modelo (no caso será a classe *GPGGAModel).*

```
@Override
public void decod(){

    //Captura última medição
    String line = sr.getDataRead();

    line = line.replace("*", ",");
    //Parsing da String
    String[] lineSplited = line.split(",");
    // Instancia Modelo
    GPGGAModel gpsModel = new GPGGAModel();
    try{

        /*Parsing e atribuir linha lida*/
        gpsModel.setMessageId( lineSplited[0].trim());
        gpsModel.setUTCTime(lineSplited[1].trim());
        gpsModel.setLatitude(new BigDecimal(lineSplited[2].trim()) );
        gpsModel.setNSIndicator(lineSplited[3].charAt(0));
        gpsModel.setLongitude(new BigDecimal(lineSplited[4].trim()));
        gpsModel.setEWIndicator(lineSplited[5].charAt(0));

        .
        .
        .
```

Figura 42 - Implementação do método *decod.*

Após o *parsing*, deve-se adicionar o modelo na lista de medições e atribuir para a variável *lastMeasurement* esta medição por meio dos métodos *setLastMeasurement* e *addToList*.

```
    setLastMeasurement(gpsModel);
    addToList(gpsModel);
}catch(NumberFormatException ex){
    System.out.println("Erro ao ler dados do dispositivo");
}
```

Figura 43 - Adicionando medição na lista e no atributo.

Após estes dois passos, o desenvolvedor deve adicionar o dispositivo na classe *DevFactory.*

```
public class DevFactory {
    public Device dispositivo;
//    retorna controlador do dispositivo
    public AbstractDevice getDev (String type){
        AbstractDevice dispositivo = null;
        switch(type){
            case "GPGGA":
                dispositivo = new GPGGA();

            break;

            default: dispositivo = null;

        }
        return dispositivo;

    }
}
```

Figura 44 - Adicionando novo dispositivo na classe Dev*Factory*

3.4.2.3 - DEFINIÇÃO DE NOVOS ALGORITMOS DE FUSÃO

Na definição de novos algoritmos de fusão, o desenvolvedor deve implementar o algoritmo no pacote *algorithm* contido dentro dos pacotes *middleware.Fusion.signal, middleware.Fusion.pixel, middleware.Fusion.features, middleware.Fusion.simbol.* O

desenvolvedor deve escolher o nível de fusão que deseja utilizar e implementar a tratativa dos dados no método *run* em um dos *enums* (*HeterogeneousFusionFunctions* e *HomogeneousFusionFunctions*) e após isso, executar o algoritmo de fusão.

Como exemplo, será demonstrado o algoritmo que efetuará a tratativa dos dados na fusão das coordenadas de três GPS's e aplicação do critério de Chauvenet.

A princípio, deve-se recuperar os dispositivos do *Middleware* e suas respectivas medições. Deve-se também instanciar a classe modelo que armazenará o resultado da fusão das coordenadas.

```
public enum HomogeneousFusionFunctions implements RunMiddlewareFusion {

    GPSCoordsAverage(){
        @Override
        public void run(Middleware mid2){
```

Figura 45 - Criando método no *enum*.

```
 //captura  dispositivo
AbstractDevice gps = gpgga.get(j);
//captura medições do dispositivo
GenericGPS genericGps = (GenericGPS) gps.getLastMeasurement();
//Instancia DataModel
GPSCoordsDataModel dtModel = new GPSCoordsDataModel();
//passar lat e lng para lista de CoordsDataModel
dtModel.setLatitude(genericGps.getLatitude());
dtModel.setLongitude(genericGps.getLongitude());


lat[j] = genericGps.getLatitude();
lng[j] = genericGps.getLongitude();
```

Figura 46 - Captura de medições e Instancia do *DataModel*.

As coordenadas são separadas em dois vetores um para latitude e um para longitude e após isso é aplicado o algoritmo do critério de Chauvenet.

```
Chauvenet chLat = new Chauvenet();
Chauvenet chLng = new Chauvenet();

chLat.run(lat);
chLng.run(lng);
```

Figura 47 - Execução do algoritmo de fusão.

Após a fusão dos dados, estes deve ser adicionados na lista da classe *Middleware* para

que a aplicação possa fazer sua utilização.

```
mid.getDataModel().add(dadosNew);
```

Figura 48 - Inserindo novos dados na lista do *Middleware*.

3.4.2.4 - OBTENÇÃO DOS DADOS E FUSÃO DE SENSORES

Para obtenção dos dados dos sensores, a classe *Middleware* oferece dois meios de obtenção das informações dos sensores. A primeira é obter uma lista de todas as medições do sensor e a segunda é obter a última medição realizada.

Para obter uma lista de medições basta executar o método *getMeasurementsList(<nome do controlador>)* e, para obter a última medição realizada, basta executar o método *getDeviceLastMeasurement(<nome do controlador>)* (ver Figuras 49 e 50).

```
List gpsYellowList = mid.getMeasurementsList("GpsYellow");
```

Figura 49 - Obtendo lista das medições do controlador *GpsYellow*.

```
GPGGAModel gpsYellow = (GPGGAModel) mid.getDeviceLastMeasurement("GpsYellow");
```

Figura 50 - Obtendo última medição do controlador *GpsYellow*.

Como a classe *Middleware* gerencia vários dispositivos, o método que retorna a última medição retorna um *Object*, sendo assim é necessário fazer um *casting* com o tipo de modelo utilizado pelo sensor.

A classe *Middleware* também possui o método *getDataModel()* onde é retornado uma lista onde são armazenados os dados após a fusão conforme Figura 51. Se este comando for executado antes de aplicar o algoritmo de fusão, será retornado uma lista vazia.

```
List dataModel = mid.getDataModel();
```

Figura 51 - Obtendo lista dos valores após fusão.

Para a execução da fusão dos sensores que estão no *Middleware* deve-se executar o método *deviceFusion()* conforme Figura 52:

```
mid.deviceFusion();
```

Figura 52 – Executando fusão de sensores.

3.5 - CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados as etapas de desenvolvimento do projeto proposto, deste o processo de concepção da arquitetura até o modo de funcionamento e implementação de novos sensores e algoritmos de fusão.

Na descrição da arquitetura proposta, foi feita uma abordagem sobre os principais componentes do *middleware* bem como a função de cada componente da arquitetura.

O componente Sensor*Reader* (elemento leitor de dados) é responsável por fazer a interação direta com a fonte de dados, obtendo as informações da fonte.

O controlador é responsável por fazer o gerenciamento do dispositibo, efetuando as operações de leitura, decodificação e estruturação dos dados no modelo do sensor.

A camada de fusão é responsável por extrair os dados (desejáveis para aplicar a fusão) dos controladores, a fim de utilizá-los como dados de entrada no algoritmo de fusão. Após aplicação da fusão este dado será estruturado novamente na classe inerente ao resultado da fusão.

Na abordagem do funcionamento, foi descrito em etapas os processos de configuração para utilização do *middleware*. Foi descrito também os processos para criação de novos componentes como classe leitora de dados, controladores e algorítmos de fusão.

No capítulo 4, será abordado os testes efetuados utilizando o *middleware* proposto para a integração de três GPS e aplicando em seus dados (latitude e longitude) o critério de Chauvenet.

CAPITULO 4 - ESTUDO DE CASO

Como estudo de caso, uma aplicação foi implementada a fim de efetuar testes e validações no projeto em discussão. Esta aplicação consiste em simular a captura das coordenadas de três GPS's (marcadores GPS 1, GPS 2 e GPS 3) e realizar a fusão destas coordenadas utilizando o *middleware*.

Para simular os dispositivos, foi desenvolvida uma aplicação *web* utilizando a API do GoogleMaps para capturar as coordenadas e também, mostrar o resultado da fusão. As coordenadas capturadas foram salvas em um arquivo de texto.

Foram utilizados dois algoritmos de fusão. O primeiro algoritmo consiste em aplicar média simples nos valores das coordenadas (latitude e longitude) e o segundo algoritmo de fusão utilizado foi o critério de Chauvenet, no qual efetua a média entre um conjunto de valores e também elimina valores discrepantes deste conjunto. Para os testes, foi considerado que os dispositivos estivessem sincronizados. Como foram utilizados medições de três GPS's, o desvio máximo aceitável é 1,38 conforme tabela 3, portanto, quando os valores da razão entre o desvio médio (de cada ponto) e o desvio padrão forem maiores que 1,38, esse valor será removido do conjunto.

4.1 – APLICAÇÃO DA MÉDIA NAS COORDENADAS

Neste estudo de caso, foi utilizado o cálculo da média das coordenadas proveniente dos GPS's utilizando o método de fusão *GPS_AVG*, portanto o seguinte processo foi efetuado:

- Inclusão das latitudes dos três pontos em um vetor;
- Inclusão das longitudes dos três pontos em um vetor;
- Obtenção da média das latitudes;
- Obtenção da média das longitudes;
- Estruturação dos valores resultantes no modelo de fusão.

A Tabela 7 demonstra os valores das latitudes dos três dispositivos em três medições realizadas. A Tabela 8 demonstra os valores das longitudes dos três dispositivos em três medições realizadas. A Tabela 9 demonstra os valores resultantes do processo de fusão, que no caso é a estruturação das médias de latitude e longitude em suas respectivas medições.

Tabela 7 – Valores das latitudes e média dos dispositivos em três medições.

N° da Medição	Latitude	Nova Latitude Baseada na Média
1	-22.208766682860954	-22.208767510602060
	-22.208768545278435	
	-22.208767303666793	
2	-22.208759233190747	-22.208768338343019
	-22.208772270113347	
	-22.208773511724964	
3	-22.20877289091915	-22.208774753336552
	-22.20877537414236	
	-22.208775994948148	

Tabela 8 – Valores das longitudes e média dos dispositivos em três medições.

N° da Medição	Longitude	Nova Longitude Baseada na Média
1	-49.95370142161846	-49.95370030403137
	-49.95370075106621	
	-49.95369873940945	
2	-49.953680634498596	-49.953690245747565
	-49.95369605720043	
	-49.95369404554367	
3	-49.95369538664818	-49.953692480921745
	-49.9536906927824	
	-49.953691363334656	

Tabela 9 - Coordenadas resultantes após processo de fusão

Nº da Medição	Coordenada (Latitude;Longitude)
1	-22.208767510602060 ; -49.95370030403137
2	-22.208768338343019 ; -49.953690245747565
3	-22.208774753336552 ; -49.953692480921745

O seguinte processo foi efetuado nos valores das coordenadas dos três GPS's utilizando o método de fusão GPSCoordsAverage que utiliza o critério de Chauvenet como algoritmo de fusão:

- Inclusão das latitudes dos três pontos em um vetor
- Inclusão das longitudes dos três pontos em um vetor
- Aplicacao do algoritmo na latitude podendo resultar na eliminação de valores discrepantes da latitude
- Aplicacao do algoritmo na longitude podendo resultar na eliminação de valores discrepantes da longitude
- Caso um valor da latitude for eliminado, seu respectivo valor da longitude será também eliminado, resultando na exclusão do ponto da amostra. Isso acontece, pois um ponto resultante de uma medição errada pode estar na mesma latitude dos demais e apresentar longitude diferente ou pode estar na mesma longitude e possuir valor de latitude diferente.

4.2 - TESTE REALIZADO EM UM TRAJETO

Neste teste realizado, foram demarcados pontos simulando um trajeto em uma rua. Para simular uma medição com erro, alguns pontos foram marcados fora da rua. As Figuras 53, 54 e 55 demonstram as marcações dos pontos enquanto a Figura 55 demonstra os dados após a fusão.

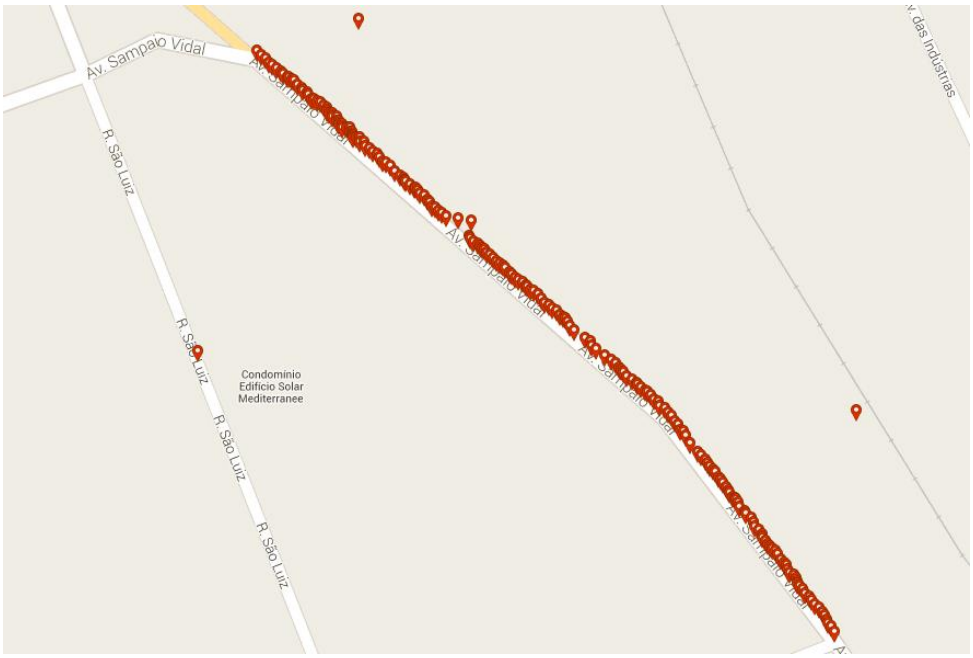

Figura 53 - Marcação do GPS 1.

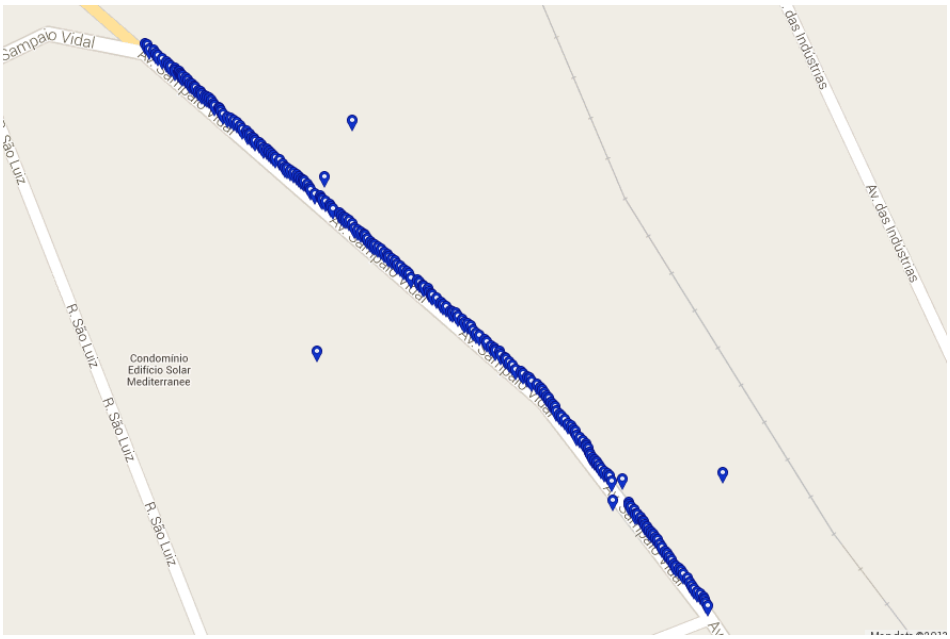

Figura 54 - Marcação do GPS 2.

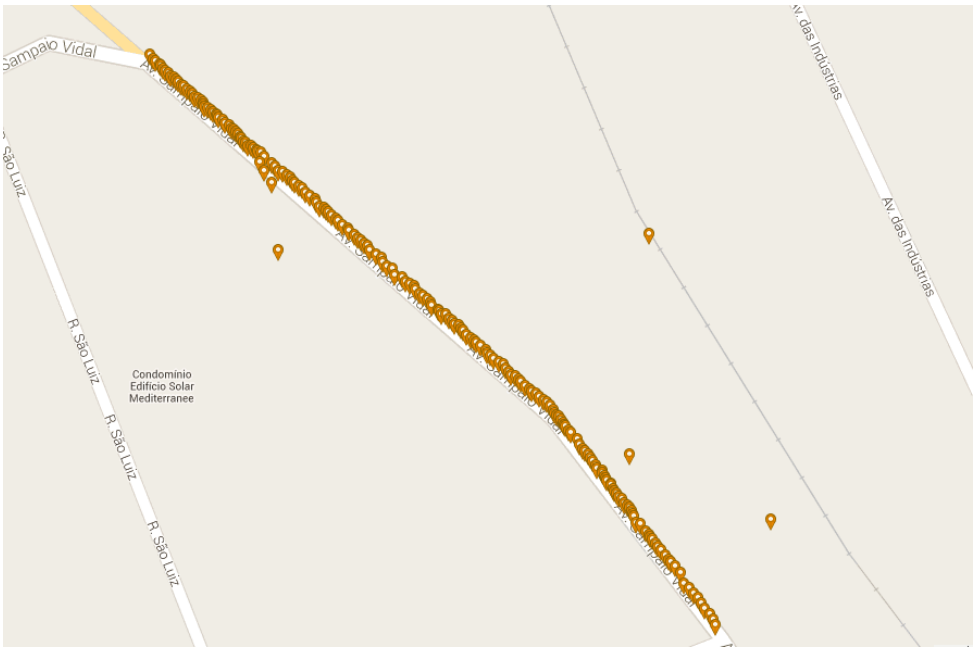

Figura 55 - Marcação do GPS 3.

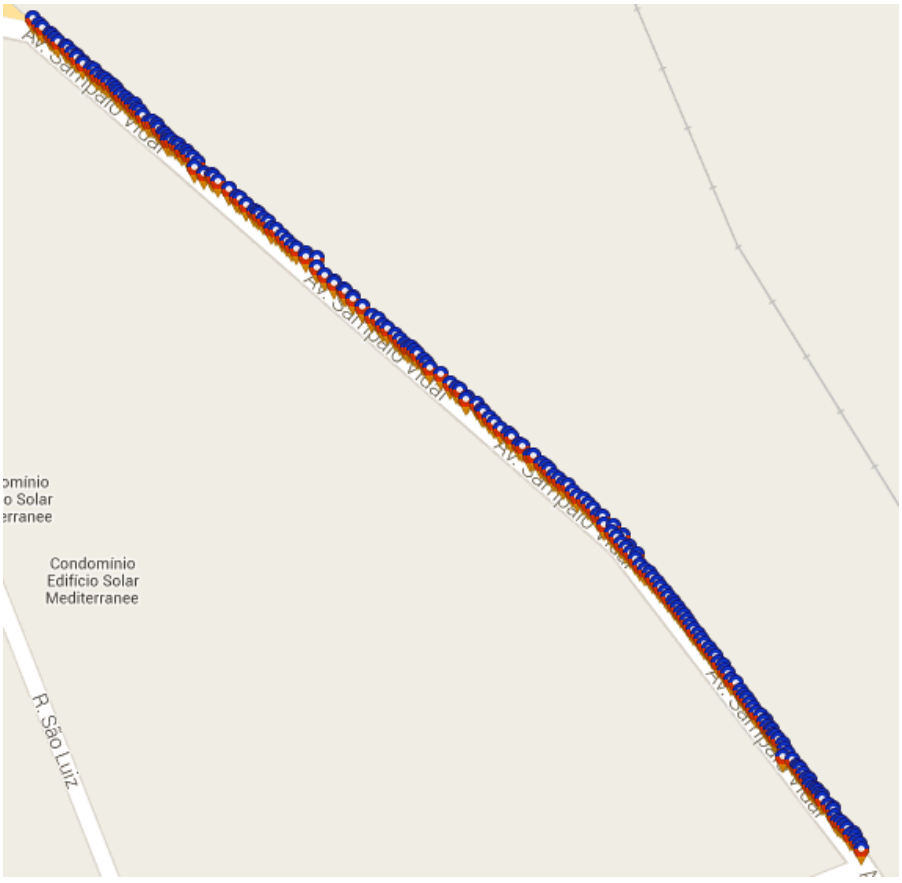

Figura 56 - Dados após a fusão.

Como observado na Figura 56, após a integração dos gps's foi realizada a fusão das coordenadas, onde é perceptível, como resultado da fusão, a eliminação dos pontos fora do trajeto. Outro ponto a ser observado é que além da eliminação dos pontos, foi que um pequeno trecho da rota marcada pelo GPS 1 não haviam medições e após a integração dos sensores, fez com que este trecho fosse coberto pelas marcações dos outros GPS's, caracterizando assim, tolerância a falhas.

4.3 - TESTES COM PONTOS UNITÁRIOS

Neste caso de teste, foram utilizadas pontos unitários de cada GPS e realizado a fusão dos dados. Os testes foram divididos em:

- Teste com pontos próximos;
- Teste com um ponto pouco distante;
- Teste com um ponto distante;
- Teste com dois pontos pouco distantes.
- Teste com dois pontos distantes.

Considera-se que o ponto correto, é o ponto representado pelo marcador de cor amarela.

4.3.1 - TESTE COM PONTOS PRÓXIMOS

O objetivo deste teste foi demonstrar de uma maneira sucinta a maneira que ocorre a integração e fusão dos pontos. As Figuras 57, 58 e 59 representam as marcações sincronizadas dos gps's. logo, quando é aplicada a fusão nestes dados, é gerado um novo ponto baseado na média da latitude e longitude.

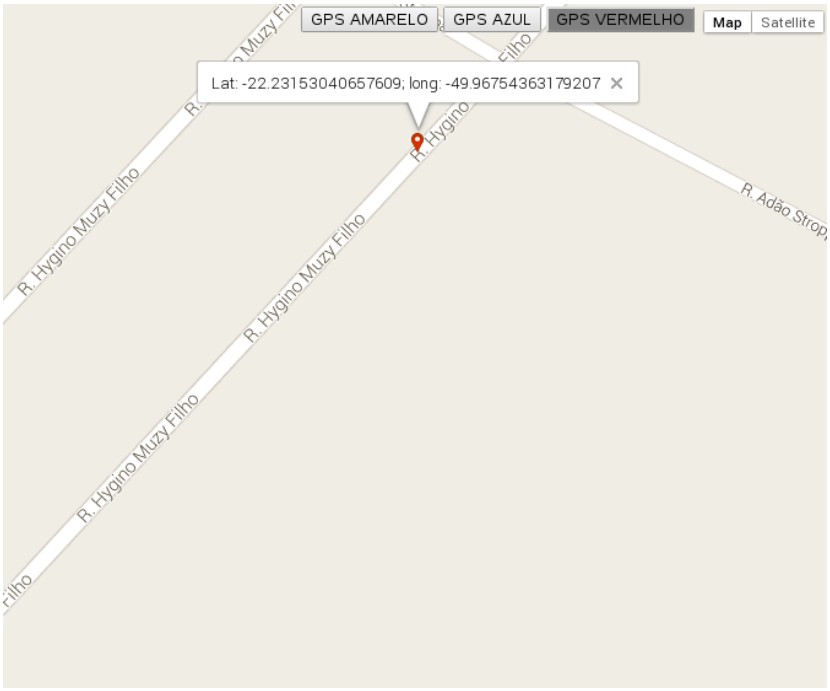

Figura 57 - Marcação do ponto do GPS 1.

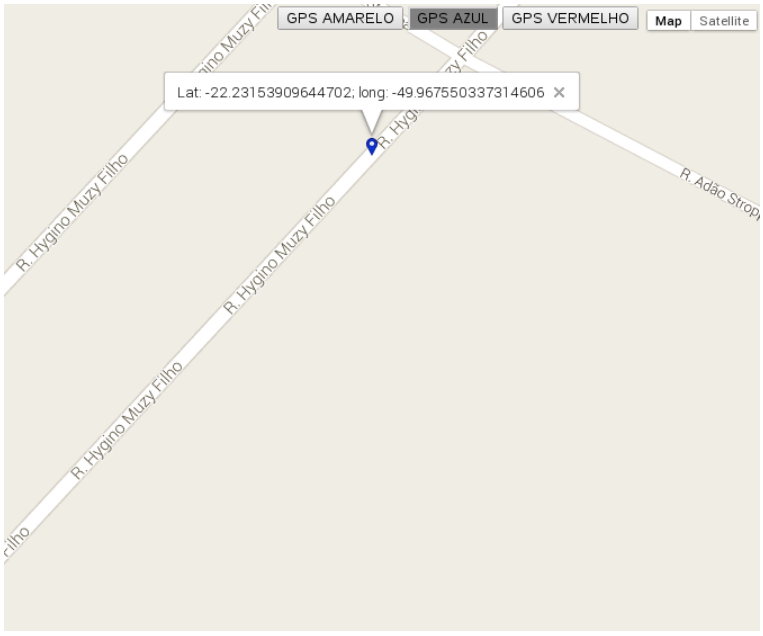

Figura 58 - Marcação do GPS 2.

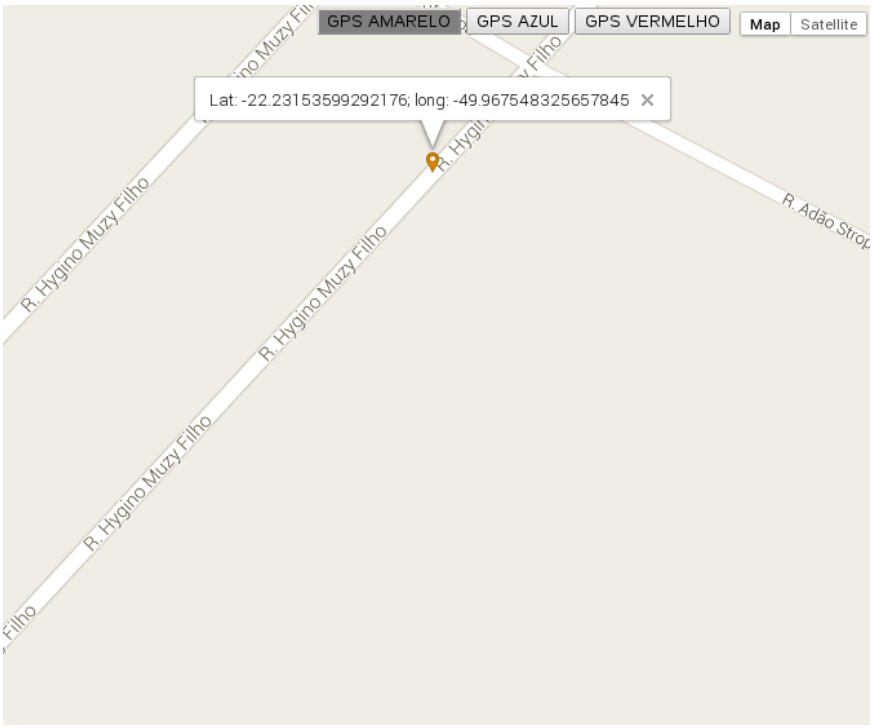

Figura 59 - Marcação do GPS 3

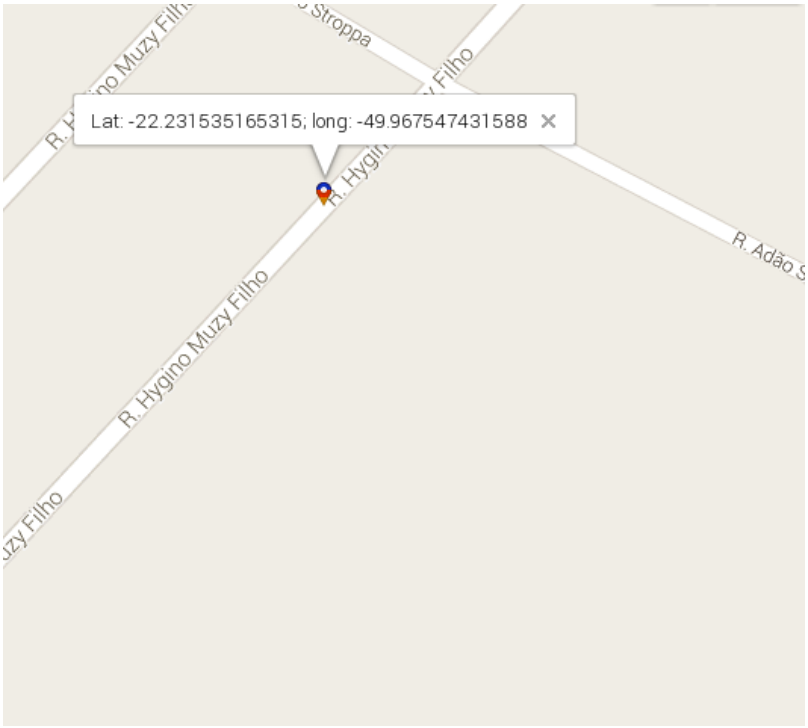

Figura 60 - Ponto após fusão.

4.3.2 - TESTE COM UM PONTO POUCO DISTANTE

Este teste demonstra a eliminação de pontos discrepantes. No caso exemplificado, o ponto apresenta uma distância relativamente pequena dos demais, representando uma medição errada de um GPS ocasionada, por exemplo, por um ruído ou interferência. O ponto discrepante ilustrado pela Figura 62 foi eliminado após a aplicação da fusão e uma nova média foi calculada com os valores das coordenadas dos pontos remanescentes (figura 61 e Figura 63). O novo ponto é representado pela Figura 64.

Figura 61 - Marcação do ponto do GPS 1.

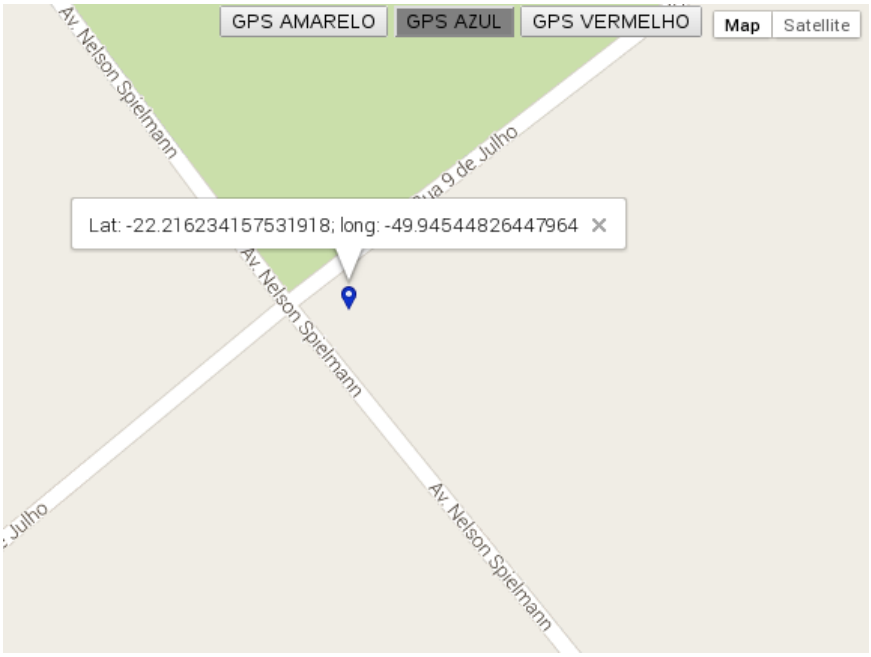

Figura 62- Marcação do GPS 2 (ponto distante).

Figura 63 - Marcação do ponto do GPS 3

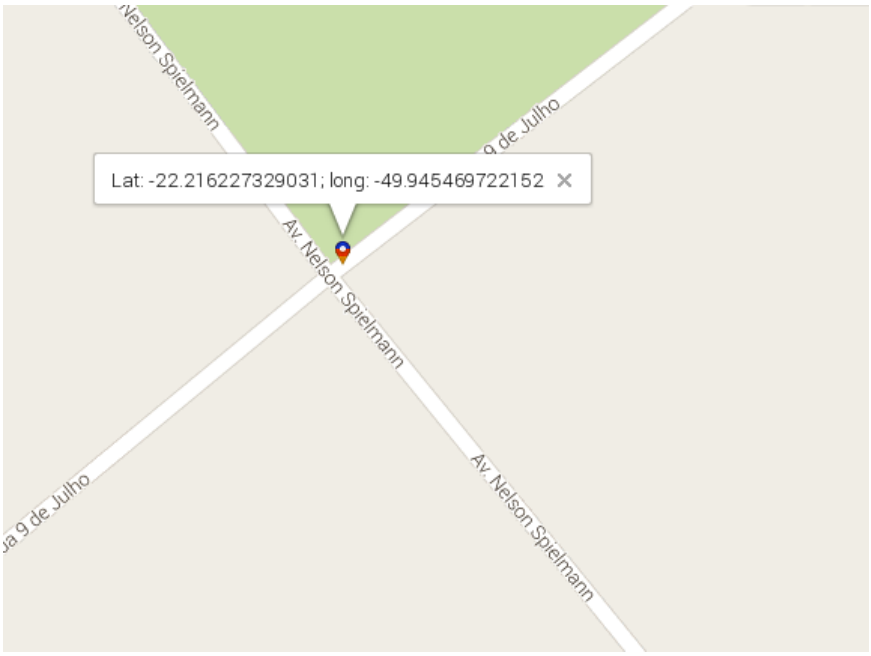

Figura 64 - Ponto após fusão dos dados.

4.3.3 - TESTE COM UM PONTO DISTANTE

Este teste também demonstra a eliminação de pontos discrepantes. Porém neste caso, o ponto apresenta uma distância maior dos demais, representando um erro maior na medição de um GPS ocasionada, por exemplo, por um ruído ou interferência. O ponto discrepante ilustrado pela Figura 65 foi eliminado após a aplicação da fusão e uma nova média foi calculada com os valores das coordenadas dos pontos remanescentes (Figura 66 e Figura 67). O novo ponto é representado pela Figura 68.

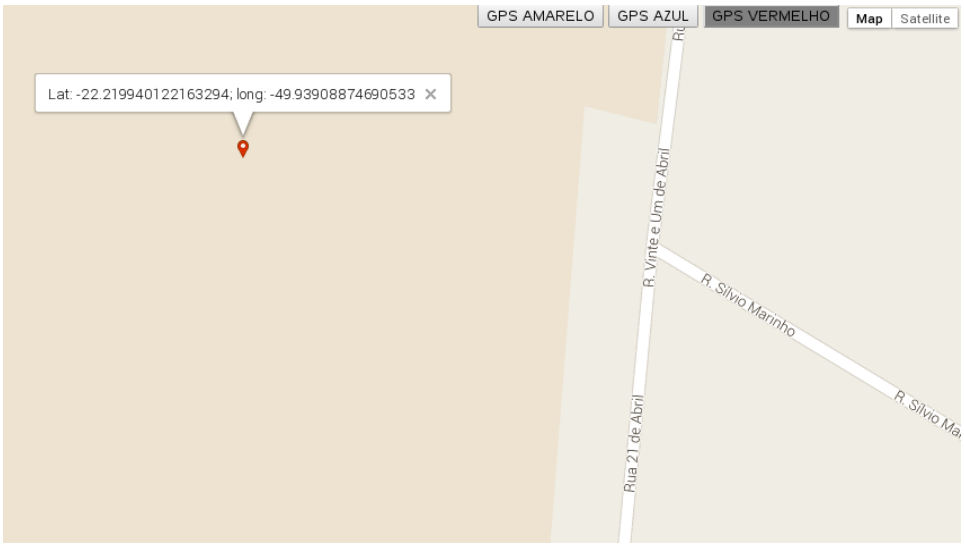

Figura 65 - Marcação do ponto do GPS 1.

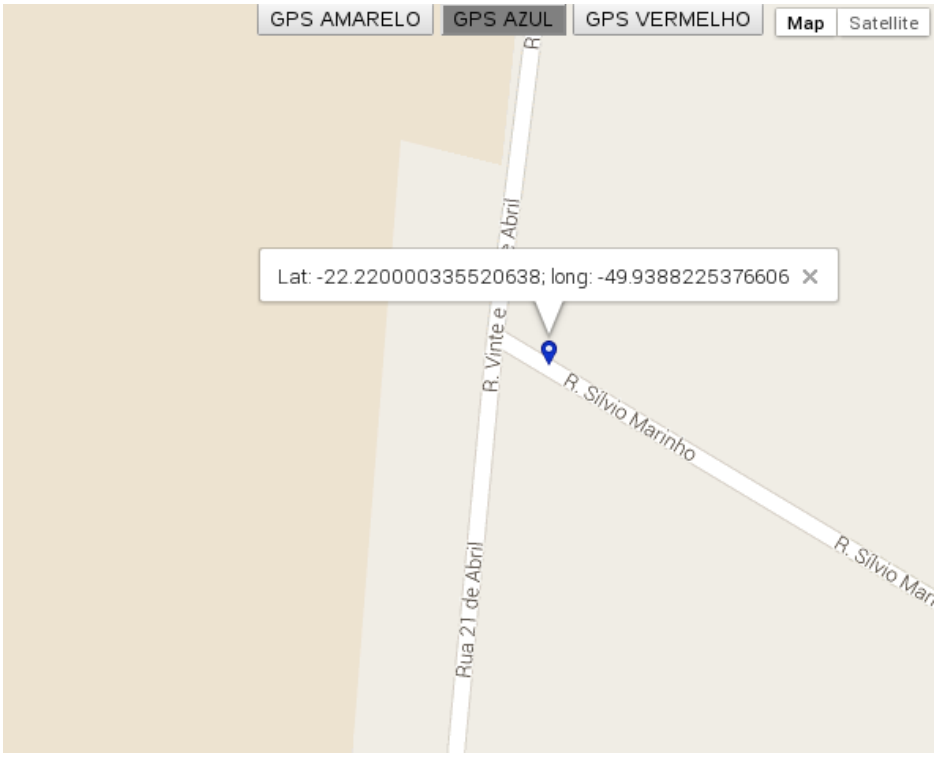

Figura 66 - Marcação do ponto do GPS 2.

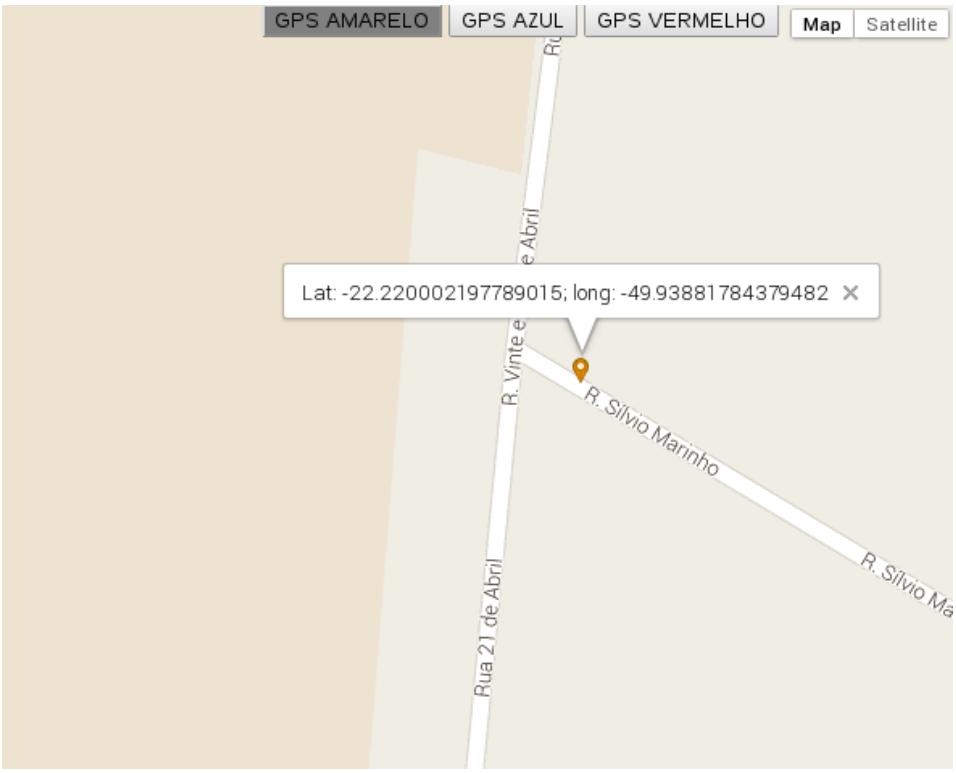

Figura 67 - Marcação do ponto do GPS 3.

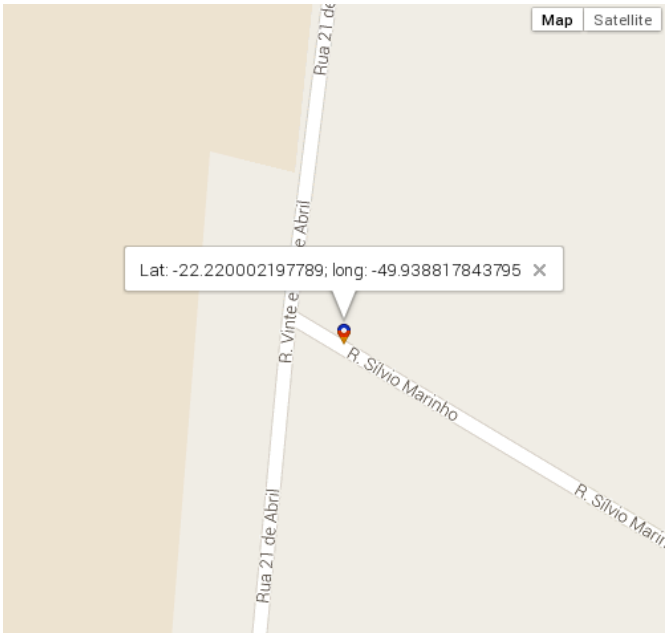

Figura 68 - Ponto após a fusão.

4.3.4 - TESTE COM DOIS PONTOS POUCO DISTANTES

Este teste tem como objetivo demonstrar a influência de um conjunto de medições

ruidosas na precisão do valor resultante. Portanto foram efetuadas duas medições ruidosas provenientes dos GPS's. Neste caso, a maioria dos valores são ruidosos tendo maior peso no valor da média entre as coordenadas. Assim sendo, a precisão da coordenada resultante da fusão são comprometidos, fazendo com que o ponto seja "atraído" para o local onde está a maioria dos pontos. As Figuras 69, 70, 71 e 72 ilustram este caso de teste.

Figura 69 - Marcação do ponto do GPS 1 (ponto distante).

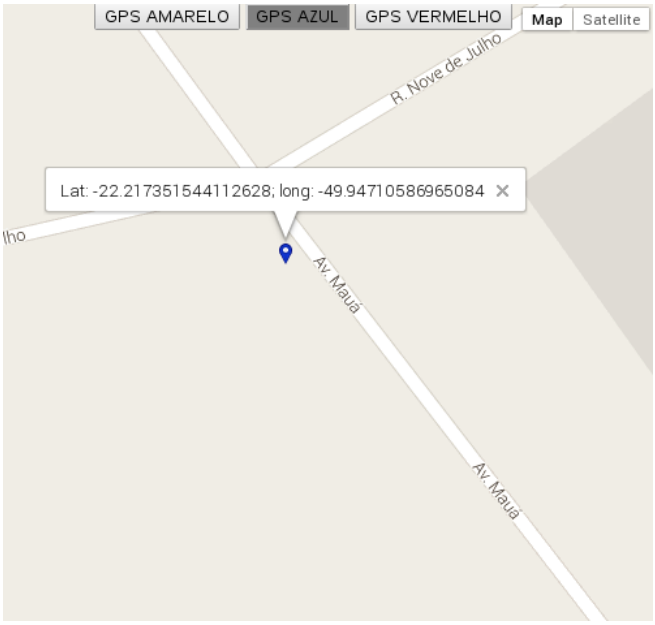

Figura 70 - Marcação do ponto do GPS 2 (ponto distante).

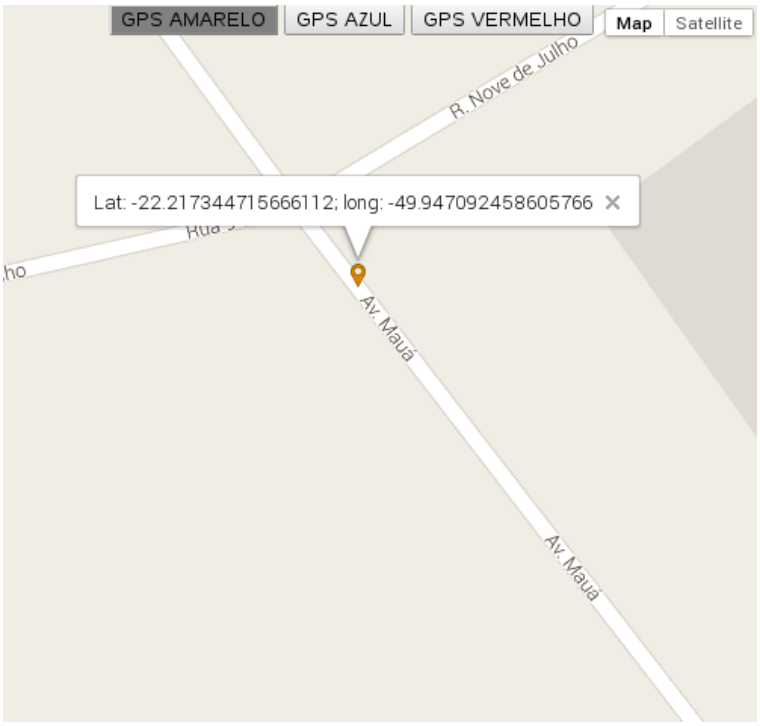

Figura 71 - Marcação do ponto do GPS 3.

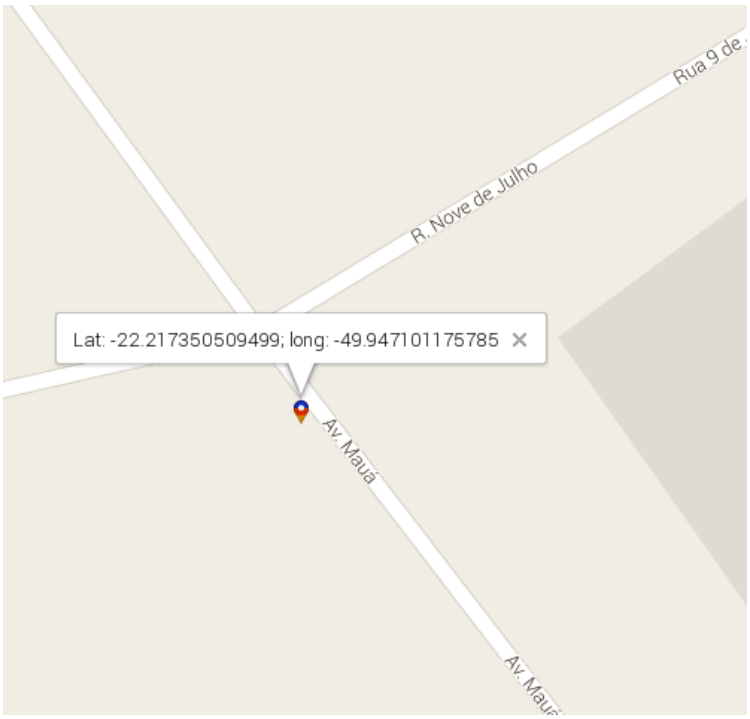

Figura 72 - Ponto após a fusão.

4.3.5 - TESTE COM DOIS PONTOS DISTANTES

Este teste também tem como objetivo demonstrar a influência de um conjunto de medições ruidosas na precisão do valor resultante. Neste caso os dois valores ruidosos são distantes entre si (Figuras 73 e 74), e o ponto de referência da medição correta é o ponto representado pela Figura 75. Assim sendo, a precisão dos valores da coordenada resultante da fusão são comprometidos, resultando em um ponto distante do correto. A diferença para o estudo de caso anterior, é que neste caso, a coordenada resultante não é "atraída" pela maioria dos pontos, já que não existe um ponto (ou pontos) que exerça um maior peso.

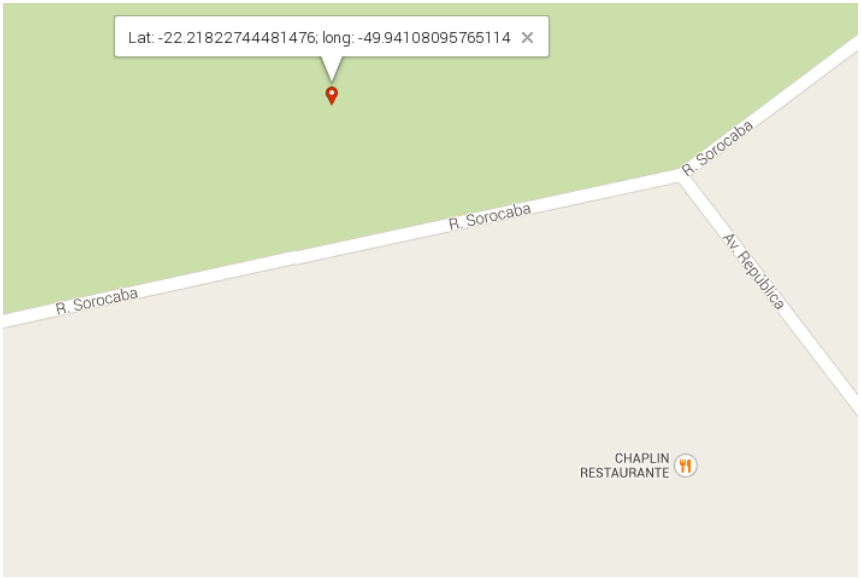

Figura 73 - Marcação do ponto do GPS 1 (ponto distante).

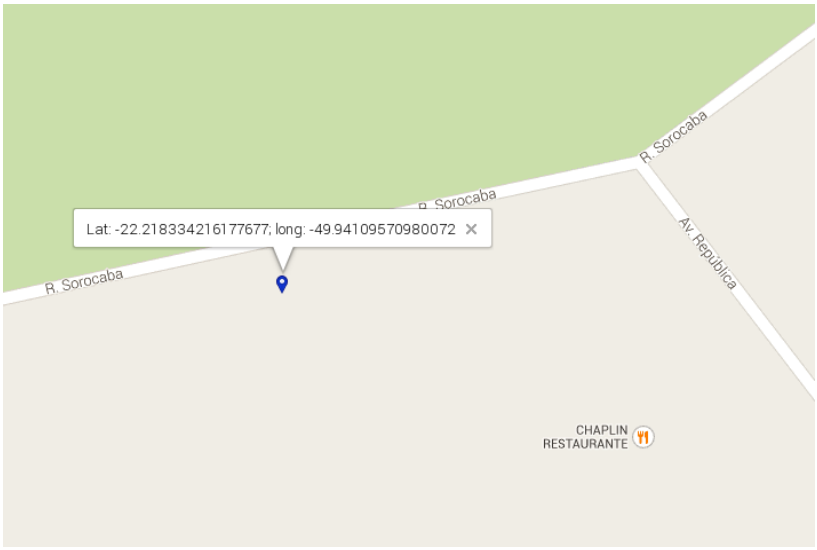

Figura 74 - Marcação do ponto do GPS 2 (ponto distante).

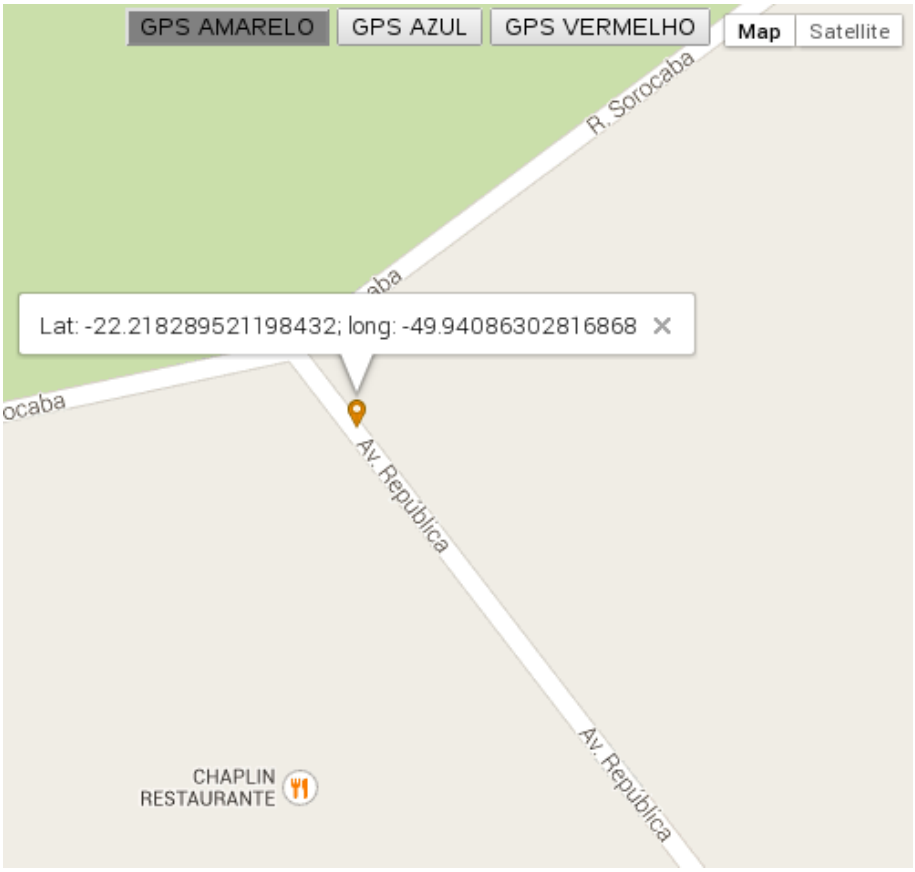

Figura 75 - Marcação do ponto do GPS 3.

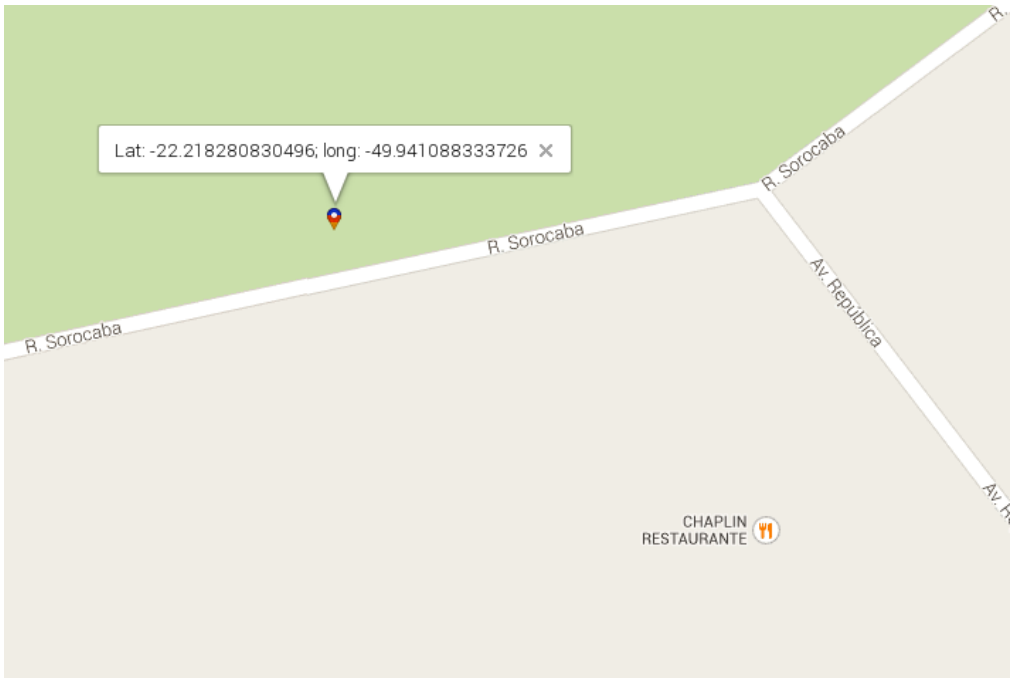

Figura 76 - Ponto após a fusão.

4.4 - AVALIAÇÃO DOS ESTUDOS DE CASO

Mediante os testes realizados, o *middleware* correspondeu bem na integração dos sensores porém, o critério de Chauvenet apresenta algumas limitações quanto ao seu uso como algoritmo de fusão visto que, ele avalia se um ponto é discrepante a partir da média dos valores.

O problema que isso pode acarretar é que se por exemplo, a maioria dos dados dos sensores estiverem com valores errados (podendo ser devido a uma interferência) e a restante minoria apresentar valores corretos, os valores corretos serão eliminados pelo algoritmo pois serão considerados discrepantes em relação à média dos valores. Esta foi a intenção na demonstração dos estudos de caso onde foi utilizado dois pontos distantes e dois pontos pouco distantes. Os dois pontos (a maioria) foram marcados de maneira a representar uma medição com erro de dois GPS's, sendo apenas um GPS representando o valor correto. O resultante da fusão foi que o ponto que estava correto, foi eliminado pelo algoritmo acarretando em um erro de precisão.

Estes problemas ocorrem pois o sistema não tem conhecimento de qual GPS está marcando o ponto correto, ou seja, não sabe diferenciar que o ponto distante não está correto. Isso leva a recorrer a técnicas que possam fazer esta estimação.

A teoria de Dempster Shafer permite combinação de evidências provindas de diversas fontes com o objetivo de alcançar um grau de credibilidade. A Regra de Bayes trabalha com probabilidades, ou seja, a probabilidade um ponto marcado pelo GPS esteja errada (ou correta). As cadeias de Markov são utilizadas para estimar estado futuro a partir do estado atual (estima estado n+1 a partir do estado n), deste modo, é possível realizar estimativas se os dados dos dispositivos representam uma medição errada ou correta.

Outra maneira de se conseguir melhora na precisão dos dados no processo de fusão seria aumentar o número de medições em função do tempo.

Porém quando a maioria dos dados está com valores próximos do correto (desejado), o algoritmo é capaz de efetuar a fusão destes valores, aumentando a precisão da medição. Este caso é ilustrado no estudo de caso com pontos próximos.

Se dentre os três valores, um valor estiver distante, este dado será eliminado da amostragem elevando a precisão do dado. Este caso é ilustrado no estudo de caso com um

ponto distante e um ponto pouco distante.

CAPÍTULO 5 – CONCLUSÕES

É notório que o desenvolvimento de aplicações que envolva sensores, é de certo modo complexa, visto que existem diversos fabricantes, diversos tipos de sensores e além disso, para interagir com estes dispositivos, geralmente é necessário a implementação de protocolos de comunicação e de rotinas inerentes às estas operações. Este problema é agravado no desenvolvimento de sistemas ciberfísicos, onde são utilizados diversos sensores e a operação destes dispositivos não é feita individualmente. Eles operam em conjunto a fim de fornecer informações mais precisas do ambiente podendo até utilizar técnicas de fusão de sensores. Portanto, o desenvolvimento de uma camada de abstração destas rotinas de acesso, torna o desenvolvimento de sistemas mais fácil, rápido e por consequência mais barato.

Deste modo, neste trabalho é descrito a implementação desta camada de abstração, ou seja, um *middleware* para integração de sensores e que oferece suporte a fusão de sensores. Além de abstrair operações de acesso aos dados dos dispositivos, ele permite a inserção de novos sensores e novos algoritmos de fusão de sensores.

Como limitações, este trabalho tem: a customização do *middleware* de forma a inserir novos sensores e algoritmos de fusão se dá alterando manualmente o código fonte do mesmo. Isso pode acarretar em um erro no funcionamento se não for implementado corretamente.

Como trabalhos futuros espera-se superar as limitações, efetuar estudos e avaliações de desempenho do *middleware* e aumentar a abrangência de dispositivos e algoritmos de fusão.

Por fim, resta avaliar a ferramenta e o quanto os benefícios esperados são efetivamente benéficos em aplicações reais.

REFERÊNCIAS BIBLIOGRÁFICAS

Broy, M. Cyber-Physical Systems: Technological and Scientific Chalenges. Universidade Técnica de Munique, Instituto de Ciência da Computação, 2010.

Cardenas, A. A. et al. Secure control: Towards survivable cyber-physical systems. In: Distributed Computing Systems Workshops, 2008. ICDCS '08. 28th International Conference, 2008. p. 495-500.

Elmenreich, W. Sensor *Fusion* in Time-Triggered Systems. Tese de Doutorado. Faculdade Tecnológica de Ciências Naturais e Informática, Viena, Áustria. 2002

Faceli, K., Combinação de Métodos de Inteligência Artificial para Fusão de Sensores. Dissertação de mestrado. Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, 2001.

Garay, J. R. B., Oliveira, A. M., Kofuji, Sérgio T. Proposed *Middleware* for Sensor Networks in Cyber-Physical System Environments. SENSORCOMM 2013 : The Seventh International Conference on Sensor Technologies and Applications, 2013.

Grewal, M. S., Andrews, A. P. Kalman Filtering: theory and pratice using MATLAB 3 ed, 2008

Hadim, S., Mohamed, N., "*Middleware* Challenges and Approaches for Wireless Sensor Networks",IEEE Distributed Systems Online, vol. 7, no. 3, 2006, art. no. 0603-o3001.

Hoang, D. D., Paik, H.Y., Kim, C. K. Service-Oriented *Middleware* Architectures for Cyber-Physical Systems. IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.1, 2012.

Huang, B. Cyber Physical Systems: A Survey. 2008

Khaleghi B., Khamis A. , Karraya F. O., Razavi, S. N. Multisensor data *Fusion*: A review of the state-of-the-art. Elsevier Journal,2011.

Lee, I., Cyber Physical Systems: Computing Revolution. CIS 480, Spring 2009

Liggins, M. E., Hall, D. L., Llinas, J. Handbook of Multisensor Data *Fusion*: Theory and Practice, Second Edition (Electrical Engineering & Applied Signal Processing Series). Springer, 2008.

Loureiro, A. A. F., Nogueira, J. M. S., Ruiz, L. B., de Freitas Mini, R. A., Nakamura, E. F., and Figueiredo, C. M. S.. Redes de sensores sem fio. In Simpósio Brasileiro de Redes de Computadores, pages 179 – 226, 2003.

Luo, R. C., KAY, Michael G.. Multisensor Integration and *Fusion* for Intelligent Machines and Systems. Norwood: Ablex Publishing Corporation, 1995. p. 688

Luo, R. C., KAY, Michael G. A Tutorial on Multisensor Integration and *Fusion*. In: IECON '90, 1990, Raleigh. 16th Annual Conference of IEEE. Pacific Grove: IEEE, 1990. p. 707 - 722.

Magalhães, N. A. F. Sistema de Apoio a Análise de Desempenho de Parques Eólicos. Dissertação de Mestrado. Faculdade de Engenharia da Universidade do Porto, 2011.

Nakamura, E. F. Fusão de Dados em Redes de Sensores sem Fio. Tese de Doutorado, Universidade Federal de Minas Gerais, 2007.

Penteado, C. G. Arquitetura modular de processador multicore, flexível, segura e tolerante a falhas, para sistemas ciberfísicos. Tese de doutorado. Universidade de São Paulo, Escola Politécnica, 2010.

Rajkumar, R., Lee, Insup., Sha, L., e Stankovic, J., Cyber-Physical Systems: The Next Computing Revolution. Design Automation Conference 2010, Anaheim, California, USA, 2010.

Salustiano, R. E. Aplicação de Técnicas de Fusão de Sensores no Monitoramento de Ambientes. Dissertação de Mestrado. Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, 2006.

Sanislav, T., Miclea, L. Cyber-Physical Systems - Concept, Challenges and Research Areas. CEAI, Vol.14, No.2, pp. 28-33, 2012

SÁ, M. P., *ConBus*: Uma Plataforma de *Middleware* de Integração de Sensores para o Desenvolvimento de Aplicações Móveis Sensíveis ao Contexto. Dissertação de Mestrado. Universidade Federal de Goiás, Instituto de Informática, 2010

Santana, D. D. S. Estimação de Trajetórias Terrestres Utilizando Unidade de Medição Inercial de Baixo Custo e Fusão Sensorial. Dissertação de Mestrado. Universidade de São Paulo, Escola Politécnica, 2005.

Welch, G., Bishop G. An Introduction to the Kalman Filter. Department of Computer Science University of North Carolina at Chapel Hill Chapel Hill, NC 27599-3175.

Wellington, S. Algorithms for Sensor Validation and Multisensor Fusion. Tese de Doutorado. Nottingham Trent University and Southamptom Institute, 2002

Simon, D. Kalman Filtering, Embedded Systems Programming, vol. 14, no. 6, pp. 72-79, Junho 2001