

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TÉRCIO PACITTI
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

HENRIQUE RIBEIRO DOS SANTOS SOARES

**CONDE: Um Sistema de Controle e Decisão para Edifícios Inteligentes
usando Redes de Sensores e Atuadores Sem Fio**

Rio de Janeiro

2012

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
INSTITUTO TERCIO PACITTI
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

HENRIQUE RIBEIRO DOS SANTOS SOARES

CONDE: Um Sistema de Controle e Decisão para Edifícios Inteligentes
usando Redes de Sensores e Atuadores Sem Fio

Dissertação de Mestrado submetida ao
Corpo Docente do Programa de Pós-
Graduação em Informática da
Universidade Federal do Rio de Janeiro
e à banca externa convidada como parte
dos requisitos necessários para obtenção
do título de Mestre em Informática.

Orientadores: Luci Pirmez
Flávia Coimbra Delicato

Rio de Janeiro
2012

S676 Soares, Henrique Ribeiro dos Santos

CONDE: um sistema de controle e decisão para edifícios inteligentes usando redes de sensores e atuadores sem fio / Henrique Ribeiro dos Santos Soares. – 2012.

000 f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti, 2012.

Orientadoras: Luci Pirmez ; Flávia Coimbra Delicato

1. Rede de Atuadores e Sensores Sem Fio. 2. Sistema de Controle e Decisão
3. Edifícios Inteligentes – Teses. I. Pirmez, Luci (Orient.). II. Coimbra, Flávia Delicato (Orient.). III. Universidade Federal do Rio de Janeiro, Instituto de Matemática, Instituto Tércio Pacitti. IV. Título.

CDD.

Henrique Ribeiro dos Santos Soares

CONDE: Um Sistema de Controle e Decisão para Edifícios Inteligentes usando
Redes de Sensores e Atuadores Sem Fio

Dissertação de Mestrado submetida ao Corpo Docente do Programa de Pós-Graduação em Informática da Universidade Federal do Rio de Janeiro e à banca externa convidada como parte dos requisitos necessários para obtenção do título de Mestre em Informática.

Aprovada em: Rio de Janeiro ____ de _____ de _____.

Prof^a. Luci Pirmez – Orientadora
D.Sc., COPPE/UFRJ, Brasil
UFRJ/PPGI

Prof^a. Flávia Coimbra Delicato – Orientadora
D.Sc., COPPE/UFRJ, Brasil
UFRJ/PPGI

Prof^a. Silvana Rossetto
D.Sc., PUC-RIO, Brasil
UFRJ/PPGI

Prof. Paulo de Figueiredo Pires
D.Sc., COPPE/UFRJ, Brasil
UFRJ/PPGI

Prof. Célio Vinícius Neves de Albuquerque
Ph.D., University of California, EUA
UFF/IC

Dedico este trabalho a todos que me amam e acompanharam o desenvolvimento deste trabalho, especialmente a minha querida Vitória, que partilhou comigo todas as etapas do trabalho, sendo essencial sobretudo nos momentos mais difíceis.

AGRADECIMENTOS

À minha querida Vitória, pela paciência, suporte, palavras de incentivo e de conforto e por ter caminhado comigo até aqui, sobrepondo tantas dificuldades. Esta conquista também é dela.

À meus pais, Deilson e Marilei, e minha irmã, Luize, e o restante da minha família, pela origem, força e amor. Sem eles, eu não existiria e, certamente, não chegaria até aqui.

Aos amigos de laboratório Érico, Hélio, Humberto, Igor, Joffre, Maicon, Rafael, Renato Pinheiro, Renato Azevedo, Sandro, Sérgio e Tiago, que durante todo o período do curso me ajudaram e doaram um pouco do seu curto tempo livre para que este trabalho fosse entregue. Muito obrigado!

Ao grande amigo Claudio, meu orientador de consideração, por todo o incentivo, interesse e disposição em ajudar não só a mim e como a todos no laboratório. Agradeço muitíssimo por tudo o que fez pelo andamento do trabalho, pelas revisões relâmpago e pela ajuda, principalmente nas publicações. Seu suporte foi fundamental para o sucesso desta empreitada.

Aos professores do PPGI pelas orientações e ensinamentos e, em especial, aos professores que participaram dos seminários de acompanhamento pelas correções e revisões dos trabalhos. A ajuda dos senhores foi fundamental para o meu sucesso.

Em especial, às minhas orientadoras, professora Luci Pirmez e professora Flávia Coimbra Delicato, por todos os momentos em que Sras. se dispuseram a me ajudar, me ensinando e me mostrando o caminho a seguir durante o curso. Professoras, o sucesso deste trabalho é um reflexo de toda a sua dedicação. Sinto-me contente de ter sido orientado pelas senhoras. Muito obrigado!

“Run, rabbit run.

Dig that hole, forget the sun,

And when at last the work is done

Don't sit down, it's time to dig another one.”

(Pink Floyd – Breathe – Roger Waters, David Gilmour, Richard Wright)

RESUMO

SOARES, Henrique Ribeiro dos Santos. **CONDE**: um sistema de controle e decisão para edifícios inteligentes usando redes de sensores e atuadores sem fio. 2012. 000 f. Dissertação (Mestrado em Informática) - Programa de Pós-Graduação em Informática, Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

A degradação ambiental e o aquecimento global são desafios alvos de intensa pesquisa e investimentos na busca de estratégias para minimizar a emissão de gases poluentes e para tornar mais eficiente o consumo de energia. Um dos campos de pesquisa no uso das TICs como provedoras de soluções para os desafios ambientais consiste no edifício inteligente (Smart Building), uma vez que os edifícios comerciais são grandes consumidores de energia e responsáveis por uma parcela significativa das emissões de gases poluentes. Este trabalho apresenta o CONDE, um sistema descentralizado de controle e decisão para aplicações de edifícios inteligentes que fazem uso de RASSF. Esta descentralização é proposta como uma forma de melhorar a eficiência energética dos edifícios inteligentes e sua tolerância a falhas. Além disto, o CONDE permite integrar diferentes aplicações de edifício inteligente através do compartilhamento de dados de monitoramento entre aplicações e do encadeamento de tomadas de decisões em aplicações distintas, buscando melhorar a eficiência energética pela minimização da realização de operações redundantes. Os experimentos mostraram que tanto a descentralização do sistema e decisão quanto a integração de aplicações contribuíram para melhorar a eficiência energética do edifício como um todo, e que a descentralização também contribuiu positivamente para a tolerância a falhas do sistema.

Palavras-chave: Rede de atuadores e sensores sem fio. Edifícios inteligentes. Eficiência Energética.

ABSTRACT

SOARES, Henrique Ribeiro dos Santos. **CONDE**: um sistema de controle e decisão para edifícios inteligentes usando redes de sensores e atuadores sem fio. 2012. 000 f. Dissertação (Mestrado em Informática) - Programa de Pós-Graduação em Informática, Instituto de Matemática, Instituto Tércio Pacitti, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2012.

Environmental degradation and global warming are challenges that require intense research and investments in search of strategies to minimize greenhouse gas emissions and to develop energy-efficient environments. A research field that makes use of Information and Communication Technologies (ICTs) as providers of solutions to environmental challenges is smart buildings, since commercial buildings are major energy consumers and are responsible for a significant portion of greenhouse gas emissions. This paper presents CONDE, a decentralized control and decision-making system for smart building applications using Wireless Sensor/Actuator Networks (WSAN). Such decentralization is proposed as a way to improve energy efficiency of smart buildings and their fault tolerance. Moreover, the CONDE is able to integrate different smart buildings applications through the sensed data sharing and decision chaining between applications, as a way to improve energy efficiency by minimizing redundant operations. The experiments showed that both system decentralization and application integration contributed in improving energy efficiency of the building and that system decentralization contributed positively to enhance the system's fault tolerance.

Keywords: Wireless actuator and sensor networks. Smart Buildings. Energy Efficiency.

LISTA DE FIGURAS

Figura 1. Exemplo de mote e de placa de sensoriamento.....	24
Figura 2. Diagrama de componentes UML da arquitetura lógica do sistema proposto.	36
Figura 3. Diagrama de componentes da arquitetura física do sistema proposta.....	44
Figura 4. Diagrama de sequência da fase de operação do CONDE	47
Figura 6. Representação gráfica do primeiro cenário utilizado nos experimentos.....	54
Figura 7. Representação gráfica do segundo cenário utilizado nos experimentos	54
Figura 8. Tempo requerido para enviar os dados necessários ao processo de inferência.....	57
Figura 9. Estatísticas de uso da memória RAM pelos nós da RASSF	58
Figura 10. Estatísticas de uso da memória flash de programa pelos nós da RASSF.....	59
Figura 11. Estatísticas de uso da memória flash de dados pelos nós da RASSF.....	59
Figura 12. Comparação em termos de atraso	61
Figura 13. Comparação em termos de energia gasta pela RASSF	61
Figura 14. Comparação em termos de eficiência do sistema	62
Figura 15. Comparação em termos de eficiência do sistema entre as soluções	66
Figura 16. Comparações em termos de economia de energia entre as aplicações	67

LISTA DE TABELAS

Tabela 1. Alguns tipos de sensores utilizados pelas aplicações descritas	27
Tabela 2. Estatísticas de uso da CPU pelos nós da RASSF	58
Tabela 3. Estatísticas do consumo de energia pelos nós da RASSF	60
Tabela 4. Efeito de falhas sobre os NATs	63
Tabela 5. Efeito de falhas sobre o NEB	63
Tabela 6. Efeito de falhas sobre os NSDs	64
Tabela 7. Efeito de falhas sobre os NSOs	65
Tabela 8. Comparação entre os experimentos simulado e real.....	67

LISTA DE EQUAÇÕES

Equação 1. Equação de Aproximação da Energia Consumida no Recebimento	55
Equação 2. Equação de Aproximação da Energia Consumida no Transmissão	56

LISTA DE ABREVIATURAS E SIGLAS

A – Ampere

API – Application Programming Interface

AVAC – Aquecimento, Ventilação e Ar-Condicionado

C – Celsius

cd – Candela

CDC – Cell Device Concentrator

CONDE – Sistema de CONtrole e DEcisão

CTP – Collection Tree Protocol

DIP – Dissimination Protocol

EB – Estação Base

h – Hora

HTTP – Hypertext Transfer Protocol

ICEM – Integrated Concurrency and Energy Management

IP – Internet Protocol

IPv4 – Internet Protocol Version 4

IPv6 – Internet Protocol Version 6

J – Joule

KP – Knowledge Processors

KWh – Kilowatt-hora

LED – Light-Emitting Diode

m – Metro

MANET – Mobile Ad-Hoc Network

MCC – Macro-Cell Concentrator

NAT – Nó Atuador

NEB – Nó Estação Base

NSD – Nó Sensor Decisor

NSO – Nó Sensor Ordinário

RAM – Random Access Memory

RASSF – Rede de Atuadores e Sensores Sem Fio

RSSF – Rede de Sensores Sem Fio

s – Segundo

SIB – Semantic Information Broker

SSN – Shared Sensor Network

TCP – Transmission Control Protocol

TIC – Tecnologia da Informação e da Comunicação

V – Volt

W – Watt

WSAN – Wireless Sensor and Actuator Network

WSN – Wireless Sensor Network

Sumário

1	Introdução.....	17
2	Conceitos Básicos.....	23
2.1	Redes de Atuadores e Sensores Sem Fio.....	23
2.2	Edifícios Inteligentes (<i>Smart Buildings</i>).....	25
2.2.1	Aplicações para Edifícios Inteligentes.....	26
2.2.2	Tipos de Sensores Utilizados em Aplicações para Edifícios Inteligentes.....	28
2.2.3	Integração entre Aplicações para Edifícios Inteligentes.....	29
3	Trabalhos Relacionados.....	31
4	CONDE: Sistema de Controle e Decisão para Edifícios Inteligentes.....	35
4.1	Arquitetura Lógica do CONDE.....	35
4.1.1	Descrição do Subsistema Inferência.....	40
4.1.2	Descrição do Subsistema de Integração.....	41
4.2	Arquitetura Física do CONDE.....	42
4.3	Funcionamento do Sistema de Decisão.....	45
4.3.1	Fase de Inicialização.....	45
4.3.2	Fase de Operação.....	46
5	Avaliação Experimental.....	49
5.1	Configuração do Ambiente.....	50
5.2	Descrição do Método de Inferência usado no protótipo.....	51
5.3	Descrição do Método de Integração usado no protótipo.....	52
5.4	Descrição do Cenário, Experimentos e Métricas.....	53
5.4.1	Experimentos de Calibração.....	56
5.4.2	Experimento para avaliar o consumo de recursos da RASSF.....	57
5.4.3	Experimento com uma plataforma de sensores reais.....	60
5.4.4	Experimento para avaliar o CONDE em termos de tolerância a falhas.....	62
5.4.5	Comparação entre o CONDE e outros sistemas.....	65
6	Conclusões.....	68
6.1	Trabalhos Futuros.....	70
	Referências.....	72
	ANEXO 1 – Base de dados Aplicações.....	77
	ANEXO 2 – Base de dados Parâmetros de Inferência.....	78
	ANEXO 3 – Base de dados Parâmetros de Integração.....	95

1 Introdução

A degradação ambiental e o aquecimento global estão entre os maiores desafios enfrentados atualmente pela população mundial. Tais desafios são alvos de intensa pesquisa e investimentos por organizações governamentais e privadas na busca de estratégias para reduzir a emissão de gases poluentes e para tornar mais eficiente o consumo de energia.

As Tecnologias da Informação e da Comunicação (TICs) desempenham um papel vital na solução dos problemas ambientais causados pela degradação da natureza. Apesar de serem parte do problema, uma vez que os equipamentos que as utilizam também consomem energia e são fontes de poluição, as TICs podem contribuir eficazmente para a redução do consumo de energia, tornando o funcionamento de equipamentos tradicionalmente utilizados em diversos campos mais eficiente e, conseqüentemente, diminuindo o desperdício tanto de energia quanto de matéria-prima (WEBER 2009). Desta forma, as TICs assumem um importante papel na busca por soluções que possibilitem o crescimento sustentável das nações.

Um dos campos de pesquisa no uso das TICs como provedoras de soluções para os desafios ambientais consiste no *edifício inteligente (Smart Building)*. Os edifícios inteligentes são definidos em (SCHOR et al. 2009) como edifícios equipados com dispositivos inteligentes instalados de forma a minimizar o consumo de energia sem comprometer o conforto e a segurança do usuário. O trabalho de (WONG et al. 2005) define requisitos a serem levados em conta na construção de edifícios inteligentes, a saber, em ordem de prioridade: (i) sustentabilidade, (ii) usabilidade e flexibilidade do espaço, (iii) minimização de custos de construção, operação e manutenção, (iv) conforto dos usuários, (v) eficiência de operação, (vi) controle em situações de desastre, (vii) treinamento de pessoal, (viii) implantação de alta tecnologia, (ix) processo de construção e estruturas e (x) saneamento.

Um passo importante rumo a um estilo de vida sustentável é melhorar a eficiência energética dos edifícios, residenciais e comerciais, pela redução do consumo de energia. Embora os avanços recentes no campo da ciência dos materiais tenham permitido obter reduções significativas quanto ao consumo de energia diretamente na estrutura dos edifícios, através, por exemplo, da utilização de materiais que absorvem menos calor no revestimento externo do edifício, ainda há problemas a serem superados. Uma das questões ainda em aberto relaciona-se à grande quantidade de energia que é desperdiçada pelo uso exagerado de

diversos equipamentos (GeSI 2008), tais como condicionadores de ar, aquecedores e até lâmpadas.

Aplicações para edifícios inteligentes mais comumente encontradas na literatura (WEBER 2009) incluem: controle de sistemas de aquecimento, ventilação e ar condicionado (AVAC), de iluminação, de sombreamento e de qualidade do ar e janelas (ventilação natural); monitoramento da atividade de dispositivos (visando desligá-los se não utilizados); controle de utilidades domésticas (por exemplo, controle de televisores, máquinas de lavar) e aplicações de segurança para proteger pessoas (controle de acesso) e equipamentos (proteção contra furto de equipamentos). Tais aplicações frequentemente monitoram grandezas físicas extraídas do ambiente monitorado, como luz, aceleração, temperatura, presença, químicos (como fumaça ou gases) e voltagem. Atualmente, a maioria dos trabalhos encontrados na literatura que propõem aplicações voltadas para edifícios inteligentes apontam soluções centralizadas de monitoramento e controle as quais fazem uso de sensores analógicos e cabeados (SHU et al. 2009).

Recentemente, foram apresentadas na literatura novas propostas de soluções de controle e monitoramento que fazem uso de Redes de Sensores Sem Fio (RSSF) voltadas para edifícios inteligentes, como (BYUN e PARK 2011), (CHRUNGOO et al. 2011), (GEZER e BURATTI 2011), (FORTINO e GUERRIERI 2011) e (SUN et al. 2011). As RSSFs, segundo (DELICATO 2005), são compostas por dispositivos pequenos, de baixo custo de produção, com fonte de energia e capacidades computacionais limitadas, dotados de componentes de sensoriamento e capazes de se comunicar através de enlaces sem fio. Elas são compostas por um grande número de nós sensores, frequentemente da ordem de centenas ou até milhares, que atuam de forma colaborativa com a finalidade de monitorar determinado ambiente. Na maior parte das propostas atuais de aplicações de edifício inteligente, a RSSF é utilizada apenas como infraestrutura de sensoriamento e de comunicação para transmitir os dados coletados pelos sensores para uma entidade central, uma estação base ou sorvedouro, onde se localiza todo o processo de controle e decisão das aplicações.

Segundo estimativas publicadas no relatório (GeSI 2008), as soluções para edifícios inteligentes utilizando RSSFs disponíveis até a publicação do mesmo podem chegar a reduzir o consumo de energia em até 13% e a emissão de gases pelos próprios edifícios em até 35%. Em síntese, as RSSFs possuem um papel de grande importância na resposta aos desafios ambientais, pelo seu potencial de contribuição para a redução de emissão de gases poluentes e pela redução do consumo de energia nos edifícios inteligentes.

Apesar da crescente aplicação das RSSFs em edifícios inteligentes, os sensores são dispositivos passivos, altamente limitados em relação à energia, processamento, memória e poder de comunicação, que basicamente coletam e transmitem dados, mas não atuam no ambiente. Por outro lado, as Redes de Atuadores e Sensores Sem Fio (RASSF) (AKYILDIZ e KASIMOGLU 2004) incluem nós atuadores, que são dispositivos com menos restrições que um nó sensor comum e podem realizar tarefas mais complexas, as quais exigem maior poder computacional e têm maior custo energético. Além disto, atuadores podem atuar ativamente no ambiente físico, em resposta a situações indicadas pelos dados coletados pelos nós sensores (LIU et al. 2010). Assim, além dos atuadores serem utilizadas como tecnologia para realizar o monitoramento de dados, podem ser utilizadas também como ferramentas poderosas em aplicações de controle. Ou seja, é possível que os nós da própria RASSF tomem decisões baseadas em dados coletados dentro da mesma e atuem rapidamente para que ações sejam executadas como, por exemplo, fazer ajustes via atuadores para reduzir a emissão de gases, para diminuir o consumo de energia elétrica ou melhorar o aproveitamento da mesma, ou ainda para promover maior conforto das pessoas presentes no edifício. O trabalho de (LEWIS 2005) define requisitos a serem levados em conta no desenvolvimento de aplicações para RASSFs, a saber: (i) suporte a um grande número de nós, (ii) baixo consumo energético, (iii) auto-organização da rede, (iv) processamento de dados de múltiplas fontes de sensoriamento e (v) habilidade de fazer consultas.

Em (DEGW et al. 1992), o conceito de edifícios inteligentes é criticado, pois, segundo os autores, os sistemas implantados nos edifícios não seriam capazes de lidar com as constantes mudanças nas organizações que os ocupam ou a tecnologia utilizada por eles se tornaria prematuramente obsoleta, provocando a necessidade constante de reformas ou até demolição. Neste contexto, as RASSF fornecem vantagens significativas sobre outras tecnologias de comunicação tradicionais, como facilidade de implantação, mobilidade, baixo custo e inteligência agregada pelo processamento nos diversos nós que compõe a RASSF (GUNGOR et al. 2010). Portanto, uma RASSF é capaz de realizar todo o processo de tomada de decisão das aplicações para edifícios inteligentes, e também é capaz de executar ações como: modificar o termostato de um ar condicionado de acordo com o número de pessoas em uma sala, desligar as luzes em uma sala vazia ou ligar ventiladores quando a qualidade do ar é considerada inadequada. Tradicionalmente, todas essas decisões acontecem fora da rede de sensores e de forma centralizada (SCHOR et al. 2009; LIU et al. 2010).

Ao se retirar os processos de controle e decisão das aplicações da entidade centralizadora e incorporá-los à RASSF (*in-network processing*), novos aspectos relacionados a esta tecnologia devem ser tratados. Um desafio chave é aumentar o tempo de vida da RASSF tanto quanto possível, mantendo a funcionalidade das aplicações e melhorando a eficiência energética dos edifícios inteligentes (SHU et al. 2009). Visto que a RASSF tem atribuições extras, além daquelas realizadas pelas RSSFs, é necessário adaptar técnicas de processamento de sinais existentes em ambientes inteligentes para reduzir ao máximo o número de transmissões de dados, com o intuito principal de economizar energia da RASSF. Outro desafio consiste em desenvolver metodologias que possibilitem integrar diferentes aplicações dentro da RASSF. A integração nesse contexto é definida como a capacidade de comunicação, troca de informação e colaboração entre aplicações para atingir objetivos comuns (NUNES 1995). Exemplos de vantagens de integrar aplicações são: (i) utilização mais eficiente de recursos, sejam eles energéticos, computacionais ou mesmo humanos, (ii) reações mais rápidas e coordenadas aos eventos físicos monitorados, (iii) capacidade de correlacionar informações de diferentes aplicações para otimizar processos de decisão, (iv) encadeamento de decisões entre aplicações integradas, isto é, uma decisão tomada em uma dada aplicação pode disparar outra decisão em uma outra aplicação distinta.

O principal intuito de integrar diferentes aplicações em edifícios inteligentes é economizar energia, através da redução de repetições improdutivas de tarefas e da colaboração entre aplicações para atingir objetivos comuns. Desta forma, minimiza-se o desperdício de energia com equipamentos evitando que os mesmos permaneçam realizando tarefas em momentos indesejados, como, por exemplo, uma aplicação de iluminação que ilumina uma sala vazia quando outra aplicação, de AVAC, por exemplo, já detectou que a sala está vazia e interrompeu sua operação (NUNES 1995). Além disto, acelera-se a operação de tarefas similares em aplicações distintas, como, ainda utilizando o exemplo da sala vazia, quando a aplicação de iluminação notifica a aplicação de refrigeração que a sala está vazia e não é mais necessário mantê-la refrigerada.

Com o objetivo de contribuir com a solução dos problemas relatados e alcançar os benefícios descritos anteriormente, este trabalho apresenta o CONDE, um sistema de **controle** e **decisão** descentralizado, para aplicações voltadas para edifícios inteligentes que utilizam RASSFs. O CONDE atende a todos os requisitos de desenvolvimento de aplicações para RASSFs e atua, sobretudo, sobre os 5 requisitos de maior prioridade dos edifícios inteligentes, a saber, (i) sustentabilidade, (ii) usabilidade e flexibilidade do espaço, (iii) minimização de

custos de construção, operação e manutenção, (iv) conforto dos usuários e (v) eficiência de operação do edifício. Os demais requisitos, por serem de natureza sobretudo estrutural, estão fora do escopo do CONDE enquanto solução para edifícios inteligente. O CONDE também não trata aplicações que tenham como requisito atraso limitado, entrega de pacotes em tempo real ou alta taxa de transferência. Isto se dá devido a limitações intrínsecas à utilização da RASSF. Esta limitação poderia ser resolvida pela utilização de protocolos específicos para RASSFs que atendam a estas necessidades, no entanto esta funcionalidade foi considerada fora de escopo.

O CONDE é classificado como um sistema descentralizado, uma vez que todo o processo de controle e decisão das aplicações para edifícios inteligentes é realizado dentro da RASSF, de forma colaborativa, por seus próprios nós. Esta descentralização é uma forma de melhorar a eficiência energética dos edifícios inteligentes. Tal melhora da eficiência energética é consequência, dentre outros fatores, da proximidade entre os locais onde o processo de tomada de decisão é realizado e onde as decisões tomadas são efetivamente executadas. Assim, como as grandezas monitoradas não são transmitidas para uma entidade centralizadora, há um ganho em tempo de resposta, de forma que a decisão é aplicada mais rapidamente, economizando tempo e energia da RASSF.

Além disto, o CONDE permite, dentro da RASSF, integrar diferentes aplicações de edifício inteligente através do compartilhamento de dados de monitoramento entre aplicações e do encadeamento de tomadas de decisões em aplicações distintas, aproveitando-se assim de um paradigma recente conhecido como Redes de Sensores Compartilhadas (*Shared Sensor Networks* – SSN) (EFSTRATIOU et al. 2010). No paradigma das SSNs, ao invés de assumir projetos de RASSF/RASSF específicos para uma dada aplicação alvo, a infraestrutura de sensoriamento é compartilhada por múltiplas aplicações, que podem, potencialmente, pertencer a diferentes usuários. Portanto, as SSNs podem ser vistas como infraestruturas para sistemas físicos integrados para diversas aplicações. Assim, as aplicações ativas na RASSF podem compartilhar informações coletadas do ambiente e uma decisão tomada em uma aplicação pode acionar eventos em outras aplicações.

O principal objetivo do trabalho é mostrar que o CONDE contribui para a melhoria da eficiência energética do edifício como um todo. Para demonstrar que o objetivo do trabalho foi atingido com sucesso, foram realizados diversos experimentos para comprovar que a descentralização contribui positivamente para o desempenho do sistema. Estes experimentos avaliaram os benefícios trazidos pelo sistema em termos de economia de energia, tolerância a

falhas do próprio sistema (diferentemente das soluções similares investigadas neste trabalho) e eficiência do sistema e o impacto do sistema sobre o edifício tanto de forma direta pela medição do consumo de recursos provocados pelo sistema, quanto de forma indireta pela tolerância a falhas que a descentralização do sistema proporciona e a economia total sobre a energia gasta pelo edifício.

Os diferenciais da solução proposta em relação às demais soluções existentes na literatura são: (i) a descentralização do processo de controle e decisão das aplicações entre os nós sensores e atuadores; (ii) a reutilização de dados de sensoriamento por parte das aplicações; e (iii) a possibilidade de encadear decisões tomadas em aplicações distintas dentro da própria RASSF, resultando assim em economia no consumo de energia da RASSF também na fase de sensoriamento.

O restante deste trabalho está dividido como segue: o Capítulo 2 apresenta os conceitos básicos necessários à compreensão deste trabalho. O Capítulo 3 apresenta os trabalhos relacionados. O Capítulo 4 apresenta o CONDE e detalha todos os componentes e sub-sistemas que o compõe. O Capítulo 5 apresenta os resultados obtidos nos experimentos conduzidos. Finalmente, o Capítulo 6 apresenta as conclusões e os trabalhos futuros.

2 Conceitos Básicos

Neste capítulo são descritos os conceitos básicos necessários para o entendimento do trabalho proposto e que fundamentam os temas abordados.

Assim, primeiramente são apresentados os conceitos de Redes de Atuadores e Sensores Sem Fio (RASSF), incluindo descrições sobre os dispositivos utilizados nas mesmas e a topologia de organização destes últimos em uma rede. Em seguida são apresentados os conceitos de Edifícios Inteligentes (*Smart Buildings*), incluindo descrições detalhadas sobre os tipos de aplicações presentes nos mesmos, sobre os tipos de sensores tradicionalmente utilizados por estas aplicações e, finalmente, sobre técnicas de integração destas aplicações que visam tornar mais eficiente o uso da energia elétrica pelo edifício inteligente como um todo.

2.1 Redes de Atuadores e Sensores Sem Fio

Segundo (DELICATO 2005), as redes de sensores sem fio (RSSF) são redes compostas por pequenos dispositivos baratos, de baixo poder de processamento e armazenamento, dotados de capacidades de sensoriamento e capazes de se comunicar entre si através de conexões ad-hoc estabelecidas por enlaces sem fio. O surgimento das RSSF se deve aos avanços das tecnologias de comunicação sem fio e dos micro-sistemas eletromecânicos.

As RSSF, em geral, são formadas por um grande número de nós sensores, frequentemente da ordem de milhares, e sua finalidade é monitorar determinado ambiente descentralizada e remotamente. Para este fim, estes nós sensores realizam leituras e as transmitem para um ponto de coleta dos dados, chamado estação base ou sorvedouro, de onde as mesmas são retiradas para posterior análise. As EBs estão, geralmente, conectadas a dispositivos com maiores recursos computacionais de processamento e armazenamento, dedicados a tratar os dados recebidos dos sensores (YICK et al. 2008).

As leituras transmitidas para a RSSF são obtidas de dispositivos de sensoriamento integrados ao nó sensor. Como exemplo destes dispositivos, pode-se citar termômetros, luminômetros, extensômetros, barômetros, acelerômetros, detectores de substâncias químicas, entre outros. Os dados analógicos obtidos do dispositivo de sensoriamento são convertidos em um conversor analógico/digital, também integrado ao nó sensor e, finalmente, transmitidos para a RSSF. A Figura 1(a) apresenta um exemplo de dispositivo que pode ser configurado para fazer parte de uma RSSF. Este dispositivo, chamado *mote*, é do modelo MICAz da

família MICA, plataforma desenvolvida pela empresa Crossbow. A Figura 1(b) apresenta um exemplo de placa de sensoriamento que pode ser acoplado ao mote MICAz para sensoriar temperatura e luminosidade.



Figura 1. Exemplo de mote e de placa de sensoriamento

Segundo (DELICATO 2005), os nós de uma RSSF são dispositivos especialmente desenvolvidos para operar com o menor gasto de energia possível. Esta política é necessária, pois a maioria dos modelos disponíveis atualmente é alimentada por baterias não-recarregáveis e é incomum que se elabore políticas de substituição destas baterias ou dos nós sem energia. Isto ocorre devido a grande quantidade de nós na RSSF e, em alguns casos, a dificuldade de acesso físico aos nós sensores. Assim, "o tempo de vida operacional das RSSF é severamente limitado pela capacidade de bateria de seus nós" (DELICATO 2005).

Em alguns projetos de RSSF, é desejável que esta seja capaz de agir ativamente sobre o ambiente de implantação, em vez de monitorá-lo passivamente. Neste caso, são incluídos à RSSF tradicional os dispositivos atuadores, fazendo aparecer o conceito de Redes de Atuadores e Sensores Sem Fio (RASSF) (AKYILDIZ e KASIMOGLU 2004). Atuadores são dispositivos muito similares aos *motes* utilizados na RSSF, entretanto eles são dotados de maior poder computacional e possuem menores restrições de energia. Além disto, os Atuadores contêm componentes interruptores integrados ao mote, como relés, transistores ou *dimmers*, por exemplo. Estes componentes interruptores, quando ligados a circuitos elétricos, podem ser usados para desencadear ações no ambiente onde a RASSF foi implantada, como apagar uma luz, ajustar o termostato de um ar-condicionado ou até enviar um sinal de desligamento a um computador.

As RASSF são redes de natureza dinâmica tanto do ponto de vista organizacional quanto do topológico. Isto significa que uma RASSF deve ser capaz de adaptar-se automaticamente a perda ou inserção de nós e se auto-gerenciar ("auto-configuração, auto-

manutenção, auto-organização, auto proteção etc" (RUIZ et al. 2004)), uma vez que, em geral, as RASSF operam sem intervenção humana. Entre os motivos que causam a perda de um nó de uma RASSF, podemos citar: o esgotamento da energia de sua bateria, sua desativação por estar em modo *sleep* ou uma condição de falha qualquer; e entre os motivos que causam a adição de um nó a uma RASSF, podemos citar: a instalação física de um novo nó ou sua reativação por sair do modo *sleep*.

As restrições de hardware e de energia fazem com que o projeto de uma RASSF seja muito influenciado por sua aplicação. Um projeto de uma RASSF, segundo (DELICATO 2005), visa atender apenas os requisitos mínimos da aplicação, de forma a consumir menos energia e, assim, maximizar seu tempo de vida. Tolerância a falhas, escalabilidade, custo de produção, ambiente operacional, topologia de rede, meio de transmissão e consumo de energia são alguns dos requisitos que podem influenciar um projeto de RASSF (RUIZ et al. 2004).

A questão do consumo energético é fundamental no projeto de um RASSF. Os motes tradicionais têm poucos recursos computacionais e energéticos, bem menos que os dispositivos Atuadores. Então, a gestão do consumo de energia, sobretudo dos primeiros, demanda o uso de técnicas particulares visando o menor consumo de energia. Todas as operações executadas, tanto pelos motes quanto pelos Atuadores, tem um custo de energia e algumas operações são mais custosas que outras. Por exemplo, sabe-se que a operação que mais onera a RASSF quanto ao consumo de energia é a transmissão de dados pelo enlace sem fio (YICK et al. 2008), portanto o uso desta operação deve ser minimizado ao máximo. O mesmo vale, em grau decrescente de custo energético para os *motes*, para o sensoriamento, o uso de LEDs, e até a execução de instruções pelo processador. Alguns autores afirmam que a energia consumida no envio de um bit de informação corresponde à energia consumida por até 1000 instruções de processamento (DIETRICH e DRESSLER 2009).

2.2 Edifícios Inteligentes (*Smart Buildings*)

Os edifícios inteligentes são, conforme descrito anteriormente, edifícios equipados com dispositivos que utilizam um conjunto de tecnologias que melhoram sua eficiência energética. Os edifícios inteligentes vêm sendo intensamente estudados por seu potencial de redução significativa do consumo de energia e das emissões de gases, sejam estes emitidos pelo próprio edifício ou no processo de geração de energia por fontes não sustentáveis, como usinas termo-elétricas, por exemplo. Segundo estimativas publicadas em (GeSI 2008), em 2020, os edifícios serão responsáveis por 22,5% das emissões em todo o mundo. Assim é de

extrema importância que se busque a eficiência energética para que as emissões de gases poluentes sejam minimizadas. Exemplos destas tecnologias incluem novos materiais utilizados na construção do edifício, como revestimentos externos que absorvem menos calor, por exemplo, ou a aplicação de TICs (Tecnologias da Informação e da Comunicação). As TICs desempenham um papel vital na solução dos problemas ambientais causados pela degradação da natureza. Isto é verdade no contexto dos edifícios inteligentes, pois as TICs têm potencial para tornar mais eficiente o uso da energia e minimizar as emissões de gases no próprio edifício.

As TICs podem ser utilizadas em diversas aplicações. Entre elas, podemos destacar: (i) sistemas de gestão de recursos do edifício, como sistemas para monitoramento de temperatura e iluminação, (ii) sistemas de automatização de procedimentos, como programas que desligam automaticamente equipamentos diversos e (iii) sistemas de segurança e controle de acesso, como mecanismos de detecção de presença que ativam câmeras ou catracas.

2.2.1 Aplicações para Edifícios Inteligentes

As aplicações para edifícios inteligentes, segundo (WEBER 2009), podem ser classificadas em três gerações distintas. Estas gerações diferem, principalmente, quanto à interação entre aplicações.

Na primeira geração, as aplicações para edifícios inteligentes são dotadas de diversos sistemas completamente auto-suficientes e independentes. Não há interação de nenhuma espécie entre estas aplicações, nem compartilhamento de infraestrutura, de forma que o controle de cada aplicação implantada no edifício é feito separadamente e cada aplicação deve ter sua própria rede dedicada para gerenciamento da aplicação. Estas aplicações foram intensamente utilizadas nas décadas de 1960 e 1970. Podemos citar como exemplos de aplicações de primeira geração: (i) aplicações de Aquecimento, Ventilação e Ar Condicionado (AVAC) e (ii) aplicações de segurança e monitoramento.

Já na segunda geração, várias aplicações passam a ser interconectadas através de uma rede dedicada que permite o gerenciamento centralizado e remoto das mesmas e algumas reações remotas como a solicitação de desligamento de equipamentos quando os mesmos não estão sendo utilizados. Apesar de utilizarem a mesma rede de comunicação, a interação entre as aplicações ainda é tímida e, na maioria das vezes, é feita manualmente. Estas aplicações foram utilizadas nas décadas de 1980, 1990 e 2000 e algumas ainda são utilizadas atualmente. Uma forma de interação muito comum entre aplicações desta geração é a integração de uma

aplicação qualquer com uma de monitoramento por câmeras. Um operador humano executa ações na primeira aplicação de acordo com o que ele pode ver nas câmeras.

Finalmente na terceira geração, os sistemas inteligentes, além de interconectados, são capazes de aprender, e adaptar suas funções de monitoramento e de controle ao ambiente monitorado. Ou seja, o sistema de decisão para aplicações de terceira geração é capaz de, em tempo real, apreender e adaptar suas funções de controle e monitoramento de acordo com dados obtidos do monitoramento do ambiente e das outras aplicações implantadas no mesmo edifício integrando todas as aplicações. Esta integração entre as diversas aplicação é percebida como se todo o edifício tivesse apenas uma grande aplicação que controla toda a automatização do mesmo. Não há ainda soluções comerciais de sistemas da terceira geração disponíveis no mercado.

Em (NUNES 1995) são apresentadas diversos exemplos de aplicações de primeira e segunda geração para edifícios inteligentes que podem fazer uso de RASSFs. Estas mesmas aplicações podem ser adaptadas ou estendidas para poderem ser classificadas como aplicações de terceira geração. Podemos destacar entre os diversos exemplos de aplicações (i) sistemas de Aquecimento, Ventilação e Ar Condicionado (AVAC) para controle de temperatura, (ii) Sistemas de controle de luminosidade e iluminação, (iii) Sistemas de controle da qualidade do ar e de janelas, (iv) Sistema de monitoramento de atividades de equipamentos, (v) Eletrodomésticos inteligentes e (vi) Sistemas de segurança e proteção.

**Tabela 1. Alguns tipos de sensores utilizados pelas aplicações descritas.
Fonte (WEBER 2009)**

	AVAC	Luminosidade e iluminação	Qualidade do ar e janelas	Monitoramento de equipamentos	Eletrodomésticos inteligentes	Segurança e proteção
Temperatura e Calor	X					X
Luminosidade		X				
Movimento e Ocupação	X	X	X	X	X	X
Fumaça			X		X	X
Estado		X	X	X	X	X
Quebra de vidros					X	X

Todas estas aplicações utilizam vários tipos de sensores diferentes para monitorar diversos eventos físicos. Embora algumas implementações das aplicações difiram quanto a quais tipos de sensores são utilizados, há um consenso sobre a maioria dos tipos de sensores utilizados. A Tabela 1 mostra quais eventos são monitorados pelas aplicações apresentadas.

2.2.2 Tipos de Sensores Utilizados em Aplicações para Edifícios Inteligentes

De acordo com a Tabela 1, embora as aplicações sejam totalmente distintas quanto a seu objetivo, elas podem utilizar diversos tipos de sensores em comum. No entanto, estes sensores freqüentemente são interpretados de forma diferentes em aplicações distintas, embora o evento físico monitorado seja medido da mesma forma.

Os sensores de temperatura e calor medem o valor absoluto da temperatura em um determinado ambiente. Exemplos de aplicações que utilizam este tipo de sensor são os sistemas AVAC e de segurança e proteção. Estas medições podem ser usadas para manter uma sala a uma temperatura agradável ou para detectar a localização de focos de incêndio e acionar alarmes, dependendo da aplicação.

Os sensores de luminosidade detectam o nível de luminosidade incidente no mesmo. Na Tabela 1 somente os sistemas de controle de luminosidade utilizam este tipo de sensor. Nestas aplicações eles são usados para regular a intensidade da iluminação não natural ou para controlar cortinas ou persianas, de forma a obter um melhor aproveitamento da luz natural ou para movimentar painéis solares a fim de maximizar a geração de energia limpa.

Os sensores de movimento e ocupação são os mais utilizados nas aplicações exemplificadas. Eles detectam qualquer tipo de movimento no ambiente monitorado ou ainda a presença de pessoas mesmo que imóveis. Entre as diversas utilidades deste tipo de sensor em aplicações para edifícios inteligentes podemos destacar o desligamento de equipamentos ou da iluminação não natural na ausência de pessoas ou o acionamento de sistemas de monitoramento por câmeras na presença delas.

Os sensores de detecção de fumaça verificam a presença de fumaça no ambiente monitorado. Exemplos de aplicações que utilizam este tipo de sensor são sistemas de controle de qualidade do ar e de proteção e segurança. Estes sensores podem ser utilizados tanto para verificar a qualidade do ar, como para detectar incêndios em uma determinada sala.

Já os sensores de estado mostram o estado de determinadas estruturas da construção inteligente sendo utilizados em praticamente todas as aplicações. Eles, em geral, são associados a algum tipo de equipamento e determinam o estado uma determinada função

(ligado/desligado, aberto/fechado, etc). Pode-se, por exemplo, saber se as persianas de uma janela estão abertas ou fechadas pelos dados transmitidos por um sensor de estado acoplado ao dispositivo que as controlam. Eles funcionam como alertas para que outros sensores possam fazer análises mais detalhadas de determinados tipos de problemas.

Finalmente os sensores de quebra de vidros são utilizados em sistemas de segurança para evitar acessos indevidos e para acionar outros sistemas. Por exemplo, caso uma janela esteja quebrada em uma determinada sala, o sistema de refrigeração pode ser desligado neste cômodo.

Embora esta listagem de tipos de sensores seja bem abrangente existem diversos outros tipos de sensores que já são utilizados em aplicações para edifícios inteligentes que não estão presentes nesta listagem. Entre eles podemos citar: sensores de umidade (CHRUNGOO et al. 2011), de vibração/aceleração (LEMOS et al. 2011), de produtos químicos (HAYES et al. 2008), de pressão (SINGH et al. 2011), de forças magnéticas (SIFUENTES et al. 2011), óticos (SIFUENTES et al. 2011), micro-câmeras integradas à RASSF (ERICKSON et al. 2009), entre outros.

2.2.3 Integração entre Aplicações para Edifícios Inteligentes

O conceito de integração de aplicações é aplicado nos edifícios inteligentes a partir da segunda, mas principalmente na terceira geração. A integração de aplicações nos edifícios inteligentes é alcançada através da comunicação entre aplicações, ou seja, através da troca de informação e da colaboração para execução de tarefas comuns.

A integração entre as aplicações pode influenciar qualquer momento do processo de decisão. Uma aplicação integrada a outra pode (i) compartilhar dados de monitoramento ou (ii) tomar decisões baseadas no estado de equipamentos ou em decisões ou eventos de outra aplicação. Segundo (NUNES 1995), aplicações consideradas de segunda geração oferecem uma série de vantagens em relação às de primeira geração. Entre estas vantagens podemos citar: (i) utilização mais eficiente dos recursos disponíveis, (ii) maior precisão das operações devido a correlação de informações, (iii) facilidade de gerência das aplicações integradas e (iv) melhor relação funcionalidade/custo.

Nas aplicações para edifícios inteligentes classificadas de terceira geração, o sistema de decisão é descentralizado, instalado em nós sensores sem fio, de forma que seja possível em tempo real adaptar suas funções de controle e monitoramento de acordo com o ambiente monitorado. Isto proverá vantagens adicionais às apresentadas para a segunda geração.

Exemplos destas vantagens são: (i) Disseminação de mudanças no modo de funcionamento das aplicações, (ii) Integração dinâmica de novas aplicações e (iii) Aprendizado sobre comportamento padrão no edifício.

3 Trabalhos Relacionados

Existem diversos trabalhos na literatura que apresentam propostas para eficiência energética de edifícios inteligentes. Entretanto, muitos deles propõem soluções específicas para uma aplicação, como, por exemplo: (SINGHVI et al. 2005), (WEN et al. 2006), (PARK et al. 2007), (TACHWALI et al. 2007), (PAN et al. 2008), (DELANEY et al. 2009), (HIROYASU et al. 2009), (AGARWAL et al. 2010), (MOHAMADDOUST et al. 2010), (YEH et al. 2010) e (CHRUNGOO et al. 2011). Outros trabalhos, como (ERICKSON et al. 2009), (MAESTRE e CAMACHO 2009), (SCHOR et al. 2009), (FILIPPONI et al. 2010), (LIU et al. 2010), (BYUN e PARK 2011), (GEZER e BURATTI 2011), (FORTINO e GUERRIERI 2011) e (SUN et al. 2011), apesar de serem capazes de controlar mais de uma aplicação, fazem uso da RSSF como infraestrutura de comunicação para transmitir os dados coletados pelos sensores para uma entidade central no edifício, não aproveitando sua capacidade, ainda que reduzida, de processamento paralelo. Os mecanismos de controle e de decisão destes trabalhos estão localizados fora da RSSF, em uma entidade central de processamento. Diferentemente desses trabalhos, o CONDE descentraliza o mecanismo de controle e decisão entre os nós da própria RASSF, valendo-se da capacidade de processamento destes nós, e assim provendo uma economia de energia da RASSF, dentre outros fatores, pela proximidade entre os locais onde a decisão é tomada e onde ela é executada. Assim, como as grandezas monitoradas não são transmitidas para uma entidade centralizadora, há um ganho em tempo de resposta e, conseqüentemente, se economiza energia pela redução no número de mensagens transmitidas.

O trabalho de (ERICKSON et al. 2009) apresenta uma prova de conceito de um sistema simples que coleta dados periodicamente de uma RSSF de micro-câmeras por um determinado período a fim de criar, em um computador fora da RSSF, modelos de predição da ocupação do edifício para ajustar o controle de uma aplicação AVAC. Estes modelos de predição podem, em trabalhos futuros, ser usados para ajustar o funcionamento de, virtualmente, qualquer aplicação em um edifício inteligente através dos modelos de predição da ocupação do edifício gerados. O principal diferencial entre o sistema proposto em (ERICKSON et al. 2009) e o CONDE é que as ações não acontecem em tempo real no primeiro, portanto o CONDE é potencialmente mais eficiente neste aspecto.

Em (SCHOR et al. 2009), a aplicação proposta para edifícios inteligentes faz uso de uma RSSF responsável por coletar dados e transmiti-los quando solicitados através de um Serviço Web (*Web Service*). Cada nó sensor tem um endereço IP único na rede e implementa

um mini servidor HTTP. Para tal, foram implementados nos nós sensores os protocolos IPv6, TCP e HTTP, os quais se referem, respectivamente, às camadas de rede, de transporte e de aplicação. Os sistemas de controle e decisão são comandados por regras simples orientadas a eventos. Diferentemente do mecanismo proposto em (SCHOR et al. 2009), o sistema proposto no presente trabalho é descentralizado, o que produz, como mencionado anteriormente, um ganho em tempo de resposta pela execução mais rapidamente das decisões, que, potencialmente, se reflete em economia de tempo e energia.

Já a proposta apresentada em (MAESTRE e CAMACHO 2009) mostra um sistema de automatização de baixo custo para residências, baseado no protocolo Universal Plug & Play (UPnP), a fim de integrar diversas aplicações para edifícios inteligentes. A arquitetura deste protocolo consiste de três componentes distintos: servidores, apresentadores e controladores. Os servidores provêm um determinado recurso (como uma aplicação) a um apresentador remoto. Já os apresentadores utilizam um determinado recurso provido por um servidor. Finalmente, os controladores medem a interação (utilizando alguma métrica como o número de mensagens, por exemplo) entre os apresentadores e os servidores. Há somente uma entidade controladora, que fica localizada em um computador, que faz o controle centralizado de toda a comunicação entre os componentes. As RSSFs são citadas como uma das entidades que podem ser integradas à solução como componentes servidores, porém não são componentes essenciais da arquitetura proposta. Além disto, caso haja algum problema no computador que centraliza o processo decisório, o funcionamento de todo o sistema fica comprometido. Diferentemente desse trabalho, o mecanismo de decisão proposto é totalmente descentralizado e realizado na íntegra pelos nós da RASSF, trazendo potenciais vantagens inerentes de tempo de resposta e economia de energia.

O trabalho de (FILIPPONI et al. 2010) apresenta uma proposta de arquitetura baseada em eventos para monitoramento de espaços públicos com sensores heterogêneos em cidades inteligentes. O objetivo desse trabalho é prover um arcabouço para a implementação de serviços de monitoramento de áreas públicas e infraestruturas genéricas. A arquitetura proposta utiliza o paradigma publicação/subscrição, onde elementos chamados *Knowledge Processors* (KPs) podem produzir e consumir notificações de eventos. Estas notificações são produzidas por um observador de um evento e descrevem este evento. Elas têm um significado fixo e pode ser grandezas sensoriadas ou um comando a ser executado. As notificações são distribuídas para outros KPs que se inscreveram neste tipo de evento através dos *Semantic Information Brokers* (SIBs) e estes as repassam para as aplicações. Esse

trabalho prevê a utilização de RSSF como ferramenta para monitoramento de grandezas físicas, entretanto os nós sensores não fazem nenhum destes papéis descritos. Os nós sensores enviam os dados coletados para fora da RSSF, onde há um KP que interpreta estes dados, cria uma notificação de evento e a publica. Ou seja, a RSSF não faz parte da arquitetura proposta. O maior diferencial do trabalho de (FILIPPONI et al. 2010) e o sistema proposto neste trabalho é que todas as funções do sistema são implementadas dentro da RASSF, que é composta por nós sensores de baixo consumo de energia.

Em (LIU et al. 2010) é proposta uma aplicação para controle de sistemas de iluminação e do estado de equipamentos e estruturas, como janelas, por exemplo. Essa proposta utiliza dois componentes principais. O primeiro é responsável pelo monitoramento de eventos físicos no ambientes por meio de uma RSSF. O outro componente é responsável pelo tratamento dos dados vindos do primeiro componente (agregações e avaliações quanto à qualidade) e contém os mecanismos de controle e decisão. Baseado nos dados de entrada fornecidos pela RSSF, a aplicação proposta é capaz de: disparar alarmes caso alguma medição ultrapasse um determinado limiar; monitorar o total de recursos utilizados por um usuário ou aplicação; determinar quais computadores estão disponíveis em um laboratório; e determinar a localização de um visitante no edifício, guiando-o até seu destino. A proposta apresentada neste trabalho difere de (LIU et al. 2010) por se tratar de um sistema totalmente descentralizado nos nós da RASSF.

O trabalho de (GEZER e BURATTI 2011) apresenta um protótipo utilizando RASSF para cenários de aplicações a serem utilizados no projeto eDIANA (*Embedded Systems for Energy Efficient Buildings*). Uma das maiores inovações apresentadas do projeto eDIANA é a criação de perfis de consumo energético para cada tipo de equipamento (*appliance*) utilizado em edifícios ou residências inteligentes. A arquitetura apresentada do projeto eDIANA é muito semelhante à arquitetura do CONDE. O ambiente monitorado é dividido em células, onde cada célula é comandada por um Dispositivo Concentrador da Célula (*Cell Device Concentrator* - CDC) e tem diversos nós periféricos. Cada célula é independente para tomar decisões com base nos dados providos pelos dispositivos periféricos da própria célula e em diretivas enviadas por uma estação base, denominada Dispositivo Macro-Concentrador de Células (*Macro-Cell Concentrator* - MCC). É direta a associação da arquitetura apresentada com a arquitetura do CONDE (CDC é semelhante ao componente que escalona as atividades do CONDE e MCC é semelhante ao componente que faz a interface entre a RASSF e a redes externa). No entanto os resultados apresentados ainda são iniciais e não são apresentados

resultados relacionados a economia de energia do edifício inteligente, tornando impraticável a comparação dos resultados obtidos neste trabalho. Além disso, o trabalho de (GEZER e BURATTI 2011), apesar de tratar várias aplicações, se quer menciona a integração destas aplicações, de forma que este é um diferencial do CONDE em relação a proposta apresentada.

4 CONDE: Sistema de Controle e Decisão para Edifícios Inteligentes

Este trabalho apresenta o CONDE, um sistema de controle e decisão descentralizado, denominado CONDE, para aplicações voltadas para edifícios inteligentes que utilizam RASSFs. Nesta proposta, os nós da RASSF realizam a coleta e a manipulação de informações dos edifícios, bem como a tomada de decisões sobre ações referentes às aplicações. Além disso, o sistema de controle e decisão proposto permite integrar diferentes aplicações através do compartilhamento de dados de monitoramento e pelo encadeamento de tomadas de decisões em aplicações distintas.

As próximas seções detalham o sistema CONDE da seguinte forma: a seção 4.1 descreve a arquitetura lógica do CONDE, apresentando seus componentes de software e como os mesmos se relacionam. A seção 4.2 apresenta a arquitetura física do sistema proposto, expondo como os componentes de software são implantados nos dispositivos físicos. Finalmente, a seção 4.3 detalha o funcionamento do sistema proposto como um todo através de um estudo de caso, descrevendo passo-a-passo as interações entre os elementos descritos nas seções anteriores. Há também neste capítulo o detalhamento do funcionamento dos subsistemas contidos no sistema proposto e de como é definida a interface entre os mesmos.

4.1 Arquitetura Lógica do CONDE

A arquitetura lógica do CONDE (Figura 2) consiste de: quatro componentes de software, (i) Gestor de Decisão, (ii) Gestor de Monitoramento, (iii) Gestor de Atuação e (iv) Gestor da RASSF; dois subsistemas, o de (i) Inferência, que contém o componente Método de Inferência e a base de dados Parâmetros de Inferência, e o de (ii) Integração, que contém o componente Método de Integração e a base de dados Parâmetros de Integração; e, finalmente, da base de dados Aplicações. As bases de dados são especificadas por arquivos XML, um para cada base, de forma que é necessário que se conheçam todas as aplicações suportadas pelo CONDE a priori. Tais bases de dados são criadas e têm dados inseridos nas mesmas antes da inicialização do CONDE.

O Gestor de Decisão é o componente central da arquitetura e é responsável por gerenciar o funcionamento dos outros componentes e dos subsistemas e por coordenar as ações executadas no ambiente monitorado. Este componente é também responsável por selecionar os dados enviados pelo componente Gestor de Monitoramento para cada processo de decisão de cada aplicação, e por enviar as decisões tomadas para o componente Gestor de Atuação. Para realizar esta seleção, o Gestor de Decisão consulta a Base de dados Aplicações, que contém dados sobre os parâmetros das aplicações, tais como seus

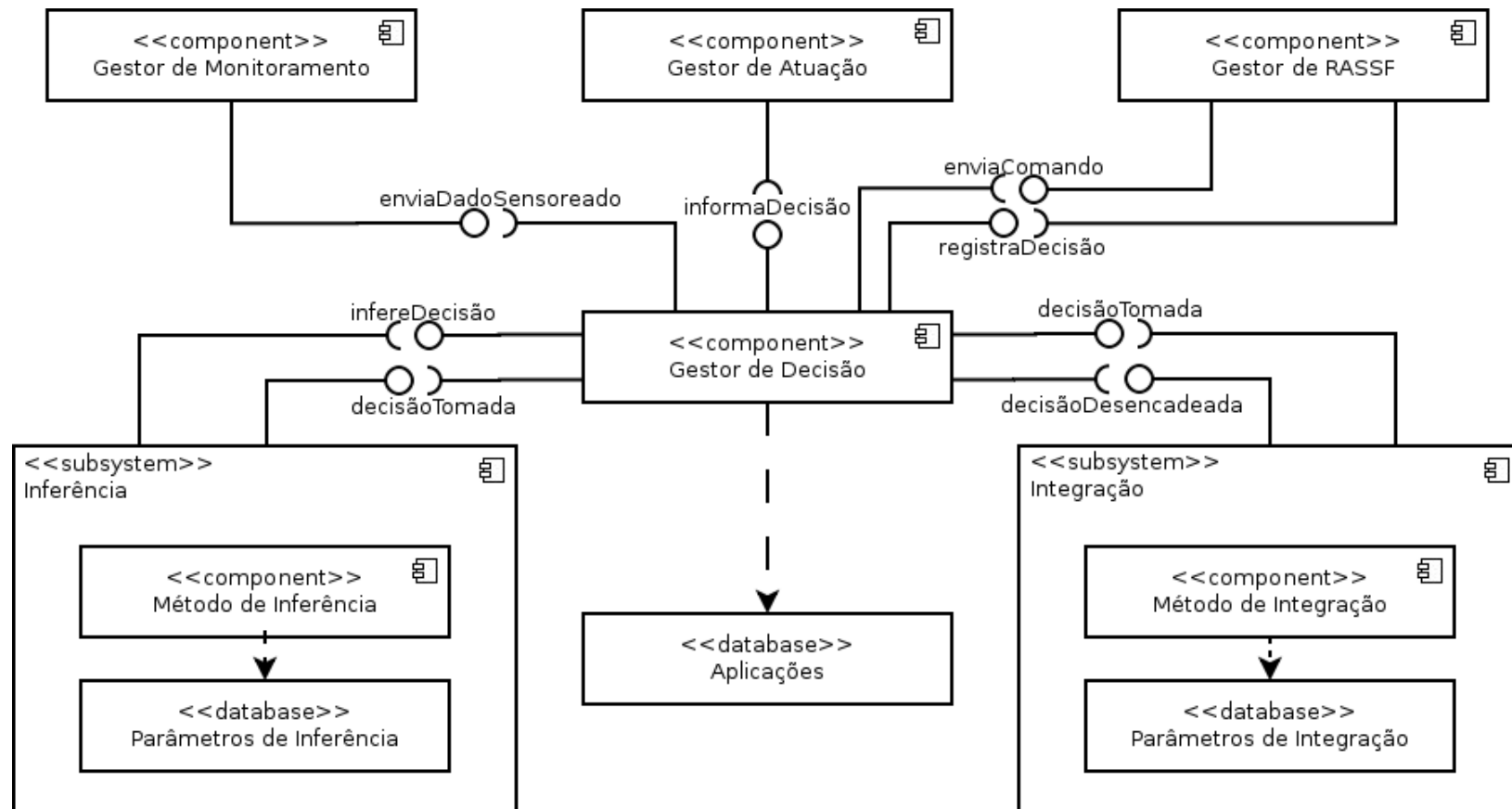


Figura 2. Diagrama de componentes UML da arquitetura lógica do sistema proposto.

identificadores, os tipos de sensores que são utilizados por cada aplicação e a periodicidade com que o processo de decisão de cada aplicação é disparado. A frequência de leitura e envio dos dados sensoriados é definida no **Gestor de Monitoramento**.

O componente **Gestor de Monitoramento** coordena as tarefas de monitoramento dos dados físicos do edifício, ou seja, este componente deve estar localizado em um dispositivo físico capaz de realizar o sensoriamento. O componente constitui basicamente em uma API com os dispositivos de sensoriamento existentes em um dado nó sensor. Cada nó pode possuir um ou mais dispositivos de sensoriamento, capazes de sensoriar diferentes variáveis ambientais, como luminosidade, temperatura, vibração, etc. É importante observar que diferentes aplicações podem solicitar a coleta de um mesmo tipo de dado, mas a periodicidade da coleta pode ser diferente para cada aplicação. Neste caso, adota-se a menor periodicidade da coleta entre as aplicações que solicitam a coleta de um mesmo tipo de dado a fim de atender a ambas. Além disto, os dados físicos são coletados independentemente do seu valor semântico para as aplicações a que se destinam. Por exemplo, quando o **Gestor de Monitoramento** obtém uma leitura de um sensor de presença, esta leitura indica apenas se alguma forma de movimento foi detectada. Somente quando esta leitura é enviada para o **Gestor de Decisão** e é associada a uma aplicação, ela pode indicar eventos de significado mais complexo como: *há um intruso no ambiente monitorado* ou *não desligue os elevadores, pois há pessoas dentro dele*. Este valor semântico só é levado em consideração durante o processo de inferência (Seção 4.1.1), momento em que este dado coletado é associado a uma aplicação.

O componente **Gestor de Monitoramento** envia periodicamente os dados coletados para o **Gestor de Decisão**. A fim de minimizar a quantidade de envios, o componente **Gestor de Monitoramento** armazena o último dado enviado de cada unidade de sensoriamento e compara cada nova leitura com este valor. Caso haja uma variação significativa entre o dado armazenado e a nova leitura, ou seja, se a diferença entre estas leituras ultrapassar o limiar determinado para este tipo de dado, o dado é enviado para o **Gestor de Decisão**. Caso contrário, ou seja, se a diferença entre o dado armazenado e a nova leitura não for considerada significativa, esta informação não é enviada, economizando assim a energia da transmissão na rede.

O componente **Gestor de Atuação** é responsável por gerenciar a atuação do sistema sobre o ambiente monitorado, ou seja, é quem executa as decisões enviadas pelo **Gestor de Decisão**. Por motivo de eficiência energética da RASSF, este componente deve estar

localizado em um dispositivo físico capaz de atuar ativamente no ambiente monitorado, ou seja, nos chamados atuadores. Os atuadores podem conter chaves elétricas, relés, transistores, LEDs e etc. Seu papel é ligar/desligar lâmpadas, refrigeradores ou outros dispositivos presentes no edifício. O componente **Gestor de Atuação** recebe do **Gestor de Decisão** um identificador da aplicação e os valores de saída provenientes dos **Subsistemas Inferência e Integração**.

O componente **Gestor da RASSF** faz a interface entre a RASSF e redes externas, como por exemplo, a Internet. Este componente é utilizado para monitorar a operação do CONDE para fins de manutenção ou auditoria e para enviar comandos para o **Gestor de Decisão** a partir de um computador fora da RASSF. O componente **Gestor da RASSF** atua como um *gateway* ou um *relay* entre a rede externa e a RASSF, isto é, este componente não interpreta as mensagens recebidas, mas somente as encaminha de uma rede para a outra. Este processo de encaminhar as mensagens entre as redes é realizado por aplicações simples, que utilizam a API de comunicação com a RASSF para intermediar esta troca de mensagens. As mensagens recebidas são as decisões provenientes do **Gestor de Decisão** e os comandos recebidos de fora da RASSF.

O **Subsistema de Inferência**, descrito mais detalhadamente na Seção 4.1.1, é responsável por decidir, com base nos dados coletados obtidos pelo componente **Gestor de Monitoramento**, e encaminhados pelo **Gestor de Decisão**, quais ações serão executadas pelo componente **Gestor de Atuação**. O componente **Método de Inferência** pertence ao **Subsistema Inferência** e obedece a interface definida por este subsistema. A definição desta interface permite que este componente seja substituído por outro que faça a inferência utilizando outro método aplicável, tornando a arquitetura do CONDE mais flexível. Este componente depende da base de dados **Parâmetros de Inferência** que guarda os dados das aplicações ativas e quaisquer outros dados utilizados pelo componente **Método de Inferência** para a decisão. Essas decisões são específicas para cada aplicação. Para tomar as decisões referentes a uma dada aplicação, o **Subsistema Inferência** utiliza o componente **Método de Inferência**, que consulta a base de dados **Parâmetros de Inferência** desta aplicação, a qual contém um conjunto de regras que serão utilizadas na decisão (o componente **Método de Inferência** está descrito em detalhes na Seção 5.2). Qualquer decisão tomada neste subsistema é encaminhada para o **Subsistema Integração** para determinar se qualquer outra decisão de uma aplicação integrada deve ser encadeada a ela.

O **Subsistema Integração** é responsável por realizar a integração e cooperação entre diferentes aplicações para edifício inteligente, de forma a possibilitar compartilhamento de dados e encadeamento de decisões levando em conta o estado de equipamentos e as decisões tomadas em aplicações distintas. O componente **Método de Integração** pertence ao **Subsistema Integração** e obedece a interface definida por este subsistema. A definição desta interface permite que este componente seja substituído por outro que faça a integração utilizando um método diferente. O componente **Método de Integração** recebe do **Gestor de Decisão** uma mensagem contendo um identificador da aplicação que está sendo avaliada e o valor das variáveis de saída referentes a esta aplicação, vindos do **Subsistema Inferência**. A partir desses dados, o componente **Método de Integração** consulta a base de dados **Parâmetros de Integração** de forma a verificar quais aplicações estão integradas à aplicação avaliada. Em seguida, o **Subsistema Integração** retorna ao **Gestor de Decisão** uma mensagem contendo um conjunto de dados referentes a cada uma das aplicações integradas a aplicação avaliada. Esse conjunto de dados contém, para cada aplicação integrada, o identificador de aplicação e um valor de saída representando uma ação referente a esta aplicação, numa estrutura de dados idêntica à saída do **Subsistema Inferência**. Decisões acionadas no **Subsistema Integração** são enviados, juntamente com a decisão tomada no **Subsistema Inferência**, para o **Gestor de Decisão**, que as encaminha para o **Gerenciador de Atuação**.

A integração de aplicações se dá através do cruzamento de variáveis de saída de uma dada aplicação, vindas do **Subsistema Inferência**, via **Gestor de Decisão**, com variáveis de saída de outras aplicações presentes no edifício inteligente. Na base de dados **Parâmetros de Integração**, por exemplo, a aplicação de iluminação está correlacionada à aplicação de AVAC pela variável de saída “controleIllum”. Está especificado na base de dados que, caso a variável de saída “controleIllum” da aplicação de iluminação esteja no estado “Desligar”, as variáveis de saída da aplicação de AVAC também devem se encontrar no estado “Desligar”. Então o **Subsistema Integração** envia uma mensagem ao **Gestor de Decisão** com o identificador da aplicação de AVAC e o valor referente às variáveis de saída no estado “Desligar”. Esta mensagem será enviada ao **Gestor de Decisão** para que as ações sejam executadas posteriormente. Da mesma forma que há essa regra de integração, existem outras que compõem a base de dados e dependem do conjunto de aplicações presentes no sistema. Mais detalhes sobre o **Subsistema Integração** estão descritos na seção 4.1.2.

4.1.1 Descrição do Subsistema Inferência

O Subsistema Inferência é responsável por processar os dados coletados no Gestor de Monitoramento e tomar decisões para cada aplicação ativa no edifício inteligente. Este subsistema é composto por um componente e uma base de dados, a saber: Método de Inferência e Parâmetros de Inferência, respectivamente. O Método de Inferência é quem define como os dados coletados serão processados a fim de tomar decisões referentes às aplicações ativas no sistema proposto. Estas decisões são tomadas com base em um conjunto de parâmetros e regras que regulam o processo de tomada de decisões em cada aplicação. Este conjunto de parâmetros e regras fica armazenado na base de dados Parâmetros de Inferência. Independente do método de inferência escolhido, a máquina de inferência se utiliza destes diferentes conjuntos de parâmetros e regras para avaliar os dados de cada aplicação.

A base de dados Parâmetros de Inferência define a semântica das variáveis de entrada e saída do método de inferência escolhido, ou seja, somente a partir deste ponto o significado dos dados coletados é levado em consideração e, conforme explicado anteriormente, este significado depende da interpretação no contexto específico de cada aplicação. As variáveis de entrada do Método de Inferência são os dados coletados pelas unidades de sensoriamento associados a cada aplicação e as variáveis de saída consistem dos tipos de decisão que a aplicação pode tomar. Por exemplo, na aplicação AVAC as variáveis de entrada são temperatura e presença de pessoas e as de saída são variação de temperatura e intensidade de ventilação. Assim, decisões tomadas na aplicação AVAC podem ajustar o termostato do ar condicionado a fim de atingir a temperatura considerada ideal armazenada na base de dados Parâmetros de Inferência e/ou acionar a ventilação do ambiente com o mesmo propósito.

A arquitetura do Subsistema Inferência é genérica, a fim de permitir que o sistema proposto possa utilizar qualquer método de inferência que o administrador do CONDE ache conveniente, bastando para isso substituir o componente Método de Inferência e sua base de dados Parâmetros de Inferência. A única restrição imposta pelo sistema proposto é que este novo componente implemente uma interface de comunicação pré-definida. Esta interface exige que o componente Método de Inferência receba via comando os dados coletados e dispare um evento com as decisões tomadas. O comando de entrada deve receber uma dupla {A, M} vinda do componente Gestor de Decisão, onde “A” é o identificador da aplicação em questão e “M” é o conjunto das variáveis de entrada desta aplicação, que são os dados

coletados associados a esta aplicação. O evento acionado pelo componente Método de Inferência deve conter a dupla {A, D}, onde “A” é o identificador da aplicação em questão e “D” é o conjunto de variáveis de saída desta aplicação, que simbolizam as decisões tomadas pelo Subsistema Inferência.

Maiores detalhes sobre o a versão atual do componente Método de Inferência e da base de dados Parâmetros de Inferência utilizados serão detalhados na Seção 5.2.

4.1.2 Descrição do Subsistema de Integração

O Subsistema Integração é responsável por integrar as diversas aplicações no edifício inteligente pelo encadeamento das decisões de uma aplicação com outra decisão tomada em uma aplicação distinta. Este subsistema é composto por um componente e uma base de dados, a saber: componente Método de Integração e base de dados Parâmetros de Integração. O componente Método de Integração (detalhado na Seção 5.3) é quem define quais ações serão desencadeadas dada uma decisão tomada no Subsistema Inferência, ou seja, que algoritmo será utilizado para tratar os dados enviados pelo Subsistema Inferência via Gestor de Decisão a fim de integrar as aplicações. A escolha das ações desencadeadas é baseada nos parâmetros e regras contidas na base de dados Parâmetros de Integração. Estes parâmetros e regras regulam o processo de escolha das ações desencadeadas em cada aplicação integrada àquela indicada pelos dados de entrada. Independente do Método de Integração, os dados necessários para a escolha das ações desencadeadas devem estar armazenados nesta base de dados.

A base de dados Parâmetros de Integração define a natureza da integração entre duas aplicações, ou seja, como as decisões tomadas passam a ter um significado no Subsistema Inferência, onde este significado pode ser associado a um contexto em que o ambiente monitorado está inserido. As ações desencadeadas no Subsistema Integração estão inseridas em um determinado contexto e agregam ao mesmo significado. Por exemplo, se uma decisão na aplicação de AVAC consiste em desligar o ar condicionado em uma sala, isto não significa obrigatoriamente que esta sala está vazia. No entanto, se a aplicação de iluminação, integrada a aplicação de AVAC, desliga também as lâmpadas desta sala, esta segunda ação agrega um significado mais amplo à primeira ação, que, neste contexto, aumenta a possibilidade de a sala, de fato, estar vazia.

Os dados de entrada do componente Método de Integração são exatamente os dados de saída do Subsistema Inferência, ou seja, o conjunto de decisões tomadas em uma das

aplicações ativas no edifício inteligente. Por exemplo, digamos que, em um dado instante, uma decisão seja tomada na aplicação Controle de Iluminação, a qual consiste em apagar as luzes de uma determinada sala. Está determinado em uma das regras da base de dados **Parâmetros de Integração** que esta decisão indica que não há mais ninguém nesta sala, então as ações desencadeadas neste contexto seriam: desligar o controle de temperatura na aplicação AVAC. A mera presença desta regra na base de dados **Parâmetros de Integração** indica que as aplicações Controle de Iluminação e AVAC estão integradas no nível de encadeamento de decisões. Neste cenário, o **Subsistema Integração** recebe como dado de entrada a decisão tomada no **Subsistema Inferência** informando que as lâmpadas devem ser desligadas e retorna como dados de saída duas outras decisões informando que os sistemas de ar condicionado e ventilação também devem ser desligados.

A arquitetura do **Subsistema Integração** é genérica a fim de permitir que o CONDE utilize qualquer método de integração que se ache conveniente pela simples substituição do componente **Método de Integração** e da base de dados **Parâmetros de Integração**. A única restrição imposta pelo CONDE é que o componente **Método de Integração** utilize a interface de comunicação pré-definida. Esta interface exige que o componente **Método de Integração** receba via comando as decisões tomadas no **Subsistema Inferência** e dispare um evento indicando as ações encadeadas aos dados de entrada. O comando de entrada deve receber uma dupla $\{A, D\}$ do **Gestor de Decisão**, onde “A” é o identificador da aplicação e “D” é o conjunto de valores de saída desta aplicação, que simbolizam as decisões tomadas pelo **Subsistema Inferência**. O evento disparado pelo **Subsistema Integração** deve conter um conjunto finito de duplas $\{\{A_1, I_1\}, \{A_2, I_2\}, \dots\}$, onde “ A_i ” é o identificador da aplicação integrada e “ I_i ” é o conjunto de valores de saída associados à “ A_i ”, que simbolizam as ações disparadas por esta aplicação, as quais agora estão encadeadas à decisão tomada no **Subsistema Inferência**.

Maiores detalhes sobre o a versão atual do componente **Método de Integração** e da base de dados **Parâmetros de Integração** utilizados serão detalhados na Seção 5.3.

4.2 Arquitetura Física do CONDE

Neste trabalho, utiliza-se uma RASSF composta por diversos nós sensores, nós atuadores e uma estação base. Os nós sensores podem assumir dois papéis diferentes na rede, a saber: (i) Nó Sensor Decisor (NSD); e (ii) Nó Sensor Ordinário (NSO). Os nós atuadores e a estação base podem assumir um único papel, Nó Atuador (NAT) e Nó Estação Base (NEB),

respectivamente. Papéis dizem respeito às responsabilidades de cada nó e às suas funcionalidades.

Durante a fase de projeto do CONDE, a possibilidade de se utilizar apenas um papel para os nós sensores foi considerada. Nesta abordagem, os nós sensores seriam responsáveis pelo monitoramento, inferência e gestão de decisões. Os nós sensores, a fim de manter a cobertura de sensoriamento sobre todo o ambiente monitorado, deveriam compartilhar os dados sensorizados entre si, de forma que todos os nós recebessem os dados necessários para tomar decisões. De posse dos mesmos dados de entrada, todos os nós sensores tomariam decisões idênticas em seus processos de inferência e transmitiriam ações repetidas para os Gestores de Atuação. Devido a estas repetições, este cenário é ineficiente em relação à energia gasta na RASSF. Por outro lado, no cenário com dois papéis para os nós sensores, apenas um o nó sensor seria responsável pela inferência e gestão de decisões e os demais ficariam responsáveis pelo monitoramento. Os nós sensores responsáveis pelo monitoramento enviam os dados sensorizados para o nó responsável pela inferência e gestão de decisões, que realiza o processo de inferência uma única vez e transmite ações apenas uma vez. Como o processo de inferência é realizado em apenas um nó sensor e as ações são transmitidas uma só vez, foi identificada uma diminuição significativa de processamento e de pacotes transmitidos, respectivamente, que motivou a escolha do cenário com dois papéis.

Os componentes de software foram implantados nos dispositivos físicos como apresentado na Figura 3. O NSD contém o componente **Gestor de Decisão**, a base de dados **Aplicações** e os **Subsistemas Inferência e Integração**. NSO, NAT e NEB contêm apenas um único componente, **Gestor de Monitoramento**, **Gestor de Atuação** e **Gestor da RASSF**, respectivamente. Os NSOs são os nós sensores em que estão localizadas as unidades de sensoriamento e contêm o componente **Gestor de Monitoramento**. Eles coletam os dados do ambiente monitorado e os enviam via rádio de acordo com as regras do componente **Gestor de Monitoramento**. Os NSDs são nós sensores que não precisam conter unidades de sensoriamento. Eles recebem os dados monitorados dos NSOs, tomam decisões e transmitem estas decisões para os NATs, para que estas sejam executadas no ambiente monitorado, e para o NEB, para fins de manutenção e auditoria do sistema. Além disto, os NSDs podem receber comandos vindos de fora da RASSF pelo NEB. Os NATs são nós atuadores, que só recebem as mensagens enviadas pelos NSDs e as executam. O NEB é uma estação base que, conforme explicado anteriormente, recolhe os dados enviados pela RASSF e é capaz de enviar comandos para os NSDs vindos de fora da RASSF.

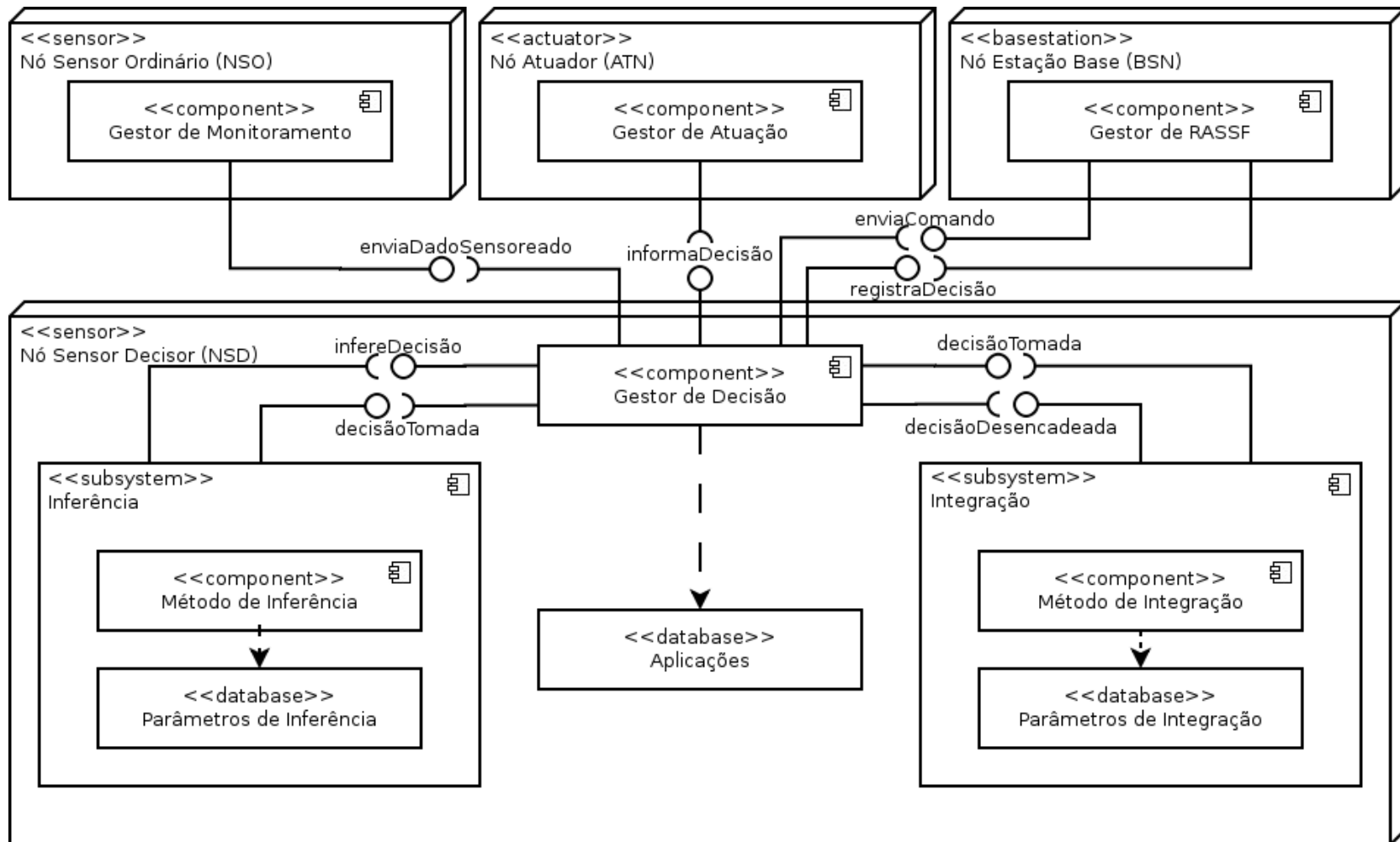


Figura 3. Diagrama de componentes da arquitetura física do sistema proposta

4.3 Funcionamento do Sistema de Decisão

O fluxo de operação do sistema proposto é dividido em duas fases: inicialização e ciclo de operação. Na primeira, são determinados os parâmetros de funcionamento do sistema. É nesta fase que as bases de dados são criadas e os dados necessários ao funcionamento do sistema são inseridos nas mesmas, isto é, (i) são definidas as aplicações a serem utilizadas no edifício inteligente em que a RASSF será implantada, (ii) são configurados/inicializados os parâmetros destas aplicações e (iii) são definidos os métodos de inferência e integração, bem como seus respectivos parâmetros. As informações sobre as aplicações e seus parâmetros são armazenadas na memória flash de armazenamento no momento da instalação do código do CONDE no nó sensor. Os sensores a serem ativados na rede dependem da aplicação. Na segunda fase, o CONDE entra em um laço composto de uma série de procedimentos por aplicação. Estes procedimentos são detalhados na Seção 4.3.2.

Para descrever com mais detalhe cada uma dessas fases e como elas ocorrem no cenário de uma RASSF, considera-se como estudo de caso um edifício onde estão implantadas as aplicações AVAC, Controle da Qualidade do Ar e de Iluminação e Monitoramento de Equipamentos. Nestas aplicações, os NSO são programados para monitorar periodicamente temperatura, luminosidade, presença de fumaça e de pessoas e estado de equipamentos, e enviar estes dados coletados a um NSD. A RASSF é sincronizada de forma os sensores podem ser configurados para intercalarem períodos de ativação (onde capturam dados e trocam mensagens) e períodos de dormência (onde são desligados para economizar energia).

4.3.1 Fase de Inicialização

O Gestor de Decisão realiza três passos na fase de inicialização: (i) as bases de dados Aplicações, Parâmetros de Inferência e Parâmetros de Integração são criadas no componente Gestor de Decisão e nos Subsistemas Inferência e Integração, respectivamente, (ii) os dados referentes às aplicações a serem executadas no edifício são adicionados a suas respectivas bases de dados, e (iii) informa ao componente Gestor de Monitoramento que as unidades de sensoriamento de temperatura, luminosidade e presença de fumaça e de pessoas devem ser ativadas, assim como os respectivos períodos de tempo durante os quais os dados devem ser coletados. Em seguida, o componente Gestor de Decisão inicia a fase de operação.

4.3.2 Fase de Operação

A fase de operação funciona conforme ilustrado no diagrama de sequência apresentado na Figura 4. O **Gestor de Decisão**, para cada aplicação ativa, dispara um conjunto de procedimentos. Inicialmente, ele recebe do **Gestor de Monitoramento** os dados coletados pelas unidades de sensoriamentos e agrupa estes dados de acordo com os tipos de dados utilizados por cada aplicação, conforme armazenado na base de dados Aplicação. Por exemplo, estão associadas à aplicação AVAC as unidades de sensoriamento que monitoram temperatura e presença de pessoas, à aplicação Controle de Iluminação as que monitoram luminosidade, presença de pessoas e estado dos equipamentos, à aplicação Controle da Qualidade do Ar as que monitoram presença de fumaça e de pessoas e estados dos equipamentos e à aplicação Monitoramento de Equipamentos as que monitoram presença de pessoas e estado de equipamentos. O componente **Gestor de Monitoramento** coleta estes dados, independentemente de qual aplicação está executando/ativa no momento, e os envia para o **Gestor de Decisão**.

Em seguida, estes dados coletados de diferentes unidades de sensoriamento são agrupados por aplicação e enviados para o **Subsistema Inferência**, o qual os processa e toma as decisões pertinentes para cada aplicação separadamente. Por exemplo, *caso não haja pessoas em uma sala não é necessário ajustar sua temperatura*, ou *caso a luminosidade natural de um determinado ambiente diminua deve-se intensificar a iluminação artificial*. Tomadas estas decisões, o **Subsistema Inferência** envia uma mensagem para o **Gestor de Decisão** informando o resultado da sua decisão e associando o identificador da aplicação com tal resultado. Por exemplo, no caso da aplicação de AVAC, é criada e enviada para o **Gestor de Decisão** uma mensagem contendo (i) o identificador da aplicação, (ii) um valor que indica a variação de temperatura que o termostato deve ajustar a fim de atingir a temperatura ideal e (iii) um valor que indica a intensidade em que o sistema de ventilação deve funcionar. Já no caso da aplicação Controle de Iluminação, uma mensagem contendo (i) o identificador da aplicação e (ii) o valor de ajuste de luminosidade é criada e enviada para o **Gestor de Decisão**. O **Gestor de Decisão** envia essa mensagem para o **Subsistema Integração**.

Neste momento, de posse dos dados recebidos do **Subsistema Inferência** via o **Gestor de Decisão**, o **Subsistema Integração**, de acordo com o método de integração escolhido, consulta a base de dados **Parâmetros de Integração** em busca das ações que deverão ser realizadas sobre o ambiente monitorado em resposta a mensagem recebida. Por

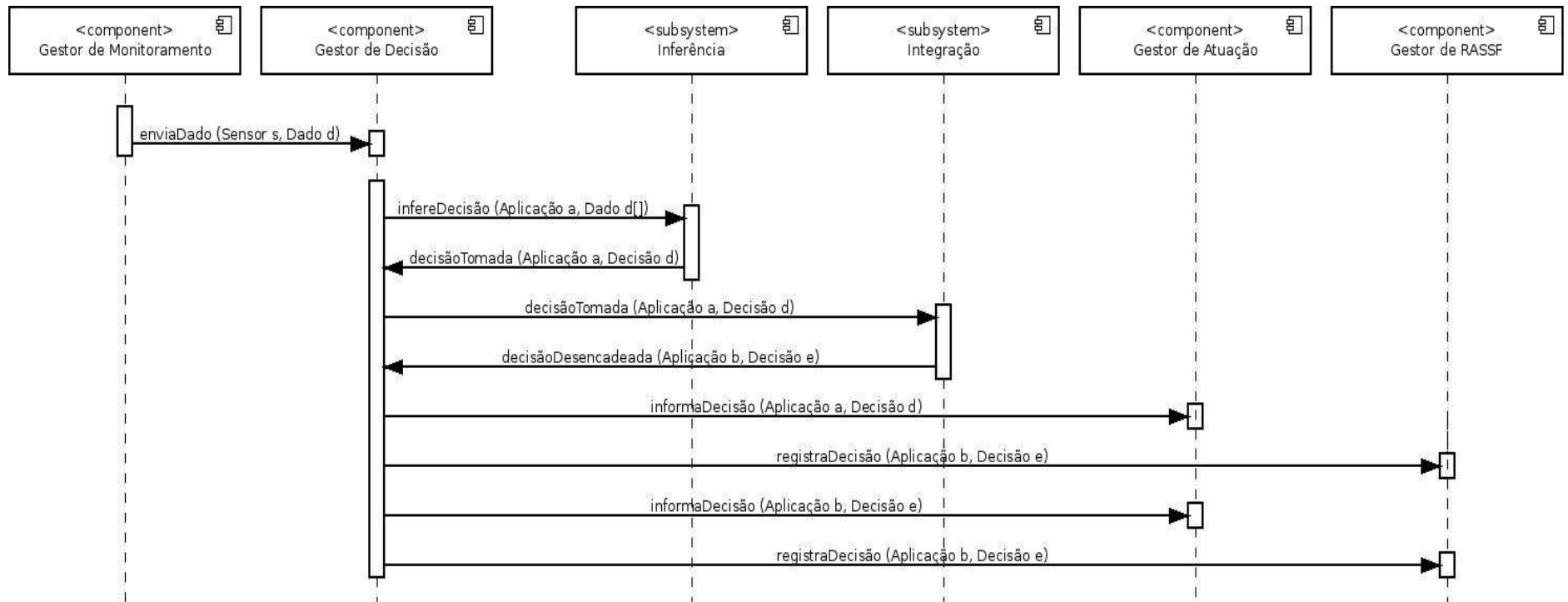


Figura 4. Diagrama de sequência da fase de operação do CONDE

exemplo, suponha que foi recebida uma mensagem com o identificador da aplicação AVAC que instrui o CONDE a desligar os controles de temperatura e ventilação da aplicação. Independentemente das leituras dos sensores de luminosidade, a aplicação de iluminação deve instruir o desligamento das lâmpadas, pois a sala provavelmente está vazia. Assim, a decisão de desligar a aplicação de AVAC, tomada no **Subsistema Inferência**, é encadeada com a decisão de desligar a aplicação de luminosidade. Então, ambas as decisões são enviadas ao **Gestor de Decisão**, que, por sua vez, vai encaminhá-las aos componentes **Gestor de Atuação** e **Gestor da RASSF**.

O **Gestor de Atuação** recebe os dados enviados pelo **Gestor de Decisão** indicando quais decisões, sejam elas originárias do **Subsistema Inferência** ou **Integração**, foram tomadas e devem ser executadas sobre o ambiente monitorado e, a seguir, as executa.

5 Avaliação Experimental

Foram realizados cinco experimentos para avaliar o CONDE. Quatro deles tiveram por objetivo confirmar (i) que é viável implantá-lo em uma RASSF, (ii) que os resultados produzidos pelo simulador podem ser equiparados a resultados obtidos em experimentos executados em uma plataforma real de sensores, dessa forma validando as simulações conduzidas e (iii) que há, de fato, benefícios em se descentralizar o sistema. O último experimento visa apresentar os resultados relacionados à eficiência energética do edifício inteligente, que é o principal problema tratado neste trabalho.

Dos cinco experimentos realizados, quatro deles foram simulados, exceto o terceiro, que abrange uma configuração de ambiente simulada e uma configuração real. O primeiro experimento realiza uma calibração do CONDE, de forma a avaliar o número ideal de NSOs por NSD implantado no edifício a fim de obter os melhores resultados em termos de tempo de resposta.

O segundo busca avaliar o CONDE sob o aspecto de consumo de recursos, ou seja, visa determinar se é viável implantar o sistema em uma RASSF real, com seus tipicamente escassos recursos.

O terceiro experimento tem como objetivo comparar os resultados obtidos na simulação com os obtidos em experimentos reais em termos de atraso, energia consumida pelos nós sensores e eficiência do sistema, a fim de validar os resultados simulados. Para esse experimento são utilizadas as métricas de avaliação descritas na Seção 5.4.

O quarto experimento testa o CONDE em termos de tolerância a falhas, avaliando seu comportamento quando múltiplos nós apresentam mal-funcionamento. Neste experimento, o CONDE foi testado levando em conta um modelo de falhas a fim de verificar a sua resistência em termos de funcionalidade do sistema quando há nós na rede apresentando falhas de parada, ou *fail-stop failures* (TANENBAUM e VAN STEEN 2007). Para isso, é considerada a existência de nós defeituosos que deixam de se comunicar durante a operação do sistema devido a falta de energia ou falhas de hardware. Com a finalidade de simular este comportamento de falha, alguns nós tiveram o seu código propositalmente alterado para deixar de interagir completamente com a RASSF. Neste experimento, o número de nós defeituosos variaram entre 0 e 4, de um em um, para cada um dos papéis atribuídos aos nós sensores. O objetivo deste experimento é provar que a descentralização do sistema contribui para sua resistência global a falhas dos nós da RASSF.

Finalmente, no quinto experimento, foi realizada uma comparação do sistema CONDE em seu estágio final de desenvolvimento com (i) o CONDE em um estágio intermediário de desenvolvimento, quando o mesmo não possuía o **Subsistema Integração**, e com (ii) a solução proposta em (TSAI et al. 2008), que apresenta um sistema de controle e decisão centralizado, dessa forma permitindo medir e comparar a eficiência energética do edifício inteligente alcançada nas três abordagens a fim de avaliar os benefícios da descentralização do sistema e da integração das aplicações.

Em todos os experimentos realizados, duas aplicações foram consideradas: AVAC e iluminação. As unidades de sensoriamento utilizadas foram temperatura, luminosidade e presença. A atividade de atuação foi emulada pelo acendimento de LEDs nos atuadores, independentemente do tipo de ação a ser executada pelos mesmos.

5.1 Configuração do Ambiente

Nos experimentos realizados, tanto em ambiente real quanto simulado, uma RASSF composta por sensores estáticos da plataforma MICAz, comercializados pela MEMSIC Technology (MEMSIC 2010), foi utilizada. Os sensores da plataforma MICAz, utilizados em todos os experimentos deste trabalho, são dotados de 4kB de memória RAM, 128 kB de memória flash para armazenamento do programa e 512 kB para armazenamento de dados, e alimentados por duas pilhas AA (1202 mAh), que juntas contém aproximadamente 16000 J de energia (KRÄMER e GERALDY 2006).

O CONDE foi desenvolvido utilizando o ambiente de desenvolvimento TinyOS (LEVIS e GAY 2009), versão 2.1.1, e usando a linguagem nesC (LEVIS e GAY 2009), uma extensão da linguagem C, que implementa um modelo de programação orientado a eventos. O TinyOS disponibiliza diversos componentes de software, incluindo os componentes que implementam a pilha de protocolos de comunicação para RSSF/RASSF. Cada componente TinyOS possui uma interface bem definida, implementada por meio de funções que são caracterizadas como tratadores de eventos ou comandos. Os experimentos reais foram realizados em ambiente fechado e controlado (laboratório). Os experimentos simulados foram executados no simulador Avrora (ALBEROLA 2011).

Os protocolos roteamento e controle de energia adotados são específicos para RASSF e são definidos por padrão no TinyOS. São eles: *Collection Tree Protocol* (CTP) (GNAWALI et al. 2009) para efetuar o roteamento dados em direção ao sorvedouro e *Integrated Concurrency and Energy Management* (ICEM) (KLUES et al. 2007) para sincronização de

transmissão dos nós da RASSF. Esta sincronização é feita quando cada nó é ligado, se integrando à RASSF, e mantida através do envio de mensagens de ajuste periódicas. Estes foram escolhidos por serem os protocolos padrão do TinyOS e por serem específicos para RASSF.

O CONDE foi implementado definindo-se sete novos componentes para o TinyOS. Os componentes ManagerC, DecisionMakerC e MonitorC implementaram respectivamente as funcionalidades dos componentes Controle, Gestor de Decisão, Monitoramento. O Subsistema Inferência foi implementado em dois componentes do TinyOS denominados InferenceC e InferenceSystemC. O Subsistema Integração também foi implementado em dois componentes do TinyOS denominados IntegrationC e IntegrationSystemC. Os Subsistemas Inferência e Integração foram desenvolvidos em dois componentes separados para que a substituição dos métodos de inferência ou integração pudesse ser feita de forma simples pela substituição dos componentes InferenceSystemC e IntegrationSystemC, respectivamente, sem grandes reformulações no código, que foi mantido o mais modular possível.

5.2 Descrição do Método de Inferência usado no protótipo

O processo de inferência realizado pelo CONDE depende do método de inferência escolhido. Teoricamente, qualquer método de inferência pode ser utilizado através da substituição do componente Método de Inferência por outro, desde que este implemente a mesma interface. A base de dados Parâmetros de Inferência é dependente do componente Método de Inferência, pois cada método de inferência utiliza os dados de forma distinta. Ela é preenchida durante a fase de inicialização do CONDE.

Para a versão atual do CONDE, o método de inferência escolhido consiste no uso de bases de regras simples, uma para cada aplicação, que consistem de parâmetros de entrada e decisões associadas e são armazenadas na base de dados Parâmetros de Inferência. O CONDE toma decisões comparando os dados de entrada com os parâmetros presentes na base de dados. Se os dados de entrada satisfizerem as condições de uma determinada regra, a decisão associada a ela é transmitida para o Gestor de Decisão.

A base de dados Parâmetros de Inferência contém as variáveis de entrada e saída do sistema e suas regras. As regras são definidas por intervalos associados a cada variável de entrada e valores de ajuste associados a cada variável de saída. Estes intervalos definem que valores as variáveis de entrada devem indicar para que uma regra seja acionada. Estes valores

de ajuste definem que ações devem ser transmitidas pelo CONDE para implantação no edifício. Desta forma, a definição genérica de uma regra de inferência aceita pelo CONDE é:

- se (entrada_1 \in intervalo_1) e (entrada_2 \in intervalo_2) e (entrada_3 \in intervalo_3) e ... então (saída_1 \rightarrow ajuste_1) e (saída_2 \rightarrow ajuste_2) e (saída_3 \rightarrow ajuste_3) e ...

Todas as regras utilizadas nos experimentos estão listadas no Anexo 2. Por exemplo: temperatura e presença estão associadas à aplicação de controle de AVAC, e podem estar nos intervalos [0, 50] e [0, 100], respectivamente, e as variáveis de saída representam tipos de decisão, como ajustar o termostato do ar-condicionado em um número determinado de graus e a intensidade da ventilação no ambiente. Assim, segundo a definição anterior de regra, um exemplo poderia ser:

- se (temperatura \in [25,31.75[) e (presença \in [50,62.5[) então (termostato -3.0 °C) e (ventilação +25%)

5.3 Descrição do Método de Integração usado no protótipo

O processo de integração realizado pelo CONDE depende do método de integração escolhido. Teoricamente qualquer método de integração pode ser utilizado através da substituição do componente Método de Integração por outro, desde que este implemente a mesma interface. A base de dados Parâmetros de Integração é dependente do componente Método de Integração, pois cada método de inferência utiliza os dados de entrada de forma distinta. Ela é preenchida durante a fase de inicialização do CONDE.

Para a versão atual do CONDE, o método de integração escolhido consiste no uso de bases de regras simples que podem encadear decisões estáticas aos dados de entrada, uma para cada tipo de decisão. Os dados de entrada consistem de decisões tomadas no Subsistema Inferência. Cada decisão é tratada individualmente pelo Subsistema Integração, mesmo que duas ou mais pertençam a mesma aplicação e tenham sido tomadas simultaneamente. Neste método, o espaço de entrada é dividido em sub-espacos de tamanho fixo e a cada um destes é associado um conjunto de decisões estáticas. Ou seja, para quaisquer valores de entrada pertencentes a um mesmo sub-espaco, o Subsistema Integração encadeará sempre as mesmas decisões a ele.

Para cada tipo de decisão passível de ser tomada por uma aplicação do edifício inteligente, há um conjunto de regras que é utilizado pelo Subsistema Integração do CONDE, os quais consistem de parâmetros de entrada e decisões associadas e são armazenados na base de dados Parâmetros de Integração. O CONDE encadeia decisões

comparando os dados de entrada com os parâmetros presentes na base de dados. Se os dados de entrada satisfizerem as condições de uma determinada regra, as decisões associadas a ela são encadeadas aos dados de entrada e transmitidas para o **Gestor de Decisão**.

A base de dados **Parâmetros de Integração** contém as variáveis de entrada e saída do sistema e suas regras. As regras são definidas por um identificador de aplicação e um intervalo de valores de ajuste associados uma variável de entrada desta aplicação e outro indicador de aplicação e valores de ajuste associados a cada variável de saída desta outra aplicação. Estes intervalos definem que valores as variáveis de entrada devem indicar para que uma regra seja acionada. Estes valores de ajuste definem que ações devem ser encadeadas às decisões já tomadas pelo CONDE para implantação no edifício. Desta forma, a definição genérica de uma regra de inferência aceita pelo CONDE é:

- se (aplicação = aplicação_1 e entrada \in intervalo) então (aplicação = aplicação_2 e (saída_1 \rightarrow ajuste_1) e (saída_2 \rightarrow ajuste_2) e (saída_3 \rightarrow ajuste_3) e ...)

Todas as regras utilizadas nos experimentos estão listadas no Anexo 3. Por exemplo: a quantidade de graus a ajustar no termostato de um ar condicionado é uma variável que representa uma decisão que pode ser tomada por uma aplicação de AVAC, e podem estar nos intervalos $[-7\text{ }^{\circ}\text{C}, 7\text{ }^{\circ}\text{C}]$, e a intensidade luminosa a ajustar no termostato de um sistema de iluminação artificial é uma variável que representa uma decisão que pode ser tomada por uma aplicação de Iluminação, e podem estar nos intervalos $[-128\text{ cd}, 128\text{ cd}]$. Estas duas variáveis pertencentes a duas aplicações distintas podem ser associadas por uma regra de integração. Assim, segundo a definição anterior de regra, um exemplo poderia ser:

- se (aplicação = AVAC e termostato $\in [-0,25\text{ }^{\circ}\text{C}; 0,25\text{ }^{\circ}\text{C}]$) então (aplicação = Iluminação e (luminosidade $\rightarrow -128\text{ cd}$))

5.4 Descrição do Cenário, Experimentos e Métricas

Para todos os experimentos, adotou-se uma rede de topologia plana e com nós estáticos tanto para os testes realizados em plataformas reais quanto simuladas. Cada experimento foi repetido 30 vezes, a fim de se obter um intervalo de confiança de 95%.

Foram utilizados dois cenários de testes distintos para os experimentos realizados com o CONDE. O primeiro cenário, como exemplificado na Figura 5, consistiu de uma área quadrada com 15m x 15m referente à simulação de uma única sala em que o CONDE seria implantado e foi projetado para que um único NSD localizado em seu centro pudesse receber dados enviados por outros nós em qualquer ponto da sala. Este cenário foi utilizado nos

experimentos de calibração, que não se beneficiaria da presença de mais de um nó NSD, e de comparação entre resultados obtidos em simulações e em experimentos reais, que foi restrito pela quantidade de plataformas de sensores reais que tínhamos disponíveis.

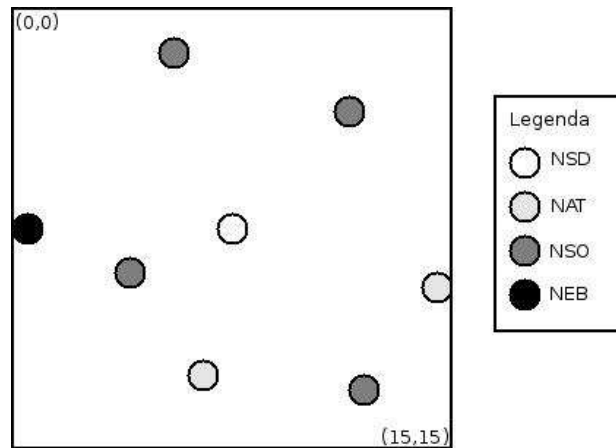


Figura 5. Representação gráfica do primeiro cenário utilizado nos experimentos

O segundo cenário, como exemplificado na Figura 6, foi utilizado nos experimentos realizados em (TSAI et al. 2008) e foi utilizado nos experimentos para permitir a comparação dos resultados obtidos neste trabalho com os de (TSAI et al. 2008). Neste segundo cenário, o edifício possui 12 andares e cada um destes andares contém 5 NSD dispostos de forma que estejam no alcance uns dos outros e pudessem se comunicar entre si. Além disso, o edifício possui um único NEB localizado no primeiro andar. Cada um destes NSDs possui 4 NSOs e 2 NATs dispostos aleatoriamente associados a ele. Assim o edifício totaliza 421 nós sensores, sendo 240 NSOs, 120 NATs, 60 NSDs e 1 NEB. Este cenário foi utilizado nos experimentos de avaliação do CONDE quanto ao consumo de recursos, de tolerância a falhas e de comparação do CONDE com outras soluções.

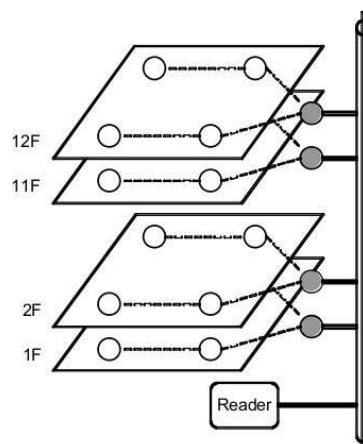


Figura 6. Representação gráfica do segundo cenário utilizado nos experimentos – Fonte: (TSAI et al. 2008)

As métricas utilizadas para a avaliação do sistema de decisão foram: (i) atraso de decisão do CONDE, (ii) energia consumida pelo edifício, (iii) energia consumida pela RASSF e (iv) eficiência do CONDE.

Define-se o *atraso de decisão do CONDE* como o tempo gasto entre o momento em que o sensoriamento é realizado até o momento em que a decisão é executada no edifício inteligente. Esta métrica é calculada, no caso dos experimentos simulados, através dos registros de tempo de cada evento emitidos automaticamente pelo simulador e, no caso dos experimentos reais, através de registros de tempo enviados em mensagens transmitidas por interfaces seriais.

Define-se a *energia consumida pelo edifício* como a quantidade de energia (kWh) gasta pelos equipamentos do edifício durante um determinado período de tempo. Esta métrica é avaliada através do consumo médio dos equipamentos do edifício (fornecido pelo *datasheet* de cada equipamento) e do intervalo de tempo durante o qual os mesmos estiveram ligados.

Define-se a *energia consumida pela RASSF* como a quantidade de energia, medida em joules (J), gasta pelos nós sensores, atuadores e a estação base, que compõem a RASSF, ao executar as aplicações em conjunto com o CONDE. Esta métrica é avaliada usando a ferramenta de análise de energia (Energy Analysis Tool) do simulador Avrora (ALBEROLA 2011), que mede a energia gasta com processamento, transmissão e recebimento de pacotes, uso de LEDs e acessos à memória flash. Esta métrica é, no entanto, tratada de forma diferente no terceiro experimento, pois uma etapa deste experimento envolve nós de uma plataforma real de sensores e estes não possuem uma API que permita obter a quantidade de energia exata gasta pelo sensor. Então, a fim de poder estimar a energia consumida neste experimento, esta métrica é avaliada através de uma aproximação feita a partir do número de mensagens transmitidas e recebidas (uma vez que é sabido que, para a maior parte das aplicações para RASSFs, a maior fonte de consumo de energia é com comunicação (DIETRICH e DRESSLER 2009)). A medição aproximada da energia consumida para avaliar o terceiro experimento foi baseada nos dados descritos a seguir e obtidos em (KRÄMER e GERALDY 2006).

A taxa de transmissão da interface de rede sem fio do dispositivo MicaZ é 4 $\mu\text{s/bit}$, a corrente elétrica no circuito para receber dados é 18,8 mA e para enviar dados é 17,4 mA e a voltagem base utilizada no dispositivo MicaZ é 3V. Assim, obtemos:

$$Q_{RX} = 3V * 18,8 \text{ mA} * 4 \mu\text{s/bit} = 0,2088 \mu\text{VAs/bit} = 0,2256 \mu\text{J/bit} \quad (1)$$

$$Q_{TX} = 3V * 17,4 \text{ mA} * 4 \text{ } \mu\text{s/bit} = 0,2088 \text{ } \mu\text{VAs/bit} = 0,2088 \text{ } \mu\text{J/bit} \quad (2)$$

Com base nas informações das Equações 1 e 2, foram adicionadas ao CONDE instruções com o objetivo de contar o número de mensagens recebidas e transmitidas; ao final do tempo do experimento, instruções para enviar esta contagem ao NEB para coleta e assim foi possível estimar a energia consumida no terceiro experimento.

A *eficiência do CONDE* é calculada como a razão entre a quantidade de decisões certas tomadas e o total de decisões tomadas pelo CONDE. Uma decisão é classificada como certa ou errada baseada nos registros das variáveis de entrada e de saída a cada processo de inferência e de integração realizado pelo CONDE. Como as grandezas físicas monitoradas eram previamente escolhidas, no caso simulado, ou conhecidas, no caso real, foi possível determinar que decisões deveriam ser sido tomadas a cada processo de decisão e, assim, classificá-las entre “certas” e “erradas”. Decisões erradas foram detectadas, sobretudo, em situações onde houve perdas de pacotes contendo dados de sensoriamento. No caso dos experimentos simulados, estes registros foram extraídos através de mensagens de controle emitidas junto aos registros de eventos já realizados pelo simulador e, no caso dos experimentos reais, foram extraídos por mensagens transmitidas por interfaces seriais.

5.4.1 Experimentos de Calibração

No primeiro experimento, simulações foram realizadas para calibrar os parâmetros do Sistema CONDE de forma a encontrar a configuração ideal quanto ao tempo de resposta do sistema enquanto se minimiza o consumo de recursos da RASSF. Neste experimento, variamos o número de NSOs, que enviam informações para um único NSD, de 1 a 10 sensores, de forma a identificar o número ideal de NSOs por NSD em termos do tempo necessário para a coleta dos dados requeridos pelo Subsistema Inferência. Os NSO foram dispostos aleatoriamente ao longo de uma área retangular 15x 15 m com o NSD tendo sido colocado no centro da área. Cada nó foi programado de forma a possuir uma identificação única e alcance de rádio omnidirecional fixo de 15 metros.

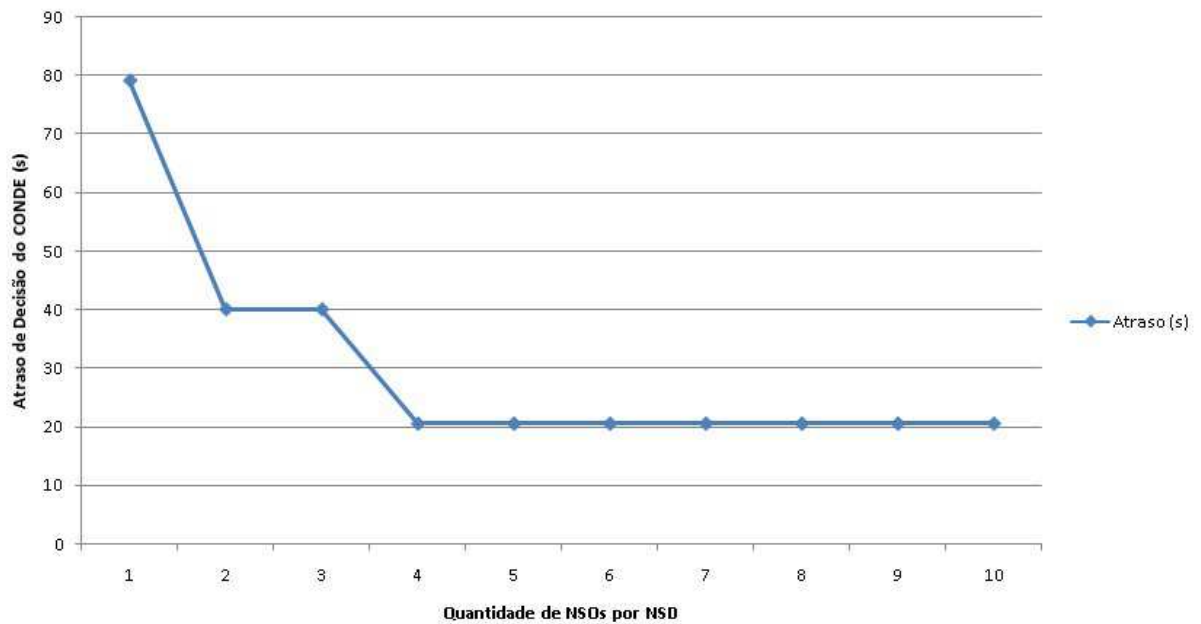


Figura 7. Tempo requerido para enviar os dados necessários ao processo de inferência

A Figura 7 mostra o tempo requerido para que o NSD receba todos os dados necessários para a inferência para diferentes quantidades de NSOs por NSD. Os melhores resultados são obtidos para 4 (quatro) ou mais NSOs por NSD, que foram de 20,637s em média. Tal resultado decorre do uso de um buffer de sensoriamento do NSD, que deve ser preenchido completamente antes que qualquer processo de inferência possa ser iniciado. Este buffer foi criado a fim de evitar que leituras provenientes de um sensor defeituoso não comprometam a decisão a ser tomada. Isto significa que, ao capturar mais amostras, eliminamos possíveis erros de leitura através da redundância. Assim, para o sistema CONDE adotou-se como sendo quatro o número de NSOs no ambiente monitorado para cada NSD.

5.4.2 Experimento para avaliar o consumo de recursos da RASSF

Neste segundo experimento, são realizadas medições para avaliar o impacto do CONDE sobre os nós da RASSF em termos de uso de CPU, uso de memória RAM, flash de programa e flash de armazenamento de dados, consumo de energia pelos nós sensores e o tempo de vida médio de todos os papéis atribuídos a eles, tendo em vista suas limitações computacionais e de energia.

A Tabela 2 mostra estatísticas de *uso de CPU* para cada papel atribuído aos nós da RASSF. O CONDE utiliza um componente nativo do TinyOS que sincroniza os nós, de forma que passem o máximo de tempo possível no modo de economia de energia. A Tabela 2 mostra que os NATs permanecem 99,078% do tempo de simulação em modo de economia de

energia, os NSDs, 97,774% e os NSOs, 98,578%, enquanto o NEB não entrou no modo de economia de energia. Esta diferença do NEB em relação aos outros papéis se dá em virtude do fato de que o NEB tem uma interface serial. Esta interface fica sempre ativa, logo a CPU fica em modo inativo (*idle*), que gasta mais energia que o modo de economia de energia (*sleep*), mas fornece uma fonte de energia alternativa pelo cabo serial para este sensor.

Tabela 2. Estatísticas de uso da CPU pelos nós da RASSF

Sensores	NAT	NEB	NSD	NSO
Tempo de CPU em modo ativo	0,243%	0,594%	0,743%	0,566%
Tempo de CPU em modo inativo (idle)	0,679%	99,406%	1,483%	0,856%
Tempo de CPU em modo de economia de energia	99,078%	0,0%	97,774%	98,578%

A Figura 8 mostra estatísticas sobre *o uso de memória RAM* para todos os papéis dos nós sensores. A quantidade de memória usada por cada papel é diretamente proporcional à sua complexidade. Pode-se notar que nenhum dos dispositivos (NAT, NEB, NSD e NSO) excedeu os 4 KB de memória disponível.

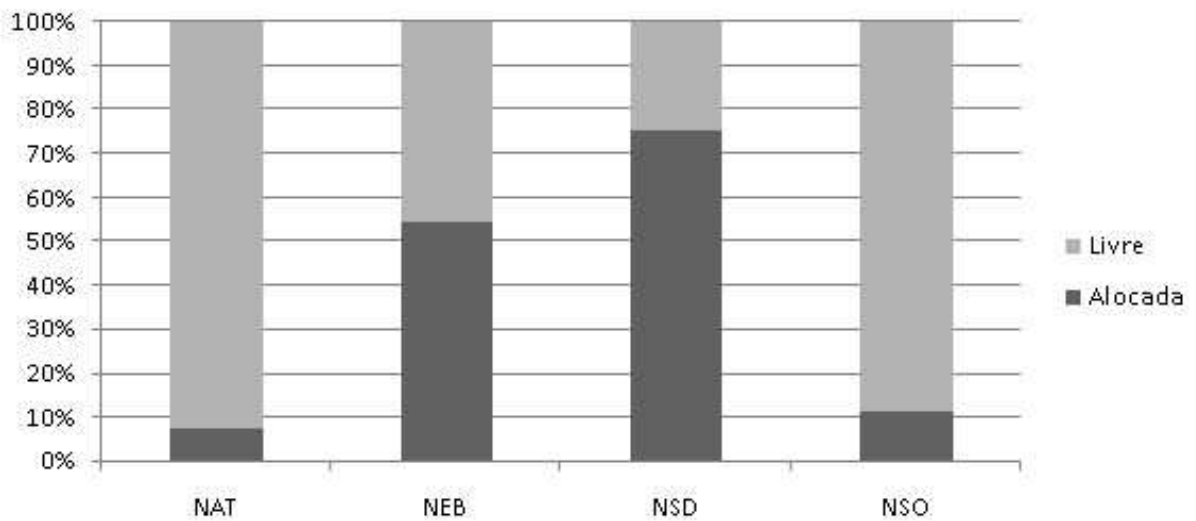


Figura 8. Estatísticas de uso da memória RAM pelos nós da RASSF

A Figura 9 apresenta estatísticas sobre *o uso da memória flash de programa* para todos os papéis dos nós sensores. Esta parte da memória flash dos nós sensores MICAz é usada para armazenar o arquivo objeto do software desenvolvido, isto é, estas estatísticas representam o impacto do CONDE nos nós em termos de memória. Como no caso anterior, a quantidade de memória usada por cada papel é diretamente proporcional à sua complexidade.

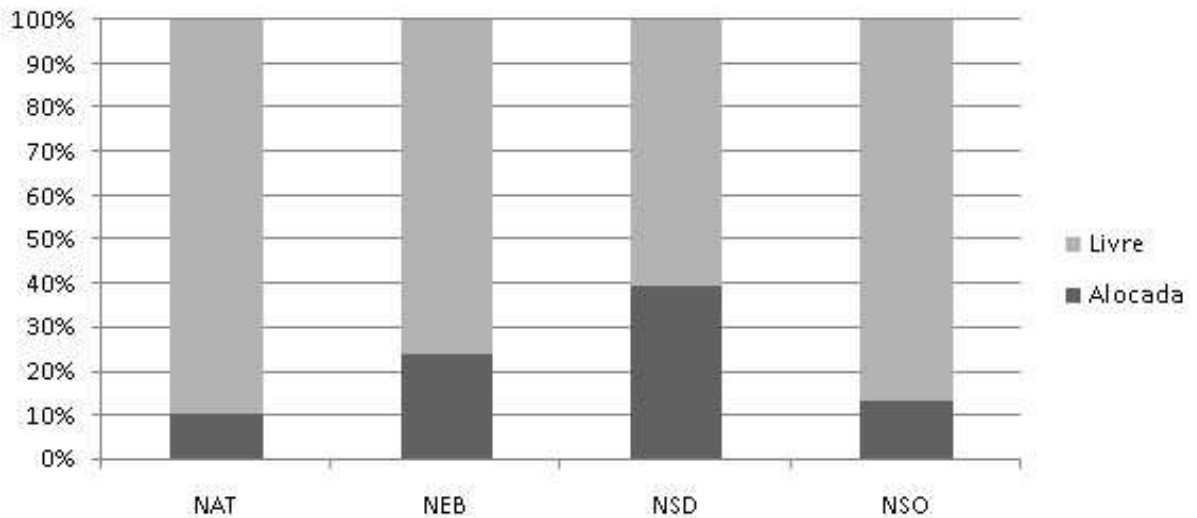


Figura 9. Estatísticas de uso da memória flash de programa pelos nós da RASSF

A Figura 10 mostra estatísticas sobre *o uso da memória flash de armazenamento de dados* para todos os papéis dos nós sensores. Esta parte da memória flash dos nós sensores MICAz é usada para armazenar dados de qualquer espécie para uso do CONDE durante sua execução. Todos os bancos de dados foram armazenados na memória flash de armazenamento de dados e, como todas as bases de dados ficam localizadas no NSD, apenas os nós do tipo NSD a utilizam. O uso da memória flash de dados aumentaria para os NSDs caso novas aplicações fossem inseridas, pois novas regras de inferência e integração seriam utilizadas para cada nova aplicação e estas seriam armazenadas nas respectivas bases de dados.

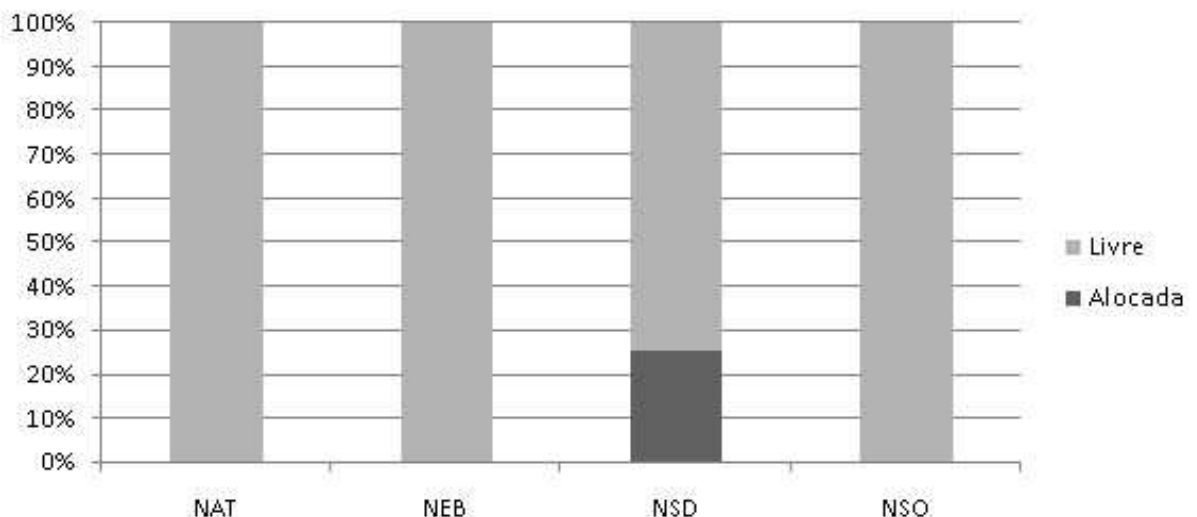


Figura 10. Estatísticas de uso da memória flash de dados pelos nós da RASSF

A Tabela 3 apresenta estatísticas sobre *a quantidade de energia gasta e o tempo de vida médio* de todos os papéis dos nós sensores. Como esperado, os NSDs gastam mais energia do que os NATs e os NSOs, mas o NEB gasta ainda mais energia. Isso acontece

porque o NEB nunca entra no modo de economia de energia, de modo que nunca dorme, gastando muito mais energia do que os outros nós. No entanto, como o NEB tem uma fonte de energia alternativa, a seu tempo de vida é virtualmente infinito e não limitado a energia da bateria. Quanto ao tempo máximo de vida da RASSF, ele fica restrito ao tempo de vida do NSD, pois este é o único papel fundamental para o funcionamento do CONDE. Ou seja, se um NAT, NEB ou NSO falha por falta de energia ou por qualquer outro motivo, o funcionamento do sistema não é comprometido; porém se o NSD falha o sistema para de funcionar na área coberta por este NSD. Portanto, o tempo máximo de vida do CONDE pode ser calculado como o tempo de vida dos nós NSD, que é de 1011,378 horas ou 42,14 dias de funcionamento ininterrupto. Se o funcionamento do CONDE ficar mais restrito, como durante o horário de expediente, por exemplo, este tempo máximo de vida tende a aumentar proporcionalmente, isto é, 112,38 dias.

Tabela 3. Estatísticas do consumo de energia pelos nós da RASSF

Sensores	NAT	NEB	NSD	NSO
Energia gasta (1 hora)	7,719 J	37,517 J	15,820 J	11,924 J
Tempo médio de vida	2072,807 h	—	1011,378 h	1341,832 h

5.4.3 Experimento com uma plataforma de sensores reais

Este experimento tem por objetivo validar os resultados obtidos em experimentos simulados, em termos de atraso, consumo de energia e eficiência do sistema, através da comparação com os resultados obtidos nos experimentos em plataforma de sensores reais. Ou seja, a finalidade do experimento é confirmar que os dados obtidos nas simulações de fato retratam os resultados que seriam obtidos se todos os testes fossem realizados em plataformas de sensores reais. Assim, o experimento foi realizado em duas etapas: a primeira em ambiente simulado e outra em plataformas de sensores reais. A etapa simulada foi realizada com o simulador Avrrora, como os experimentos anteriores, e a etapa real foi realizada com dispositivos MicaZ em ambiente controlado (no laboratório de Redes e Sistemas Distribuídos do iNCE/UFRJ).

Em termos de atraso de decisão, ambas as fases do experimento apresentaram comportamento semelhante, como mostrado na Figura 11. O atraso na fase simulada foi ligeiramente maior do que na fase real, sendo, em média, 33.142 e 33.057 ms, respectivamente.

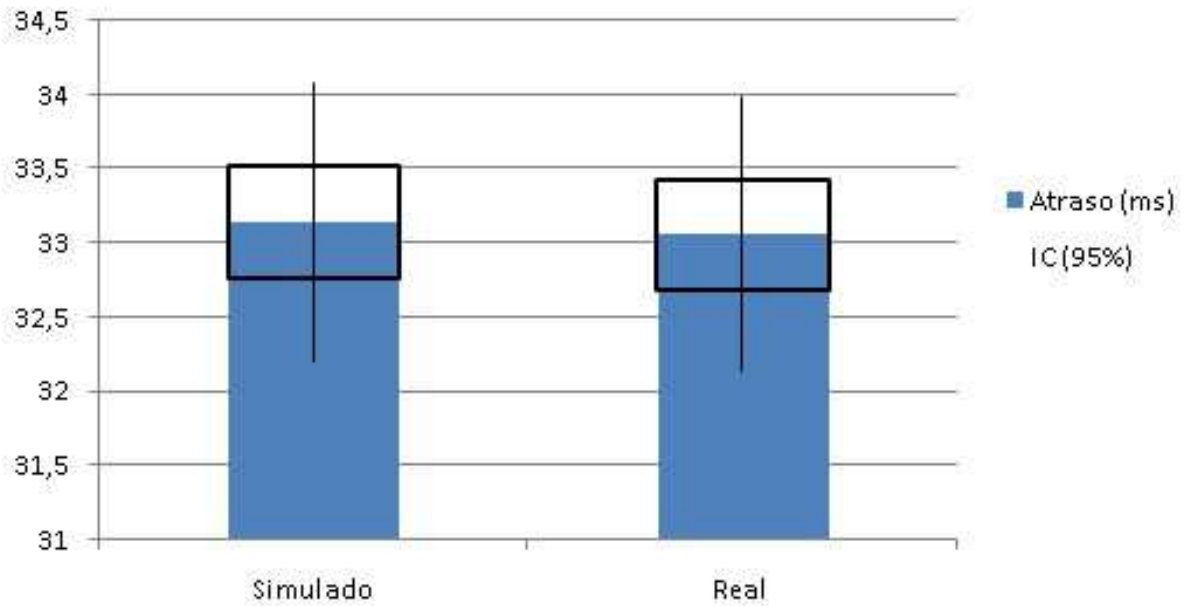


Figura 11. Comparação em termos de atraso

Em termos de consumo de energia, conforme observado na Figura 12 (onde DP é o desvio padrão e IC é o intervalo de confiança), a etapa real do experimento consumiu menos energia que a etapa simulada, em média 34.148 J e 34.835 J, respectivamente. Isto se deu como consequência de uma pequena diminuição na quantidade de pacotes recebidos. Esta diminuição foi atribuída a interferência na comunicação via rádio na RASSF que ocorre no ambiente real, a qual não foi prevista na simulação de ruídos utilizada na versão simulada do experimento.

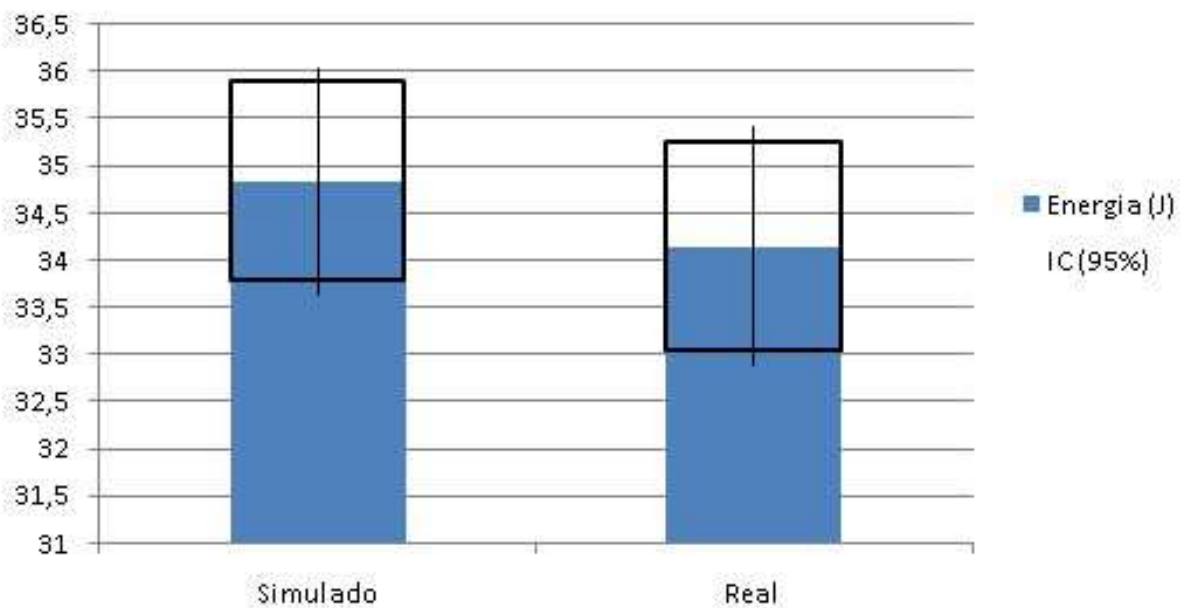


Figura 12. Comparação em termos de energia gasta pela RASSF

Em termos de eficiência do sistema, a fase simulada apresentou resultados ligeiramente melhores do que a fase real do experimento, como mostrado na Figura 13. Em média, a eficiência do sistema era 95.556% na fase simulada e 94.333% na fase real. Tal diferença pode ser explicada por interferência na comunicação via rádio na fase real que induzida alguma perda de pacote que não poderia ser comparada com a componente de simulação de ruído do Avrra.

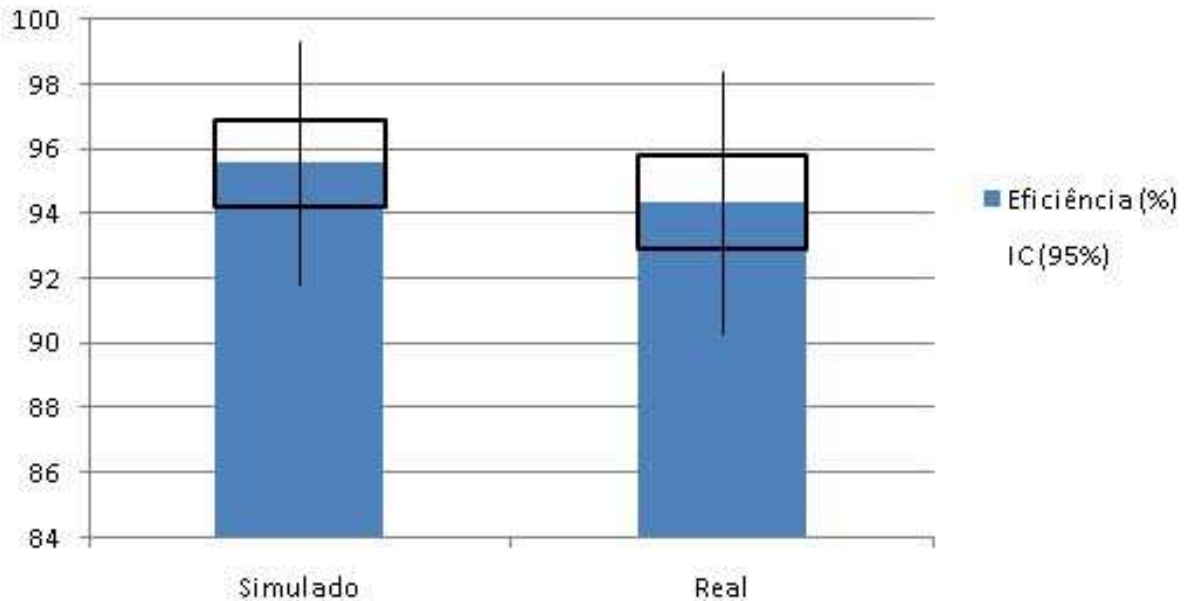


Figura 13. Comparação em termos de eficiência do sistema

5.4.4 Experimento para avaliar o CONDE em termos de tolerância a falhas

Neste experimento, foi avaliada a robustez do CONDE a falhas nos nós sensores da RASSF. O objetivo deste experimento é provar que a descentralização do CONDE entre os nós da RASSF melhora a tolerância do sistema como um todo a falhas de parada, ou *fail-stop failures* (TANENBAUM e VAN STEEN 2007). Este tipo de parada ocorre quando um nó pára de interagir completamente com o sistema, deixando de cumprir sua funcionalidade. Diferentemente dos sistemas centralizados, o CONDE é capaz de suportar algumas falhas em dispositivos que compõe a RASSF sem provocar a interrupção completa do serviço.

Para avaliar a resistência do CONDE a falhas, foram realizados testes utilizando o seguinte modelo de falhas. Para cada papel que os nós da RASSF podem assumir foram simuladas falhas de parada disparadas a partir de temporizadores. Ou seja, a simulação começa com todos os nós funcionando normalmente e, a cada período de 30 minutos, o CONDE é desligado completamente em um nó que representa o mesmo papel escolhido aleatoriamente. Este processo foi repetido até que 50% dos nós do tipo sendo testado

estivessem “falhado”. Assim foram feitas simulações para falhas de parada nos nós NAT, NEB, NSD e NSO e foi avaliado o impacto destas falhas sobre o CONDE em termos de atraso e de comprometimento do funcionamento do sistema. Também foram analisadas as implicações de falhas ocorridas em nós NAT ou NSO associados a um mesmo NSD e de falhas em NSDs sobre o esquema de roteamento da RASSF.

Os NATs são os dispositivos atuadores que executam as decisões tomadas nos NSD. Estes nós não interagem ativamente com nenhum nó da RASSF, ele somente recebem as decisões transmitidas pelos NSDs e as executam. Portanto falhas de parada em dispositivos NAT não têm implicações sobre o atraso e não comprometem o funcionamento do sistema em si. O único efeito que as falhas de parada causam sobre o sistema é que as decisões tomadas não são executadas sobre o ambiente. Assim este tipo de falha é facilmente detectável por um operador ou usuário.

Tabela 4. Efeito de falhas sobre os NATs

Falhas	Atraso	Funcionamento	Efeito
1 NAT	—	—	Este NAT não executa as decisões enviadas
2 NATs / NSDs distintos	—	—	Estes NATs não executam as decisões enviadas
2 NATs / NSD único	—	—	Estes NATs não executam as decisões enviadas
3 NATs / NSDs distintos	—	—	Estes NATs não executam as decisões enviadas
3 NATs / NSD único	—	—	Estes NATs não executam as decisões enviadas

O NEB é o nó estação base que recebe da RASSF as decisões tomadas para registro e auditoria e envia comandos para os NSDs. Em todos os experimentos, a RASSF tinha apenas um NEB. Assim, se houver uma falha de parada no nó NEB, as decisões tomadas em seu período de inatividade não serão registradas e não será possível enviar comandos para dentro da RASSF, mas não haverá impacto sobre o atraso ou sobre o funcionamento do sistema.

Tabela 5. Efeito de falhas sobre o NEB

Falhas	Atraso	Funcionamento	Efeito
1 NEB	—	—	Não registro de decisões tomadas e não é possível enviar comandos

Os NSDs são os nós que tomam as decisões de acordo com os dados sensorizados enviados pelos NSOs associados a cada NSD e enviam as decisões tomadas para os NATs associados, para sua execução, e para o NEB, para registro e auditoria. Falhas de parada em nós NSD tornam o CONDE inoperante em toda a área de atuação do nó em questão, se não

houver nenhuma espécie de redundância sobre a área de atuação do NSD que falhou, ou seja, os dados sensorizados pelos NSOs não são utilizados e os NATs não recebem decisões para execução. Outra implicação de falhas em um NSD é a interrupção do repasse de decisões tomadas por outros NSDs pela RASSF. Se um NSD pertencente a uma rota de repasse de decisões para o NEB falha, esta rota é interrompida e deve ser recalculada. Se houver outra rota possível, a rota é recalculada e o repasse das decisões continua normalmente. Caso contrário, o repasse de decisões dos NSDs que dependiam do NSD que falhou fica interrompido. Falhas nos NSDs não causam impacto sobre o atraso do sistema, mas causam impacto sobre o funcionamento do sistema.

Tabela 6. Efeito de falhas sobre os NSDs

Falhas	Atraso	Funcionamento	Efeito
1 NSD	—	Interrupção local	Sistema inoperante na área de atuação do NSD que falhou
2 NSDs / rotas não afetadas	—	Interrupção local	Sistema inoperante na área de atuação dos NSDs que falharam
2 NSDs / rotas afetadas	—	Interrupção local	Sistema inoperante na área de atuação dos NSDs que falharam e interrupção do registro dos NSDs cujas rotas dependem dos NSD que falharam
3 NSDs / rotas não afetadas	—	Interrupção local	Sistema inoperante na área de atuação dos NSDs que falharam
3 NSDs / rotas afetadas	—	Interrupção local	Sistema inoperante na área de atuação dos NSDs que falharam e interrupção do registro dos NSDs cujas rotas dependem dos NSD que falharam

Os NSOs são os nós que fazem o sensoriamento das grandezas físicas no edifício inteligente e enviam estes dados sensorizados aos NSD para que estes possam tomar decisões. Quando um NSO sofre uma falha de parada, ele pára de enviar os dados de sensoriamento para seu NSD associado. Porém, como há 4 NSOs associados a cada NSD, conforme os resultados obtidos no primeiro experimento, o CONDE continua funcionando normalmente com 3 NSOs associados, sem no entanto operar no estado de atraso mínimo. Este atraso acontece somente localmente na área regida pelo NSD cujo NOS associado falhou e não reflete um atraso em todo o sistema. No pior caso, onde todos os NSOs associados a um NSD falham, o NSD não recebe nenhum dado de sensoriamento, então, apesar de não ter nenhuma falha, não pode tomar decisões. Neste caso, o NSD fica num estado inativo esperando receber leituras suficientes para encher seu buffer de leitura e finalmente poder tomar as decisões.

Tabela 7. Efeito de falhas sobre os NSOs

Falhas	Atraso	Funcionamento	Efeito
1 NSO	+ 94,597 %	—	Este NSO não envia dados de sensoriamento
2 NSOs / NSDs distintos	+ 94,597 %	—	Estes NSOs não enviam dados de sensoriamento
2 NSOs / NSD único	+ 94,597 %	—	Estes NSOs não enviam dados de sensoriamento
3 NSOs / NSDs distintos	+ 94,597 %	—	Estes NSOs não enviam dados de sensoriamento
3 NSOs / NSD único	+ 283,868 %	—	Estes NSOs não enviam dados de sensoriamento
4 NSOs / NSDs distintos	+ 100 %	—	Estes NSOs não enviam dados de sensoriamento
4 NSOs / NSD único	—	Interrupção local	O NSD não recebe dados de sensoriamento, logo não pode tomar decisões

É importante frisar que em nenhuma das possíveis hipóteses de falha descritas neste experimento houve interrupção total do CONDE. Em todos os casos a interrupção foi parcial e restrita a uma região específica. Um cenário de interrupção total do sistema só aconteceria pela falha de todos os NSDs ou de todos os NSOs. Portanto, o CONDE tem melhor resistência a falhas de parada que outras propostas de sistemas centralizados, onde uma falha de parada na entidade centralizadora, onde se localiza o sistema de controle e decisão, provocaria a interrupção do funcionamento de todo o sistema.

5.4.5 Comparação entre o CONDE e outros sistemas

Neste experimento, o sistema CONDE completo, ou seja, capaz de executar todas as suas funcionalidades conforme especificadas no presente trabalho, foi comparado com duas outras soluções: (i) o CONDE incapaz de efetuar a integração de aplicação pelo encadeamento de decisões e (ii) uma abordagem centralizada (na qual não são utilizados atuadores, ou seja, trata-se de uma RSSF) apresentada em (TSAI et al. 2008). Tal comparação permite analisar principalmente as vantagens de uma abordagem descentralizada, onde se explora a capacidade de processamento dentro da RASSF, e da estratégia de integrar aplicações no nível da decisão. Tais vantagens serão avaliadas em termos do consumo de energia (tanto da RASSF quanto do edifício) e da eficiência das duas abordagens, que é representada pela quantidade de decisões corretas. Na abordagem centralizada, os sensores enviam os dados coletados para a estação base e esta por sua vez transmite os dados para fora da rede, determina a ação a ser tomada e depois envia a resposta aos nós atuadores, ou seja, não existem nós decisores no sistema.

O cenário usado em (TSAI et al. 2008) consiste de uma casa com 2 ambientes, um monitorado pelo sistema de decisão e outro utilizando equipamentos de refrigeração tradicionais (não dotados de capacidades “inteligentes”). O sistema proposto em (TSAI et al. 2008) é um sistema de controle de temperatura baseado em *smart-skins*. *Smart-skins* são elementos auto-reguláveis da interface externa do ambiente, que permitem avaliar o ambiente externo ao edifício inteligente e se adaptar a mudanças de temperatura. Os *smart-skins* incluem RSSF para fazer o monitoramento do ambiente. Diferentemente de nossa abordagem, a decisão de mudança de estados nos *smart-skins* não é realizada internamente a própria rede.

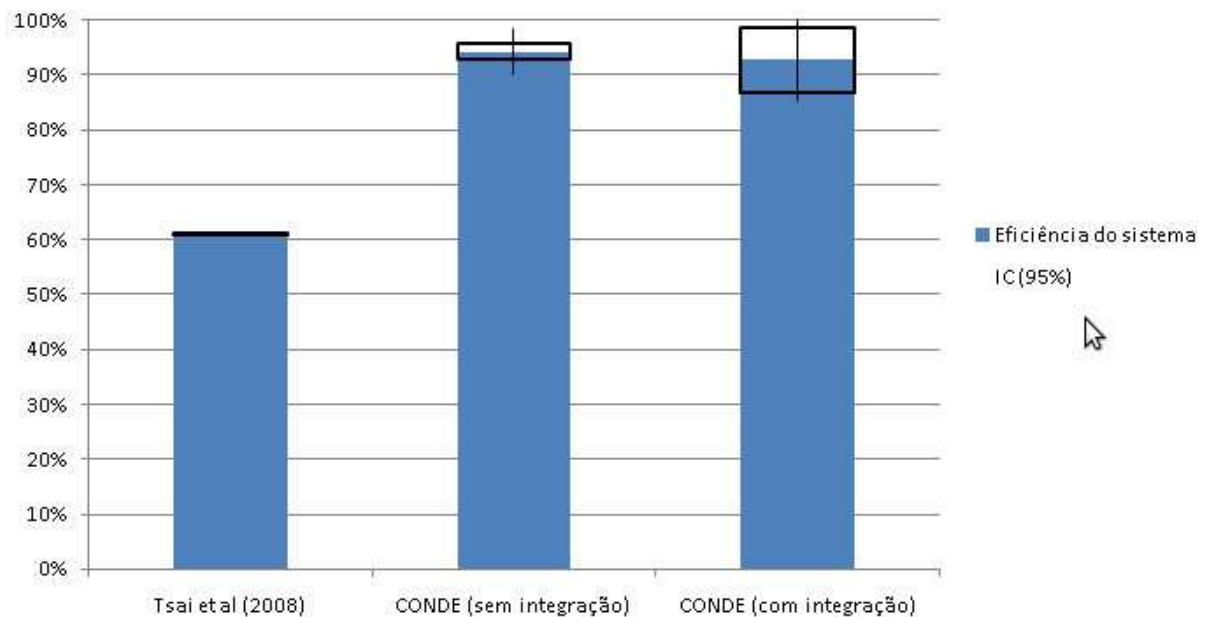


Figura 14. Comparação em termos de eficiência do sistema entre as soluções

No cenário proposto, as salas foram monitoradas de 10 em 10 minutos durante 3 horas. Os resultados de eficiência do sistema são apresentados na Figura 14. Pode-se observar na Tabela 8 que ambas as soluções descentralizadas tiveram resultados melhores que a solução centralizada. As soluções com e sem integração do CONDE foram respectivamente 31,643% e 32,222% mais eficientes que o sistema proposto em (TSAI et al. 2008). Pode-se observar também que, em média, o CONDE sem integração foi ligeiramente melhor que o CONDE com integração em termos de eficiência do sistema. Isto pode ser explicado por uma inconsistência entre as decisões tomadas no **Subsistema Inferência** e as desencadeadas no **Subsistema Integração** numa mesma aplicação. Se estes dois tipos de decisão forem ligeiramente diferentes em uma mesma aplicação, as duas decisões, embora redundantes, podem ser executadas no ambiente monitorado, causando esta diferença na eficiência do sistema.

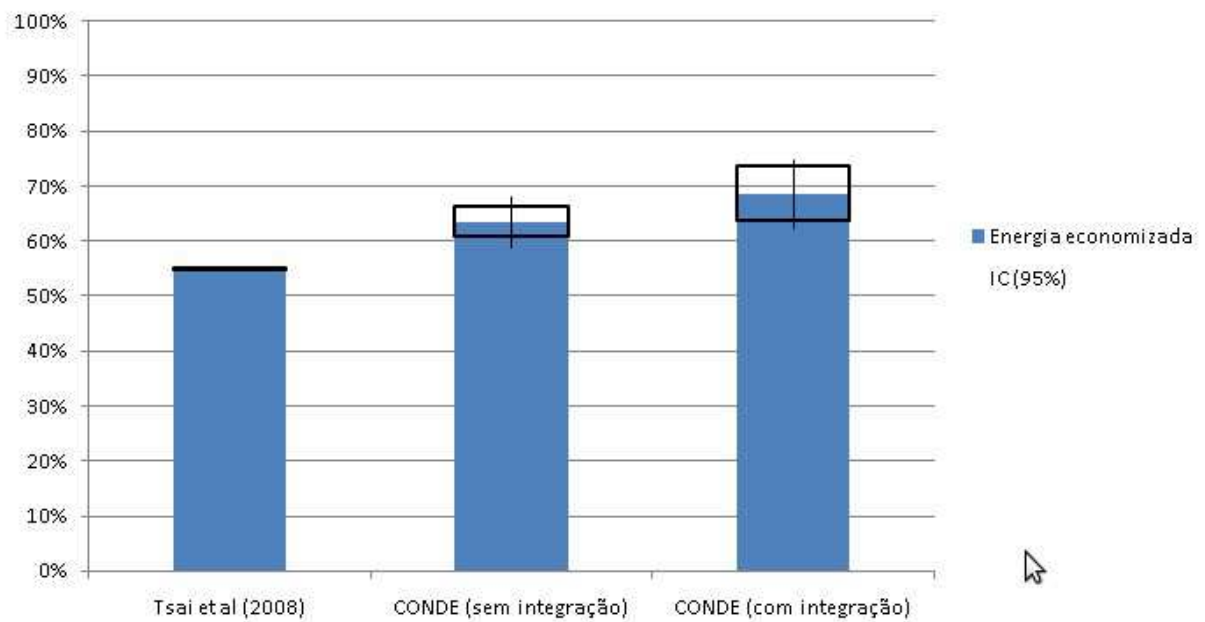


Figura 15. Comparações em termos de economia de energia entre as aplicações

Os resultados de economia de energia são apresentados na Figura 15. Em relação ao consumo de energia no edifício inteligente, ambas as soluções descentralizadas, CONDE (sem integração) e CONDE (com integração) apresentaram melhores resultados que a solução centralizada, isto é, uma economia de 8,529% e 13,624%, respectivamente, em comparação com (TSAI et al. 2008), como pode ser observado na Tabela 8. Por ser descentralizado, o tempo de resposta dos nós decisores do CONDE é muito menor que em um sistema de decisão centralizado, uma vez que a decisão é transmitida em apenas um salto, enquanto em (TSAI et al. 2008) os dados coletados e a decisão tomada percorrem vários saltos saindo e entrando na RSSF, respectivamente.

Tabela 8. Comparação entre os experimentos simulado e real

Experimentos	CONDE (sem Integração)			CONDE (com Integração)			(TSAI et al. 2008)
	Média	DP	IC	Média	DP	IC	Média
Eficiência do sistema	94,333%	4,026	±1,441	92,754%	7,398	±5,864	61,111%
Energia economizada no edifício	63,529%	4,652	±2,753	68,624%	6,183	±5,014	55%

6 Conclusões

Neste trabalho foi apresentado o CONDE, um sistema de decisão descentralizado e colaborativo para aplicações de RASSF utilizadas em edifícios inteligentes a fim de melhorar a eficiência energética do edifício e, conseqüentemente, contribuir para a redução da emissão de gases poluentes pelo próprio edifício. O CONDE foi implementado para ser associado a aplicações da terceira geração dos edifícios inteligentes e para ser implantado em RASSF, de forma a descentralizar o processo de decisão das aplicações. O sistema apresentado é dito colaborativo, pois vários nós sensores atuam juntos, comunicando-se entre si, a fim de produzir uma decisão que melhore a eficiência energética do edifício inteligente, sem provocar desconforto ao usuário presente no edifício ou esgotar os recursos limitados da RASSF em que o sistema foi implantado.

Para melhorar a eficiência energética do edifício inteligente, o CONDE em sua versão atual utiliza um algoritmo simples baseado em regras pré-definidas que ficam armazenadas na memória *flash* de armazenamento dos nós sensores que executam o papel Nó Sensor Decisor. Estes nós recebem como entrada os dados de monitoramento enviados pelos nós que executam o papel Nó Sensor Ordinário para definir possíveis ações a serem realizadas no ambiente de implantação da RASSF. Uma vez que as decisões são tomadas, elas são enviadas aos nós que executam o papel Nó Atuador para execução sobre o ambiente monitorado. Este algoritmo baseado em regras foi escolhido por sua simplicidade, fator fundamental para projetos de RASSF, e a facilidade de definição das regras de inferência.

Foram realizados diversos experimentos a fim de comprovar que: (i) os resultados obtidos através de simulações de RASSF eram confiáveis e poderiam ser utilizados (eventualmente extrapolados para outros cenários), (ii) implementar um sistema de controle e decisão para aplicações implantadas em edifícios inteligentes baseado em uma RASSF era factível, (iii) descentralizar o sistema entre os nós que compunham a RASSF produzia um ganho em termos de tolerância a falhas sobre as soluções tradicionais, e (iv) esta descentralização também contribuía em termos de economia total de energia gasta pelo edifício como um todo. Por meio dos testes efetuados, foi comprovada a viabilidade de implantar o CONDE da forma como foi idealizado na RASSF e que descentralizar todo o sistema de controle e decisão entre os nós sensores da RASSF contribuiu positivamente em termos de tolerância a falhas e economia global de energia no edifício.

Nos experimentos realizados para comprovar os resultados obtidos pelo simulador poderiam ser utilizados, foram implementados dois cenários semelhantes, sendo que um deles

utilizava plataformas de sensores reais compostas por motes MICAz e o outro era simulado através do simulador Avrora. Neste experimento foram comparadas métricas de atraso de transmissão, energia consumida e eficiência do CONDE na tomada de decisões, a fim de aferir se os resultados obtidos nas simulações seriam similares aos resultados obtidos em experimentos conduzidos em plataformas reais de sensoriamento. A avaliação destas métricas mostrou que os resultados obtidos nas simulações eram muito similares aos obtidos nos experimentos reais e, portanto, poderiam ser utilizados como resultados válidos.

Nos experimentos que visaram comprovar que a implementação do CONDE em uma RASSF era factível, foram avaliadas métricas relevantes a projetos de RASSFs, a saber: (i) impacto de processamento causado, (ii) consumo de memórias RAM, *flash* de programa e *flash* de armazenamento e (iii) consumo de energia pelos nós da RASSF. A avaliação destas métricas mostrou que a implementação do CONDE em uma RASSF era, de fato, possível. Uma vez comprovado este fato, foi possível realizar os demais testes para a medição da eficácia e eficiência do CONDE.

Nos experimentos realizados para verificar o comportamento do CONDE frente a falhas de parada nos nós sensores, foi verificado o impacto causado no sistema pelo desligamento de nós, simulando falhas de *hardware* ou esgotamento de energia. As métricas utilizadas foram o impacto sobre o funcionamento do sistema como um todo, ou seja, se a falha causaria interrupção na operação do sistema, e efeitos que estas falhas causariam, ainda que o serviço provido não fosse totalmente interrompido. A avaliação dos diversos cenários de falha testados mostrou que o CONDE é capaz de resistir a várias falhas de parada em nós cumprindo os diversos papéis definidos no trabalho sem comprometer o funcionamento do sistema.

Finalmente, no experimento que visou comprovar que o CONDE pode contribuir para a eficiência energética global do sistema, o CONDE foi comparado com a solução apresentada em (TSAI et al. 2008) em termos de energia consumida pelo edifício e eficiência do sistema ao tomar decisões. Além disto, uma versão anterior do CONDE, a qual não inclui o Subsistema Integração, foi também comparada com estas duas soluções, a fim de medir também que impacto em termos de consumo de energia e eficiência este componente causa no sistema. A avaliação destas métricas comprovou que o CONDE obteve melhores resultados tanto em termos de economia de energia quanto em eficiência do sistema quando comparado à solução apresentada em (TSAI et al. 2008). Considerando a implementação atual do CONDE e do Subsistema Integração, o CONDE utilizando o Subsistema Integração tem

melhores resultados em termos de economia de energia, mas tem resultados ligeiramente piores em termos de eficiência do sistema quando comparado ao próprio CONDE não utilizando o **Subsistema Integração**. Esta diferença na métrica eficiência do sistema pode ser atribuída ao algoritmo escolhido para o **Subsistema Integração**, que pode não ser o ideal para realizar esta tarefa, ou às regras de integração que, poderiam não estar perfeitamente configuradas, gerando este tipo de anomalia.

6.1 Trabalhos Futuros

Quanto aos trabalhos futuros a serem realizados com o CONDE, uma das vertentes de investigação consiste em se analisar a adoção de métodos de agregação de dados de sensoriamento obtidos no edifício inteligente. Isto pode trazer ganhos quanto a energia gasta pelos nós sensores que compõem a RASSF utilizada, além de potencialmente acelerar ainda mais a resposta do sistema aos estímulos ambientais providos pelo edifício.

Outra vertente de trabalhos futuros a serem investigados consiste na utilização de outros métodos de inferência e integração. O CONDE foi implementado de forma modular, a fim de fornecer interfaces de programação que permitissem a substituição destes métodos de forma simples, pela substituição dos componentes TinyOS em que eles foram implementados. Durante a elaboração deste trabalho, foi cogitada a possibilidade de utilizar um sistema de inferência nebuloso como sistema de inferência do CONDE, pois este método teria vantagem pela diminuição do número de regras armazenadas na memória *flash* de armazenamento. Entretanto devido a limitações de tempo não foi possível testar este método. Teoricamente, um sistema de inferência nebuloso utilizaria um algoritmo mais complexo para tratamento das regras, mas em contrapartida utilizaria muito menos regras do que o método de inferência utilizado no CONDE atualmente. Este ganho em memória de armazenamento permitiria que fossem armazenadas mais aplicações no mesmo NSD, trazendo assim mais funcionalidades ao edifício inteligente.

Outra possibilidade ainda é a implementação de métodos de colaboração entre NSDs, onde NSDs em uma mesma sala ou ambiente pudessem trocar decisões tomadas em suas áreas de influência a fim de se obterem resultados mais corretos e precisos, levando em conta decisões tomadas com dados de sensoriamento colhidos em locais fora do alcance destes NSDs. Talvez um sistema de votação entre os NSDs fornecesse uma visão mais ampla sobre um ambiente que fosse grande o suficiente para utilizar mais de um NSD.

Outro aspecto não investigado por este trabalho é a questão da segurança do sistema. O CONDE não inclui nenhum mecanismo de segurança em sua implementação atual, de forma que o sistema é vulnerável a uma série de ataques. Podemos citar, como por exemplo: (i) ataques de negação de serviço sobre os NSDs, onde um dispositivo *jammer* bloquearia a comunicação dos NSDs, provocando a indisponibilidade do sistema como um todo, (ii) ataques de personificação tanto sobre os NSDs quanto sobre NATs, onde um atacante poderia injetar pacotes contendo dados de sensoriamento ou decisões arbitrárias, forçando o sistema a processá-los como legítimos por não haverem métodos de validação de integridade presentes no sistema, (iii) ataques sobre as tabelas de roteamento dos NSDs, onde um atacante forçaria a criação de *loops* nas tabelas de roteamento dos NSDs, de forma que os dados enviados nunca chegassem ao NEB, ou ainda (iv) ataques de injeção, onde um atacante injetaria nós sob seu controle que poderiam executar qualquer dos papéis descritos neste trabalho e colher dados para enviá-los ao atacante posteriormente, entre muito outros. Assim, há diversas possibilidades de trabalhos futuros na área de segurança.

Referências

- AGARWAL, Y. et al. Occupancy-driven energy management for smart building automation. In: ACM WORKSHOP ON EMBEDDED SENSING SYSTEMS FOR ENERGY-EFFICIENCY IN BUILDING, 2., 2010. Zürich. **Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building**, New York: ACM, 2010. p. 1–6.
- ALBEROLA, R. P. ; PESCH, D. AvroraZ: extending Avrora with an IEEE 802.15.4 compliant radio chip model. In: ACM WORKSHOP ON PERFORMANCE MONITORING AND MEASUREMENT OF HETEROGENEOUS WIRELESS AND WIRED NETWORKS, 3., 2008. Vancouver. **Proceedings of the 3rd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks**. New York: ACM, 2008.
- AKYILDIZ, I. F. ; KASIMOGLU, I. H. Wireless sensor and actor networks: research challenges. **Ad Hoc Networks**, [S.l.] v. 2, n. 4, p. 351–367. Oct. 2004.
- AKYILDIZ, I. F. ; VURAN, M. C. **Wireless sensor networks**. Chichester: John Wiley & Sons, 2010. (Ian F. Akyildiz Series in Communication and Networking).
- BUCKL, C. et al. Services to the field: an approach for resource constrained sensor/actor networks. In: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS, 2009, Bradford, UK. **Proceedings of the International Conference on Advanced Information Networking and Applications Workshops**, Washington: IEEE, 2009. p. 476–481.
- BUYS, L. et al. Smart housing and social sustainability: learning from the residents of queensland's research house. **Australian Journal of Emerging Technologies and Society**, Hawthorn, v. 3, n. 1, p. 43–57, 2005.
- BYUN, J. ; PARK, S. Development of a self-adapting intelligent system for building energy saving and context-aware smart services, **IEEE Transactions on Consumer Electronics**, v. 57, n. 1, p. 90–98. Feb. 2011.
- CHRUNGOO, A. et al. On adaptive HVAC based on a de-centralized algorithm using K:1 transmission protocol for autonomous wireless sensor network. In: WCSN - INTERNATIONAL CONFERENCE ON WIRELESS COMMUNICATION AND SENSOR NETWORKS, 6., 2011. Allahabad, India. **Proceedings of the Sixth International Conference on Wireless Communication and Sensor Networks**, Allahabad, India: Indian Institute of Information Technology, 2011. p. 1–6.
- DEGW London Limited ; TEKNIBANK (Milan, Italy) ; EUROPEAN INTELLIGENT BUILDING GROUP. **The intelligent building in europe: executive summary**. London: British Council for Offices, 1992. (Technical Report, 3).
- DELANEY, D. T. ; O'HARE, G. M. P. ; RUZZELLI, A. G. Evaluation of energy-efficiency in lighting systems using sensor networks. In: ACM WORKSHOP ON EMBEDDED SENSING SYSTEMS FOR ENERGY-EFFICIENCY IN BUILDINGS, 1., 2009, Berkeley, CA. **Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings**, New York: ACM, 2009. p. 61–66.
- DELICATO, F. C. **Middleware orientado a serviços para redes de sensores sem fio**. 2005. Tese (Doutorado em Engenharia Elétrica) - Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia (COPPE), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

- DIETRICH, I. ; DRESSLER, F. On the lifetime of wireless sensor networks. **ACM Transactions on Sensor Networks**, New York, v. 5, n. 1, p. 1–39. Feb. 2009.
- EFSTRATIOU, C. et al. A shared sensor network infrastructure. In: **SenSys '10 - ACM CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS**, 8., 2010, Zürich. **Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems**, New York: ACM, 2010. p. 367–368.
- ERICKSON, V. L. et al. Energy efficient building environment control strategies using real-time occupancy measurements. In: **ACM WORKSHOP ON EMBEDDED SENSING SYSTEMS FOR ENERGY-EFFICIENCY IN BUILDINGS**, 1., 2009, Berkeley, CA. **Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings**, New York: ACM, 2009. p. 19–24.
- FILIPPONI, L. et al. Smart city: an event driven architecture for monitoring public spaces with heterogeneous sensors. In: **INTERNATIONAL CONFERENCE ON SENSOR TECHNOLOGIES AND APPLICATIONS**, 4., 2010, Venice. **Proceedings of the International Conference on Sensor Technologies and Applications**, Piscataway, NJ: IEEE, 2010. p. 281–286.
- FORTINO, G. ; GUERRIERI, A. Decentralized and embedded management for smart buildings. In: **WORKSHOP ON APPLICATIONS OF SOFTWARE AGENTS**, 2011, Novi Sad, Serbia. **Proceedings of the Workshop on Applications of Software Agents**, Sad, Serbia: CEUR, 2011. p. 3–7.
- GEZER, C. ; BURATTI, C. A ZigBee smart energy implementation for energy efficient buildings. In: **VEHICULAR TECHNOLOGY CONFERENCE**, 73., 2011, Budapest. **Proceedings of the IEEE 73rd Vehicular Technology Conference**, Piscataway, NJ: IEEE, 2011. p. 1–5.
- GeSI (Global e-Sustainability Initiative). **SMART 2020: enabling the low carbon economy in the information age**. [S.l.]: 2008. (United States Report Addendum). Technical Report The Climate Group.
- GNAWALI, O. et al. Collection tree protocol. In: **SenSys '09 - ACM CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS**, 7., 2009. Berkeley, CA. **Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems**, New York: ACM, 2009. páginas 1–14.
- GUNGOR, V.C. ; LU, B. ; HANCKE, G. P. Opportunities and challenges of wireless sensor networks in smart grid. **IEEE Transactions on Industrial Electronics**, New York, v. 57, n. 10, p. 3557–3564, Oct. 2010.
- HAYES, J. et al. Evaluation of a low cost wireless chemical sensor network for environmental monitoring. In: **IEEE CONFERENCE ON SENSORS**, 7., 2008, Lecce, Italy. **Proceedings on the IEEE Conference on Sensors**, [S.l.]: IEEE, 2008. p. 530–533.2008..
- HIROYASU, T. et al. Intelligent lighting control user interface through design of illuminance distribution. In: **INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS DESIGN AND APPLICATIONS**, 9., 2009. Pisa. **Proceedings of the International Conference on Intelligent Systems Design and Applications**, Piscataway, NJ: IEEE, 2009. p. 714–719.
- HYNES, G. ; REYNOLDS, V. ; HAUSWIRTH, M. Enabling mobility between context-aware smart spaces. In: 2009. **INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION NETWORKING AND APPLICATIONS WORKSHOPS**, 2009,

- Bradford, UK. **Proceedings of the International Conference on Advanced Information Networking and Applications Workshops**, Washington: IEEE, 2009. p. 255–260.
- KAWSAR, F. ; LYARDET, F. ; NAKAJIMA, T. Three challenges for future smart object systems. In: AML-Blocks '08: EUROPEAN WORKSHOP ON SMART PRODUCTS: BUILDING BLOCKS OF AMBIENT INTELLIGENCE, 2., 2008. Nürnberg. **Proceedings of the 2nd European Workshop on Smart Products: Building Blocks of Ambient Intelligence**, Norderstedt, Germany: Books on Demand Norderstedt, 2008, p. 35–37.
- KRÄMER, M. ; GERALDY, A. **Energy measurements for MicaZ node**. Stuttgart: Universität Stuttgart. 2006. p. 61–68. (Technical Report 5.GI/ITG KuVS.Fachgespräch DahtloseSensornetze).
- KLUES, K. et al. Integrating concurrency control and energy management in device drivers. **ACM SIGOPS Operating Systems Review**, New York, v. 41, n. 6, p. 251–264. Dec. 2007.
- LEMOS, E. T. et al. Algoritmo distribuído para detecção de dano em aero geradores utilizando redes de atuadores e sensores sem fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 29., 2011. Campo Grande. **Anais do XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, Porto Alegre: SBC, 2011. p. 439–452.
- LEVIS, P. ; GAY, D. **TinyOS programming**. Cambridge: Cambridge University Press, 2009. ISBN: 978-0-52-189606-1
- LEWIS, F. L. **Wireless sensor networks, smart environments: technologies, protocols, and applications**, Hoboken, NJ: John Wiley & Sons, 2005. (Wiley Series on Parallel and Distributed Computing). ISBN: 978-0-47-154448-7.
- LIN, K. ; LEVIS, P. Data discovery and dissemination with DIP, In: IPSN '08: INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING IN SENSOR NETWORKS, 7., 2008, St. Louis, Missouri. **Proceedings of the 7th International Conference on Information Processing in Sensor Networks**, New York: ACM/IEEE, 2008. p. 433–444.
- LIU, M. et al. SmartCIS: integrating digital and physical environments. **ACM SIGMOD Record**, New York; v. 39, n. 1, p. 48–53. Mar. 2010.
- MAESTRE, J. M. ; CAMACHO E. F. Smart home interoperability: the DomoEsi project approach. **International Journal of Smart Home**, Tasmania, v. 3, n. 3, p. 31–44, 2009.
- MEMSIC. MicaZ wireless measurement system datasheet. 2012. Disponível em: <http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html>. Acesso em: jun. 2012.
- MOHAMADDOUST, R. et al. Designing the lighting control system based on WSN with optimization of decision making algorithm. In: CICON 2010 - INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND COMMUNICATION NETWORKS, 2010. Bhopal, India. **Proceedings of the International Conference on Computational Intelligence and Communication Networks**, Los Alamitos, CA: IEEE, 2010. p. 279–283.
- NUNES, R. Integração de serviços para edifícios inteligentes. 1995. Tese (Doutorado em Engenharia Electrotécnica e de Computadores) - Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, 1995.

- PAN, M. S. et al. A WSN-based intelligent light control system considering user activities and profiles. **IEEE Sensors Journal**, New York, v. 8, n. 10, p. 1710–1721. Oct. 2008.
- PARK, H. ; BURKE, J. ; SRIVASTAVA, M. B. Design and implementation of a wireless sensor network for intelligent light control. In: IPSN '07 - INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING IN SENSOR NETWORKS, 6., 2007, Cambridge, Mass. **Proceedings of the 6th International Conference on Information Processing in Sensor Networks**, New York: ACM, 2007. p. 370–379.
- RUIZ, L. B. et al. Arquiteturas para redes de sensores sem fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 22., 2004. Gramado. **Livro Texto de Mini-Cursos do XXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, Porto Alegre: SBC, 2004. p. 167–218.
- SCHOR, L. ; SOMMER, P. ; WATTENHOFER, R. Towards a zero-configuration wireless sensor network architecture for smart buildings. In: ACM WORKSHOP ON EMBEDDED SENSING SYSTEMS FOR ENERGY-EFFICIENCY IN BUILDINGS, 1., 2009, Berkeley, CA. **Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings**, New York: ACM, 2009. .p. 31–36.
- SHU, F. ; HALGAMUGE, M. N. ; CHEN, W. Building automation systems using wireless sensor networks: radio characteristics and energy efficient communication protocols. **Electronic Journal of Structural Engineering**, Special Issue: Sensor Network on Building Monitoring: from Theory to Real Application, 2009. p. 66–73.
- SIFUENTES, E. ; CASAS, O. ; PALLAS-ARENY, R. Wireless magnetic sensor node for vehicle detection with optical wake-up. **IEEE Sensors Journal**, New York, v. 11, n. 8, p. 1669–1676. Aug. 2011.
- SINGH, R. ; MISHRA, S. ; JOSHI, P. Pressure monitoring in wireless sensor network using Zigbee transceiver module. In: INTERNATIONAL CONFERENCE ON COMPUTER AND COMMUNICATION TECHNOLOGY, 2 , 2011, Allahabad, India. **Proceedings of the 2nd International Conference on Computer and Communication Technology**, Allahabad, India: IEEE, 2011. p. 225–229.
- SINGHVI, V. et al. Intelligent light control using sensor network. In: SenSys '05 - INTERNATIONAL CONFERENCE ON EMBEDDED NETWORKED SENSOR SYSTEMS, 3. 2005, San Diego, CA. **Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems**, New York: ACM, 2005. p. 218–229.
- SUN, Y. ; ZHAO, G. ; LUO, H. Smart building control based on wireless sensor-actuator networks. **Chinese Journal of Electronics**, [S.l.]: v. 20, n. 3, p. 437-442. Jul. 2011.
- TACHWALI, Y. ; REFAI, H. ; FAGAN, J. E. Minimizing HVAC energy consumption using a wireless sensor network. In: IECON 2007 - ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY, 33., 2007. Taipei, Taiwan. **Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society**, 2007. Piscataway NJ: IEEE, 2007. v. 1. Piscataway, p. 439–444.
- TANENBAUM, A. S. ; VAN STEEN, M. Distributed systems: principles and paradigms. Upper Saddle River: Pearson Prentice Hall, 2007. ISBN: 978-0-13-239227-3.
- TSAI, K. C. ; SUNG, J. T. ; JIN, M. H. Wireless sensor network and sensor fusion technology for ubiquitous smart living space applications. In: ISUC '08 - INTERNATIONAL SYMPOSIUM ON UNIVERSAL COMMUNICATION, 2., 2008. Osaka Japan.

Proceedings of the 2008 Second International Symposium on Universal Communication, Los Alamitos CA: IEEE, 2008. p. 295–302.

- WEBER, V. Smart sensor networks: technologies and applications for green growth. Washington DC: OECD Directorate for Science, Technology and Industry. 2009. (Technical Report).
- WEN, Y. J. ; GRANDERSON, J. ; AGOGINO, A. M. Towards embedded wireless-networked intelligent daylighting systems for commercial buildings. In: SUTC 06 - INTERNATIONAL CONFERENCE ON SENSOR NETWORKS, UBIQUITOUS, AND TRUSTWORTHY COMPUTING, 2006. Taichung, Taiwan. **Proceedings of the International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing**, Los Alamitos CA: IEEE, 2006. v. 1, p. 326–331.
- WONG, J. K. W. ; LI, H. ; WANG, S. W. Intelligent building research: a review. **Automation in Construction**, Amsterdam, v. 14, n. 1, p. 143–159. Jan. 2005.
- YEH, L. W. et al. Autonomous light control by wireless sensor and actuator networks. **IEEE Sensor Journal**, New York, v. 10, n. 6, p. 1029–1041. Jun. 2010.
- YICK, J. ; MUKHERJEE, B. ; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, Amsterdam, v. 52, n. 12, p. 2292-2330. Aug. 2008.

ANEXO 1 – Base de dados Aplicações

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

<applications xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="applications.xsd">
  <application>
    <applicationId>TAG_AVAC_APPLICATION</applicationId>
    <timerScale>600</timerScale>
    <sensorId>TAG_TEMPERATURE_SENSOR</sensorId>
    <sensorId>TAG_PRESENCE_SENSOR</sensorId>
  </application>
  <application>
    <applicationId>TAG_LUMINANCE_APPLICATION</applicationId>
    <timerScale>150</timerScale>
    <sensorId>TAG_PHOTO_SENSOR</sensorId>
    <sensorId>TAG_PRESENCE_SENSOR</sensorId>
  </application>
</applications>
```

ANEXO 2 – Base de dados Parâmetros de Inferência

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

<inference xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ruled_inference.xsd">
  <application>
    <applicationId>TAG_AVAC_APPLICATION</applicationId>
    <rule>
      <input>0.0 6.25</input>
      <input>0.0 12.5</input>
      <output>0.0 0.0</output>
    </rule>
    <rule>
      <input>0.0 6.25</input>
      <input>12.50 25.0</input>
      <output>1.0 25.0</output>
    </rule>
    <rule>
      <input>0.0 6.25</input>
      <input>25.0 37.5</input>
      <output>1.0 25.0</output>
    </rule>
    <rule>
      <input>0.0 6.25</input>
      <input>37.50 50.0</input>
      <output>3.0 50.0</output>
    </rule>
    <rule>
      <input>0.0 6.25</input>
      <input>50.0 62.5</input>
      <output>3.0 50.0</output>
    </rule>
    <rule>
      <input>0.0 6.25</input>
      <input>62.50 75.0</input>
      <output>5.0 75.0</output>
    </rule>
    <rule>
      <input>0.0 6.25</input>
```

```
        <input>75.0 87.5</input>
        <output>5.0 75.0</output>
</rule>
<rule>
    <input>0.0 6.25</input>
    <input>87.50 100.0</input>
    <output>7.0 100.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
    <input>0.0 12.5</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
    <input>12.50 25.0</input>
    <output>0.0 25.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
    <input>25.0 37.5</input>
    <output>1.0 25.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
    <input>37.50 50.0</input>
    <output>1.0 25.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
    <input>50.0 62.5</input>
    <output>3.0 50.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
    <input>62.50 75.0</input>
    <output>3.0 50.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
```

```
        <input>75.0 87.5</input>
        <output>5.0 50.0</output>
</rule>
<rule>
    <input>6.25 12.5</input>
    <input>87.50 100.0</input>
    <output>5.0 75.0</output>
</rule>
<rule>
    <input>12.50 18.75</input>
    <input>0.0 12.5</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>12.50 18.75</input>
    <input>12.50 25.0</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>12.50 18.75</input>
    <input>25.0 37.5</input>
    <output>0.0 25.0</output>
</rule>
<rule>
    <input>12.50 18.75</input>
    <input>37.50 50.0</input>
    <output>1.0 25.0</output>
</rule>
<rule>
    <input>12.50 18.75</input>
    <input>50.0 62.5</input>
    <output>1.0 25.0</output>
</rule>
<rule>
    <input>12.50 18.75</input>
    <input>62.50 75.0</input>
    <output>3.0 25.0</output>
</rule>
<rule>
    <input>12.50 18.75</input>
```



```
<input>75.0 87.5</input>
<output>3.0 50.0</output>
</rule>
<rule>
  <input>12.50 18.75</input>
  <input>87.50 100.0</input>
  <output>3.0 50.0</output>
</rule>
<rule>
  <input>18.75 25.0</input>
  <input>0.0 12.5</input>
  <output>0.0 0.0</output>
</rule>
<rule>
  <input>18.75 25.0</input>
  <input>12.50 25.0</input>
  <output>0.0 0.0</output>
</rule>
<rule>
  <input>18.75 25.0</input>
  <input>25.0 37.5</input>
  <output>0.0 0.0</output>
</rule>
<rule>
  <input>18.75 25.0</input>
  <input>37.50 50.0</input>
  <output>0.0 0.0</output>
</rule>
<rule>
  <input>18.75 25.0</input>
  <input>50.0 62.5</input>
  <output>1.0 25.0</output>
</rule>
<rule>
  <input>18.75 25.0</input>
  <input>62.50 75.0</input>
  <output>1.0 25.0</output>
</rule>
<rule>
  <input>18.75 25.0</input>
```

```
        <input>75.0 87.5</input>
        <output>1.0 25.0</output>
</rule>
<rule>
    <input>18.75 25.0</input>
    <input>87.50 100.0</input>
    <output>1.0 25.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
    <input>0.0 12.5</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
    <input>12.50 25.0</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
    <input>25.0 37.5</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
    <input>37.50 50.0</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
    <input>50.0 62.5</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
    <input>62.50 75.0</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
```

```
        <input>75.0 87.5</input>
        <output>-1.0 25.0</output>
</rule>
<rule>
    <input>25.0 31.25</input>
    <input>87.50 100.0</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
    <input>0.0 12.5</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
    <input>12.50 25.0</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
    <input>25.0 37.5</input>
    <output>0.0 25.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
    <input>37.50 50.0</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
    <input>50.0 62.5</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
    <input>62.50 75.0</input>
    <output>-3.0 25.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
```

```
        <input>75.0 87.5</input>
        <output>-3.0 50.0</output>
</rule>
<rule>
    <input>31.25 37.5</input>
    <input>87.50 100.0</input>
    <output>-3.0 50.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
    <input>0.0 12.5</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
    <input>12.50 25.0</input>
    <output>0.0 25.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
    <input>25.0 37.5</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
    <input>37.50 50.0</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
    <input>50.0 62.5</input>
    <output>-3.0 50.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
    <input>62.50 75.0</input>
    <output>-3.0 50.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
```

```
        <input>75.0 87.5</input>
        <output>-5.0 50.0</output>
</rule>
<rule>
    <input>37.50 43.75</input>
    <input>87.50 100.0</input>
    <output>-5.0 75.0</output>
</rule>
<rule>
    <input>43.75 50.0</input>
    <input>0.0 12.5</input>
    <output>0.0 0.0</output>
</rule>
<rule>
    <input>43.75 50.0</input>
    <input>12.50 25.0</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>43.75 50.0</input>
    <input>25.0 37.5</input>
    <output>-1.0 25.0</output>
</rule>
<rule>
    <input>43.75 50.0</input>
    <input>37.50 50.0</input>
    <output>-3.0 50.0</output>
</rule>
<rule>
    <input>43.75 50.0</input>
    <input>50.0 62.5</input>
    <output>-3.0 50.0</output>
</rule>
<rule>
    <input>43.75 50.0</input>
    <input>62.50 75.0</input>
    <output>-5.0 75.0</output>
</rule>
<rule>
    <input>43.75 50.0</input>
```

```

        <input>75.0 87.5</input>
        <output>-5.0 75.0</output>
    </rule>
    <rule>
        <input>43.75 50.0</input>
        <input>87.50 100.0</input>
        <output>-7.0 100.0</output>
    </rule>
</application>
<application>
    <applicationId>TAG_LUMINANCE_APPLICATION</applicationId>
    <rule>
        <input>0.0 128.0</input>
        <input>0.0 12.5</input>
        <output>0.0</output>
    </rule>
    <rule>
        <input>0.0 128.0</input>
        <input>12.50 25.0</input>
        <output>16.0</output>
    </rule>
    <rule>
        <input>0.0 128.0</input>
        <input>25.0 37.5</input>
        <output>32.0</output>
    </rule>
    <rule>
        <input>0.0 128.0</input>
        <input>37.50 50.0</input>
        <output>64.0</output>
    </rule>
    <rule>
        <input>0.0 128.0</input>
        <input>50.0 62.5</input>
        <output>128.0</output>
    </rule>
    <rule>
        <input>0.0 128.0</input>
        <input>62.50 75.0</input>
        <output>128.0</output>
    </rule>

```

```
</rule>
<rule>
    <input>0.0 128.0</input>
    <input>75.0 87.5</input>
    <output>128.0</output>
</rule>
<rule>
    <input>0.0 128.0</input>
    <input>87.50 100.0</input>
    <output>128.0</output>
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>0.0 12.5</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>12.50 25.0</input>
    <output>16.0</output>
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>25.0 37.5</input>
    <output>32.0</output>
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>37.50 50.0</input>
    <output>64.0</output>
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>50.0 62.5</input>
    <output>64.0</output>
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>62.50 75.0</input>
    <output>64.0</output>
```

```
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>75.0 87.5</input>
    <output>128.0</output>
</rule>
<rule>
    <input>128.0 256.0</input>
    <input>87.50 100.0</input>
    <output>128.0</output>
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>0.0 12.5</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>12.50 25.0</input>
    <output>16.0</output>
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>25.0 37.5</input>
    <output>16.0</output>
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>37.50 50.0</input>
    <output>32.0</output>
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>50.0 62.5</input>
    <output>32.0</output>
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>62.50 75.0</input>
    <output>32.0</output>
```



```
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>75.0 87.5</input>
    <output>64.0</output>
</rule>
<rule>
    <input>256.0 384.0</input>
    <input>87.50 100.0</input>
    <output>64.0</output>
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>0.0 12.5</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>12.50 25.0</input>
    <output>-16.0</output>
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>25.0 37.5</input>
    <output>-16.0</output>
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>37.50 50.0</input>
    <output>16.0</output>
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>50.0 62.5</input>
    <output>16.0</output>
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>62.50 75.0</input>
    <output>16.0</output>
```

```
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>75.0 87.5</input>
    <output>16.0</output>
</rule>
<rule>
    <input>384.0 512.0</input>
    <input>87.50 100.0</input>
    <output>16.0</output>
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>0.0 12.5</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>12.50 25.0</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>25.0 37.5</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>37.50 50.0</input>
    <output>-16.0</output>
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>50.0 62.5</input>
    <output>-16.0</output>
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>62.50 75.0</input>
    <output>-16.0</output>
```

```
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>75.0 87.5</input>
    <output>-16.0</output>
</rule>
<rule>
    <input>512.0 640.0</input>
    <input>87.50 100.0</input>
    <output>-16.0</output>
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>0.0 12.5</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>12.50 25.0</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>25.0 37.5</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>37.50 50.0</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>50.0 62.5</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>62.50 75.0</input>
    <output>-32.0</output>
```

```
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>75.0 87.5</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>640.0 768.0</input>
    <input>87.50 100.0</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>0.0 12.5</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>12.50 25.0</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>25.0 37.5</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>37.50 50.0</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>50.0 62.5</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>62.50 75.0</input>
    <output>-64.0</output>
```

```
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>75.0 87.5</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>768.0 896.0</input>
    <input>87.50 100.0</input>
    <output>-32.0</output>
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>0.0 12.5</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>12.50 25.0</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>25.0 37.5</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>37.50 50.0</input>
    <output>-128.0</output>
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>50.0 62.5</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>62.50 75.0</input>
    <output>-64.0</output>
```

```
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>75.0 87.5</input>
    <output>-64.0</output>
</rule>
<rule>
    <input>896.0 1024.0</input>
    <input>87.50 100.0</input>
    <output>-64.0</output>
</rule>
</application>
</inference>
```

ANEXO 3 – Base de dados Parâmetros de Integração

```

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

<integration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ruled_integration.xsd">
  <application>
    <applicationId>TAG_AVAC_APPLICATION</applicationId>
    <rule>
      <input index="0">-1.0 1.0</input>
      <action application="TAG_LUMINANCE_APPLICATION">-
16.0</action>
    </rule>
    <rule>
      <input index="0">6.0 7.0</input>
      <action application="TAG_LUMINANCE_APPLICATION">
32.0</action>
    </rule>
    <rule>
      <input index="1">0.0 20.0</input>
      <action application="TAG_LUMINANCE_APPLICATION">-
32.0</action>
    </rule>
    <rule>
      <input index="1">80.0 100.0</input>
      <action application="TAG_LUMINANCE_APPLICATION">
32.0</action>
    </rule>
  </application>
  <application>
    <applicationId>TAG_LUMINANCE_APPLICATION</applicationId>
    <rule>
      <input index="0">-128.0 -64.0</input>
      <action application="TAG_AVAC_APPLICATION">0.0
0.0</action>
    </rule>
    <rule>
      <input index="0">-16.0 16.0</input>
      <action application="TAG_AVAC_APPLICATION">1.0
25.0</action>
    </rule>
  </application>
</integration>

```

```
        <rule>
            <input index="0">64.0 128.0</input>
            <action application="TAG_AVAC_APPLICATION">5.0
75.0</action>
        </rule>
    </application>
</integration>
```