



**MÁRIO JORGE
ANDRADE TEIXEIRA**

**REDE DE SENSORES PARA RECOLHA E ANÁLISE
DE DADOS AMBIENTAIS**



**MÁRIO JORGE
ANDRADE TEIXEIRA**

**REDE DE SENSORES PARA RECOLHA E ANÁLISE
DE DADOS AMBIENTAIS**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. Joaquim Manuel Henriques de Sousa Pinto, Professor Auxiliar do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro e do Dr. Cláudio Teixeira, equiparado a Investigador Auxiliar do Departamento de Eletrónica Telecomunicações e Informática da Universidade de Aveiro.

o júri

presidente

Prof. Doutor José Manuel Matos Moreira
Professor Auxiliar, Universidade de Aveiro

Doutora Mariana Curado Malta
Professora Adjunta, Instituto Superior de Contabilidade e Administração do Porto

Doutor Cláudio Jorge Vieira Teixeira
Equiparado a Investigador Auxiliar, Universidade de Aveiro

agradecimentos

Um breve mas sentido agradecimento a todos os que me ajudaram ao longo do meu percurso até aqui, e um desejo de um dia poder retribuir em igual medida.

A todos, muito Obrigado.

palavras-chave

Software; arquitetura; rede; sensores; dados ambientais; informação; dinâmico.

resumo

Os fatores ambientais influenciam toda a atividade humana, podendo fazer a diferença entre o sucesso e o insucesso dessas atividades. Devido ao crescimento da quantidade de dados ambientais medidos, e do crescente número de tipos desses mesmos dados, têm sido cada vez mais utilizados sistemas de base eletrônica e informática para fazerem a recolha, o armazenamento e até a análise desses mesmos dados, permitindo aos seus utilizadores tomar decisões mais informadas nas suas mais diversas atividades.

Esta dissertação tem como principal objetivo a projeção e implementação de uma solução que permita a recolha de dados ambientais por uma rede sem fios de sensores, e o armazenamento e mostragem da informação recolhida dessa forma.

A solução projetada deve ser simplista, de baixo custo, e rápida e eficaz na aquisição de dados e no seu processamento e armazenamento. Deve ainda possibilitar ao utilizador a análise da informação recolhida de forma consistente e em tempo real, e de forma dinâmica considerando os tipos de dados recolhidos.

keywords

Software; architecture; network; sensors; environmental data; information; dynamic.

abstract

Environmental factors influence all kinds of human activity, sometimes making the difference between the success and failure of a certain undertaking. Due to the growth of the amount of environmental data being measured, and the growing number of different types of said data, systems based on electronics and informatics have been increasingly used to gather, store and even analyze this data, allowing users to make more informed decisions in their diverse activities.

This theses' main objective is to design and implement a solution that allows environmental data gathering by a wireless sensor network, and the storage and display of the information gathered by that network.

The designed solution should be a simple and low cost one, and it should be quick and effective in data acquisition, processing and storing. It should also allow the user to analyze the gathered information in a consistent, real time manner, and dynamically adjusting to the types of data gathered.

ÍNDICE

ÍNDICE.....	i
ÍNDICE DE FIGURAS.....	iii
ÍNDICE DE TABELAS	v
LISTA DE ACRÓNIMOS.....	vii
1. INTRODUÇÃO	1
1.1. Motivação	1
1.2. Objetivos.....	5
1.3. Metodologia de Desenvolvimento	6
1.4. Estrutura da Dissertação.....	7
2. ESTADO DA ARTE	9
2.1. Fire Systems.....	9
2.2. Vineyard Production Monitoring	13
2.3. OnFarm	17
2.4. AMI Turf Irrigation System.....	21
2.5. Uso de <i>ArcGIS</i> no auxílio à produção de Pêra Rocha.....	23
2.6. Análise Comparativa	25
3. CONCEÇÃO E DESENVOLVIMENTO.....	29
3.1. Equipamento	29
3.2. Requisitos.....	32
3.3. Arquitetura	35
3.4. Implementação e Protótipo	38
3.4.1. Modelo de Base de Dados	38
3.4.2. Web Services	41
3.4.3. <i>Gateway</i> da Rede – Monitor Série	43
3.4.4. Motes.....	45
3.4.5. Visualização de Dados.....	48
3.5. Testes de Comunicação.....	55
4. CONCLUSÕES E TRABALHO FUTURO	61

4.1. Conclusões.....	61
4.2. Problemas encontrados	62
4.3. Trabalho Futuro	63
4.4. Considerações Finais.....	64
5. BIBLIOGRAFIA	65
ANEXO A.....	69

ÍNDICE DE FIGURAS

Figura 1 Aplicação web Fire Systems para sistema indoor	9
Figura 2 Aplicação mobile Fire Systems para sistema indoor	10
Figura 3 Aplicação web Fire Systems para sistema outdoor.....	11
Figura 4 Aplicação mobile Fire Systems para sistema outdoor.....	11
Figura 5 Análise e divisão de zonas de um terreno para o Vineyard Procution Monitoring.....	14
Figura 6 Ecrã da ferramenta Sensitive	15
Figura 7 Ecrã da ferramenta Sensitive.....	16
Figura 8 OnFarm Grower Dashboard.....	18
Figura 9 Exemplo de gráfico do sistema OnFarm	18
Figura 10 Exemplo de alertas do sistema OnFarm	19
Figura 11 Mapa de horas de frio em ArcGIS.....	24
Figura 12 Gráfico do acumulado de horas de frio para vários locais em ArcGIS .	24
Figura 13 Dispositivo Waspote.....	29
Figura 14 Arquitetura do sistema para a recolha de dados.....	35
Figura 15 Arquitetura do sistema para a visualização de informação	36
Figura 16 Diagrama de sequência da recolha de dados	37
Figura 17 Modelo da base de dados	38
Figura 18 Parte do diagrama de sequência da recolha de dados para o servidor de Web Services	41
Figura 19 Interface da aplicação monitor de porta série	43
Figura 20 Interface da seleção da porta série	44
Figura 21 Código fonte de programa Waspote.....	46
Figura 22 Componente gauge.....	49
Figura 23 Gauge com alarme de valor mínimo	50
Figura 24 Gauge com alarmes de valor mínimo e valor máximo.....	50
Figura 25 Gauge com intervalo de valores definidos	51
Figura 26 Gauge com intervalo de valores definidos e um alarme.....	51
Figura 27 Componente chart.....	51
Figura 28 Componente chart com duas séries.....	52

Figura 29 Componente chart com duas séries com alarmes de valor mínimo.....	53
Figura 30 Componente mapa.....	53
Figura 31 Componente mapa com um valor atual que faz disparar um alarme ...	54
Figura 32 Comunicação, entre a rede e a aplicação através da porta série, de duas <i>frames</i> ASCII	56
Figura 33 Comunicação, entre a rede e a aplicação através da porta série, de duas <i>frames</i> binárias	57
Figura 34 Comunicação, entre a rede e a aplicação, de duas <i>frames</i> binárias e invocação do <i>webservice</i> de <i>parsing</i>	58
Figura 35 Comunicação entre a rede e a aplicação e invocação do <i>webservice</i> de <i>parsing</i> , conversão e armazenamento	58
Figura 36 Verificação da comunicação entre o servidor de <i>webservices</i> e a base de dados	59
Figura 37 Diagrama de Base de Dados	69

ÍNDICE DE TABELAS

Tabela 1 Tabela comparativa das soluções 1	25
Tabela 2 Tabela comparativa das soluções 2	27
Tabela 3 Tabela representativa da forma de criação de componentes.....	49
Tabela 4 Entidade Site	70
Tabela 5 Tabela Node	70
Tabela 6 Tabela Sensor	71
Tabela 7 Tabela SensorAlarm.....	72
Tabela 8 Tabela ValueType	72
Tabela 9 Tabela TypeRange	73
Tabela 10 Tabela SensorData.....	73
Tabela 11 Tabela SensorMoteSensor	74
Tabela 12 Tabela MoteSensor	75
Tabela 13 Tabela SensorTable	75
Tabela 14 Tabela FieldType.....	76

LISTA DE ACRÓNIMOS

ALADIN	A ire L imitée A daptation dynamique D éveloppement I nter N ational
API	A pplication P rogramming I nterface
ASCII	A merican S tandard C ode for I nformation I nterchange
ESRI	E nvironmental S ystems R esearch I nstitute
GPRS	G eneral P acket R adio S ervice
HTTP	H yper T ext T ransfer P rotocol
IDE	I ntegrated D evelopment E nvironment
IDW	I nverse D istance W eighting
IEEE	I nstitute of E lectrical and E lectronics E ngineers
IoT	I nternet- o f- T hings
PC	P ersonal C omputer
RTC	R eal T ime C lock
SaaS	S oftware- a s- a - S ervice
SD	S ecure D igital
SIG	S istema de I nformação G eográfica
SNR	S ignal-to- N oise R atio
SOAP	S imple O bject A ccess P rotocol
SQL	S tructured Q uery L anguage
URL	U niform R esource L ocator
USB	U niversal S erial B us
WSN	W ireless S ensor N etwork

1. INTRODUÇÃO

1.1. Motivação

Toda a atividade humana é afetada por fatores ambientais, como por exemplo a temperatura, a humidade, a qualidade do ar, entre outros. Estes fatores podem por vezes fazer a diferença entre o sucesso e o insucesso de um determinado empreendimento. No entanto são na maioria dos casos fatores mensuráveis, e o Homem desenvolveu ao longo do tempo meios para os medir, e capacidades para os analisar.

Sendo possível medir estes fatores e fazer a recolha dos dados ambientais daí provenientes, e ainda havendo o conhecimento para fazer a análise dos mesmos, transformando-os em informação útil e relevante, podem ser tomadas decisões que tornem determinada atividade menos susceptível de ser influenciada negativamente por um destes fatores ou sua variação, ou até fazer com que essa atividade beneficie de determinado contexto ambiental, ou adaptá-la a uma mudança de um desses aspetos.

Este processo de recolha e análise de dados ambientais é usado em diversos contextos e de diversas formas. No entanto, a capacidade humana nem sempre é suficiente para explorar ao máximo as possibilidades que este processo pode dar. Por exemplo, a recolha de dados de múltiplas localizações, num contexto em que é necessário cobrir uma grande área, pode tornar-se demasiado lenta, mais ainda se considerarmos a possibilidade de algumas destas localizações serem de difícil acesso. Do ponto de vista da análise dos dados recolhidos, um exemplo de recolha de dados como o anterior vai gerar grandes quantidades de dados, difíceis de separar, filtrar e posteriormente analisar de forma integrada para gerar informação útil e em tempo útil.

Como resposta a este problema, surge de forma cada vez mais difundida a utilização de equipamentos e sistemas informáticos para a recolha e tratamento de dados ambientais, em áreas muito variadas como a engenharia civil, a

indústria, a agricultura, o meio ambiente, entre outros (Barroca, Borges, Fernando, Monteiro, Górski. & Castro-Gomes, 2013 ; Sung & Hsu, 2011 ; Lo, Lynch & Liu, 2015 ; AlSkaif, Zapata & Bellalta, 2015 ; Dener & Bostancioğlu, 2015).

A utilização de sistemas informáticos permite resolver problemas como a recolha de dados de múltiplas localizações de forma simultânea ou o tratamento de grandes quantidades de dados, assim facilitando a sua análise, e aumentando a velocidade de todo este processo.

Estes sistemas de informação são feitos, tipicamente, com base em *Wireless Sensor Networks*, ou WSN. As WSN são formadas por pequenos dispositivos que incorporam um micro-controlador, uma ou várias unidades de memória, uma unidade de alimentação, um ou mais módulos de comunicação, e utilizam sensores para a obtenção de dados. Estes dispositivos são designados por *motes*, e são capazes de ler informação através dos sensores que possuem e comunicar os valores obtidos entre si e/ou para o exterior da rede (Barroca et. al, 2013).

As WSN destacam-se sobretudo devido à sua facilidade de utilização e ao seu baixo custo, e o interesse crescente nestas redes pode ser atribuído diretamente aos benefícios da não-necessidade de instalar fios e à sua aplicabilidade em múltiplos contextos, tão distintos como, por exemplo, o controlo de tráfego ou aplicações da área da saúde (Lo et. al, 2015 ; AlSkaif et. Al, 2015). Estas redes têm ainda outras características que potenciam o seu uso para o fim pretendido, como o seu baixo consumo, o facto da sua instalação ser sem fios permite não danificar o ambiente no caso de uma instalação ao ar livre, e ainda a possibilidade de re-programar dispositivos sem ser necessário aceder fisicamente ao local onde este está instalado (Dener et. al, 2015).

Os protocolos usados tipicamente em WSNs são protocolos derivados do padrão IEEE 802.15.4. Este padrão e os protocolos dele derivados têm potenciado a proliferação de WSNs nas mais variadas áreas, fundamentalmente devido ao seu baixo consumo energético e o seu baixo custo (Chaloo, Oladeinde, Yilmazer, Ozcelik & Chaloo, 2012). Para além destas vantagens, as redes do tipo

802.15.4 são as que funcionam melhor em ambientes com SNR baixo (Lian, Hsiao & Sung, 2013).

Entre as várias áreas de aplicação de WSNs, podemos destacar:

- Engenharia Civil;
- Segurança em Ambiente Industrial;
- Monitorização do Ambiente;
- *Tracking*;
- Produção Agrícola;
- Controlo de Tráfego;
- Saúde;
- Monitorização de Localizações Remotas;
- Monitorização do Comportamento dos Consumidores.

Na área da Engenharia Civil, estas redes são usadas na monitorização de estruturas, por exemplo para medir a temperatura e humidade em estruturas de betão, o que permite detetar antecipadamente problemas com a estrutura, antecipando o planeamento e a execução de possíveis programas de manutenção, minimizando os custos relevantes do processo. Assim, os sensores e os sistemas de monitorização associados formam um elemento importante na inspeção e gestão deste tipo de estruturas (Barroca et. al, 2013 ; Lo et. al, 2015).

Na área da Segurança em Ambiente Industrial, este tipo de redes são usadas para medir vários parâmetros como a temperatura ou a concentração de dióxido de carbono para aumentar a segurança dos trabalhadores no ambiente de trabalho (Sung et. al, 2011).

Na Monitorização do Ambiente, temos, entre outros, sistemas de medição de parâmetros meteorológicos, sistemas de prevenção de fogos e sistemas de medição de poluição ambiental (Lo et. al, 2015 ; AISkaif et. al, 2015 ; Dener et. al, 2015).

Na área do *Tracking*, redes deste tipo são usadas para fazer tracking quer a objetos, quer a animais (Lo et. al, 2015 ; AISkaif et. al, 2015).

As WSN são muito usadas em aplicações na área da Agricultura, com aplicações em ambiente de agricultura de precisão, no controlo da irrigação, por exemplo (Ojha, Misra & Raghuwanshi, 2015).

Outras áreas de aplicação incluem o Controlo de Tráfego, aplicações na área da Saúde, Monitorização de Localizações Remotas ou Monitorização do Comportamento dos Consumidores (AlSkaif et. al, 2015 ; Dener et. al, 2015), o que mais uma vez demonstra a aplicabilidade deste tipo de redes sensoriais a vários contextos bastante diversificados, e mostra a sua flexibilidade.

1.2. Objetivos

Partindo do que foi discutido no sub-capítulo anterior, observa-se que a utilização de sistemas informáticos, nomeadamente apoiados por redes do tipo WSN, para a obtenção e processamento de dados ambientais nos mais variados contextos é não só possível como aconselhável, já que o poder dos atuais sistemas de informação ultrapassa em larga escala aquilo que é possível fazer, em termos de recolha e de análise de dados, por uma ou várias pessoas, e portanto a sua utilização para esse fim traz mais-valias inegáveis a quem os utiliza.

Apesar da existência de muitas soluções funcionais em várias áreas, existe ainda um grande espaço para a entrada de novas soluções e novos produtos, tendo em conta a especificidade de cada contexto, por exemplo no que toca aos parâmetros a medir e à dimensão da rede pretendida.

Pretende-se nesta dissertação desenvolver uma solução que seja dinâmica para se adaptar facilmente aos vários contextos em que possa vir a ser utilizada, e que seja de baixo custo para permitir a viabilidade da sua utilização por pessoas e empreendimentos com poder económico médio e médio-baixo, e que lhes permita potenciar as suas atividades e/ou os seus produtos com um aumento de qualidade significativo.

Para isso, pretende-se projetar e implementar redes de sensores do tipo WSN para a obtenção de dados ambientais, e um sistema que permita a recolha dos mesmos, e o processamento e armazenamento de informação deles obtida. Deve existir comunicação em tempo real entre as WSN e o sistema, que deve ser capaz de os registar e de agir com base neles. Pretende-se ainda desenhar algumas técnicas de visualização da informação que possam ser interessantes neste contexto.

Tudo isto pretende-se numa solução de baixo custo, simplista, que seja fiável em funcionamento em tempo real, e que seja o mais dinâmica possível e facilmente escalável.

1.3. Metodologia de Desenvolvimento

Para atingir os objetivos propostos para esta dissertação, e que foram enunciados em maior detalhe no sub-capítulo anterior, nomeadamente a projeção e implementação de redes WSN e de um sistema de recolha, processamento e armazenamento dos mesmos, para além de serviços e formas de visualização da informação obtida, foi utilizada uma metodologia estruturada mas não rígida, e com uma abordagem de implementação sobretudo incremental.

A necessidade da criação efetiva de uma solução tangível, com capacidades comprovadas na realização das tarefas pretendidas com a obtenção dos resultados pretendidos, dentro de um quadro temporal com restrições, bem como a abrangência de meios, componentes, e tecnologias diferentes a utilizar numa solução completa como a pretendida, são os fatores principais que levaram à utilização desta metodologia e deste modelo de metodologia.

Esta metodologia e esta abordagem levam a que, após a projeção da arquitetura geral, seja projetada e implementada uma parte de cada vez, pretendendo-se que cada um destes incrementos venha aumentar as capacidades do sistema em lidar com os dados obtidos e em transmitir esses mesmos dados de forma encadeada até à fase do seu armazenamento, tal como arquitetado anteriormente.

1.4. Estrutura da Dissertação

Esta dissertação está dividida em **cinco capítulos**, dos quais esta introdução é o **primeiro**.

O **segundo capítulo** apresenta algumas soluções e produtos com os mesmos moldes, de algumas áreas que se entendem particularmente relevantes.

No **terceiro capítulo** é descrita a concepção e implementação do sistema e dos seus vários componentes.

O **quarto capítulo** contém algumas conclusões e considerações sobre o trabalho realizado.

Por último, no **quinto capítulo** estão as referências bibliográficas.

2. ESTADO DA ARTE

Neste capítulo vamos apresentar algumas soluções estudadas e alguns produtos e serviços comercializados com base neste tipo de redes sensoriais, nos princípios enunciados no capítulo 1 e nos objetivos de recolha, tratamento e análise de dados ambientais.

Estas soluções foram encontradas com base numa revisão de literatura, e ainda, em alguns casos, com o auxílio dos orientadores.

2.1. Fire Systems

Dener, Özkök & Bostancıoğlu (2015) criaram uma solução de sistema orientado para a monitorização ambiental, tendo como objetivo a prevenção de fogos, denominado por *Fire Systems*.

O *Fire Systems* baseia-se numa *Wireless Sensor Network*, com vários nós a transmitir dados para o sistema, que os armazena e pode ainda apresentá-los em tempo real ao utilizador. Os nós da WSN usam a tecnologia ZigBee, baseada no padrão IEEE 802.15.4, para a comunicação com o *gateway* do sistema.

Este sistema monitoriza possíveis aumentos de temperatura, e permite assim a prevenção de fogos. A aplicação de visualização de dados corre nas plataformas web e mobile.

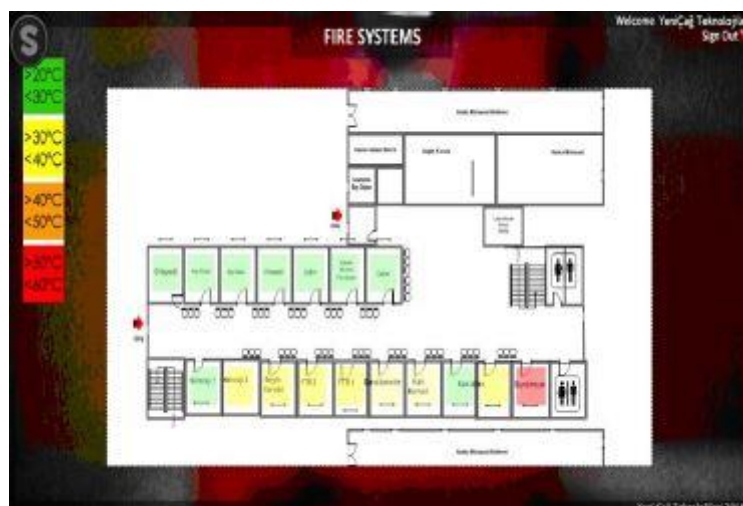


Figura 1 Aplicação web Fire Systems para sistema indoor

Novamente, a monitorização da informação recolhida pelo sistema pode ser feita através da plataforma web ou da aplicação mobile, da forma ilustrada pelas Figuras 3 e 4.

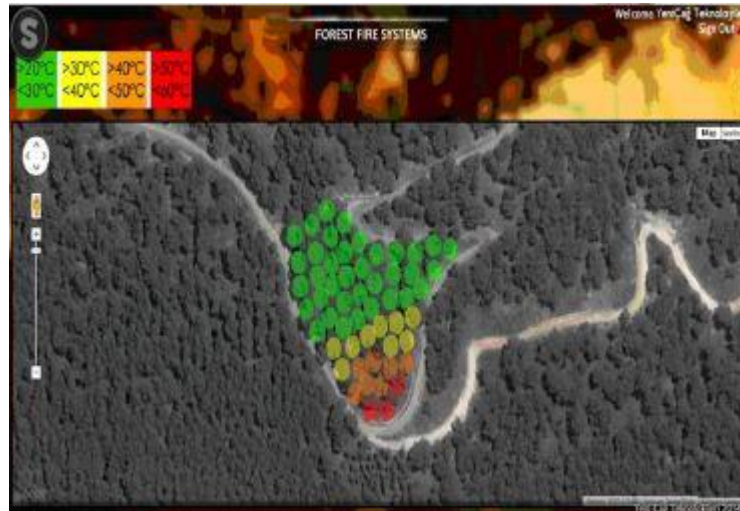


Figura 3 Aplicação web Fire Systems para sistema outdoor

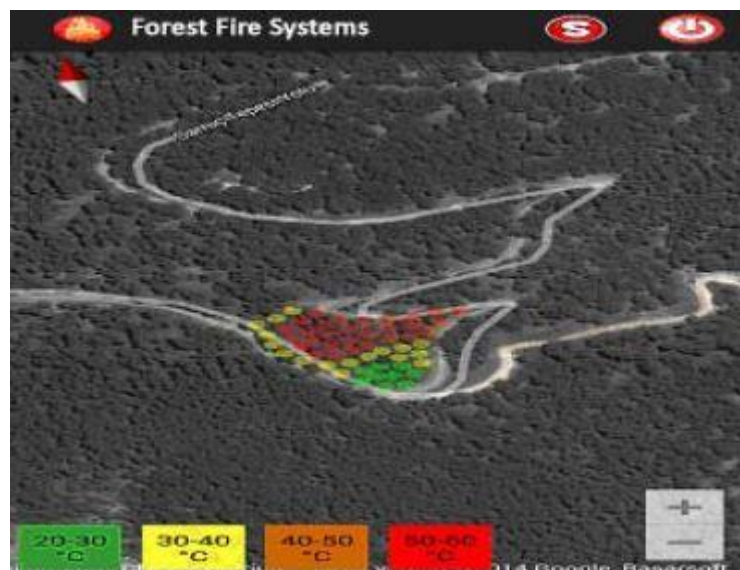


Figura 4 Aplicação mobile Fire Systems para sistema outdoor

Repare-se na apresentação da escala de cores:

- As zonas a verde são zonas em que a temperatura está dentro do normal;
- As zonas a amarelo são zonas em que a temperatura está ligeiramente acima do normal;

- As zonas a laranja são zonas em que a temperatura pode indicar uma situação anormal ou de perigo;
- As zonas a vermelho são zonas em que a elevada temperatura indica a existência de perigo.

Comparando o *Fire Systems* com outros sistemas de segurança de incêndio, verifica-se que a facilidade de uso do sistema e o seu baixo custo, conseguidos através do uso de WSN, são vantagens muito interessantes em ambiente de aplicação real (Dener et. al, 2015).

2.2. Vineyard Production Monitoring

Díaz, Pérez, Mateos, Marinescu & Guerra (2011) procuraram criar não apenas um sistema de monitorização da produção de vinhas, mas toda uma metodologia para a criação de sistemas suportados por WSNs para a monitorização de parâmetros ambientais em contexto de produção agrícola.

Este interesse entende-se pela importância que a utilização deste tipo de redes sensoriais pode ter na melhoria dos resultados de produções deste setor, já que a produção tem uma dependência muito forte das condições ambientais existentes. Assim, o investimento de produções agrícolas na informatização e na colocação deste tipo de redes e sistemas será um garante da obtenção de maior quantidade e qualidade de produtos, sendo particularmente úteis em produções em que se procura a maior qualidade possível do produto, como é o caso do setor vinícola.

Assim, ao aplicar este tipo de tecnologias à monitorização de vinhas, como um sistema de agricultura de precisão, pretende-se aumentar a qualidade da produção e reduzir os custos da mesma, além de reduzir o efeito de doenças que ataquem a colheita (Ojha et. al, 2015). A colocação de WSNs ajuda a estimar a variabilidade de parâmetros importantes para a atividade agrícola ao longo do tempo, o que permitirá atingir estes objetivos.

A definição e instalação da rede sensorial deve ser adaptada a cada local, e para isso um terreno onde se pretenda a instalação do sistema é primeiramente avaliado, com base em fatores geográficos, meteorológicos e de acordo com as características do solo, e dividido em zonas, de acordo com a semelhança dos fatores avaliados. Um exemplo de um exercício de análise e divisão de um terreno em zonas pode ser visto na Figura 5.

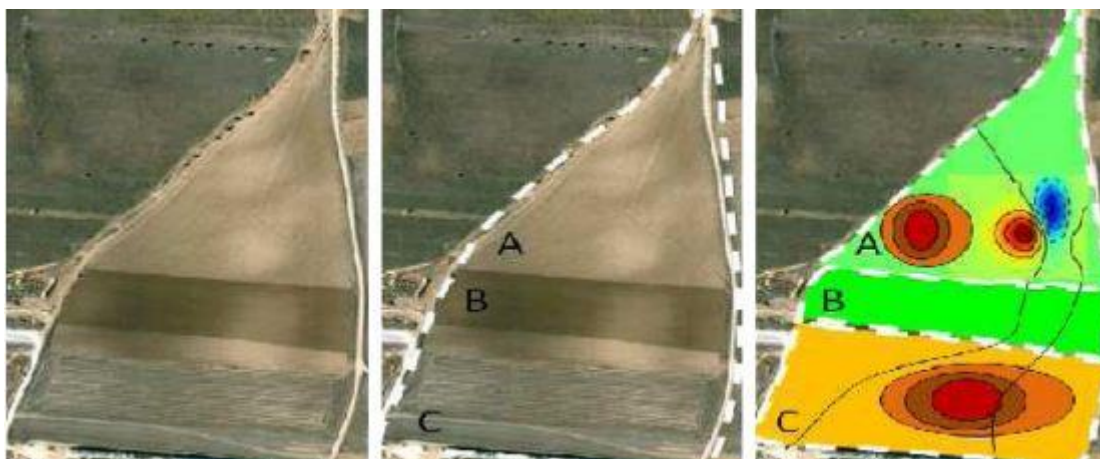


Figura 5 Análise e divisão de zonas de um terreno para o Vineyard Production Monitoring

Como podemos ver na Figura 5, o terreno foi dividido em 3 zonas (A, B, C), com base nos fatores já mencionados. Posteriormente é definido, para cada zona, que diferentes tipos de sensores serão instalados, consoante os parâmetros que se pretendam medir em cada uma delas, e tendo em conta os parâmetros essenciais ao tipo de produção presente no local.

O sistema criado neste caso contém cerca de 30 dispositivos, ou *motes*, que utilizam o protocolo de comunicação ZigBee, e que possuem capacidade de comunicação entre os 75 metros e os 100 metros.

A rede possui nós de obtenção de dados, que os passam ao sistema, mas também nós atuadores, que neste caso são usados para controlar o sistema de irrigação. Esse controlo é exercido com base nos valores lidos pelos nós de obtenção de dados, para os vários parâmetros analisados.

O *deployment* do sistema consiste em dois momentos:

- Instalar os elementos, em que é colocado em cada *mote* o código executável respetivo;
- Colocar fisicamente os *motes* nos respetivos locais, ativá-los e dar assim início à aplicação WSN.

A partir daqui, o sistema funciona, resumidamente, da seguinte forma:

- Os dados vindos da WSN chegam à estação base através do *gateway* e são guardados na base de dados;
- Os dados presentes na base de dados são processados e apresentados aos utilizadores;

- Os utilizadores utilizam a informação obtida para tomarem decisões acerca das ações mais apropriadas no caso de situações ou problemas inesperados.

Para a apresentação dos dados guardados na base de dados aos utilizadores, deve ser usada uma aplicação assistente que extraia e apresente a informação atual. Esta aplicação assistente pode ser uma *app* tradicional, mas é não só recomendado o uso de uma aplicação através da Internet, devido às suas vantagens ao nível da mobilidade e portabilidade entre vários tipos de plataformas, mas também criadas duas ferramentas web, que em conjunto pretendem criar uma *interface* simples mas poderosa para a análise de dados:

- *Sensitive*: Uma ferramenta que permite aos utilizadores analisar os dados obtidos sob a forma de gráficos. Começa por se ligar à base de dados, obter os dados pretendidos sob a forma de valores numéricos, e representa esses valores num gráfico com linhas ou barras. A Figura 6 mostra um exemplo de valores obtidos pela WSN presente na vinha e a sua apresentação sob a forma de gráficos.
- *Sensonic*: Uma ferramenta que pretende facilitar a análise de gráficos por parte dos utilizadores. Esta ferramenta permite que os utilizadores adicionem um ou mais gráficos gerados pela ferramenta *Sensitive* num painel onde se podem juntar vários gráficos relacionados. Podem ser usados vários painéis, cada um contendo vários gráficos. Qualquer gráfico gerado pelo *Sensitive* pode ser adicionado a um painel através do URL associado aos dados representados no gráfico. A Figura 7 mostra o ecrã principal desta aplicação.

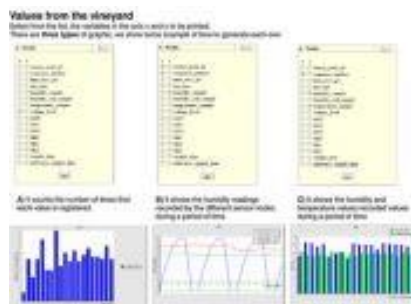


Figura 6 Ecrã da ferramenta *Sensitive*

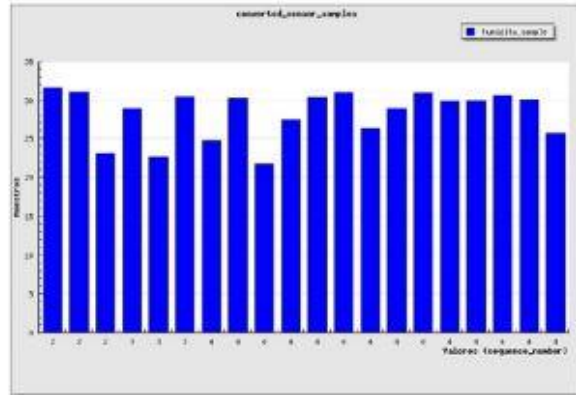
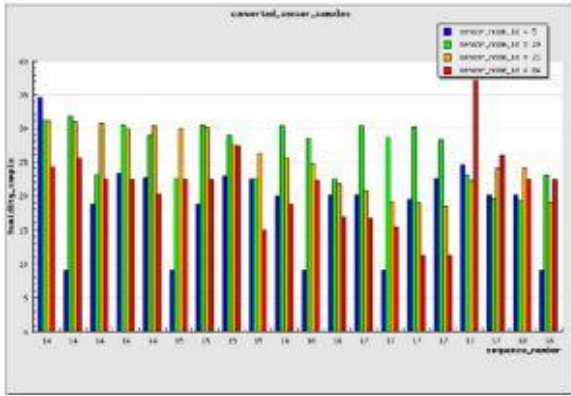
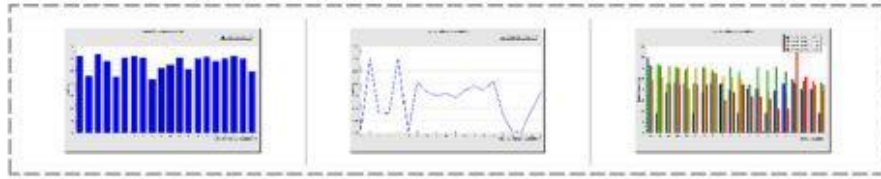


Figura 7 Ecrã da ferramenta Sensonic

2.3. OnFarm

O OnFarm apresenta-se como uma *suite* de serviços de gestão agrícola baseados em *Internet of Things* (IoT).

Pretende oferecer aos seus clientes a capacidade de obter uma visão em tempo real de um grande número de pontos de obtenção de dados variáveis, visão essa necessária à criação de condições de trabalho e de obtenção de resultados óptimos no contexto de produção agrícola.

Oferece uma solução à incompatibilidade dos dados obtidos por sensores diferentes, extraindo informação co-relacionada em tempo real através de WSNs, e responde às crescentes dificuldades que os empreendimentos têm em gerir e analisar todos os dados que atualmente conseguem obter, oferecendo uma plataforma de software baseada em *Software as a Service* (SaaS), agregando dados em informação e apresentando-a de forma compreensível e intuitiva, através de vários tipos de relatórios em tempo real.

Trabalha em ligação com parceiros que fornecem hardware, software, ou serviços na área, incluindo algumas das maiores companhias do setor, criando uma interface do tipo *Plug and Play* entre os equipamentos ou serviços fornecidos pelos vários parceiros e o próprio sistema OnFarm.

Assim, esta solução não contém qualquer instalação de redes sensoriais, mas antes fornece serviços de agregação de dados e de visualização de informação em tempo real sobre redes e equipamentos já existentes.

Os serviços de visualização correm através da *web*, e portanto podem ser utilizados em várias plataformas, como PC, MAC ou *tablets*.

A informação obtida pode ser apresentada de várias formas, como por exemplo:

- *Dashboard*;
- Gráficos;
- Mapas.

O *Grower Dashboard* é a principal ferramenta de visualização de informação, e permite a visualização simultânea de uma grande quantidade de dados, representados de várias formas. Permite a personalização de *dashboards*,

e permite o acesso a qualquer informação obtida por qualquer dispositivo ou em qualquer localização, em tempo real. Na Figura 8 podemos ver um exemplo do *Grower Dashboard*.

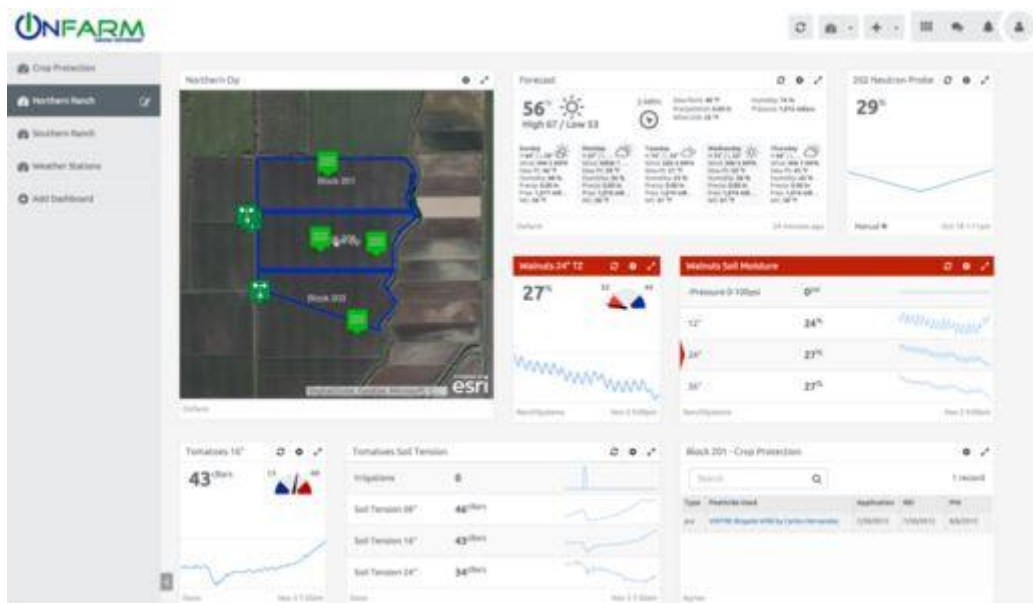


Figura 8 OnFarm Grower Dashboard

A visualização de informação com recurso a gráficos surge incluída no *Grower Dashboard*, mas também pode ser feita separadamente, para um maior nível de detalhe na análise da variação do valor observado. A Figura 9 mostra um gráfico gerado separadamente.



Figura 9 Exemplo de gráfico do sistema OnFarm

Os mapas são uma forma rápida e intuitiva de obter uma visão geral sobre o estado de uma propriedade. Através da sua análise, o utilizador pode ficar rapidamente a saber se algum ponto tem tido leituras de valores que possam estar perto de ultrapassar um *threshold* definido, ou até que o ultrapassem.

Outras funcionalidades importantes do OnFarm são:

- Colaboração;
- Calendarização;
- Alertas.

O sistema OnFarm disponibiliza como tarefa de colaboração a possibilidade de várias pessoas acederem a algumas opções do mesmo, para além de disponibilizar um sistema de troca de mensagens para possibilitar uma fácil comunicação entre esses agentes.

Está incluída uma ferramenta de calendarização, que permite adicionar e consultar eventos que sejam relevantes de forma fácil, para além de ser possível que eventos sejam adicionados de forma automática, mediante configuração prévia.

O sistema permite ainda a definição de alarmes, e no caso de um valor lido pela rede sensorial ultrapassar um valor pré-definido, o que faz disparar um alarme, ser notificado um agente desse facto, de acordo com a atribuição de responsabilidades existente. Os alertas podem ser enviados para o telemóvel e/ou para um endereço de correio eletrónico. A Figura 10 mostra um exemplo de alertas recebidos num telemóvel.

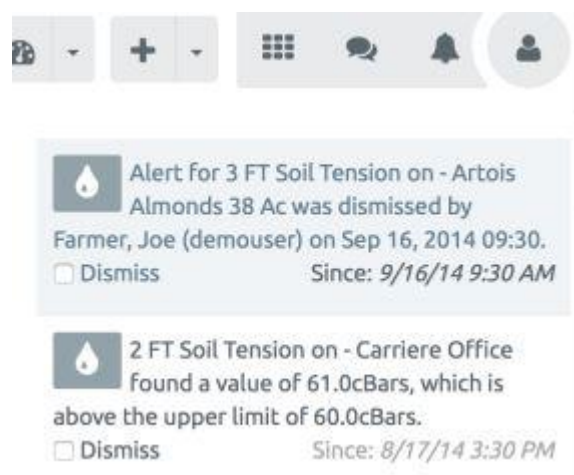


Figura 10 Exemplo de alertas do sistema OnFarm

O produto está disponível em dois pacotes, o pacote *Basic* e o pacote *Pro*. Ambos os pacotes incluem o *Grower Dashboard*, o sistema de alarmes e alertas, a ferramenta de conversação, as capacidades de construção de gráficos e mapas, a ligação a meios públicos de dados meteorológicos, e o serviço de armazenamento e *backup* de dados. Existe ainda a possibilidade de incluir uma ou mais *Virtual Weather Stations*.

O pacote *Basic* está limitado a um utilizador e a um máximo de três estações de recolha de dados, colocados em apenas uma propriedade contígua. O seu custo anual é de USD 100 (cerca de € 88/ano).

O pacote *Pro* não tem qualquer limitação ao nível da integração dos dados, recebendo dados de um número ilimitado de estações, instaladas em uma ou mais áreas sem qualquer limite fixo, e sem qualquer limitação no número de utilizadores do sistema. O seu custo anual é de USD 500 por utilizador (cerca de € 440/utilizador/ano).

2.4. AMI Turf Irrigation System

Desenhado, desenvolvido e comercializado pela empresa *Aqua Management*, este é um sistema de controlo de irrigação, pensado para o contexto da manutenção de relvados, e surge como uma solução orientada para a indústria da irrigação de relvados para o problema da gestão eficiente da água com baixo custo.

É um sistema baseado na *cloud*, que considera parâmetros medidos no terreno, como a evapotranspiração, as condições meteorológicas, e o fluxo de água no terreno para efetuar uma gestão da irrigação do terreno em causa. A partir destes valores, o sistema consegue efetuar uma gestão da irrigação do terreno através de nós atuadores, de forma automática, mas também permite que esse controlo seja feito de forma manual pelo utilizador, através de uma aplicação baseada na *web*, que pode ser acedida através de um qualquer computador, *tablet*, ou *smartphone*.

A utilização do fluxo de água no terreno como um parâmetro para o controlo dinâmico da irrigação desse mesmo terreno permite uma gestão assente na relação entre a quantidade real de água que o terreno recebeu e a quantidade necessária, calculada através dos outros parâmetros, como a evotranspiração e os dados meteorológicos. Esta gestão é mais eficiente que gestões tradicionais, que se baseiam tipicamente apenas no tempo em que o sistema de irrigação esteve ligado, inferindo a partir daí a quantidade de água recebida pelo terreno em causa.

Os valores da evapotranspiração e as condições meteorológicas são obtidos pelo sistema através de sistemas de informação públicos e/ou privados que disponibilizam esse tipo de valores para o local pretendido, ao invés que o fluxo de água no terreno é medido por nós que possuem sensores com essa capacidade. Outro fator utilizado pelo sistema, e que é obtido por sensores no local, é a deteção de fugas de água, que é vista como parte essencial de um sistema dedicado à gestão eficiente de água num sistema de irrigação. A deteção de uma fuga de água no local leva à emissão de um alerta de forma automatizada, podendo ainda, se o sistema estiver configurado para tal, desligar

de forma automática o sistema de irrigação, para que seja procedida à reparação da causa da fuga. Após o problema ter sido solucionado, o sistema pode voltar ao seu normal funcionamento.

O sistema disponibiliza ao utilizador, e mediante as medições efetuadas ao longo do tempo no que toca ao fluxo de água no terreno, relatórios derivados desse consumo de água ao longo do tempo, e que permitam ao utilizador orçamentar o uso de água à escala diária, semanal, mensal, ou até anual, o que pode ser especialmente útil em locais do globo em que o consumo de água tem um preço muito elevado.

A informação de preço do sistema não está atualmente disponível.

2.5. Uso de *ArcGIS* no auxílio à produção de Pêra Rocha

Silva, Câmara, Nunes & Abreu (2011) estudaram a aplicação de SIGs na agrometeorologia, e o seu potencial como auxílio à produção agrícola.

Para esse estudo, um dos casos de estudo foi o da Pêra Rocha do Oeste, uma variedade portuguesa de pêra, que está entre as mais produzidas do continente europeu, e é um produto com Denominação de Origem Produzida. No caso específico deste produto, o parâmetro mensurável mais relevante para a produção com qualidade é o número de horas de frio a que este está exposto, considerando uma hora de frio como horas com temperatura do ar a 2 metros inferior a 7.2 graus centígrados.

Os dados utilizados, relativos a este parâmetro, são os do modelo de previsão numérica do tempo designado por ALADIN, um modelo que funciona à escala europeia. Os dados recolhidos, para a região de produção da Pêra Rocha do Oeste, são da responsabilidade do Instituto de Meteorologia. O modelo ALADIN tem uma resolução de 9 quilómetros, e no estudo feito é utilizado o método de interpolação IDW para a obtenção de valores para um novo conjunto de pontos, que constituem uma nova grelha com resolução de 1 km, por forma a dar mais especificidade local aos resultados desta análise, tendo em conta a escala continental do modelo ALADIN, e a escala regional da produção da Pêra Rocha do Oeste.

É utilizado o sistema ArcGIS, um SIG desenvolvido pela companhia ESRI, para representar esta informação de forma simples e intuitiva, pretendendo-se a partir daqui analisar o acumulado de horas de frio que já ocorreram em cada local da região, e a partir daí ajudar os produtores à escolha do momento mais oportuno à colheita das suas pêras. A observação é feita de forma diária, produzindo a cada dia um novo número acumulado de horas de frio em cada local, e traduzindo essa informação sobre a forma de um mapa da região, como aquele que pode ser visto na Figura 11.

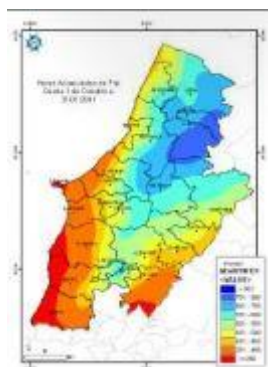


Figura 11 Mapa de horas de frio em ArcGIS

Nesta Figura pode ver-se como a escala de cores representa o número acumulado de horas de frio a que cada local já esteve sujeito durante aquele ciclo de produção de Pêra Rocha do Oeste, sendo o azul a cor indicativa de maior número acumulado de horas de frio, e o vermelho a cor indicativa do menor valor desse parâmetro. Pode assim seguir-se a evolução da situação em toda a região de forma diária, e analisar a partir daí o período de colheita mais apropriado para cada local e produtor, maximizando a qualidade do produto em toda a região.

Esta informação pode ainda representada sob a forma de gráficos, como aquele que podemos ver na Figura 12, em que se vê o acumulado de horas para vários locais, e está colocada uma linha verde que marca o número necessário de horas de frio a partir do qual se pode programar a colheita para uma qualidade máxima do produto. O período entre o alcance deste número de horas de frio e a colheita deve ser entre 135 e 140 dias.



Figura 12 Gráfico do acumulado de horas de frio para vários locais em ArcGIS

2.6. Análise Comparativa

De seguida procura-se comparar algumas das características das soluções e produtos apresentadas anteriormente.

Característica		<i>Fire Systems</i>	<i>Vineyard Production Monitoring</i>	OnFarm	AMI <i>Turf Irr. System</i>	<i>ArcGIS Pêra Rocha</i>
Contexto		Prevenção de Fogos	Produção Agrícola	Produção Agrícola	Gestão de Água	Produção Agrícola
WSN	Rede Sensorial Própria	Sim	Sim	Não	Sim	Não
	Parâmetros Medidos	Temperatura	Vários parâmetros relevantes	Potencialmente todos	Fluxo de água e fugas	Dados públicos
	Nós atuadores	Não	Sim	Potencialmente sim	Sim	Não

Tabela 1 Tabela comparativa das soluções 1

A Tabela 1 contém as características principais das várias soluções apresentadas, no que toca ao contexto para o qual cada uma está direcionada e ao tipo e constituição das redes de sensores que os suportam.

Podemos verificar que a solução *Fire Systems* está direcionada para o contexto da prevenção de fogos, quer em ambiente *indoor*, quer em ambiente florestal, ou *outdoor*. O AMI *Turf Irrigation System* está direcionado para a gestão eficiente de recursos hídricos na irrigação de relvados. Quer a solução *Vineyard Production Monitoring*, quer o sistema OnFarm, quer ainda o uso de *ArcGIS* no auxílio à produção de Pêra Rocha estão direcionados para o contexto da produção agrícola, pretendendo assim contribuir para a melhoria da qualidade e quantidade dos produtos dos seus clientes. O *Vineyard Production Monitoring* e o

OnFarm estão direcionados para a medição de dados em tempo real e à sua utilização no dia-a-dia da produção agrícola, enquanto o uso de *ArcGIS* no auxílio à produção de Pêra Rocha está mais direcionado para medição diária de um parâmetro e à sua utilização na calendarização da fase de colheita, de forma anual.

No que toca às WSN que fornecem dados ambientais aos sistemas, verificamos uma diferença fundamental entre as soluções *Fire Systems*, *Vineyard Production Monitoring* e *AMI Turf Irrigation System*, e o sistema *OnFarm*, e ainda a solução de uso de *ArcGIS* no auxílio à produção de Pêra Rocha, que consiste no facto de as três primeiras desenharem e instalarem uma rede de sensores própria para a obtenção de dados, o sistema *OnFarm* funcionar em ligação com equipamentos e serviços já existentes, e a última utilizar apenas dados obtidos por um instituto público, o que perfaz uma diferença fundamental entre estas várias soluções.

A solução *Fire Systems*, tendo em conta a sua especificidade, pretende apenas medir a temperatura em vários locais, ao contrário da *Vineyard Production Monitoring*, que cria redes sensoriais com capacidades para medir vários parâmetros relevantes no contexto da produção de vinhas. Já o *OnFarm* tem capacidade para funcionar com dados obtidos por qualquer tipo de sensores, tendo em conta que não define a constituição dos nós da rede de sensores e tem capacidade para receber dados de vários tipos de dispositivos. O *AMI Turf Irrigation System* instala capacidades de medição do fluxo de água num terreno para além de sensores de identificação de fugas de água. Por último, o uso de *ArcGIS* no auxílio à produção de Pêra Rocha utiliza apenas dados públicos.

A *Vineyard Production Monitoring*, o *AMI Turf Irrigation System* e o *OnFarm* utilizam ainda nós atuadores, sendo que no caso do *OnFarm* apenas se estes estiverem presentes na rede que sustenta o sistema.

Na Tabela 2 volta-se a analisar estas soluções, mas desta vez com o foco nas possibilidades dadas aos utilizadores para a consulta e visualização da informação e outras funcionalidades importantes que possuam.

Característica		<i>Fire Systems</i>	<i>Vineyard Production Monitoring</i>	OnFarm	<i>AMI Turf Irr. System</i>	<i>ArcGIS Pêra Rocha</i>
Visualização da Informação	Plataforma	Web; Mobile	Web	Web	Web	Web; Desktop
	Plantas	Sim	Não	Não	Não	Não
	Mapas	Sim	Não	Sim	Sim	Sim
	Gráficos	Não	Sim	Sim	Não	Sim
	<i>Dashboard</i>	Não	Não	Sim	Não	Não
Outras Funcionalidades	Colaboração	Não	Não	Sim	Não	Não
	Calendarização	Não	Não	Sim	Não	Não
	Alarmes	Não	Sim	Sim	Sim	Não
	Alertas	Não	Não	Sim	Sim	Não

Tabela 2 Tabela comparativa das soluções 2

Em termos das plataformas nas quais os utilizadores das várias soluções podem utilizar as aplicações respetivas para a visualização da informação, verifica-se que as cinco soluções e sistemas apostam na plataforma web, procurando aproveitar a sua portabilidade. Além disso, a solução *Fire Systems* contém também aplicações *mobile*, e o *ArcGIS* pode ser usado como uma ferramenta *desktop*.

No que toca às representações da informação, a *Fire Systems* usa plantas para representar a informação de sistemas *indoor* e mapas para representar a informação de sistemas *outdoor*; a solução *Vineyard Production Monitoring* faz

uso intensivo de gráficos, enquanto o OnFarm apresenta uma maior variedade, utilizando mapas, gráficos e um *Dashboard* configurável que conjuga os outros tipos de representação. O AMI *Turf Irrigation System* utiliza mapas na *interface* com o utilizador, e o uso de *ArcGIS* no auxílio à produção de Pêra Rocha produz mapas e gráficos para a representação da informação.

Em termos de alarmes, as soluções *Vineyard Production Monitoring* e AMI *Turf Irrigation System* têm capacidades de identificação de valores fora de um determinado *threshold* e a atuação através dos nós atuadores que possuem. O sistema OnFarm, além das capacidades de atuação via nós atuadores, caso existam na rede em causa, tem ainda a funcionalidade de enviar alertas a um determinado utilizador que esteja pré-configurado, podendo ser enviado para um telemóvel ou para um e-mail.

O OnFarm contém ainda outras funcionalidades importantes, como as capacidades de colaboração e comunicação, a ferramenta de calendarização, e o sistema de alertas após a identificação de um alarme.

Todas estas capacidades de representação de informação de múltiplas maneiras e funcionalidades extra presentes no sistema OnFarm quando comparado com as outras soluções analisadas mostram a diferença fundamental que este sistema apresenta em relação aos outros: é um sistema de análise e representação de informação, sem especificação da recolha de dados, contrariamente ao que pretendem ser as soluções *Fire Systems*, *Vineyard Production Monitoring* e AMI *Turf Irrigation System*.

Estas soluções são bastante específicas no seu contexto de aplicação. Assim, entendo existir espaço para o desenvolvimento de uma solução mais abrangente, que possa ser utilizada numa maior variedade de contextos, e que se adapte aos mesmos.

3. CONCEÇÃO E DESENVOLVIMENTO

3.1. Equipamento

Tal como já foi referido nos capítulos anteriores, o objetivo desta dissertação é o de projetar e implementar uma rede de sensores do tipo WSN para a obtenção de dados ambientais e criar um sistema que permita a recolha e armazenamento dos mesmos através de comunicação em tempo real, projetando a rede e o sistema correspondente para o seu correto funcionamento para a maior variedade de valores e tipos de dados ambientais possíveis.

A rede será constituída por vários pontos, ou nós, que representam locais de interesse, e nos quais interessa medir determinados parâmetros ambientais, definidos quer pelo contexto global da rede, quer pela especificidade de cada localização. Em cada um destes pontos deverão existir equipamentos preparados para proceder à recolha dos dados pretendidos, através de sensores, e ao seu envio para o sistema, através da comunicação que se estabelece com este.

Nesta solução serão usados dispositivos *Waspote*, produzidos pela companhia *Libelium*. Estes dispositivos, ou *motes*, permitem a integração de vários tipos de sensores, bem como o estabelecimento de uma rede de comunicação entre os vários dispositivos num local, tipicamente com recurso a módulos *XBee*, módulos esses baseados no *standard* IEEE 802.15.4. Na Figura 13 podemos ver um destes dispositivos.



Figura 13 Dispositivo Waspote

Estes dispositivos funcionam de forma cíclica, permitindo leitura e envio de dados, num funcionamento mais básico, mas suportam interrupções programáveis, quer interrupções geradas de forma assíncrona, quer interrupções geradas de forma síncrona.

Interrupções geradas de forma assíncrona incluem interrupções geradas quando um valor lido por um sensor está fora de um intervalo esperado, interrupções geradas pelo acelerómetro interno do dispositivo, tipicamente para identificar situações de queda livre ou de impacto, e interrupções geradas pelo módulo *XBee*.

As interrupções geradas de forma síncrona podem ser geradas pelo *Watchdog*, com intervalos de tempo de 32ms a 8s, ou pelo RTC interno, com um intervalo de tempo desde 1 segundo até vários dias. Interrupções deste tipo permitem gerir o consumo de energia pelo *mote* através da utilização de estados de baixo consumo, pelo que estão presentes, tipicamente, em qualquer programa.

Os dispositivos *Waspote* podem ser alimentados por bateria (quer recarregável, com menor capacidade, quer não-recarregável, com várias e maiores capacidades), por painel solar (rígido ou flexível), ou por USB (podem ainda ser usados adaptadores 220V-USB, ou isqueiro automóvel - USB).

Para a *interface* entre os dispositivos *Waspote* e um PC, é utilizado um *Waspote Gateway*, um dispositivo de capacidades limitadas, que funciona apenas como uma *bridge* entre uma rede do *standard* 802.15.4 ou *ZigBee*, e uma porta USB de um PC. Este dispositivo permite a monitorização básica de uma rede de *Waspotes*.

Para a programação dos *motes*, um IDE específico *open-source*, o *Waspote-IDE*, está disponível. Os *motes* são programados na linguagem C++.

De notar ainda a capacidade do *Waspote* em utilizar um cartão SD, o que permite, por exemplo, a salvaguarda de dados sensoriais que por qualquer razão não tenham sido corretamente enviados pela rede.

Cada dispositivo *Waspote* deverá utilizar um componente designado por *sensor board*, que faz a ligação entre os sensores e as entradas analógicas e digitais da placa principal. Este facto, exemplo da modularidade do dispositivo *Waspote*, torna-o passível de ser utilizado num grande número de contextos e

situações, bastando para isso ser utilizada uma *sensor board* que responda às necessidades desse mesmo contexto.

A *sensor board* é também responsável por fazer quaisquer adaptações que sejam necessárias, ao nível eletrónico, aos valores que recebe de cada sensor, para garantir a correção e compatibilidade dos mesmos. Existem várias *sensor boards*, cada uma pensada para um tipo de aplicação, cada uma das quais com um número e tipo pré-definido de sensores a utilizar.

Assim sendo, os *Waspnotes* funcionam com vários tipos de sensores em simultâneo, e com sensores muito variados, que medem os parâmetros que se julgam ser mais relevantes no contexto indicado para cada *sensor board*.

Para responder às necessidades do projeto a implementar, utilizámos um *kit* de avaliação de *Waspnotes*. Este *kit* contém os itens mais encomendados pelos utilizadores destes equipamentos. Este *kit* contém quatro *Waspnotes* e um *Waspnote Gateway*, para além de cinco *sensor boards* diferentes, a saber: *Gases Board*, *Events Board*, *Smart Cities Board*, *Smart Metering Board* e *Agriculture PRO Board*.

Este *kit* contém vários sensores de vários tipos, dos quais destacamos:

- 4 sensores de temperatura do ar MCP9700A;
- 1 sensor de humidade do ar 808H5V5;
- 1 sensor de temperatura do solo PT-1000;
- 1 sensor de humidade do solo *Watermark*;
- 1 *Weather Station* WS-3000.

3.2. Requisitos

Antes de começar o desenvolvimento da solução pretendida, importa perceber e definir as necessidades e requisitos do sistema, sabendo que se pretende uma comunicação em tempo real entre a rede de sensores e o sistema, e se pretende um sistema dinâmico e capaz de se adaptar a diferentes tipos de dados ambientais.

Em primeiro lugar, os dispositivos que vão constituir a nossa WSN, os *Waspote*, constroem as suas *frames* de comunicação, usadas entre outros fins para o envio dos dados obtidos pelos sensores a eles ligados, com um formato bem definido pelo seu fabricante, e que suporta dois tipos de *frames*:

- *Frames* ASCII, *frames* a utilizar quando se pretende que a sua estrutura e conteúdo seja facilmente perceptível e compreensível;
- *Frames* Binárias, desenhadas sobretudo para criar *frames* mais comprimidas, diminuindo o tamanho da *frame*. A desvantagem deste tipo de *frames* é a sua fraca legibilidade.

A constituição de ambos os tipos de *frame* tem duas partes: um cabeçalho, em que são incluídas algumas informações indicativas do dispositivo criador da *frame*, como o seu identificador de série e a sua *tag* de identificação, e sobre a própria *frame*, como o número da *frame* e um indicador do seu tamanho; e um *payload*, que é constituído por um determinado número de campos que permitem a identificação de um sensor, cada um seguido pelo valor medido por esse sensor.

A utilização destes dispositivos leva à receção de *frames* com este formato por parte do sistema, e, apesar de o tipo de *frames* utilizadas por cada dispositivo ser programável, é um requisito do sistema que este seja capaz de trabalhar com qualquer um deles.

Uma segunda questão prende-se com a recolha dos dados obtidos pela WSN pelo sistema, isto é, na definição de um *gateway* da rede para o sistema e da comunicação entre as partes. A forma recomendada pela *Libelium*, o fabricante dos dispositivos *Waspote*, é através do seu *router wireless Meshlium*, um equipamento com capacidades para receber *frames* através dos vários

métodos de comunicação disponíveis para utilização nos *motes*, bem como para as reenviar via *Ethernet/WiFi* ou via 3G/GPRS, para além de as poder armazenar numa base de dados local ou externa do tipo *MySQL*.

Este equipamento tem um custo elevado, pelo que se pretende desenvolver *middleware* próprio, capaz de recolher, tratar e armazenar a informação recolhida pela rede de sensores.

A opção tomada é usar o *Waspote Gateway*, em ligação com um computador, para a recolha dos dados, tratando o computador eventualmente de reencaminhar os dados no sistema. Esta opção permite logo à partida monitorizar o tráfego na rede de sensores em tempo real, o que é uma vantagem não só durante o processo de desenvolvimento da solução, mas também em fase de funcionamento com a possibilidade de serem adicionadas mais capacidades de monitorização e visualização do tráfego da rede.

Tendo em conta que o *Waspote Gateway* funciona apenas como uma *bridge* entre a rede de sensores e a porta USB do computador, inviabilizando qualquer tratamento dos dados nessa fase, e não disponibilizando qualquer *interface* para a recolha dos mesmos, será necessário criar um monitor de porta série para recolher os dados em bruto, e tratá-los a partir daí. Esse será também um requisito do sistema.

É também necessário o tratamento das *frames* criadas pela rede sensorial, através de um processo de *parsing*, que identifique a origem e a informação relevante contida na mesma, deconstruindo-a de forma metódica. O sistema deve ser capaz de o fazer para qualquer um dos tipos de *frames*, e sem conhecimento prévio desse mesmo tipo.

Outra das condicionantes da projeção do sistema passa pela possível não-suficiência de um dispositivo *Waspote* para a obtenção de todos os valores pretendidos para um determinado local. Apesar de ser possível ignorar esta condicionante, e simplesmente representar dois nós com localização muito semelhante ou idêntica, entendemos que a criação de uma camada de abstração que permita a utilização dos dados de vários *motes* como sendo oriundas do mesmo ponto deve ser tida como um requisito do sistema, já que isso permitirá

um comportamento mais dinâmico do sistema face a alterações na estrutura da WSN.

Quer o *parsing* das *frames* recebidas pelo sistema, quer a sua conversão tendo a conta a camada de abstração entre *motes* e nós anteriormente enunciada, são fases do tratamento de dados que se pretende que sejam efetuadas antes do armazenamento em base de dados. É estipulado que esse tratamento dos dados seja realizado por um componente diferente dos utilizados para a recolha dos dados, mas em comunicação com estes, de forma a permitir a substituição da forma de recolha de dados posteriormente, se necessário, ou o funcionamento do sistema para várias formas de recolha de dados em simultâneo.

A visualização da informação não é o foco principal. No entanto, serão estudadas e desenvolvidas algumas formas de representação e visualização da mesma, com o principal objetivo de estudar formas de representação de dados automática para valores e tipos de valores dinâmicos. O desenvolvimento destas formas de visualização deverá ter por base a web.

3.3. Arquitetura

Partindo dos objetivos desta dissertação e dos requisitos definidos anteriormente, incluindo a comunicação em tempo real entre a WNS e o sistema, o método de recolha de dados e o de ligação entre as duas componentes, a necessidade de *parsing* das *frames* e processamento das mesmas em fase anterior ao armazenamento, e ao facto de se pretender esse *parsing* e esse processamento num componente externo ao método de ligação rede / sistema, foi definida a arquitetura de sistema representada nas Figuras 14 e 15.

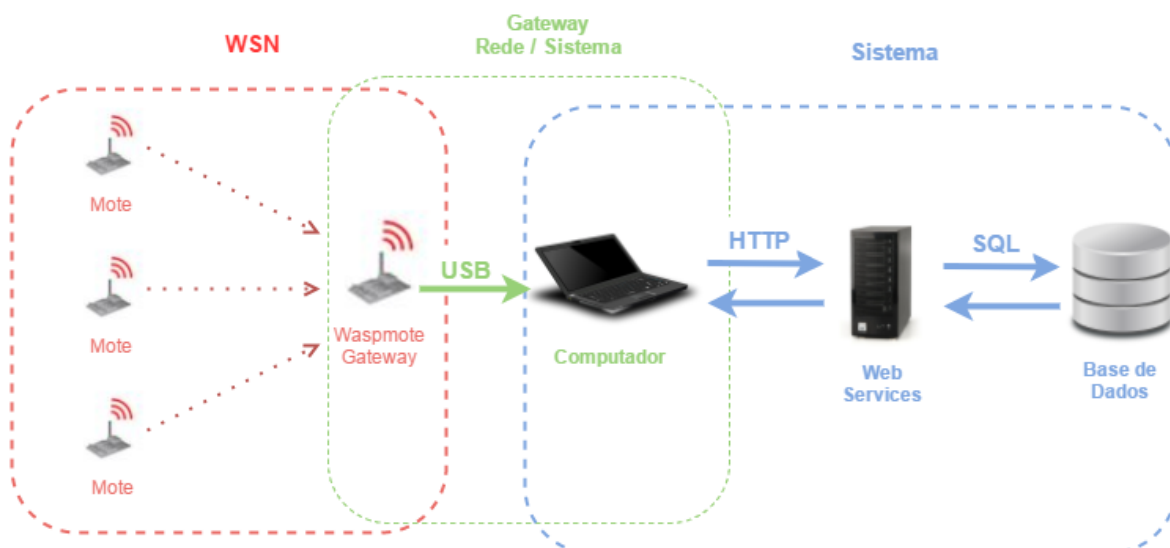


Figura 14 Arquitetura do sistema para a recolha de dados

Na Figura 14 podemos ver a arquitetura definida do ponto de vista da recolha, processamento e armazenamento de dados.

Os vários *motes* constituintes da rede sensorial vão recolhendo dados e enviando-os para o equipamento *Waspnote Gateway*, através de comunicação sem fios baseada no *standard* IEEE 802.15.4.

O *Waspnote Gateway* recebe os dados dessa comunicação sem fios e atua como um *bridge*, passando os dados para a porta USB a que se encontra ligado.

Um programa que funciona como um monitor de porta série a correr num Computador, lê os dados provenientes da rede de sensores e entregues pelo

Waspote Gateway numa das portas USB, identificando as *frames* e enviando-as em bruto na invocação de um *Web Service*.

Este *Web Service*, ao ser invocado, vai realizar o *parsing* à *frame* recebida, e procede à conversão da informação contida na mesma de informação por *mote* para informação por nó. Neste processo é utilizada a ligação à Base de Dados, que contém a configuração da rede necessária à realização desta conversão, através de SQL. Nesta fase é também realizado qualquer processamento adicional que seja necessário aos dados recolhidos pelos sensores, como conversão entre valores numa unidade para outra unidade, entre outros.

Após este processo, o *Web Service* envia a informação para ser armazenada na Base de Dados através de SQL, tipicamente em uma entrada numa tabela para cada leitura de um sensor presente na *frame* analisada.

A Base de Dados armazena a informação enviada pelo *Web Service*, e dá início ao processo de respostas: da Base de Dados para o *Web Service*, e do *Web Service* para o programa monitor que o invocou.

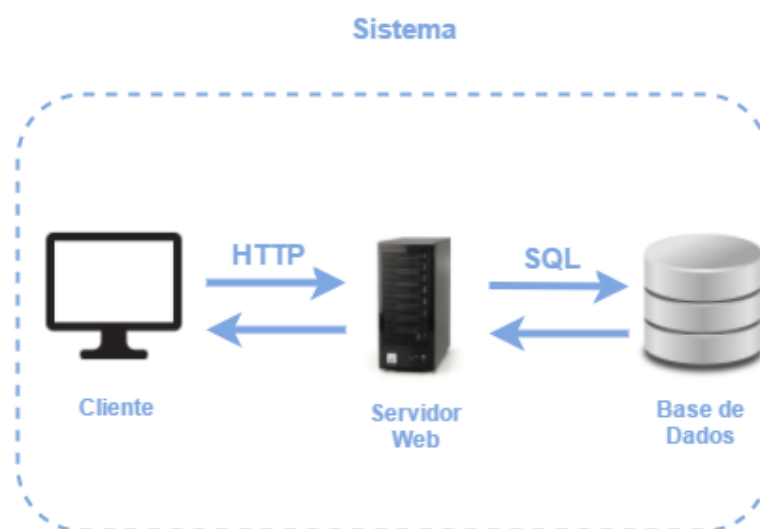


Figura 15 Arquitetura do sistema para a visualização de informação

Na Figura 15 podemos ver a arquitetura definida do ponto de vista da consulta da informação, e conseqüentemente do estudo das técnicas de visualização.

O Cliente, através da aplicação *web* desenvolvida, requisita um determinado conjunto de informação ao Servidor *Web*, através do protocolo HTTP. O Servidor *Web* requisita essa mesma informação à Base de Dados, que a devolve.

Ao receber a informação que havia sido requisitada pelo Cliente, o Servidor *Web* processa-a de forma a representá-la de forma intuitiva e em tempo real ao utilizador, e tendo criado um conjunto de componentes de visualização de forma dinâmica, cria a resposta ao pedido do Cliente sob a forma de uma página *web*.

Por forma a tornar mais compreensível a sequência de ações que são executadas pelos vários componentes durante o processo de recolha, processamento, e armazenamento de dados em tempo real, e da comunicação entre eles, apresenta-se na Figura 16 um diagrama de sequência com essas informações.

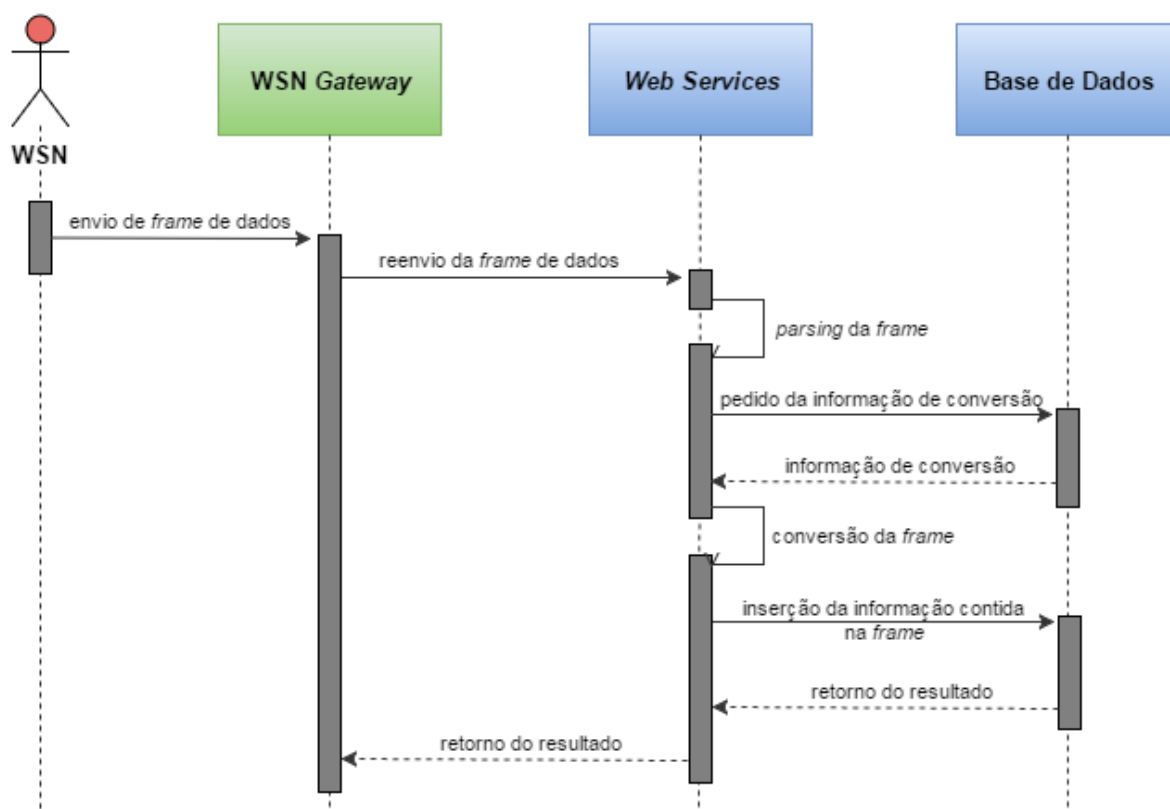


Figura 16 Diagrama de sequência da recolha de dados

3.4. Implementação do Protótipo

Neste subcapítulo iremos discutir com algum pormenor a implementação dos vários componentes especificados e definidos nos requisitos enumerados anteriormente e presentes na arquitetura já descrita.

3.4.1. Modelo de Base de Dados

Para o armazenamento da informação obtida pelas redes sensoriais, é necessário uma base de dados, e que esta tenha uma estrutura e um modelo que a torne capaz de cumprir as suas importantes funções no sistema. Tendo em conta a forma como toda a informação que vai estar contida nesta base de dados se relaciona, definiu-se o seguinte modelo de base de dados.

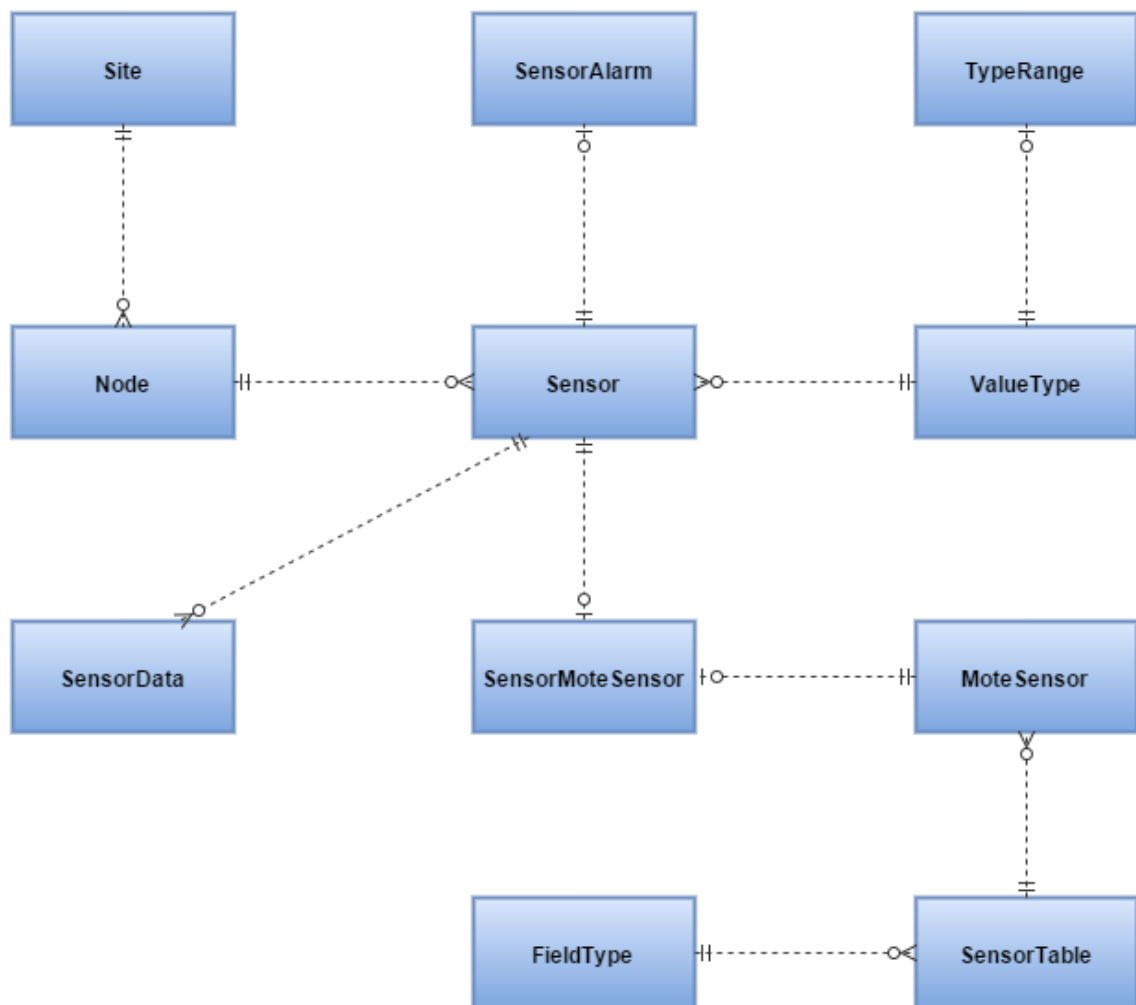


Figura 17 Modelo da base de dados

Neste contexto, a tabela *Site* contém os dados relativos a essa mesma entidade, que consiste na agregação de um conjunto de locais onde estarão nós da rede de obtenção de dados, relacionados entre si.

A tabela *Node* contém a informação de cada um desses nós da rede, pelo que se relaciona com a tabela *Site*. A tabela *Node* inclui informações como a localização geográfica de cada nó, com campos que armazenam a sua latitude e longitude.

A tabela *Sensor*, por sua vez, representa um sensor da rede WSN, com informação importante como o nó a que pertence, o que dá a localização do mesmo, o parâmetro ambiental lido por aquele sensor, e a profundidade a que esse sensor se encontra, na instalação física da rede.

A tabela *SensorAlarm* possibilita a definição de alarmes a serem acionados quando o valor de um sensor ultrapassa um determinado *threshold*. Os valores mínimos e máximos considerados normais para cada sensor podem aqui ser definidos.

A tabela *Sensor* relaciona-se ainda com a tabela *ValueType*, que contém os vários parâmetros ambientais que é possível medir. A existência desta tabela permite relacionar os vários sensores que meçam o mesmo parâmetro ambiental, ao mesmo tempo que mantém a possibilidade de serem acrescentados novos parâmetros a qualquer momento, se tal fôr desejado. Ainda neste contexto, é possível definir um *range* de valores para um parâmetro, se tal fizer sentido. Essa definição é feita através de uma entrada na tabela *TypeRange*, sendo possível definir limites mínimos e máximos. Como um exemplo da utilidade desta tabela, podemos referir o caso do parâmetro do nível pH, que varia numa escala entre 0 e 14.

Para representar e armazenar toda a informação proveniente da rede, nomeadamente os valores obtidos pelos sensores da rede, existe a tabela *SensorData*. Esta tabela contém a referência ao sensor que leu determinado valor, o valor lido e data e hora a que esse valor foi lido. Através da referência e da relação entre a tabela *SensorData* e a tabela *Sensor*, podem obter-se outras informações relevantes como a localização onde o valor foi lido, ou até a profundidade a que se encontra o sensor que o obteve.

A tabela *MoteSensor* representa a informação da estrutura da rede, ao nível dos dispositivos *Waspmote* e dos sensores ligados a cada um deles. A correlação entre o sensor da estrutura física da rede, cuja entrada está na tabela *MoteSensor*, e o sensor da estrutura virtual da rede, cuja representação se encontra na tabela *Sensor*, é feita através de uma entrada na tabela *SensorMoteSensor*.

Por último, as tabelas *SensorTable* e *FieldType* contêm informação necessária ao *parsing* das *frames* recebidas pelo sistema. A tabela *SensorTable* contém a informação que relaciona o ID de cada sensor físico atribuído pelo fabricante dos dispositivos com as características do valor correspondente na *frame*, como o tipo de dados e o tamanho dos mesmos. A tabela *FieldType* permite o relacionamento entre as entradas da tabela *SensorTable* com o mesmo tipo de dados, o que é útil no momento do *parsing*, para além de permitir a adição de novos tipos de dados, caso surjam entretanto sensores que os utilizem.

Uma descrição mais detalhada deste modelo de base de dados e das suas tabelas encontra-se no Anexo A.

Esta base de dados foi criada em *SQL Server*.

3.4.2. Web Services

Na comunicação entre a rede e a base de dados, está colocado um servidor de *Web Services*, cuja finalidade principal é efetuar o processamento necessário sobre as *frames* de dados enviadas pelos *motes* da rede através do *gateway*, mas também tem a finalidade de criar uma camada de abstração entre a WSN e a base de dados, recebendo dados em bruto da rede e convertendo-os em informação útil e armazenável na base de dados cujo modelo foi definido no ponto anterior.

Este *Web Service*, programado na linguagem Java, está disponibilizado de forma a ser invocado através da especificação de protocolo SOAP, a funcionar sobre HTTP. A intenção da invocação do *Web Service* com o envio da *frame* em bruto, associado ao método de envio, leva-nos à utilização de uma codificação em *base64*, de forma a lidar com questões de formato e conteúdo da *frame*.

Na Figura 18, recorda-se a secção relevante do diagrama de sequência apresentado na Figura 16.

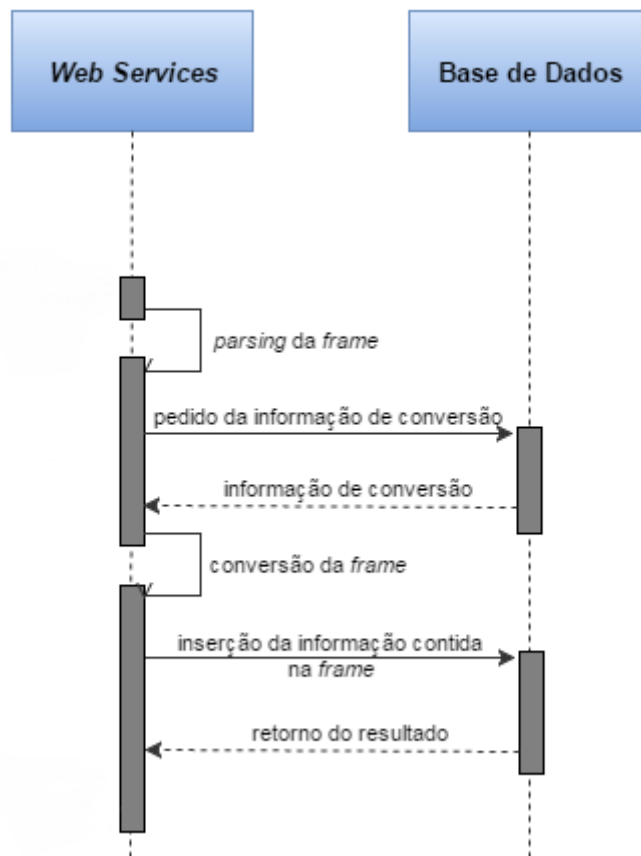


Figura 18 Parte do diagrama de sequência da recolha de dados para o servidor de Web Services

Como podemos verificar, as tarefas desempenhadas pelo *Web Service* invocado, após a recepção dos dados em brutos e sua descodificação para o formato original, são:

- Realizar o *parsing* da *frame*;
- Realizar a conversão dos dados contido na *frame*, em informação útil baseada em nós e sensores definidos na base de dados;
- Armazenar os dados na base de dados.

3.4.3. Gateway da Rede – Monitor Série

Para tornar possível a comunicação entre a rede de dispositivos *Wasp mote* e o resto do sistema, nomeadamente o servidor de *Web Services*, foi definido o *gateway* da WSN como o conjunto de um dispositivo *Wasp mote Gateway*, um equipamento que faz a ligação entre a rede do protocolo IEEE 802.15.4 e uma porta USB, e um computador.

Para esse fim, foi desenvolvido um protótipo de uma aplicação para o computador, em C#, que lhe permite ler os dados em bruto da porta USB a que está ligado o *Wasp mote Gateway*, identificar e delimitar as *frames* que estão a ser recebidas, e invocar o *Web Service* pretendido.

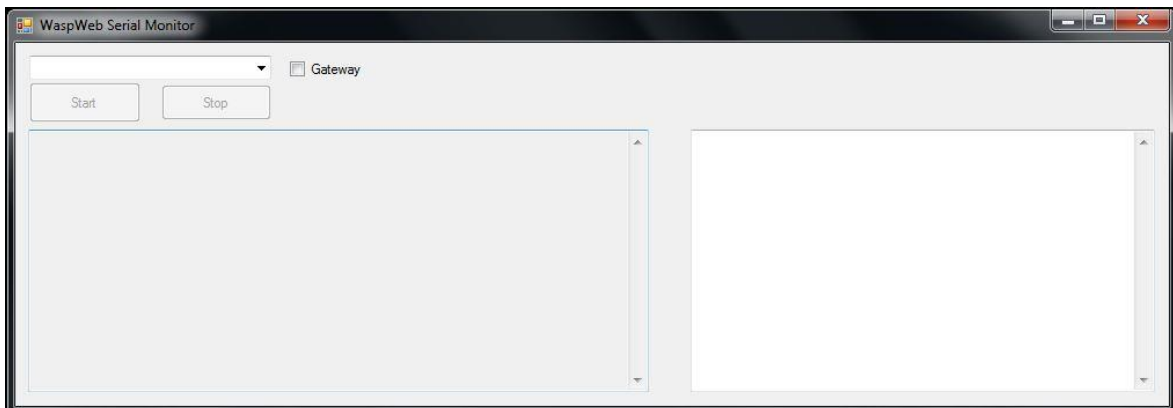


Figura 19 Interface da aplicação monitor de porta série

Na Figura 19 podemos ver a *interface* desta aplicação. Nela, podemos controlar o processo de receção de *frames* da rede, e envio desta para o servidor de *Web Services*, e detetar possíveis anomalias neste processo que estejam a acontecer.

Toda a informação recebida na porta USB que estiver a ser monitorizada é escrita na caixa da esquerda em bruto. Ao mesmo tempo, vai sendo analisada pela aplicação, que identifica e delimita as *frames* recebidas, e as envia para o servidor, tal como já foi referido.

É possível monitorizar qualquer uma das portas USB que tenham *drivers* apropriados ao seu funcionamento como uma porta de série em conjunto com os dispositivos nela ligados. Para a escolha da porta a monitorizar, basta seleccionar a porta pretendida na lista apresentada, tal como se mostra na Figura 20.

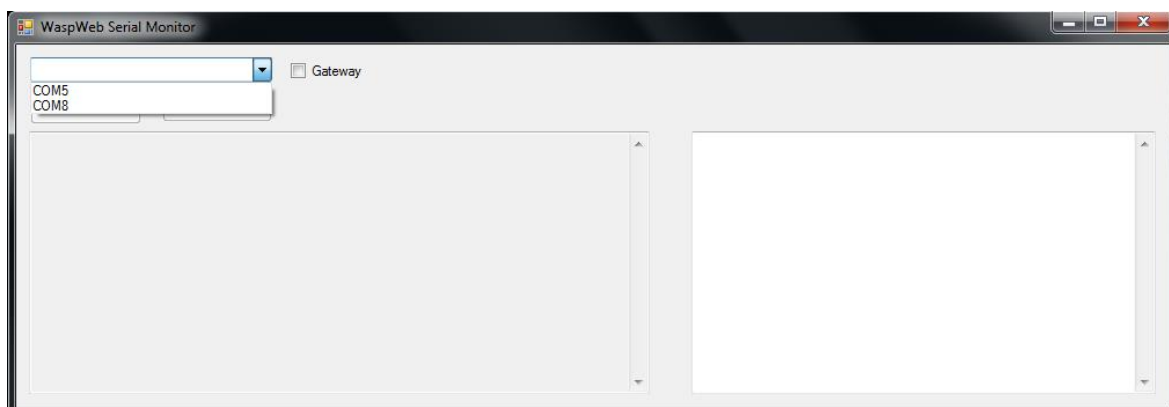


Figura 20 Interface da seleção da porta série

A aplicação está preparada para funcionar quer monitorizando uma porta à qual está ligado diretamente um dispositivo *Waspnote* diretamente, ou monitorizando uma porta à qual está ligado o *Waspnote Gateway*, que é o caso utilizado em funcionamento normal da solução. Devido aos diferentes parâmetros de comunicação em série dos dispositivos *Waspnote* e do *Waspnote Gateway*, nomeadamente ao nível do *baud rate*, deve ser explicitado previamente que tipo de dispositivo está ligado na porta que se pretende monitorizar, através da seleção ou não da *check box* junto da *label* "Gateway".

3.4.4. Motes

Na base de todo o sistema, está a rede de dispositivos, ou *motes*, encarregues de medir os parâmetros ambientais desejados. Durante o desenvolvimento da solução, foram utilizadas redes de até 3 *motes* a trabalharem em simultâneo, e, tendo em conta a quantidade e o tipo dos sensores disponíveis para se ligarem a estes dispositivos e recolherem dados, a configuração típica dos dispositivos, em termos de sensores ligados, era um sensor de temperatura e um de humidade, ou apenas um de temperatura, variando conforme o dispositivo em causa.

Tendo a programação de cada *mote* a sua especificidade, quer pelo uso de diferentes sensores, quer pelo uso de uma *sensor board* diferente, os programas executáveis para estes dispositivos têm todos uma estrutura semelhante, e o código dos programas utilizados na rede desta solução eram por isso bastante parecidos. Esses programas são escritos na linguagem C++.

O código apresentado na Figura 21 é um exemplo representativo desses programas:

```
#include <WaspXBee802.h>
#include <WaspFrame.h>
#include <WaspSensorAgr_v20.h>

// Ponteiro para uma estrutura de pacote XBee
packetXBee* packet;
// MAC Address Destino
char* MAC_ADDRESS="000000000000FFFF";
// Identificador do nó
char* NODE_ID="NAGR01";
// Tempo de sleep DD:hh:mm:ss
char* sleepTime = "00:00:05:00";
// Declaração das variáveis para a medição dos valores dos sensores
float temperatureValues[10], humidityValues[10];

void setup()
{
  // 0. Iniciar a porta USB para debugging
  USB.ON();

  // 1. Composição da mensagem inicial
  ...
  // 2. Envio da mensagem inicial
  ...
}
```

```

void loop()
{
  // 3. Medir os valores pretendidos
  USB.println(F("Measuring sensors..."));
  SensorAgrv20.ON();
  RTC.ON();
  delay(100);

  // Ligar os sensores e esperar pela sua estabilização
  SensorAgrv20.setSensorMode(SENS_ON, SENS_AGR_HUMIDITY);
  SensorAgrv20.setSensorMode(SENS_ON, SENS_AGR_TEMPERATURE);
  delay(15000);
  // Obter 10 medições
  for (int i=0;i<10;i++){
    temperatureValues[i]=SensorAgrv20.readValue(SENS_AGR_TEMPERATURE);
    humidityValues[i] = SensorAgrv20.readValue(SENS_AGR_HUMIDITY);
  }
  // Desligar os sensores
  SensorAgrv20.setSensorMode(SENS_OFF, SENS_AGR_TEMPERATURE);
  SensorAgrv20.setSensorMode(SENS_OFF, SENS_AGR_HUMIDITY);
  // Calcular a média
  for (int i=1; i<10; i++){
    temperatureValues[0] = temperatureValues[0]+temperatureValues[i];
    humidityValues[0]+=humidityValues[i];
  }
  temperatureValues[0]/=10;
  humidityValues[0]/=10;

  SensorAgrv20.OFF();
  RTC.getTime();

  // 4. Composição da mensagem
  // 4.1 Criar uma nova frame
  frame.createFrame(ASCII);
  // 4.2 Adicionar os campos da frame
  frame.addSensor(SENSOR_TCA, temperatureValues[0]);
  frame.addSensor(SENSOR_HUMA, humidityValues[0]);
  frame.addSensor(SENSOR_TIME, RTC.hour, RTC.minute, RTC.second );
  // 4.3 Imprimir a frame
  frame.showFrame();

  // 5. Enviar a mensagem
  ...
  // 6. Entrar em modo Deep Sleep
  USB.println(F("Going to sleep..."));
  USB.println();
  PWR.deepSleep(sleepTime, RTC_OFFSET, RTC_ALM1_MODE1, ALL_OFF);
}

```

Figura 21 Código fonte de programa Waspnote

Como se pode ver, o funcionamento dos dispositivos é cíclico, em que os sensores lêem os valores dos parâmetros ambientais, e o dispositivo processa-os,

constrói uma *frame*, envia a *frame* para o *gateway* e entra em modo de baixo consumo, repetindo o processo ao fim do tempo estipulado.

Apesar de termos usado valores reais obtidos pelo equipamento que tínhamos à disposição, tornou-se necessário criar um simulador, para a obtenção de ainda mais valores em simultâneo, e ainda de diferentes valores e parâmetros que aqueles que os sensores de que dispunhamos podiam ler.

As principais características desse simulador são:

- A capacidade de simular múltiplos *motes* em simultâneo;
- O facto de cada *mote* simulado ter uma *frame* completamente diferente, inclusive no tipo de *frame*, ou ASCII, ou binário;
- A capacidade de simular um número diferente de sensores em cada *mote*, e sensores de tipo diferente;
- Envio de *frames* ao longo de um tempo estabelecido, e com intervalos definidos para cada *mote*;
- Variação aleatória do valor de cada sensor, à volta do valor estabelecido para esse mesmo sensor, e dentro de limites escolhidos para cada sensor.

Este simulador envia as *frames* criadas diretamente para o servidor de *Web Services*, e permite-nos simular nós mais completos em conjunto com os dispositivos reais.

3.4.5. Visualização de Dados

Apesar de não ser o principal foco desta dissertação, desenvolveu-se um estudo de possíveis formas de visualização da informação obtida e armazenada pelo sistema, que sejam intuitivas e de fácil compreensão para o utilizador.

O principal desafio na criação destas formas de visualização prende-se com a forma dinâmica como todo o sistema foi desenhado, incluindo a própria base de dados e consequentemente a forma como a informação está armazenada. Isto leva a que num momento em que um utilizador pretenda ver a informação contida no sistema sobre um conjunto de nós que constituam o grupo a que tem acesso, sejam completamente variáveis os seguintes fatores chave:

- O número de nós;
- O número de sensores para cada nó, que podem ser diferentes para nós diferentes;
- Para cada sensor em cada nó, o parâmetro ambiental que mede, e a profundidade a que o mede.

O protótipo de aplicação, programado em C#, pretende responder a estes desafios, tendo sobretudo duas preocupações chave: a criação dos componentes necessários à representação dos valores de todos os sensores pertencentes a todos os nós em causa, sem criar componentes em excesso; e a reutilização do mesmo tipo de componentes para a representação de valores dos vários parâmetros ambientais, e provenientes de todo o tipo de profundidades, adaptando as características desses mesmos componentes às necessidades no momento da criação de cada um.

Ao ser efetuado um pedido pelo utilizador, é feita uma abordagem orientada ao nó, isto é, são obtidos os vários nós que fazem parte do pedido, e segue-se a criação de componentes nó a nó. A visualização da informação em cada nó é feita em duas vistas: uma vista mais resumida, em que é apresentada a informação relativa aos valores atuais dos parâmetros medidos naquele nó (último valor medido), e uma vista mais detalhada, que para cada nó apresenta mais informação, como médias e variações dos parâmetros ao longo do tempo.

Tomando o caso da visão resumida de cada nó, em que são apresentados os últimos valores medidos pelos sensores desse nó, como um exemplo da forma dinâmica como os componentes são criados para a mostragem desses valores, apresenta-se a Tabela seguinte que representa a forma como essa criação de componentes é feita:

Profundidade\Parâmetro	Parâmetro 1	...	Parâmetro N
Profundidade 1	Valor(1,1)	...	Valor(N,1)
...
Profundidade M	Valor(1,M)	...	Valor(N,M)

Tabela 3 Tabela representativa da forma de criação de componentes

Assim, para cada nó, serão no máximo necessários $N \times M$ componentes para mostrar a informação relevante proveniente do nó e da sua localização.

Para cada par <Parâmetro, Profundidade> é verificado se existe algum sensor que recolha medições desse parâmetro e a essa profundidade, e, caso exista, é criado um componente que faça a mostragem do último valor obtido.

O componente usado para a exibição do valor atual de um determinado parâmetro é do tipo *gauge*, e é utilizado um componente *gauge* da API de Visualização da *Google (Google Visualization API)*. A Figura 22 mostra este componente.



Figura 22 Componente gauge

A utilização deste componente vai para além da simples mostra do valor pretendido ao utilizador. É possível definir a escala, o valor mínimo e máximo do *gauge*, e ainda a representação gráfica de zonas que equivalham a intervalos de valores. Esta última opção de personalização é ideal para a representação dos alarmes definidos para o sensor em causa, sendo utilizada para esse fim.



Figura 23 Gauge com alarme de valor mínimo

Na Figura 23 podemos ver a representação de um valor para o qual está definido um limite mínimo, a partir do qual será disparado um alarme. Neste caso o valor está num intervalo em que se podem considerar normal. Já na Figura 24 vemos um *gauge* representativo dum valor proveniente de um sensor que tem definidos um alarme de valor mínimo admissível e um alarme de valor máximo admissível. Neste caso, o valor está ligeiramente acima do que pode ser considerado normal, e existe uma situação de alarme.



Figura 24 Gauge com alarmes de valor mínimo e valor máximo

Importa também referir como são determinados os valores mínimos e máximos de cada *gauge*. A prioridade neste capítulo vai para os intervalos de valores que estejam definidos explicitamente para o parâmetro representado pelo *gauge*, caso existam. Caso não estejam definidos limites mínimos e máximos, ou caso esteja definido apenas um deles, é utilizado o valor a mostrar e possíveis

alarmes que estejam definidos para aquele sensor para calcular os restantes limites.



Figura 25 Gauge com intervalo de valores definidos

Na Figura 25 podemos ver um *gauge* criado para representar um valor de nível de pH, para o qual está definido o intervalo de valores possíveis, que é de entre 0 e 14. Na Figura 26 podemos ver a forma como o componente lida com sucesso com a representação dos limites dos valores de nível de pH e a existência de um alarme em simultâneo.



Figura 26 Gauge com intervalo de valores definidos e um alarme

Olhando agora para a representação de informação mais detalhada referente a cada nó, são utilizados componentes *Chart* da *framework* da *Microsoft* para mostrar a variação de valores ao longo do tempo.

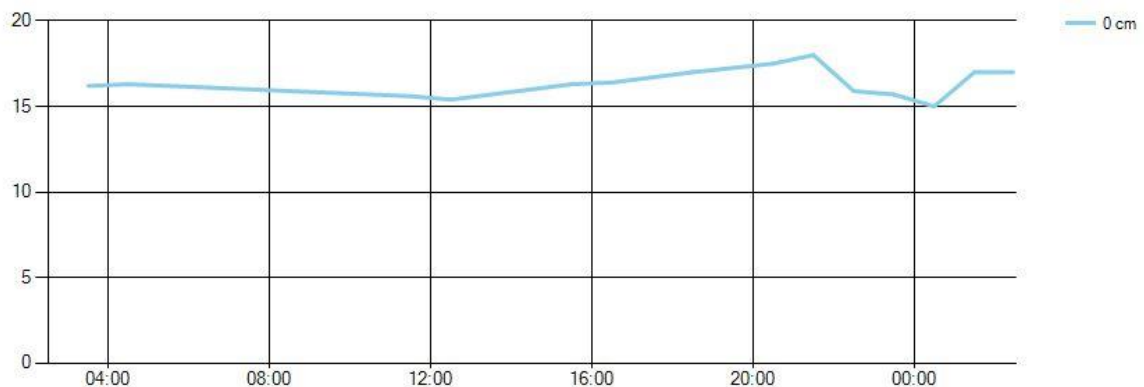


Figura 27 Componente chart

A criação deste tipo de componentes, no contexto de um nó, obedece aos seguintes princípios:

- Um *chart* para cada variável ambiental;
- Uma série de valores para cada profundidade a que valores dessa variável ambiental sejam medidos no nosso em causa.

Assim, e de forma dinâmica, a aplicação cria um *chart* semelhante àquele que se encontra na Figura 27, no caso de uma variável ambiental que é medida apenas a uma profundidade no nó em causa, ou um *chart* semelhante ao apresentado na Figura 28, no caso de essa variável ser medida em duas profundidades diferentes nessa localização.

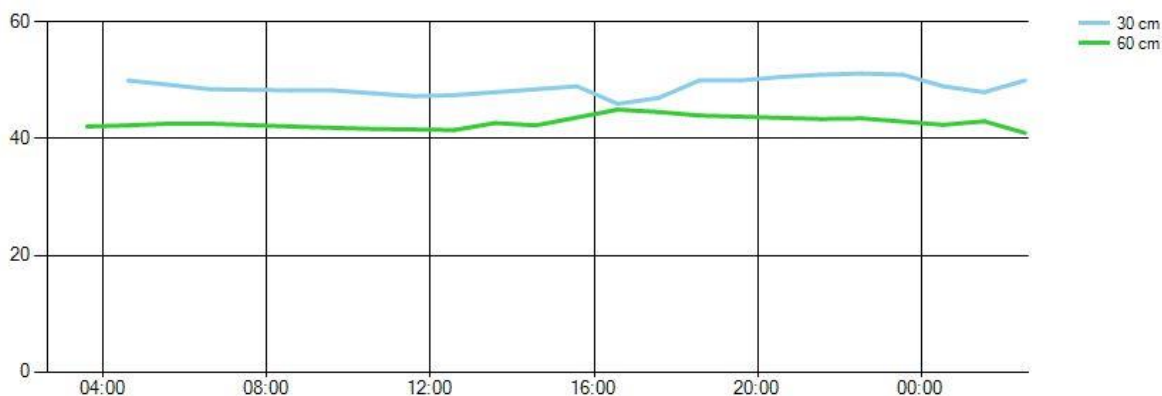


Figura 28 Componente *chart* com duas séries

De referir que os *charts* estão configurados para mostrar a evolução dos valores obtidos durante as últimas 24 horas, mas o princípio será o mesmo para outros intervalos temporais.

Outra informação que é apresentada de forma intuitiva num *chart*, se estiver definida para os sensores em causa, são os alarmes mínimos e máximos definidos para os mesmos. Essa apresentação faz-se com recurso a linhas horizontais, de cor correspondente à da série em questão, conforme se vê na Figura 29, caso em que existem duas séries, e duas linhas horizontais que representam dois alarmes mínimos, um referente aos valores lidos a cada uma das profundidades.

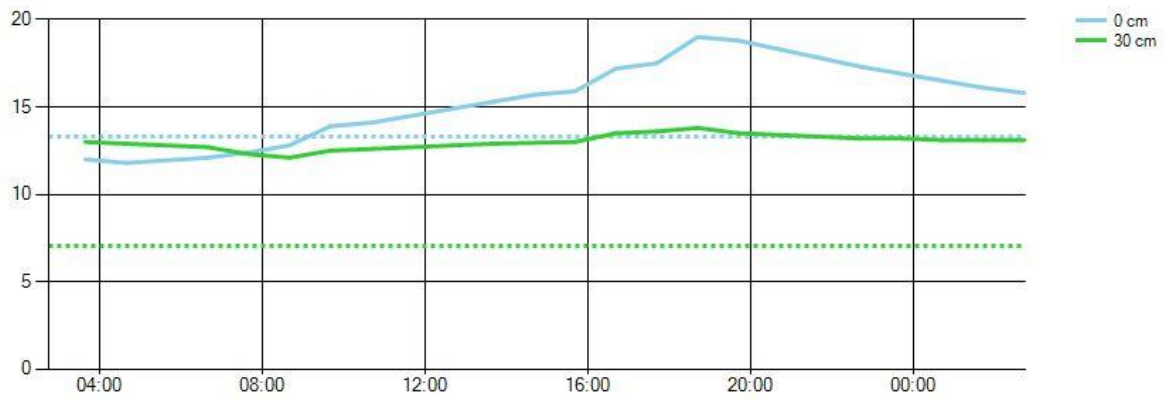


Figura 29 Componente chart com duas séries com alarmes de valor mínimo

São também apresentados, nesta secção de detalhes referentes a cada nó, a média das últimas 24 horas de cada valor, apresentação feita em texto.

Para além de todas estas informações apresentadas referentes a cada nó e a cada sensor, e de todos os componentes criados de forma dinâmica já enumerados e explicados em detalhe, foi ainda desenvolvida uma forma de visualização mais geral, que engloba vários nós do sistema.

Esta forma baseia-se num componente do tipo mapa, sendo usado aqui a *API Google Maps*, sobejamente conhecida. O objetivo da introdução de um mapa no nosso protótipo passa por dar uma visão rápida e intuitiva da localização dos nós e do seu estado, antes de o utilizador passar à análise detalhada da informação de cada um deles.

A Figura 30 mostra este componente em execução.

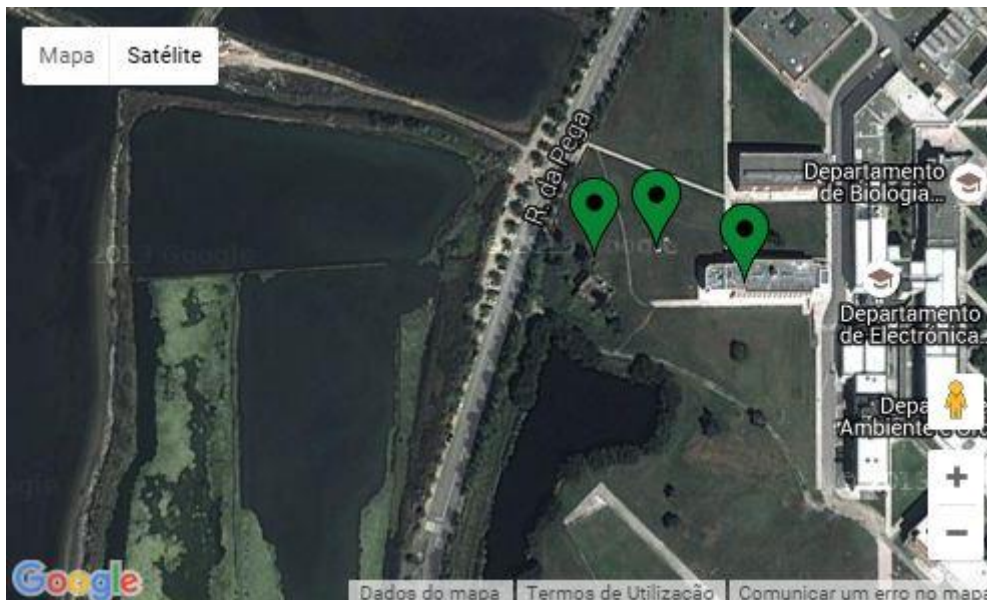


Figura 30 Componente mapa

Este componente utiliza ainda os dados sobre os valores atuais dos sensores de cada nó, e os valores de alarme que estejam definidos para cada um

desses sensores, para alertar o utilizador de forma rápida caso algum valor esteja fora do intervalo pretendido. Uma situação em que existe pelo menos um sensor num dos nós com o último valor acima do valor definido como máximo, ou com o último valor abaixo do valor definido como mínimo, e a representação pelo mapa desse facto, pode ser vista na Figura 31.

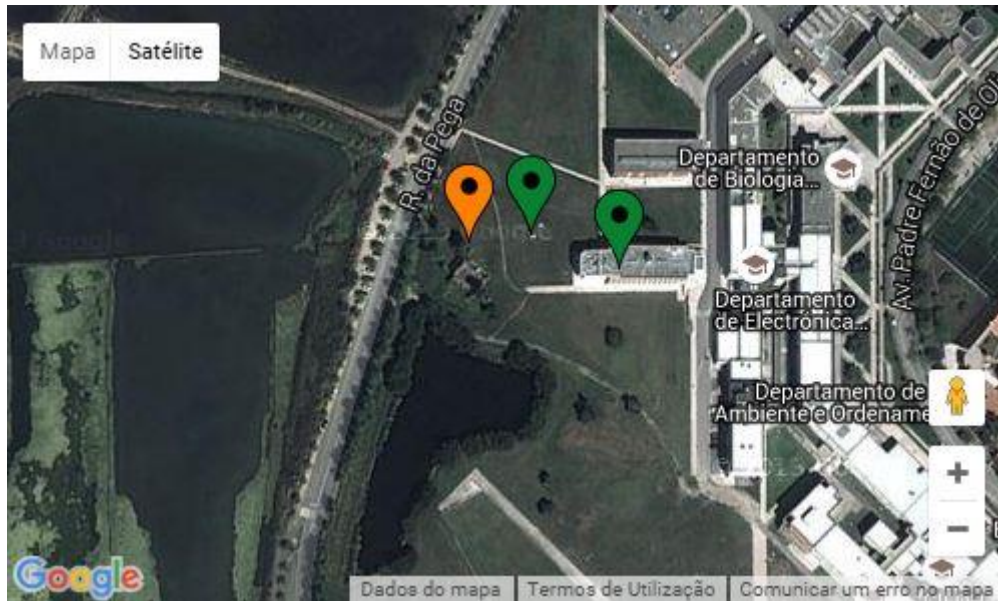


Figura 31 Componente mapa com um valor atual que faz disparar um alarme

3.5. Testes de Comunicação

Neste subcapítulo serão apresentados exemplos de comunicação entre os vários componentes da solução, como:

- Comunicação entre os *motes*, o *gateway* e a aplicação monitor de porta série;
- Comunicação entre a aplicação monitor de porta série e o servidor de *Web Services*;
- Comunicação entre o servidor de *Web Services* e a base de dados.

A comunicação entre os *motes* e a aplicação monitor de porta série pode fazer-se de duas formas:

- Diretamente entre o *mote* e a aplicação, quando o *mote* está ligado à porta USB do computador;
- Passando pelo *Wasp mote Gateway*, em modo de funcionamento normal, comunicando com este através da rede WSN. O *gateway* passa então os dados à porta USB à qual se encontra ligado, e a aplicação recolhe-os.

Em cada uma destas duas formas os *motes* podem enviar *frames* de dois tipos, *frames* ASCII e *frames* do tipo binário.

Foram realizados testes de comunicação entre *motes* e a aplicação nos quatro cenários possíveis:

- *Mote* diretamente ligado à porta USB e a enviar *frames* ASCII;
- *Mote* diretamente ligado à porta USB e a enviar *frames* do tipo binário;
- *Mote* a comunicar com o *gateway* por rede sem fios e este ligado à porta USB, com o *mote* a enviar *frames* ASCII;
- *Mote* a comunicar com o *gateway* por rede sem fios e este ligado à porta USB, com o *mote* a enviar *frames* do tipo binário.

Em todos estes cenários foi comprovada a correta comunicação entre os *motes* e a rede por eles formada e a aplicação, mediante a correta programação dos *motes* e a correta configuração da aplicação. Esta comunicação é de extrema importância, já que é ela que vai permitir que os dados lidos pelos sensores

cheguem ao sistema, onde serão tratados e processados, antes de serem armazenados como informação útil.

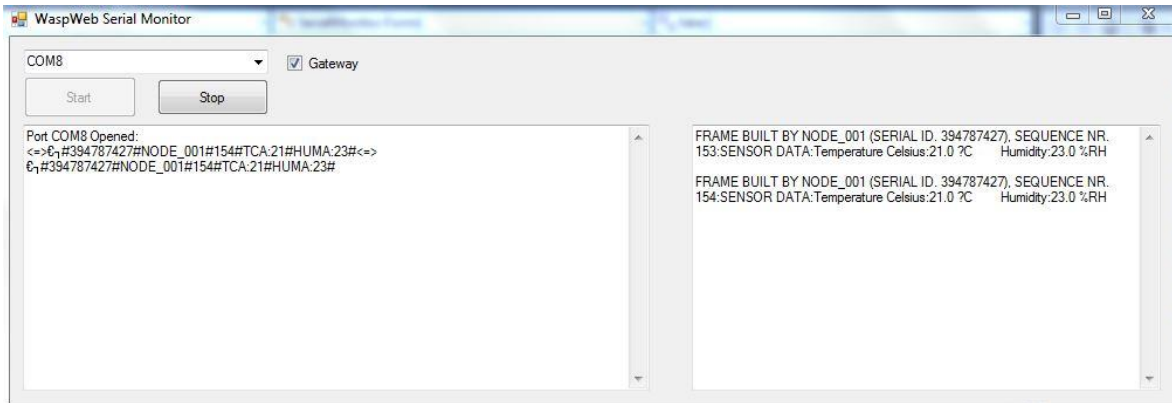


Figura 32 Comunicação, entre a rede e a aplicação através da porta série, de duas *frames* ASCII

Na Figura 32 podemos ver um exemplo de *frames* ASCII criadas por um dispositivo *Waspnote*, enviadas por este através da rede WSN para o *gateway*, e entregues pela porta USB pelo *gateway* à aplicação monitor de porta série. Este tipo de *frames* é em grande parte legível, e neste exemplo podemos ainda ver a resposta do serviço de *parsing* invocado após a recepção das *frames*.

Na Figura 33, vemos um exemplo de *frames* do tipo binário, criadas pelo mesmo dispositivo *Waspnote* após a sua configuração para o efeito. Estas *frames* chegam da mesma forma à aplicação monitor de porta série, e são tratadas da mesma forma. No entanto, este tipo de *frames* não é de fácil compreensão, oferecendo em troca um tamanho consideravelmente menor, o que facilita a sua transmissão, bem como permite poupar energia ao nível dos dispositivos da rede sensorial, fator chave neste tipo de tecnologias. Podemos então ver a identificação da constituição das *frames*, *byte a byte*, e a correta identificação de cada *frame* recebida em série.

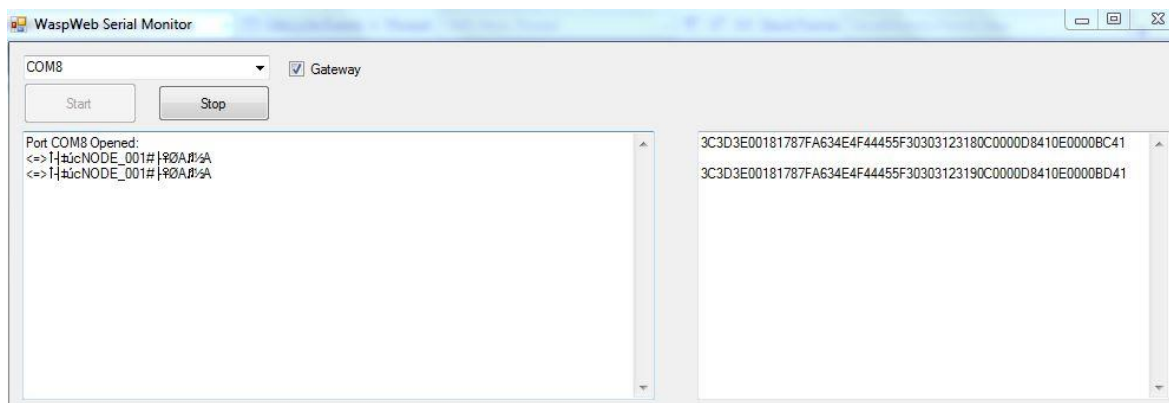


Figura 33 Comunicação, entre a rede e a aplicação através da porta série, de duas *frames* binárias

Já a comunicação entre a aplicação que monitoriza a informação recebida da rede sensorial e o servidor de *Web Services* é sempre realizada mediante a receção de uma *frame* proveniente da rede e entregue na porta USB, após ter sido identificada e delimitada.

Esta comunicação, que não é mais que a invocação de um *web service*, faz-se através do protocolo HTTP, e o servidor produz respostas conforme o serviço que tiver sido requisitado.

Foram realizados testes nos seguintes cenários:

- Envio de *frame* no formato ASCII para o serviço de *parsing*;
- Envio de *frame* no formato binário para o serviço de *parsing*;
- Envio de *frame* no formato binário para o serviço de *parsing*, conversão e armazenamento.

Em todos os casos pode observar-se a correta comunicação entre estes dois componentes do sistema, e verifica-se a receção da resposta do servidor pela aplicação, e ainda a sua correção tendo em conta a *frame* enviada na invocação do *web service*.

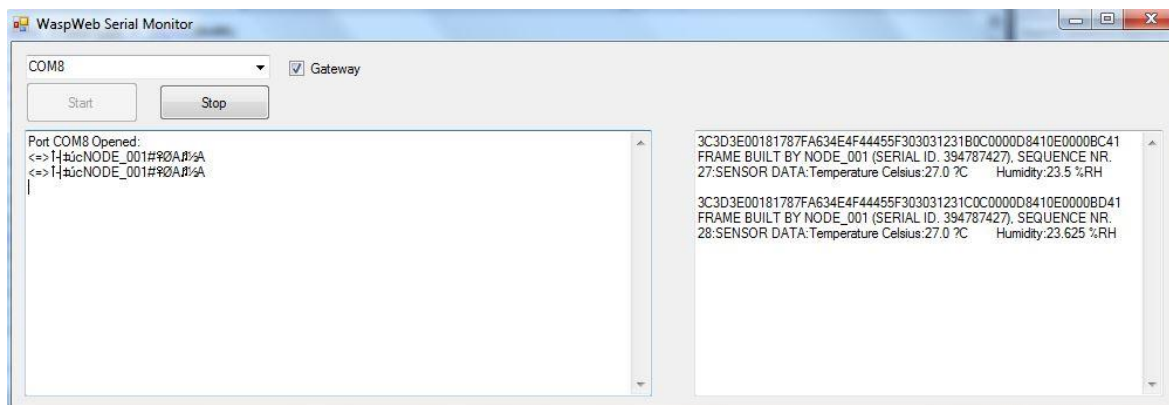


Figura 34 Comunicação, entre a rede e a aplicação, de duas *frames* binárias e invocação do *webservice* de *parsing*

Na Figura 34 podemos ver um exemplo em que *frames* provenientes de um dispositivo *Wasp mote*, e do tipo binário, são recebidas pela aplicação monitor de porta série e de seguida usadas na invocação do serviço de *parsing*, e podemos ainda observar a receção da resposta à invocação do serviço para cada *frame*.

Já na Figura 35, vemos um exemplo em que *frames* criadas pelo mesmo dispositivo, e também do tipo binário, são recebidas pela aplicação e usadas na invocação do *webservice* que permite o *parsing*, a conversão e o armazenamento da informação extraída das mesmas na base de dados. Observa-se a resposta recebida pela aplicação, e proveniente do servidor, à invocação com sucesso do *webservice* para cada uma das *frames*.

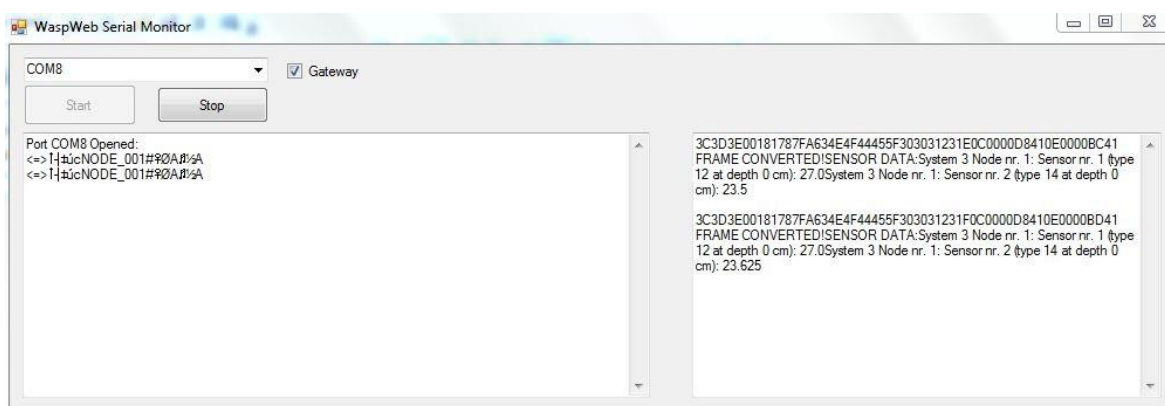


Figura 35 Comunicação entre a rede e a aplicação e invocação do *webservice* de *parsing*, conversão e armazenamento

Por último, foi testada a comunicação entre o *web service* e a base de dados, mediante uma invocação do serviço de *parsing*, conversão e armazenamento pela aplicação, tal como visto no último exemplo.

Após o processamento da *frame* realizado pelo servidor de *Web Services*, a informação recolhida da *frame* pode dar origem a uma ou mais entradas na tabela *SensorData* da base de dados. Essa introdução dos novos dados foi verificada, e mostra-se correta nos testes realizados, o que permite verificar a comunicação entre os dois componentes.

A Figura 36 demonstra este comportamento, com a reflexão na base de dados da informação contida nas *frames* observadas na Figura 35.

	ID	SiteID	NodeNumber	SensorNumber	Value	Date Time
1812	1812	4	1	1	17	2015-11-17 22:33:33.150
1813	1813	4	1	1	14	2015-11-18 03:33:22.690
1814	1814	4	1	1	13	2015-11-18 05:33:16.030
1815	1815	4	1	1	14	2015-11-18 08:33:11.830
1816	1816	4	1	1	13	2015-11-18 11:33:03.173
1817	1817	4	1	1	11	2015-11-18 14:32:58.937
1818	1818	4	1	1	9	2015-11-18 18:32:53.553
1819	1819	3	1	1	27	2015-11-19 15:26:22.610
1820	1820	3	1	2	23,5	2015-11-19 15:26:22.847
1821	1821	3	1	1	27	2015-11-19 15:26:23.017
1822	1822	3	1	2	23,625	2015-11-19 15:26:23.017

Figura 36 Verificação da comunicação entre o servidor de *webservices* e a base de dados

4. CONCLUSÕES E TRABALHO FUTURO

Tendo sido apresentada a teoria por trás desta dissertação, e tendo também sido explicitada a forma como foi desenvolvida a solução, as decisões tomadas e a implementação das mesmas, fazemos agora algumas considerações e tiramos conclusões sobre o trabalho realizado.

4.1. Conclusões

O objetivo desta dissertação passava por projetar e implementar uma solução abrangente, que permitisse realizar leitura de dados ambientais por vários dispositivos ligados por redes do tipo *Wireless Sensor Network*, e efetuar a sua recolha, processamento e armazenamento, para além de desenvolver um estudo de formas dinâmicas de representação da informação obtida. Esta abrangência levou à utilização de vários componentes, desde dispositivos com capacidades de leitura de dados ambientais e de comunicação, meios de recolha de dados da rede formada por esses dispositivos, uma camada intermédia, por via de um servidor de *Web Services*, que realizasse o tratamento dos dados, transformando-os em informação útil, e ainda uma base de dados, devidamente estruturada, para o armazenamento dos dados, para além de um servidor *web* que se ligue a essa mesma base de dados e que permita a visualização da informação recolhida pelo utilizador.

O protótipo desenvolvido, já descrito com detalhe no ponto 3.4, é uma implementação da arquitetura já referida, e pretende-se que o seu funcionamento seja o idealizado nessa mesma arquitetura. Além disso, um dos pontos chave passa pela capacidade de os componentes lidarem com os tipos de dados e com a informação de forma dinâmica. Este protótipo responde a estas questões levantadas, desde o modelo de dados, a forma como os dados sensoriais são recolhidos e processados, e ainda as formas de representação dos mesmos ao utilizador.

4.2. Problemas encontrados

Ao longo do tempo em que a solução foi sendo projetada e implementada, surgiram alguns problemas, desde problemas pontuais, de correção simples, quer problemas estruturais, que obrigaram a mudanças com alguma profundidade.

Entre esses problemas estruturais estão problemas relacionados com o modelo de dados, com casos em que esse modelo foi sendo aperfeiçoado através da introdução de novos campos e/ou tabelas, quer para o suporte de novas funcionalidades, quer para aumentar o comportamento dinâmico do sistema.

Alguns problemas relacionados com a transmissão de *frames* em bruto através de HTTP foram também encontrados, e a sua resolução baseou-se no uso da codificação em *base64* dos dados da *frame* em fase prévia ao seu envio. Esta codificação pode aumentar o tamanho total dos dados a enviar até 33%, mas garante o correto funcionamento do sistema para todas as *frames*, qualquer que seja o seu tipo e conteúdo.

Vários outros problemas menores foram sendo encontrados e resolvidos. No entanto, existem questões em que podem ser introduzidas melhorias.

4.3. Trabalho Futuro

Apesar do trabalho realizado durante este período, existem muitas situações que podem ser melhoradas, especialmente se tivermos em conta a abrangência da solução desenvolvida.

Em primeiro lugar, a aplicação monitor de porta série, usada para a recolha dos dados provenientes da WSN e para o seu envio para o servidor de *Web Services*, pode ser melhorada, quer ao nível da *interface* com o utilizador, quer através do acrescento de funcionalidades de monitorização da rede que dêem mais informação ao utilizador.

Para além disso, podem ser desenvolvidas novas formas de recolha de dados de uma rede WSN, para que o sistema seja exequível noutro tipo de cenários em que esta forma de recolha não pode ou não deve ser utilizada.

Podem ainda ser estudadas novas formas de visualização de informação, para além do aperfeiçoamento das formas já desenvolvidas. Isto para além de uma *interface* mais intuitiva para o utilizador, que englobe todos estes componentes.

Funcionalidades de controlo de sessão, associadas a outras questões de segurança na transmissão dos dados, representam também um campo que é possível explorar para melhoria da qualidade e fiabilidade da solução apresentada.

Por último, o modelo de dados deverá continuar a ser ajustado às novas exigências que possam surgir.

4.4. Considerações Finais

Através da investigação, projeção e implementação de uma solução abrangente, e da ultrapassagem de vários problemas que foram aparecendo ao longo de todo este processo, foram desenvolvidas capacidades e conhecimentos em vários domínios e sobre variadas tecnologias e plataformas.

Este conhecimento surge como uma mais valia no percurso académico, mas também no percurso pessoal.

5. BIBLIOGRAFIA

ALADIN Consortium. “ALADIN” (2015), [Online]. Available: <http://www.cnrm.meteo.fr/aladin/> [Acedido em Outubro 2015].

AlSkaif, T., Zapata, M. G., & Bellalta, B. - Game theory for energy efficiency in Wireless Sensor Networks: Latest trends. Journal of Network and Computer Applications. Vol. 54 (2015), p. 33- 61. doi:10.1016/j.jnca.2015.03.011

AMI. “AMI Turf Irrigation Controllers” (2015), [Online] . Available: <https://aquamanagement.com/vertical-solutions/ami-turf-irrigation-controllers/> [Acedido em Outubro 2015].

Barroca, N., Borges, L., Fernando, V., Monteiro, F., Górski, M. & Castro-Gomes, J. - Wireless sensor networks for temperature and humidity monitoring within concrete structures. Construction and Building Materials. Vol. 40 (2013), p. 1156 – 1166. doi:10.1016/j.conbuildmat.2012.11.087.

Chaloo, R., Oladeinde, A., Yilmazer, N., Ozcelik, S. & Chaloo, L. - An Overview and Assessment of Wireless Technologies and Co- existence of ZigBee, Bluetooth and Wi-Fi Devices. Procedia Computer Science. Vol. 12 (2012), p. 386- 391. doi:10.1016/j.procs.2012.09.091

Dener, M. & Bostancıoğlu, C. - Smart Technologies with Wireless Sensor Networks. Procedia – Social and Behavioral Sciences. Vol. 195 (2015), p. 1915 – 1921. doi:10.1016/j.sbspro.2015.06.202

Dener, M., Özköka, Y., & Bostancıoğlu, C. - Fire Detection Systems in Wireless Sensor Networks. Procedia – Social and Behavioral Sciences. Vol. 195 (2015), p. 1846 – 1850. doi:10.1016/j.sbspro.2015.06.408

Diaz, S. E., Pérez, J. C., Mateos, A. C., Marinescu, M., & Guerra, B. B. - A novel methodology for the monitoring of the agricultural production process based on wireless sensor networks. Computers and Eletronics in Agriculture. Vol. 76, nº 2 (2011), p. 252 – 265. doi:10.1016/j.compag.2011.02.004

Esri. “ArcGIS” (2015), [Online]. Available: <http://www.esri.com/software/arcgis/capabilities> [Acedido em Outubro 2015].

Lian, Hsiao, Sung - Intelligent multi-sensor control system based on innovative technology integration via ZigBee and Wi-Fi networks. Journal of Network and Computer Applications. doi 10.1016/j.jnca.2012.12.012. Vol. 36, nº 2 (2013) p. 756 - 767

Libelium. “Waspote Documentation” (2015), [Online]. Available: <http://www.libelium.com/development/waspote/documentation/> [Acedido em Abril 2015].

Lo, C., Lynch, J. P., & Liu, M. - Distributed model-based nonlinear sensor fault diagnosis in wireless sensor networks. Mechanical Systems and Signal Procecing. Vol. 66-67 (2016), p. 470 – 484. doi:10.1016/j.ymsp.2015.05.011

Ojha, T., Misra, S., & Raghuwanshi, N. S. - Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. Computers and Eletronics in Agriculture. Vol. 118 (2015), p. 66 – 84. doi:10.1016/j.compag.2015.08.011

OnFarm. “OnFarm: Grow Informed” (2015), [Online]. Available: <http://www.onfarm.com/> [Acedido em Setembro 2015].

Silva, A., Câmara, N., Nunes, L. & Abreu, F. - Aplicações SIG em Agrometeorologia: casos da temperatura na pêsra rocha e na vinha. 9º Encontro de Utilizadores ESRI - EUE (2011).

Sung, W. & Hsu, Y. - Designing an industrial real-time measurement and monitoring system based on embedded system and ZigBee. Expert Systems with Applications. Vol. 38, nº 4 (2011), p. 4522 – 4529. doi:10.1016/j.eswa.2010.09.126

ANEXO A

Neste anexo, será descrito com maior detalhe o modelo de base de dados utilizado, passando pela descrição detalhada das várias tabelas, e utilidade dos seus vários campos.

Na Figura 37 podemos ver o diagrama da base de dados.

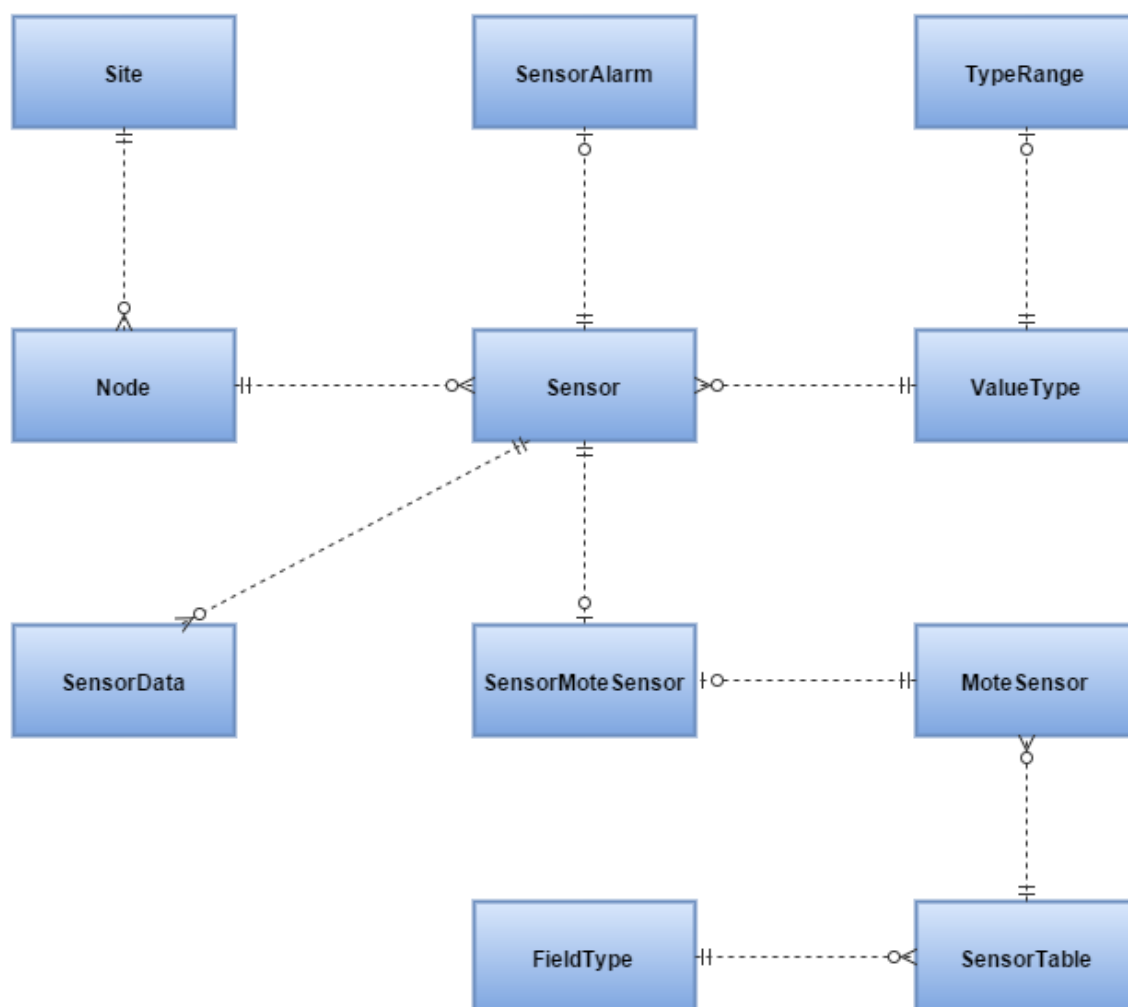


Figura 37 Diagrama de Base de Dados

Segue-se a descrição detalhada de cada uma das tabelas.

Site

A tabela *Site* permite guardar informação específica de cada conjunto de nós relacionados, bem permite que esse conjunto seja referenciado inequivocamente, para efeitos de agregação e visualização de informação, bem como possivelmente para controlo de acesso.

Atributo	Descrição
<u>SiteID</u>	Identificador do Site
SiteName	Nome do Site

Tabela 4 Entidade Site

A entidade modelada por esta tabela relaciona-se com a entidade *Node*, numa relação de um-para-muitos.

Node

A tabela *Node* permite guardar informação específica de cada nó de recolha de dados ambientais, o que permitirá relacionar a informação recolhida com a posição geográfica em que foi recolhida.

Atributo	Descrição
<u>SiteID</u>	Identificador do Site
<u>NodeNumber</u>	Identificador do Nó no contexto do Site
NodeLatitude	Latitude geográfica do Nó
NodeLongitude	Longitude geográfica do Nó

Tabela 5 Tabela Node

A entidade *Node* relaciona-se com a entidade *Site* numa relação de muitos-para-um, e ainda com a entidade *Sensor*, numa relação de um-para-muitos.

Sensor

Esta tabela permite armazenar a informação relativa a cada sensor que se encontre no terreno, o que permitirá mapear os dados recebidos pelo sistema e permitirá ainda a sua análise de forma posterior.

Atributo	Descrição
<u>SiteID</u>	Identificador do Site
<u>NodeNumber</u>	Identificador do Nó no contexto do Site
<u>SensorNumber</u>	Identificador do Sensor no contexto do Nó
TypeID	Identificador da variável medida pelo Sensor
Depth	Profundidade a que o Sensor se encontra

Tabela 6 Tabela Sensor

A entidade modelada por esta tabela relaciona-se com a entidade *Node* da forma já referida. Para além dessa relação, esta entidade possui outras, a saber:

- com a entidade *SensorAlarm*, de um-para-zero-ou-um;
- com a entidade *ValueType*, de muitos-para-um;
- com a entidade *SensorData*, de um-para-muitos;
- com a entidade *MoteSensor*, através da tabela de relação *SensorMoteSensor*.

SensorAlarm

Esta tabela permite armazenar informação acerca de valores mínimos e máximos que se entendam aceitáveis para determinada variável ambiental medida por um determinado sensor. A análise da informação presente nesta tabela permite o lançamento de alertas aquando da receção de um valor não desejado.

Atributo	Descrição
<u>SiteID</u>	Identificador do Site
<u>NodeNumber</u>	Identificador do Nó no contexto do Site
<u>SensorNumber</u>	Identificador do Sensor no contexto do Nó
MinAlarm	Valor mínimo aceitável para o Sensor
MaxAlarm	Valor máximo aceitável para o Sensor

Tabela 7 Tabela SensorAlarm

Esta tabela relaciona-se com a tabela Sensor, tal como já foi referido.

ValueType

A tabela *ValueType* permite guardar informação relativa a uma determinada variável medida por um ou mais sensores de forma transversal a todo o sistema. Permite ainda referenciar essa variável no relacionamento entre as demais entidades.

Atributo	Descrição
<u>TypeID</u>	Identificador da variável ambiental
TypeName	Nome da variável
TypeUnit	Unidade dos valores utilizados

Tabela 8 Tabela ValueType

A entidade *ValueType* relaciona-se com a entidade Sensor numa relação de tipo um-para-muitos, e ainda com a entidade *TypeRange*, numa relação de um-para-zero-ou-um.

TypeRange

Esta tabela permite armazenar valores de intervalos que sejam intrínsecos a uma determinada variável, para efeitos de visualização, e possivelmente de deteção de erros.

Atributo	Descrição
<u>TypeID</u>	Identificador da variável ambiental
MinRange	Valor mínimo intrínseco
MaxRange	Valor máximo intrínseco

Tabela 9 Tabela TypeRange

Esta entidade relaciona-se com a entidade *ValueType*, da forma já anteriormente referida.

SensorData

Esta tabela permite o armazenamento, e posterior processamento e visualização, da informação obtida por todos os sensores da rede sensorial, tendo em conta a sua localização, o tipo de valor recolhido, e a referência temporal da sua obtenção.

Atributo	Descrição
<u>ID</u>	Identificador único da informação
SiteID	Identificador do Site
NodeNumber	Identificador do Nó no contexto do Site
SensorNumber	Identificador do Sensor no contexto do Nó
Value	Valor obtido pelo Sensor
DateTime	Data e hora da obtenção do Valor

Tabela 10 Tabela SensorData

Esta entidade relaciona-se com a entidade Sensor, numa relação de muitos-para-um.

SensorMoteSensor

Esta tabela permite especificar a correspondência entre sensores virtuais do sistema, e sensores reais da plataforma *Waspote* presentes no terreno.

Atributo	Descrição
<u>SiteID</u>	Identificador do Site
<u>NodeNumber</u>	Identificador do Nó no contexto do Site
<u>SensorNumber</u>	Identificador do Sensor no contexto do Nó
MoteSerialID	Identificador único do dispositivo Waspote
MoteSensorID	Tipo do sensor físico no dispositivo Waspote
MoteIndex	Índice da colocação do sensor físico nas frames

Tabela 11 Tabela SensorMoteSensor

Esta tabela permite mapear a relação entre as entidades Sensor e *MoteSensor*.

MoteSensor

Esta entidade permite a especificação de sensores da plataforma *Waspote* no sistema, para permitir a conversão de informação recebida pelo sistema e proveniente desse tipo de sensores em informação mapeada com os sensores virtuais definidos.

Atributo	Descrição
<u>MoteSerialID</u>	Identificador único do dispositivo Waspote
<u>MoteSensorID</u>	Tipo do sensor físico no dispositivo Waspote
<u>MoteIndex</u>	Índice da colocação do sensor físico nas frames

Tabela 12 Tabela MoteSensor

Esta entidade relaciona-se com a entidade Sensor, através da tabela de relação *SensorMoteSensor*, e com a entidade *SensorTable*, numa relação de muitos-para-um.

SensorTable

Esta tabela permite mapear os tipos de sensores disponíveis para a plataforma *Waspote*, de forma a que seja possível efetuar o *parsing* aos dados recebidos em bruto pelo sistema. A informação contida nesta tabela é definida pelo fabricante, mas é mutável, pelo que está armazenada em base de dados e é utilizada em *runtime* pelos serviços de *parsing*.

Atributo	Descrição
<u>SensorID</u>	Identificador do tipo de sensor físico
Name	Nome do sensor físico
AsciID	Identificador ASCII do tipo de sensor físico
NumberOfFields	Número de campos do valor
TypeID	Tipo dos campos do valor
FieldSize	Tamanho dos campos do valor
Unit	Unidade dos valores recolhidos pelo sensor

Tabela 13 Tabela SensorTable

Esta tabela relaciona-se com a tabela *MoteSensor*, numa relação de um-para-muitos, e com a tabela *FieldType*, numa relação de muitos-para-um.

FieldType

Esta tabela permite armazenar informação relativa a um tipo de dados em que informação recolhida por sensores físicos *Waspote* é enviada na *frame*.

Atributo	Descrição
<u>TypeID</u>	Identificador do tipo de dados
TypeName	Nome do tipo de dados

Tabela 14 Tabela *FieldType*

Esta tabela relaciona-se com a tabela *SensorTable*, numa relação de um-para-muitos.