



Rui Pedro dos
Santos Oliveira

**Sistema de monitorização e controlo da produção
de salicórnio na Ria de Aveiro**

A decidir...

DOCUMENTO PROVISÓRIO





Rui Pedro dos
Santos Oliveira

**Sistema de monitorização e controlo da produção
de salicórnia na Ria de Aveiro**

A decidir...

DOCUMENTO PROVISÓRIO



**Rui Pedro dos
Santos Oliveira**

**Sistema de monitorização e controlo da produção
de salicórnio na Ria de Aveiro**

A decidir...

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 Doutor Joaquim Manuel Henriques de Sousa Pinto, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor José Alberto Gouveia Fonseca, Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

o júri / the jury

presidente / president

ABC

Professor Catedrático da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)

vogais / examiners committee

DEF

Professor Catedrático da Universidade de Aveiro (orientador)

GHI

Professor associado da Universidade J (co-orientador)

KLM

Professor Catedrático da Universidade N

**agradecimentos /
acknowledgements**

Um agradecimento especial aos meus pais e irmão por todo o apoio prestado ao longo deste meu percurso académico, como também a toda a minha família. A todos os meus colegas e amigos, que conheci ao longo destes 5 anos na Universidade de Aveiro, um grande obrigado

Desejo também pedir desculpa a todos que tiveram de suportar o meu desinteresse pelas tarefas mundanas do dia-a-dia, ...

palavras chave

Cultivo da salicórnia, irrigação, sensores, atuadores, web, monitorização, atuação remota.

resumo

Nos dias que correm, é frequente um trabalho ser avaliado pela sua aparência em vez de o ser pelo seu conteúdo. Sendo assim, sem descurar este último, nesta tese descrevemos maneiras revolucionárias de transformar um documento sólido e austero num documento sólido e belo, capaz de fazer chorar de alegria (ou de inveja) qualquer leitor, mesmo quando este não percebe nada do que lá está escrito.

A exploração de novas descobertas na área da percepção visual, nomeadamente no que se refere à apreciação de obras de arte geniais, ...

keywords Cultivo da salicórnia, irrigação, sensores, atuadores, web, monitorização, atuação remota.

abstract Nowadays, it is usual to evaluate a work ...

Conteúdo

| | |
|---|-----|
| Lista de Figuras | vii |
| Lista de Tabelas | ix |
| Acrónimos | xi |
| 1 Introdução | 1 |
| 1.1 Motivação | 1 |
| 1.2 Objetivos | 2 |
| 1.3 Organização do documento | 2 |
| 2 Salicórnia: caracterização, importância e cultivo | 5 |
| 2.1 Características da planta | 5 |
| 2.2 Importância da planta | 7 |
| 2.2.1 Aplicações alimentares | 7 |
| 2.2.2 Aplicações medicinais | 7 |
| 2.3 Condições ideais de cultivo da salicórnia | 7 |
| 2.4 Considerações finais | 7 |
| 3 Soluções para Internet of Things | 9 |
| 3.1 Evolução tecnologia: o IoT | 9 |
| 3.1.1 Vantagens | 11 |
| 3.1.2 Desvantagens | 11 |
| 3.2 Arquitetura em Internet of Things (IoT) | 11 |
| 3.3 Aplicações relacionadas | 11 |
| 3.3.1 Multi-monitorização de estufas agrícolas | 12 |
| 3.3.2 Agroopar | 12 |
| 3.4 Sistema de Monitorização de Estufas Agrícolas | 12 |

| | |
|--|-----------|
| 4 Estado da arte | 13 |
| 4.1 Sistema de gestão de base de dados (Sistema de Gestão de Base de Dados (SGBD)) | 13 |
| 4.1.1 PostgreSQL | 13 |
| 4.1.2 MySQL | 13 |
| 4.1.3 SQL server | 13 |
| 4.1.4 Comparação e solução adotada | 13 |
| 4.2 Desenvolvimento web | 15 |
| 4.2.1 Django | 15 |
| 4.2.2 Faslk | 15 |
| 4.2.3 ASP.net | 15 |
| 4.2.4 Conclusões e solução adotada | 15 |
| 4.3 Desenvolvimento mobile | 16 |
| 4.3.1 Plataformas nativas | 16 |
| 4.3.2 Multi-plataforma | 16 |
| 4.3.3 Conclusões e solução adotada | 16 |
| 4.4 REST Frameworks | 17 |
| 4.4.1 Django Rest Framework | 17 |
| 4.4.2 Flask-RESTful | 17 |
| 4.4.3 Conclusões e solução adotada | 17 |
| 4.5 Documentação automática | 17 |
| 4.5.1 Documentação API | 17 |
| 4.6 Sensores | 18 |
| 4.6.1 Sensor de temperatura | 18 |
| 4.6.2 Sensor de luminosidade (GL5528) | 19 |
| 4.7 Tecnologias de comunicação usadas em IoT | 19 |
| 4.7.1 RFID/NFC | 19 |
| 4.7.2 Bluetooth | 19 |
| 4.7.3 WiFi | 19 |
| 4.7.4 Zigbee | 20 |
| 4.7.5 LoRa | 20 |
| 4.7.6 Sigfox | 20 |
| 4.7.7 GPRS/GSM | 20 |
| 4.7.8 Comparação de tecnologias de comunicação | 21 |
| 4.7.9 Módulo bluetooth | 21 |

| | |
|---|-----------|
| 5 Sistema de controlo e monitorização | 23 |
| 5.1 Descrição global do sistema | 23 |
| 5.2 Componentes | 24 |
| 5.2.1 Controller Module | 25 |
| 5.2.2 Sensor Module | 26 |
| 5.3 Design funcional | 27 |
| 5.4 Considerações finais | 27 |
| 6 Sistema de informação: análise de requisitos e arquitetura | 29 |
| 6.1 Análise de requisitos | 29 |
| 6.1.1 Requisitos funcionais | 29 |
| Dashboard | 29 |
| Aplicação mobile | 30 |
| 6.1.2 Requisitos não funcionais | 30 |
| 6.1.3 Entidades envolventes | 30 |
| 6.1.4 Casos de utilização | 31 |
| 6.2 Estrutura da base de dados | 32 |
| 6.3 Design técnico | 34 |
| 6.4 Arquitetura lógica | 35 |
| 6.4.1 Camada de apresentação | 35 |
| 6.4.2 Camada de lógica de negócio | 35 |
| 6.4.3 Camada de acesso a dados | 35 |
| 6.5 Arquitetura física | 35 |
| 6.5.1 Sistema de informação | 35 |
| 6.5.2 Aplicação web | 35 |
| 6.5.3 Aplicação mobile | 35 |
| 6.6 Diagrama de componentes | 35 |
| 6.7 Sistema de interação | 35 |
| 6.8 Descrição | 35 |
| 6.9 Arquitetura geral | 36 |
| 6.10 Requisitos de funcionamento | 37 |
| 6.11 API | 38 |
| 6.12 Valores simulados | 38 |
| 6.13 Deploy do projecto | 38 |
| 6.14 Considerações finais | 38 |

| | |
|---|-----------|
| 7 Simulação em hardware | 39 |
| 7.1 Micro-controladores | 39 |
| 7.1.1 Arduino Nano | 40 |
| 7.1.2 Raspberry pi | 42 |
| 7.2 Sensores | 43 |
| 7.2.1 Temperatura | 43 |
| 7.2.2 Luminosidade | 44 |
| 7.2.3 Sensor de nível líquido | 45 |
| 7.2.4 Simulador de bomba para transferências de águas (led) | 45 |
| 7.3 Comunicação | 46 |
| 7.4 Interligação de componentes | 47 |
| 7.5 Considerações finais | 47 |
| 8 Sistema de deteção de intrusos | 49 |
| 8.1 Biblioteca de processamento de imagem: OpenCV | 49 |
| 8.1.1 Conclusões | 50 |
| 8.2 Algoritmo de deteção de intrusos | 51 |
| 8.3 Testes | 52 |
| 8.4 Implementação | 52 |
| 8.4.1 Flask | 52 |
| 8.4.2 Servidor web NGNIX | 52 |
| 8.5 Considerações finais | 52 |
| 9 Resultados | 53 |
| 9.1 Testar API | 53 |
| 9.2 Aplicação mobile | 53 |
| 9.3 Deteção de intrusos | 53 |
| 10 Conclusão e trabalho futuro | 55 |
| 10.1 Conclusão | 55 |
| 10.2 Trabalho futuro | 55 |
| 10.3 Considerações finais | 55 |
| A Untitled appendix #A | 59 |
| B Mockups da aplicação mobile | 61 |
| C Trigger SQL | 63 |

| | |
|---|-----------|
| D Resultados processamento de imagem | 65 |
| D.1 Frame 1 | 66 |
| D.2 Frame 2 | 68 |
| D.3 Frame 3 | 70 |
| D.4 Frame 4 | 72 |
| E Interface gráfica | 75 |
| F Descrição formal dos casos de uso gerais | 77 |
| G Interligação de componentes em hardware | 79 |

Lista de Figuras

| | | |
|------|---|----|
| 1.1 | Salicornia proveniente da ria de Aveiro | 1 |
| 2.1 | <i>Salicornia ramosissima</i> : na primavera e no outono respetivamente à esquerda e à direita (Fotografia por José M. G. Pereira) | 6 |
| 2.2 | Ciclo de vida da <i>Salicornia ramosissima</i> (Fotografia por José M. G. Pereira) | 6 |
| 3.1 | Evolução da internet em cinco fases (Adaptado de [1]) | 10 |
| 3.2 | Pirâmide do conhecimento: modelo DIKW | 11 |
| 5.1 | Ilustração principais componentes | 23 |
| 5.2 | Ilustração de uma ”quinta”onde se produz salicornia | 24 |
| 5.3 | Pirâmide do conhecimento: modelo DIKW | 25 |
| 6.1 | Pirâmide do conhecimento: modelo DIKW | 31 |
| 6.2 | Pirâmide do conhecimento: modelo DIKW | 32 |
| 6.3 | Arquitetura lógica | 35 |
| 7.1 | Flower one. | 40 |
| 7.2 | Flower two. | 40 |
| 7.3 | Flower one. | 42 |
| 7.4 | Flower two. | 42 |
| 7.5 | Flower one. | 43 |
| 7.6 | Flower two. | 43 |
| 7.7 | Flower one. | 44 |
| 7.8 | Esquema eletrotécnico | 44 |
| 7.9 | Flower one. | 45 |
| 7.10 | Flower two. | 45 |
| 7.11 | Flower one. | 45 |
| 7.12 | Flower two. | 45 |

| | |
|--|----|
| 7.13 Arquitetura lógica | 46 |
| 7.14 Flower one. | 47 |
| 7.15 Flower two. | 47 |
| 8.1 Logótipo OpenCV | 50 |
| 8.2 Raspberry Pi Camera Board V2 8MP 1080p | 50 |
| D.1 Pirâmide do conhecimento: modelo DIKW | 66 |
| D.2 Pirâmide do conhecimento: modelo DIKW | 68 |
| D.3 Pirâmide do conhecimento: modelo DIKW | 70 |
| D.4 Pirâmide do conhecimento: modelo DIKW | 72 |
| E.1 Pirâmide do conhecimento: modelo DIKW | 75 |
| E.2 Pirâmide do conhecimento: modelo DIKW | 76 |
| G.1 Pirâmide do conhecimento: modelo DIKW | 80 |

Lista de Tabelas

| | | |
|-----|---|----|
| 6.1 | My caption | 33 |
| 6.2 | My caption | 38 |
| 7.1 | Características do sensor TTC 104 | 41 |
| 7.2 | Comparação entre versão 2 e 3 do Raspberry Pi | 42 |
| 7.3 | Características do sensor TTC 104 | 43 |
| 7.4 | Características do sensor GL5528 | 44 |
| 9.1 | Um nome qualquer | 53 |
| B.1 | Um nome qualquer | 61 |
| D.1 | Your caption here | 67 |
| D.2 | Your caption here | 69 |
| D.3 | Your caption here | 71 |
| D.4 | Your caption here | 73 |
| F.1 | Casos de utilização: Login/Logout | 77 |
| F.2 | Casos de utilização: Login/Logout | 77 |

Acrónimos

| | |
|-------------|---|
| NTC | Negative Temperature Coefficient |
| HTML | HyperText Markup Language |
| HTTP | HyperText Transfer Protocol |
| UID | Unique Identification Number |
| GSM | Global System for Mobile Communications |
| IoT | Internet of Things |
| RFID | Radio-Frequency IDentification |
| GPRS | General Packet Radio Service |
| SM | Sensor Module |
| CM | Controller Module |
| SGBD | Sistema de Gestão de Base de Dados |
| URL | Uniform Resource Locator |
| WWW | World Wide Web |

Introdução



Figura 1.1: Salicornia proveniente da ria de Aveiro

1.1 Motivação

<http://eusougourmet.blogspot.pt/2011/09/compre-o-que-e-nosso-salicornia.html>

* O gênero salicornia inclui cerca de 117 espécies, sendo *Salicornia herbacea*, *Salicornia bigelovii*, *Salicornia europea*, *Salicornia prostata*, *Salicornia mmosissima* e *Salicornia virginica* aquelas com maior ocorrência. [2]

A que serve de mote a esta dissertação ...

Os recursos naturais, nomeadamente, plantas, animais e minerais, são utilizados desde a antiguidade pelo ser humano, não apenas como fonte de alimentos mas também para o tratamento de diversas doenças []. Muitas das espécies que nascem em todo o mundo inicialmente são consideradas pragas, contudo e após alguns estudos intensivos à espécie são descobertas verdadeiras pérolas. Um exemplo disso é a salicornia.

A salicornia é a planta que iremos dar destaque durante este projeto. Esta planta é por vezes utilizada como substituta do sal marinho[] e utilizada para os mais diversos fins. Iremos abordar alguns deles mais à frente.

A salicornia nasce e cresce naturalmente ao longo dos estuários e sapais (salinas) costeiras do Mediterrâneo].

Esta é uma planta suculenta adaptada a ambientes salinos (halófita) que se desenvolve maioritariamente em ambientes aquáticos com elevado teor de sal.]

Existem mais de de as mais comuns são:

Existem cerca de uma centena de espécies do género *Salicornia* L.], as mais comum encontram-se destacadas de seguida:

Salicornia virginica: é uma planta com flor e pode ser encontrada na região mediterrânica

Salicornia europea: resce em várias zonas de entre-marés salinas

Salicornia maritima: *Salicornia bigelovii*: *Salicornia perennis*: *Salicornia ramosissima*:

A evolução tecnológica é algo que sempre esteve presente na vida do ser humano desde os seus primórdios até aos dias atuais, sendo que se tem verificado um aumento desta relação com o humano e principalmente com o ritmo da própria evolução. As tecnologias, de uma maneira geral, são todas as invenções produzidas pelo homem, para aumentar a sua atividade no planeta e simplificar o modo de vida que quem o habita [1]. O conceito de “Internet das coisas” (do inglês “Internet of Things”, IoT) é fruto desta evolução tecnológica, já que permite a ligação dos mais diversos dispositivos eletrónicos à Internet.

1.2 Objetivos

Este trabalho tem como objetivo o desenvolvimento

- Criação de uma plataforma web que permita:
 - Disponibilizar a leitura dos mais diversos sensores de sensores (temperatura, salinidade...)
 - Permitir gerar alarmes de inundação, sendo estes enviados via SMS ou email para o cliente.
 - Atuar remotamente para drenagem de água em excesso existente nas leiras
 - Sistema de transmissão de vídeo disparada por eventos gerados pelos sensores
- Criação de uma aplicação móvel que permita receber alarmismos de situações anómalas.

1.3 Organização do documento

No Capítulo 2 apresenta-se

o projeto CAMBADA e identifica-se os pontos chave tanto do software como do hardware.

No Capítulo 3

No Capítulo 4 é....

Para finalizar, no Capítulo 5 apresentam-se conclusões sobre o trabalho desenvolvido e eventuais melhorias para o futuro.

Salicórnia: caracterização, importância e cultivo

A palavra salicórnia deriva do latim tardio *sal*, que significa sal, e *cornus* que significa corno. Etimologicamente a palavra salicórnia significa cornos salgados[3]. A espécie de salicórnia que servirá de mote à elaboração desta dissertação é a única existente em Portugal designada por *Salicornia ramosissima J. Woods (S. ramosissima)*[4], uma espécie do género *Salicornia L.*, pertencente à família das beterrabas denominada de *Chenopodiaceae* [5].

Nesta secção será apresentada a *Salicornia ramosissima* que impulsionará toda esta dissertação. Serão descritas as principais características desta planta, principais propriedades e as diferentes aplicações alimentais existentes no mercado.

2.1 Características da planta

A salicórnia é uma espécie halófita, ou seja adaptada a viver em ambientes com elevado teor salino[6], sendo uma das mais evoluídas da sua família. É uma planta anual de dimensão pequena, aparentemente sem folhas, ereta, os seus caules são carnudos e suculentos, simples e/ou extremamente ramificados, segmentados por articulações[7], geralmente com menos de 30 cm de altura[2].

A salicórnia tem uma coloração normalmente verde-escuro mas a sua ramagem torna-se verde-amarelado ou mesmo vermelho-púrpura no outono[7]. A figura 2.1 ilustra a respetiva coloração na primavera e no outono. Na Inglaterra, a salicórnia é conhecida como *purple glasswort*, podendo este nome estar na origem desta pigmentação caraterística[8]. Em Portugal e Espanha é conhecida vulgarmente como erva-salada, sal verde e/ou espargos do mar[9].



Figura 2.1: *Salicornia ramosissima* : na primavera e no outono respetivamente à esquerda e à direita (Fotografia por José M. G. Pereira)

A *Salicornia ramosissima* desenvolve-se preferencialmente no litoral costeiro, em pântanos e sapais salgados ou em margens de salinas temporariamente alagadas. Encontra-se distribuída maioritariamente na parte oeste da Europa e a oeste da região do Mediterrâneo, sendo uma das espécies mais abundante[10]. Pode ser encontrada em todo o litoral da Península Ibérica, embora com menos frequência no Minho[7]. Em Portugal, é encontrada ao longo da costa, mais frequentemente nas margens dos canais da Ria de Aveiro e Ria Formosa, no Algarve[9].

Esta planta é uma das mais estudadas a nível mundial[10], possuindo um ciclo de vida anual bem definido, com gerações discretas e as suas sementes são hermafroditas[11]. A salicórnia cresce habitualmente entre março, início da sementeira e novembro fechando assim o ciclo com a produção de sementes. Entre maio e agosto decorre a colheita da planta[9] utilizada para os mais diversos fins. A floração ocorre fundamentalmente no mês de outubro[10]. A figura 2.2 representa evolução do estado da planta para as diferentes fases do seu ciclo de vida.



Figura 2.2: Ciclo de vida da *Salicornia ramosissima* (Fotografia por José M. G. Pereira)

2.2 Importância da planta

Uma das características que tornam o género *Salicornia L.* uma planta tão popular são as suas elevadas propriedades nutricionais, nomeadamente a nível de minerais e vitaminas antioxidantes, como vitamina C e β -caroteno. A salicornia é também uma fonte de proteínas e possui um alto teor total de lípidos e ómega-3[ref].

Desde a descoberta da salicórnia que esta é usada a nível culinário mas também no tratamento e prevenção de algumas doenças. Seguidamente iremos aprofundar cada uma dessas aplicações esclarecendo a sua relevância.

2.2.1 Aplicações alimentares

Espécies do género *Salicornia L.* estão incluídas na alimentação humana, desde a antiguidade, sendo normalmente consumida crua, cozinhada ou seca, podendo ser triturada. Quando crua é usada como acompanhamento das mais diversas refeições enquanto que seca ou triturada é usada como especiaria, podendo ser utilizada como tempero na confecção de peixes, marisco ou carnes. O sal verde é um grande substituto do sal comum, pois é rico em substâncias depurativas e diuréticas. Os seus caules carnudos são bastante requisitados para cozinhas *gourmet*, não só pelo seu sabor salgado, mas também pelo seu elevado valor nutricional. [reff]

2.2.2 Aplicações medicinais

A nível medicinal, existem inúmeros estudos que revelam as propriedades químicas que esta planta detém. Existem estudos que demonstram estas propriedades na prevenção e tratamento de algumas doenças, tais como, a hipertensão, cefaleias e escorbuto, diabetes, obesidade, cancro, entre outras.

2.3 Condições ideais de cultivo da salicórnia

O crescimento da *Salicornia ramosissima* é influenciada pela salinidade do meio. Um estudo realizado por Silva et al.[11] comprova que esta planta halófita apresenta um crescimento ideal a salinidades baixas ou moderadas, em vez de salinidades elevadas, pelo que é considerada uma halófita não obrigatória.

2.4 Considerações finais

Soluções para Internet of Things

Nesta secção encontra-se descrita uma pequena introdução ao conceito de *Internet of Things* e respetiva importância no contexto deste projeto. São também apresentadas as principais tecnologias de comunicação possível de utilização e respetiva comparação entre elas. Por fim, serão apresentados alguns projetos/aplicações relacionadas com esta dissertação.

3.1 Evolução tecnologia: o IoT

Antes de descrever a importância e o conceito de IoT, é necessário entender as diferenças entre os termos Internet e World Wide Web (WWW), que são usados indistintamente pela sociedade. A Internet é a camada ou rede física composta por *switches*, *routers* e outros equipamentos[12]. A sua principal função é transportar informações de um ponto para outro de forma rápida, confiável e segura. Por outro lado, a Web pertence à camada de aplicações que opera sobre a Internet cuja função é oferecer uma interface que transforme as informações que fluem pela Internet em algo útil. Ao longo do tempo, a Web passou e continua a passar por várias etapas evolucionárias, identificadas como:

- Web 1.0 – passado: esta primeira etapa foi inventada por Tim Berners Lee em 1989[13]. Nesta fase surgiram os principais conceitos que conhecemos da Internet atual: Localizador Uniforme de Recursos (do inglês Uniform Resource Locator (URL)), Linguagem de Marcação de Hipertexto (do inglês HyperText Markup Language (HTML)) e Protocolo de Transferência de Hipertexto (do inglês HyperText Transfer Protocol (HTTP)). Ainda nesta primeira fase, mas mais tarde, em 1998 foi criado por Larry Page e Sergey Brin o Google que criou simplicidade nas pesquisas na Web[14].

- Web 2.0 – presente: a Web cresceu muito e muito rapidamente. A versão mais próxima da visão de Tim Berners Lee – colaborativa, usado como meio de interação, comunicação global e elevado compartilhamento de informação.
- Web 3.0 – futuro: para o futuro prevê-se que os conteúdos online possão vir a estar organizados de forma semântica, muito mais personalizados para cada utilizador, sites, aplicações inteligentes e/ou publicidade baseada nas pesquisas e nos comportamentos.

O aparecimento do IoT foi extraordinariamente importante já que se trata da primeira evolução real da Internet, um salto que levará, no futuro, ao desenvolvimento de aplicações revolucionárias com potencial para melhorar significativamente a forma como a sociedade vive, aprende, trabalha e se diverte. O IoT já transformou a Internet em algo sensorial, através da medição de diferentes características, como por exemplo a temperatura, a pressão, as vibrações, a iluminação, a humidade, o stress, entre outras.

A figura 3.1 representa a evolução da Internet em cinco fases. Inicialmente surge a conexão entre dois computadores que permite a criação de uma rede, posteriormente nasce o conceito de WWW ligando um grande número de computadores entre si. Seguidamente, surgiu a Internet móvel que permitiu conectar dispositivos moveis à Internet, possibilitando a ligação da sociedade através das redes sociais. Finalmente, a internet está a evoluir para o IoT, permitindo ligar objetos do quotidiano ao sistema global de redes de computadores [1].

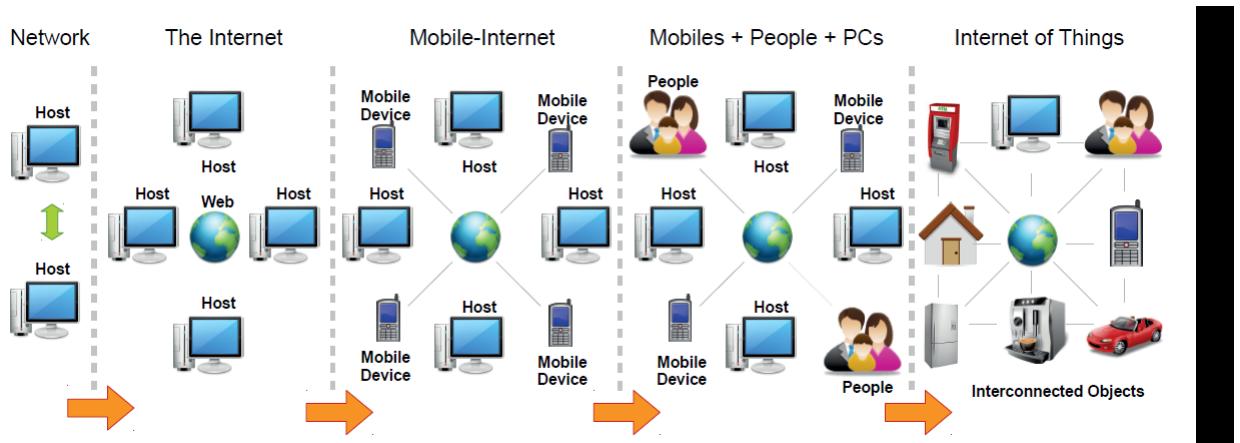


Figura 3.1: Evolução da internet em cinco fases (Adaptado de [1])

Uma das principais vantagens do IoT é a sua ligação evidente a todos os objetos, o que por si só é uma ideia avassaladora. O volume de dados gerado por este tipo de ligação pode ser interpretado pelo modelo DIKW que em inglês significa Data-Information-Knowledge-Wisdom [15]. Este modelo, também conhecido como pirâmide do conhecimento (Figura G.1), é uma hierarquia informacional utilizada especialmente nas áreas da ciência da informação e

na gestão do conhecimento, onde cada camada acrescenta certos atributos sobre a anterior.

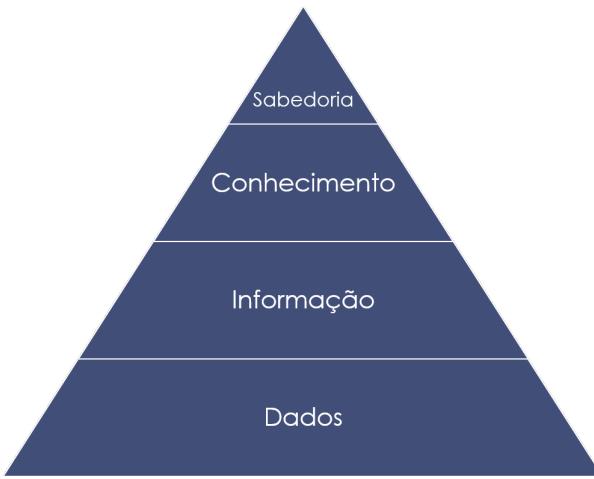


Figura 3.2: Pirâmide do conhecimento: modelo DIKW

A ligação dos objetos à Internet acarreta benefícios visíveis à nossa sociedade, possibilitando um maior controlo e entendimento de como os sistemas interagem entre si e proporcionando uma melhor qualidade de vida a todos. Embora as vantagens se sobreponham às desvantagens não nos podemos esquecer que existem alguns problemas a nível segurança, privacidade, legislação e identidade.

3.1.1 Vantagens

3.1.2 Desvantagens

3.2 Arquitetura em IoT

IoT

3.3 Aplicações relacionadas

Seja para comparar, seja para replicar boas funcionalidades, ou seja para conseguir oferecer algo mais ao utilizador final, quando se pretende desenvolver uma determinada aplicação, é importante proceder a uma avaliação de aplicações da mesma área se encontram no mercado. Assim, são aqui abordadas algumas das aplicações relacionadas que são mais utilizadas ou que mais se aproximam daquilo que se pretende para a aplicação a desenvolver neste projeto, tendo em conta os diferentes sistemas operativos.

3.3.1 Multi-monitorização de estufas agrícolas

<https://repositorio.ipcb.pt/bitstream/10400.11/949/1/Multimonitorizacao.pdf>

3.3.2 Agroopar

<http://www.vidarural.pt/agroopar-os-custos-na-mao-do-agricultor/>

3.4 Sistema de Monitorização de Estufas Agrícolas

[16]

4

Estado da arte

Nesta secção, são apresentados os resultados da pesquisa efectuada sobre o estado da arte das ferramentas com funcionalidades que deverão estar presentes no sistema desenvolvido. Pretende-se apresentar de forma geral todas as tecnologias possíveis de utilização e respetiva comparação.

4.1 Sistema de gestão de base de dados (SGBD)

Um SGBD é um conjunto de software responsáveis pela gestão de uma base de dados.

4.1.1 PostgreSQL

O PostgreSQL é um sistema de gestão de base de dados do tipo objeto-relacional uma vez que permite um modelo de dados orientado a objetos i.e. possibilita a manipulação de objetos, classes e heranças diretamente no esquemas da base de dados. Segundo o site oficial do PostgreSQL este é considerado um SGBD bastante poderoso e com desenvolvimento *open sources*. []

4.1.2 MySQL

4.1.3 SQL server

4.1.4 Comparação e solução adotada

Os próprios criadores do Django recomendam a utilização do PostgreSQL, indicando que alcança um bom equilíbrio entre custo, características, rapidez e estabilidade

No entanto, é pertinente fazer uma comparação entre o PostgreSQL e outras ferramentas open-source como o MySQL. Embora as diferenças entre as duas ferramentas não sejam muito grandes, podemos ter também em conta a performance de uma e outra. Uma comparação feita usando o benchmark TPC-H 8 mostra que a performance do PostgreSQL é ligeiramente superior à do MySQL na maioria das queries [22].

4.2 Desenvolvimento web

Para o desenvolvimento da dashboard poderiam ser adotadas duas estratégias distintas para o desenvolvimento web:

- Manipulação local usando javascript do DOM.
- Acesso ao servidor que serve conteúdos criados em função dos pedidos do cliente

Neste contexto poderiam ser utilizados

Angular, React

Servidor serve conteudos criados em função dos pedidos do cliente

4.2.1 Django

Assim, e de acordo com as explicações dos autores da ferramenta [18], as principais vantagens tiradas da utilização da framework Django são: Boa documentação; Facilidade e rapidez de desenvolvimento e deployment; Estabilidade; Escalabilidade.

4.2.2 Faslk

4.2.3 ASP.net

4.2.4 Conclusões e solução adotada

4.3 Desenvolvimento mobile

4.3.1 Plataformas nativas

4.3.2 Multi-plataforma

<http://websocialdev.com/lista-de-frameworks-para-desenvolvimento-mobile/>

4.3.3 Conclusões e solução adotada

4.4 REST Frameworks

4.4.1 Django Rest Framework

Django REST Framework é uma ferramenta considerada 'poderosa e flexível para a construção de APIs Web' [], que pode ser usada juntamente com a framework de desenvolvimento de aplicações Web Django, que quando integrada no desenvolvimento de um determinado *backend* permite a implementação de serviços do tipo REST.

A API navegável Web é uma vitória usabilidade enorme para os desenvolvedores.

Políticas de autenticação , incluindo pacotes para OAuth1a e OAuth2 .

Serialização que suporta tanto ORM e não ORM fontes de dados.

Customizável todo o caminho - basta usar vistas regulares baseadas na função , se você não precisar dos mais poderosos recursos .

Extensa documentação , e grande apoio da comunidade .

Utilizado e confiável por empresas internacionalmente reconhecidas, incluindo Mozilla , Red Hat , Heroku , e Eventbrite .

4.4.2 Flask-RESTful

4.4.3 Conclusões e solução adotada

com autenticação via token

4.5 Documentação automática

4.5.1 Documentação API

utilizado swagger; apenas permite acesso a quem está logado... incorporar layout do swagger com o do salidashboard

app mobile microcontroladores -i controller modulers

documentação com swager

4.6 Sensores

Esta secção tem como objetivo fazer um estudo comparativo entre diferentes tecnologias usadas para a medição dos vários parâmetros ambientais necessários ao controlo e monitorização da salicornia. Todas as soluções adaptadas tem termos de hardware escolhidas devido à possuir-las.

4.6.1 Sensor de temperatura

Existem vários tipos de sensores de temperatura baseados em princípios de funcionamento distintos.

- **Termopares:**
- **RTDs:**
- **Termístор:**
- **Circuito integrado:**

4.6.2 Sensor de luminosidade (GL5528)

O LDR (Light Dependent Resistor) é um componente cuja resistência varia de acordo com a intensidade da luz. Quanto mais luz incidir sobre o componente, menor a resistência. Este sensor de luminosidade pode ser utilizado em projetos com arduino e outros microcontroladores para alarmes, automação residencial, sensores de presença e etc.

4.7 Tecnologias de comunicação usadas em IoT

Nesta secção serão apresentados alguns das tecnologias de comunicação mais utilizados em *Internet of Things* que permite a troca de informações entre dispositivos e respetiva comparação entre eles.

4.7.1 RFID/NFC

A identificação por radiofrequência, conhecida por tecnologia Radio-Frequency IDentification (RFID), é um método de identificação automático através de sinais de rádio. Consiste essencialmente no armazenamento e posterior envio de informação por meio de ondas electro-magnéticas para circuitos integrados compatíveis em rádio frequência. Os actuais sistemas de RFID possuem grande capacidade de identificação e localização de bens ou pessoas levou, o que fez com que esta tecnologia começasse assumisse um papel importante na indústria e no comércio. A comunicação é efetuada entre uma etiqueta, ou marca, e um leitor.

De forma conceptual, o leitor RFID é responsável por transmitir um sinal de rádio através da antena para a tag, e esta responde emitindo para o leitor RFID o seu Unique Identification Number (UID).

4.7.2 Bluetooth

Bluetooth é uma especificação de rede sem fio de âmbito pessoal (Wireless personal area networks – PANs) consideradas do tipo PAN ou mesmo WPAN

4.7.3 WiFi

rede sem fio IEEE 802.11, que também são conhecidas como redes Wi-Fi ou wireless, foram uma das grandes novidades tecnológicas dos últimos anos. Atuando na camada física, o 802.11 define uma série de padrões de transmissão e codificação para comunicações sem fio, sendo os mais comuns: FHSS (Frequency Hopping Spread Spectrum), DSSS (Direct Sequence Spread Spectrum) e OFDM (Orthogonal Frequency Division Multiplexing). Atualmente, é o

padrão de fato em conectividade sem fio para redes locais. Como prova desse sucesso pode-se citar o crescente número de Hot Spots e o fato de a maioria dos computadores portáteis novos já saírem de fábrica equipados com interfaces IEEE 802.25. A Rede IEEE possui como principal característica transmitir sinal sem fio através de ondas!

4.7.4 Zigbee

Zigbee designa um conjunto de especificações para a comunicação sem-fio entre dispositivos eletrônicos, com ênfase na baixa potência de operação, na baixa taxa de transmissão de dados e no baixo custo de implementação. Tal conjunto de especificações define camadas do modelo OSI subsequentes àquelas estabelecidas pelo padrão IEEE 802.15.4.

4.7.5 LoRa

A tecnologia Lora

Wide-Area Network Low-Power (LPWAN) ou Low-Power Rede (LPN) é um tipo de telecomunicações sem fio de rede projetada para permitir comunicações de longo alcance em uma baixa taxa de bits entre as coisas (objetos relacionados), tais como sensores operados em uma bateria.

As tecnologias WAN de baixa potência são projetadas para ambientes de rede máquina a máquina (M2M). Com a diminuição dos requisitos de energia, maior alcance e menor custo do que uma rede móvel, os LPWANs são pensados para permitir uma gama muito mais ampla de aplicativos M2M e Internet of Things (IoT), que foram limitados por orçamentos e problemas de energia.

4.7.6 Sigfox

Uma empresa francesa que constrói redes sem fio para conectar objetos de baixa energia, como medidores de energia elétrica , smartwatches e máquinas de lavar, que precisam estar continuamente ligados e emitindo pequenas quantidades de dados. Sua tecnologia é voltada para a Internet das Coisas (IoT).

4.7.7 GPRS/GSM

O General Packet Radio Service (GPRS) é uma tecnologia que aumenta as taxas de transferência de dados nas redes Global System for Mobile Communications (GSM) existentes.

Vantagens em relação ao GSM

4.7.8 Comparação de tecnologias de comunicação

4.7.9 Módulo bluetooth

Sistema de controlo e monitorização

Este capítulo tem como objetivo a descrição do sistema que resultou do trabalho prático desta dissertação. Para esse fim, cada elemento pertencente ao sistema é caracterizado de acordo com as suas funções e especificidades. É também descrito como os elementos interagem entre si.

5.1 Descrição global do sistema

Este sistema tem como objetivo a supervisão remota da produção de salicornia, permitindo não só a monitorização dos dados adquiridos pelos sensores, como também da atuação remota de determinados comandos. Neste contexto também será possível a aquisição de imagens que possibilitará a deteção de intrusos nas quintas onde se produz esta espécie.

O esquema da figura 5.1 ilustra todos os componentes de um modo geral e as diferentes plataformas com que o cliente pode interagir.

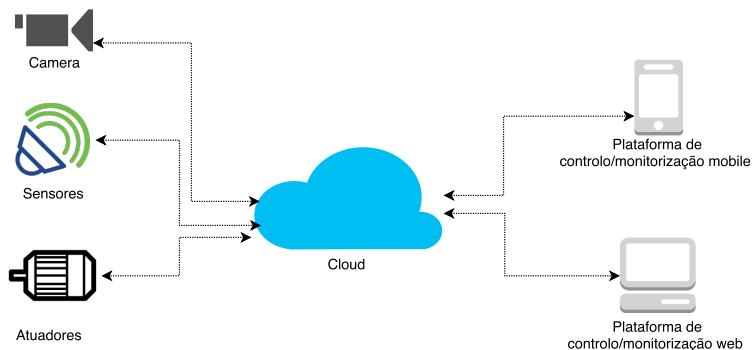


Figura 5.1: Ilustração principais componentes

Como vimos no capítulo 3, uma plantação de salicórnia carece de um controlo relativamente fino de certos parâmetros ambientais sobretudo da salinidade do terreno onde ela cresce. A salinidade do terreno depende, por sua vez, das chuvas, da salinidade da água dos canais da ria. Nas quintas onde se cultiva salicórnia, a produção faz-se numa espécie de leiras limitadas por pequenos canais de irrigação. Esses canais podem ser cheios de água salgada proveniente dos esteiros que rodeiam a quinta. Essa operação implica a abertura de válvulas de admissão dessa água, medida do nível da maré nos canais, monitorização da qualidade e salinidade da água exterior.

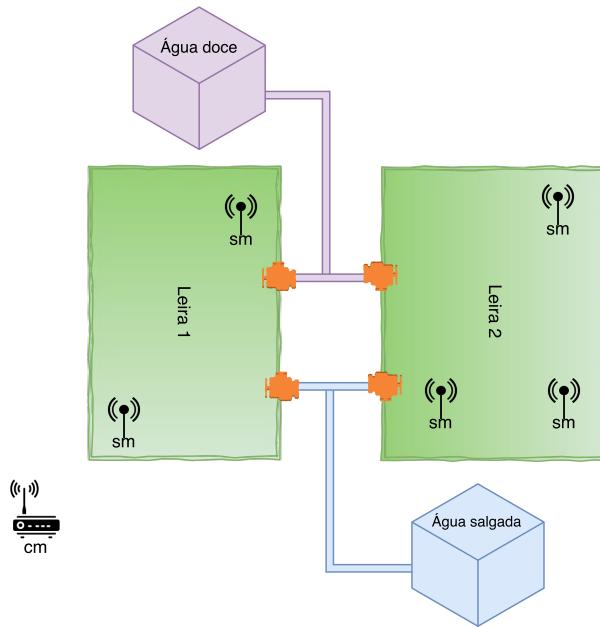


Figura 5.2: Ilustração de uma ”quinta” onde se produz salicornia

A figura 5.2 ilustra esquematicamente a localização de duas leiras com determinados canais de irrigação para água doce e água salgada.

Neste contexto, cada grupo de sensores espalhado por cada leira irá comunicar com um módulo central, originando uma topologia de rede em estrela. Por sua vez, este módulo irá comunicar diretamente com a servidor que possibilitará que os dados sejam tratados e disponíveis para visualização ao cliente. Pressupõe-se portanto que este ultimo módulo tenha necessariamente ligação à rede de modo a conseguir consumir a API REST por HTTP desenvolvida para o efeito.

5.2 Componentes

No contexto desta dissertação é necessário reter dois conceitos principais, são eles:

- **Sensor Module (SM):** consiste num microcontrolador responsável pela aquisição de dados provenientes dos mais diversos tipos de sensores. Cada *sensor module* terá que utilizar um determinado módulo de comunicação de modo a possibilitar a comunicação com um módulo central. Para além disso, pretende-se que o sensor module possa ter controlo sobre si, ou seja, inteligência própria.
- **Controller Module (CM):** consiste num microcontrolador responsável pela receção dos dados proveniente dos *sensor modules*. Pretende-se que este módulo envie informações para os sensor modules quando requisitados pelo utilizador. O principal objetivo deste módulo consiste em receber a informação proveniente dos sensor modules e respectivo envio para o servidor em cloud.

A figura 5.3, ilustra a comunicação entre três sensor modules com um controller moduler. Cada um desses sensor modules possui um conjunto específico de sensores que comunicam com controller module através de um determinado modulo de comunicação. Posteriormente, o controller module possui um determinado protocolo de comunicação que permite a utilização da API REST.

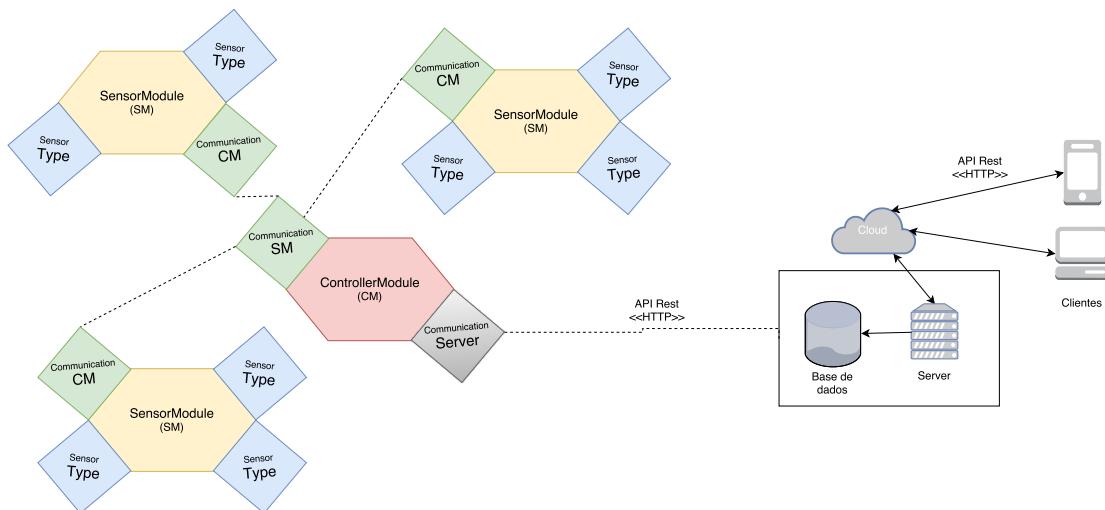


Figura 5.3: Pirâmide do conhecimento: modelo DIKW

Seguidamente serão especificados todos os detalhes de cada modulo, uma vez que serão considerados na modulação deste sistema.

5.2.1 Controller Module

nome bateria memoria comunicacao localizacao

5.2.2 Sensor Module

nome tipos de senores tempo em que é recebida info bateria localizacao

5.3 Design funcional

5.4 Considerações finais

Sistema de informação: análise de requisitos e arquitetura

6.1 Análise de requisitos

6.1.1 Requisitos funcionais

Dashboard

- A interface do sistema deve permitir que o utilizador, seja ele qual for, entre ou faça *login* no sistema.
- A interface do sistema deve permitir que o utilizador, seja ele qual for, saia ou faça (*logout*) no sistema.
- O dashboard deverá permitir que qualquer utilizador possa recuperar a sua chave de acesso ao sistema.
- O sistema deve permitir que qualquer utilizador se possa registar embora sempre associado a uma determinada empresa.
- O dashboard deverá permitir ao administrador a adição de novas empresas e a gestão de todos os utilizadores.
- O sistema deve permitir que qualquer utilizador possa adicionar, editar ou remover:
 - Tipos de sensores
 - Tipos de comunicação

- Controller modules
 - Tipo de comunicação a um controller moduel
 - Sensor modules a um determinado Controller modules
 - Um ou vários tipos de comunicação de um Sensor moduel
 - Um ou vários sensores a um sensor moduel em que cada sensor é de um determinado tipo
- O sistema deve permitir o armazenamento de informações do cliente.
 - O sistema deve permitir a atualização das informações do cliente.

Aplicação mobile

6.1.2 Requisitos não funcionais

Requisitos não funcionais são todos os requisitos da aplicação relacionados com performance, escalabilidade, segurança, disponibilidade e usabilidade. Estes não são necessariamente pedidos pelo cliente.

- O sistema deverá executar em qualquer plataforma.
- O sistema deverá disponibilizar uma API para que possam ser criados novos produtos com base neste
-

6.1.3 Entidades envolventes

No contexto deste sistema existem três entidades distintas que são importantes numerar:

- **General user:** este poderá registar-se e associando-se a uma determinada empresa registada no sistema. Após a validação por parte da empresa este utilizador poderá aceder à sua área reservada através das dashboard ou aplicação mobile.
- **Company user:** utilizador que gera todos os general users que se possam associar a si. Deste modo, este utilizador poderá validar os general users que a si se associam ou eliminá-los caso a permissão não seja desejada.
- **Administrador:** vulgarmente denominado por Admin. Pretende-se que apenas exista uma único administrador. Genericamente este utilizador tem a possibilidade de poder adicionar novas empresas ao sistema i.e. criar novas utilizador com permissões específicas.

6.1.4 Casos de utilização

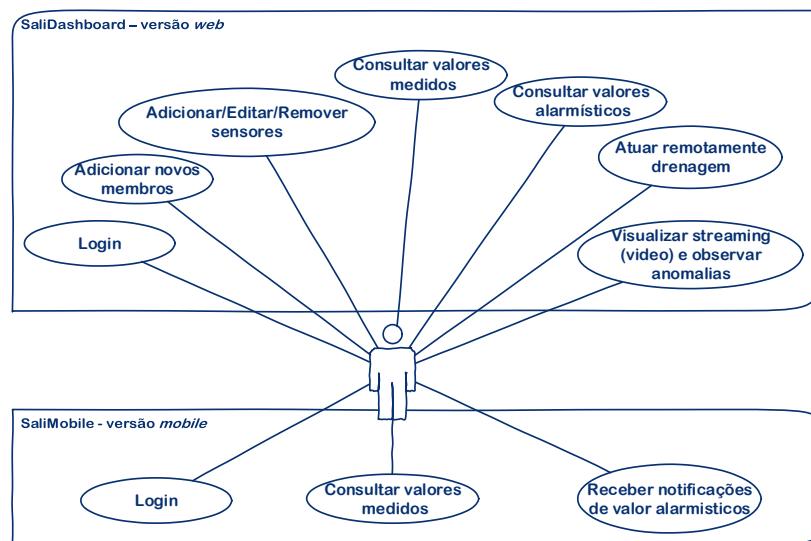


Figura 6.1: Pirâmide do conhecimento: modelo DIKW

6.2 Estrutura da base de dados

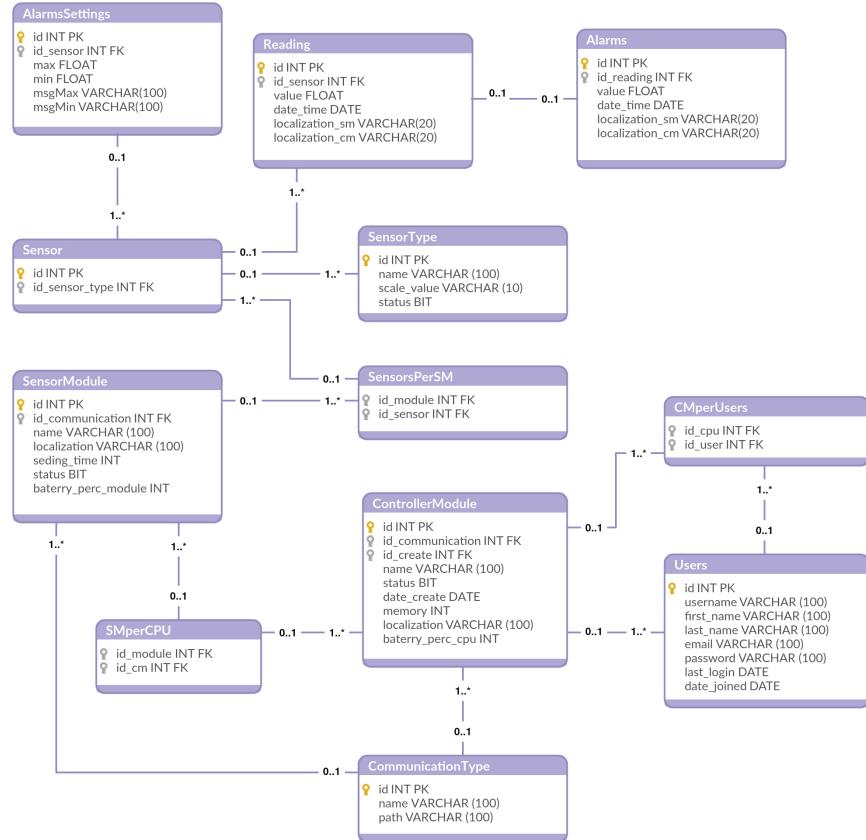


Figura 6.2: Pirâmide do conhecimento: modelo DIKW

| Nome | Identificador | Descrição |
|-------------------|-------------------|-----------|
| User | auto-incrementado | dasdas |
| ControllerModule | | |
| SensorModule | | |
| CommunicationType | | |
| SensorType | | |
| Sensor | | |
| Reading | | |
| AlarmsSettings | | |
| Alarms | | |

Tabela 6.1: My caption

6.3 Design técnico

6.4 Arquitetura lógica

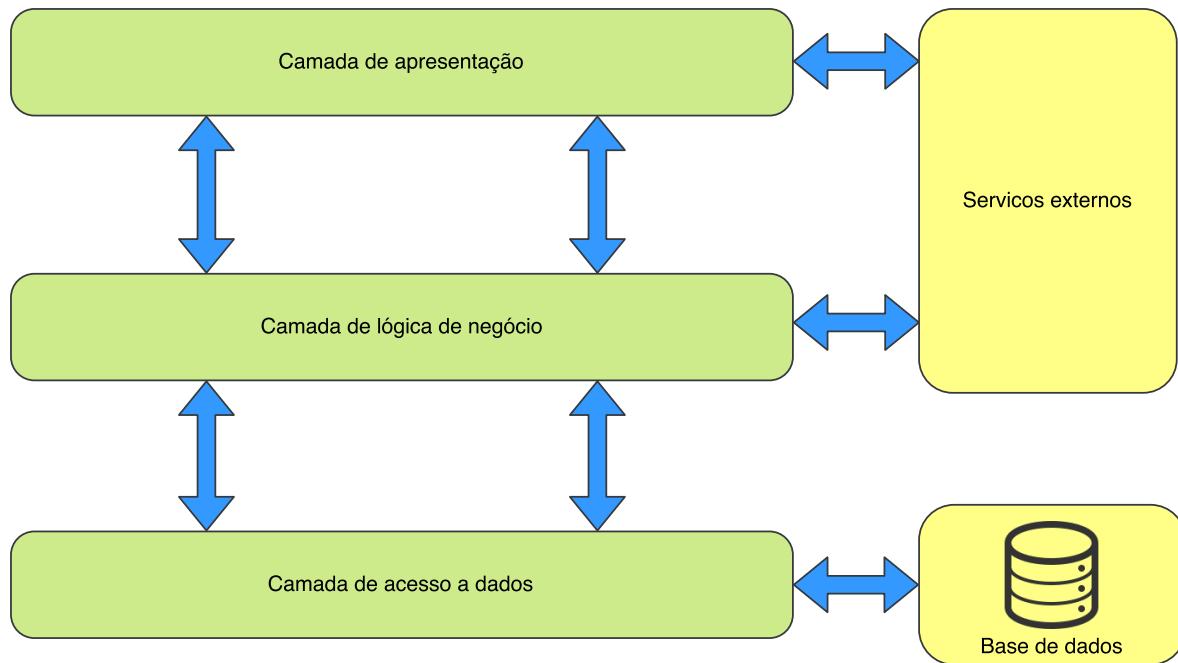


Figura 6.3: Arquitetura lógica

6.4.1 Camada de apresentação

6.4.2 Camada de lógica de negócio

6.4.3 Camada de acesso a dados

6.5 Arquitetura física

6.5.1 Sistema de informação

6.5.2 Aplicação web

6.5.3 Aplicação mobile

6.6 Diagrama de componentes

6.7 Sistema de interação

6.8 Descrição

6.9 Arquitetura geral

6.10 Requisitos de funcionamento

6.11 API

Os métodos da API permitem executar as funções REST. Assim, torna-se fundamental perceber estes métodos para ter um melhor conhecimento do funcionamento do sistema. Como tal, de seguida, são descritos os métodos mais importantes que dão suporte a cada uma das funções REST.

- **GET:**
- **POST:**
- **DELETE:**

| End point | Tipo | Descrição |
|-------------------|-------------------|-----------|
| User | auto-incrementado | dasdas |
| ControllerModule | | |
| SensorModule | | |
| CommunicationType | | |
| SensorType | | |
| Sensor | | |
| Reading | | |
| AlarmsSettings | | |
| Alarms | | |

Tabela 6.2: My caption

6.12 Valores simulados

6.13 *Deploy* do projecto

<https://jee-appy.blogspot.com.tr/2015/04/deploy-django-project-on-apache-using.html>

Características da máquina virtual

Description: Ubuntu 14.04.1 LTS 64 bits RAM 2GB

6.14 Considerações finais

Simulação em hardware

Com o objetivo de simular a API desenvolvida pretendeu-se encontrar hardware que encaixasse no contexto do problema. Para tal, foram utilizados dois micro-controladores e alguns sensores. Neste capítulo será descrito cada um deles e o processo de desenvolvimento da respetiva simulação.

7.1 Micro-controladores

Para do cenário anteriormente descrito foram utilizados dois micro-controladores bastante comuns no mercado. Um arduino e um raspberry pi.

7.1.1 Arduino Nano

O Arduino é fruta da evolução de um projeto italiano desenvolvido no ano de 2005, cujo o objetivo consistiu em ser utilizado em projetos escolares de forma a ter um orçamento menor que outros sistemas de prototipagem disponíveis naquela época.

Tal como descrito no seu site oficial, um Arduino consiste numa plataforma *open-source* de prototipagem eletrónica com hardware e software flexíveis e fáceis de serem utilizados.

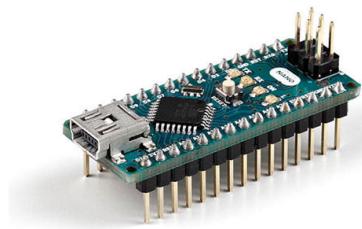


Figura 7.1: Flower one.

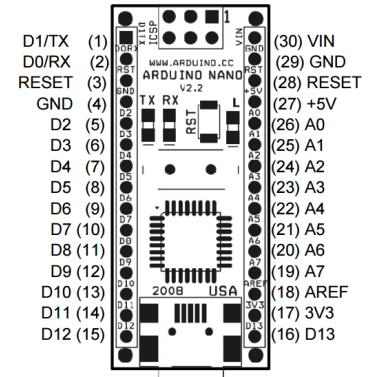


Figura 7.2: Flower two.

Principais características do Arduino Nano:

- Microcontrolador: É o cérebro do Arduino. Este é o dispositivo programável que roda o código que enviamos à placa. Nesta placa o microcontrolador ATmega328 é utilizado, este dispõem de 32kb de memória flash e 2kb de memória ram.
- Conector USB: Conector que conecta o Arduino ao computador além de alimentar a placa.
- Pinos de Entrada e Saída: Pinos que podem ser programados para agirem como entradas ou saídas fazendo com que o Arduino interaja com o meio externo.
- Pinos de Alimentação: Fornecem diversos valores de tensão que podem ser utilizados para transmitir energia elétrica aos componentes do seu projeto.
- Botão de Reset: Botão que reinicia o dispositivo.
- Conversor Serial-USB e LEDs TX/RX: O conversor Serial-USB permite que o microcontrolador e o computador se comuniquem, nesta placa o microcontrolador Atmega16U2 é programado para agir como conversor. Os LEDs TX e Rx acendem quando o Arduino está transmitindo e recebendo dados pela porta serial respectivamente.

- Conector de Alimentação: Permite com que uma fonte alimente a placa. Caso o Arduino esteja sendo alimentado pela porta USB e por uma fonte o hardware seletor escolherá automaticamente a melhor fonte.
- LED de Alimentação: Indica se a placa está a transmitir energia.
- LED Interno: LED ligado ao pino digital 13.

Na tabela 7.1 encontram-se as principais características do Arduino utilizado.

| | |
|---------------------|----------------------------------|
| Microcontrolador | ATmega328 |
| Tensão de Operação | 5V |
| Tensão de Entrada | 7-12V |
| Portas Digitais | 14 (6 podem ser usadas como PWM) |
| Portas Analógicas | 8 |
| Corrente Pinos I/O | 40mA |
| Memória Flash | 32KB (2KB usado no bootloader) |
| SRAM | 2KB |
| EEPROM | 1KB |
| Velocidade do Clock | 16MHz |
| Dimensões | 45 x 18mm |

Tabela 7.1: Características do sensor TTC 104

O que pode ser feito? Evolução...

Componentes IDE

<https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>

7.1.2 Raspberry pi



Figura 7.3: Flower one.

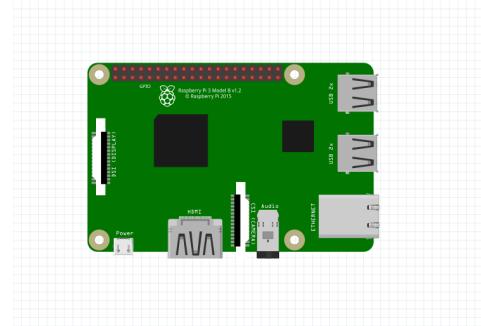


Figura 7.4: Flower two.

| | Raspberry Pi 3 Model B | Raspberry Pi 2 Model B 1.2 |
|-------------------------------|--|--|
| Processor Chipset | Broadcom BCM2837 64Bit Quad Core Processor powered Single Board Computer running at 1.2GHz | Broadcom BCM2837 64Bit Quad Core Processor powered Single Board Computer running at 900MHz |
| Processor Speed | QUAD Core @1.2 GHz | QUAD Core @900 MHz |
| RAM | 1GB SDRAM @ 400 MHz | 1GB SDRAM @ 400 MHz |
| Storage | MicroSD | MicroSD |
| USB 2.0 | 4x USB Ports | 4x USB Ports |
| Max Power Draw/voltage | 2.5A @ 5V | 1.8A @ 5V |
| GPIO | 40 pin | 40 pin |
| Ethernet Port | Yes | Yes |
| WiFi | Built in | No |
| Bluetooth LE | Built in | No |

Tabela 7.2: Comparaçao entre versão 2 e 3 do Raspberry Pi

7.2 Sensores

Nesta secção serão apresentados os sensores utilizados na simulação e as suas principais características. Todos os sensores foram escolhidos tendo em conta o seu enquadramento no projeto e a sua disponibilidade no laboratório.

7.2.1 Temperatura

Como sensor de temperatura foi utilizado um termíster do tipo Negative Temperature Coefficient (NTC). Como vimo anteriormente, um termíster é um semicondutor sensível à temperatura i.e. cujo o coeficiente de variação da resistência com a temperatura é negativa, ou seja, quando a temperatura sobe então consequentemente a resistência diminui.

Na figura x encontra-se o esquema eletrónico deste componente.



Figura 7.5: Flower one.

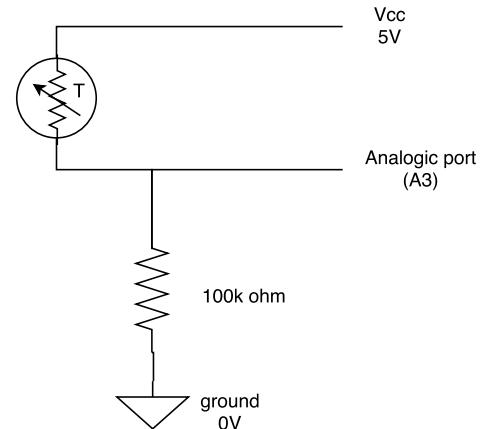


Figura 7.6: Flower two.

| | |
|--------------------|----------------|
| Dimensão | 5mm |
| Resistência | 100K Ω |
| Valor máximo | +125C |
| Valor mínimo | -30C |
| Nível de confiança | + - 10% |
| Preço | 0.35 €/unidade |

Tabela 7.3: Características do sensor TTC 104

7.2.2 Luminosidade

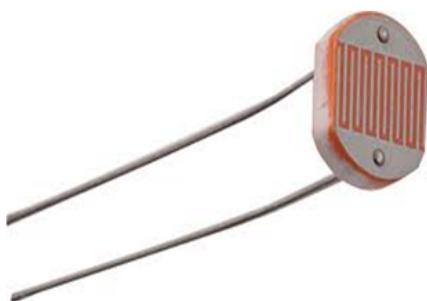


Figura 7.7: Flower one.

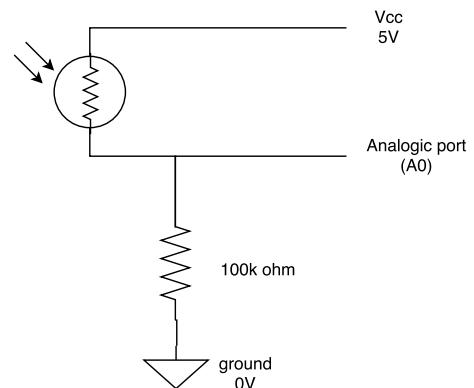


Figura 7.8: Esquema eletrotécnico

| | |
|----------------------------|-----------------------|
| Diâmetro | 5mm |
| Tensão máxima | 150VDC |
| Potência máxima: | 100mW |
| Tensão de operação: | -30 C a 70 C |
| Espectro: | 540nm |
| Comprimento com terminais: | 32mm |
| Resistência no escuro: | 1 M (Lux 0) |
| Resistência na luz: | 10-20 Komega (Lux 10) |

Tabela 7.4: Características do sensor GL5528

7.2.3 Sensor de nível líquido

Water Level Switch Liquid Level Sensor Plastic Ball Float



Figura 7.9: Flower one.



Figura 7.10: Flower two.

7.2.4 Simulador de bomba para transferências de águas (led)

dsadsa



Figura 7.11: Flower one.

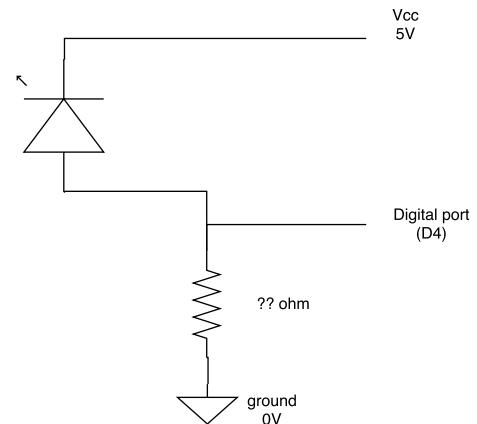


Figura 7.12: Flower two.

7.3 Comunicação

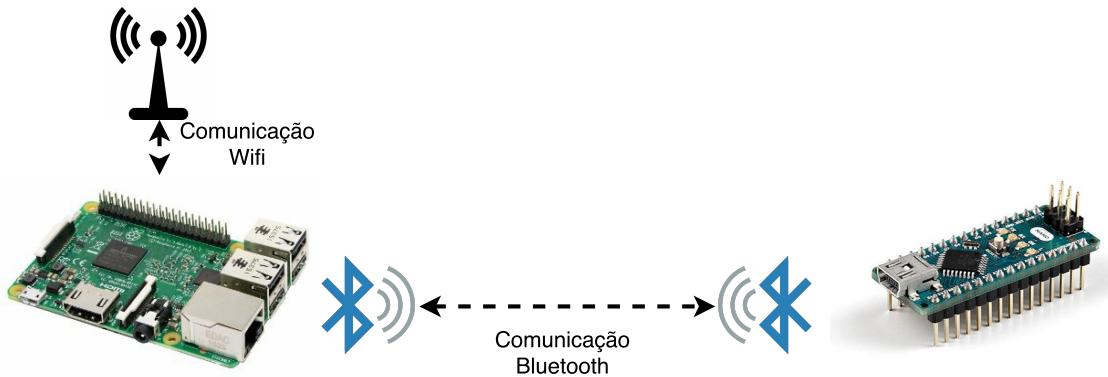


Figura 7.13: Arquitetura lógica

Numa primeira fase precedeu-se à comunicação através de porta serie.

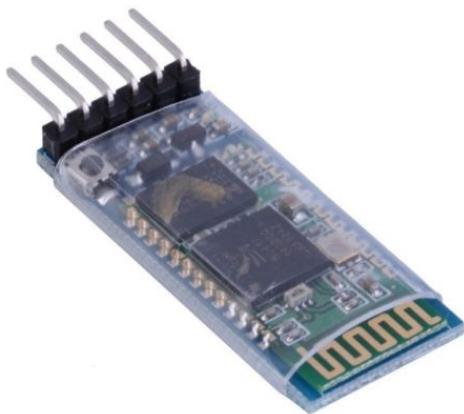


Figura 7.14: Flower one.

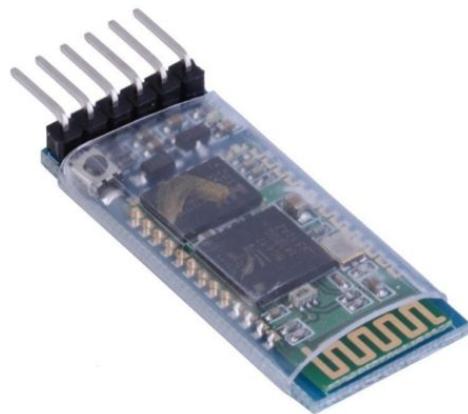


Figura 7.15: Flower two.

<http://www.instructables.com/id/Modifying-the-AT-Codes-on-a-HC-05-With-the-Code-ZS/>

<http://www.arduinoecia.com.br/2013/03/modulo-bluetooth-jy-mcu-configuracao.html>

Testar ligação com modulo foi usada app bluetooth Terminal HC-05

7.4 Interligação de componentes

7.5 Considerações finais

↓1 fase testar coneccao arduino to rasp via porta serie; foi criado um script em python para processar info e enviar para o servidor através da API

↓2 fase : necessidade de tornar um módulo isolado sem necessidade de fio; foi testado um modulo wifi e bluetooth;

↓neste contexto modulo wifi nao!... pretende-se que os sensor moduels sejam de baixo custo e low power. foi utilizado um modulo bluetooth; foi testada a conexao da comm bluetooth através de uma client disponveil na google play bluetooth terminal HC-05

↓ pq nao foi usado um sensor de salinidade?

Sistema de deteção de intrusos

No contexto desta dissertação houve necessidade de implementar um sistema de video-stream que permitisse detetar intrusos, maioritariamente pessoas ou animais de grande porte, que possam invadir as quintas onde se produz salicornia. Esta necessidade prende-se essencialmente com elevado custo do hardware do sistema de monitorização e também de eventuais instrumentos de elevado custo necessários ao cultivo desta espécie (e.g. geradores, máquinas elétricas para poda).

Neste capítulo será descrita a tecnologia de processamento de imagem utilizada tal como o algoritmo disponibilizado pela mesma.

8.1 Biblioteca de processamento de imagem: OpenCV

O OpenCV, também conhecido por *Open Source Computer Vision Library*, é uma biblioteca de software de visão por computador de código *open source* (figura 8.1). OpenCV foi construído para fornecer uma infra-estrutura comum para aplicações de visão computacional e para criar o uso da percepção da máquina nos produtos comerciais.

A biblioteca possui mais de 2500 algoritmos otimizados, que inclui um conjunto abrangente de algoritmos clássicos e avançados de visão computacional e algoritmos de *machine learning*. Esses algoritmos podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, detetar movimentos numa câmara, seguir um objetos em movimento, produzir nuvens de pontos 3D de câmaras estéreo, entre outros. OpenCV tem mais de 47 mil pessoas na comunidade de usuários e o número estimado de downloads superior a 7 milhões. A biblioteca é amplamente utilizada em empresas e grupos de pesquisa [17].

O OpenCV é usado principalmente em aplicações de visão em tempo real. Esta biblioteca tem interfaces nas mais diversas linguagens: C++, C, Python, Java e MATLAB, embora seja nativamente escrito em C. OpenCV tem suporte para Windows, Linux e Mac OS[17].



Figura 8.1: Logótipo OpenCV

8.1.1 Conclusões

Desde logo a escolha da tecnologia para processamento de imagem recaiu sobre o opencv não apenas por ser uma biblioteca bastante popular e possuir bastantes algoritmos implementados mas também por eu próprio possuir já algum background e projetos desenvolvidos neste contexto.

Pretendeu-se que este processamento fosse implementado em material já adquirido sem necessidade de gastos. Optou-se então por utilizar um *Raspberry Pi* que juntamente com um *Raspberry Pi camera module* (figura 8.2) permitirá a aquisição de imagem e servirá também como *controller module* ao sistema de aquisição de dados.

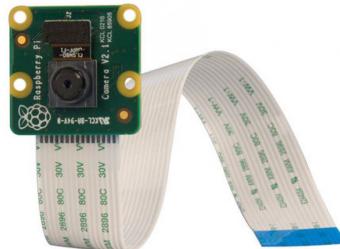


Figura 8.2: Raspberry Pi Camera Board V2 8MP 1080p

Eis algumas das características principais do Raspberry Pi Camera Board V2:

- lente de foco fixo on-board
- 150 milímetros CSI cabo da câmara incluída
- 8 megapixels do sensor com capacidade de resolução nativa de 3.280 imagens estáticas de pixels x 2464

- Suporta 1080p30, 720p60 e 640x480p90 vídeo
- Tamanho 25 milímetros x 23 milímetros x 9 mm
- Peso pouco mais de 3 g
- Liga-se à placa de framboesa Pi por meio de um cabo de fita curta (fornecido)
- Camera v2 é compatível com a última versão do Raspbian, sistema operacional preferido do Raspberry Pi

No que toca ao desenvolvimento, optou-se por utilizar o package picamera. Este pacote fornece uma interface em Python (disponível para qualquer versão) para o módulo de câmera Raspberry Pi¹, permitindo uma fácil interação entre a aquisição da imagem e respetivo processamento. Neste contexto optou-se obviamente por utilizar a interface Python da biblioteca do OpenCV.

8.2 Algoritmo de deteção de intrusos

De modo a estudar alguns algoritmos de deteção de pessoas foram estudados alguns artigos neste contexto.

Para a resolução deste problema foi efetuados

HOGDescriptor: classe que implementa um histograma de gradientes orientado ([Dalal2005]) detetor de objetos.

```
hog = cv2.HOGDescriptor() hog.setSVMClassifier(cv2.HOGDescriptor.getDefaultPeopleDetector())
```

Usado biblioteca do opencv que permite detectar HOGDescriptor

Deteção de intrusos:

<http://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv/>

versão simplificada: <http://www.pyimagesearch.com/2015/02/16/faster-non-maximum-suppression-python/>

Servidor em flask

deploy <https://iotbytes.wordpress.com/python-flask-web-application-on-raspberry-pi-with-nginx-and-uwsgi/>

Dataset:

é usado um detector HOG juntamente com um classificador linear SVM

parametros do método detectMultiScale do opencv

- **img**: parâmetro obrigatório.

¹<http://picamera.readthedocs.io/en/release-1.13/>

- `hitThreshold`: parâmetro opcional.
- `winStride`: parâmetro opcional.
- `padding`: parâmetro opcional. Os valores típicos para preenchimento incluem `(8, 8)` , `(16, 16)` , `(24, 24)` , e `(32, 32)` .
- `scale`: parâmetro opcional.
- `finalThreshold`: parâmetro opcional.
- `useMeanShiftGrouping`: parâmetro opcional.

Neste contexto apenas foram utilizados os seguintes parâmetros `winStride`, `scale`, `padding`.

8.3 Testes

Foram considerados 4 frames de imagens e no apêndice X

8.4 Implementação

8.4.1 Flask

Flask é considerada uma microframework web desenvolvida em Python e baseado nas bibliotecas WSGI Werkzeug e Jinja2. Escolhi esta microframework pois pretende-se que esta seja executada num microcontrolador com baixos recursos. Para além disso, considera-se ser de fácil aprendizagem relativamente ao Django (já abordado na capítulo XX) e com uma ótima documentação.

8.4.2 Servidor web NGNIX

8.5 Considerações finais

Resultados

9.1 Testar API

Tabela 9.1: Um nome qualquer

| Posição | País | IDH |
|---------|-----------|------|
| 1 | Noruega | .955 |
| 2 | Austrália | .938 |
| 3 | EUA | .937 |
| 4 | Holanda | .921 |
| 5 | Alemanha | .920 |

- Arduino nano
- Sensor de temperatura
- Sensor de luminosidade
- Sensor de umidade do solo
- Nivel do liquido

Porta serie ligada a um rasp.

9.2 Aplicação mobile

9.3 Deteção de intrusos

10

Conclusão e trabalho futuro

10.1 Conclusão

10.2 Trabalho futuro

10.3 Considerações finais

Bibliografia

- [1] T. Our, “Resume : Context Aware Computing for The Internet of Things : A Survey Article 2013,” pp. 1–5, 2013.
- [2] V. Isca, A. Seca, D. Pinto, and A. Silva, *An overview of Salicornia genus: the phytochemical and pharmacological profile*, natural pr ed., V. Gupta, Ed. Daya Publishing House, New Delhi, 2014.
- [3] T. Davidson, *Chambers twentieth century dictionary*, W. & R. Ch, Ed., London, 1903.
- [4] João Silva, “Sal verde, National Geographic.” [Online]. Available: <https://nationalgeographic.sapo.pt/23-arquivo/as-nossas-historias/298-sal-verde> [Accessed: 2017-02-01]
- [5] S. Beer and O. Demina, “A new species of Salicornia (Chenopodiaceae) from European Russia,” pp. 253–257, 2005.
- [6] M. Ferri and N. Menezes, *Glossário Ilustrado de Botânica*, 1st ed., Livraria Nobel, Ed., Brasil, 1981.
- [7] M. H. A. Silva, “Aspectos morfológicos e ecofisiológicos de algumas halófitas do sapal da Ria de Aveiro,” Ph.D. dissertation, Universidade de Aveiro, 2000. [Online]. Available: <http://ria.ua.pt/handle/10773/925>
- [8] A. J. Davy, G. F. Bishop, and C. S. B. Costa, “*Salicornia* L. (*Salicornia pusilla* J. Woods, *S. ramosissima* J. Woods, *S. europaea* L., *S. obscura* P.W. Ball & Tutin, *S. nitens* P.W. Ball & Tutin, *S. fragilis* P.W. Ball & Tutin and *S. dolichostachya* Moss),” *Journal of Ecology*, vol. 89, no. 4, pp. 681–707, 2001.
- [9] R. Pinto, “Expresso — A planta que é uma alternativa ao sal: antes era uma praga, agora é uma erva gourmet,” 2015. [Online]. Available: <http://bit.ly/1PR7KAG> [Accessed: 2017-02-01]

- [10] E. Figueroa, J. Jimenez-Nieva, J. Carranza, and C. Gonzalez Vilches, “Distribucion y Nutricion Mineral de Salicornia ramosissima J. Woods, Salicornia europaea L. y Salicornia dolichostachya Moss. en el estuario de los rios Odiel y Tinto (Huelva, SO España),” *Limnetica*, vol. 3, no. 2, pp. 307–310, 1987.
- [11] H. Silva, G. Caldeira, and H. Freitas, “Salicornia ramosissima population dynamics and tolerance of salinity,” *Ecological Research*, vol. 22, no. 1, pp. 125–134, 2007.
- [12] D. Evans, “A Internet das Coisas Como a próxima evolução da Internet está mudando tudo,” pp. 5–7, 2011.
- [13] B. Getting, “Basic Definitions: Web 1.0, Web 2.0, Web 3.0 — Practical Ecommerce.” [Online]. Available: <http://www.practicalecommerce.com/articles/464-Basic-Definitions-Web-1-0-Web-2-0-Web-3-0> [Accessed: 2017-02-20]
- [14] J. Lovato, “Google’s evolution in 10 years,” 2014. [Online]. Available: <http://www.mediavisioninteractive.com/blog/search-enginenews/looking-back-moving-forward-google-evolution/> [Accessed: 2017-02-20]
- [15] J. Rowley, “The wisdom hierarchy: representations of the DIKW hierarchy,” *Journal of Information Science*, vol. 33, no. 2, pp. 163–180, 2007. [Online]. Available: <http://alturl.com/7qike>
- [16] A. S. H. G. Abreu, “Sistema de Monitorização de Estufas Agrícolas,” Ph.D. dissertation, University of Aveiro, 2012. [Online]. Available: <http://ria.ua.pt/handle/10773/10269>
- [17] Itseez, “About - OpenCV library.” [Online]. Available: <http://opencv.org/about.html> [Accessed: 2017-05-19]

A

Untitled appendix #A

Write something here...

B

Mockups da aplicação mobile

Tabela B.1: Um nome qualquer

| Posição | País | IDH |
|---------|-----------|------|
| 1 | Noruega | .955 |
| 2 | Austrália | .938 |
| 3 | EUA | .937 |
| 4 | Holanda | .921 |
| 5 | Alemanha | .920 |

C

Trigger SQL

```
1 CREATE OR REPLACE FUNCTION alarm_occurred() returns trigger as
2   $alarm$
3
4   DECLARE
5     varmax FLOAT;
6     varmin FLOAT;
7
8   BEGIN
9
10    varmax := (select max from saliapp_alarmssettings where
11      id_sensor_id= new.id_sensor_id);
12    varmin := (select min from saliapp_alarmssettings where
13      id_sensor_id= new.id_sensor_id);
14
15    IF (new.value >= varmax) THEN
16      insert into saliapp_alarms (id_reading_id, checked, max_or_min)
17        VALUES (new.id, 'f', 't');
18
19    RETURN new;
20
21  END IF;
22
23  IF (new.value <= varmin) THEN
24    insert into saliapp_alarms (id_reading_id, checked, max_or_min)
25      VALUES (new.id, 'f', 'f');
26
27  RETURN new;
28
29  END IF;
30
```

```
19 RETURN NULL;
20 END
21 $alarm$
22 LANGUAGE plpgsql;
23
24 create trigger trigger_alarm_occurred after insert on
25   saliapp_reading
26 for each row execute procedure alarm_occurred();
27
28 DROP FUNCTION alarm_occurred();
29
30 DROP TRIGGER trigger_alarm_occurred ON saliapp_reading;
```

D

Resultados processamento de imagem

Características do computador

- CPU: Intel Core i7-3630QM CPU @ 2.40GHz x 8
- SO version: Ubuntu 16.04.2 LTS
- Intel Corporation 3rd Gen Core processor Graphics Controller (rev 09) NVIDIA Corporation GF108M [GeForce GT 635M] (rev a1)

D.1 Frame 1



Figura D.1: Pirâmide do conhecimento: modelo DIKW

Características:

- Dimensões (px):
- Tamanho (MB):
- Número de pessoas existentes:

| winStride | padding | scale | detection (number) | execution time (seg) |
|------------------|----------------|--------------|---------------------------|-----------------------------|
| (2, 2) | (8, 8) | 0.5 | 4 | 0.184819936752 |
| (4, 4) | (8, 8) | 0.5 | 3 | 0.0488700866699 |
| (8, 8) | (8, 8) | 0.5 | 1 | 0.0153889656067 |
| (2, 2) | (8, 8) | 1.0 | 4 | 0.17699098587 |
| (4, 4) | (8, 8) | 1.0 | 3 | 0.0484340190887 |
| (8, 8) | (8, 8) | 1.0 | 1 | 0.0148591995239 |
| (2, 2) | (8, 8) | 1.5 | 6 | 0.177606105804 |
| (4, 4) | (8, 8) | 1.5 | 5 | 0.0484080314636 |
| (8, 8) | (8, 8) | 1.5 | 2 | 0.0160319805145 |
| (2, 2) | (16, 16) | 0.5 | 4 | 0.193215847015 |
| (4, 4) | (16, 16) | 0.5 | 3 | 0.0518131256104 |
| (8, 8) | (16, 16) | 0.5 | 1 | 0.0164451599121 |
| (2, 2) | (16, 16) | 1.0 | 4 | 0.193369865417 |
| (4, 4) | (16, 16) | 1.0 | 3 | 0.05233502388 |

| | | | | |
|--------|----------|-----|---|-----------------|
| (8, 8) | (16, 16) | 1.0 | 1 | 0.0161139965057 |
| (2, 2) | (16, 16) | 1.5 | 6 | 0.193920850754 |
| (4, 4) | (16, 16) | 1.5 | 5 | 0.0550818443298 |
| (8, 8) | (16, 16) | 1.5 | 2 | 0.0162160396576 |
| (2, 2) | (24, 24) | 0.5 | 4 | 0.203732967377 |
| (4, 4) | (24, 24) | 0.5 | 3 | 0.0558068752289 |
| (8, 8) | (24, 24) | 0.5 | 1 | 0.0173289775848 |
| (2, 2) | (24, 24) | 1.0 | 4 | 0.203326940536 |
| (4, 4) | (24, 24) | 1.0 | 3 | 0.0569319725037 |
| (8, 8) | (24, 24) | 1.0 | 1 | 0.0179741382599 |
| (2, 2) | (24, 24) | 1.5 | 6 | 0.20330619812 |
| (4, 4) | (24, 24) | 1.5 | 5 | 0.0555651187897 |
| (8, 8) | (24, 24) | 1.5 | 2 | 0.0173530578613 |

Tabela D.1: Your caption here

D.2 Frame 2



Figura D.2: Pirâmide do conhecimento: modelo DIKW

| winStride | padding | scale | detection (number) | execution time (seg) |
|------------------|----------------|--------------|---------------------------|-----------------------------|
| (2, 2) | (8, 8) | 0.5 | 11 | 0.335342168808 |
| (4, 4) | (8, 8) | 0.5 | 4 | 0.0799450874329 |
| (8, 8) | (8, 8) | 0.5 | 0 | 0.0238499641418 |
| (2, 2) | (8, 8) | 1.0 | 11 | 0.293792009354 |
| (4, 4) | (8, 8) | 1.0 | 4 | 0.0808959007263 |
| (8, 8) | (8, 8) | 1.0 | 0 | 0.024552822113 |
| (2, 2) | (8, 8) | 1.5 | 10 | 0.310877084732 |
| (4, 4) | (8, 8) | 1.5 | 6 | 0.0828230381012 |
| (8, 8) | (8, 8) | 1.5 | 1 | 0.031553030014 |
| (2, 2) | (16, 16) | 0.5 | 11 | 0.356366157532 |
| (4, 4) | (16, 16) | 0.5 | 5 | 0.0858371257782 |
| (8, 8) | (16, 16) | 0.5 | 0 | 0.0261859893799 |
| (2, 2) | (16, 16) | 1.0 | 11 | 0.324184179306 |
| (4, 4) | (16, 16) | 1.0 | 5 | 0.0870020389557 |
| (8, 8) | (16, 16) | 1.0 | 0 | 0.0258660316467 |
| (2, 2) | (16, 16) | 1.5 | 10 | 0.321846008301 |
| (4, 4) | (16, 16) | 1.5 | 7 | 0.0916659832001 |
| (8, 8) | (16, 16) | 1.5 | 1 | 0.0345950126648 |
| (2, 2) | (24, 24) | 0.5 | 11 | 0.343872070312 |
| (4, 4) | (24, 24) | 0.5 | 5 | 0.0918598175049 |
| (8, 8) | (24, 24) | 0.5 | 0 | 0.0270938873291 |

| | | | | |
|--------|----------|-----|----|-----------------|
| (2, 2) | (24, 24) | 1.0 | 11 | 0.344779968262 |
| (4, 4) | (24, 24) | 1.0 | 5 | 0.090653181076 |
| (8, 8) | (24, 24) | 1.0 | 0 | 0.0263440608978 |
| (2, 2) | (24, 24) | 1.5 | 10 | 0.355221986771 |
| (4, 4) | (24, 24) | 1.5 | 7 | 0.0967049598694 |
| (8, 8) | (24, 24) | 1.5 | 1 | 0.0326068401337 |

Tabela D.2: Your caption here

D.3 Frame 3



Figura D.3: Pirâmide do conhecimento: modelo DIKW

| winStride | padding | scale | detection (number) | execution time (seg) |
|------------------|----------------|--------------|---------------------------|-----------------------------|
| (2, 2) | (8, 8) | 0.5 | 8 | 1.25844407082 |
| (4, 4) | (8, 8) | 0.5 | 5 | 0.359390974045 |
| (8, 8) | (8, 8) | 0.5 | 1 | 0.131782054901 |
| (2, 2) | (8, 8) | 1.0 | 8 | 1.27126002312 |
| (4, 4) | (8, 8) | 1.0 | 5 | 0.355902910233 |
| (8, 8) | (8, 8) | 1.0 | 1 | 0.131030082703 |
| (2, 2) | (8, 8) | 1.5 | 16 | 1.26964783669 |
| (4, 4) | (8, 8) | 1.5 | 12 | 0.364797115326 |
| (8, 8) | (8, 8) | 1.5 | 1 | 0.197186946869 |
| (2, 2) | (16, 16) | 0.5 | 8 | 1.3578350544 |
| (4, 4) | (16, 16) | 0.5 | 5 | 0.357763051987 |
| (8, 8) | (16, 16) | 0.5 | 1 | 0.132702112198 |
| (2, 2) | (16, 16) | 1.0 | 8 | 1.27961397171 |
| (4, 4) | (16, 16) | 1.0 | 5 | 0.367429971695 |
| (8, 8) | (16, 16) | 1.0 | 1 | 0.132242918015 |
| (2, 2) | (16, 16) | 1.5 | 17 | 1.28247308731 |
| (4, 4) | (16, 16) | 1.5 | 12 | 0.403631925583 |
| (8, 8) | (16, 16) | 1.5 | 1 | 0.207641839981 |
| (2, 2) | (24, 24) | 0.5 | 8 | 1.43096494675 |
| (4, 4) | (24, 24) | 0.5 | 5 | 0.369131088257 |
| (8, 8) | (24, 24) | 0.5 | 1 | 0.134386062622 |
| (2, 2) | (24, 24) | 1.0 | 8 | 1.34318900108 |
| (4, 4) | (24, 24) | 1.0 | 5 | 0.371593952179 |

| | | | | |
|--------|----------|-----|----|----------------|
| (8, 8) | (24, 24) | 1.0 | 1 | 0.134378194809 |
| (2, 2) | (24, 24) | 1.5 | 17 | 1.39831089973 |
| (4, 4) | (24, 24) | 1.5 | 13 | 0.444314002991 |
| (8, 8) | (24, 24) | 1.5 | 1 | 0.137616872787 |

Tabela D.3: Your caption here

D.4 Frame 4



Figura D.4: Pirâmide do conhecimento: modelo DIKW

| winStride | padding | scale | detection (number) | execution time (seg) |
|------------------|----------------|--------------|---------------------------|-----------------------------|
| (2, 2) | (8, 8) | 0.5 | 7 | 1.3150138855 |
| (4, 4) | (8, 8) | 0.5 | 2 | 0.36035490036 |
| (8, 8) | (8, 8) | 0.5 | 0 | 0.129312992096 |
| (2, 2) | (8, 8) | 1.0 | 7 | 1.24681711197 |
| (4, 4) | (8, 8) | 1.0 | 2 | 0.358268976212 |
| (8, 8) | (8, 8) | 1.0 | 0 | 0.130249023438 |
| (2, 2) | (8, 8) | 1.5 | 17 | 1.58746790886 |
| (4, 4) | (8, 8) | 1.5 | 12 | 0.513493061066 |
| (8, 8) | (8, 8) | 1.5 | 1 | 0.197572946548 |
| (2, 2) | (16, 16) | 0.5 | 7 | 1.36501693726 |
| (4, 4) | (16, 16) | 0.5 | 2 | 0.363034009933 |
| (8, 8) | (16, 16) | 0.5 | 0 | 0.132270812988 |
| (2, 2) | (16, 16) | 1.0 | 7 | 1.29145503044 |
| (4, 4) | (16, 16) | 1.0 | 2 | 0.359399080276 |
| (8, 8) | (16, 16) | 1.0 | 0 | 0.132076025009 |
| (2, 2) | (16, 16) | 1.5 | 19 | 1.61724209785 |
| (4, 4) | (16, 16) | 1.5 | 13 | 0.467741012573 |
| (8, 8) | (16, 16) | 1.5 | 1 | 0.170053005219 |
| (2, 2) | (24, 24) | 0.5 | 7 | 1.33659911156 |
| (4, 4) | (24, 24) | 0.5 | 2 | 0.365787982941 |
| (8, 8) | (24, 24) | 0.5 | 0 | 0.133852005005 |
| (2, 2) | (24, 24) | 1.0 | 7 | 1.29908204079 |
| (4, 4) | (24, 24) | 1.0 | 2 | 0.377649784088 |

| | | | | |
|--------|----------|-----|----|---------------|
| (8, 8) | (24, 24) | 1.0 | 0 | 0.13329410553 |
| (2, 2) | (24, 24) | 1.5 | 19 | 1.32506895065 |
| (4, 4) | (24, 24) | 1.5 | 13 | 0.38186788559 |
| (8, 8) | (24, 24) | 1.5 | 1 | 0.1673848629 |

Tabela D.4: Your caption here

E

Interface gráfica

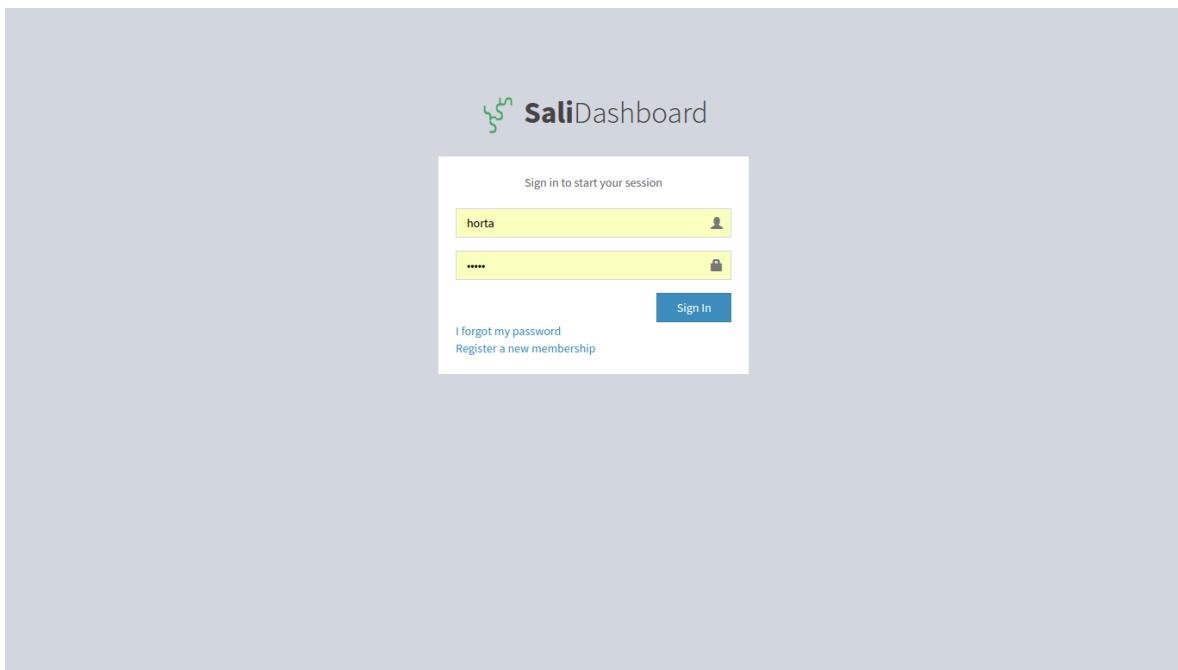


Figura E.1: Pirâmide do conhecimento: modelo DIKW

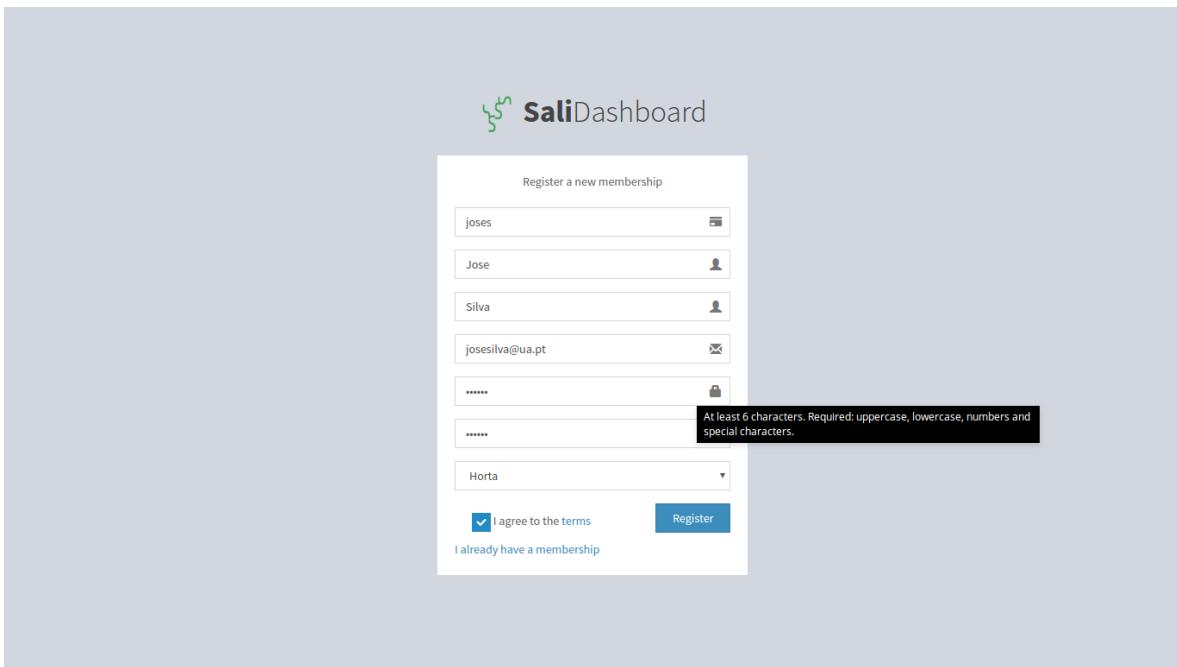


Figura E.2: Pirâmide do conhecimento: modelo DIKW

F

Descrição formal dos casos de uso gerais

| | |
|----------------------------|---|
| Caso de utilização: | -dasdasdasdasddddddddd -dasdasd -dasdasda |
| Autor: | |
| Descrição: | |
| Pré-condições: | |
| Pós-condições: | |

Tabela F.1: Casos de utilização: Login/Logout

| | |
|----------------------------|---|
| Caso de utilização: | -dasdasdasdasddddddddd -dasdasd -dasdasda |
| Autor: | |
| Descrição: | |
| Pré-condições: | |
| Pós-condições: | |

Tabela F.2: Casos de utilização: Login/Logout

G

Interligação de componentes em hardware

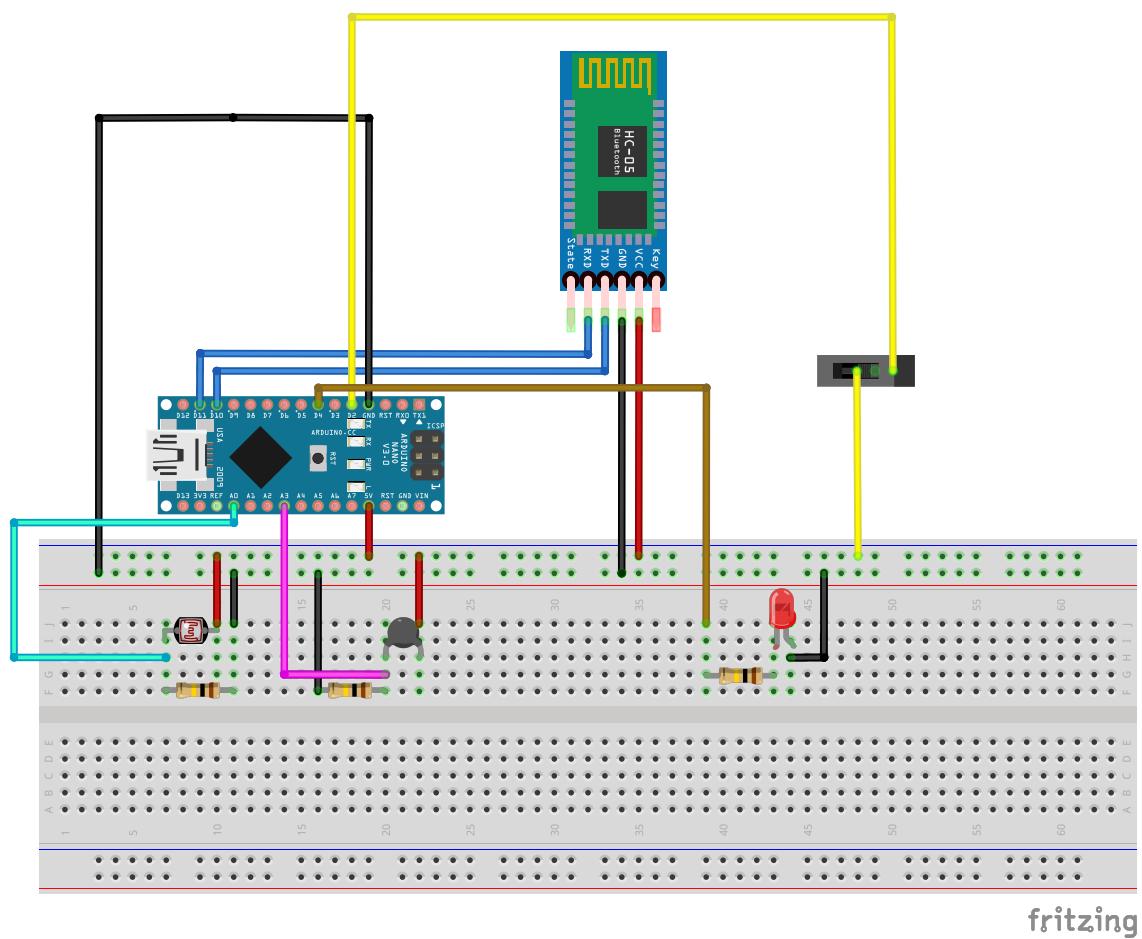


Figura G.1: Pirâmide do conhecimento: modelo DIKW