



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE SISTEMAS DE INFORMAÇÃO

ALEXANDRA VALESCA FERREIRA BRITO

**UMA METODOLOGIA PARA A INTEGRAÇÃO DE REDES DE SENSORES SEM
FIO E COMPUTAÇÃO EM NUVEM**

QUIXADÁ

2014

ALEXANDRA VALESCA FERREIRA BRITO

**UMA METODOLOGIA PARA A INTEGRAÇÃO DE REDES DE SENSORES SEM
FIO E COMPUTAÇÃO EM NUVEM**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de Concentração: Computação.

Orientadora: Prof. Dra. Atslands Rego da Rocha.

Coorientador: Prof. Dr. Flávio Rubens de Carvalho Sousa.

QUIXADÁ

2014

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca do Campus de Quixadá

B862m Brito, Alexandra Valesca Ferreira Brito
Uma metodologia para a integração de redes de sensores sem fio e computação em nuvem /
Alexandra Valesca Ferreira Brito. – 2014.
51 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de
Sistemas de Informação, Quixadá, 2014.

Orientação: Profa. Dra. Atslands Rego da Rocha
Coorientação: Prof. Dr. Flávio Rubens de Carvalho Sousa
Área de concentração: Computação

1. Computação em nuvem 2. Redes remotas (Redes de computadores) 3. Sistemas de comunicação
sem fio I. Título.

CDD 621.382

ALEXANDRA VALESCA FERREIRA BRITO

**UMA METODOLOGIA PARA A INTEGRAÇÃO DE REDES DE SENSORES SEM
FIO E COMPUTAÇÃO EM NUVEM**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação.

Aprovado em ___/___/_____.

BANCA EXAMINADORA

Prof. Dra. Atslands Rego da Rocha (Orientadora)

Universidade Federal do Ceará (UFC)

Prof. Dr. Flávio Rubens de Carvalho Sousa (Coorientador)

Universidade Federal do Ceará (UFC)

Prof. Dr. Lincoln Souza Rocha

Universidade Federal do Ceará (UFC)

Dedico este trabalho à Deus e ao meu pai Joel Ferreira da Silva (*In memoriam*).

AGRADECIMENTOS

À Deus, por Ele ter me concedido o dom da vida e tudo o que tornou possível que eu chegasse até aqui.

Aos meus pais, Liduína Brito e Joel Ferreira da Silva (*In memoriam*) e à minha madrinha Fábiana, por toda a dedicação, pela educação que me foi dada e por sempre terem investido e incentivado meus estudos.

Aos meus irmãos, Joelma e Carlos Augusto pela amizade, motivação e pelos momentos de descontração proporcionados.

Ao meu esposo, Paulo Henrique por todo o amor, carinho e compreensão que me encorajaram a concluir essa importante etapa da minha vida.

À todos os familiares que me incentivaram, em especial à minha tia “Fransquinha” e à minha sogra Maria Lucila por todo o auxílio e motivação.

Aos meus orientadores, Atslands Rocha e Flávio Sousa pela oportunidade de realizar esse trabalho e por todo o apoio que me foi prestado.

Ao professor Marcos Dantas, pelas sugestões e contribuições feitas acerca deste trabalho.

Ao bolsista do Núcleo de Capacitação em Desenvolvimento de Sistemas Embarcados – NDSE, Carlos Carvalho, pela paciência e por todo o conhecimento compartilhado.

À todos os meus colegas e amigos, em especial ao meu grande amigo: José Diógenes, por sempre ter me ajudado quando precisei e pela maravilhosa convivência nesses quatro anos. Agradeço também aos meus amigos Bruno Leandro, Micaelly Priscila e Thalita Maria, por todos os momentos de estudo e lazer que pudemos compartilhar.

“Tudo o que o homem não conhece, não existe para ele. Por isso o mundo tem, para cada um, o tamanho que abrange o seu conhecimento.”

Carlos Bernardo González Pecotche

RESUMO

As Redes de Sensores sem Fio (RSSFs) são redes constituídas por inúmeros dispositivos denominados nós sensores que possuem capacidade de sensoriamento, armazenamento e processamento de dados, mas seus recursos computacionais como memória e energia são limitados. A Computação em Nuvem por sua vez, dispõe de um ambiente flexível que permite o armazenamento e processamento de enormes quantidades de dados, com o custo baseado no uso. Devido ao recente crescimento da utilização de RSSFs em basicamente todos os lugares, tornou-se necessário suprir as limitações encontradas nessas redes. Assim, baseado nas características promissoras da Computação em Nuvem, tornou-se então viável integrar essas duas tecnologias a fim de suprir as necessidades das RSSFs. Dessa forma, este trabalho apresenta uma metodologia para integração de RSSFs e Computação em Nuvem para mostrar como as características singulares de cada uma destas duas tecnologias podem ser melhor utilizadas.

Palavras-chave: Redes de Sensores sem Fio. Computação em Nuvem. Integração de RSSFs e Computação em Nuvem.

ABSTRACT

The Wireless Sensor Networks (WSNs) consist of numerous networking devices denominated sensor nodes that are capable of sensing, processing and data storage, but their computational resources such as memory and energy are limited. Cloud Computing provides a flexible environment that allows the storage and processing of large amounts of data, with the cost based on usage. Due to use of WSNs in basically everywhere, it became necessary to overcome the limitations found in these networks. Thus, based on the promising features of Cloud Computing, and then became feasible to integrate these two technologies in order to meet the needs of WSNs. Thus, this present work presents a methodology for integrating WSNs and Cloud Computing to show how the unique characteristics of each of these two technologies can be best utilized.

Keywords: Wireless Sensor Networks. Cloud Computing. Integration of WSNs and Cloud Computing

LISTA DE FIGURAS

FIGURA 1 – VISÃO GERAL DA SOLUÇÃO.....	17
FIGURA 2 – AGREGAÇÃO DE DADOS COLETADOS INDIVIDUALMENTE	20
FIGURA 3 – HARDWARE BÁSICO DE UM NÓ SENSOR	22
FIGURA 4 – PLATAFORMA ARDUINO.....	24
FIGURA 5 – IDE DO ARDUINO.....	25
FIGURA 6 – MÓDULOS XBEE E XBEE-PRO	25
FIGURA 7 – SOFTWARE XCTU	26
FIGURA 8 – AMBIENTE DE COMPUTAÇÃO EM NUVEM.....	27
FIGURA 9 – INTERAÇÃO DE UMA RSSF COM A NUVEM.....	35
FIGURA 10 – TOPOLOGIA DA RSSF	36
FIGURA 11 – NÓ SENSOR	38
FIGURA 12 – NÓ ATUADOR	39
FIGURA 13 – NOVA TOPOLOGIA DA RSSF	41

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CPU	Central Processing Unit
EC2	Amazon Elastic Cloud Computing
IaaS	Infraestrutura como um Serviço
IDE	Integrated Development Environment
NR-17	Norma Regulamentadora 17
PaaS	Plataforma como um Serviço
RSAFs	Redes de Sensores e Atuadores sem Fio
RSSFs	Redes de Sensores sem Fio
SaaS	Software como um Serviço
SGBDs	Sistemas de Gerenciamento de Banco de Dados
TI	Tecnologia da Informação
UFC	Universidade Federal do Ceará
VM	Virtual Machine

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS.....	15
1.1.1	<i>Objetivo geral</i>	15
1.1.2	<i>Objetivos específicos</i>	16
2	TRABALHOS RELACIONADOS	17
2.1	LEITÃO <i>ET AL.</i> (2012).....	17
2.2	ALAMRI <i>ET AL.</i> (2013).....	18
3	REVISÃO BIBLIOGRÁFICA	19
3.1	REDES DE SENSORES	19
3.1.1	<i>Características das RSSFs</i>	19
3.1.2	<i>Organização das RSSFs</i>	22
3.1.2.1	Organização Plana	22
3.1.2.2	Organização Hierárquica	23
3.1.3	<i>Plataforma Arduino</i>	23
3.1.3.1	XBee	25
3.2	COMPUTAÇÃO EM NUVEM.....	27
3.2.1	<i>Características Essenciais</i>	28
3.2.2	<i>Modelos de Serviços</i>	29
3.2.2.1	Software como um Serviço (SaaS).....	29
3.2.2.2	Plataforma como um Serviço (PaaS).....	29
3.2.2.3	Infraestrutura como um Serviço (IaaS)	30
4	PROCEDIMENTOS METODOLÓGICOS	31
5	INTEGRAÇÃO DE RSSFS E COMPUTAÇÃO EM NUVEM	32
5.1	PASSO 1: IDENTIFICAR OS ELEMENTOS DE UMA RSSF	32
5.1.1	<i>Nós sensores</i>	32
5.1.2	<i>Nós atuadores</i>	33
5.1.3	<i>Nós coordenadores</i>	33
5.2	PASSO 2: DEFINIR OS SERVIÇOS A SEREM UTILIZADOS NA NUVEM.....	33
5.3	PASSO 3: INTERAÇÃO ENTRE A RSSF E A NUVEM.....	34
5.3.1	<i>Configuração do nó coordenador</i>	34

5.3.2	<i>Comunicação da RSSF com a Nuvem</i>	34
6	EXEMPLO DE USO DA METODOLOGIA	36
6.1	APLICAÇÃO E RSSF.....	36
6.1.1	<i>Elementos da RSSF</i>	37
6.1.1.1	Nós sensores	37
6.1.1.2	Nó atuador	38
6.1.1.3	Nó coordenador	39
6.2	SERVIÇOS DA NUVEM	39
6.3	INTERAÇÃO DA RSSF COM A NUVEM	40
6.3.1	<i>Configuração do nó coordenador</i>	41
6.3.2	<i>Comunicação da RSSF com a Nuvem</i>	41
6.3.3	<i>Dados coletados</i>	42
7	ANÁLISE DO EXEMPLO DE USO	43
7.1	BENEFÍCIOS PARA A RSSF	43
7.2	BENEFÍCIOS DO USO DA NUVEM	43
8	CONSIDERAÇÕES FINAIS	45
8.1	CONCLUSÃO	45
8.2	TRABALHOS FUTUROS	45
	REFERÊNCIAS	46
	APÊNDICE A – CÓDIGO DOS SCRIPTS UTILIZADOS NA METODOLOGIA	48
	ANEXO A – EXTENSÃO DO CÓDIGO DA APLICAÇÃO UTILIZADA COMO EXEMPLO DE USO DA METODOLOGIA	51

1 INTRODUÇÃO

Atualmente, aplicações para Redes de Sensores sem Fio (RSSFs) têm sido utilizadas em diversas áreas importantes, como agricultura, mobilidade urbana, industrial e assistência à saúde. Esse tipo de rede é constituído por uma grande quantidade de dispositivos sem fio, geralmente chamados de nós sensores, os quais podem se conectar a uma rede de computadores, tal como a internet, através de um *gateway*. Os nós sensores são distribuídos em uma determinada região de interesse com o intuito de monitorar e coletar informações desta região. Para isso, em cada nó sensor é possível se ter diferentes unidades de sensoriamento, como temperatura, umidade, aceleração e acústico.

Os sensores possuem recursos computacionais limitados como memória, processamento e principalmente energia. As RSSFs devem ser autônomas, ou seja, os nós sensores devem se posicionar, comunicar, recolher e processar informações de forma independente (LOUREIRO, 2006). Entretanto, a autonomia de uma RSSF é comprometida pela quantidade limitada de energia disponível em cada sensor na rede, pois assim, é necessário que seja feita a manutenção da rede para reabastecer os sensores. Desta forma, como atualmente as RSSFs podem ser implantadas em praticamente todos os lugares, o principal objetivo destas redes tem sido não realizar a manutenção dos sensores ou pelo menos reduzir a frequência de manutenção. Além disso, as RSSFs geralmente realizam o processamento de uma imensa quantidade de dados obtidos pelos inúmeros sensores dispostos na determinada área de interesse. Assim, uma quantidade significativa de energia é gasta com tarefas de comunicação, armazenamento e processamento nos nós.

Como o ambiente de RSSF é dinâmico e a quantidade de dados coletados pode variar de acordo com os requisitos de monitoramento, faz-se necessário armazenar e processar estes dados de forma rápida, característica presente nos ambientes de Computação em Nuvem. Assim sendo, a Computação em Nuvem pode ajustar a capacidade de recursos alocados de acordo com a demanda da RSSF e, assim, fornecer uma solução eficiente neste contexto. A Computação em Nuvem está se tornando, portanto, uma tecnologia promissora para suprir as limitações de recursos das RSSFs, pois permite armazenar e processar grandes volumes de dados de forma simples e eficiente (SOUSA *et al.*, 2009). O principal objetivo desta recente tecnologia é fornecer uma alta flexibilidade de computação, armazenamento e serviços de software de forma escalável e virtualizada, com o custo baseado no uso (ALAMRI *et al.*, 2013).

Neste contexto, integrar Redes de Sensores Sem Fio e Computação em Nuvem surge como uma alternativa a ser considerada, pois desta forma, as necessidades computacionais dos sensores poderiam ser supridas pelos serviços oferecidos pela Computação em Nuvem. Contudo, o processo de integração destas tecnologias tem recebido pouca atenção (ALAMRI *et al.*, 2013; LEITÃO *et al.*, 2012). Assim, para o desenvolvimento deste trabalho foi feita uma análise parcial das vantagens e desvantagens (Seção 2) de cada um destes trabalhos relacionados.

Visando abordar uma solução para a limitação de recursos das RSSFs, este trabalho apresenta uma metodologia para a integração de RSSF e Computação em Nuvem. Esta metodologia visa melhorar o uso das RSSF por meio da característica de elasticidade da nuvem, onde novos recursos poderão ser adicionados à nuvem de acordo com a demanda da RSSF. Além disso, esta metodologia será independente das tecnologias utilizadas nos sensores ou na nuvem, possibilitando seu uso em diferentes ambientes.

Além desta seção de introdução, este trabalho está estruturado em mais sete seções. A seção 2 apresenta os trabalhos relacionados que serviram de base para este trabalho. A seção 3 apresenta a revisão bibliográfica, na qual são expostos os conceitos de RSSFs e Computação em Nuvem. A seção 4 apresenta a metodologia para a Integração de RSSFs e Computação em Nuvem definida neste trabalho. Na seção 5 são apresentados os procedimentos metodológicos que foram necessários para a execução do trabalho. Na seção 6 é apresentado um exemplo de uso da metodologia. A seção 7 apresenta uma análise realizada a partir do exemplo de uso, e finalmente, na seção 8 são apresentadas as considerações finais e os trabalhos futuros.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver uma metodologia que permita a integração entre Redes de Sensores sem Fio e Computação em Nuvem. Esta metodologia abordará uma solução para alguns dos desafios existentes em RSSFs, como a eficiência energética através do processamento e armazenamento dos dados obtidos, utilizando a Computação em Nuvem.

1.1.2 Objetivos específicos

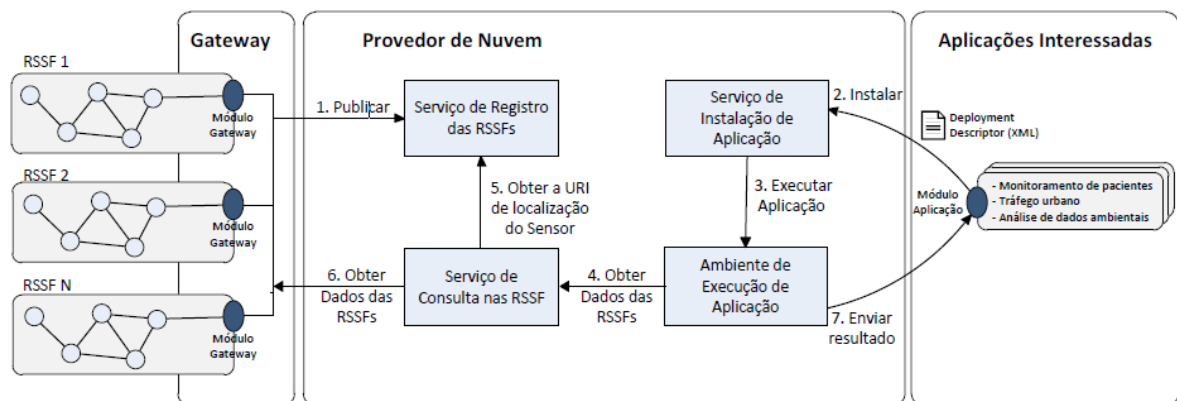
- Propor uma metodologia para integração de RSSFs e Computação em Nuvem;
- Identificar uma aplicação para RSSFs que mostre os benefícios da integração das duas tecnologias;
- Realizar um estudo de caso de integração de RSSFs e Computação em Nuvem, utilizando a aplicação para RSSFs escolhida a fim de validar a metodologia.

2 TRABALHOS RELACIONADOS

2.1 Leitão *et al.* (2012)

Leitão *et al.* 2012 propõe um *framework* que tem como objetivo facilitar a integração de RSSFs e Computação em Nuvem simplificando o desenvolvimento de aplicações para RSSF que utilizam serviços da Nuvem. A figura 1 mostra a visão geral da solução proposta por este autor.

Figura 1 – Visão geral da solução



Fonte: Leitão (2012).

Conforme mostrado no artigo, nesta figura percebemos a existência de três ambientes distintos que trabalham em conjunto – gateway, provedor de nuvem e aplicações interessadas pelos dados coletados. Assim, baseando-se em cenários de aplicações geograficamente distribuídas, a definição de um mecanismo de comunicação entre esses ambientes heterogêneos, torna-se o foco principal desta abordagem. Como solução para essa questão, o autor definiu que todos os serviços disponibilizados seriam expostos através de *Web services*, garantindo assim uma maior integração com outras tecnologias. Para isso, foi desenvolvida uma funcionalidade para adicionar um servidor web embarcado à solução.

Por fim, para demonstrar como a integração de RSSFs e Nuvem pode ser facilitada através do *framework* proposto utilizando esta solução, o autor realizou um estudo de caso integrando uma RSSF real chamada *Wisebed* com uma aplicação implantada na Nuvem *Amazon EC2*. Tal aplicação teve como principal objetivo, monitorar as variações significativas na quantidade de sensores presentes em um *testbed* de RSSF.

2.2 Alamri *et al.* (2013)

Alamri *et al.* 2013 apresenta um estudo abrangente sobre a integração de RSSF e Computação em Nuvem, abordando uma visão geral dos principais conceitos, definições, arquiteturas e aplicações desta infraestrutura. Para isso, inicialmente ele mostra as principais características de RSSFs, enfatizando as suas limitações e em seguida apresenta as principais características da Computação em Nuvem, justificando a utilização desta tecnologia para suprir as limitações de RSSFs.

Além disso, no estudo proposto por este autor são apresentadas as principais vantagens alcançadas com a integração de RSSFs e Computação em Nuvem. Tais vantagens são: análise de dados; escalabilidade; colaboração; visualização; provisionamento livre de armazenamento e processamento; provisionamento dinâmico de serviços; múltiplos prestadores de serviço; automação de serviços; flexibilidade; agilidade de serviços; otimização de recursos; e resposta em tempo real.

Outros importantes tópicos abordados neste artigo são: um modelo de ciclo de vida de um serviço e uma abordagem da arquitetura de integração de RSSFs e Computação em Nuvem em camadas; criação de serviços para a integração de RSSFs e Computação em Nuvem e renovação de suas capacidades; e um provisionamento de multisserviços em múltiplas plataformas. Este artigo apresenta também uma comparação técnica de diferentes abordagens de infraestruturas para a integração de RSSFs e Computação em Nuvem, diversos problemas que possam surgir com a integração dessas tecnologias e ainda como lidar com tais problemas. Por fim, segundo o autor, a pesquisa desafia as soluções existentes e abordagens, bem como traz futuras direções de pesquisa.

3 REVISÃO BIBLIOGRÁFICA

Esta seção aborda os principais conceitos de Redes de Sensores sem Fio e Computação em Nuvem, estudados para o desenvolvimento deste trabalho.

3.1 Redes de Sensores

As Redes de Sensores sem Fio surgiram com um grande avanço tecnológico nas áreas de sensores, circuitos integrados e comunicação sem fio ocorrido a partir do final dos anos de 1990 (LOUREIRO *et al.*, 2003). Geralmente, estas redes são formadas por uma enorme quantidade de dispositivos autônomos, chamados nós sensores, que coletam e processam as informações desejadas das áreas de interesse e se comunicam para efetuar sua disseminação. Assim, as RSSFs tem como principal objetivo monitorar e controlar um determinado ambiente de interesse sem intervenção humana (RUIZ *et al.*, 2004).

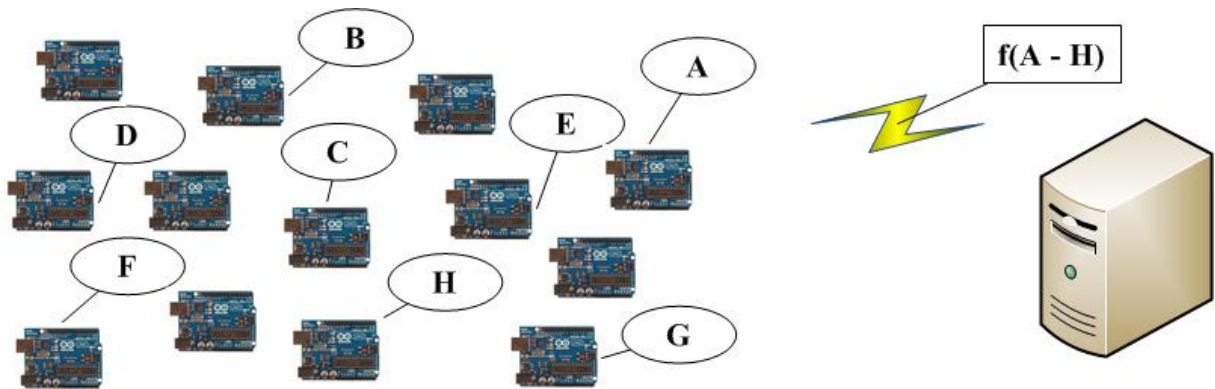
3.1.1 Características das RSSFs

As Redes de Sensores sem Fio possuem características singulares que variam de acordo com fatores como as áreas de interesse, requisitos das aplicações, entre outros. O conhecimento das características das RSSFs é um ponto relevante para o desenvolvimento de aplicações para as mesmas.

Segundo Loureiro *et al.* (2003), as principais características das RSSFs são:

- Endereçamento dos sensores ou nós: o endereçamento de cada sensor pode ser feito individualmente ou não, pois é determinado pela aplicação, ou seja, é feito de acordo com a necessidade de cada aplicação específica.
- Agregação dos dados: é a capacidade que uma RSSF possui de agregar ou sumarizar os dados coletados pelos sensores. Quando a rede possui essa funcionalidade, os dados coletados são agregados ainda na rede, ou seja, antes de serem enviados à estação base. Com isso, há uma redução na quantidade de mensagens transmitidas pela rede (Figura 2).

Figura 2 – Agregação de dados coletados individualmente



Fonte: elaborada pela autora.

Na figura 2, é mostrada uma RSSF com vários nós sensores e em vez de cada um deles se comunicar com o servidor, eles primeiramente sumarizam os dados na rede, e por fim apenas um envia todos os dados agregados de uma vez só ($f(A - H)$).

- Mobilidade dos sensores: define os sensores em dinâmicos ou estáticos, de acordo com o sistema onde está sendo feita a coleta de dados. Assim, a mobilidade dos sensores mostra se os sensores podem ser móveis ou não dependendo da necessidade da área de interesse onde a RSSFs está implantada.
- Restrições dos dados coletados: verifica se existe algum tipo de restrição nos dados coletados pelos sensores. Os dados coletados podem ter restrições de tipo, tempo de envio e tamanho.
- Quantidade de sensores: as RSSFs são constituídas geralmente por um número significativo de nós sensores. Portanto, a escalabilidade dos sensores nestas redes é um ponto importante a ser considerado.
- Limitação da energia disponível: a quantidade de energia disponível em cada sensor determina o seu tempo de vida. Logo, aplicações, protocolos e algoritmos para RSSFs não devem ser escolhidos apenas por sua robustez ou capacidade, a quantidade de energia consumida, também deve ser considerada. Os elementos consumidores de energia em uma RSSF normalmente dependem de uma bateria com capacidade finita armazenada e são listados a seguir:
 - Rádio: representa todo o sistema de transmissão e recepção de dados, amplificador e antena. O consumo de energia deste componente varia acordo com a operação. Geralmente a transmissão de dados consome mais energia que a sua recepção;

- Processador: elemento que realiza o processamento central do nó sensor. O seu consumo de energia depende da velocidade do relógio e do modo de operação e pode ser medido pelo número de ciclos de relógio;
- Sensores: representam as unidades de sensoriamento. O consumo de energia depende do modo de operação e do tipo de grandeza medida.
- Auto-organização da rede: em uma RSSF, os sensores podem ser perdidos devido à destruição física, podem falhar por esgotamento de bateria ou podem ficar incomunicáveis por causa de problemas no canal de comunicação sem fio. Além disso, novos sensores podem ser inseridos na rede ou sensores inativos podem tornar-se ativos. Em ambos os casos, é necessário a existência de mecanismos de auto-organização para gerenciar estas mudanças, de forma que a rede continue funcionando corretamente. Este gerenciamento deve ser automático e periódico, pois um gerenciamento manual é inviável devido a problemas de escalabilidade.
- Tarefas colaborativas: Em uma RSSF, as tarefas são executadas de forma colaborativa. Assim, o principal objetivo de uma RSSF é executar estas tarefas provendo mecanismos de comunicação, verificando e estimando os eventos de interesse. Entretanto, devido às restrições das RSSFs, normalmente os dados são sumarizados para melhorar o desempenho no processo de detecção de eventos. A aplicação em execução é que define como será o processo de sumarização.
- Capacidade de responder a consultas: as consultas realizadas sobre determinada informação coletada em uma região de interesse podem ser enviadas para um nó individual ou um grupo de nós. Porém, dependendo do grau de sumarização executado, a transmissão de dados pela rede até o nó sorvedouro pode se tornar inviável. Nestes casos, pode ser necessária a definição de vários nós sorvedouros para coletar os dados de um determinado local e responder as consultas referentes aos nós definidos para ele.

Neste trabalho, foram consideradas algumas das características das RSSFs, pois para realizar a integração de uma RSSF com a Nuvem é preciso saber qual serviço da Nuvem será utilizado para suprir as necessidades da RSSF de acordo com as suas características.

3.1.2 Organização das RSSFs

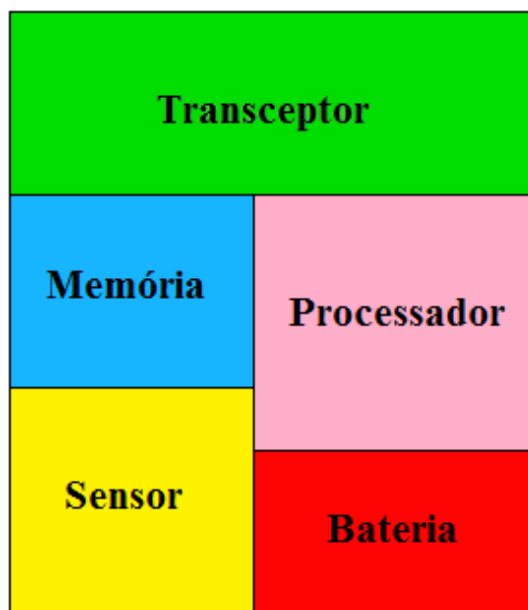
Uma RSSF pode ser organizada de forma plana ou hierárquica. Tais formas de organização de RSSFs são apresentadas a seguir.

3.1.2.1 Organização Plana

Na organização plana, os principais elementos de uma RSSF são os nós sensores e nó(s) coordenador(es). Os nós sensores são dispositivos autônomos que realizam tarefas de sensoriamento e podem processar dados coletados localmente ou de forma distribuída entre os nós “vizinhos”.

Os componentes básicos de um nó sensor são: transceptor para comunicação sem fio, fonte de energia (bateria), unidade de sensoriamento, memória e processador (Figura 3). O componente lógico de um nó sensor é o software executado no processador. Um nó sensor possui pequena capacidade computacional e pouca disponibilidade de energia, mas quando vários sensores trabalham colaborativamente, torna-se possível a realização de uma grande tarefa (RUIZ *et al.*, 2004).

Figura 3 – Hardware básico de um nó sensor



Fonte: elaborada pela autora.

As informações coletadas pelos nós sensores podem ser enviadas para um nó coordenador. Os nós coordenadores de uma RSSFs são os elementos que a rede utiliza para comunicar-se com um servidor ou com outras redes. Em algumas RSSFs, os nós coordenadores podem ser implementados em apenas um nó sensor. Além disso, o nó coordenador geralmente possui uma maior capacidade computacional, não tendo restrições de recursos, e funciona como ponto de entrada para os requisitos de aplicações e pontos de encontro para os dados coletados pelos sensores (ROCHA, 2012).

3.1.2.2 Organização Hierárquica

Na organização hierárquica, além de nós sensores e nós coordenadores, existe a figura de um nó líder de *cluster*, surgindo assim a ideia de clusterização. A clusterização pode ser definida como a divisão de uma RSSF em grupos (*clusters*), os quais possuem um líder (*cluster-head* – *CH*) que é responsável por enviar os dados coletados pelos nós sensores deste *cluster*, para o nó(s) coordenador(es) da rede por meio de uma comunicação multissalto (ROCHA, 2012).

O uso da técnica de clusterização em RSSFs tem como principais benefícios prover escalabilidade e prolongar o tempo de vida da rede, devido à economia de energia. Esta economia de energia acontece, pois geralmente, a distância entre os nós sensores de um determinado *cluster* para o seu respectivo líder, é menor do que a distância dos nós sensores e o nó coordenador, ou seja, com uma distância menor a energia gasta para enviar os dados também será menor (ROCHA, 2012).

Outra função do líder de *cluster* é coordenar a comunicação entre os nós sensores do seu respectivo *cluster* e realizar a fusão dos dados coletados por estes nós sensores. A decisão de qual nó sensor terá a função de líder de *cluster*, pode ser tomada previamente pelo desenvolvedor da RSSF ou dinamicamente pelos nós sensores. Esta decisão é relevante para os métodos de clusterização, pois as funcionalidades delegadas ao líder promovem um maior consumo de energia do que outros nós sensores do *cluster* (ROCHA, 2012).

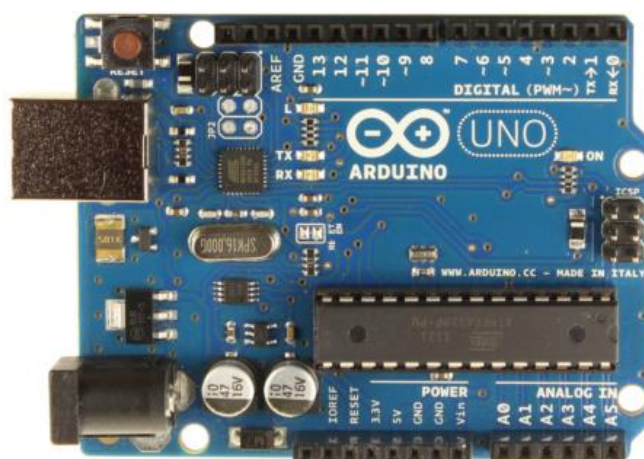
3.1.3 Plataforma Arduino

Segundo McRoberts (2011, p. 22), “o Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software”. Portanto, podemos considerar que um Arduino é um

microcontrolador em que se pode programar o processamento de entradas e saídas entre o dispositivo e os componentes externos conectados a ele.

A plataforma Arduino (Figura 4) possui hardware e software de código-fonte aberto, sendo composta por uma placa microcontroladora e um ambiente de desenvolvimento (*Integrated Development Environment – IDE*).

Figura 4 – Plataforma Arduino

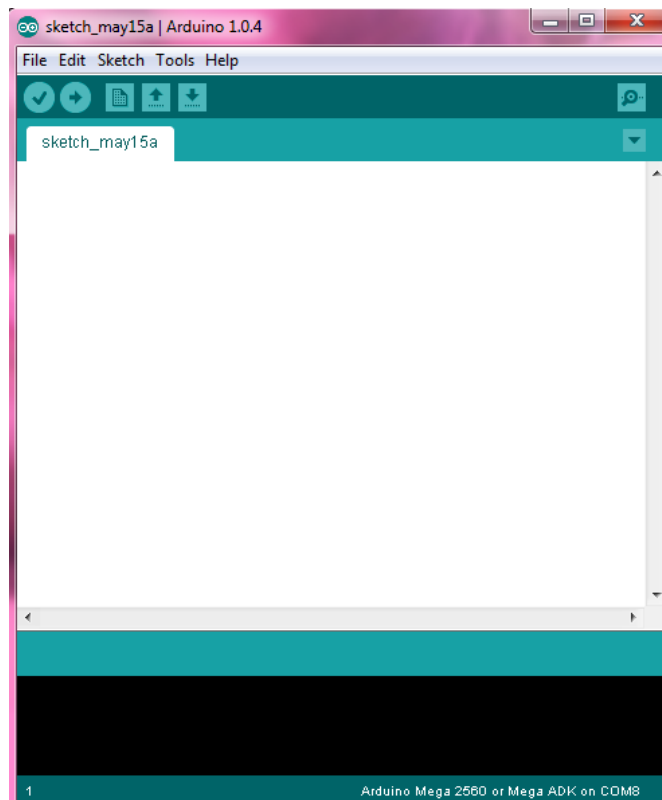


Fonte: Arduino S.A. (2013).

O microcontrolador utilizado na placa do Arduino é constituído por portas analógicas, portas para comunicação serial e entradas e saídas digitais. Além disso, existem módulos externos que aumentam as funcionalidades da placa do Arduino, chamado de *Shields*. Estes módulos se comunicam com o microcontrolador através de pinos ou barramento serial, permitindo a expansão das funcionalidades, pois possuem outras interfaces, como *Ethernet* ou *Wi-fi* (CUNHA, 2013).

A IDE (Figura 5) oferece um ambiente para a escrita do código, *debuga* e importa os comandos para o microcontrolador. O software responsável por passar instruções ao microcontrolador é escrito na linguagem C/C++, o que melhora diversos aspectos da programação de baixo nível. Além disso, o Arduino pode ser utilizado para desenvolver diversos dispositivos interativos, tendo entradas a partir de uma variedade de sensores ou interruptores, podendo controlar uma variedade de luzes, motores e outras saídas físicas (MELLIS, 2009).

Figura 5 – IDE do Arduino



Fonte: elaborada pela autora.

3.1.3.1 XBee

O XBee é um módulo de comunicação que provê conectividade sem fio para as placas Arduino e possui um microcontrolador com o protocolo ZigBee embarcado. Esse módulo possui duas versões, o XBee e o XBee-PRO (Figura 6).

Figura 6 – Módulos XBee e XBee-PRO

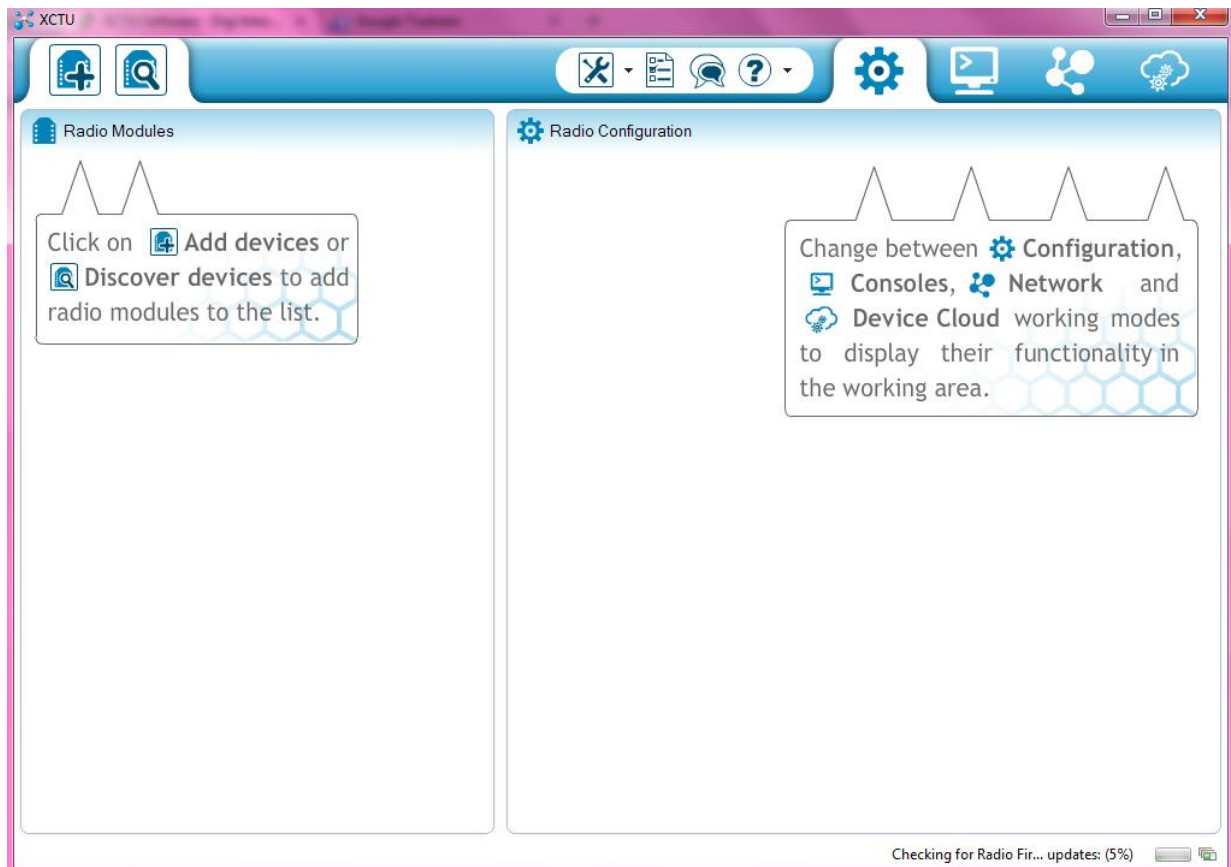


Fonte: Digi International (2014).

As duas versões oferecem as mesmas funcionalidades de conectividade e facilidade de programação e configuração (DIGI, 2014), a diferença está apenas na potência de transmissão (ESCHNER, 2011).

Para realizar a configuração do módulo XBee é necessária a utilização do software XCTU (Figura 7). O XCTU é um software multi-plataforma livre, projetado para permitir que os desenvolvedores possam interagir com módulos XBee através de uma interface gráfica simples de usar (DIGI, 2014). Ele inclui ferramentas que tornam mais fácil realizar a configuração e o teste de módulos Xbee.

Figura 7 – Software XCTU



Fonte: elaborada pela autora.

O protocolo ZigBee utilizado pelo XBee é um protocolo destinado a aplicações embarcadas que visa o baixo consumo de energia e baixa transferências de dados (CUNHA, 2013). A tecnologia ZigBee foi desenvolvida com o objetivo de criar redes em malha (*mesh*), mas suporta outras topologias como redes em estrela (*star*) e redes em árvore (*cluster tree*), admitindo a formação de redes *ad-hoc*, explicando a sua grande utilização em rede de sensores sem fio, ainda mais por fornecer confiabilidade e maior alcance (SALEIRO, 2013).

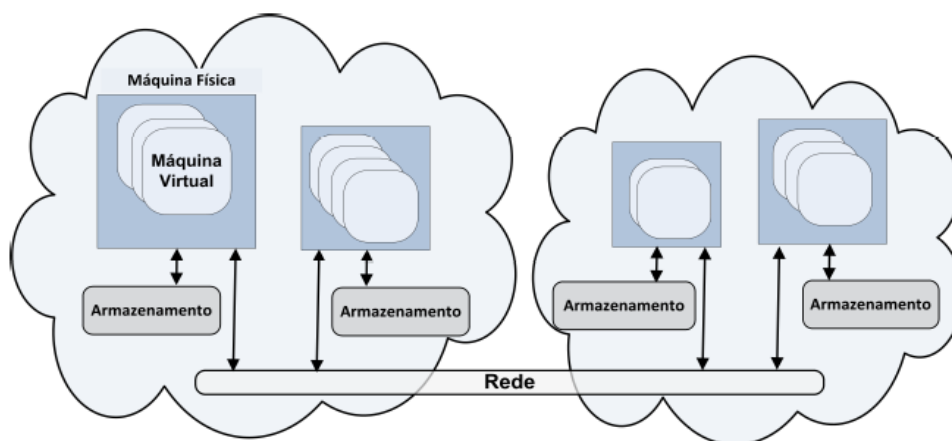
3.2 Computação em Nuvem

A Computação em Nuvem é uma tendência recente da tecnologia. ‘Nuvem’ é uma metáfora para a internet e seus serviços, sendo composta por milhares de computadores interconectados. O objetivo da Computação em Nuvem é proporcionar serviços de tecnologia da informação (TI) de acordo com a demanda de cada usuário. Os dados podem ser acessados a partir de um navegador web – independentemente de sua localização geográfica. Isso significa que usuários podem armazenar e acessar seus documentos pessoais na internet por meio de aplicativos como *Dropbox* ou *Sugarsync*, por exemplo; podem utilizar um editor de textos on-line como o *Google Docs*, por exemplo; e empresas ou universidades podem utilizar grandes infraestruturas de computadores, como *Amazon Web Services*, para processar e analisar dados, reduzindo custos e permitindo mais foco em inovação (SOUSA *et al.*, 2009).

Na nuvem os recursos são monitorados, possibilitando a tarifação de acordo com o uso, da mesma maneira que os serviços de telefonia e energia. Com isso, os usuários podem alugar capacidade de computação e armazenamento, em vez de investir na aquisição e instalação de equipamentos.

A infraestrutura do ambiente de Computação em Nuvem geralmente é formada por milhares de máquinas físicas de baixo custo, conectadas através de uma rede (Figura 8). Estas máquinas físicas possuem as mesmas configurações de software, mas podem ter capacidade de hardware como CPU, armazenamento e memória diferentes. Além disso, cada máquina física possui um determinado número de máquinas virtuais (VM) executando interiormente, de acordo com a capacidade de hardware disponível na máquina física (SOUSA *et al.*, 2010).

Figura 8 – Ambiente de Computação em Nuvem



Fonte: Sousa *et al.* (2010).

3.2.1 Características Essenciais

Em Sousa *et al.* (2010) são definidas as características essenciais da Computação em Nuvem, bem como os benefícios oferecidos pelas soluções desta tecnologia. Assim, algumas destas características, quando unidas, fazem com que a Computação em Nuvem se diferencie de outros paradigmas, tornando-se única. Características como: a elasticidade rápida de recursos; o amplo acesso e a medição de serviço, por exemplo, são características básicas para uma solução de Computação em Nuvem. As características essenciais da Computação em Nuvem serão descritas a seguir.

- *Self-service* sob demanda: recursos computacionais como tempo de processamento no servidor ou armazenamento na rede podem ser adquiridos pelo usuário de acordo com a sua necessidade, sem precisar realizar uma interação humana com os provedores dos serviços.
- Amplo acesso: os recursos disponibilizados pela rede podem ser acessados por meio de mecanismos padronizados, para que dispositivos como celulares, laptops e PDAs possam ter acesso a estes recursos por meio de plataformas do tipo *thin*.
- *Pooling* de recursos: os recursos computacionais do provedor são organizados em um *pool* para permitir que estes recursos possam servir a vários usuários, utilizando diferentes recursos físicos e virtuais, atribuídos dinamicamente e ajustados de acordo com a necessidade dos usuários. Além disso, os usuários não precisam saber especificamente a localização física dos recursos adquiridos, eles podem definir a localização de forma abstrata, informando apenas o nome do país, estado ou base de dados.
- Elasticidade rápida: os usuários podem adquirir recursos de forma rápida e elástica, de acordo com a demanda que possuem, ou seja, caso a demanda do usuário aumente, ele solicitará o aumento dos recursos, caso contrário ele irá liberar os recursos. Com isso, os usuários interpretam que os recursos disponíveis para uso são ilimitados e podem ser adquiridos em qualquer quantidade e a qualquer momento.
- Serviço medido: os sistemas em nuvem controlam e otimizam o uso de recursos automaticamente, através de uma capacidade de medição. A automação é realizada no nível de abstração apropriado para o tipo de serviço –

armazenamento, processamento, largura de banda e contas de usuário ativas. Assim, o uso dos recursos pode ser monitorado e controlado, possibilitando transparência para o provedor e o usuário do serviço utilizado.

3.2.2 Modelos de Serviços

O ambiente de Computação em Nuvem possui três modelos de serviços – Software como um Serviço (SaaS), Plataforma como um Serviço (PaaS) e Infraestrutura como um Serviço (IaaS) (SOUSA *et al.*, 2009). Tais modelos são definidos na próxima subseção.

3.2.2.1 Software como um Serviço (SaaS)

O modelo de serviço de Software como um Serviço (SaaS) oferece sistemas de software com finalidades específicas e são disponibilizados para os usuários por meio da internet. Tais sistemas de software podem ser acessados através de vários dispositivos do usuário utilizando uma interface *thin cliente* como um navegador Web. No SaaS, o usuário não pode administrar ou controlar a infraestrutura subjacente – rede, servidores, sistemas operacionais, armazenamento ou características individuais da aplicação – exceto configurações específicas. Assim, os desenvolvedores podem focar em inovação e não em infraestrutura, tornando o desenvolvimento de sistemas de softwares mais rápido.

O fato de o software estar na Web, permite que o usuário o acesse de qualquer lugar e a qualquer momento, possibilitando uma maior integração entre unidades de uma mesma empresa, por exemplo, ou com outros serviços de software. Com isso, novos recursos podem ser adicionados automaticamente aos sistemas de software sem que os usuários percebam, tornando transparente a evolução e atualização dos sistemas. Além disso, o SaaS também diminui os custos, uma vez que a aquisição de licenças de sistemas de softwares não é mais necessária. Como exemplos de SaaS podemos destacar os serviços do *Google Docs*.

3.2.2.2 Plataforma como um Serviço (PaaS)

O modelo de Plataforma como um Serviço (PaaS) proporciona uma infraestrutura de alto nível de integração para implementar e testar aplicações na nuvem. No PaaS, o usuário também não pode administrar ou controlar a infraestrutura subjacente – rede, servidores, sistemas operacionais ou armazenamento, mas pode controlar as aplicações implantadas e,

possivelmente, configurar as aplicações hospedadas nesta infraestrutura. A PaaS oferece sistema operacional, linguagens de programação e ambientes de desenvolvimento para as aplicações, auxiliando o desenvolvimento de sistemas de software, já que fornece ferramentas de desenvolvimento e colaboração entre desenvolvedores.

Geralmente, os desenvolvedores dispõem de ambientes escaláveis, mas eles precisam aceitar algumas restrições sobre o tipo de software que pode ser desenvolvido. Estas restrições podem ser limitações que o ambiente impõe na concepção das aplicações ou até mesmo a utilização de Sistemas de Gerenciamento de Banco de Dados (SGBDs) do tipo chave-valor, ao invés de SGBDs relacionais. Além disso, a PaaS permite que os usuários utilizem serviços de terceiros, aumentando o uso do modelo de suporte, no qual os usuários se inscrevem para solicitações de serviços de TI ou para resoluções de problemas pela Web, melhorando o gerenciamento do trabalho e as responsabilidades das equipes de TI das empresas. Como exemplo de PaaS podemos destacar a *Google App Engine*.

3.2.2.3 Infraestrutura como um Serviço (IaaS)

O modelo de Infraestrutura como um Serviço (IaaS) é responsável por prover toda a infraestrutura necessária para a PaaS e o SaaS. A finalidade do IaaS é tornar mais fácil e acessível o fornecimento de recursos como servidores, rede, armazenamento e outros recursos de computação fundamentais para construir um ambiente sob demanda, como sistemas operacionais e aplicativos. Algumas características da IaaS são: uma interface única para administração da infraestrutura; uma *Application Programming Interface* (API) para interação com *hosts*, *switches*, balanceadores, roteadores e por fim, o suporte para a adição de novos equipamentos de forma simples e transparente. Geralmente, o usuário não administra ou controla a infraestrutura da nuvem, mas ele pode controlar os sistemas operacionais, o armazenamento e os aplicativos implantados, podendo também selecionar componentes de rede, tais como firewalls.

O termo IaaS pode ser definido como uma infraestrutura computacional baseada em técnicas de virtualização de recursos de computação. Esta infraestrutura pode escalar de forma dinâmica, aumentando ou diminuindo os recursos de acordo com as necessidades das aplicações. Assim, ao invés de comprar novos servidores e equipamentos de rede para a ampliação de serviços, pode-se aproveitar os recursos disponíveis e adicionar novos servidores virtuais à infraestrutura existente dinamicamente. Como exemplo de IaaS podemos destacar o *Amazon Elastic Cloud Computing* (EC2).

4 PROCEDIMENTOS METODOLÓGICOS

A integração de Redes de Sensores sem Fio e Computação em Nuvem tem se tornado bastante oportuna devido às características singulares de cada uma destas áreas. Assim, enquanto as RSSFs podem ser instaladas em quase todos os lugares e possuem altas restrições de hardware, a Computação em Nuvem oferece recursos computacionais ilimitados para serem utilizados de acordo com a necessidade de cada aplicação (LEITÃO, 2012).

Neste contexto, para a realização deste trabalho foi necessário inicialmente realizar um levantamento bibliográfico sobre Redes de Sensores sem Fio e Computação em Nuvem e uma análise de alguns trabalhos relacionados. Assim, foi analisado como a integração é realizada nestes trabalhos, enfatizando seus pontos fortes e fracos.

A segunda etapa consistiu em definir a metodologia. A partir da leitura e análise dos trabalhos relacionados, foram descritos os passos que compõem a metodologia e como eles deverão ser executados.

A terceira etapa consistiu em identificar uma aplicação para RSSFs adequada que possa ressaltar a relevância da integração de RSSFs com a Computação em Nuvem. Para isso, foram considerados os aspectos de armazenamento e processamento de dados feito pelos sensores desta aplicação. Dessa forma, definimos quais serviços da Nuvem podem ser utilizados para suprir a necessidade de recursos da RSSF.

A quarta etapa consistiu em aplicar a metodologia. Após a escolha da aplicação para RSSF e dos serviços da Nuvem a serem utilizados, foi feita a integração da RSSF com a Nuvem, de acordo com a metodologia proposta.

A quinta e última etapa consistiu em realizar um estudo de caso, no qual foram feitos experimentos baseados na integração da aplicação para RSSFs com a Computação em Nuvem. Neste estudo de caso, a aplicação para RSSFs seguiu os passos descritos na metodologia para se comunicar com a Nuvem, a fim de obter os serviços que lhe são necessários.

5 INTEGRAÇÃO DE RSSFs E COMPUTAÇÃO EM NUVEM

Nesta seção apresentamos a metodologia que foi definida para realizar a integração entre RSSFs e Computação em Nuvem. Os passos definidos na metodologia são descritos a seguir.

5.1 Passo 1: Identificar os elementos de uma RSSF

Conforme visto em Leitão *et al.* (2012), para realizar a integração de RSSFs com a Nuvem é necessário a divisão do ambiente em módulos, ou seja precisamos separar a RSSF, o meio intermediário e a Nuvem. Assim, nesta metodologia, inicialmente definimos como deve ser configurado o ambiente da RSSF.

Dessa forma, para integrar uma RSSF com a Nuvem precisamos saber quais são os elementos necessários para se construir uma RSSF e qual a função de cada um deles dentro da rede. Geralmente, uma RSSF é constituída pelos seguintes elementos: nós sensores, nós atuadores e nós coordenadores. A configuração destes elementos da RSSF pode ser feita de várias formas, nesta metodologia não definimos nenhum método de configuração destes elementos.

5.1.1 Nós sensores

Os nós sensores são os elementos base de uma RSSF. Eles têm como função monitorar o ambiente onde estão dispostos, coletar os dados de interesse destes ambientes e enviar tais dados diretamente para o nó coordenador ou para o nó atuador que em seguida, enviará para o nó coordenador.

A quantidade de nós sensores que compõem uma RSSF varia de acordo com a necessidade da aplicação desenvolvida para a RSSF, mas geralmente, uma RSSF é composta por inúmeros nós sensores. Portanto, antes de integrar uma RSSF com a Nuvem, é importante que a aplicação seja bem definida, pois desta forma saberemos a quantidade de sensores que a RSSF deve ter e como cada nó sensor irá se comportar na rede.

5.1.2 Nós atuadores

Os nós atuadores têm como função somente receber os dados coletados pelos nós sensores, realizar as ações especificadas pela aplicação desenvolvida para a RSSF e se necessário, enviar os dados coletados pelos nós sensores, para o nó coordenador. É importante ressaltar que estes nós não realizam coleta de dados, apenas executam ações.

A quantidade de nós atuadores que uma RSSF deve possuir também é definida a partir especificação da aplicação e da quantidade de nós sensores que compõem a rede. Dessa forma, uma rede pode possuir ou não nós atuadores, o usuário precisa identificar a necessidade de executar ou não ações dentro da RSSF, pois se a mesma for responsável apenas por coletar dados, a existência de um nó atuador é desnecessária.

5.1.3 Nós coordenadores

Os nós coordenadores têm como função receber todos os dados coletados pelos nós sensores e realizar a comunicação da RSSF com o meio externo, ou seja, com um servidor ou com outras redes. Os nós coordenadores podem ser definidos como o *gateway* da RSSF (LEITÃO *et al.*, 2012). Dessa forma, para realizar a comunicação da RSSF com o meio externo, o nó coordenador precisa ser configurado de acordo com a especificação da aplicação.

Geralmente, os nós coordenadores podem ser implementados em apenas um nó sensor, mas em algumas RSSFs é necessária a definição de dois ou mais nós coordenadores. Isso acontece devido a complexidade da aplicação desenvolvida para uma determinada RSSF e devido as funções exercidas pelos demais elementos da RSSF.

5.2 Passo 2: Definir os serviços a serem utilizados na Nuvem

Antes de integrar uma RSSF com a Nuvem é necessário definir quais serviços oferecidos pela Nuvem serão utilizados na integração. Essa fase da metodologia é importante, pois devido ao tamanho do ambiente da Nuvem, torna-se impossível integrar uma RSSF com toda a Nuvem (ALAMRI *et al.*, 2013). Portanto, esse passo da metodologia deve ser executado baseado na necessidade da aplicação desenvolvida para a RSSF, pois é a partir da especificação da aplicação que sabemos quais são as necessidades da RSSF que precisam ser supridas. Por exemplo, se temos uma aplicação para uma RSSF que realiza uma coleta

massiva de dados, é cabível que utilizemos um serviço de armazenamento de dados na Nuvem, pois desta forma estaremos economizando a energia dos nós sensores.

Além disso, definir os serviços a serem utilizados na Nuvem é necessário, pois é a partir desta definição que escolheremos qual instância da nuvem poderemos utilizar. Ao escolher a instância da Nuvem, podemos escolher também quais softwares – Banco de Dados, plataformas de desenvolvimento – iremos instalar nesta instância para poder realizar as ações necessárias pela RSSF na Nuvem.

5.3 Passo 3: Interação entre a RSSF e a Nuvem

Baseado em Alamri *et al.*, 2013, após definir uma RSSF e quais serviços da Nuvem serão utilizados, precisamos entender como pode acontecer a interação entre uma RSSF e a Nuvem. Esta interação é detalhada a seguir.

5.3.1 Configuração do nó coordenador

Inicialmente é preciso configurar o nó coordenador que, como já foi mostrado, é o módulo *gateway* da RFFS (LEITÃO *et al.*, 2013), ou seja, é o nó capaz de se comunicar com o meio externo à RSSF. Esta configuração deve ser feita de acordo com a especificação da aplicação e da definição de quais serviços da Nuvem serão utilizados. Por exemplo, se a aplicação define que os dados coletados pela RSSF devem ser processados na Nuvem e depois retornados para o nó coordenador, o nó coordenador deverá ser configurado para enviar os dados coletados pela RSSF e receber os dados processados pela Nuvem.

Realizar a configuração do nó coordenador de uma RSSF requer certa minuciosidade, pois é a partir da configuração deste nó que a integração da RSSF com a Nuvem será possível. Caso não haja a configuração correta do nó coordenador, torna-se impossível realizar a integração de uma RSSF com a Nuvem.

5.3.2 Comunicação da RSSF com a Nuvem

Após a configuração do nó coordenador, a RSSF poderá se comunicar com a Nuvem, entretanto ela não pode se comunicar diretamente (ALAMRI *et al.*, 2013). Assim, para que a RSSF se comunique com a Nuvem, é necessário que o usuário defina um método intermediário que permita a comunicação da RSSF com a Nuvem, de acordo com a aplicação

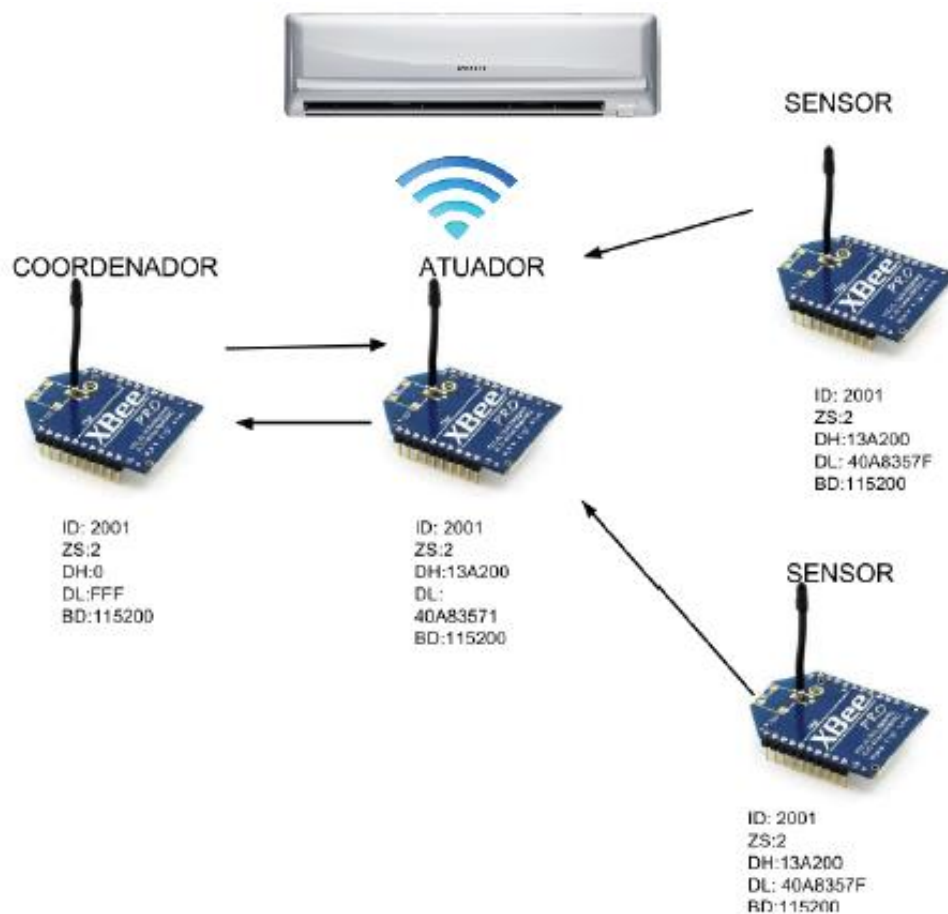
6 EXEMPLO DE USO DA METODOLOGIA

Nesta seção, é apresentado um exemplo de uso da metodologia, realizando a integração de uma RSSF com a Nuvem. Assim, é mostrado como podem ser executados todos os passos da metodologia definida neste trabalho.

6.1 Aplicação e RSSF

Para realizar este estudo de caso, inicialmente foi escolhida uma aplicação para uma RSSF com a finalidade de integrar tal rede à Nuvem. Dessa forma, foi escolhida uma aplicação que realiza a automação de condicionadores de ar para ser utilizada nos laboratórios e salas da Universidade Federal do Ceará (UFC) (CRUZ, 2013). Esta aplicação foi desenvolvida para uma RSSF cuja topologia é mostrada na figura 10.

Figura 10 – Topologia da RSSF



Fonte: Cruz (2013).

Esta aplicação tem como principal objetivo, buscar a comodidade do usuário e o uso da energia de forma eficaz (CRUZ, 2013), os quais são os princípios da automação. Assim, a justificativa do desenvolvimento desta aplicação foi a observação que devido ao crescimento da estrutura física e do número de aparelhos condicionadores de ar do Campus da UFC – Quixadá, o consumo de energia será maior e possivelmente o desperdício também, pois as pessoas responsáveis por ligar/desligar os aparelhos terão um maior trabalho de executar tais tarefas e precisarão de mais tempo para executá-las.

Esta aplicação foi desenvolvida utilizando a plataforma Arduino. Para realizar a comunicação entre os componentes da RSSF foram utilizados o módulo de comunicação XBee e o protocolo de comunicação ZigBee (CRUZ, 2013).

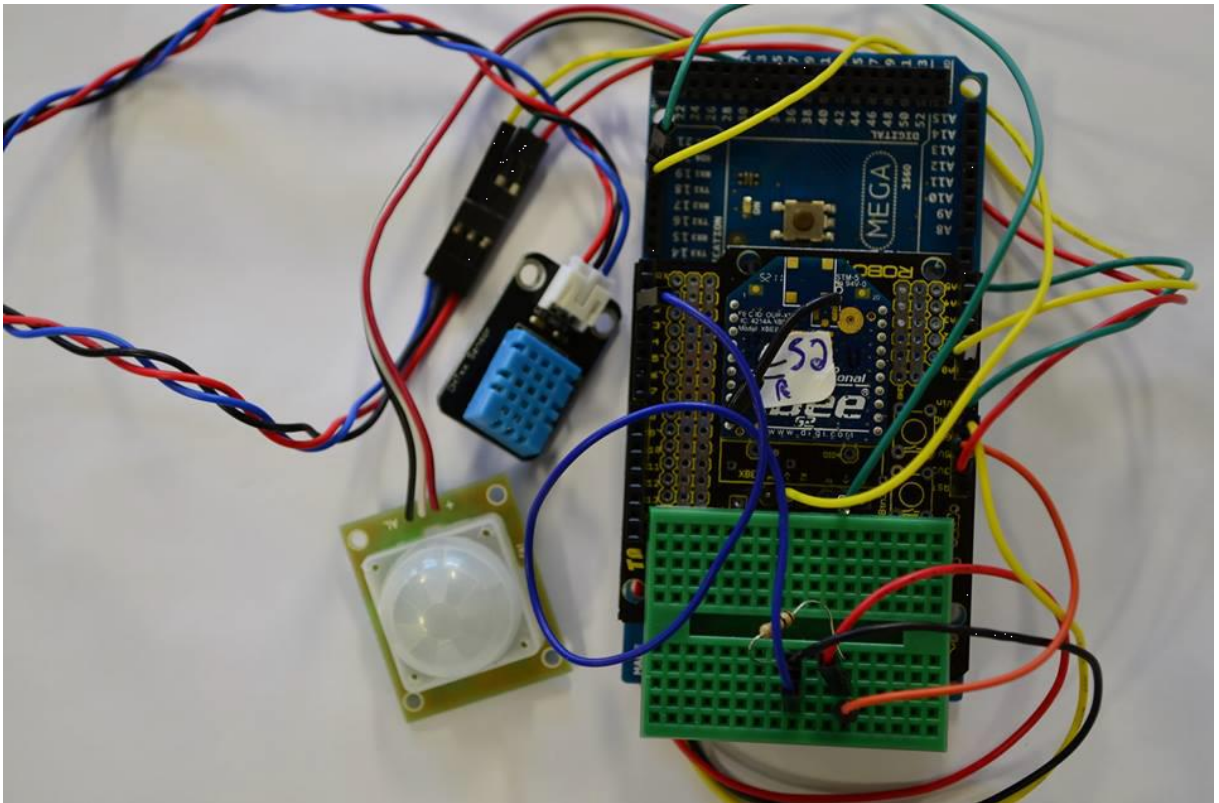
6.1.1 Elementos da RSSF

Esta aplicação foi desenvolvida para uma RSSF configurada com os seguintes elementos: nós sensores, nó atuador e nó coordenador. É importante salientar que esta RSSF é especificamente uma RSAF (Rede de Sensores e Atuadores sem Fio). Assim, o escopo desse exemplo de uso desta metodologia aplica-se a esse tipo de RSSFs.

6.1.1.1 Nós sensores

Esta RSSF é composta por dois nós sensores (Figura 11) que têm como função monitorar os laboratórios e salas onde estão dispostos, coletar os dados de temperatura destes ambientes e enviar tais dados para o nó atuador. A coleta da temperatura de uma determinada sala é feita no intervalo de tempo determinado pela aplicação.

Figura 11 – Nó sensor

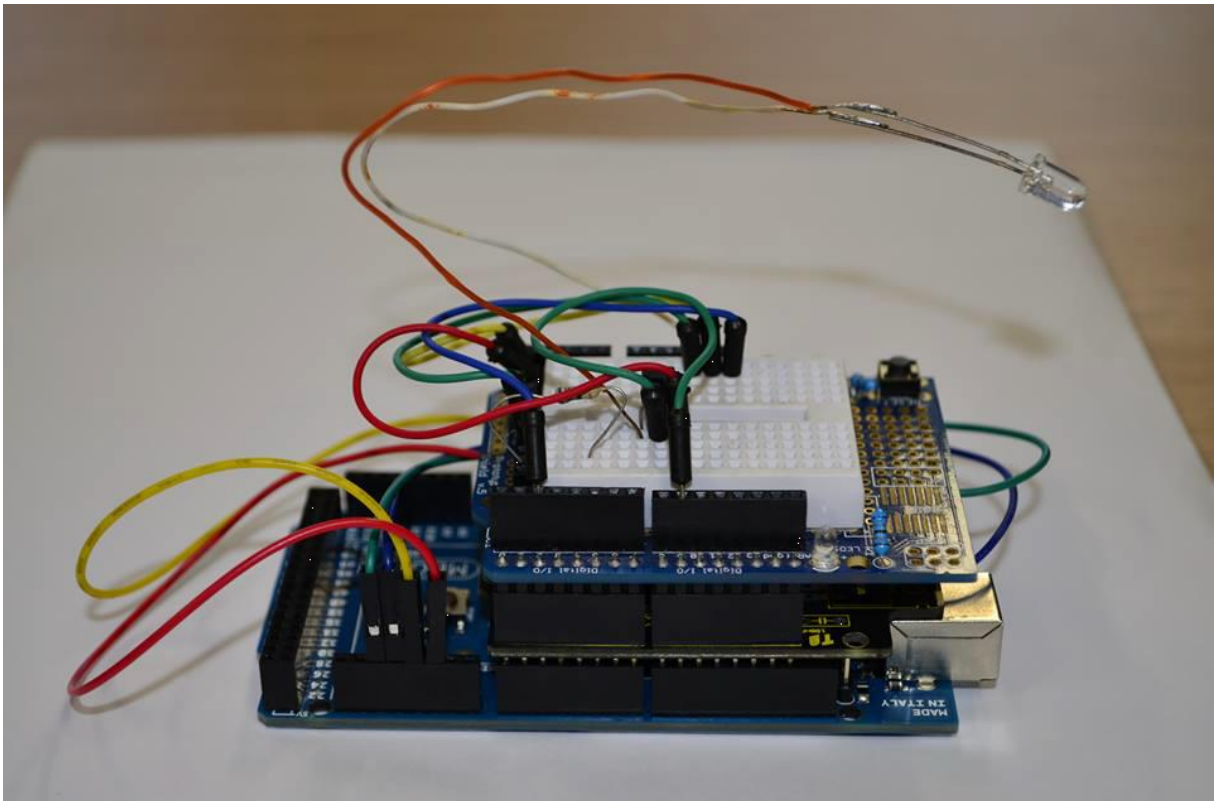


Fonte: Cruz, 2013

6.1.1.2 Nó atuador

O nó atuador (Figura 12) desta RSSF recebe os dados coletados pelos nós sensores, calcula a temperatura média do ambiente em um determinado intervalo de tempo, e regula (se necessário) a temperatura do condicionador de ar da sala, de acordo com a especificação da aplicação, neste caso entre 20 e 23° C, seguindo a norma de conforto térmico da Norma Regulamentadora N° 17 (NR-17) (CRUZ, 2013).

Figura 12 – Nó atuador



Fonte: Cruz, 2013

6.1.1.3 Nó coordenador

O nó coordenador desta RSSF tem como função receber do nó atuador, todos os dados coletados pelos nós sensores e enviar tais dados para um servidor, onde os mesmos serão armazenados. Neste estudo de caso, em vez do nó coordenador se comunicar com um servidor, ele se comunica com a Nuvem.

6.2 Serviços da Nuvem

Neste exemplo de uso da metodologia, foi definido que os serviços oferecidos pela Nuvem a serem utilizados pela a aplicação são: processamento e armazenamento de dados. Dessa forma, em vez de o nó atuador receber os dados dos nós sensores e o mesmo já efetuar a mudança de temperatura, o nó atuador envia tais dados para o nó coordenador, e este por sua vez envia os dados para a Nuvem. Ao chegarem à Nuvem, os dados de temperatura são armazenados em um Banco de Dados e processados de acordo com a especificação da

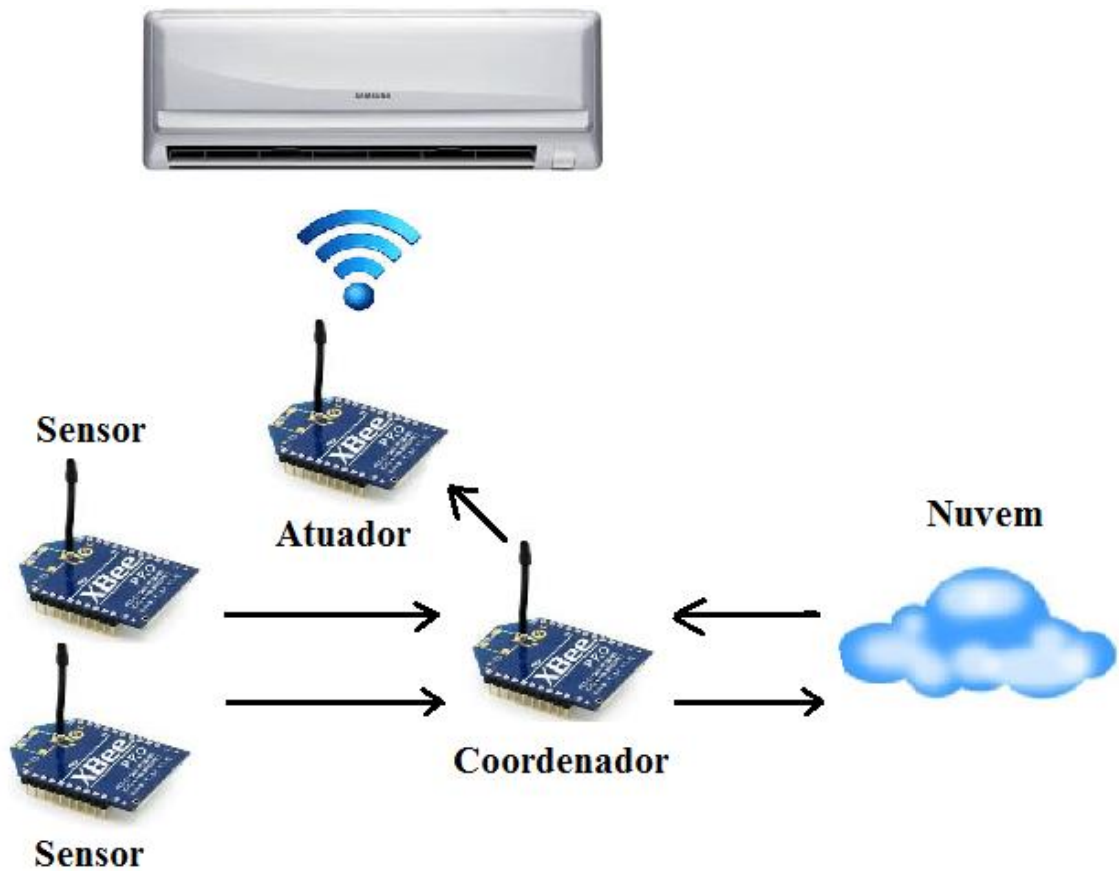
aplicação. Logo após o processamento, os dados processados são enviados de volta para o nó coordenador e em seguida, tais dados são repassados para o nó atuador para realizar a mudança de temperatura (se necessário) de acordo com a definição da aplicação. Assim, se tem um histórico dos valores coletados pelos sensores que podem fornecer uma análise posteriormente do contexto da universidade.

Após a definição dos serviços da Nuvem que serão utilizados pela RSSF, decidimos utilizar a instância da Nuvem da *Amazon EC2* com base em Leitão *et al.*, 2012 que utilizou tal instância em seu estudo de caso e devido esta instância ser disponibilizada gratuitamente. Assim, instalamos e configuramos o Banco de Dados *NoSQL MongoDB* nessa instância para armazenar os dados da RSSF. A escolha do Banco de Dados *MongoDB* foi feita devido a este banco de dados ser não relacional, o que atende as necessidades desta aplicação. Por fim, decidimos utilizar o compilador da linguagem de programação *Python* para executar o *script* do *socket* servidor na Nuvem, devido ao conhecimento desta linguagem.

6.3 Interação da RSSF com a Nuvem

Primeiramente, para integrar esta RSSF com a Nuvem, antes de configurar o nó coordenador, modificamos um pouco a forma como os elementos da RSSF interagem. Assim, em vez dos nós sensores enviarem os dados de temperatura para o nó atuador e este enviar para o nó coordenador, os nós sensores enviam os dados diretamente para o nó coordenador. O nó atuador, por sua vez, tem agora a função apenas de receber os dados de temperatura já processados e efetuar a mudança de temperatura do condicionador de ar. Com isso, temos uma nova topologia da RSSF, como mostrado na figura 13.

Figura 13 – Nova topologia da RSSF



Fonte: Elaborada pela autora.

6.3.1 Configuração do nó coordenador

Dessa forma, configuramos o nó coordenador da seguinte forma:

1. Receber os dados de temperatura que são enviados pelo nós sensores;
2. Enviar tais dados para a instância na Nuvem;
3. Receber os dados processados que a instância na Nuvem retornar;
4. Enviar os dados processados para o nó atuador.

6.3.2 Comunicação da RSSF com a Nuvem

Neste exemplo de uso da metodologia, a comunicação entre a RSSF com a Nuvem foi feita por meio da implementação de dois *sockets*, um servidor e um cliente. O *socket* servidor ficou na instância da Nuvem, preparado para receber os dados, que por sua

vez foram enviados através do *socket* cliente que ficou em uma máquina física onde o nó coordenador estava conectado também.

Dessa forma, para que o nó coordenador realize as ações definidas na configuração, implementamos três *scripts* na linguagem *Python*: o *SerialRead*, o *SocketServidor* e o *SocketCliente* (APÊNDICE A). Além disso, para que o nó coordenador se comunique com os demais nós da rede e com a Nuvem utilizando estes *scripts*, precisamos utilizar uma máquina física (computador) onde o mesmo está conectado.

Assim, o *script SerialRead* foi executado na máquina física onde o nó coordenador estava conectado e foi responsável por receber os dados de temperatura coletados pela RSSF, através do nó coordenador e enviar tais dados para o *SocketCliente*.

O *script SocketCliente* também foi executado na máquina física e foi responsável por receber os dados de temperaturas do *SerialRead* e enviá-los para a Nuvem que receberá tais dados através do *SocketServidor*.

O *script SocketServidor* por sua vez, foi executado na instância da Nuvem *Amazon EC2* já configurada e foi responsável por receber os dados de temperatura enviados pelo *SocketCliente*, armazená-los no Banco de Dados *MongoDB* e processá-los fazendo o cálculo da média da temperatura do ambiente. Após o cálculo, o *SocketServidor* enviou o valor da temperatura ambiente para o nó coordenador que por sua vez, enviou para o nó atuador responsável por mudar a temperatura.

6.3.3 Dados coletados

Durante a realização do estudo de caso, ao integrarmos a RSSF com a Nuvem obtivemos onze mil dados de temperatura coletados pelos nós sensores. Os dados foram enviados para a Nuvem, onde foram armazenados no Banco de Dados e em seguida, foram processados (média dos valores recebidos). A coleta dos dados foi realizada durante 16 horas e a frequência de coleta dos dados de temperatura foi de 0,5 segundo, ou seja, a cada 0,5 segundo, os nós sensores coletavam e enviavam dados para o nó coordenador. A Nuvem levou apenas 0,01 segundo para calcular cada média de temperatura e o nó atuador recebeu 960 valores de temperatura média do ambiente.

Nesta seção, foi mostrado um estudo de caso como demonstração de uso da metodologia proposta. Na próxima seção será apresentada a análise feita a partir do estudo de caso.

7 ANÁLISE DO EXEMPLO DE USO

7.1 Benefícios para a RSSF

Ao concluirmos o estudo de caso, pudemos observar que os entre os benefícios da integração desta RSSF com a Nuvem estão: a economia de processamento de dados e a redução das transmissões de mensagens no nó atuador. Entretanto, o principal benefício foi o armazenamento dos dados coletados.

Dessa forma, ao definir que o processamento dos dados da RSSF seria feito na Nuvem, em vez do nó atuador ter a função de receber, processar todos os dados coletados e executar uma ação, ele agora apenas recebe o resultado dos dados processados e executa a ação necessária. Reduzindo assim, o processamento na RSSF.

Assim, de acordo com os resultados obtidos, foram analisados que sem a integração da RSSF com a Nuvem, o nó atuador teria recebido e transmitido onze mil dados de temperatura. Além disso, ele teria executado o cálculo da temperatura média de todos esses dados e ainda teria executado todas as ações definidas pela aplicação. No entanto, como já foi mostrado, durante o estudo de caso, o nó atuador recebeu apenas novecentos e sessenta dados, os quais já tinham sido processados, e executou as ações com base nos resultados recebidos.

Portanto, podemos concluir que foi obtida uma economia considerável tanto de processamento, como transmissão de dados do nó atuador. Entretanto, é imprescindível ressaltar que a grande vantagem da integração desta aplicação com a Nuvem é o armazenamento, pois de acordo com a justificativa da aplicação, a mesma foi desenvolvida para ser aplicada a todo o campus da UFC em Quixadá, gerando assim uma quantidade imensa de dados coletados pelos sensores. Ao integrarmos esta RSSF à Nuvem não será preciso preocupar-se com o tamanho ou quantidade de dados, devido à elasticidade do ambiente da Nuvem.

7.2 Benefícios do uso da Nuvem

Geralmente, dependendo da quantidade de sensores e/ou da frequência da coleta de dados efetuada pelas mesmas, torna-se necessário utilizar uma infraestrutura capaz de armazenar e processar esse grande volume de dados, garantindo um bom funcionamento da RSSF. Nesse caso, a infraestrutura oferecida pela Nuvem surge como melhor alternativa para suprir as necessidades que podem ser encontradas nas RSSFs, pois além de suprir

necessidades básicas como armazenamento e processamento de dados, a Nuvem também pode realizar análises mais complexas sobre os dados coletados sem precisar de um enorme consumo de recursos computacionais.

No estudo de caso realizado, apesar de não terem sido utilizados muitos nós sensores, a rede foi configurada para coletar e enviar dados para a Nuvem frequentemente (a cada 0,5 segundos). Assim, conseguimos obter uma quantidade significativa de dados (11.000 temperaturas) armazenados e processados pela Nuvem, de maneira satisfatória, mostrando o quão oportuna a integração de RSSFs e Nuvem tem se tornado. Assim, com base nos resultados obtidos neste estudo de caso, pudemos reforçar a justificativa do uso da Nuvem por RSSFs.

Nesta seção, foi mostrada a análise realizada a partir do estudo de caso realizado neste trabalho. Na próxima seção são apresentadas as considerações finais acerca do trabalho realizado.

8 CONSIDERAÇÕES FINAIS

Nesta seção são apresentadas algumas considerações finais acerca do desenvolvimento deste trabalho.

8.1 Conclusão

Este trabalho propôs a elaboração de uma metodologia para a integração de RSSFs e Computação em Nuvem, baseado no recente fato da Computação em Nuvem estar se tornando uma alternativa viável para a solução das limitações encontradas nas RSSFs. Assim, foi definida uma metodologia para integrar RSSFs e Nuvem independente de plataforma, ambiente de desenvolvimento ou linguagem de programação, com a finalidade de que esta metodologia possa ser utilizada por qualquer RSSF ou serviço oferecido pela Computação em Nuvem.

Por fim, a metodologia definida neste trabalho foi validada por meio da realização de um estudo de caso, no qual a metodologia foi aplicada a uma aplicação específica para uma RSSF utilizando os serviços da Computação em Nuvem necessários. Ao realizar este estudo de caso, foi possível mostrar alguns dos benefícios que a integração de RSSFs e Computação em Nuvem pode proporcionar. Com isso, pode-se concluir que este trabalho alcançou seus objetivos de forma satisfatória.

8.2 Trabalhos Futuros

Durante o desenvolvimento da metodologia proposta, foram identificadas algumas importantes possibilidades de melhorias que podem ser aplicadas a este trabalho, sendo elas:

- Aplicar a metodologia a outras aplicações simultâneas para RSSFs para uma melhor validação da mesma;
- Testar a metodologia aplicando a mesma a aplicações para RSSFs maiores e que utilizem mais armazenamento e processamento da Nuvem e/ou outros serviços oferecidos por ela;
- Avaliar mais detalhadamente os benefícios que o uso desta metodologia proposta traz para a integração de RSSFs e Computação em Nuvem;
- Aplicar a metodologia a aplicações compartilhadas para RSSFs.

REFERÊNCIAS

- ALAMRI, A. *et al.* **A survey on sensor-cloud: architecture, applications, and approaches.** International Journal of Distributed Sensor Networks, Saudi Arabia, Volume 2013, Article ID 917923, 18 pages, 2013.
- ARDUINO. **Arduino.** [site corporativo]. Disponível em: < <http://arduino.cc/> >. Acesso em: 09 de dezembro de 2013.
- CRUZ, M. M.. **Climaduino: Sistema Embarcado de Climatização para Edifícios Inteligentes.** Trabalho de Conclusão de Curso (Tecnólogo em Redes de Computadores), Universidade Federal do Ceará, Quixadá, 2013.
- CUNHA, T. F.. **Controle centralizado de equipamentos de ar condicionado via rede sem fio ZigBee.** Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Instituto Federal de Santa Catarina, 2013, São José.
- DHARMASIRI, H. M. L.; GOONETILLAKE, M. D. J. S.. **A Federated Approach on Heterogeneous NoSQL Data Stores.** *In:* International Conference on Advances in ICT for Emerging Regions (ICTer), p. 234 – 239, Colombo, Sri Lanka, 2013.
- DIALLO, O.; RODRIGUES, J. J. P.; SENE, M.. **Real-time data management on wireless sensor networks: A survey.** Elsevier: Journal of Network and Computer Applications, v. 35, p. 1013-1021, dezembro, 2011.
- DIGI INTERNATIONAL. **XBee.** [site corporativo]. Disponível em: <<http://www.digi.com/xbee/>>. Acesso em: 15 de maio de 2014.
- ESCHNER, R. H.. **Sistema de automação residencial baseado em sensores ZigBee.** Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.
- EVANS, B.. **Beginning arduino programming.** Apress, 2011.
- LEITÃO, L. M.; SAMPAIO, A.; HOLANDA FILHO, R.. **NaturalCloud: um framework para integração de Rede de Sensores sem Fio na Nuvem.** *In:* X Workshop em Clouds, Grids e Aplicações – WCGA, 2012, Ouro Preto. Anais X Workshop em Clouds e Aplicações, 2012. p. 44-55.
- LOUREIRO, A. A. F. *et al.* **Redes de Sensores sem Fio.** *In:* Simpósio Brasileiro de Redes de Computadores, 21., 2003, Natal.
- LOUREIRO, A. A. F.. **Redes de Sensores sem Fio.** *In:* Seminário: Grandes Desafios da Pesquisa em Computação para o período 2006-2016, 2006, São Paulo.
- McROBERTS, M.. **Arduino básico.** São Paulo: Novatec, 2011.
- MELLIS, D. A.. **Arduino - Introduction.** Disponível em: <<http://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 09 de dezembro de 2013.

MONGODB. **MongoDB**. [site corporativo]. Disponível em: < <http://www.mongodb.org/>>. Acesso em 19 de maio de 2014.

MONGODB. **PyMongo 2.7rc0 Documentation**. Disponível em: <<http://www.mongodb.org/>>. Acesso em 19 de maio de 2014.

NETO, O. S.. **PythonBrasil**. Disponível em: <<http://www.python.org.br/wiki>>. Acesso em 19 de maio de 2014.

POTDAR, V.; SHARIF, A.; CHANG, E.. **Wireless Sensor Networks: A Survey**. *In:* International Conference on Advanced Information Networking and Applications Workshops, 2009, Bradford, United Kingdom.

PYTHON. **Python**. [site corporativo]. Disponível em: < <https://www.python.org/>>. Acesso em 19 de maio de 2014.

ROCHA, A. R.. **Um serviço de clusterização semântica para Redes de Sensores sem Fio autonômicas**. Tese de Doutorado (Doutorado em Engenharia de Teleinformática), Universidade Federal do Ceará, Fortaleza, 2012.

ROCHA, A. R.; PIRMEZ, L.; DELICATO, F. C.; LEMOS, E.; SANTOS, I.; GOMES, D. G.; SOUZA, J. N. S. **WSNs clustering based on semantic neighborhood relationships**. Elsevier: Computer Networks, v. 56, n.5, p. 1627-1645, março, 2012.

RUIZ, L. B.; CORREIA, L. H. A.; VIEIRA, L. F.; MACEDO, D. F.; NAKAMURA, E. F.; FIGUEIREDO, C. M. S.; VIEIRA, M. A.; BECHELANE, E. H.; LOUREIRO, A. F.; NOGUEIRA, J. M.. **Arquiteturas para Redes de Sensores Sem Fio**. *In:* Minicursos do 22º Simpósio Brasileiro de Redes de Computadores. Gramado, Rio Grande do Sul, 2004, p.167-218.

SALEIRO, M.; EY, E.. **ZigBee: uma abordagem prática**. Disponível em <http://lusorobotica.com/ficheiros/Introducao_ao_Zigbee_-_por_msaleiro.pdf> Acesso em: 09 de dezembro de 2013.

SOUSA, F. R. C.; MOREIRA, L. O.; MACÊDO, J. A. de ; MACHADO, J. C.. **Gerenciamento de dados em Nuvem: conceitos, sistemas e desafios**. *In:* Simpósio Brasileiro de Banco de Dados, 2010, Belo Horizonte.

SOUSA, F. R. C.; MOREIRA, L. O.; MACHADO, J. C.. **Computação em Nuvem: conceitos, tecnologias, aplicações e desafios**. *In:* Escola Regional de Computação Ceará, Maranhão, Piauí, 2009, Parnaíba.

YICK, J.; MUKHERJEE, B.; GHOSAL, D.. **Wireless sensor network survey**. Elsevier: Computer Networks, v. 52, p. 2292-2330, abril, 2008.

APÊNDICE A – CÓDIGO DOS *SCRIPTS* UTILIZADOS NA METODOLOGIA***SCRIPT “SERIALREAD”***

```
#!/usr/bin/env python
# -*- utf-8 -*-

from FilterData import FilterData
from threading import Thread
import serial
import time
import datetime

class SerialRead(Thread):

    def __init__(self, porta, baudRate):
        Thread.__init__(self)
        # open serial port
        try:
            self.serial_port = serial.Serial(porta, int(baudRate), timeout=2)
        except serial.SerialException as e:
            print("nao foi possivel abrir a porta serial '{}': {}".format(porta, e))

    def run(self):

        contador = 0
        #le os dados da porta serial
        while True:
            a = []
            for c in self.serial_port.readline():
                a.append(c)
            a = ''.join(a)

            # wait for new data after each line
            timeout = time.time() + 0.1
            while not self.serial_port.inWaiting() and timeout > time.time():
                pass
            if a != "":
                FilterData.setData(a)
                #print a

            a = []
            time.sleep(1)
            #dados.setData()

ler = SerialRead('/dev/ttyUSB0', 115200)
ler.start()
```


SCRIPT "SOCKETSERVIDOR"

```

#!/usr/bin/env python

import socket
from pymongo import Connection

host = ''
port = 5678
temperatura = 0.0
temPresenca = 0
media = 0.0
ligado = 0

addr = (host, port)
serv_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serv_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
serv_socket.bind(addr)
serv_socket.listen(10)

while(1):
    print 'aguardando conexao'
    conn, cliente = serv_socket.accept()
    print 'conectado'
    print "aguardando mensagem"
    recebe = conn.recv(1024)

    print "mensagem recebida: "+ recebe
    con = Connection('127.0.0.1', 27017, safe=True)
    db = con['sensores']
    users = db.users
    resultado = db.users.find().sort([["_id", -1]].limit(30)

    for p in resultado:
        media = media + x['t']
        if p['p'] > 0:
            temPresenca = 1

    if temPresenca == 1:
        temPresenca = 0.0
        temperatura = (media / 30)
        media = 0.0
        print temperatura

        if ligado == 0:
            print "liga o aparelho"
            conn.send("1a")
            ligado = 1

        if temperatura < 20.0:
            print "aumenta"
            conn.send("1c")
        elif temperatura > 23.0:
            print "diminui"
            conn.send("1d")
    else:
        print "Sem presenca: desliga"
        conn.send("1b")
        ligado = 0

serv_socket.close()

```

SCRIPT "SOCKETCLIENTE"

```
#!/usr/bin/env python

import socket
import serial
import time

ip = '54.186.82.197'
port = 5678

def escrever_porta(string):
    try:

        Obj_porta = serial.Serial('/dev/ttyUSB0', 115200)
        Obj_porta.writelines(string)
        Obj_porta.close()
        print "ok"
        time.sleep(60)

    except serial.SerialException:
        print"ERRO: Verifique se ha algum dispositivo conectado na porta!"

while(1):
    addr = ((ip,port))
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    try:
        client_socket.connect(addr)
        client_socket.send("a")
        print 'mensagem enviada'
        recebido = client_socket.recv(1024)
        print recebido
        escrever_porta(recebido)
        client_socket.close()

    except socket.error as e:
        print("Cannot connect to ")
        print(ip," on port: ", port)
        print(e)
```

ANEXO A – EXTENSÃO DO CÓDIGO DA APLICAÇÃO UTILIZADA COMO EXEMPLO DE USO DA METODOLOGIA

Fonte: Cruz, 2013

CÓDIGO DOS SENSORES

```
#include <EasyTransfer.h>

#include <dht11.h>

dht11 sensor;
EasyTransfer ET;

struct PACOTE{
    float temperatura;
    int presenca;
};
PACOTE pacote;

void setup(){

    pinMode(3, INPUT);
    Serial.begin(115200);
    Serial1.begin(115200); //serial xbee

    ET.begin(details(pacote), &Serial1);
}

void loop(){

    int presenca = digitalRead(3);
    if(presenca > 0){
        pacote.presenca = 0;
    }
    else{
        pacote.presenca = 1000;
    }

    int chk = sensor.read(2);
    Serial.println(digitalRead(3));

    pacote.temperatura = (float)sensor.temperature;

    ET.sendData();
}
```

CÓDIGO DO ATUADOR

```
char texto;

void setup(){

  Serial.begin(115200);
  Serial1.begin(115200);

}

void loop(){

  while(Serial1.available() > 0){

    if (Serial1.read() == 49){
      Serial.println("mensagem pra mim!");
      acao(Serial1.read());
    }

  }

}

void acao(char comando){

  texto = comando;
  if(texto == 'a'){
    Serial.println("liga");
  }
  else if(texto == 'b'){
    Serial.println("desliga");
  }
  else if(texto == 'c'){
    Serial.println("aumenta");
  }
  else if(texto == 'd'){
    Serial.println("diminui");
  }
}
```