

**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS  
FATEC PROFESSOR JESSEN VIDAL**

**JEFFERSON RIBEIRO UCHÔAS**

**UTILIZAÇÃO DA INTERNET DAS COISAS PARA O  
DESENVOLVIMENTO DE UM SISTEMA DE MEDIÇÃO DE  
ÍNDICES METEOROLÓGICOS**

São José dos Campos  
2013

**JEFFERSON RIBEIRO UCHÔAS**

**UTILIZAÇÃO DA INTERNET DAS COISAS PARA O  
DESENVOLVIMENTO DE UM SISTEMA DE MEDIÇÃO DE  
ÍNDICES METEOROLÓGICOS**

Trabalho de Graduação apresentado à  
Faculdade de Tecnologia de São José dos  
Campos, como parte dos requisitos  
necessários para a obtenção do título de  
Tecnólogo em Banco de Dados.

**Orientador: Me. Giuliano Araujo Bertoti**

São José dos Campos  
2013

## **Dados Internacionais de Catalogação-na-Publicação (CIP)**

### **Divisão de Informação e Documentação**

RIBEIRO UCHÔAS, Jefferson  
Utilização da Internet das Coisas para o desenvolvimento de um sistema de medição de índices meteorológicos.

São José dos Campos, 2013.

70f.

Trabalho de Graduação – Curso de Tecnologia em Banco de Dados,  
FATEC de São José dos Campos: Professor Jessen Vidal, 2013.

Orientador: Me. Giuliano Araujo Bertoti.

1. Áreas de conhecimento. I. Faculdade de Tecnologia. FATEC de São José dos Campos: Professor Jessen Vidal. Divisão de Informação e Documentação. II. Título

## **REFERÊNCIA BIBLIOGRÁFICA –**

RIBEIRO UCHÔAS, Jefferson. **Utilização da Internet das Coisas para o desenvolvimento de um sistema de medição de índices meteorológicos**. 2013. 70f. Trabalho de Graduação - FATEC de São José dos Campos: Professor Jessen Vidal.

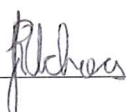
## **CESSÃO DE DIREITOS –**

NOME DO AUTOR: Jefferson Ribeiro Uchôas

TÍTULO DO TRABALHO: Utilização da Internet das Coisas para o desenvolvimento de um sistema de medição de índices meteorológicos

TIPO DO TRABALHO/ANO: Trabalho de Graduação / 2013.

É concedida à FATEC de São José dos Campos: Professor Jessen Vidal permissão para reproduzir cópias deste Trabalho e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.



Jefferson Ribeiro Uchôas

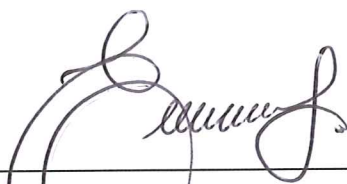
Rua Rubens Leitão Filho, 260, Jardim Ismênia

CEP 12221-020 – São José dos Campos – São Paulo

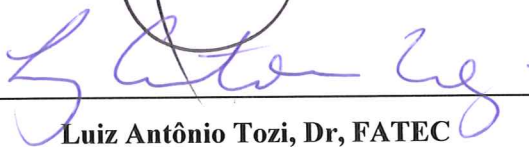
**JEFFERSON RIBEIRO UCHÔAS**

**UTILIZAÇÃO DA INTERNET DAS COISAS PARA O  
DESENVOLVIMENTO DE UM SISTEMA DE MEDIÇÃO DE  
ÍNDICES METEOROLÓGICOS**

Trabalho de Graduação apresentado à  
Faculdade de Tecnologia de São José dos  
Campos, como parte dos requisitos  
necessários para a obtenção do título de  
Tecnólogo em Banco de Dados.



**Eliane Nikoluk Scachetti, Dr<sup>a</sup>, Polícia Militar do Estado de São Paulo**



**Luiz Antônio Tozi, Dr, FATEC**



**Denise Alves de Barros, Tecnóloga, FATEC**



**Giuliano Araujo Bertoti, Me, FATEC**

17/12/2013

**DATA DA APROVAÇÃO**

A DEUS que, pela Sua infinita misericórdia, me deu forças para continuar e me abriu o entendimento e a Jesus Cristo, autor e consumador da minha fé.

## **AGRADECIMENTOS**

Agradeço ao professor e orientador Giuliano Bertoti pelo apoio e encorajamento contínuos para a conclusão deste Trabalho, aos demais professores pelos conhecimentos transmitidos, à minha esposa e filha que muito me apoiaram e tiveram grande paciência nos momentos nos quais tive que me entregar aos estudos, e à minha mãe que jamais duvidou desta vitória.

“Em todo trabalho há proveito; meras palavras,  
porém, só encaminham para a penúria.”

Provérbios 14:23

## RESUMO

Grande parte dos moradores de cidades brasileiras vive em moradias precárias nas áreas de grande fragilidade ambiental, sem nenhuma infraestrutura urbana e em situação de extrema vulnerabilidade a desastres que decorrem de inundações, enxurradas ou deslizamentos de terra. Além disso, as mudanças climáticas colocam em risco a vida humana cada vez com mais frequência. Tal cenário aponta a necessidade de adoção de medidas preventivas que permitam uma correta percepção dos fatores climáticos que podem colocar a vida das pessoas em risco visando prever e controlar os efeitos negativos de eventos adversos sobre a população, os bens e serviços e o meio ambiente. O objetivo deste trabalho é desenvolver e implementar um sistema de medição de índices meteorológicos capaz de enviar alertas antecipados sobre a possibilidade de ocorrência de desastres naturais. O dispositivo Arduino foi utilizado para coletar e enviar os dados do ambiente para o serviço de web Xively de Internet das Coisas e também para um web service com arquitetura RESTful. O web service foi desenvolvido em linguagem Java enquanto a aplicação para interação com o usuário foi desenvolvida em HTML e CSS. A fim de validar a confiabilidade do dispositivo, as variáveis ambientais coletadas foram comparadas com os dados oficiais da Defesa Civil de São José dos Campos. Por se tratar de plataforma aberta e de baixo custo para a construção, o sistema proporciona livre acesso a qualquer pessoa ou instituição interessada em utilizá-lo ou mesmo modificá-lo.

**Palavras-Chave:** Arduino, Internet das Coisas, sistemas de alerta, deslizamento de encostas, meio-ambiente, desastres naturais, meteorologia.



## ABSTRACT

Many residents of Brazilian cities live in substandard housing, located in high environmental fragile areas, without urban infrastructure and in extremely vulnerable situations due to disasters arising from floods, mudslides and landslides. In addition, climate changes represent a risk to human lives in such conditions. The scenario described shows the need to adopt preventive actions that allow a correct perception of the climate factors, to prevent and control the negative effects of adverse events on the population, goods, services and the environment. The present work aims to develop and implement a system for measuring meteorological index in order to send early warnings on the possibility of natural disasters. Arduino technology has been applied to collect and send environment data to the Xively Internet of Things web service and also to a RESTful based web service. The web service is developed in Java, whereas the web application which enables user interaction is developed in HTML and CSS. In order to validate the reliability of the device implemented, measurements data have been collected and assessed in comparison to the official data of São José dos Campos Civil Defense. Being an open platform and low cost building system, it provides free access to people or institutions interested in using it or even modifying it.

**Keywords:** Arduino, Internet of Things, alert systems, landslides, environment, natural disasters, meteorology.

## LISTA DE FIGURAS

Figura 1 – Pluviômetros nos municípios do Estado de São Paulo	16
Figura 2 – Arduino <i>Ethernet</i>	21
Figura 3 – Objeto JSON	24
Figura 4 – Dispositivos conectados ao Xively	25
Figura 5 – Crescimento de ativações de dispositivos Android	28
Figura 6 – Arquitetura global	31
Figura 7 – Sensor pluviométrico ( <i>rain gauge</i> )	33
Figura 8 – Esquema do circuito do sensor pluviométrico ( <i>rain gauge</i> )	34
Figura 9 – Esquema do circuito do sensor anemômetro ( <i>anemometer</i> )	35
Figura 10 – Sensores anemômetro ( <i>anemometer</i> ) e cata-vento ( <i>wind vane</i> )	35
Figura 11 – Esquema do circuito do sensor cata-vento ( <i>wind vane</i> )	36
Figura 12 – Sensor DHT22 (RHT03)	37
Figura 13 – Esquema do circuito do sensor DHT22 (RHT03)	37
Figura 14 – Placa Proto Shield	38
Figura 15 – Ponto de Dados da aplicação <i>Weather Station</i>	40
Figura 16 – Objeto BSon da aplicação <i>Weather Station</i>	41
Figura 17 – Canal de dados <i>temperature</i> da <i>Weather Station</i>	42
Figura 18 – Comunicação Dispositivos/OpenShift	43
Figura 19 – Proto Shield e sua conexão com a placa Arduino Ethernet	46
Figura 20 – Módulo físico da <i>Weather Station</i> montado na caixa hermética	47
Figura 21 – <i>Weather Station</i> montada	47
Figura 22 – Informações sobre vento, umidade relativa do ar e chuva	50
Figura 23 – Informações sobre temperatura	50
Figura 24 – Informações sobre temperatura máxima e mínima	51
Figura 25 – Informações sobre chuva acumulada	51
Figura 26 – Acesso a partir de uma Smart TV e de um Notebook	52
Figura 27 – Acesso a partir de um Tablet e de um Smartphone	52
Figura 28 – Telas da aplicação para dispositivos Android	53
Figura 29 – Dispositivos registrados para o recebimento de alertas	54
Figura 30 – Plataforma de Coleta de Dados	55
Figura 31 – Metadados da PCD de São José dos Campos	56
Figura 32 – Meteograma de precipitação e temperatura	57
Figura 33 – Dados de precipitação registrados pela <i>Weather Station</i>	58
Figura 34 – Dados de temperatura registrados pela <i>Weather Station</i>	58
Figura 35 – Dados registrados pela PCD	59
Figura 36 – Dados registrados pela <i>Weather Station</i>	59
Figura 37 – Precipitações do mês de outubro de 2013 - Defesa Civil Estadual	60
Figura 38 – Precipitações do mês de outubro de 2013 - <i>Weather Station</i>	60
Figura 39 – Temperatura do mês de outubro de 2013 - Defesa Civil Estadual	61
Figura 40 – Temperaturas do mês de outubro de 2013 - <i>Weather Station</i>	61
Figura 41 – Distância entre a PCD e a <i>Weather Station</i>	63

## LISTA DE QUADROS

Quadro 1 – Relação entre inclinação da paleta, resistência e voltagem	36
Quadro 2 – Endereços para requisição ao <i>web service</i>	44
Quadro 3 – Custo do <i>hardware</i> da <i>Weather Station</i>	48

## SUMÁRIO

1 – INTRODUÇÃO	14
1.1 – Motivação	14
1.2 – Objetivos	17
1.2.1 – Objetivo Geral	17
1.2.2 – Objetivos Específicos	17
1.3 – Metodologia	17
1.4 – Organização do Trabalho	18
2 – REVISÃO BIBLIOGRÁFICA	19
2.1 – Hidrometria	19
2.2 – Arduino	20
2.2.1 – Arduino Ethernet	21
2.3 – Sensores	21
2.3.1 – Sensor pluviométrico ( <i>Rain Gauge</i> )	22
2.3.2 – Sensor anemômetro ( <i>Anemometer</i> )	22
2.3.3 – Sensor cata-vento ( <i>Wind Vane</i> )	22
2.3.4 – Sensor de temperatura e umidade (DHT22)	22
2.4 – RESTful	22
2.5 – JSON	23
2.6 – Xively	24
2.7 – OpenShift	25
2.8 – MongoDB	26
2.9 – <i>Internet</i> das Coisas	26
2.10 – Cidades inteligentes	27
2.11 – Android	28
3 – DESENVOLVIMENTO	30
3.1 – Arquitetura Global	30
3.2 – Estrutura do <i>software</i> desenvolvido para o Arduino Ethernet	32
3.3 – Sensores	33
3.3.1 – Sensor pluviométrico ( <i>Rain Gauge</i> )	33
3.3.2 – Sensor anemômetro ( <i>Anemometer</i> )	34
3.3.3 – Sensor cata-vento ( <i>Wind Vane</i> )	35
3.3.4 – Sensor de temperatura e umidade (DHT22)	36
3.4 – Hardware para interligação dos sensores	37
3.5 – Comunicação do Arduino com a plataforma Xively	38
3.6 – Comunicação do Arduino com a plataforma OpenShift	39
3.7 – <i>Web Service</i> RESTful	39
3.8 – Objeto JSON	40
3.9 – Banco de dados não relacional MongoDB	40
3.10 – Comunicação dos dispositivos com a plataforma Xively	41
3.11 – Comunicação dos dispositivos com a plataforma OpenShift	43
3.12 – Aplicativo Android	44
4 – ANÁLISE DOS RESULTADOS	46
4.1 – Arquitetura de <i>Hardware</i> de Baixo Custo	46
4.2 – Arquitetura de <i>Software Open-Source</i>	49
4.3 – Aplicação para Interação com o Usuário	49
4.4 – Estudo de Casos	55

4.4.1 – Comparação com um meteograma	57
4.4.2 – Comparação com dados observados	59
4.4.3 – Comparação com uma série mensal de dados	60
4.5 – Observações	62
5 – CONSIDERAÇÕES FINAIS	64
5.1 – Contribuições e Conclusões	64
5.2 – Trabalhos futuros	65
REFERÊNCIAS	67

# **1 – INTRODUÇÃO**

## **1.1 – Motivação**

O relatório global anual de clima e catástrofes da Aon Benfield aponta que os incidentes naturais ocorridos em 2011 causaram impactos significativos em várias partes do mundo. Financeiramente foi o ano mais caro da história. Do mesmo modo que em todos esses locais, o cenário natural do Brasil vem sendo alterado. Tal país, que possui a característica de não ser afetado por desastres naturais, tem visto esta realidade sendo alterada a cada dia. Este cenário aponta a necessidade de emitir alertas sobre a ocorrência de eventuais desastres em locais que podem colocar a vida das pessoas em risco (CEMADEN, 2013).

Os assentamentos habitacionais da população de baixa renda revelam as marcas da exclusão social que comandou a história do Brasil. Grande parte dos moradores de cidades latino-americanas vive em moradias precárias nas áreas de grande fragilidade ambiental e sem nenhuma infraestrutura urbana, ou seja, em situação de extrema vulnerabilidade a desastres que decorrem de inundações, enxurradas ou deslizamentos de terra. As mudanças climáticas colocam em risco a vida humana cada vez com mais frequência (MINISTÉRIO DO MEIO AMBIENTE, 2013).

Quando uma tempestade severa produz chuvas e ventos fortes e se desloca sobre áreas urbanas com grande densidade demográfica, pode gerar inundações nas áreas próximas aos rios, bem como escorregamentos de encostas nas áreas consideradas de risco, destruindo e alagando casas, podendo levar pessoas a morte e deixar inúmeras famílias desabrigadas (MARCELINO, 2007).

É dever da União, dos Estados, do Distrito Federal e dos Municípios adotar as medidas necessárias à redução dos riscos de desastre. Entretanto, há necessidade que a sociedade civil também seja engajada a participar ativamente na solução de tal problemática (LEI FEDERAL Nº 12.608, de 10 de abril de 2012).

Inundações, alagamentos, enxurradas, deslizamentos, estiagens, secas e vendavais são os tipos de desastres naturais mais comuns registrados no Brasil. Nos centros urbanos, onde ocorre a maior concentração da população brasileira, os eventos que causam os maiores impactos são as inundações, as enxurradas e os deslizamentos de solo ou rocha (CEMADEN, 2013).

Apesar da maioria dos desastres naturais que ocorrem no Brasil não poderem ser evitados, bem se sabe que as atividades e intervenções do homem podem agravar ou minimizar seus impactos. Os desmatamentos, queimadas, assoreamento de rios, ocupação

desordenada, acúmulo de lixo, entre outras formas de desrespeito à natureza podem agravar sobremaneira os danos causados pelos desastres naturais. A fim de diminuir a possibilidade de tais danos podem ser adotadas medidas preventivas de cunho educativo como mapeamento das áreas de risco, análise de vulnerabilidade, etc. São medidas de baixo custo e fácil implantação que, apesar dos resultados a médio e longo prazo, permitem uma correta percepção do risco (MARCELINO, 2007).

A Lei Federal nº 12.608 institui a Política Nacional de Proteção e Defesa Civil e autoriza a criação de sistema de informações de monitoramento de desastres, em ambiente informatizado, atuando por meio de bases compartilhadas.

O Decreto Estadual nº 42.565 redefine o Plano Preventivo de Defesa Civil - PPDC específico para Escorregamentos nas Encostas da Serra do Mar, abrangendo os municípios de Cubatão, Guarujá, Santos e São Vicente, localizados na Baixada Santista, e Caraguatatuba, Ilhabela, São Sebastião e Ubatuba, localizados no Litoral Norte do Estado de São Paulo. Em seu anexo aponta a chuva como o principal agente deflagrador dos escorregamentos.

O PPDC, neste cenário, almeja possibilitar a previsão de condições de chuvas potenciais à ocorrência de escorregamentos, tanto naturais quanto induzidos, definindo valores acumulados de chuvas de 3 (três) dias, diferenciados para cada município, com base na condição pluviométrica que desencadeou escorregamentos verificados no início de 1988 e em anos anteriores. Tais valores são dados em mm / m<sup>2</sup>.

Foi adotado o Valor Acumulado de Chuvas (VAC) de 100 mm para os municípios de Santos, São Vicente e Guarujá e de 120 mm para Caraguatatuba, Ilhabela, São Sebastião e Ubatuba (DECRETO ESTADUAL Nº 42.565, de 1 de dezembro de 1997).

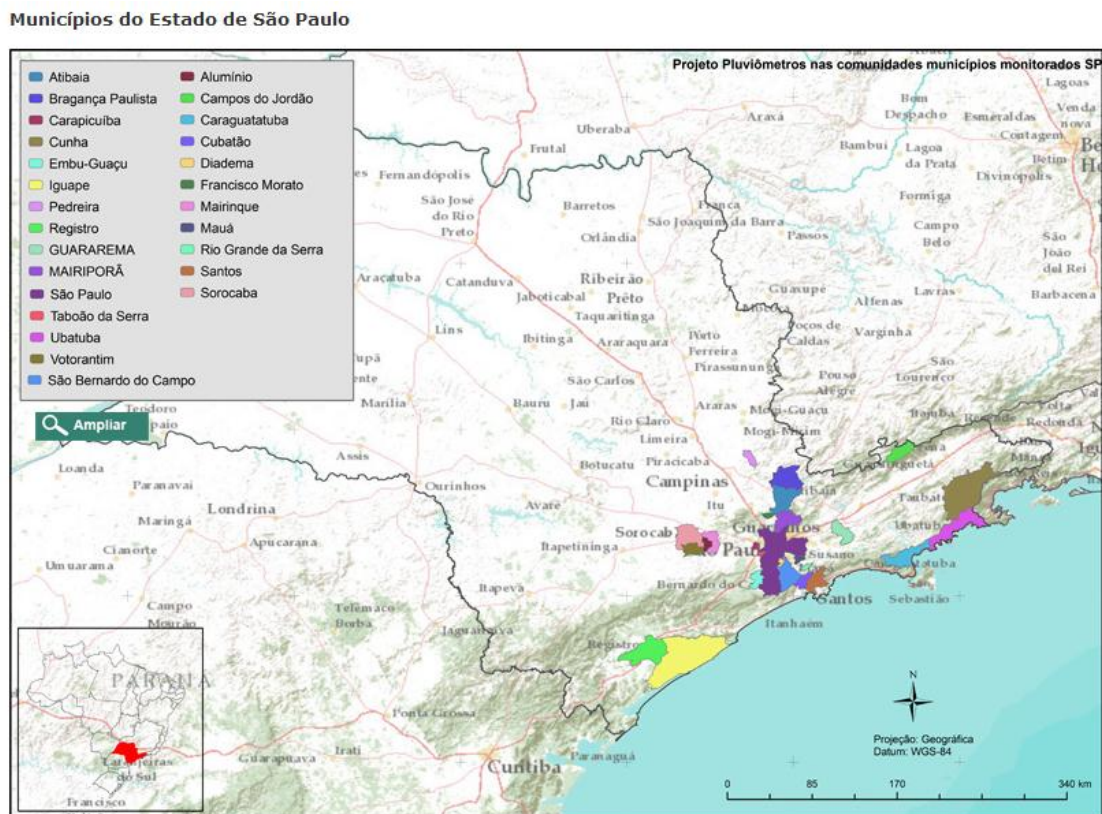
No ano de 2012 foi publicada a Resolução nº 18-610, a qual além de incluir novos municípios e regiões no PPDC, apontou que estudos desenvolvidos em diferentes países e também pelo Instituto de Pesquisas Tecnológicas do Estado de São Paulo (IPT) e Instituto Geológico (IG), no Brasil, reconheceram a importância de picos intensos de chuvas precedidos por um acumulado pluviométrico anterior como deflagradores de inundações e escorregamentos.

Foram então definidos novos valores acumulados de chuvas de três dias, diferenciados para cada região, a saber: além do Litoral Norte, já definido anteriormente com o VAC de 120 mm, para os municípios do Vale do Paraíba, exceto São Luiz do Paraitinga, definiu-se o valor de 100 mm e para o município de São Luiz do Paraitinga e municípios da Serra da Mantiqueira o valor de 80 mm.

O CEMADEN desenvolve o projeto "Pluviômetros nas Comunidades", visando envolver a população que vive em áreas de risco, distribuindo pluviômetros semiautomáticos, equipamentos utilizados para medir a quantidade de chuva, para serem instalados em áreas de risco e operados por equipes da comunidade local, especialmente treinadas. O projeto procura promover o engajamento e a conscientização dos moradores, bem como completar a rede de informações hidro meteorológicas do país.

Os municípios identificados com risco potencial de ocorrência de deslizamentos de encostas e enxurradas são alvo do monitoramento realizado pelo CEMADEN e CENAD. No estado de São Paulo são vinte e seis municípios com pluviômetros semiautomáticos instalados nas comunidades, como se pode observar na Figura 1:

**Figura 1 – Pluviômetros nos municípios do Estado de São Paulo**



Fonte: CEMADEN, 2013

Apesar de estar aumentando o número de municípios participantes do programa, ainda é custosa a implantação de pluviômetros automáticos e a confiabilidade dos dados medidos pelos pluviômetros manuais depende necessariamente da conscientização dos voluntários, visto que precisam realizar a leitura de forma constante e sem falhas.



## 1.2 – Objetivos

Nas subseções a seguir serão apresentados os objetivos deste trabalho.

### 1.2.1 – Objetivo Geral

O objetivo deste trabalho é utilizar a *Internet* das Coisas para o desenvolvimento de um sistema automático e de baixo custo para medição de índices meteorológicos.

### 1.2.2 – Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Monitorar os eventos hidrológicos potencialmente causadores de desastres;
- Produzir alertas antecipados sobre a possibilidade de ocorrência de desastres naturais;
- Integrar informações em um sistema capaz de subsidiar os órgãos públicos na previsão e no controle dos efeitos negativos de eventos adversos sobre a população, os bens e serviços e o meio ambiente.

## 1.3 – Metodologia

Esta seção descreve a metodologia e os materiais empregados no desenvolvimento do projeto.

Será desenvolvido um sistema de medição de índices meteorológicos composto por dois módulos: um módulo lógico e um módulo físico.

O módulo físico será composto por componentes físicos disponíveis para utilização no mercado, a saber, uma placa Arduino *Ethernet*, que pode ser considerada um computador com plataforma simples de *hardware* livre que possui um micro controlador ATmega328 e uma placa com suporte de entrada/saída de dados (MARCELINO, 2007), (SABO, 2011), (ARDUINO, 2013).

O Arduino Ethernet possui uma interface *Ethernet* embutida que possibilita a comunicação com um servidor *web* a ser utilizado para persistir as informações coletadas e também um módulo de cartão SD, que poderá ser utilizado para armazenamento local destes dados.

O dispositivo ainda contará com um conjunto de sensores que fará as medições do ambiente em uma série temporal predefinida.

Será montado um ambiente de teste local, uma vez que este projeto trata-se de um protótipo experimental. Os testes serão realizados durante trinta dias, de modo a proporcionar precisão na coleta de dados e confiabilidade dos resultados.

Para a programação do módulo Arduino será utilizada a linguagem *Processing*, própria da plataforma, a qual é baseada nas linguagens C e C++. O servidor *web* a ser utilizado será a plataforma Xively, com transferência de dados baseada no padrão RESTful e objetos do tipo JSON (FIELDING, 2000), (ARDUINO, 2013), (XIVELY, 2013), (PROCESSING, 2013).

Também será disponibilizado o acesso aos dados coletados através de aplicativo específico desenvolvido para a plataforma Android (GOOGLE PLAY, 2013).

#### **1.4 – Organização do Trabalho**

Este trabalho está organizado nos seguintes capítulos:

- a) Capítulo 2: Revisão bibliográfica, que contempla a descrição das principais tecnologias e plataformas utilizadas nesta solução;
- b) Capítulo 3: Desenvolvimento do projeto;
- c) Capítulo 4: Análise dos resultados, no qual são interpretados os resultados obtidos;
- d) Capítulo 5: Considerações finais, no qual é discutido o que foi observado e o que poderemos aprender com esta experiência.

## 2 – REVISÃO BIBLIOGRÁFICA

O objetivo deste capítulo é alertar para a necessidade de desenvolver na população a cultura da percepção de riscos de desastres naturais e apresentar as tecnologias existentes no mercado que possibilitam o desenvolvimento de uma solução inteligente para a redução destes riscos.

O capítulo está dividido em Hidrometria (2.1), Arduino (2.2), Sensores (2.3), RESTful (2.4), JSON (2.5), Xively (2.6), OpenShift (2.7), MongoDB (2.8), *Internet* das Coisas (2.9), Cidades Inteligentes (2.10) e Android (2.11).

### 2.1 – Hidrometria

A intensidade e frequência das chuvas é uma das principais preocupações da sociedade devido aos efeitos potencialmente danosos que podem acarretar quando ocorrem em excesso, podendo causar impactos nos setores natural, social e econômico (CEMADEN, 2013).

A hidrometria ou medição das variáveis hidrológicas do ambiente tem como objetivo obter dados básicos sobre precipitações, níveis de água, vazões, entre outros, e a sua variação no tempo e espaço (GUIMARÃES, 2008).

As chuvas diárias, também chamadas precipitações, geralmente são registradas através de pluviógrafos, pluviômetros de operação manual, pluviômetros semiautomáticos e mais raramente através de estações automáticas (KÜCHLER, 2010).

O pluviômetro é um dispositivo formado por uma superfície horizontal responsável pela captação da água, a qual é delimitada por um anel metálico, possuindo também um reservatório para acumular a água recolhida. Já o pluviógrafo é um equipamento que tem a capacidade de registrar de forma contínua a precipitação em um determinado local. O registro pode ser realizado de forma analógica ou digital. Os aparelhos analógicos possuem um mecanismo formado por um tambor giratório contendo um papel especial (diagrama) e uma agulha traçadora que registra em um gráfico a quantidade de chuva acumulada (GUIMARÃES, 2008).

## 2.2 – Arduino

Arduino é uma plataforma aberta (*open-source*) voltada para a construção de protótipos eletrônicos baseada em *hardware* de fácil adaptação e *software* simples, facilmente manipulável por qualquer pessoa interessada em criar objetos ou ambientes interativos.

Uma placa Arduino pode ser conectada a diversos sensores capazes de sentir o ambiente ao seu redor, podendo inclusive interagir com o ambiente através de controladores, motores e outros atuadores. É possível programar o microcontrolador da placa utilizando linguagem de programação e ambiente de desenvolvimento próprios do Arduino. Um projeto que envolva Arduino pode ser desenvolvido para trabalhar sozinho ou integrado a um *software* rodando em um computador (ARDUINO, 2013).

Arduino é composto de duas partes principais: o *hardware*, ou simplesmente a placa Arduino, com a qual se trabalha construindo objetos ou dispositivos e a interface de desenvolvimento através da qual se escreve o seu código, um pequeno programa de computador que será utilizado para controlar a placa (BANZI, 2009).

Existem placas pré-montadas à venda no mercado, entretanto também podem ser construídas à mão, devido à sua simplicidade. A plataforma de desenvolvimento, os drives e projetos de referência podem ser baixados gratuitamente e, por estarem disponíveis sob uma licença de código aberto, sua utilização e adaptação são livres (ARDUINO, 2013).

As instruções de programas para Arduino são escritas utilizando uma linguagem de programação de código aberto chamada *Processing* e podem ser desenvolvidas em um ambiente de programação chamado PDE (*Processing Development Environment*) escrito em Java, e que pode rodar em três modos: modo Java, modo JavaScript e modo Android. O PDE traduz as instruções (*sketches*) para a linguagem de programação C e as transfere para o compilador AVR-GCC existente no PDE. Com o código compilado, o próprio PDE pode ser utilizado para fazer o *Upload* para a placa.

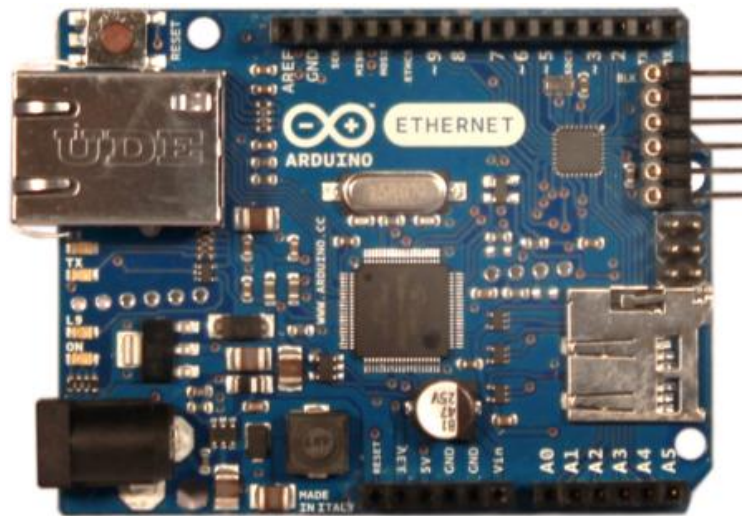
A linguagem de programação *Processing* foi inicialmente desenvolvida para ensinar fundamentos de programação a estudantes, entretanto, evoluiu para uma ferramenta mais completa capaz de gerar trabalhos profissionais e atualmente é muito utilizada por estudantes, pesquisadores, profissionais e entusiastas da prototipagem para a criação de estruturas sofisticadas (PROCESSING, 2013).

### 2.2.1 – Arduino Ethernet

O Arduino Ethernet é baseado em um microcontrolador ATmega328, um componente de alto desempenho e baixo consumo de energia, possui 14 pinos de entrada/saída digitais, 6 entradas analógicas, oscilador de cristal de 16 MHz, conexão RJ45, conector de alimentação, um botão de *reset*, além de leitor de cartão microSD, que pode ser usado para armazenar arquivos locais ou recebidos através da rede. A interface *Ethernet* é do tipo Wiznet, a mesma encontrada na placa Ethernet Shield. O fato de não possuir chip controlador USB não chega a ser um problema, visto que pode ser programado através de um cabo FTDI (ARDUINO, 2013), (INÁCIO, 2012).

Opcionalmente pode ser adicionado um módulo PoE (*Power over Ethernet*) como fonte de alimentação da placa (ARDUINO, 2013). Na Figura 2 se observa a placa Arduino Ethernet:

**Figura 2 – Arduino Ethernet**



Fonte: ARDUINO, 2013

### 2.3 – Sensores

Sensores são dispositivos capazes de converter estímulos físicos para sinais elétricos, através da ação de componentes, os quais podem ser analógicos ou eletrônicos ativos. Os sinais convertidos podem ser lidos e interpretados, funcionando como uma interface entre o mundo físico e os dispositivos eletrônicos (SOUZA JUNIOR, 2012).

### **2.3.1 – Sensor pluviométrico (*Rain Gauge*)**

O sensor pluviométrico utilizado na estação é do tipo caçamba basculante com esvaziamento automático. Possui um sistema que permite que a cada 0,011" (0,2794 milímetros) de chuva ocorra um fechamento momentâneo de contato que pode ser gravado com um contador digital ou uma interrupção da entrada do microcontrolador (ARGENTDATA, 2013).

### **2.3.2 – Sensor anemômetro (*Anemometer*)**

O sensor anemômetro do tipo copo mede a velocidade do vento fechando contato quando um ímã passa pelo interruptor. Um vento com a velocidade de 1,492 MPH (2,4 km / h) faz com que a chave feche uma vez por segundo (ARGENTDATA, 2013).

### **2.3.3 – Sensor cata-vento (*Wind Vane*)**

O sensor do tipo cata-vento possui oito interruptores, cada um deles ligado a um resistor diferente. Uma palheta magnética pode fechar dois interruptores de uma vez, permitindo que até 16 diferentes posições possam ser indicadas. Um resistor externo pode ser utilizado para formar um divisor de tensão, produzindo uma tensão de saída que pode ser medida com um conversor de analógico para digital (ARGENTDATA, 2013).

### **2.3.4 – Sensor de temperatura e umidade (DHT22)**

O sensor de temperatura e umidade modelo DHT22 é composto por um sensor capacitivo e um módulo digital de alta precisão. É necessário um resistor externo para produzir uma tensão de saída que pode ser medida com um receptor digital (AOSONG, 2013).

## **2.4 – RESTful**

Em engenharia de *software* geralmente se define arquitetura de um programa de computador como o conjunto de regras que especificam os tipos de componentes e conectores que serão utilizados para compor o sistema. Os serviços *web* atuais estão utilizando a

abordagem RESTful (*REpresentational State Transfer*) como uma alternativa ao uso de tecnologias baseadas em SOAP (*Simple Object Access Protocol*) por tratar-se de um protocolo leve e possuir capacidade de transmissão de dados diretamente via HTTP (*HyperText Transfer Protocol*) utilizando uma arquitetura cliente-servidor na qual os serviços *web* são vistos como recursos e são identificados por suas URLs (*Uniform Resource Locator*) (RESTFUL 2013), (ORACLE, 2013).

Uma solução baseada no estilo REST possui sua execução focada no recurso a ser usado (*resource-oriented*), ou seja, os métodos *GET*, *PUT*, *POST*, *DELETE*, etc. são implícitos, enquanto que nos serviços *web* tradicionais que são focados na atividade a ser usada (*activity-oriented*) o desenvolvedor precisa declarar o método HTTP a utilizar nas ações a serem realizadas (CRUD). Pode até ser notada alguma analogia entre os métodos HTTP e operações SQL, as quais também dependem de alguns verbos comuns (*INSERT*, *SELECT*, *UPDATE*, *DELETE*), entretanto, o REST faz uso de todo o potencial que a arquitetura *web* oferece, tornando-se parte dela. No estilo REST, o protocolo HTTP é implícito e não requer HTTP para troca de dados (RESTFUL, 2013), (LISBOA, 2012).

## 2.5 – JSON

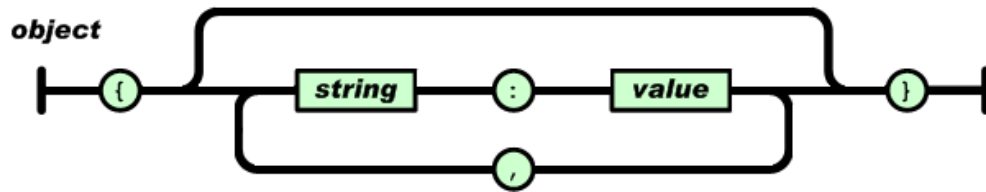
JSON (*JavaScript Object Notation*) é um protocolo de troca de dados baseado na linguagem de programação *JavaScript Standard ECMA-262 3ª Edição*, de dezembro de 2009, fácil de ser entendida por seres humanos e de ser analisada e gerada por máquinas. Pode-se dizer que o JSON é um formato de texto que usa convenções familiares aos programadores das linguagens C, C ++, C #, além de Java, JavaScript, Perl, Python, e muitas outras (JSON, 2013).

Para a construção/geração de um objeto JSON são utilizadas duas estruturas suportadas praticamente por todas as linguagens de programação modernas:

- uma coleção de pares do tipo nome/valor, conhecida em outras linguagens como objeto, *struct*, dicionário, tabela *hash*, lista encadeada, entre outras;
- uma lista ordenada de valores, conhecida pelos desenvolvedores como matriz, vetor, lista, sequência, etc.

Um objeto JSON é um conjunto desordenado de pares do tipo nome/valor com a seguinte notação: {nome:valor}. Os pares nome/valor são separados por vírgula (JSON, 2013). A Figura 3 mostra a estrutura de um objeto JSON:

**Figura 3 – Objeto JSON**



Fonte: JSON, 2013

## 2.6 – Xively

Xively é uma plataforma *web* escalável que possibilita a conexão entre dispositivos de *hardware* e aplicações visando fornecer dados armazenados em tempo real. Quando foi criada recebeu o nome de Pachube, tendo seu nome alterado para Cosm no ano de 2012. Em 2013 foi remodelada, agregando recursos de *PaaS (Platform as a Service)* e novamente renomeada, passando a se chamar Xively. Sua API aberta possibilita que empresas e desenvolvedores autônomos possam criar dispositivos, desenvolver protótipos e disponibilizar seus dados em um serviço *web* armazenados na nuvem que propicia fácil acesso e clara visualização através de gráficos (XIVELY 2013), (INÁCIO, 2012), (SOUZA JUNIOR, 2012).

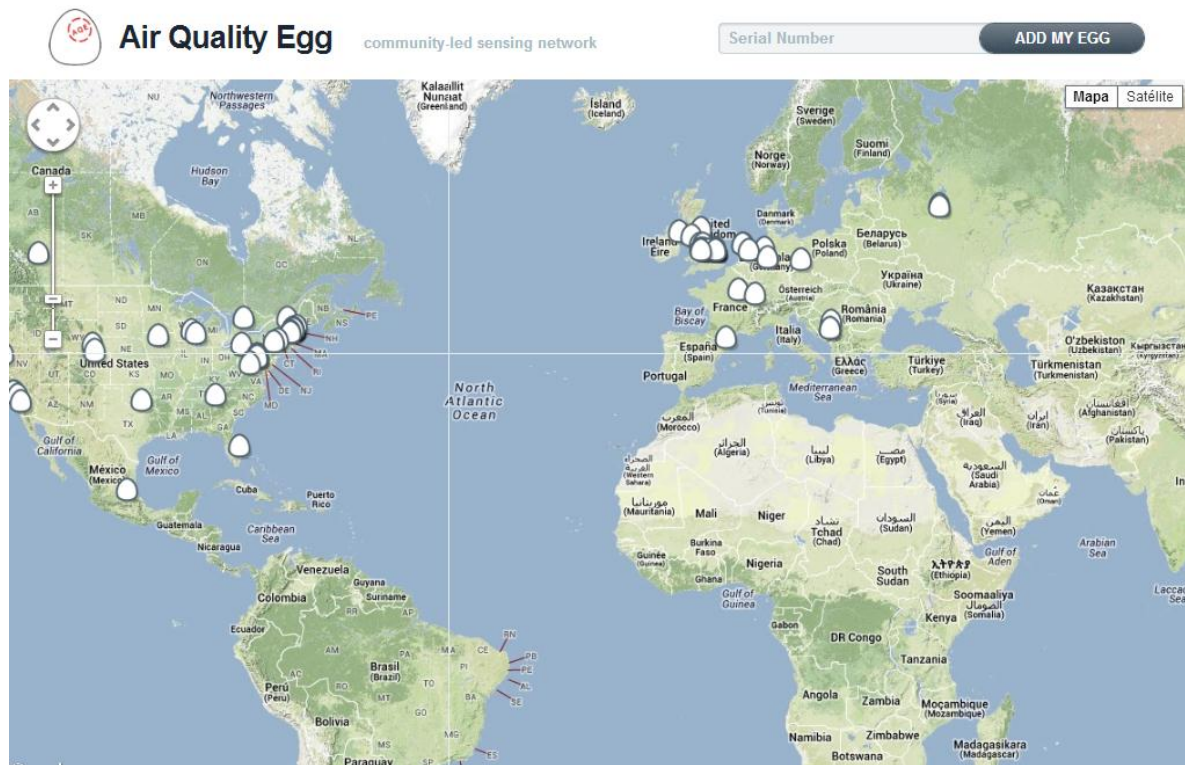
A plataforma Xively, pertencente ao grupo LogMeIn, oferece uma forma de lançar no mercado produtos voltados à internet sem a necessidade de construir uma infraestrutura de *back-end* (XIVELY, 2013).

A comunidade Xively, formada por desenvolvedores, fabricantes e entusiastas que trabalham com uma grande variedade de *hardware*, projetos e produtos, está engajada em disseminar a ideia da *Internet das Coisas*, estabelecendo movimentos voltados à construção de novos dispositivos que interagem entre si por intermédio de uma simples conexão com a *Internet*. É possível escolher entre os protocolos XML, JSON e dados CSV para interação bidirecional entre dispositivos e a *web* através da API RESTful (XIVELY, 2013), (INÁCIO, 2012).

A plataforma Xively, além de ser uma forma de criar protótipos de novos produtos, também é um serviço que ajuda as empresas a levar produtos ao mercado em grande escala (XIVELY, 2013). Na Figura 4 pode-se ver diversos dispositivos conectados à plataforma Xively ao redor do mundo:



**Figura 4 – Dispositivos conectados ao Xively**



Fonte: AIRQUALITYEGG, 2013

## 2.7 – OpenShift

OpenShift é uma plataforma de *Cloud Computing* do tipo *Platform-as-a-Service* (PaaS) oferecida pela empresa RedHat, que fornece aos programadores uma solução escalável para desenvolvimento e hospedagem de *software* em um ambiente de nuvem público ou privado. Os desenvolvedores e arquitetos de soluções podem facilmente introduzir novas tecnologias, estruturas, serviços e integrações através da API pública oferecida pela plataforma e do seu código aberto provido pelo projeto Origin (OPENSIFT, 2013).

A plataforma OpenShift foi projetada para manter os aplicativos disponíveis para os usuários e garante ser resiliente a falhas, persistindo todos os estados das aplicações em um *cluster* MongoDB rápido e confiável.

As aplicações inativas ou raramente visitadas podem ser desativadas e reativadas a qualquer momento de modo a permitir a alocação de recursos conforme necessário. Além disso, os administradores podem aumentar a capacidade adicionando um novo *hardware* (físico ou virtual) para atender o aumento da demanda (OPENSIFT, 2013).

## 2.8 – MongoDB

MongoDB é um banco de dados de código aberto orientado a documentos, no qual cada registro é considerado um documento. É escrito na linguagem C++ e é considerado um dos principais bancos de dados NoSQL (MONGODB, 2013).

O banco de dados MongoDB utiliza o BSON, um formato de serialização usado para armazenar documentos e fazer chamadas de procedimento remoto. BSON é uma junção das palavras “*binary*” e “JSON”. Desta forma, pode-se considerar que um BSON é uma representação binária de documentos JSON (*JavaScript Object Notation*).

Os dados em MongoDB possuem um esquema flexível e são armazenados em estruturas chamadas de coleções. Um documento pertencente a uma coleção não precisa de uma estrutura definida. Os documentos de uma mesma coleção não precisam ter o mesmo conjunto de campos ou estrutura, e campos comuns em documentos de uma coleção podem conter diferentes tipos de dados.

Cada documento só precisa conter os campos relevantes para a entidade ou objeto que representa. A flexibilidade do esquema possibilita a modelagem dos documentos no MongoDB de modo que possam se assemelhar e refletir objetos em nível de aplicativo.

No desenvolvimento de modelos de dados, como em toda a modelagem de dados, devem-se considerar as propriedades inerentes e exigências dos objetos da aplicação e as relações entre objetos da aplicação (MONGODB, 2013).

## 2.9 – Internet das Coisas

A atual fase de desenvolvimento tecnológico permitiu que a microeletrônica e a comunicação sem fio se integrem de uma maneira tal que proporcionou que sejam desenvolvidos sensores com a capacidade de comunicar-se entre si, possibilitando a criação de redes interconectadas de sensores. Tais sensores podem ser utilizados para o desenvolvimento de dispositivos para automação doméstica, monitoramento ambiental, entre outros. À medida que processadores dedicados passam a se incorporar a dispositivos eletrônicos e a *internet* vem ampliando a conectividade entre as residências e empresas, cria-se a visão de uma *Internet das Coisas*. O termo *Internet das Coisas*, inicialmente definido como uma rede capaz de localizar diversos objetos identificados por RFID, nos dias de hoje refere-se a uma sociedade de objetos ubíquos, na qual diferentes objetos estão conectados,

combinando RFID, redes de sensores e tecnologias ubíquas (GAMA, 2012), (GUARDIA, 2012).

Os dispositivos móveis, os sensores do tipo RFID e as redes sem fio formam a chamada computação ubíqua da era da *Internet* das Coisas. Tal tecnologia permite que o ciberespaço interaja com os lugares e os objetos do dia-a-dia, de forma tal que a informação eletrônica é produzida, distribuída, acessada e consumida a partir dos mais diversos dispositivos e objetos em todo e qualquer lugar (LEMOS, 2009).

Nos dias atuais ainda não há uma homogeneidade entre os diversos dispositivos existentes, visto que são utilizados diferentes protocolos para proporcionar sua interconexão e também não há uma definição sobre as melhores soluções para o armazenamento dos dados oriundos destes sensores. Entretanto, estão sendo criadas cadeias de coleta de dados de sensores que possam captar informação do meio ambiente e torna-la acessível, de forma fácil e uniforme, a aplicações de alto nível, de modo que possibilite o uso de computação ubíqua no domínio de cidades inteligentes (GUARDIA, 2012), (QUINTAS, 2010).

## **2.10 – Cidades inteligentes**

As informações e dados transmitidos todos os dias aos cidadãos através de diversos meios eletrônicos são geralmente disponibilizados de forma bruta e incompreensível, sendo desperdiçados ou mesmo perdidos se não existirem mecanismos e soluções com a capacidade de recebê-los e trata-los de forma inteligente. Cidades Inteligentes são definidas como o uso de Tecnologias da Informação e Comunicação para analisar e integrar os dados recebidos de sistemas centrais a fim de que possam ser utilizados como base para tomada de decisões (AVELAR, 2012).

A gestão de uma cidade inteligente deve estar voltada para o investimento em tecnologia da informação e suas estruturas físicas, visando aumentar a eficiência da comunicação móvel, cuidar da qualidade do ar e da água, coletar e organizar dados do ambiente e compartilhá-los de forma colaborativa com todas as entidades envolvidas na gestão social de modo a ser capaz de se recuperar de desastres com rapidez (STRAPAZZON, 2009).

As cidades inteligentes podem impulsionar a prosperidade dos seus cidadãos e propiciar o seu crescimento econômico e sustentável, disponibilizando ferramentas para análise de dados e tomada de decisões, como a coordenação dos recursos visando à resolução

de problemas de forma proativa, tendo como cenário a cidade como um todo a fim de otimizar o emprego dos recursos orçamentários (SMARTER CITIES, 2013).

## 2.11 – Android

O sistema operacional Android está instalado em milhões de dispositivos móveis em mais de 190 países ao redor do mundo. Considerada atualmente a maior plataforma instalada em dispositivos móveis, vem crescendo rapidamente, disponibilizando a seus usuários inúmeros aplicativos, jogos e outros conteúdos digitais. A plataforma Android proporciona aos desenvolvedores um excelente ambiente para a criação e distribuição de aplicativos e jogos para usuários de dispositivos móveis (ANDROID, 2013).

O Android se tornou rapidamente o sistema operacional móvel que mais cresce no mundo devido às contribuições da comunidade de código aberto Linux, bem como aos parceiros desenvolvedores de *hardware* e *software*.

Segundo seus desenvolvedores, todos os dias mais de um milhão de novos dispositivos Android são ativados em todo o mundo, o que demonstra que a plataforma se tornou favorita para consumidores e desenvolvedores, visto que sua inovação permite construir poderosas aplicações diferenciadas, que utilizam as mais recentes tecnologias móveis (ANDROID, 2013). A Figura 5 ilustra o crescimento de ativações de dispositivos Android:



Fonte: ANDROID, 2013

Analisando o considerável crescimento do número de ativações de dispositivos Android e sua crescente popularização, percebe-se que o campo para o desenvolvimento de soluções neste cenário é fértil e possibilitará um sensível avanço na prevenção de desastres ambientais através da observação e ampla divulgação das variáveis do ambiente.

### 3 – DESENVOLVIMENTO

O objetivo deste capítulo é descrever a arquitetura utilizada para o desenvolvimento da solução, a qual doravante será tratada como *Weather Station*.

Todos os recursos de *hardware* e *software* utilizados para o desenvolvimento da *Weather Station* são baseados em plataforma aberta (*open-source*) a fim de proporcionar livre acesso a qualquer pessoa interessada em utilizá-la ou mesmo modificá-la.

O capítulo está dividido em Arquitetura Global (3.1), Estrutura do *software* desenvolvido para o Arduino (3.2), Sensores (3.3), *Hardware* da *Weather Station* (3.4), Comunicação do Arduino com a plataforma Xively (3.5), Comunicação do Arduino com a plataforma OpenShift (3.6), *Web Service* RESTful (3.7), Objeto JSON (3.8), Banco de dados não relacional MongoDB (3.9), Comunicação dos dispositivos com a plataforma Xively (3.10), Comunicação dos dispositivos com a plataforma OpenShift (3.11) e Aplicação Android (3.12).

#### 3.1 – Arquitetura Global

O sistema está dividido basicamente em duas partes: Física (*hardware*) e lógica (*software*). A estrutura física consiste em um módulo Arduino Ethernet conectado à rede através do protocolo TCP/IP e que possui 04 (quatro) sensores interligados, utilizados para captar as seguintes variáveis do ambiente: velocidade do vento, volume de chuva, origem do vento, temperatura e umidade relativa do ar.

O sistema lógico é formado por um *software* gerenciador do Arduino, um *Feed* disponibilizado pela plataforma Xively (XIVELY, 2013), um *Web Service* (RESTFUL, 2013) construído sobre a plataforma OpenShift (OPENSIFT, 2013) e uma aplicação para acesso aos dados desenvolvida para o Sistema Operacional Android (ANDROID, 2013).

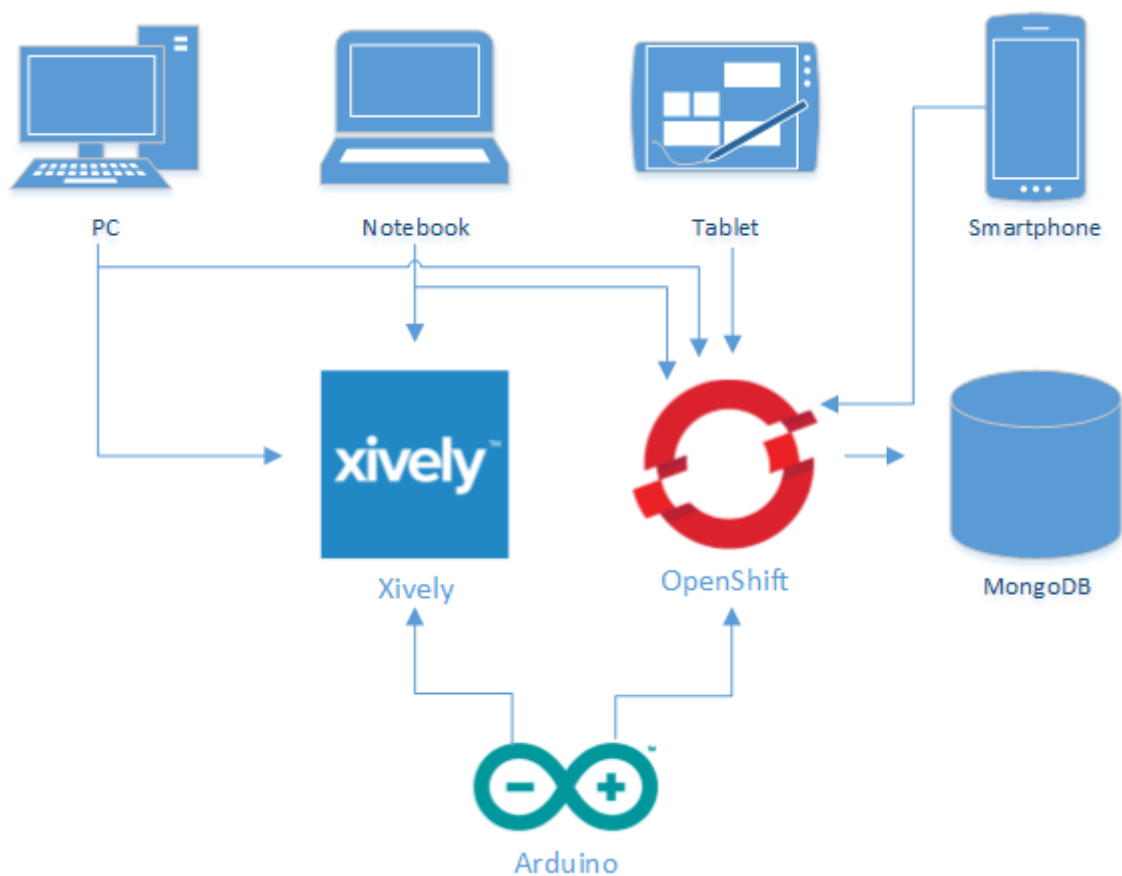
O *software* do Arduino se conecta ao serviço Xively utilizando a biblioteca de mesmo nome fornecida pela própria plataforma e envia os dados através de requisições HTTP PUT (BERNERS-LEE, 1999). Já o envio dos dados para o *Web Service* hospedado no OpenShift é realizado através de requisições HTTP POST (BERNERS-LEE, 1999). Esses dados são tratados e armazenados em um banco de dados NoSQL (SADALAGE, 2013), também disponibilizado pela plataforma OpenShift. O banco de dados utilizado para a solução é o MongoDB (MONGODB, 2013).

A escolha de um banco de dados não relacional visa possibilitar maior liberdade para alteração na modelagem dos dados que serão persistidos, visto não haver necessidade de se

preocupar com alteração de tabelas, criação de novos campos, alteração de relacionamentos, chaves estrangeiras, entre outras características de um banco relacional. Uma alteração no modelo dos dados será persistida no banco de dados NoSQL sem a necessidade de alteração na sua estrutura (COUCHDB, 2013).

A Figura 6 apresenta a arquitetura global do sistema:

**Figura 6 – Arquitetura global**



Fonte: Adaptação de ARDUINO, XIVELY e OPENSIFT, 2013

A seguir será apresentada a descrição de cada um dos módulos da *Weather Station*:

- **Arduino** – O *software* desenvolvido para o microcontrolador do Arduino Ethernet, sua implantação, bem como as bibliotecas utilizadas serão descritos em detalhes no item 3.2;
- **Sensores** – Os sensores utilizados na solução, sua calibragem, os trechos de código responsáveis pelo tratamento dos dados coletados, bem como os esquemas de ligação serão descritos em detalhes no item 3.3;

- *Hardware* – O *hardware* contendo os circuitos para interligação dos sensores à placa Arduino será descrito no item 3.4;
- *Xively* – O *feed* cadastrado na plataforma Xively, os canais de dados (*datastreams*) e os protocolos de conexão e transferência de dados serão descritos em detalhes no item 3.5;
- *OpenShift* – No item 3.6 serão descritos em detalhes a comunicação do módulo Arduino e transferência de dados para o *Web Service* hospedado na plataforma OpenShift;
- *Web Service RESTful* – A estrutura, as classes e os serviços oferecidos pelo *Web Service* serão detalhados no item 3.7;
- *Objeto JSON* – A troca de dados entre o *Web Service* e a aplicação para dispositivos móveis será descrita em detalhes no item 3.8;
- *Banco de dados não relacional MongoDB* – O item 3.9 apresentará a estrutura dos dados persistidos no Banco de Dados e a coleção onde estes objetos serão armazenados;
- *Comunicação dos dispositivos clientes com a plataforma Xively* – O acesso aos dados disponibilizados pela plataforma Xively será descrito em detalhes no item 3.10;
- *Comunicação dos dispositivos clientes com o Web Service desenvolvido sobre a plataforma OpenShift* – No item 3.11 serão descritos os detalhes de acesso aos dados disponibilizados pelo *Web Service* aos clientes *Desktop*, *Web* e *Móveis*;
- *Android* – A aplicação desenvolvida para a plataforma Android será descrita no item 3.12.

### **3.2 – Estrutura do *software* desenvolvido para o Arduino Ethernet**

Para o desenvolvimento do *software* gerenciador do Arduino Ethernet foram incluídas as seguintes bibliotecas:

- *Biblioteca SPI.h* : possibilita a utilização da porta SPI, que possui um padrão de comunicação serial Síncrono. Nessa porta são conectados cartões de memória (SD), por exemplo;
- *Biblioteca Ethernet.h* : possibilita que a placa Arduino se conecte à *internet*, aceitando tanto conexões de saída como de entrada. Suporta até quatro conexões simultâneas.
- *Biblioteca HttpClient.h* : possibilita a utilização dos métodos HTTP pela placa Arduino;



- Biblioteca Xively.h : permite a comunicação da placa Arduino com a plataforma Xively;
- Biblioteca DHT: possibilita acesso aos métodos de leitura de temperatura e umidade do sensor DHT22.

### 3.3 – Sensores

A fim de viabilizar a captação das informações do ambiente é necessária a utilização de sensores, que são dispositivos capazes de converter estímulos físicos para sinais elétricos, de modo a conectar o mundo físico com os dispositivos eletrônicos.

Para o desenvolvimento do sistema, foram utilizados um sensor pluviométrico (3.3.1), um sensor anemômetro (3.3.2), um sensor cata-vento (3.3.3) e um sensor de temperatura e umidade (3.3.4).

#### 3.3.1 – Sensor pluviométrico (*Rain Gauge*)

A medição da quantidade de precipitação no ambiente é realizada por um sensor pluviométrico tipo caçamba basculante com capacidade de 0,2794 milímetros de líquido. Cada vez que a caçamba se enche ocorre um fechamento momentâneo de contato que pode ser gravado com um contador. A Figura 7 mostra o sensor utilizado na *Weather Station* para a medição da quantidade de chuva:

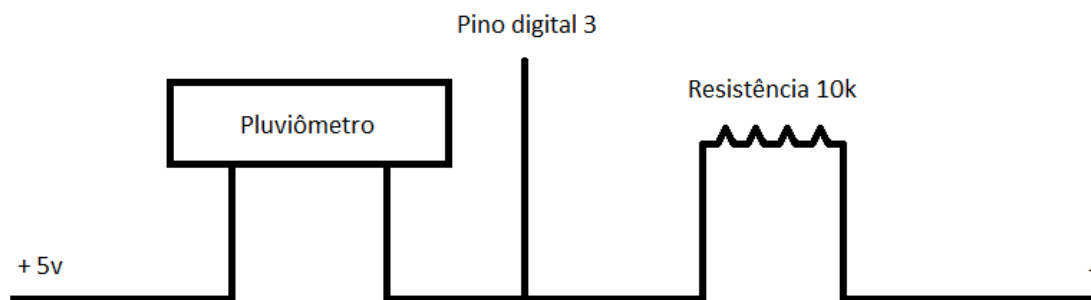
**Figura 7 – Sensor pluviométrico (*rain gauge*)**



Fonte: AUTOR, 2013

O algoritmo utilizado é composto de um contador que conta a quantidade de vezes que o contato é fechado num intervalo de quinze minutos e o multiplica pela capacidade de líquido da caçamba basculante. Considerando que o microcontrolador da placa Arduino pode interpretar um contato como fechado diversas vezes antes que esteja novamente aberto, a cada incremento do contador ocorre um *delay* de 125 milissegundos, de modo a não comprometer a medição do nível de precipitação. Na Figura 8 observa-se o esquema do circuito do sensor *rain gauge*:

**Figura 8 – Esquema do circuito do sensor pluviométrico (*rain gauge*)**



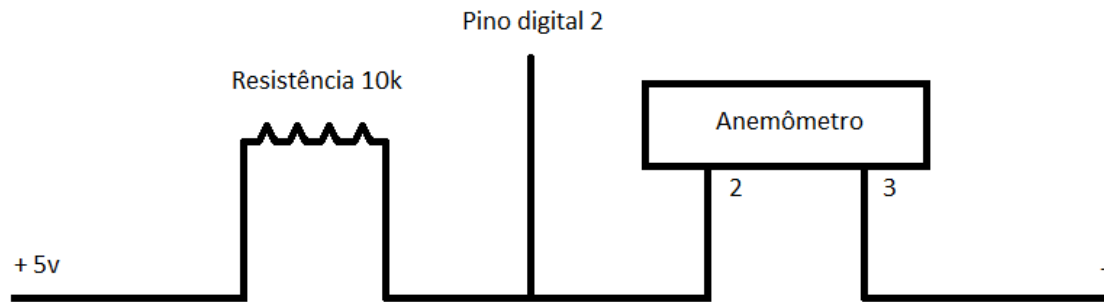
Fonte: AUTOR, 2013

### 3.3.2 – Sensor anemômetro (*Anemometer*)

A velocidade do vento é medida com um sensor anemômetro do tipo copo, o qual gera uma interrupção por segundo quando submetido a um vento de velocidade de 1,492 MPH (2,4 km/h).

O algoritmo não é complexo, pois possui um contador que verifica quantas interrupções foram registradas pelo microcontrolador durante um período de cinco segundos (5000 milissegundos) e realiza o cálculo da velocidade do vento multiplicando o contador por 2400 e dividindo pelos milissegundos. A Figura 9 ilustra o esquema do circuito do sensor *anemometer*:

**Figura 9 – Esquema do circuito do sensor anemômetro (*anemometer*)**



Fonte: AUTOR, 2013

### 3.3.3 – Sensor cata-vento (*Wind Vane*)

A medição da origem do vento é realizada através de um sensor do tipo cata-vento que possui oito interruptores internos, cada um deles ligado a um resistor diferente. A palheta magnética interna do dispositivo pode fechar dois interruptores de uma vez, permitindo que até 16 diferentes posições possam ser indicadas. A Figura 10 mostra os sensores *anemometer* e *wind vane* utilizados na *Weather Station*:

**Figura 10 – Sensores anemômetro (*anemometer*) e cata-vento (*wind vane*)**



Fonte: AUTOR, 2013

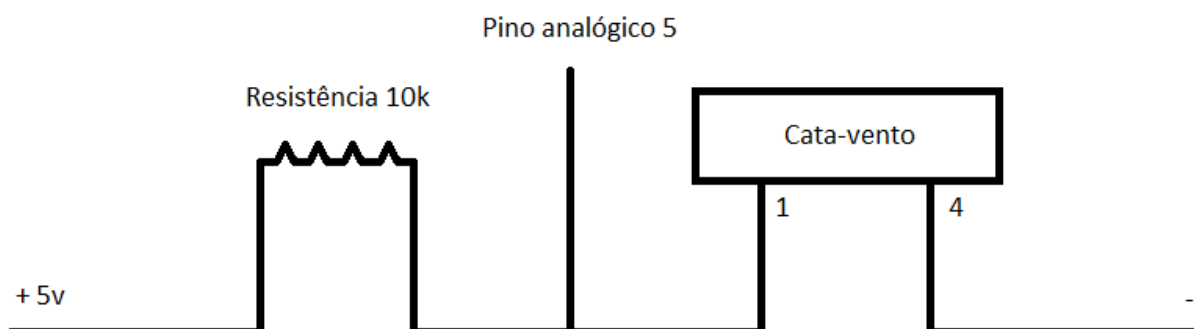
O circuito do sensor foi montado com um resistor externo de 10 K e alimentado com uma corrente de 5v produzindo as tensões de saída descritas no Quadro 1:

**Quadro 1 – Relação entre inclinação da paleta, resistência e voltagem**

Direção (Graus)	Resistência (Ohms)	Voltagem (V=5v, R=10k)
0	33k	3.84v
45	8.2k	2.25v
90	1k	0.45v
135	2.2k	0.90v
180	3.9k	1.40v
225	16k	3.08v
270	120k	4.62v
315	64.9k	4.78v

Fonte: ARGENTDATA, 2013

O algoritmo possui um ponteiro contendo oito posições correspondentes aos pontos cardeais, a saber: “W”, “NW”, “N”, “SW”, “NE”, “S”, “SE” e “E”, correspondentes às oito inclinações da paleta magnética de modo a corresponder o valor resistivo gerado pelo sensor com uma posição do ponteiro. O esquema do circuito do sensor *wind vane* é mostrado na Figura 11:

**Figura 11 – Esquema do circuito do sensor cata-vento (*wind vane*)**

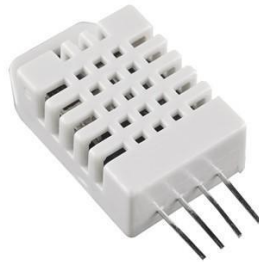
Fonte: AUTOR, 2013

### 3.3.4 – Sensor de temperatura e umidade (DHT22)

O sensor de temperatura e umidade modelo DHT22, também conhecido entre os fabricantes como RHT03, utilizado no projeto é composto por um sensor de temperatura do

tipo NTC (*Negative Temperature Coefficient*), um sensor de umidade capacitivo e um módulo digital de alta eficiência pré-calibrado de fábrica. A Figura 12 mostra o sensor DHT22:

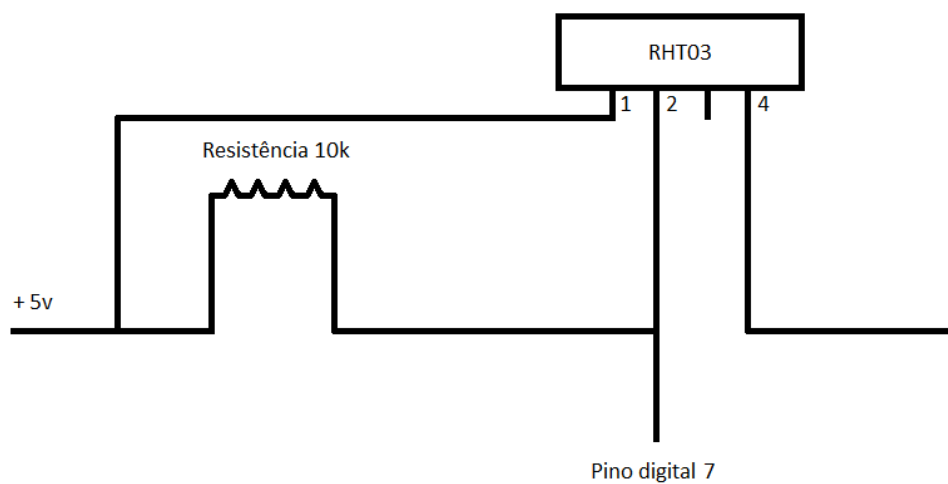
**Figura 12 – Sensor DHT22 (RHT03)**



Fonte: AOSONG, 2013

A *Weather Station* utiliza um resistor de 10K Ohms alimentado com 5v para a conexão com o sensor. O esquema do seu circuito é mostrado na Figura 13:

**Figura 13 – Esquema do circuito do sensor DHT22 (RHT03)**

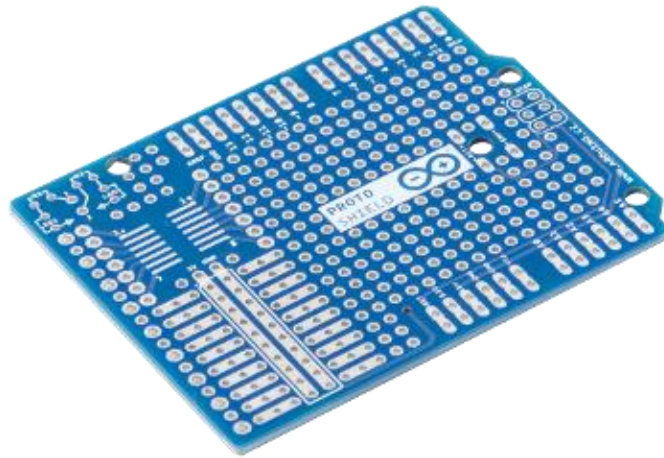


Fonte: AUTOR, 2013

### 3.4 – Hardware para interligação dos sensores

Para a construção dos circuitos que possibilitarão a interligação dos sensores foi utilizada uma placa do tipo Proto Shield fabricada pela própria Arduino. A Figura 14 mostra a placa utilizada na *Weather Station*:

**Figura 14 – Placa Proto Shield**



Fonte: ARDUINO, 2013

A fim de possibilitar uma perfeita adaptação dos circuitos necessários para o funcionamento do sistema a placa é fornecida como exibida na figura, sem nenhum componente, de modo que há plena liberdade para a sua montagem.

### **3.5 – Comunicação do Arduino com a plataforma Xively**

A comunicação do Arduino com a plataforma Xively se dá através da porta *Ethernet* e utiliza para tal as bibliotecas *Ethernet.h* e *HttpClient.h*, necessárias para a comunicação de dados com o servidor. Os dados são enviados para a plataforma utilizando-se requisições do tipo HTTP PUT (BERNERS-LEE, 1999). O método de envio de dados é definido pela própria plataforma, que é construída com os princípios RESTful.

Para a geração dos canais de dados (*datastreams*) que serão enviados para a plataforma Xively é necessária a biblioteca *Xively.h*. Cada dado obtido de um sensor é inserido em um *datastream*. Quando há mais de um *datastream*, é necessário criar um *array* de *datastreams*, que conterá todos os canais de dados gerados. A biblioteca gera *datastreams* no formato JSON.

A *Weather Station* possui cinco *datastreams*, denominados *anemometer*, *raingauge*, *temperature*, *humidity* e *wind vane*, correspondentes aos sensores responsáveis pela captura da velocidade do vento, volume de chuva, temperatura, umidade relativa do ar e origem do vento respectivamente.

### 3.6 – Comunicação do Arduino com a plataforma OpenShift

A comunicação do Arduino com a plataforma OpenShift também é feita através da porta *Ethernet*, utilizando as bibliotecas *Ethernet.h* e *HttpClient.h*. Os dados são enviados para o *web service* hospedado na plataforma OpenShift utilizando-se requisições do tipo HTTP POST (BERNERS-LEE, 1999).

Os dados recebidos pelo *web service* são tratados e validados quanto ao formato e origem antes de serem persistidos no banco de dados, de modo a garantir a confiabilidade das informações. Para que seja validada, a *String* recebida pelo *web service* deve ser formada obrigatoriamente pela expressão *string-valor-valor-valor-string-string*, que correspondem respectivamente a: código validador, valor do anemômetro, valor do pluviômetro, valor da temperatura, origem do vento e umidade.

O *software* da *Weather Station* possui classes e métodos que possibilitam a comunicação com outros sistemas através de requisições GET e POST (BERNERS-LEE, 1999), bem como uma interface amigável que possibilita ao usuário comum visualizar os dados medidos através de um navegador de *internet*.

A página principal da *Weather Station* está escrita em linguagem html 5 e javascript e os gráficos foram desenvolvidos com a API Google Charts. Está disponível na *url*: <http://weatherstation-hitechdv.rhcloud.com/>.

### 3.7 – Web Service RESTful

O *web service* construído para a *Weather Station* é do tipo RESTful e foi desenvolvido em linguagem Java, utilizando a API JBoss jaxrs 2.1.0. O acesso aos métodos HTTP, implícitos neste tipo de *web service*, se dá através de anotações (@GET, @POST), as quais também são utilizadas para definir os parâmetros a receber, o caminho da aplicação, a indicação do formato de dados que produz e que pode consumir (@PathParam, @Path, @Produces, @Consumes), entre outras definições (FIELDING, 2000).

Todas as dependências da aplicação estão definidas no arquivo pom.xml, de maneira que o mecanismo de gerenciamento do Maven faz as importações para o seu repositório local de dependências, caso não existam.

A aplicação está dividida em diversas classes: Classes DAO (*Data Access Object*), responsáveis pelas operações de leitura e gravação no Banco de Dados, Classe *Factory*, responsável pela criação e gerenciamento de conexões com o Banco de Dados, Classes *Model*, responsáveis pela criação dos Objetos e Classe *Rest*, responsável por disponibilizar os

serviços oferecidos pelo *web service*. O caminho da classe *Rest* está mapeado no arquivo *web.xml* (FREEMAN, 2005), (GAMMA, 2006), (SOMMERVILLE, 2007).

### 3.8 – Objeto JSON

Tanto a aplicação disponibilizada na plataforma Xively, quanto a aplicação construída no OpenShift fornecem informações para aplicações externas através de requisições aos seus serviços. Um dos formatos fornecidos por ambas as plataformas é o formato JSON (JSON, 2013).

Por tratar-se de uma estrutura no formato texto que pode ser analisada pela máquina e de tamanho consideravelmente reduzido, o Objeto JSON não exige grandes trocas de dados entre cliente/servidor, de modo que é indicado para interação com aplicações *mobile*.

A Figura 15 exibe um exemplo de parte de um ponto de dados da *Weather Station* no formato JSON obtido da plataforma Xively:

**Figura 15 – Ponto de Dados da aplicação *Weather Station***

```
{ "id": 187368934, "title": "Arduino Weather Station", "private": "false", "tags": ["estação meteorológica", "weather station"], "feed": "https://api.xively.com/v2/feeds/187368934.json", "status": "live", "updated": "2013-05-30T01:00:10.251420Z", "created": "2013-05-16T12:01:26.622711Z", "email": "jefuchoas@gmail.com", "creator": "https://xively.com/users/jefuchoas", "version": "1.0.0" }
```

Fonte: AUTOR, 2013

Como se pode verificar, trata-se de um conjunto de pares do tipo chave: valor, no qual cada valor pode ser acessado referenciando-se sua chave específica.

### 3.9 – Banco de dados não relacional MongoDB

Os dados registrados pela *Weather Station* e informações como temperatura média diária, volume de chuva acumulada, entre outras, são persistidos no Banco de Dados não relacional MongoDB.

A plataforma OpenShift oferece suporte nativo ao Banco de Dados MongoDB, de modo que a *string* (OPENSIFT\_MONGODB\_DB\_HOST), bem como a porta (OPENSIFT\_MONGODB\_DB\_PORT) de conexão com o banco são gerenciadas internamente à aplicação (OPENSIFT, 2013).



O banco de dados da aplicação possui a seguinte estrutura: uma base de dados denominada *weatherstation* e quatro coleções denominadas *dados\_meteorologicos*, *dados\_diarios*, *devices* e *cellphones* nas quais são persistidos, respectivamente, os dados registrados pelos sensores, as informações diárias manipuladas pela aplicação os dispositivos Android com a aplicação instalada e os telefones celulares cadastrados para receber alertas.

Os dados são persistidos no formato do próprio modelo do objeto construído pelo *Bean* Java, de modo que sua estrutura é conhecida. Basta receber o objeto BSon do banco e convertê-lo novamente para o modelo Java e então acessar seus atributos através dos seus métodos *Getters*.

A Figura 16 ilustra um objeto BSon armazenado no Banco de Dados da aplicação. Nela se observa o atributo *\_id* criado automaticamente durante a inserção dos dados, a data e a hora da gravação dos dados, as variáveis do ambiente enviadas pelo Arduino, bem como as informações inseridas pela estação.

**Figura 16 – Objeto BSon da aplicação *Weather Station***

```
{
  "_id": ObjectId("52589837e4b0bee1f1046220"),
  "anemometer": 7.6799998283386,
  "raingauge": 0,
  "temperature": 17.200000762939,
  "windvane": "SE",
  "humidity": 99.900001525879,
  "data": "11-10-2013",
  "hora": "21: 30: 46",
  "moisture": 17.180000305176,
  "origin": "Sudeste",
  "intensity": "Não chove agora"
}
```

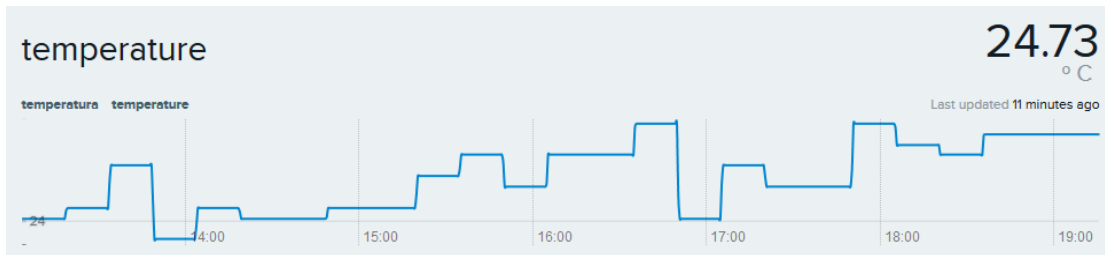
Fonte: AUTOR, 2013

### 3.10 – Comunicação dos dispositivos com a plataforma Xively

Para a disponibilização dos dados ao usuário final a *Weather Station* possui um *feed* chamado “*Arduino Weather Station*”, o qual exibe as informações dos canais de dados enviadas pelo Arduino em forma de gráfico. O *feed* é identificado publicamente pelo id 187368934 e pode ser acessado pelos usuários através da *url* <https://xively.com/feeds/187368934>.

A Figura 17 ilustra o canal de dados da *Weather Station* que exibe os últimos valores de temperatura medidos pelo Arduino:

**Figura 17 – Canal de dados *temperature* da *Weather Station***



Fonte: ARDUINO WEATHER STATION, 2013

Os dados disponibilizados pela *Weather Station* são públicos e também podem ser acessados por outras aplicações através de solicitações GET diretamente à plataforma Xively, de maneira que outros desenvolvedores, órgãos do governo, instituições de defesa civil, entre outras podem utilizar tais informações para integração com seus sistemas de monitoramento e prevenção de desastres.

Os pontos de dados são disponibilizados por padrão no formato JSON. Uma requisição para a *url* <https://api.xively.com/v2/feeds/187368934> retornará um objeto JSON. Entretanto, por padrão todas as requisições HTTP GET retornam informações temporais em UTC. Para que as informações temporais (*timestamps*) sejam exibidas no horário do Brasil, é necessário inserir a informação da zona desejada na própria *url* da requisição. Deste modo, a *url* que retornará um objeto JSON com *timestamp* no horário do Brasil será <https://api.xively.com/v2/feeds/187368934?timezone=-3>.

Também podem ser configurados gatilhos (*triggers*) que disponibilizam recursos de atualizações automáticas para um serviço da *internet* através do envio de solicitações HTTP POST para uma *url* específica quando uma condição for satisfeita, como, cada vez que um valor for atualizado, quando uma medição for maior que um determinado valor, etc. A resposta da plataforma Xively, tanto para os gatilhos quanto para as requisições HTTP GET, pode ser obtida nos formatos JSON, XML e CSV.

Para que uma aplicação externa obtenha os pontos de dados registrados na plataforma Xively no formato XML, a *url* de requisição deve necessariamente informar o formato desejado. A *url* que retornará um ponto de dados XML e com *timestamp* no horário brasileiro será <https://api.xively.com/v2/feeds/187368934.xml?timezone=-3>.

É possível requisitar as informações de um *datastream* específico passando o *id* desejado na própria *url* de requisição, sem a necessidade de receber um *feed* completo para depois acessar suas chaves internas. Desta forma, caso uma aplicação externa queira obter

apenas os dados de temperatura, basta fazer a requisição <https://api.xively.com/v2/feeds/187368934.json?datastreams=temperature>.

Outra possibilidade viável é obter uma série histórica de pontos de dados armazenados. Um exemplo de requisição para que uma aplicação obtenha todos os pontos de dados armazenados na base de dados nas últimas seis horas seria <https://api.xively.com/v2/feeds/187368934.json?duration=6hours&interval=0>.

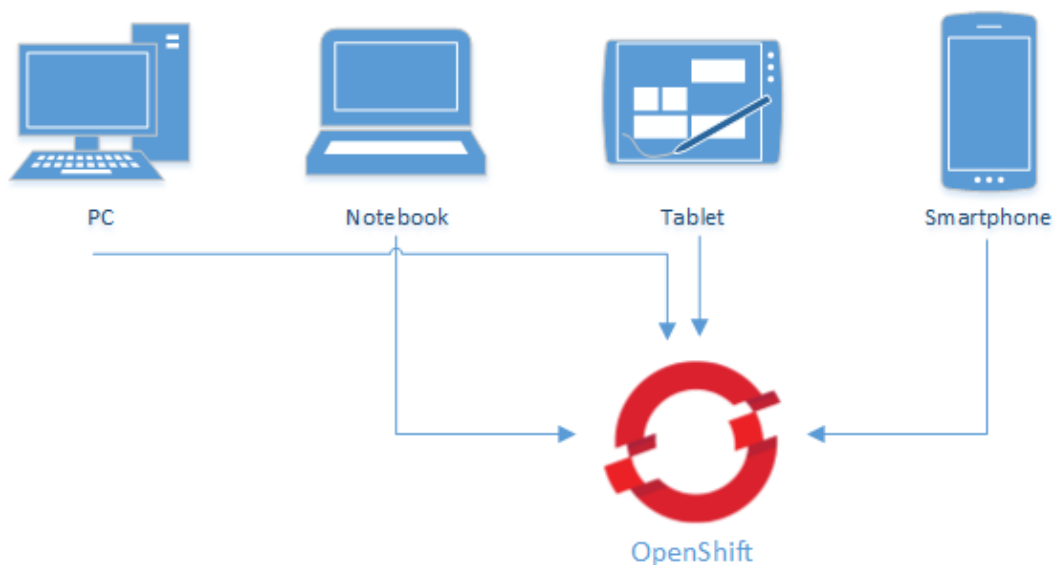
### 3.11 – Comunicação dos dispositivos com a plataforma OpenShift

A aplicação construída sobre a plataforma OpenShift é composta por um *web service* que responde a requisições HTTP e devolve objetos no formato JSON para aplicações de terceiros e dispositivos *mobile*. Possui também uma interface *web* amigável que pode ser acessada por navegadores de *internet* como Internet Explorer, Google Chrome, Firefox, Opera, entre outros.

A interface *web* exibe informações sobre as medições de temperatura, umidade relativa do ar, volume de chuva, velocidade e origem do vento atuais, e também gráficos contendo a média histórica das medições de temperatura, temperaturas máxima e mínima, além do volume de chuva acumulado. Também disponibiliza consultas a medições de datas e horários anteriores.

A Figura 18 exemplifica o acesso dos dispositivos à plataforma OpenShift:

**Figura 18 – Comunicação Dispositivos/OpenShift**



Fonte: Adaptação de OPENSIFT, 2013

O *web service* aceita requisições GET de dispositivos *mobile* e envia como resposta um objeto completo no formato JSON contendo todas as informações das medições atuais e também objetos temporais contendo informações de registros em datas e horários específicos. Tais informações são fornecidas no formato JSON e podem ser utilizadas para integração com outros sistemas.

Os dados disponíveis podem ser acessados através de requisições aos endereços listados no Quadro 2:

**Quadro 2 – Endereços para requisição ao *web service***

Informação	Endereço de requisição
Última medição	<a href="http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosMeteorologicos">http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosMeteorologicos</a>
Dados do dia corrente	<a href="http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosDiarios">http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosDiarios</a>
Últimas 16 medições	<a href="http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosMeteorologicosList">http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosMeteorologicosList</a>
Últimos 15 registros diários	<a href="http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosDiariosList">http://weatherstation-hitechdv.rhcloud.com/rest/dados/dadosDiariosList</a>
Chuva acumulada de 3 dias	<a href="http://weatherstation-hitechdv.rhcloud.com/rest/dados/threeDaysRain">http://weatherstation-hitechdv.rhcloud.com/rest/dados/threeDaysRain</a>

Fonte: AUTOR, 2013

### 3.12 – Aplicativo Android

O aplicativo desenvolvido para a plataforma Android possibilita acesso a todos os dados disponibilizados pela *Weather Station*. Os dados são acessados através de requisições GET às *url* do *web service*, o qual fornece as respostas no formato JSON.

As informações recebidas da *Weather Station* são tratadas na camada de negócios do aplicativo e convertidos em objetos, os quais são acessados pela camada de visualização.

O aplicativo Android conta com um pequeno banco de dados chamado SQLite, nativo da plataforma, no qual são persistidas as informações da última leitura. Desta forma, se não houver disponibilidade de acesso à *internet* no momento da inicialização do aplicativo serão exibidas as últimas informações salvas localmente.

A biblioteca SQLite é uma opção de escolha de banco de dados para dispositivos de recursos limitados como celulares, PDAs e MP3 *players* por ser muito compacta. Com todos os recursos ativados, o tamanho da biblioteca pode ser inferior a 500KB, dependendo da plataforma de destino e as configurações de otimização do compilador. Como pode ser executada em um espaço de pilha mínimo (4KB) ou mesmo pouca pilha (100KB), o

desempenho é geralmente muito bom, mesmo em ambientes com pouca memória (SQLITE, 2013).

A aplicação ainda conta com um sistema de alertas desenvolvido a partir da API *Google Cloud Messaging*, que possibilita que o servidor encaminhe mensagens curtas para os dispositivos que possuam o aplicativo *Weather Station* instalado (GCM, 2013). A API GCM é uma solução do tipo *push-based*, na qual, ao contrário da solução cliente-servidor (*pull-based*), é o servidor o responsável por notificar os assinantes quando da ocorrência de um evento (RAMOS, 2012).

O serviço *Google Cloud Messaging* (GCM) é gratuito e proporciona aos desenvolvedores o envio de dados do servidor para os aplicativos desenvolvidos na plataforma Android. Pode ser uma simples mensagem que informa ao aplicativo que existem dados mais recentes e que podem ser buscados no servidor ou mesmo uma mensagem de até 4 KB contendo dados úteis para o usuário (GOOGLE PLAY, 2013).

Para utilizar o aplicativo em um dispositivo Android, basta baixa-lo através do link <http://weatherstation-hitechdv.rhcloud.com/download/weatherstation.apk>. O aplicativo é compatível com *smartphones* e *tablets*.

A aplicação também conta com um serviço de alertas via SMS (*Short Message Service*) visando alcançar também as pessoas que não possuam dispositivos com o Sistema Operacional Android ou nem mesmo acesso à *internet*.

O SMS é um serviço que permite a troca de mensagens de texto curtas entre os usuários das redes de telefonia celular. Os dispositivos de telefonia móvel trocam mensagens com as torres de comunicação a fim de informar sua localização, de modo que a antena que melhor captar o sinal enviado fica responsável por fazer as ligações com o aparelho. Para viabilizar a entrega das mensagens SMS as operadoras de telefonia celular aproveitam essas trocas de informações obrigatórias e insere a mensagem entre os seus caracteres livres. Desta maneira as mensagens são entregues aos dispositivos destinatários praticamente sem nenhum custo para a operadora (LEITÃO, 2009), (TEIXEIRA, 2008).

## 4 – ANÁLISE DOS RESULTADOS

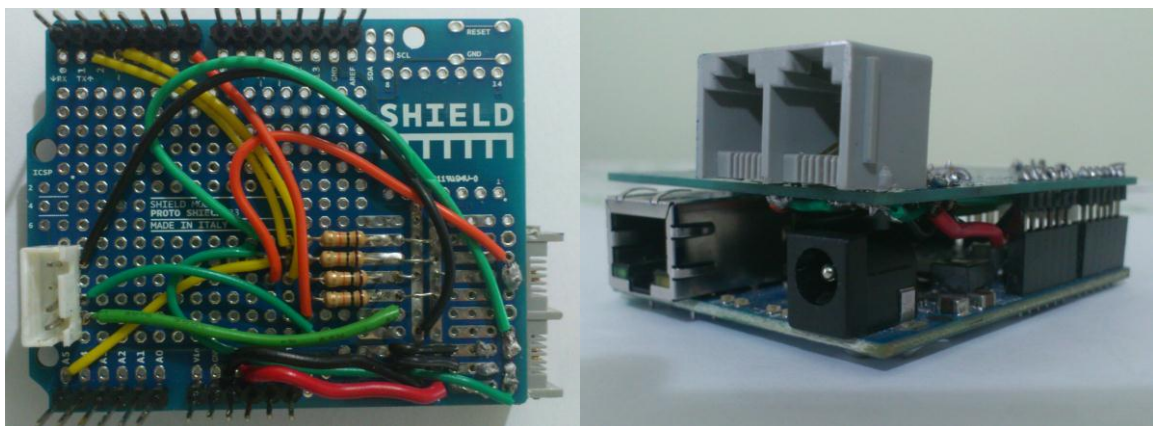
Neste capítulo serão apresentados e discutidos os resultados obtidos durante o desenvolvimento deste trabalho, a saber:

- Arquitetura de *Hardware* de Baixo Custo, item 4.1;
- Arquitetura de *Software Open-Source*, item 4.2;
- Aplicação para Interação com o Usuário, item 4.3;
- Estudo de Casos, item 4.4;
- Observações, item 4.5.

### 4.1 – Arquitetura de *Hardware* de Baixo Custo

O sistema de *hardware* desenvolvido para a captação das informações meteorológicas conta com uma placa do tipo Proto Shield, na qual foram montados os circuitos responsáveis pela interligação com os sensores. Todos os circuitos foram ligados a conectores a fim de facilitar a montagem do sistema. Foi utilizado um conector do tipo PEMC1-04 para o circuito do sensor DHT22, um conector do tipo RJ-11 para o circuito do sensor *rain gauge* e um conector do tipo RJ-11 para os circuitos dos sensores *anemometer* e *wind vane*. Também foram soldados vinte e oito pinos na placa Proto Shield visando possibilitar a conexão com a placa Arduino Ethernet de modo a dar acesso a todos os pinos digitais e analógicos para a montagem dos circuitos. Para acondicionar o sensor DHT22 foram utilizados dois copos plásticos contendo uma camada de cortiça entre eles a fim de proporcionar isolamento térmico contra os raios solares. A Figura 19 apresenta a Proto Shield e sua conexão com a placa Arduino Ethernet:

**Figura 19 – Proto Shield e sua conexão com a placa Arduino Ethernet**



Fonte: AUTOR, 2013

O módulo foi acoplado a uma caixa de PVC de maneira que seu *hardware* ficará protegido contra danos acidentais. A Figura 20 mostra o módulo montado e já instalado em uma caixa hermética, bem como o adaptador POE (*Power-Over-Ethernet*) utilizado para fornecer alimentação através do cabo UTP:

**Figura 20 – Módulo físico da *Weather Station* montado na caixa hermética**



Fonte: AUTOR, 2013

O módulo já acondicionado à caixa hermética foi fixado junto com os sensores em um cano de ferro. O sistema foi instalado na cobertura do prédio residencial denominado “Condomínio Aquarela”, na região Leste do município de São José dos Campos. A Figura 21 exhibe o sistema instalado:

**Figura 21 – *Weather Station* montada**



Fonte: AUTOR, 2013

Todo o *hardware* da *Weather Station* foi desenvolvido com materiais encontrados para aquisição no mercado ao menor custo possível. O custo total do *hardware* está discriminado no Quadro 3:

**Quadro 3 – Custo do *hardware* da *Weather Station***

<b>Componente</b>	<b>Quantidade</b>	<b>Valor Unitário</b>	<b>Valor Total</b>
Arduino Ethernet	1	US\$ 52,36	US\$ 52,36
Arduino Proto Shield	1	US\$ 3,94	US\$ 3,94
Resistor 10 KOhms	4	US\$ 0,12	US\$ 0,48
Conector RJ-11	2	US\$ 0,66	US\$ 1,32
Conector PEMC1-04	1	US\$ 1,32	US\$ 1,32
Caixa PVC	1	US\$ 1,50	US\$ 1,50
Caixa Hermética	1	US\$ 6,80	US\$ 6,80
Adaptador POE	1	US\$ 3,80	US\$ 3,80
Sensor DHT22	1	US\$ 4,80	US\$ 4,80
Sensor Wind Vane	1	US\$ 23,00	US\$ 23,00
Sensor Anemometer	1	US\$ 20,00	US\$ 20,00
Sensor Rain Gauge	1	US\$ 15,00	US\$ 15,00
Kit de montagem dos Sensores	1	US\$ 10,00	US\$ 10,00
Suporte	1	US\$ 4,60	US\$ 4,60
Cabo UTP	15 metros	US\$ 0,59	US\$ 8,85
Copo plástico	2	US\$ 0,50	US\$ 1,00
Fio Jumper	6	US\$ 0,33	US\$ 1,98
Fonte de Alimentação	1	US\$ 6,90	US\$ 6,90
Barra de Pinos Macho 180°	1	US\$ 0,66	US\$ 0,66
<b>Custo total</b>			US\$ 168,31

Fonte: AOSONG, ARDUINO, ARGENTDATA e EBAY, 2013

Como o custo para a construção do módulo físico é baixo, a própria população que vive em áreas de risco, na ausência de assistência dos órgãos governamentais, pode se mobilizar para implantar a solução apontada por este trabalho visando criar um sistema de monitoramento de desastres local, de modo a evitar situações como as ocorridas na região serrana do estado do Rio de Janeiro no ano de 2011, quando centenas de vidas foram perdidas (VEJA, 2011).



## 4.2 – Arquitetura de *Software Open-Source*

O módulo lógico da *Weather Station* é composto por um *software* gerenciador do Arduino que envia os dados captados do ambiente para um *Feed* disponibilizado na plataforma Xively e para o *Web Service* RESTful desenvolvido em Java, o qual foi construído sobre a plataforma OpenShift (RESTFUL, 2013), (OPENSIFT, 2013), (XIVELY, 2013).

As informações sobre as precipitações são captadas em tempo real e as demais são coletadas a cada quinze minutos. Tais registros são persistidos no banco de dados não relacional MongoDB operando sobre a plataforma OpenShift. A solução conta ainda com uma aplicação para acesso aos dados desenvolvida para o Sistema Operacional Android (MONGODB, 2013), (ANDROID, 2013).

Os recursos de *software* utilizados para o desenvolvimento do protótipo são baseados em plataforma aberta (*open-source*) a fim de proporcionar livre acesso a qualquer pessoa interessada em utilizá-los ou mesmo modificá-los.

O código fonte do projeto pode ser encontrado para consulta e utilização no repositório <https://github.com/jefuchoas/AWS/>.

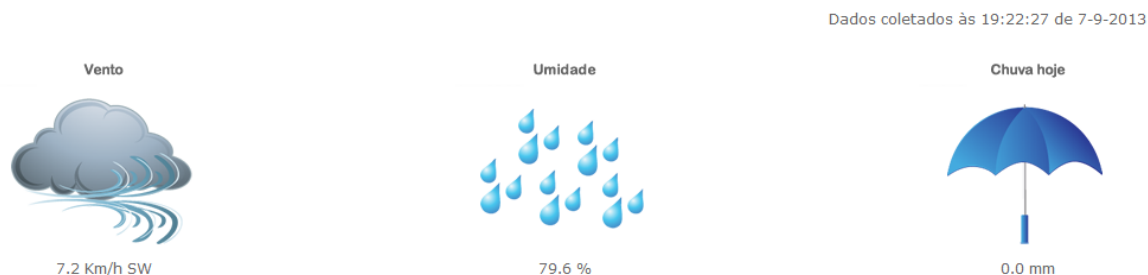
## 4.3 – Aplicação para Interação com o Usuário

A aplicação para interação com o usuário possui uma interface com acessibilidade (W3C, 2013). Nela são descritas as *urls* para requisições ao *Web Service*, bem como são disponibilizados os *links* para o *feed* publicado na plataforma Xively e um *link* para o *download* da aplicação desenvolvida para dispositivos Android.

No cabeçalho da página principal é exibido o logotipo da *Weather Station*, os botões de navegação e os dados da localização geográfica da sua instalação.

Logo abaixo do cabeçalho é exibido o horário e a data da última coleta de dados, bem como as informações sobre velocidade e origem do vento, umidade relativa do ar e a chuva acumulada do dia corrente. Na Figura 22 são mostradas estas informações:

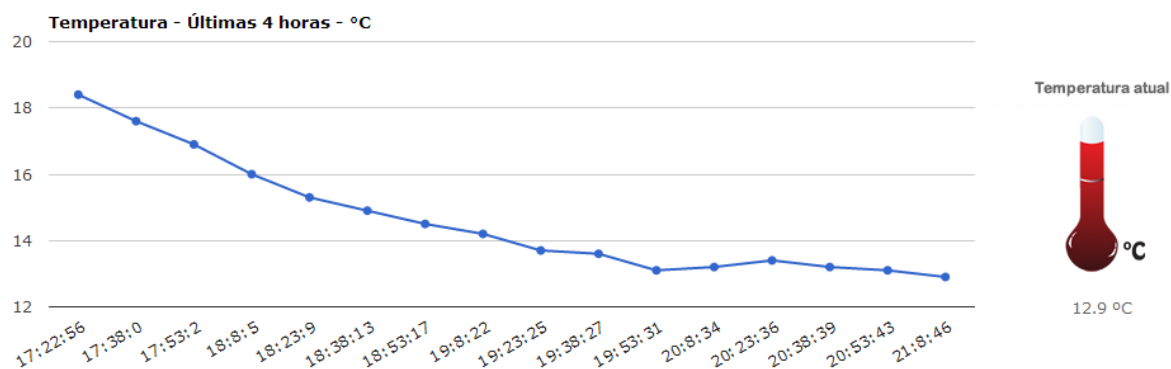
**Figura 22 – Informações sobre vento, umidade relativa do ar e chuva**



Fonte: AUTOR, 2013

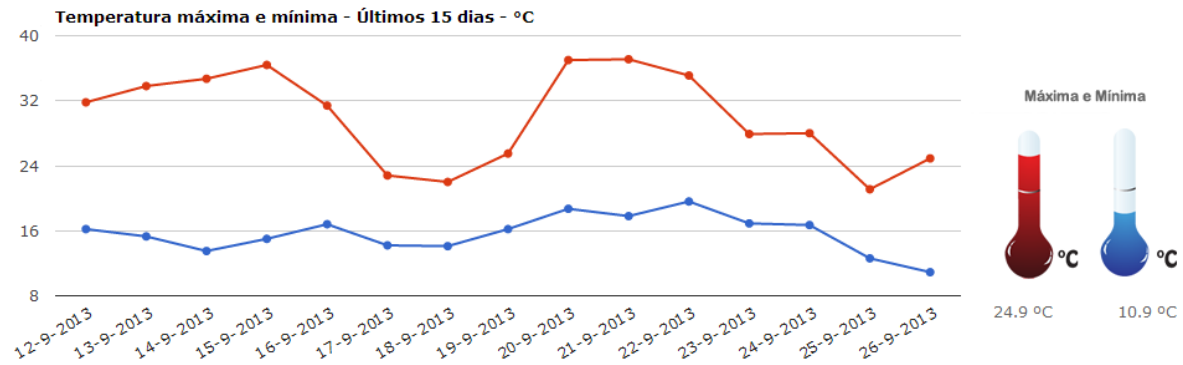
A interface apresenta um gráfico que aponta as últimas dezesseis leituras registradas na base de dados, o que equivale a quatro horas de registros, uma vez que o *hardware* envia os dados captados a cada quinze minutos. Ao lado do gráfico é exibida a informação sobre o registro mais recente de temperatura. A Figura 23 exibe as informações sobre temperatura:

**Figura 23 – Informações sobre temperatura**



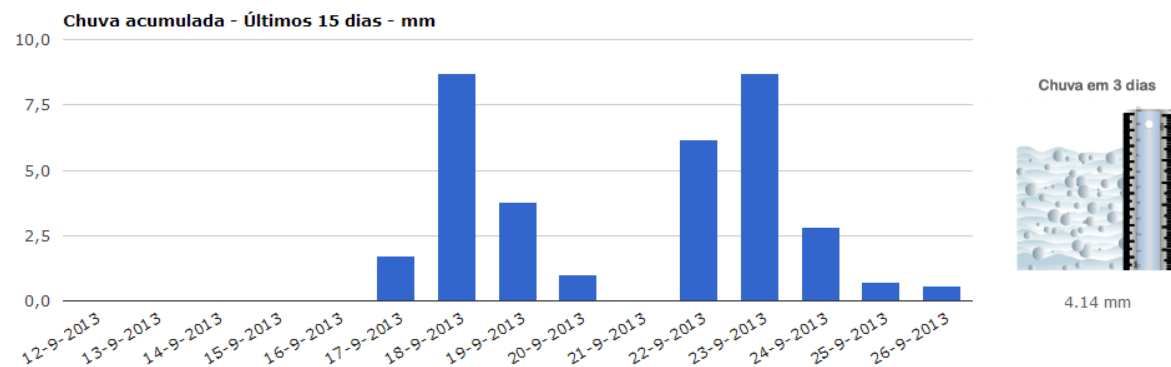
Fonte: AUTOR, 2013

O gráfico apresentado na Figura 24 exibe as informações sobre os últimos quinze registros de temperatura máxima e mínima, bem como o registro referente ao dia corrente.

**Figura 24 – Informações sobre temperatura máxima e mínima**

Fonte: AUTOR, 2013

O gráfico mostrado na Figura 25 apresenta os registros históricos de chuva acumulada nos últimos quinze dias de coleta. A chuva acumulada em três dias é utilizada como parâmetro para gerar alertas e desencadear ações visando à retirada das pessoas das áreas de risco, caso o total de chuva acumulada ultrapasse o valor histórico definido para a região. Ao lado do gráfico é exibido o valor acumulado de três dias. Como a inserção dos dados é dinâmica, é possível acompanhar a evolução da chuva em janelas de quinze minutos, de modo que torna viável uma ação preventiva em tempo hábil para evitar catástrofes.

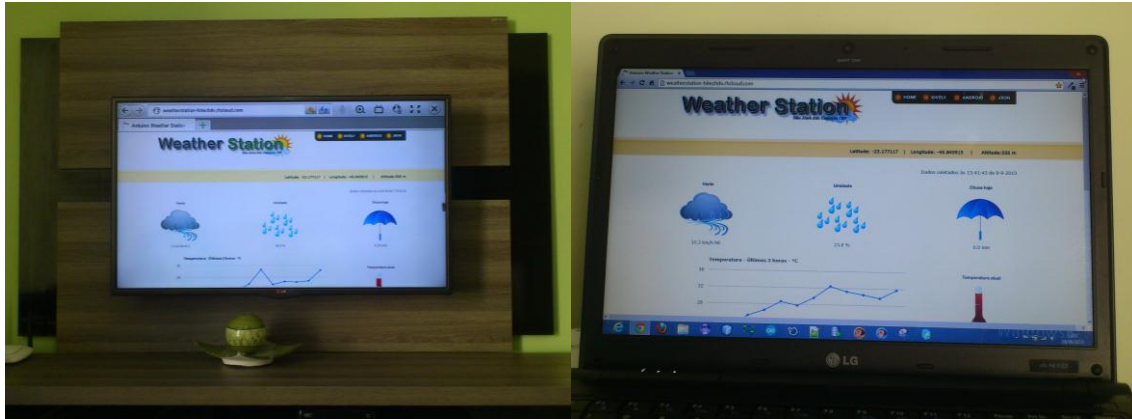
**Figura 25 – Informações sobre chuva acumulada**

Fonte: AUTOR, 2013

A interface *web* da *Weather Station* é compatível com todos os navegadores de *internet* utilizados na atualidade. Seu *layout* é perfeitamente adaptável aos mais variados tipos de dispositivo, como Notebooks, Smart TVs, Tablets, Smartphones, entre outros, proporcionando boas condições de leitura das informações, mesmo nas telas de menor

tamanho. A Figura 26 mostra o acesso à interface da *Weather Station* a partir de uma Smart TV e de um Notebook:

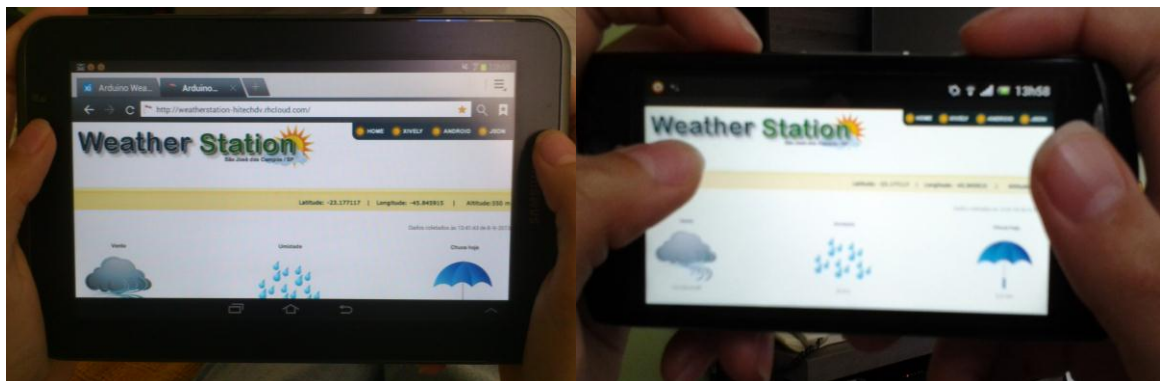
**Figura 26 – Acesso a partir de uma Smart TV e de um Notebook**



Fonte: AUTOR, 2013

O acesso à aplicação a partir de dispositivos com telas menores também é perfeitamente viável. A Figura 27 mostra o acesso a partir de um Tablet e de um Smartphone:

**Figura 27 – Acesso a partir de um Tablet e de um Smartphone**



Fonte: AUTOR, 2013

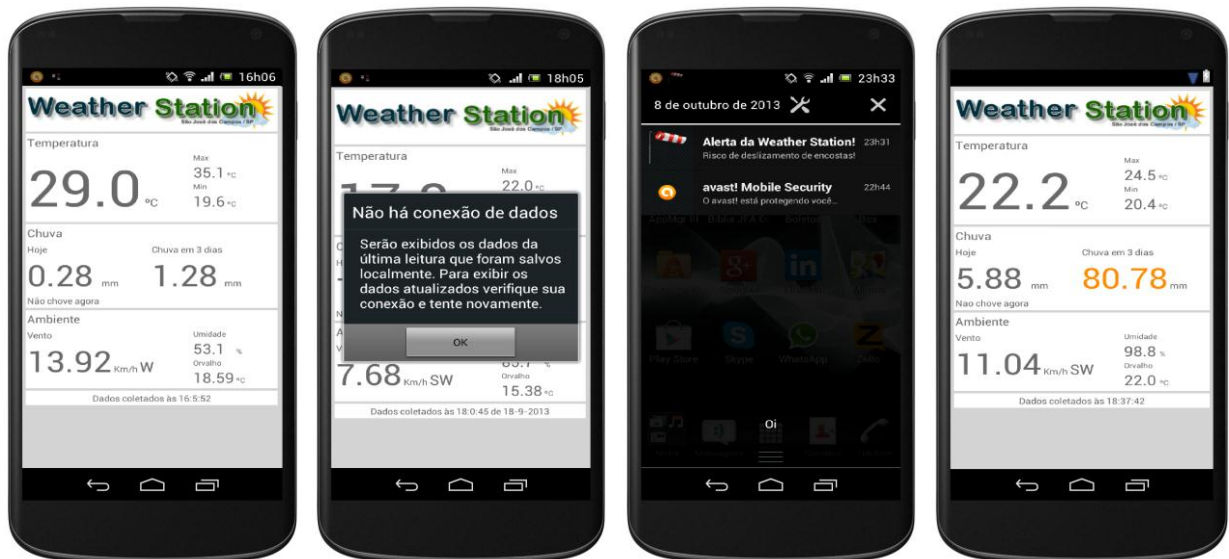
Além da interface *web*, a *Weather Station* possui uma aplicação para dispositivos Android, através da qual o usuário acessa em uma única página customizada para dispositivos móveis todos os dados disponibilizados pela estação, de forma clara.

Se não houver conexão de dados, serão exibidas as últimas informações gravadas na base de dados local, as quais foram coletadas durante a última leitura válida.

Ao se aproximar de um nível crítico a informação sobre o volume acumulado de chuva em três dias será exibida na cor laranja e, ao atingir o nível crítico, a informação será exibida na cor vermelha, de modo a alertar o usuário sobre o perigo de um deslizamento de terra nas áreas de risco apenas com a observação da tela.

Além disso, o sistema de alertas notifica o usuário, através de notificações do tipo *push* (RAMOS, 2012), quando o nível acumulado de chuvas de três dias ultrapassa a média histórica da região, gerando risco de deslizamento de encostas. A figura 28 apresenta respectivamente a tela inicial da aplicação com todas as informações coletadas do *web service*, a tela com a mensagem sobre a falta de conexão de dados, alertando o usuário que serão exibidos os dados da última leitura válida, a tela do dispositivo exibindo a notificação recebida do servidor e a tela exibindo as informações do nível de chuva acumulada na cor laranja, por ter se aproximado do nível crítico definido para o município de São José dos Campos:

**Figura 28 – Telas da aplicação para dispositivos Android**



Fonte: AUTOR, 2013

Ao ser executada pela primeira vez a aplicação se registra no serviço GCM e recebe um *id* do dispositivo (*device id*). Este *id* é então enviado ao *web service* e é armazenado na base de dados da *Weather Station* na coleção *devices* do banco MongoDB. No caso do usuário

não querer receber notificações poderá desativar o serviço através do menu “Alerta” existente na aplicação, o qual poderá ser reativado a qualquer momento. A Figura 29 ilustra a lista de dispositivos habilitados para o recebimento de alertas da estação:

**Figura 29 – Dispositivos registrados para o recebimento de alertas**

#2	Atualizar   Deletar   New Field   Duplicar   Atualizar   Texto   Expand
	<pre>{   "_id": ObjectId("52567970e4b0dd42e0575695"),   "deviceId": "APA91bEOxWjYvkzd29fd4bDU1un3Nx108EBVr- YqNfZxWN30ep1CPaKY8b0idRvC_UEQ6oYjJGU160dJ7EGK6Z7QUKvJQ0edvJQ6B12fnMviqkg8pvuQfmb4TzUQymRq4aUxUOWINghuAZ_87-90pl [...]" }</pre>
#1	Atualizar   Deletar   New Field   Duplicar   Atualizar   Texto   Expand
	<pre>{   "_id": ObjectId("525676afe4b0dd42e0575692"),   "deviceId": "APA91bFukJL6dwS99Q4hKaiwQo1vhYvqUD94TPrN59nwPDO1GYspLsxBFRQLUN46tg1eZoCZ_rAgjVjOtdMDS11LbY- CZZrdgZ6QSNgPTzuovS8d5ezvTTFtTJvV6rsQ8DWT_2knSyWncP8xuUWxqnk [...]" }</pre>

Fonte: AUTOR, 2013

Considerando o fato de que muitas pessoas não possuem dispositivos com o Sistema Operacional Android e devido à necessidade de que o maior número possível de moradores das áreas de risco tenha acesso à informação sobre as condições climáticas, também foi desenvolvido um sistema de alertas através de mensagens SMS disponibilizadas pelas operadoras de telefonia celular, as quais serão disparadas seguindo os mesmos critérios das notificações *push* do Android.

Para a utilização desta tecnologia a *Weather Station* está integrada com um *web service* dedicado ao envio de mensagens SMS disponibilizado gratuitamente pela empresa SSLti, sediada no município de São José dos Campos.

Para passar a receber os alertas gerados pelo sistema basta cadastrar o número do celular na interface *web* da *Weather Station*. Os celulares cadastrados permanecem salvos na coleção *cellphones* da estação até que o próprio usuário os remova da base de dados, o que também pode ser feito através da interface *web*.

O *web service* construído para a aplicação poderá ser utilizado e integrado por qualquer dispositivo, visto que utiliza a arquitetura RESTful e gera as informações no formato de *array* e objetos JSON. Através desta arquitetura qualquer desenvolvedor poderá utilizar as informações para a integração e o envio de dados para os mais variados tipos de dispositivos como o *Google Glass*, entre outros.

Mais que um sistema automático e de baixo custo para medição de índices meteorológicos, a *Weather Station* é uma solução que visa produzir alertas antecipados sobre

a possibilidade de ocorrência de desastres naturais em locais que podem colocar a vida das pessoas em risco de modo a minimizar suas consequências, bem como possibilita a integração de diversos sistemas que tenham o mesmo objetivo.

#### 4.4 – Estudo de Casos

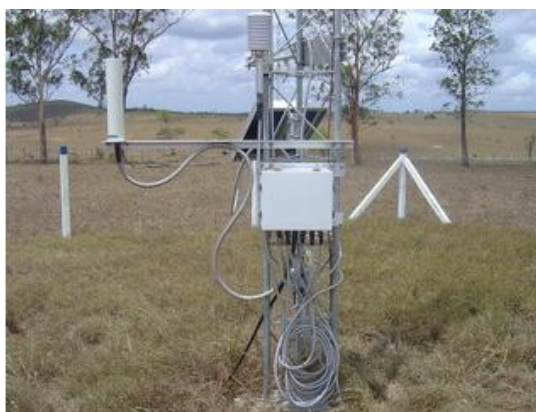
A fim de validar os dados coletados pela *Weather Station*, faz-se necessária a comparação com os modelos matemáticos e as amostras de dados observados pelos órgãos oficiais que operam os sistemas de coletas de informações do ambiente.

Tal abordagem visa dar credibilidade aos alertas gerados pela estação quando o índice de três dias de chuva acumulada atingir os níveis de risco definidos para o município onde está instalada.

Desta forma, foi escolhida, para a validação dos dados coletados, a informação disponibilizada pela Defesa Civil do município de São José dos Campos através do site da Coordenadoria Estadual de Defesa Civil, o qual pode ser acessado no endereço [http://www.defesacivil.sp.gov.br/v2010/portal\\_defesacivil/conteudo/meteorologia.asp?lk=3](http://www.defesacivil.sp.gov.br/v2010/portal_defesacivil/conteudo/meteorologia.asp?lk=3), contendo o registro dos dados observados.

Os dados inseridos no sistema da Coordenadoria Estadual de Defesa Civil são provenientes da Plataforma de Coleta de Dados (PCD) instalada no aeroporto “Professor Urbano Ernesto Stumpf”, no município de São José dos Campos, a qual é controlada pelo Centro Técnico Aeroespacial (CTA) e é parte da REDEMET. Tais dados são manipulados pela empresa Somar Tecnologia (SOMAR, 2013), (SINDA, 2013). Na Figura 30 se observa o exemplo de uma PCD:

**Figura 30 – Plataforma de Coleta de Dados**





Uma PCD moderna normalmente é destinada a interação com satélites e é equipada com baterias e células solares para o suprimento de energia. Atualmente existem diversos satélites que contam com equipamentos capazes de receber os dados das PCD e retransmiti-los para os Centros de Processamento em terra (INPE, 2013). A Figura 31 apresenta a localização e os dados da PCD do tipo Agromet do Sistema Nacional de Dados Ambientais (SINDA) do município de São José dos Campos, através da qual se observa seu código, usuário, tipo de estação e sua localização geográfica:

**Figura 31 – Metadados da PCD de São José dos Campos**

Estação		
Código: 30893	Nome: Sao Jose dos Campos	Usuário: INPE-FAPESP
Tipo: AGROMET	Município: Sao Jose dos Campos - SP	Status: Ativa
Altitude: 621 m	Longitude: -45.87º	Latitude: -23.20º

Alterar Gráfico (Selecione a variável ou a data)			
Variáveis:	Dia:	Mês:	Ano:
Precipitacao Acumulada	20	10	2013
<input type="button" value="ok"/>			

Dados Numéricos

Gráfico da Variável - Visualização para 3 dias (Use as setas para movimentar o período)

Período disponível: (2008-04-15 06h00) até (2013-06-23 09h00)

SINDA/INPE  
Estação: Sao Jose dos Campos (30893) Tipo: AGROMET

[Exibir mapa ampliado](#)

Fonte: SINDA 2013

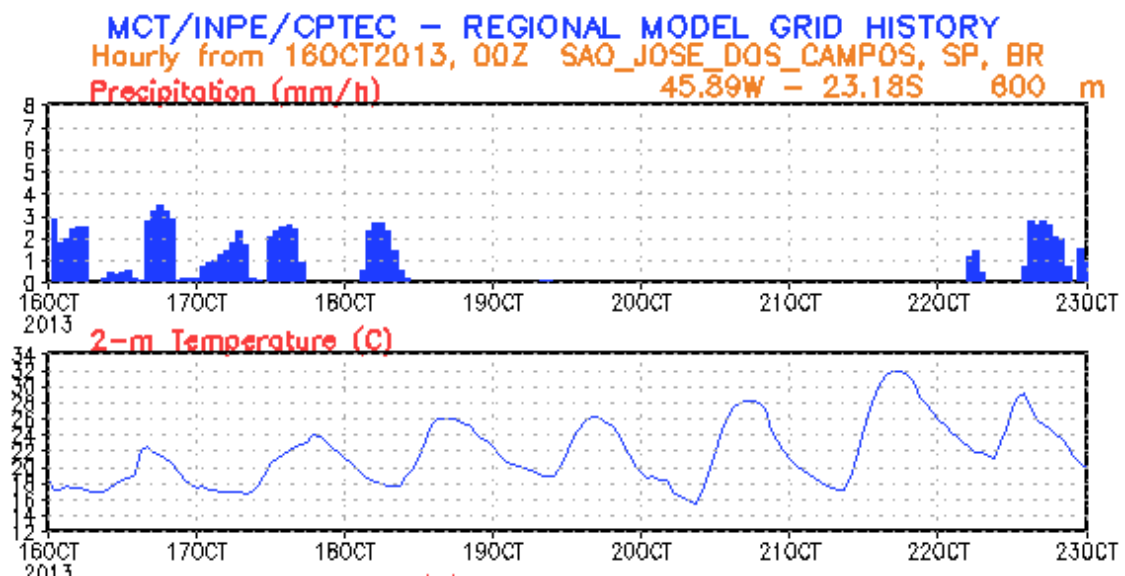
Além da validação através dos dados da Defesa Civil, as leituras da estação também foram validadas através do registro do meteograma de uma semana e dos dados observados, ambos disponibilizados pelo CPTEC-INPE no endereço <http://www.cptec.inpe.br/cidades/tempo/4963>. Um meteograma é um modelo matemático com a previsão das variáveis climáticas ao longo do tempo (INPE, 2013).



#### 4.4.1 – Comparação com um meteograma

Foi selecionada a semana de 16 a 23 de outubro de 2013 para uma primeira análise dos dados registrados pela *Weather Station*. O meteograma apresenta as estimativas de precipitação diária distribuídas em faixas de horário, de modo que há necessidade de se somar os valores previstos para se chegar ao valor diário. A Figura 32 apresenta o meteograma com precipitação e temperatura para a semana escolhida:

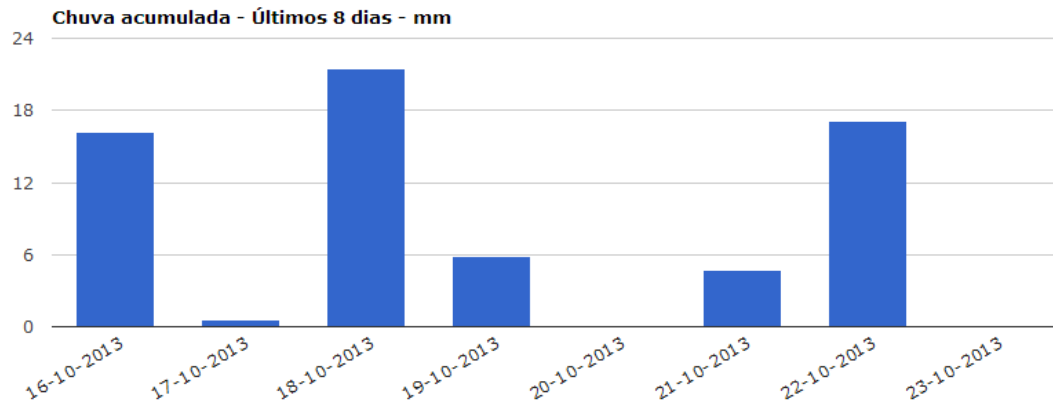
**Figura 32 – Meteograma de precipitação e temperatura**



Fonte: CPTEC 2013, INPE 2013

As variáveis do ambiente registradas pela *Weather Station* se aproximam dos valores estimados no meteograma do CPTEC-INPE. Observa-se que o gráfico mostra a possibilidade de chuvas no período de 16 a 18 de outubro, seguido de um período seco e novo período chuvoso nos dias 22 e 23. Tais estimativas foram confirmadas na maior parte dos dias, conforme se observa na Figura 33, a qual apresenta os dados de precipitação diária registrados durante a semana de 16 a 23 de outubro de 2013 pelo sistema:

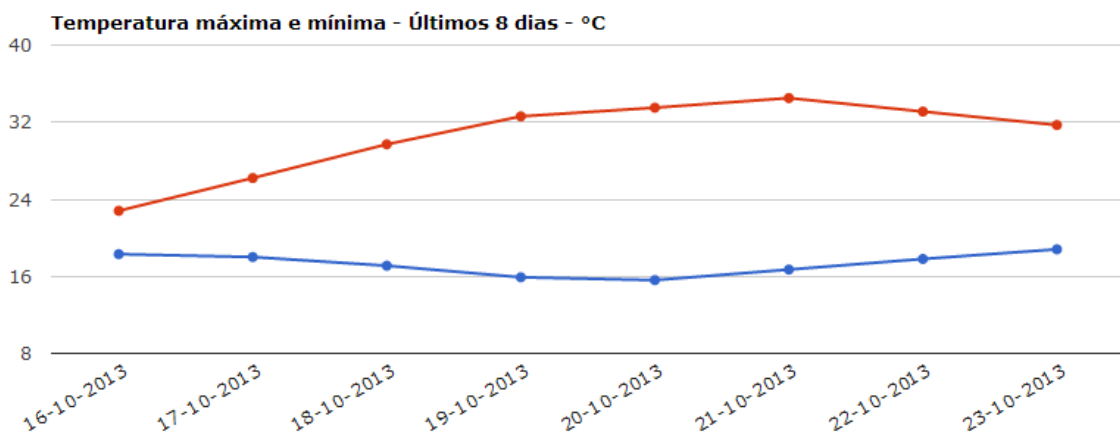
**Figura 33 – Dados de precipitação registrados pela *Weather Station***



Fonte: AUTOR, 2013

A Figura 34 apresenta a evolução da temperatura durante o período em análise. Esta ficou próxima da estimada no meteograma. A temperatura mínima girou em torno dos dezoito graus no período de 16 a 18 de outubro, chegando a dezesseis graus nos dias 19 e 20 e uma leve alta até próximo dos dezoito graus nos dias 22 e 23. Da mesma forma a temperatura máxima também acompanhou a evolução estimada no meteograma, atingindo o pico no dia 21:

**Figura 34 – Dados de temperatura registrados pela *Weather Station***



Fonte: AUTOR, 2013

#### 4.4.2 – Comparação com dados observados

Além do meteograma, também foram comparados os dados observados pela PCD da Redemet instalada no aeroporto de São José dos Campos. A leitura utilizada para a análise se deu às 21h00 de 20 de outubro de 2013. A Figura 35 apresenta os dados da PCD:

**Figura 35 – Dados registrados pela PCD**



Fonte: CPTEC 2013, INPE 2013

A Figura 36 apresenta as variáveis registradas pela *Weather Station* na mesma faixa de horário da leitura da PCD. Tanto a temperatura como a umidade relativa do ar apontam valores aproximados, apesar da distância entre as estações:

**Figura 36 – Dados registrados pela *Weather Station***



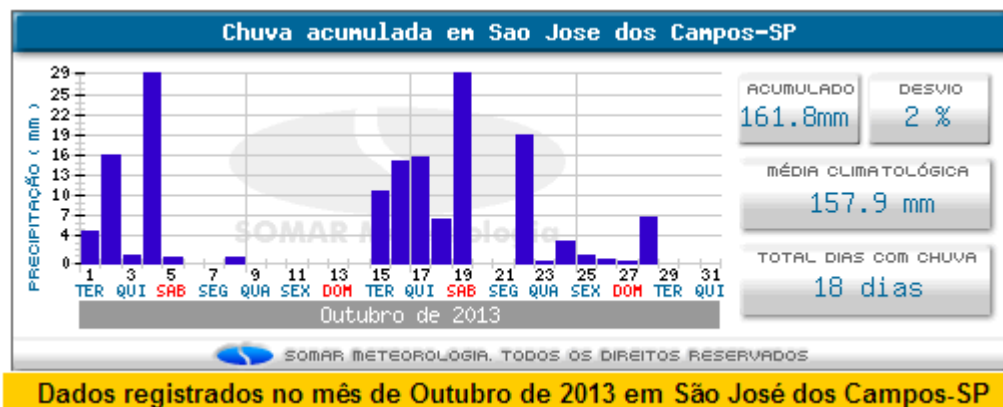
Fonte: AUTOR, 2013

A variável que mais se distanciou da medição da PCD foi a velocidade do vento, entretanto, há que se observar que o vento pode sofrer grandes variações, passando de valores muito baixos a valores elevados em questão de minutos.

#### 4.4.3 – Comparação com uma série mensal de dados

Uma maneira eficaz de se validar a leitura da *Weather Station* foi compará-la com o registro mensal de dados observados. Para tal análise foram utilizadas as informações oficiais do município de São José dos Campos. A Figura 37 apresenta os registros disponibilizados no site da Coordenadoria Estadual de Defesa Civil:

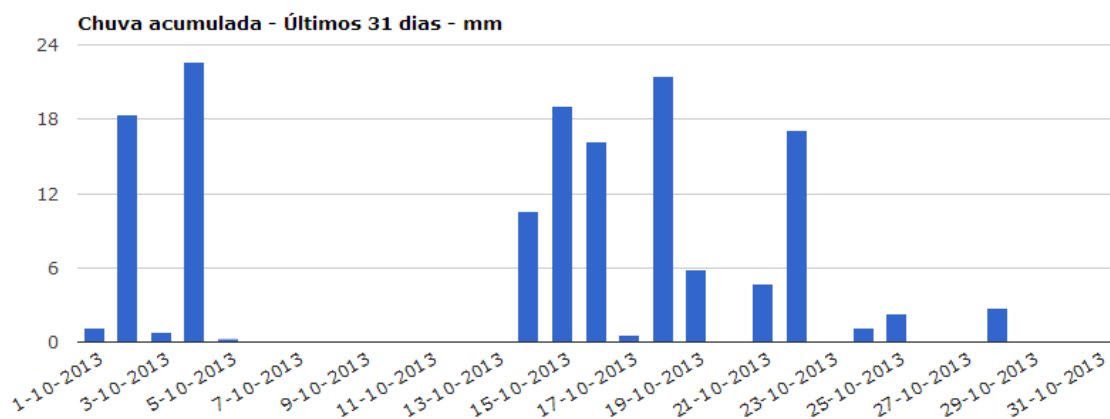
**Figura 37 – Precipitações do mês de outubro de 2013 - Defesa Civil Estadual**



Fonte: Coordenadoria Estadual de Defesa Civil 2013

Na Figura 38 se observa que os dados registrados pela *Weather Station* sobre o volume de precipitação no município de São José dos Campos se aproximam dos dados oficiais:

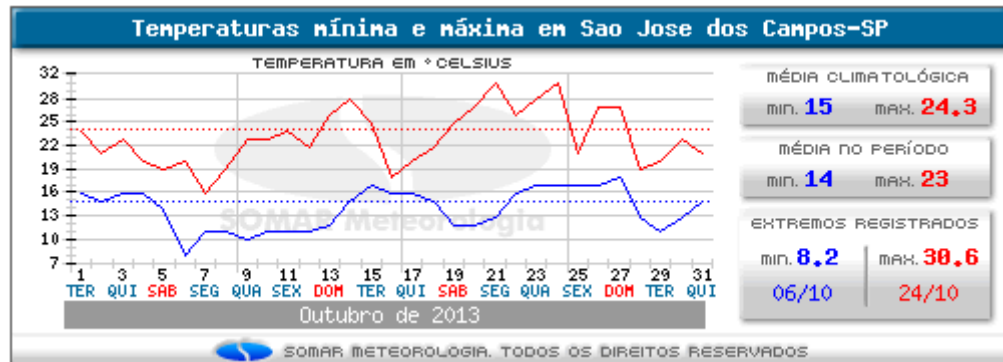
**Figura 38 – Precipitações do mês de outubro de 2013 - Weather Station**



Fonte: AUTOR, 2013

A Figura 39 apresenta os registros de temperatura mínima e máxima disponibilizados no site da Coordenadoria Estadual de Defesa Civil:

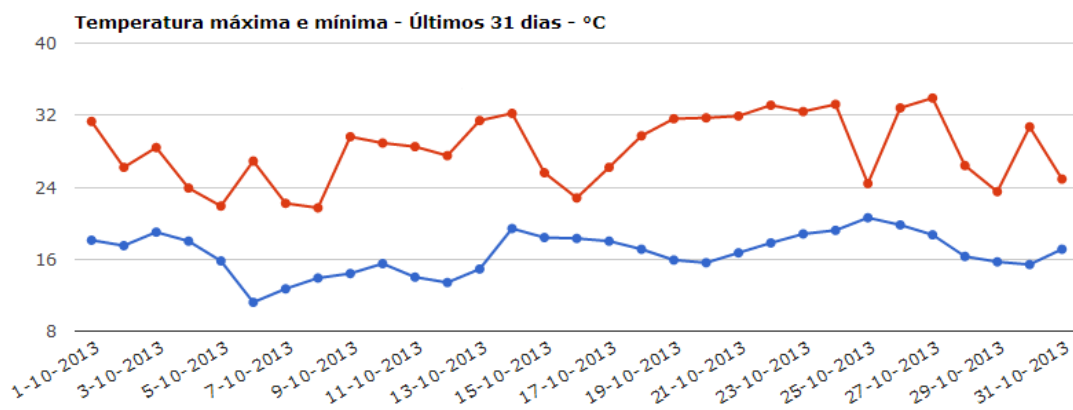
**Figura 39 – Temperatura do mês de outubro de 2013 - Defesa Civil Estadual**



Fonte: Coordenadoria Estadual de Defesa Civil 2013

Na Figura 40 se observa que os dados registrados pela *Weather Station* sobre as temperaturas máxima e mínima durante o mês de outubro de 2013 no município de São José dos Campos também se aproximam dos dados oficiais:

**Figura 40 – Temperaturas do mês de outubro de 2013 - Weather Station**



Fonte: AUTOR, 2013

#### 4.5 – Observações

Antes de qualquer comparação dos valores observados, cabe esclarecer que, embora a medição e os aparelhos utilizados para a aquisição de dados de chuva sejam simples e sua medição não seja complexa, é raro encontrar uma série confiável de dados pluviométricos.

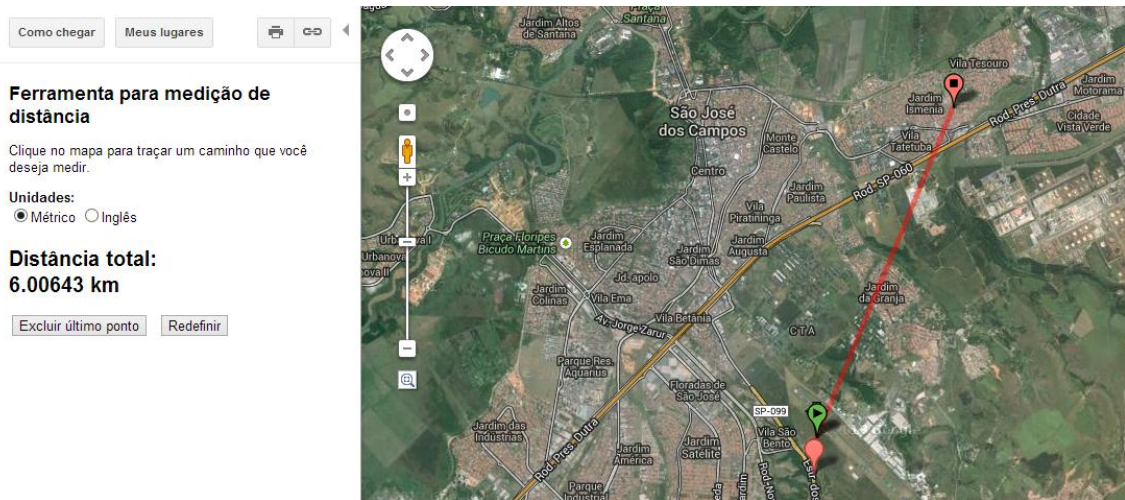
Como disse Chevallier, é necessário ter um bom conhecimento dos métodos de coleta, da correta forma de instalação dos aparelhos utilizados e ainda da personalidade dos observadores, antes de criticar ou de analisar a consistência dos dados (CHEVALLIER, 1993).

A Organização Meteorológica Mundial (OMM, 2013) define que a interceptação da chuva deve ser feita a uma altura média de 1 a 1,5 metros da superfície solo e que os dispositivos devem estar distantes de árvores, edificações ou qualquer outro obstáculo que possa interferir na coleta. No caso de não existir área livre, o equipamento deve ser instalado a uma distância de pelo menos duas vezes a altura do ponto onde está o obstáculo. Também não é aconselhável a instalação do dispositivo próximo a fontes de calor como superfícies de concreto, para que não haja interferência nos valores de temperatura do ar, nem em áreas muito abertas, visto que as correntes de vento podem influenciar na leitura.

Cabe observar que, por falta de espaço físico adequado, a *Weather Station* está instalada na cobertura do Condomínio Residencial Aquarela, situado na Zona Leste do município de São José dos Campos. Por estar instalada próximo ao telhado do condomínio, que é uma fonte de calor considerável, os valores de temperatura do ar medidos durante o período de incidência de luz solar sofreram interferência, apresentando valores acima da média.

Os níveis de chuva acumulada aferidos pela *Weather Station* estão próximos dos valores oficiais da Coordenadoria Estadual de Defesa Civil. A pequena diferença se dá devido à distância entre os pontos de coleta, visto que pode estar chovendo em uma região e em outro ponto do município não cair um pinga de chuva sequer. Foi utilizada a ferramenta para medição de distância do *Labs do Google Maps* para este cálculo. A Figura 41 apresenta a distância aproximada de 6 quilômetros entre as estações:

**Figura 41 – Distância entre a PCD e a Weather Station**



Fonte: MAPS 2013

Neste cenário observa-se que os dados registrados pela *Weather Station* são confiáveis, respeitadas as características do local onde está instalada, visto que apresentou grande similaridade com os registros oficiais. Diante disto, para que o sistema de alertas seja eficaz e alcance seu objetivo principal, ou seja, a antecipação da possibilidade de ocorrência de desastres naturais, há necessidade de que seja instalado nas proximidades das áreas de risco, onde ocorrem os eventos que causam os maiores impactos sobre a população.

## 5 – CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de um sistema automático e de baixo custo para medição de índices meteorológicos utilizando a *Internet das Coisas*. As contribuições e trabalhos futuros serão apresentados nas seções a seguir:

### 5.1 – Contribuições e Conclusões

As contribuições deste trabalho são:

- Possibilidade de monitoração automática dos eventos hidrológicos potencialmente causadores de desastres;
- Produção de alertas antecipados sobre a possibilidade de ocorrência de desastres naturais;
- Integração entre diversas plataformas de desenvolvimento através do *web service* RESTful que gera dados no formato JSON;
- Integração de informações em sistema capaz de subsidiar a comunidade e os órgãos públicos para a previsão e o controle dos efeitos negativos de eventos adversos sobre a população, os bens e serviços e o meio ambiente;
- Recursos de *hardware* e *software* baseados em plataforma aberta (*open-source*).

A partir destas contribuições pode-se concluir que:

- A *Internet das Coisas* possibilita a interação entre objetos e o ciberespaço de forma tal que a informação eletrônica seja produzida, distribuída, acessada e consumida a partir dos mais diversos dispositivos e objetos em todo e qualquer lugar;
- Um *web service* com a arquitetura RESTful que gera informações no formato de *array* e objetos JSON pode ser utilizado e integrado por qualquer dispositivo de modo que qualquer desenvolvedor pode utilizar as informações para a integração e o envio de dados para os mais variados tipos de dispositivo como o *Google Glass*, *smartphones*, *tablets*, entre outros;



- Por se tratar de plataforma aberta e de baixo custo para a construção, além de proporcionar livre acesso a qualquer pessoa interessada em utilizá-la ou mesmo modificá-la, na ausência de assistência dos órgãos governamentais, a população que vive em áreas de risco pode se mobilizar para implantar a solução;
- É perfeitamente viável a geração de alertas antecipados sobre a possibilidade de ocorrência de desastres naturais de modo a possibilitar que os moradores de áreas de risco possam agir em tempo hábil para evitar a perda de centenas de vidas.

A seguir são apresentadas experiências obtidas ao longo do desenvolvimento deste trabalho:

- É aconselhável que sejam seguidas as instruções da Organização Meteorológica Mundial quanto ao local e cuidados para a instalação do equipamento de coleta de dados para que se alcance um resultado confiável;
- O sensor pluviométrico deve ser instalado em uma superfície rígida, que não sofra vibrações, uma vez que se trata de um sensor constituído de um reservatório basculante cujo conteúdo é computado ao fechar um contato no circuito elétrico. Caso ocorra excesso de vibrações no suporte do sensor podem ocorrer contatos não esperados no circuito gerando valores errôneos na leitura dos níveis de chuva.

## 5.2 – Trabalhos futuros

As contribuições alcançadas com este Trabalho não encerram as pesquisas relacionadas à utilização da *Internet* das Coisas para o desenvolvimento de um sistema automático e de baixo custo para medição de índices meteorológicos, mas abrem oportunidades para alguns trabalhos futuros:

- Instalar a estação nas proximidades das áreas de risco para que o sistema de alertas seja mais eficaz;
- Recriar a estação utilizando alimentação por energia solar, de modo a garantir maior tempo de operação, economizar energia e utilizar recursos naturais;
- Recriar a estação na área de risco com sistema de alarme sonoro e/ou luminoso a fim de alertar a população quanto à possibilidade de deslizamento de encostas;

- Criar uma rede de estações nas áreas de risco e integrar suas leituras em um sistema único que possibilite ao usuário escolher qual estação irá acompanhar;
- Integrar o *web service* da *Weather Station* com um sistema de monitoramento fluviométrico a fim de possibilitar não só a geração de alertas sobre riscos de deslizamento de encostas, como também sobre os riscos de alagamentos e inundações;
- Integrar a aplicação com o *Google Glass*, possibilitando que os usuários possam ver todas as informações meteorológicas na lente do óculos;
- Adicionar recursos de geolocalização ao aplicativo Android de modo que o usuário receba automaticamente informações da estação mais próxima da sua residência;
- Adicionar um servidor de SMS próprio de modo a suprimir a dependência de um serviço de terceiros.

## REFERÊNCIAS

- AIRQUALITYEGG, **Airqualityegg**, Disponível em <http://airqualityegg.com/> Acessado em 18/05/2013.
- ANDROID, **Google**. Disponível em <http://developer.android.com/> Acessado em 17/03/2013.
- AOSONG. **Aosong**. Disponível em: <http://www.aosong.com/> Acessado em 12/09/2013.
- APACHE COUCHDB, **Apache CouchDB**. Disponível em <http://couchdb.apache.org/> Acessado em 05/03/2013.
- ARDUINO. **Arduino**. Disponível em: <http://Arduino.cc/> Acessado em 05/03/2013.
- ARDUINO WEATHER STATION. Disponível em <https://xively.com/feeds/187368934> Acessado em 12/09/2013.
- ARGENTDATA. **Agent Data**. Disponível em: <http://www.argentdata.com/> Acessado em 12/09/2013.
- AVELAR, E. A. M., AVELAR, L. M., SILVA, D. P., DIAS, K. L., *Arquitetura de Comunicação para Cidades Inteligentes: Uma proposta heterogênea, extensível e de baixo custo*, **Centro de Informática – Universidade Federal de Pernambuco (UFPE)**, 2012, p.1.
- BANZI, M, **Getting started with Arduino**, O’ Reilly Media Inc, 2009, p.19.
- BERNERS-LEE, T., FIELDING, R. T., GETTYS, J., MOGUL, J., FRYSTYK, H., MASINTER, L., LEACH, P., **Hypertext Transfer Protocol -- HTTP/1.1**, 1999 – Disponível em <http://www.hjtp.at/doc/rfc/rfc2616.html>, Acessado em 02/04/2013.
- CEMADEN - **CENTRO NACIONAL DE MONITORAMENTO E ALERTAS DE DESASTRES NATURAIS** - Disponível em: <http://www.cemaden.gov.br> Acessado em 26/02/2013.
- CHEN, Chiachung, **Measurement**, Elsevier, 2009.
- CHEVALLIER, P. *Aquisição e processamento de dados. In: TUCCI, C.E.M. Hidrologia: Ciência e aplicação*. Porto Alegre, Ed. **UFRGS/EDUSP/ ABRH**, 1993. p.485-525.
- COORDENADORIA ESTADUAL DE DEFESA CIVIL. Disponível em <http://www.defesacivil.sp.gov.br/> Acessado em 23/10/2013.
- CPTEC, **Centro de Previsão de Tempo e Estudos Climáticos**. Disponível em <http://www.cptec.inpe.br/> Acessado em 20/10/2013.
- DECRETO ESTADUAL Nº 42.565, **Estado de São Paulo**, 1 de dezembro de 1997.
- EBAY, **Ebay**. Disponível em <http://www.ebay.com/> Acessado em 08/05/2013.

FIEDLING, R. T. *Architectural Styles and the Design of Network-based Software Architectures. Dissertation* (Doctor of Philosophy in Information and Computer Science) - **University of California**, Irvine, 2000.

FREEMAN, Eric, FREEMAN, Elisabeth, **Padrões de Projeto, Série Use a Cabeça**, Alta Books – O'Reilly, 2005.

GAMA, K. S., *Integração de dispositivos de cidades inteligentes ao armazenamento de dados na nuvem*, **Centro de Estudos e Sistemas Avançados do Recife (CESAR)**, 2012, p.4-5.

GAMMA, Erich, JOHNSON, Ralph, HELM, Richard, VLISSIDES, John, **Padrões de Projetos: Soluções Reutilizáveis**, Bookman, 2006.

GCM, **Google Cloud Messaging** – Disponível em <http://developer.android.com/google/gcm/index.html>, Acessado em 06/10/2013.

GOOGLE PLAY, **Google Play** – Disponível em <https://support.google.com/googleplay/android-developer/>, Acessado em 06/10/2013.

GUARDIA, H. C., **Conhecimentos em Rede**, Departamento de Computação - Universidade Federal de São Carlos (UFSCar) - Tecnologias, Infraestrutura e Software, São Carlos, v. 1, n. 2, set-dez 2012 - ISSN 2316-2872, p. 98-101.

GUIMARÃES, C. L., COSTA, I. C., FURTADO, D. A. e KONIG, A., Gestão integrada de recursos hídricos, **Universidade Federal de Campina Grande – UFCG, Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq**, Capítulo 5, p.97-102, 2008.

INÁCIO, M. J., MAIA, R. D., NETTO, J. C., OLIVEIRA, H. R., SOUZA, T. A., CASTRO, F. T. e MARIA, D. G. S., Sistema embarcado baseado em hardware livre com utilização de computação nas nuvens aplicado no monitoramento de ambientes - **REVISTA PESQUISA & EXTENSÃO – V.2, NÚMERO 1, 2012 – ISSN 2236-7160**, P.41

INPE, **Instituto Nacional de Pesquisas Espaciais**. Disponível em <http://www.inpe.br/> Acessado em 20/10/2013.

JSON, **Json**. Disponível em: <http://www.json.org/> Acessado em 18/03/2013.

KÜCHLER, O. A., SILVA, A. C. T., MODRO, N. R. e MARTINS, J. E. M. P., Sistema para aquisição de dados pluviométricos - **XXX ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO**, 2010, p.2.

LEI FEDERAL nº 12.608, de 10 de abril de 2012.

LEITÃO, F. A. Leitão, FREIRE, S. S., LIMA, S. R., **Evolução e coexistência do serviço de mensagens SMS em IMS**, Universidade do Minho, Braga, Portugal, 2009.

LE MOS, A., JOSGRILBERG, F., **Comunicação e mobilidade - aspectos socioculturais das tecnologias móveis de comunicação no Brasil**, Universidade Federal da Bahia – EDUFBA, 2009, p.92.

LISBOA, C. L. V., **Uma Abordagem Comparativa entre as Arquiteturas de Web Services Tradicionais e Web Services Restful**, Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS), 2012, disponível em <http://gppd-wiki.inf.ufrgs.br/mediawiki/>.

MAPS, **Google Maps**. Disponível em <https://maps.google.com.br/> Acessado em 23/10/2013.

MINISTÉRIO DO MEIO AMBIENTE - Disponível em: <http://www.mma.gov.br/cidades-sustentaveis/urbanismo-sustentavel/preven%C3%A7%C3%A3o-de-desastres> Acessado em 02/03/2013.

MARCELINO, E. V. **Desastres naturais e geotecnologias: Conceitos Básicos**; INPE – INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS, 2007, p.6, 12-13.

MONGODB, **Projeto MongoDB**. Disponível em <http://www.mongodb.org/> Acessado em 17/05/2013; **Companhia MongoDB**. Disponível em <http://www.10gen.com/> Acessado em 17/05/2013.

OMM, **Organização Meteorológica Mundial**. Disponível em [http://www.wmo.int/pages/index\\_en.html](http://www.wmo.int/pages/index_en.html) Acessado em 20/10/2013.

OPENSIFT, **OpenShift**. Disponível em <https://www.openshift.com/> Acessado em 14/05/2013.

ORACLE, **Oracle**. Disponível em <http://www.oracle.com/> Acessado em 05/03/2013.

PROCESSING. **Processing**. Disponível em <http://www.processing.org> Acessado em 21/03/2013.

QUINTAS, R. V. C., **Plataforma de integração com rede das coisas (Web of things)**, Faculdade de Ciências, Universidade de Lisboa, 2010, p.4.

RESOLUÇÃO Nº 18-610, **Casa Militar do Estado de São Paulo - Cedec**, 23 de novembro de 2012.

RESTFUL, **RESTful Web Services**, disponível em <http://www.oracle.com/technetwork/>, <http://netbeans.org> e <http://www.ibm.com/developerworks/webservices/library/ws-restful/> Acessado em 19/03/2013.

RAMOS, J. M. M. G., *Livefeeds - Desenho e Avaliação de um Algoritmo de Filiação com Visibilidade Parcial*, Faculdade de Ciências e Tecnologia, **Universidade Nova de Lisboa**, 2012.

SABO, P.H., MARTINI, J. A. e GONÇALVES, R. A. L.: Rede de sensores sem fio para monitoramento frutícola, **IV EPAC - Encontro Paranaense de Computação ISSN:1981-8653**, p.134-135, 2011.

SADALAGE, P. J., FOWLER, M., **NoSQL Essencial**, Novatec 2013, ISBN: 978-85-7522-338-3.

SINDA, **Sistema Integrado de Dados Ambientais**. Disponível em <http://sinda.crn2.inpe.br/PCD> Acessado em 20/10/2013.

SMARTER CITIES, **IBM**. Disponível em [http://www.ibm.com/smarterplanet/us/en/smarter\\_cities](http://www.ibm.com/smarterplanet/us/en/smarter_cities) Acessado em 19/03/2013.

SOMAR, **Somar Meteorologia**, Disponível em <http://www.somarmeteorologia.com.br/> Acessado em 20/10/2013.

SOMMERVILLE, Ian; **Engenharia de Software**, 8ª Edição, Pearson Addison Wesley, 2007.

SOUZA JUNIOR, Ariovaldo - *Interface for User-Environment Interaction using Open Source Tools and Internet of Things*. **FATEC São José dos Campos**, 2012. 71f.

SQLITE, **SQLite**. Disponível em <http://www.sqlite.org/> Acessado em 29/09/2013.

STRAPAZZON, C. L., Convergência tecnológica nas políticas urbanas: pequenas e médias “cidades inteligentes” **Revista Jurídica, Curitiba, n. 22, Temática n. 6**, 2009, p.89-108.

TEIXEIRA, J., Língua Portuguesa e as Novas Tecnologias de Comunicação: as dinâmicas da(s) escrita(s), in *Diacrítica – Série Ciências da Linguagem*, Nº 22.1, 2008, pp. 107-127, **Centro de Estudos Humanísticos da Universidade do Minho**. ISSN 0807-8967

VEJA, **Acervo digital Veja**. Um banho de Lama na população. Edição 2200, de 19 de Janeiro de 2011, pág. 48 a 59. Disponível em <http://veja.abril.com.br/acervodigital/> Acessado em 12/09/2013.

W3C, **World Wide Web Consortium**. Disponível em <http://www.w3.org/> Acessado em 29/09/2013.

XIVELY, **Xively**. Disponível em <https://xively.com/> Acessado em 18/05/2013.