



Licenciatura em
Gestão de Sistemas e Computação

Tipo de Trabalho
Trabalho Final de Licenciatura

Título do Trabalho
Modelo de Sistema baseado em Rede de Sensores sem Fios com Tecnologia *Open Source*
para Gerar Alertas de Incêndios Florestais

Elaborado por
José Lavigne Neto

Nº de estudante
20111593

Orientado por
Professor Doutor José Braga de Vasconcelos

Barcarena, __ Novembro de 2018

Universidade Atlântica

Licenciatura em Gestão de Sistemas e Computação

**Modelo de Sistema baseado em Rede de Sensores sem Fios com Tecnologia *Open Source*
para Gerar Alertas de Incêndios Florestais**

Trabalho Final de Licenciatura

Elaborado por José Lavigne Neto

Discente Nº 20111593

Orientador Professor Doutor José Braga de Vasconcelos

Barcarena

Novembro de 2018

O autor é o único responsável pelas ideias expressas neste relatório.

Agradecimentos

***“Os sonhos são projetados por nosso arquiteto interior,
mas a realização está em nossas mãos.”***

Friedrich Nietzsche

Chegar ao alto de uma montanha exige o sacrifício de ter de completar todo o caminho e superar todas as dificuldades. É na verdade o acumular de sucessivos passos realizados com sucesso, mas que foram feitos um de cada vez e que têm de ser concluídos conscientemente.

A verdade é que no início, ao olhar para cima e ver o caminho todo pela frente, temos de acreditar que iremos conseguir chegar lá. Acreditar que é possível vencer todos os desafios. O sonho e a vontade de vencer fizeram-me seguir em frente e deram-me força durante todo o caminho para continuar durante muitas horas de estudo e poucas horas de sono.

A cada obstáculo encontrado, percebi que nunca estive sozinho e que outros também conspiravam para o meu sucesso. Os desafios lançados pelos Professores tornaram-se em bagagem: Conhecimento para carregar por este e muitos outros caminhos que virão. A estes vai o meu primeiro agradecimento.

Companheiros de caminhada foram também, os colegas do curso de Gestão de Sistemas e Computação, além de colegas de outros cursos que iam na mesma direção, cada um deles superando os seus desafios. A estes além do obrigado por partilharem a viagem, também lhes remeto os parabéns.

À minha Mãe, irmãos e principalmente minha filha, Louise, que fizeram ao meu lado esta escalada desde o início e mais recentemente, a minha amada Neuza, que agora também me acompanha, tenho mesmo muito o que agradecer.

Completar este trabalho de Fim de Curso é o desafio que falta para finalmente chegar ao topo, realizar o sonho e provar que querer é poder. Ao Professor José Braga de Vasconcelos que aceitou ser o meu Orientador para que eu pudesse realizar este trabalho de final de curso vai um agradecimento especial. Sem a sua preciosa ajuda este trabalho não teria sido possível.

Quero partilhar este momento com todos os que colaboraram para que eu aqui chegasse e dizer que a Universidade nos transforma, enriquece-nos com conhecimentos valiosos e abre novos horizontes. Para terminar, digo apenas algo que aprendi neste caminho:

“O saber não ocupa espaço, e sim o vazio de nada saber”

Gabriel de Queiroz Ribeiro.

Resumo

A floresta é um bem comum e tem de ser protegida para garantir a sua preservação e todos os seus benefícios ambientais e sociais.

Em todo o mundo, devido à mudança das condições climáticas verifica-se a ocorrência de temperaturas mais elevadas por períodos mais longos e uma diminuição dos períodos de chuva. Como resultado destas alterações, Portugal passou a enfrentar incêndios mais violentos sendo que alguns atingiram grandes proporções chegando mesmo a ser declarados como incontroláveis, pelas autoridades competentes.

As opções e recursos são limitados e nem sempre eficazes em termos de tempo e custos para prevenir, evitar e combater este problema. Para a prevenção podem ser adotadas diversas medidas, tais como: Consciencialização social, reordenamento da floresta, implementação de leis que restringem a utilização de fogo em períodos de risco elevado, melhoria dos equipamentos de prevenção, qualificação das forças de combate (profissionalização dos bombeiros), diminuição do tempo de reação entre o início do fogo, a sua deteção e o efetivo combate ao mesmo, entre outros.

Neste trabalho de fim de curso é apresentado um modelo conceitual de sistema capaz de recolher dados em tempo real a partir de uma WSN (“*Wireless Sensor Network*”) que serão armazenados numa base de dados localizada num servidor central. A análise dos dados recolhidos permitirá ter uma visão dos incêndios em tempo real, bem como permitirá gerar alertas por e-mail e SMS para os intervenientes parametrizados como destinatários no Sistema.

Considerando a variável custo, condição fundamental para que o modelo apresentado seja viável e para isso, o sistema será desenvolvido com a utilização de software open source ou gratuito e open hardware.

Palavras chave

Open Source, Open hardware, Sistema de Deteção de Incêndio, IoT, Web Services, Rede MESH, Interoperabilidade, GPS, Arduíno, Raspberry Pi.

Abstract

The forest is a common good and it must be protected to ensure its preservation and all the benefits widely known in the field of ecology and also in a social perspective.

Currently in the whole world, especially in Portugal, due to the climatic changes and other conditions, higher temperatures appears to be increasing for longer periods and only decreasing in rainy periods. As a result of these variables together, Portugal suffered notables and devastating fires, reaching large proportions, some of those, even being declared as wildfires by the authorities.

The options and resources are limited and not always effective. In terms of cost and time to prevent, avoid or to fight this problem. Prevention measures are multiple, such as: social awareness, reordering of the forest, implementation of laws restricting people to make fire during periods of risk, improvement of prevention equipment, training of combat forces (professionalization of fireman), reduction of time reaction between the beginning of the fire, its detection and the fire fight itself, among others.

In this dissertation/essay its presented a conceptual model of a system, which is capable to collect data in real from a WSN (Wireless Sensor Network) that will be stored in a data base that exist in a central server. The data analysis collected will allow to have a view of the fires in real time over a map, and also will allow to generate alert messages through several ways, such as e-mails and SMS (Short Message Services) to parameterized intervenients identified as recipients, recorded in the database of the system

It is taken in care the costs involved, a major condition for viability of this model. The system is based on open source or free software and open hardware.

Keywords

Open Source, Open hardware, Fire Detection System, IoT, Web Services, MESH Networks, GPS, Arduino, Raspberry Pi.

Lista de Abreviaturas e Siglas

FLOSS	Free / Libre and Open Source Software
GNSS	Global Navigation Satellite System
GPS	Global Positioning System.
HTTP	Hypertext Transfer Protocol.
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
IoT	Internet of Things.
OSDL	Open Source Development Lab.
OSI	Open Source Initiative.
RSSF	Rede de Sensores sem Fios
SIRESP	Sistema Integrado de Redes de Emergência e Segurança de Portugal
SMS	Short Message Service
TCP / IP	Transmission Control Protocol / Internet Protocol
VNC	Virtual Network Computing
WSN	Wireless Sensors Network

Índice

1. Introdução.....	12
1.1. Contexto	12
1.2. Questão de Investigação	13
1.3. Objetivos da Investigação.....	13
1.4. Método de investigação	13
2. Revisão da Literatura	15
2.1. Evolução das Redes	15
2.2. O Protocolo HTTP.....	17
2.3. O servidor web	18
2.4. O protocolo TCP / IP.....	18
2.5. Software Open Source e Open Hardware.....	19
2.5.1. Software Open Source.	19
2.5.2. Open Hardware.	21
2.6. Alguns protocolos de Rede Relacionados	23
2.7. GPS – Sistema de Posicionamento Global	25
2.8. Sensores	26
2.8.1. Sensor de Humidade e Temperatura – DHT22.....	26
2.8.2. Sensor Recetor de GPS – GY-NEO-6M.....	26
3. Descrição do Modelo.	28
3.1. Escolha do Hardware.....	28
3.2. Definição de Módulo na rede de sensores sem fios.....	29
3.3. Protocolos de Rede utilizados.....	30
3.4. A Rede Mesh	31
3.5. O Servidor Central.....	32
4. Conclusões	37
5. Bibliografia.....	38
5.1. Artigos impressos	38
5.2. Artigos eletrónicos.....	38

Índice de Figuras

Figura 1 - Diagrama de Etapas do Método de Pesquisa DSR	14
Figura 2 - ARPANet em 1980.	16
Figura 3 - Protocolo HTTP. Interação entre Browser e Servidor Web.	18
Figura 4 - Arduino Modelo Mega 2560 Rev3	22
Figura 5 - Raspberry Pi Modelo 3 B+	22
Figura 6 - Um dos Clusters do supercomputador formado por placas de Raspberry Pi 3.....	23
Figura 7 - Aplicações para o Zigbee.....	24
Figura 8 - Sensor de Humidade e Temperatura DHT22.....	26
Figura 9 - Recetor de sinal GPS modelo GY-NEO6M V2.....	27
Figura 10 - Algumas Topologias de Redes	31
Figura 11 - Fluxo de decisão de envio de Pacotes na rede Mesh	32
Figura 12 - Tabela 01 - Leituras registadas por tratar.	33
Figura 13 - Tabela 02 - Histórico de Leituras	34
Figura 14 - Tabela 03 Intervenientes	35

1. Introdução

Este trabalho destina-se à conclusão do Curso de Gestão de Sistemas e Computação, onde se pretende apresentar o modelo de um sistema de monitorização e recolha de dados para gerar alertas automáticos na deteção rápida de incêndios.

Este modelo está baseado em tecnologias da informação, o que permite comprovar os conhecimentos adquiridos ao longo desta Licenciatura.

1.1. Contexto

No verão de 2017, Portugal foi assolado por incêndios demasiado violentos. Após assistir a tamanha destruição e perda de vidas humanas, ficou evidente que o sistema encarregue de ajudar nas comunicações dos meios de combate a incêndio não foi suficientemente fiável. Este sistema tem um custo muito elevado para o país, como se pode ler no site do Jornal de Notícias, em notícia veiculada do dia 01 julho 2017, visualizada em 04 novembro 2018:

*O contrato celebrado em 2006 entre o Estado e a SIRESP só termina em 2021(...)
Até lá, esta parceria público-privada (PPP) vai custar mais 196 milhões, segundo o último boletim trimestral da Unidade Técnica de Apoio, somando aos cerca de 500 milhões já pagos. (Ramos, 2017)*

Então desafiei-me a aprender mais sobre sistemas de deteção de incêndio, como funcionam e como implementar uma rede de sensores sem fios (WSN – “Wireless Sensord Network”). Com base nos meus conhecimentos prévios de eletrónica e nos adquiridos ao longo desta licenciatura proponho-me descrever um modelo de rede de sensores sem fios que monitorize variáveis indicativas de condições propícias à ocorrência de incêndios (humidade e temperatura) e que a partir da interpretação automática dos valores medidos tenha a capacidade de detetar a possível existência de incêndios e gerar os devidos alertas.

Esta rede de sensores deverá utilizar as últimas placas de “open hardware” existentes, tais como Arduino e Raspberry Pi, aproveitando a sua facilidade de utilização destas com a interligação dos seus módulos periféricos. Uma das características mais importantes destas placas é a flexibilidade de aplicações que podem ser adequadas caso a caso dependendo do objetivo proposto. Outra facilidade é a quantidade de módulos periféricos existentes e disponíveis para compra até mesmo na internet a preços muito acessíveis.

1.2. Questão de Investigação

As novas tecnologias de open hardware e softwares “*open source*” têm sofrido um grande desenvolvimento, diversificação e isso tem causado uma forte redução nos preços associados a projetos criados com a utilização destas tecnologias.

Poderá um sistema de Tecnologias de Informação baseado em open hardware e software “*open source*” ajudar a prevenir/detetar incêndios?

1.3. Objetivos da Investigação

O que se propõe neste trabalho é a criação de um modelo de uma Rede de Sensores Sem Fios (RSSF) baseada em “*open hardware*” e “*software open source*” ou gratuito.

Cada elemento da rede deverá ser capaz de extrair a sua localização (coordenadas GPS), temperatura e humidade e a seguir transmiti-las para um servidor central.

Este guarda-as na base de dados e compara-as com valores de referência de forma a decidir se devem ou não ser enviados alertas.

1.4. Método de investigação

O método de investigação mais apropriado para a realização deste trabalho é o DSR – “*Design Action Research*”, uma vez que é um método caracterizado por produzir resultados de pesquisa relevantes, baseando-se em ações concretas, destinadas a resolver uma situação/problema imediato. O processo de investigação é constituído por uma fase de observação e outra de intervenção.

Existem cinco ciclos no DSR, a referir:

- **Diagnóstico:** Corresponde à identificação dos principais problemas que são as causas subjacentes do desejo da organização para a mudança. É holístico, não envolve redução ou simplificação. Permite desenvolver uma hipótese de trabalho sobre a natureza da organização e a área do problema;

- **Planeamento da Ação:** especifica as ações organizacionais que devem melhorar o problema. A descoberta das ações previstas é guiada pelo quadro teórico, o que indica tanto um estado futuro desejado para a organização, e as mudanças que permitirão atingir tal estado. O plano estabelece a meta e abordagem para mudança;
- **Empreender ações:** implementa a ação planeada. Os investigadores e profissionais colaboram na intervenção ativa na organização cliente, fazendo com que certas alterações sejam implementadas;
- **Avaliação:** após a conclusão das ações, os investigadores e profissionais avaliam os resultados. A avaliação inclui determinar se os efeitos teóricos da ação foram realizados, e se estes efeitos resolveram os problemas. Onde a mudança foi bem-sucedida, a avaliação deve criticamente questionar se a ação empreendida, entre as ações organizacionais rotineiras e não rotineiras, foi a real causa do sucesso. Onde a mudança não foi bem-sucedida, devem ser estabelecidos alguns quadros, para a próxima interação do ciclo de “*action research*”;
- **Aprendizagem:** a aprendizagem é geralmente um processo contínuo.

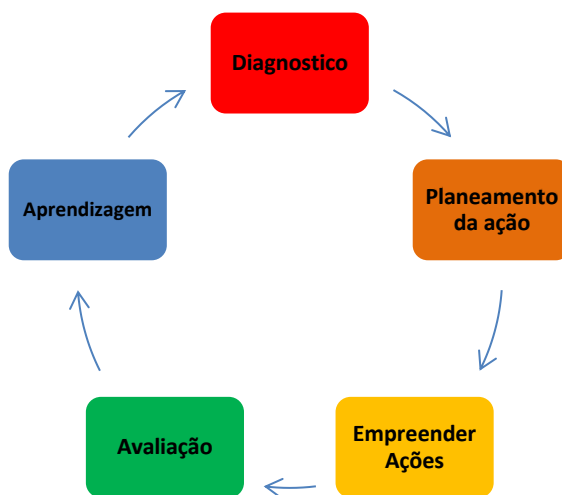


Figura 1 - Diagrama de Etapas do Método de Pesquisa DSR

2. Revisão da Literatura

Neste capítulo é apresentada a informação recolhida e analisada durante o processo de revisão de literatura, onde são apresentados conceitos relacionados com o tema desenvolvido neste trabalho para melhor compreensão das tecnologias utilizadas.

2.1. Evolução das Redes

Década de 1960: O período posterior à Segunda Guerra Mundial foi marcado pelo rápido desenvolvimento da eletrónica aplicada. Nesse cenário, a Tecnologia da Informação foi uma das beneficiadas, uma vez que as equipas de pesquisa e desenvolvimento reunidas para intercetar e decodificar as comunicações/mensagens serviram para desvendar as estratégias dos inimigos.

Com o fim do conflito, as pesquisas continuaram e foram absorvidas pelo meio académico, repassando o conhecimento para os estudantes daquela época. Isso permitiu que houvesse uma continuidade no desenvolvimento e nas pesquisas.

Esta década ficou marcada pelas redes telefónicas com foco na transmissão de voz numa taxa constante entre o transmissor e o recetor. Surgiram os terminais interativos, que permitiam ao utilizador aceder remotamente a computadores da época por meio de uma ligação telefónica. Naquela época, a demanda por um meio de comunicação entre computadores ainda era pequena, pois o número de máquinas existentes era reduzido.

Criação do ARPANet: Durante a Guerra Fria, a agência ARPA (*“Advanced Research Projects Agency”*) surge como uma iniciativa do governo norte-americano para efetuar investigação e desenvolvimento. As pesquisas da ARPA no campo militar incluíram um programa de pesquisas no campo da computação.

A rede ARPANet tornar-se-ia o embrião da Internet e foi desenvolvida pela Arpa em 1969, com o objetivo de interligar bases militares e departamentos de pesquisa sob a gestão do governo americano.

Décadas de 1970 e 1980: Em 1972, a ARPANet contava com cerca de 30 computadores interligados, desde instalações militares a empresas, para compartilhar recursos e troca de dados entre si. Cada nó da rede era interligado a, no mínimo, 2 outros nós para garantir que haveria sempre outro caminho de ligação em caso de falha. Em 1973, os protocolos TCP e IP foram combinados e a ARPANet ganhou escalabilidade e capacidade de interligar computadores de características diferentes. O protocolo TCP/IP tornou-se padrão para a internet.

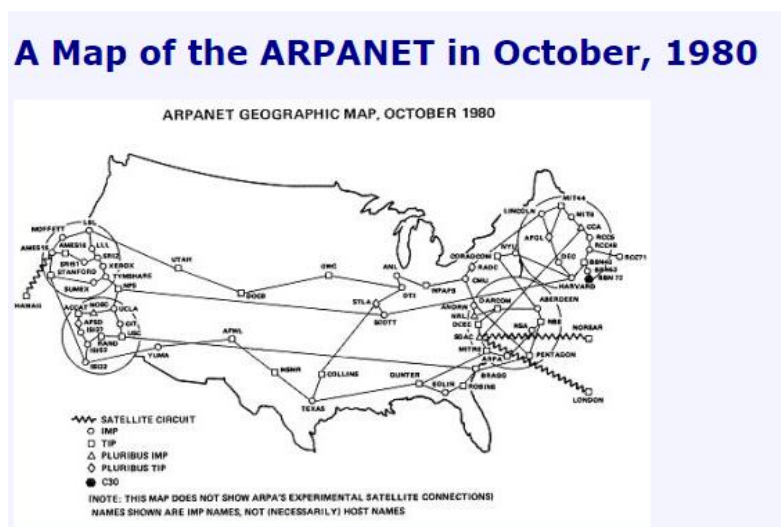


Figura 2 - ARPANet em 1980.

Evolução da Eletrônica: O avanço da microeletrônica permitiu o surgimento de novos computadores no final da década de 1970 e início dos anos 80. Esta nova tecnologia permitiu substituir as válvulas que consumiam muita energia, tornavam os computadores grandes e pesados e limitavam a velocidade de processamento em máquinas menores, com maior capacidade de processamento e menor consumo de energia, além de serem mais fiáveis. Isso permitiu que aparecessem alguns computadores em residências e fez com que o número de computadores interligados fosse superior a duas centenas.

Década de 1990 aos dias de hoje

No início da década de 1990, um pesquisador do Centro Europeu para Física Nuclear (CERN), localizado na fronteira entre França e Suíça, desenvolveu o que conhecemos hoje como web. Tim Berners-Lee seria o criador da WWW (“*World Wide Web*”) que consiste num sistema de hipertexto capaz de exibir conteúdos localizados noutro local da rede. Este sistema seria o protocolo HTTP.

2.2. O Protocolo HTTP

Criado no início da década de 1990, o HTTP é um protocolo extensível que evoluiu ao longo dos anos, sendo um protocolo de camada de aplicação (Modelo OSI) que é enviado em TCP ou em TLS (conexão TCP criptografada). Devido à sua extensibilidade, este protocolo é usado para obter documentos hipertexto, imagens e vídeos ou publicar conteúdo em servidores, como nos resultados de formulários.

O protocolo HTTP define uma série de Métodos de Requisição responsáveis por indicar a ação a ser executada num determinado recurso e, embora esses métodos possam ser descritos como substantivos, são normalmente referenciados como "HTTP Verbs". Cada um deles implementa uma diferente função sendo que alguns recursos são comuns entre todos os verbos, como por exemplo, qualquer método de requisição pode ser do tipo safe, “*idempotent*” ou “*cacheable*”.

Método GET	- Solicita a representação de um recurso específico.
Método HEAD	- É idêntico ao método GET, porém sem conter o recurso "body"(corpo).
Método POST	- Utilizado para submeter uma entidade a um recurso específico, às vezes causando uma mudança no estado do recurso ou solicitando alterações ao servidor.
Método PUT	- Substitui as atuais representações de um recurso pela carga de dados da requisição.
Método DELETE	- Remove um recurso específico.
Método CONNECT	- Estabelece um túnel para conexão com o servidor a partir do recurso alvo;
Método OPTIONS	- Utilizado para descrever as opções de comunicação com o recurso alvo.
Método TRACE	- Executa uma chamada de “ <i>loopback</i> ” de teste durante o caminho de conexão com o recurso alvo.
Método PATCH	- Utilizado para aplicar modificações parciais num recurso.

2.3. O servidor web

Quando se fala de um Servidor web (“web server”), pode-se estar a referir sobre a componente de hardware ou software. Normalmente o conceito abrange as duas partes a trabalhar em conjunto.

Na componente de hardware, o servidor web é um computador com a capacidade de armazenar ficheiros (em diversos formatos: HTML, imagens, CSS, etc) e transmiti-los ao computador quando solicitado.

Para a componente de software, o servidor web refere-se aos programas que gerem o funcionamento dos serviços e definem a forma como os utilizadores acedem à informação pretendida, localizada naquele servidor. Compreende os URLs (endereços “web”) e HTTP (protocolo utilizado para visualizar as chamadas páginas web).

Funcionamento: um navegador de internet (“Browser”) é um programa que fará uma requisição por meio do protocolo HTTP, sempre que for necessário obter um ficheiro armazenado (hospedado) num servidor web. Quando esta requisição for tratada, o servidor responderá ao pedido e enviará o documento requerido, também por meio de HTTP.

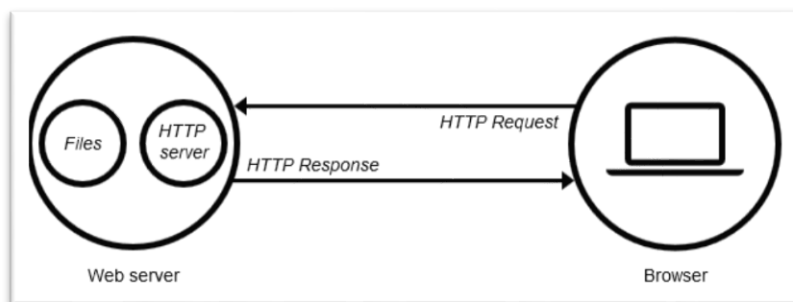


Figura 3 - Protocolo HTTP. Interação entre Browser e Servidor Web.

2.4. O protocolo TCP / IP.

É o principal protocolo de envio e recebimento de dados. Foi desenvolvido pelo Departamento de Defesa dos Estados Unidos (DoD) que tinha urgência em conseguir uma forma de transmitir e receber dados com segurança sob quaisquer circunstâncias. A criação do modelo TCP/IP ajudou a resolver esse problema e tornou-se no padrão para a Internet.

2.5. Software Open Source e Open Hardware.

A evolução das Tecnologias da Informação nos últimos anos foi tão grande que hoje temos à disposição uma grande variedade de hardware e software a preços reduzidos quando comparados com os valores praticados à 10 anos atrás.

Uma vertente desta evolução permitiu o aparecimento de Software Open Source. Estes surgiram com o objetivo de acelerar o desenvolvimento e corrigir falhas permitindo a partilha de conhecimento. O surgimento deste conceito para hardware é mais recente e teve como objetivo produzir computadores a preços reduzidos sem um objetivo específico, ou seja, cada utilizador pode modificar ou adequar às suas necessidades um componente de hardware que satisfaça os princípios daquilo que se pretende construir.

2.5.1. Software Open Source.

Open source é um termo em inglês que significa “código aberto” e isso diz respeito ao código-fonte de programas que poderão ser modificados por qualquer utilizador, de acordo com o que necessitar. O termo foi criado pela OSI (Open Source Initiative) que o utiliza sob um ponto de vista essencialmente técnico.

O software open source não possui um custo de licença, por isso permite um maior investimento em serviços e formação com o objetivo de obter maior retorno dos investimentos em TI. É comum que estas ferramentas sejam compartilhadas online por qualquer pessoa, sem restrições.

O termo “*open source*”, bem como o principal objetivo, foi criado por Eric Raymond entre outros fundadores da OSI com a finalidade de apresentar o software livre a empresas de uma maneira mais comercial, evitando um discurso ético e de direitos.

A nomenclatura “*Open Source*” apareceu durante uma reunião ocorrida em fevereiro de 1998, num debate que envolveu personalidades que posteriormente se tornariam referência neste tema: Todd Anderson, Chris Peterson, Larry Augustin, Jon “Maddog”, Sam Ockman e Eric Raymond, são alguns exemplos.

A sigla FLOSS, que em inglês significa “*Free/Libre and Open Source Software*”, é uma forma agregadora de utilizar os conceitos de Software Livre e Código Aberto a favor dos mesmos softwares, visto que ambos se diferenciam apenas na argumentação.

Os desenvolvedores deste conceito afirmam que não se trata de um movimento anticapitalista e sim de uma alternativa para a produção de software. Esse modelo colaborativo presente no open source levou o direito do autor a ser olhado de forma diferente.

A criação da organização “*Open Source Development Lab*” (OSDL) é um exemplo dos grandes esforços realizados por várias empresas tais como: IBM, Dell, Intel e HP, a trabalhar para a criação de tecnologias de código aberto.

Existem 10 regras importantes a ser respeitados para que um software possa ser considerado “*Open Source*”:

1 - **Distribuição livre:** A licença do programa não deve de forma alguma restringir o acesso gratuito por meio de venda ou mesmo de permutas.

2 - **Código fonte:** O software deve conter um código fonte que deve permitir a sua distribuição também na forma compilada. Caso o programa não seja distribuído com seu código fonte, o desenvolvedor deve fornecer um meio para se obter o mesmo. O código fonte deve ser legível e inteligível para qualquer programador.

3 - **Trabalhos derivados:** A licença do software deve fornecer permissão para que modificações possam ser realizadas, bem como trabalhos derivados. Também deve permitir que os mesmos sejam distribuídos, após modificação, com os termos da licença original.

4- **Integridade do autor do código fonte:** A licença deve, de maneira clara e explícita, permitir a distribuição do programa construído por meio do código fonte modificado. No entanto, a licença pode requerer que programas derivados tenham um nome ou número de versão distintos do programa original. Isso dependerá da preferência do desenvolvedor do código.

5 - **Não discriminação:** A licença deve estar disponível para qualquer pessoa.

6 - **Não discriminação contra áreas de atuação:** A licença deve permitir que qualquer pessoa de qualquer ramo específico possa fazer a utilização do programa. Não deve impedir, por exemplo, que uma empresa faça uso do seu código.

7 - **Distribuição da Licença:** Os direitos associados ao software devem ser aplicáveis para todos aqueles cujo programa é redistribuído, sem que exista a necessidade da execução de uma nova licença.

8 - **Licença não específica a um produto:** O programa não pode fazer parte de outro software, sendo que para utilizá-lo é obrigatório que seja distribuído todo o programa. Se o programa é extraído dessa distribuição, é necessário assegurar que todas as partes sejam disponibilizadas e redistribuídas para todos, visto que todos possuem os mesmos direitos que aqueles que são garantidos em conjunção com a distribuição do programa original.

9 - **Licença não restritiva:** A licença não pode ser considerada “*open source*” se colocar restrições noutros programas que são distribuídos em conjunto com o programa licenciado.

10 - **Licença neutra em relação à tecnologia:** A licença deve permitir que sejam adotadas interfaces, estilos e tecnologias sem restrições. Isso quer dizer que nenhuma cláusula da licença pode estabelecer regras para que estes requisitos sejam aplicados ao programa.

2.5.2. Open Hardware.

Open Hardware (“*hardware*” livre) refere-se às especificações de design de um objeto físico que está autorizado para que qualquer pessoa o possa estudar, modificar, recriar e distribuir. Todas as especificações técnicas de fabricação estão disponíveis para consulta.

Os modelos mais conhecidos e utilizados são o **Arduino** e o **Raspberry Pi**.

Arduino - foi criado em 2005 pelo professor Massimo Banzi na Itália, que queria ensinar aos seus alunos programação e eletrónica, mas não existiam placas no mercado com valores acessíveis para estudantes e, portanto, isso dificultava e limitava a sua aquisição. Para tentar resolver esta situação, ele decidiu criar uma placa de baixo custo que fosse semelhante a um computador. A sua criação recebeu o nome de **Arduíno** e é um sucesso.

Atualmente existem à venda vários modelos de placas semelhantes à original, porém com componentes construídos por diversos fabricantes. Isto deve-se ao fato de que o **Arduíno** original é “*Open Hardware*”, ou seja, qualquer placa criada com a mesma estrutura do **Arduíno**

original e que utilize a sua linguagem padrão consegue realizar as mesmas funções que um Arduino original.

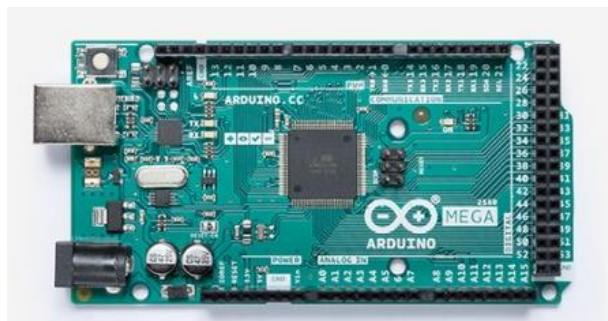


Figura 4 - Arduino Modelo Mega 2560 Rev3

Raspberry Pi - A Fundação Raspberry Pi é uma organização sem fins lucrativos que nasceu com o objetivo de ensinar informática e programação às crianças. Por isso, desenvolveu, em Inglaterra o Raspberry Pi, um microcomputador numa única placa, de baixo custo e pouco maior que um cartão de crédito, mas com a capacidade de processamento que permite viabilizar projetos. O que o torna tão barato é a sua arquitetura: possui portas de comunicação (GPIO, USB, Ethernet, Wi-Fi e Bluetooth), processador, memória, uma saída de vídeo HDMI e uma interface leitora de cartão de memória que será utilizado como disco rígido.

A simplicidade, versatilidade e baixo custo formam em conjunto a fórmula do sucesso do Raspberry Pi.

Os primeiros modelos, Raspberry A e Raspberry B, foram comercializados, respetivamente, por U\$ 25 e U\$ 35. Atualmente, o mais recente modelo, Raspberry Pi Zero WH, tem 6,5 cm de largura por 3 cm de profundidade, vem com Wi-Fi embutido, Bluetooth e pinos GPIO e custa ainda menos do que os modelos anteriores: US\$ 18.



Figura 5 - Raspberry Pi Modelo 3 B+

O Raspberry Pi, criado em 2012, também serviu de base para várias aplicações. Pode-se verificar o caso real da empresa norte-americana BitScope que utilizou 750 placas de Raspberry Pi 3 para criar um supercomputador. As placas foram reunidas em cinco “clusters” e com isso conseguiram um equipamento barato para testes de softwares que posteriormente foram instalados em supercomputadores convencionais.

Este projeto foi desenvolvido em parceria com o Laboratório Nacional de Los Alamos, do Departamento de Energia dos Estados Unidos, e permitiu economizar cerca de 250 milhões de Euros.



Figura 6 - Um dos Clusters do supercomputador formado por placas de Raspberry Pi 3.

Projetos como este comprovam a fiabilidade do hardware e a viabilidade de projetos baseados em Open hardware.

2.6. Alguns protocolos de Rede Relacionados

O **IEEE** (Instituto de Engenheiros Eletricistas e Eletrônicos) é uma associação sem fins lucrativos criada nos Estados Unidos em 1963. Um de seus objetivos é gerar conhecimento nos campos de engenharia elétrica, eletrônica e computação. Ao longo dos anos, tem definido padrões de protocolos para a área de comunicação sem fio visando o crescimento e a interoperabilidade de tecnologias já existentes no mercado ou definindo os ‘standards’ para as novas tecnologias.

Norma **IEEE 802.3** - especifica a camada física e a sub-camada MAC da camada de ligação de dados do Modelo OSI para o **protocolo Ethernet**.

Norma **IEEE 802.11** - também conhecida como **rede Wi-Fi** foi uma das grandes novidades tecnológicas dos últimos anos. Atua na camada física e define uma série de padrões de transmissão e codificação para comunicações sem fio.

Norma **IEEE 802.15.1** – mais conhecida como **Bluetooth**, é o conjunto de especificações para comunicações wireless de curta distância. Estas especificações foram desenvolvidas, inicialmente, para efetuar a transferência de dados entre computadores pessoais para dispositivos periféricos, tais como telemóveis. O Bluetooth usa a topologia estrela que permite que um conjunto de nós possa comunicar com um nó central.

Norma **IEEE 802.15.4** – foi criada com o objetivo de atender a necessidade de comunicação de dispositivos com baixa capacidade de transmissão e baixo consumo de energia, utilizando taxas de transmissão reduzidas. Este protocolo também é conhecido como Zigbee e IoT. A camada MAC suporta as topologias “*Peer-to-peer*” onde cada nó da rede consegue comunicar-se diretamente com outro nó sem a necessidade de passar por um nó central da rede (topologia estrela). Esta camada contém mecanismos de validação e rejeição de mensagens para garantir a correta transmissão de pacotes. A utilização deste padrão permite que os dispositivos possam ficar em estado de suspensão por longos períodos de tempo para reduzir o consumo de energia. Estes dispositivos possuem a capacidade de transmissão de indireta de dados, permitindo desta forma obter alternativas para o reencaminhamento de pacotes conseguindo obter a confirmação de que os pacotes foram entregues no destino.



Figura 7 - Aplicações para o Zigbee.

2.7. GPS – Sistema de Posicionamento Global

O sistema GPS “*Global Positioning System*” é um sistema de navegação global por satélite, GNSS “*Global Navigation Satellite System*”, que permite ao utilizador com um equipamento recetor obter serviços de posicionamento. É um sistema composto por satélites e outros dispositivos que tem como objetivo prestar informações precisas sobre o posicionamento individual no globo terrestre.

O sistema está ativo desde 1995 e foi criado pelo Departamento de Defesa Americano para fins militares, mas também pode ser aproveitado no meio civil, principalmente na aviação. Um conjunto de 24 satélites é o elemento principal que envia as informações para que qualquer dispositivo recetor calcule a sua posição usando um processo chamado de triangulação.

Neste processo, um mínimo de quatro satélites que estejam próximos do recetor estão constantemente a enviar sinais de rádio, contendo a posição atual do satélite e o instante (tempo) em que aquele pulso foi emitido.

O cálculo é feito comparando o tempo em que o sinal foi enviado com o momento em que ele foi recebido. Considerando que a radiofrequência viaja na velocidade da luz e aplicando algumas correções, é possível determinar a distância exata entre o recetor e o satélite.

Cruzando essa informação com a de três outros satélites na área, obtêm-se a posição do recetor. Além da latitude e longitude, o sistema de triangulação também permite saber a altura do recetor em relação ao nível do mar.

O recetor GPS está continuamente a escutar os sinais enviados pelos satélites.

- Um sinal GPS pode-se dividir em 3 partes:
 - Código pseudo-aleatório: identifica o satélite em uso.
 - Dados efémeros: contêm informação sobre o estado atual do satélite, assim como a data e hora, que são essenciais para o cálculo da posição do recetor.
 - Dados almanaque: dão informação sobre a posição de satélite GPS ao longo do dia. Cada satélite transmite a sua informação orbital assim como a dos outros satélites do sistema.

2.8. Sensores

2.8.1. Sensor de Humidade e Temperatura – DHT22

O DHT22 é um sensor de temperatura e humidade que permite medir temperaturas entre 40 graus negativos e 80 graus positivos (Celsius) e humidade entre 0 a 100%, possuindo apenas um pino com saída digital (Pino 2 – DATA).

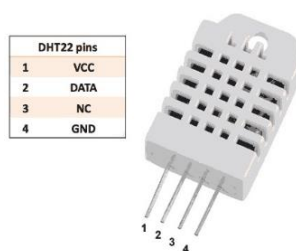


Figura 8 - Sensor de Humidade e Temperatura DHT22

2.8.2. Sensor Recetor de GPS – GY-NEO-6M

O módulo GPS-NEO-6M utiliza comunicação serial com dois pinos (Rx - Receção e Tx - Transmissão) permitindo a comunicação com os mais diversos tipos de equipamentos.

Especificações:

- Módulo GPS GY-NEO6M V2
- Alimentação: 2.7 à 5V DC
- Corrente de operação: 45 mA
- Comunicação serial/TTL
- Antena embutida
- Bateria para “backup” de dados
- Nível de sinal compatível 3.3 e 5V
- Taxa de comunicação padrão: 9600 bps
- Temperatura de operação: -40 a 85°C
- Precisão: 5m
- Dimensões: 35 x 25 x 25mm



Figura 9 - Recetor de sinal GPS modelo GY-NEO6M V2

3. Descrição do Modelo.

3.1. Escolha do Hardware

Para a elaboração deste trabalho foi necessário pesquisar sobre os diversos computadores existentes no mercado de modo a poder perceber as diferentes características técnicas e funcionais de cada modelo de computador existente, para poder escolher o que mais se adequa ao funcionamento proposto.

Depois de comparar alguns computadores fabricados em Open Hardware foram considerados os principais requisitos para a sua escolha:

- Preço e qualidade
- Facilidade de aquisição
- Fiabilidade
- Disponibilidade e compatibilidade de software open source, livre ou gratuito
- IDE de desenvolvimento e linguagem de programação
- Flexibilidade e adaptabilidade
- Documentação técnica disponível
- Periféricos existentes, compatíveis e fiáveis

Para cada unidade da rede de sensores, o **Arduino** foi considerado como a melhor escolha por cumprir os requisitos. Enquanto que para o “*gateway*” e computador central, o **Raspberry Pi** é a melhor escolha por podermos escolher entre versões diferentes de sistemas operativos, principalmente o Raspbian, uma versão do Debian Linux especificamente adaptada para o Raspberry Pi e isso facilita muito a sua programação.

Outro fator importante a considerar é que o sistema operativo do Raspberry Pi, o Raspbian, uma versão do Linux Debian modificada e adaptada especificamente para o Raspberry Pi já vem com os recursos necessários para atualização automática pela internet e permite o acesso remoto por meio do software Real VNC (“*Virtual Network Computing*”). O VNC é um protocolo que permite, ver e controlar outro computador remotamente, através da internet.

3.2. Definição de Módulo na rede de sensores sem fios

Passa a ser denominado de Módulo, cada unidade da rede de sensores sem fios. O Módulo é composto por um computador Arduíno, 2 sensores DHT-22, software para leitura e transmissão da mensagem com os valores lidos, placa de rede Zigbee, antena, fios, bateria e painel solar, montado numa caixa adequada e que possa ser fixada em campo aberto ou presa a um poste.

A rede de sensores é composta pela interligação dos módulos em rede sem fios que ficam permanentemente a recolher informações de coordenadas GPS, Data, hora, temperatura-1, humidade-1, temperatura-2, humidade-2 e voltagem da bateria. A informação de data e hora será extraída do sinal recebido dos satélites GPS.

Como dito anteriormente, cada Arduíno estará equipado com dois sensores DHT22, que fornecem a humidade 1, temperatura 1, lidas do sensor 1 e humidade 2 e temperatura 2, lidas do sensor 2. Por cada leitura, fará a comparação dos valores lidos de cada sensor entre si para garantir que os sensores estão a funcionar adequadamente e que as leituras podem ser consideradas válidas. Isto será verdadeiro quando as leituras forem iguais ou tiverem um desvio entre si de até 5%, ou seja, garantir uma fiabilidade de 95%. Se um desvio maior for identificado, a leitura dos sensores será ignorada e o módulo fará então uma pausa de 3 minutos antes de retornar a realizar novas leituras. Caso sejam verificadas 5 leituras sucessivas incorretas, o sistema transmitirá os valores dos sensores a zero, juntamente com as coordenadas válidas de GPS, data, hora e voltagem da bateria de forma a indicar ao servidor central de que existe uma falha na leitura dos sensores.

A voltagem da bateria é outro ponto a monitorar, mas caberá ao servidor central validar se existe a necessidade de manutenção para o módulo que é identificado pelas referências de coordenadas GPS. A voltagem recomendada para alimentação do Arduíno está entre os 7 e os 20 volts.

Caso o módulo deixe de receber o sinal dos satélites GPS, de onde extrai as coordenadas, data e hora, continuará a enviar as últimas coordenadas válidas lidas anteriormente e passa a transmitir a data e hora preenchidas com zeros. O servidor central interpretará isto como indicação de falha de sinal GPS naquele módulo, que é identificado pelas coordenadas que continuam a ser recebidas.

3.3. Protocolos de Rede utilizados

É necessário ter uma infraestrutura de comunicações fiável para transmitir a informação recolhida por cada módulo e enviá-la ao servidor central, para isso são considerados alguns protocolos de rede, tais como:

Ethernet, que foi prontamente considerado inviável para ser usado pelas limitações de instalação de cabos físicos que levariam a um grande número de falhas. Este protocolo será utilizado para que o “*gateway*” consiga fazer a ligação entre a rede de sensores e o servidor central.

Wi-Fi, este padrão de redes poderia ser viável por fornecer a ligação sem fios, no entanto por operar na faixa de frequência de 2,4GHz fica suscetível a mais interferências aumentando a probabilidade de falhas.

IEEE 802.15.4, este protocolo demonstrou ser o mais adequado para a rede Mesh pois permite interligar um grande número de dispositivos sem fios. Também permite pelas suas características a retransmissão de pacotes entre os módulos de forma que estes cheguem ao destino. O protocolo também prevê um modo de funcionamento com baixo consumo de energia o que prolonga a vida útil da bateria.

A interligação entre a rede Mesh e o servidor central é feita por um Gateway que é constituído por um Raspberry Pi que já vem de série equipado em seu hardware, com uma porta Ethernet.

A rede é constituída por vários “*gateways*”, porém somente um estará ativo de cada vez. Quando for necessário, é possível remotamente ativar outro Gateway. Situação que poderá ocorrer em caso de incendio ou avaria. Ao ativar outro “*gateway*” a rede de sensores volta a ser interligada ao servidor central e o sistema volta a estar operacional. Essa flexibilidade garante maior fiabilidade ao sistema.

3.4. A Rede Mesh

A rede Mesh ficará a funcionar utilizando o protocolo da tecnologia Zigbee, é composta por três tipos de nós: “Coordinator”, “Router” e “End Device”.

O nó “Coordinator” é responsável por iniciar e gerir a rede, após iniciar a rede, o nó coordenador trabalha como “router”.

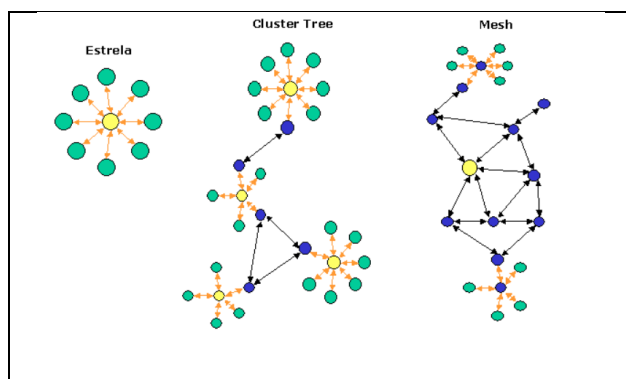


Figura 10 - Algumas Topologias de Redes

Se a rede estiver a operar em modo “beacon-active”, o nó “coordinator” está constantemente a enviar “beacon frames” para sincronizar a rede. Se a rede estiver numa topologia em árvore, todos os nós “router” recebem “beacons” dos seus pais e geram “beacons” para enviar aos nós que pertencem ao seu cluster.

O nó “coordinator” é o responsável por agregar a informação, existindo apenas um por rede, tipicamente o de maior poder computacional, neste caso o Raspberry Pi.

O nó “router” é o responsável pelo encaminhamento dos pacotes entre os nós que não conseguem comunicar diretamente com o coordenador.

O nó “End Device” é o responsável pela função de sensor e apenas se comunica com o nó router ou nó coordenador. Este nó tem a capacidade de entrar em modo “sleep” por forma a reduzir o consumo de energia prolongando a vida útil da bateria.

A tecnologia ZigBee pode ser implementada em qualquer uma das topologias de rede: Árvore, malha ou estrela. Na **topologia em estrela**, a gestão e comunicação da rede é centralizada e os nós “End Device” nunca comunicam entre si, toda a comunicação passa pelo nó “coordinator” que, após inicializar a rede, passa a trabalhar como nó “router”, identificado com um PAN ID

“*Personal Area Network IDentification*” único. Os pontos desfavoráveis nas topologias em estrela são o consumo energético do nó central e a gestão de redes de grande escala.

A **topologia em malha** “*Mesh Network*” também trabalha de forma centralizada, mas pode usar vários nós “**router**” centrais que comunicam entre si e disponibilizam múltiplas rotas entre os nós, dando à rede uma maior flexibilidade e melhor gestão de energia que a rede em estrela.

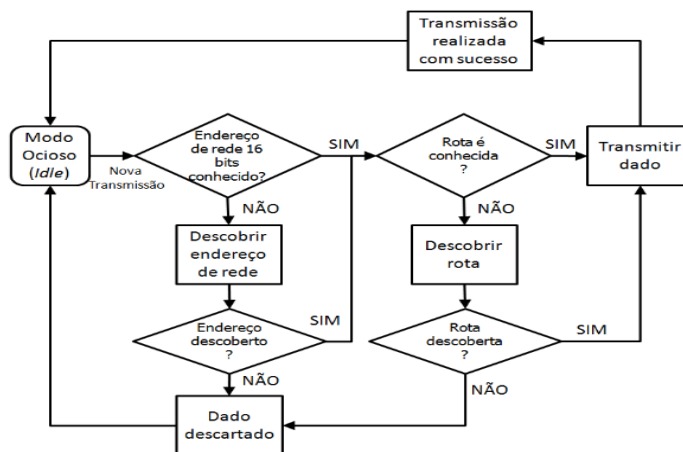


Figura 11 - Fluxo de decisão de envio de Pacotes na rede Mesh

O “**gateway**” é um nó da rede que tem a função de receber os pacotes de informação da rede de módulos e reenvia-los para o servidor central.

3.5. O Servidor Central

No servidor central deverão ser instalados os seguintes softwares:

- APACHE - Este software permite o funcionamento do Servidor Web (web server). Trata as mensagens recebidas e regista os dados recebidos nas bases de dados.
- MariaDB - Servidor de Base de Dados. (MariaDB substituiu recentemente o MySQL.)
- Ciadel - Servidor de email
- Wammu - Envio de SMS.

Os dados recebidos dos diversos módulos são tratados para que sejam extraídas as leituras dos sensores que a seguir são registadas numa tabela com a seguinte estrutura:

Name	Data Type	Unsigned	Auto Increment	Not Null	Default	Collation
☒ Hora	TIME	☐	☐	☒		
☒ Latitude	DECIMAL(12, 6)	☐	☐	☒		
☒ Longitude	DECIMAL(12, 6)	☐	☐	☒		
☐ Temperatura_1	DECIMAL(6, 2)	☐	☐	☒		
☐ Temperatura_2	DECIMAL(6, 2)	☐	☐	☒		
☐ Humidade_1	DECIMAL(3, 0)	☐	☐	☒		
☐ Humidade_2	DECIMAL(3, 0)	☐	☐	☒		
☐ Voltagem_Bateria	DECIMAL(3, 1)	☐	☐	☒		
☐ Status	CHAR(1)	☐	☐	☒	'C'	latin1_swedish_ci
☐ Timestamp_Criacao	TIMESTAMP	☐	☐	☒	CURRENT...	
☐ Timestamp_Modificacao	TIMESTAMP	☐	☐	☒	CURRENT...	

Figura 12 - Tabela 01 - Leituras registadas por tratar.

Esta tabela poderá ser criada com a execução do comando SQL:

```
CREATE TABLE test.leituras_por_tratar (
    Data DATE NOT NULL,
    Hora TIME NOT NULL,
    Latitude DECIMAL(12, 6) NOT NULL,
    Longitude DECIMAL(12, 6) NOT NULL,
    Temperatura_1 DECIMAL(6, 2) NOT NULL,
    Temperatura_2 DECIMAL(6, 2) NOT NULL,
    Humidade_1 DECIMAL(3, 0) NOT NULL,
    Humidade_2 DECIMAL(3, 0) NOT NULL,
    Voltagem_Bateria DECIMAL(3, 1) NOT NULL,
    Status CHAR(1) NOT NULL DEFAULT 'C',
    Timestamp_Criacao TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    Timestamp_Modificacao TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (Data, Hora, Latitude, Longitude)
)
ENGINE = INNODB,
CHARACTER SET latin1,
COLLATE latin1_swedish_ci;

ALTER TABLE test.leituras_por_tratar
    ADD UNIQUE INDEX UK_LEITURAS_POR_TRATAR(Data, Hora, Latitude, Longitude);
```

Existe um processo que trata os dados desta tabela ciclicamente e durante o processamento dever validar os valores dos seguintes dados:

- Valor de temperatura_1 não deve ter um desvio superior a 5% do valor lido em temperatura_2.
- Valor de humidade_1 não deve ter um desvio superior a 5% do valor lido em humidade_2.
- O valor lido de temperatura_1 não pode ser superior a 55 graus pois essa condição quando verificada deve gerar alertas de possibilidade de ocorrência de incêndio.

Depois de tratar os dados, deve atualizar o campo STATUS para 'T' (Tratado), atualizar o campo TIMESTAMP_MODIFICACAO com '*current timestamp*' e mover cada registo tratado para a tabela 02, LEITURAS_HIST com a mesma estrutura da tabela 01.

Name	Data Type	Unsigned	Auto Increment	Not Null	Default	Collation
☒ Data	DATE	☐	☐	☒		
☒ Hora	TIME	☐	☐	☒		
☒ Latitude	DECIMAL(12, 6)	☐	☐	☒		
☒ Longitude	DECIMAL(12, 6)	☐	☐	☒		
☐ Temperatura_1	DECIMAL(6, 2)	☐	☐	☒		
☐ Temperatura_2	DECIMAL(6, 2)	☐	☐	☒		
☐ Humidade_1	DECIMAL(3, 0)	☐	☐	☒		
☐ Humidade_2	DECIMAL(3, 0)	☐	☐	☒		
☐ Voltagem_Bateria	DECIMAL(3, 1)	☐	☐	☒		
☐ Status	CHAR(1)	☐	☐	☒	T	latin1_swedish_ci
☐ Timestamp_Criacao	TIMESTAMP	☐	☐	☒	CURRENT...	
☐ Timestamp_Modificacao	TIMESTAMP	☐	☐	☒	CURRENT...	

Figura 13 - Tabela 02 - Histórico de Leituras

```
CREATE TABLE test.leituras_Hist (
  Data DATE NOT NULL,
  Hora TIME NOT NULL,
  Latitude DECIMAL(12, 6) NOT NULL,
  Longitude DECIMAL(12, 6) NOT NULL,
  Temperatura_1 DECIMAL(6, 2) NOT NULL,
  Temperatura_2 DECIMAL(6, 2) NOT NULL,
  Humidade_1 DECIMAL(3, 0) NOT NULL,
  Humidade_2 DECIMAL(3, 0) NOT NULL,
```

```
Voltagem_Bateria DECIMAL(3, 1) NOT NULL,
Status CHAR(1) NOT NULL DEFAULT 'T',
Timestamp_Criacao TIMESTAMP NOT NULL,
Timestamp_Modificacao TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (Data, Hora, Latitude, Longitude)
)
ENGINE = INNODB,
CHARACTER SET latin1,
COLLATE latin1_swedish_ci;

ALTER TABLE test.leituras_Hist
ADD UNIQUE INDEX UK_LEITURAS_POR_TRATAR(Data, Hora, Latitude, Longitude);
```

Na terceira tabela são registados os dados das pessoas que devem ser avisadas em caso de incêndio ou elevada probabilidade de risco de incêndio. Utilizando o cálculo da distância entre a localização do possível incêndio e as coordenadas GPS registadas na tabela de intervenientes, o sistema determina se deve ou não enviar email e/ou SMS.

Name: Database: Comment:

Engine:

Columns

	Name	Data Type	Unsigned	Auto Increment	Not Null	Default	Collation
☒	Sequencia	INTEGER(11)	☒	☒	☒		
☐	Nome	VARCHAR(255)	☐	☐	☒		
☐	Telefone1	DECIMAL(20, 0)	☐	☐	☒	0	
☐	Telefone2	DECIMAL(20, 0)	☐	☐	☒	0	
☐	Latitude	DECIMAL(12, 6)	☐	☐	☒		
☐	Longitude	DECIMAL(12, 6)	☐	☐	☒		
☐	Email1	VARCHAR(255)	☐	☐	☒		
☐	Email2	VARCHAR(255)	☐	☐	☒		

Figura 14 - Tabela 03 Intervenientes

```
CREATE TABLE test.intervenientes (
  Sequencia INTEGER(11) UNSIGNED NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(255) NOT NULL,
  Telefone1 DECIMAL(20, 0) NOT NULL DEFAULT 0,
  Telefone2 DECIMAL(20, 0) NOT NULL DEFAULT 0,
  Latitude DECIMAL(12, 6) NOT NULL,
  Longitude DECIMAL(12, 6) NOT NULL,
  Email1 VARCHAR(255) NOT NULL DEFAULT ,
  Email2 VARCHAR(255) NOT NULL DEFAULT ,
  PRIMARY KEY (Sequencia)
)
ENGINE = INNODB;
```

Deverá existir outra tabela com o registo dos alertas enviados e servirá para manter um histórico de envio de alertas, tipo de alerta enviado: SMS ou email, data e hora de envio.

4. Conclusões

O modelo aqui proposto com vários computadores “Open hardware” fabricados numa única placa e com capacidade de processamento adequado poderá ter um impacto favorável na prevenção de fogos florestais.

As aplicações de software open source referidos conseguem oferecer fiabilidade na execução das tarefas com robustez e segurança necessários para a sua aplicação neste modelo. O software Apache ainda é o mais utilizado em todo o mundo em web servers juntamente com os bancos de dados MySQL ou MariaDB. (O MariaDB foi desenvolvido a partir do MySQL depois deste último ter sido adquirido pela empresa Oracle e que não deu seguimento a evolução no desenvolvimento do MySQL como era esperado.)

Pretende-se desenvolver uma implementação real do modelo aqui proposto em pequena escala, pois é necessário validar na prática, a distância ideal entre cada módulo da rede de sensores, de forma a reduzir o número de elementos por quilómetro quadrado, garantindo que a rede fique totalmente operacional, estável e sem falhas. A distância entre módulos tem impacto direto na determinação dos custos de implementação.

A implementação deste modelo, a ser utilizado como fonte complementar de informação em tempo real e registo de histórico de temperatura e humidade na área monitorada, aliada à capacidade de gerar alertas de forma automática poderá ser uma solução viável quer em termos operacionais, quer pelos baixos custos envolvidos na montagem e manutenção desta rede.

Este projeto pode ser viável, podendo ajudar a detetar precocemente o risco elevado de incêndio, e desta forma, funcionando como fonte de informação poderia colaborar numa perspetiva estratégica no combate ao incêndio e eventualmente poderá minimizar os cenários de destruição semelhantes aos que foram vividos em anos passados.

5. Bibliografia

5.1. Artigos impressos

Kurose, J. F. & Ross K.W. (2006). **Redes de Computadores e a Internet: Uma Abordagem Top-down**

Soares, F. M. R. (2015). **Localização baseada em medidas de distância para sistemas interiores de larga escala**. Dissertação para obtenção do Grau de Mestre em Engenharia Eletrotécnica e de Computadores. Instituto Superior Técnico, de Lisboa.

5.2. Artigos eletrónicos

Almeida, R. 2003. Sistema Nacional de Detecção de Incêndios Florestais, 10 páginas. Acedido em 11/10/2018. Disponível em <http://scrf.igeo.pt/documentacao/ralmeida/SNDIF.pdf>

Digi, Digi International (2018). XBEE2/XBEEPRO2 Zigbee RF Modules User Guide. 2018 Digi International. All rights reserved. Acedido em 11/10/2018. Disponível em <https://www.digi.com/resources/documentation/digidocs/pdfs/90000976.pdf>

Franco, H. & Moleiro, R. (2018). Fogo no Pinhal de Leiria foi Criminoso – Expresso. Acedido em 11/10/2018. Disponível em <http://expresso.sapo.pt/sociedade/2018-01-20-Fogo-no-Pinhal-de-Leiria-foi-criminoso>

Saleiro, M. & Ey, E. (Data não informada). ZigBee – uma abordagem prática, 52 Páginas. Acedido em 20/10/2018. Universidade do Algarve – Escola Superior de Tecnologia – EEE. Disponibilidade em https://lusorobotica.com/ficheiros/Introducao_ao_Zigbee_-_por_msaleiro.pdf

Os diretores do vídeo foram Eguren, R. & Alaejos, R., encomendado por Laboral Centro de Arte y Creación Industrial (2010). Arduino – O Documentário. Acedido em 05/11/2018. Youtube, publicado pelo canal PY2287SWL. Disponibilidade em https://www.youtube.com/watch?v=SiTmTP_F6so

Martins, S. D. R. (2010). Incêndios Florestais: Comportamento, Segurança e Extinção, 96 páginas. Acedido em 05/11/2018. Faculdade de Ciências e Tecnologia Departamento de Engenharia Mecânica. Disponibilidade em <https://estudogeral.sib.uc.pt/bitstream/10316/14378/1/Inc%C3%AAndios%20florestais,%20comportamento,%20seguran%C3%A7a%20e%20extin%C3%A7%C3%A3o.pdf>

Young, J. (2016). Thread Wireless Technology Overview. Acedido em 05/11/2018. Youtube, publicado pelo Canal Digi International Inc. Disponibilidade em <https://www.youtube.com/watch?v=2-69-DcM2NA>

Ramos, R. (2017). SIRESP vai custar mais 196 milhões de euros, 1 página. Jornal de Notícias. Acedido em 05/11/2018. Disponibilidade em <https://www.jn.pt/nacional/interior/siresp-vai-custar-mais-196-milhoes-de-euros-8606715.html>