



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 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...**

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

<b>Lista de Figuras</b>	v
<b>Lista de Tabelas</b>	vii
<b>Acrónimos</b>	ix
<b>1 Introdução</b>	1
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Organização do documento . . . . .	2
<b>2 Conceito de IoT no cultivo da Salicórnia</b>	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 Evolução tecnológica: o IoT . . . . .	8
2.4.1 Vantagens . . . . .	10
2.4.2 Desvantagens . . . . .	10
2.5 Considerações finais . . . . .	10
<b>3 Estado da arte</b>	11
3.1 Sistema de gestão de base de dados . . . . .	11
3.1.1 MySQL . . . . .	11
3.1.2 SQL server . . . . .	11
3.1.3 PostgreSQL . . . . .	11
3.1.4 Comparação e solução adotada . . . . .	11

---

3.2	Desenvolvimento web . . . . .	13
3.2.1	ASP.net . . . . .	13
3.2.2	Flask . . . . .	13
3.2.3	Django . . . . .	13
3.2.4	Conclusões e solução adotada . . . . .	13
3.3	Desenvolvimento mobile . . . . .	14
3.3.1	Plataformas nativas . . . . .	14
3.3.2	Multi-plataforma . . . . .	14
3.3.3	Conclusões e solução adotada . . . . .	14
3.4	REST Frameworks . . . . .	15
3.4.1	Flask-RESTful . . . . .	15
3.4.2	Django Rest Framework . . . . .	15
3.4.3	Conclusões e solução adotada . . . . .	15
3.5	Micro-controladores . . . . .	16
3.5.1	Arduino Nano . . . . .	16
3.5.2	Raspberry Pi . . . . .	17
3.6	Sensores . . . . .	18
3.6.1	Sensor de salinidade . . . . .	18
3.6.2	Sensor de temperatura . . . . .	18
3.6.3	Sensor de luminosidade . . . . .	19
3.7	Tecnologias de comunicação . . . . .	20
3.7.1	Zigbee . . . . .	20
3.7.2	LoRa . . . . .	20
3.7.3	Sigfox . . . . .	20
3.7.4	Bluetooth . . . . .	20
3.7.5	WiFi . . . . .	21
3.7.6	Comparação de tecnologias de comunicação . . . . .	21
3.8	Aplicações relacionadas . . . . .	22
3.8.1	Multi-monitorização de estufas agrícolas . . . . .	22
3.8.2	Agroopar . . . . .	22
3.8.3	outras que vale a pena para comparacao.. . . . .	22
<b>4</b>	<b>Sistema de controlo e monitorização: arquitetura e modelação</b>	<b>23</b>
4.1	Descrição global do sistema . . . . .	23
4.2	Componentes . . . . .	25
4.2.1	Controller Module . . . . .	26

4.2.2	Sensor Module . . . . .	26
4.3	Análise de requisitos . . . . .	27
4.3.1	Requisitos funcionais . . . . .	28
4.3.2	Requisitos não funcionais . . . . .	30
4.4	Modelação . . . . .	31
4.4.1	Entidades envolventes . . . . .	31
4.4.2	Casos de uso . . . . .	32
4.4.3	Modelo de dados . . . . .	33
4.5	Arquitetura lógica . . . . .	35
4.5.1	Camada de apresentação . . . . .	35
4.5.2	Camada de lógica de negócio . . . . .	36
4.5.3	Camada de acesso a dados . . . . .	36
4.6	Arquitetura física . . . . .	36
4.6.1	Sistema de informação . . . . .	36
Application Programming Interface (API)	. . . . .	36
4.6.2	Aplicação web . . . . .	37
4.6.3	Aplicação mobile . . . . .	37
4.6.4	Simulação em hardware . . . . .	38
Sensores utilizados . . . . .	. . . . .	39
Comunicação . . . . .	. . . . .	42
4.7	Diagrama de componentes . . . . .	44
4.8	Considerações finais . . . . .	45
<b>5</b>	<b>Implementação</b> . . . . .	<b>47</b>
5.1	Sistema de informação . . . . .	47
5.1.1	Aplicação web . . . . .	47
5.1.2	Aplicação mobile . . . . .	47
5.1.3	<i>Deploy</i> do projecto . . . . .	47
5.2	Simulação em hardware . . . . .	48
5.2.1	Arduino . . . . .	48
5.2.2	Sensores . . . . .	48
5.2.3	Comunicação . . . . .	49
Raspberry Pi . . . . .	. . . . .	49
Comunicação . . . . .	. . . . .	49
5.2.4	Considerações finais . . . . .	50
5.3	Sistema de deteção de intrusos . . . . .	50

5.3.1	Biblioteca de processamento de imagem: OpenCV . . . . .	50
5.3.2	Algoritmos de deteção de intrusos . . . . .	52
5.3.3	Testes . . . . .	53
5.3.4	Implementação . . . . .	53
5.4	Considerações finais . . . . .	53
<b>6</b>	<b>Resultados</b>	<b>55</b>
6.1	Aplicação web . . . . .	55
6.2	Aplicação mobile . . . . .	55
6.3	Testes API . . . . .	55
6.4	Simulação em hardware . . . . .	55
6.5	Sistema de deteção de intrusos . . . . .	55
<b>7</b>	<b>Conclusão e trabalho futuro</b>	<b>57</b>
7.1	Conclusão . . . . .	57
7.2	Trabalho futuro . . . . .	57
7.3	Considerações finais . . . . .	57
<b>A</b>	<b>Untitled appendix #A</b>	<b>62</b>
<b>B</b>	<b>Mockups da aplicação mobile</b>	<b>63</b>
<b>C</b>	<b>Trigger SQL</b>	<b>65</b>
<b>D</b>	<b>Resultados processamento de imagem</b>	<b>67</b>
D.1	Frame 1 . . . . .	68
D.2	Frame 2 . . . . .	70
D.3	Frame 3 . . . . .	72
D.4	Frame 4 . . . . .	74
<b>E</b>	<b>Interface gráfica</b>	<b>77</b>
<b>F</b>	<b>Descrição formal dos casos de uso gerais</b>	<b>79</b>
<b>G</b>	<b>Interligação de componentes</b>	<b>81</b>

# **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
2.3	Evolução da internet em cinco fases . . . . .	9
2.4	Pirâmide do conhecimento: modelo DIKW . . . . .	9
3.1	Arduin Nano . . . . .	17
3.2	Identificação dos pinos no Arduino Nano . . . . .	17
3.3	Raspberry Pi 3 . . . . .	18
3.4	Identificação dos principais componentes no Raspberry Pi 3 . . . . .	18
4.1	Ilustração principais componentes . . . . .	23
4.2	Ilustração de uma "quinta" onde se produz salicornia . . . . .	24
4.3	Pirâmide do conhecimento: modelo DIKW . . . . .	25
4.4	Fase de desenvolvimento de um software . . . . .	27
4.5	Pirâmide do conhecimento: modelo DIKW . . . . .	32
4.6	Pirâmide do conhecimento: modelo DIKW . . . . .	32
4.7	Pirâmide do conhecimento: modelo DIKW . . . . .	33
4.8	Arquitetura lógica . . . . .	35
4.9	TTC 104 NTC . . . . .	39
4.10	Esquema eletrotécnico da ligação do sensor de temperatura . . . . .	39
4.11	Sensor foto-resistência GL5528 . . . . .	40
4.12	Esquema eletrotécnico da ligação do sensor de luminosidade . . . . .	40
4.13	<i>Water Level Switch Liquid Level Sensor Plastic Ball Float</i> . . . . .	41
4.14	Esquema eletrotécnico da ligação do sensor de nível líquido . . . . .	41
4.15	Led simples. . . . .	41

4.16	Esquema eletrotécnico da ligação do led . . . . .	41
4.17	Arquitetura lógica . . . . .	42
4.18	Flower one. . . . .	43
4.19	Esquema eletrotécnico da ligação do módulo bluetooth . . . . .	43
5.1	Logótipo OpenCV . . . . .	51
5.2	Raspberry Pi Camera Board V2 8MP 1080p . . . . .	51
D.1	Pirâmide do conhecimento: modelo DIKW . . . . .	68
D.2	Pirâmide do conhecimento: modelo DIKW . . . . .	70
D.3	Pirâmide do conhecimento: modelo DIKW . . . . .	72
D.4	Pirâmide do conhecimento: modelo DIKW . . . . .	74
E.1	Pirâmide do conhecimento: modelo DIKW . . . . .	77
E.2	Pirâmide do conhecimento: modelo DIKW . . . . .	78
G.1	Pirâmide do conhecimento: modelo DIKW . . . . .	81

# **Lista de Tabelas**

3.1	Características do sensor TTC 104 . . . . .	17
3.2	Comparação entre versão 2 e 3 do Raspberry Pi . . . . .	19
4.1	My caption . . . . .	34
4.2	My caption . . . . .	36
4.3	Características do sensor TTC 104 . . . . .	39
4.4	Características do sensor GL5528 . . . . .	40
4.5	Características do sensor GL5528 . . . . .	43
B.1	Um nome qualquer . . . . .	63
D.1	Your caption here . . . . .	69
D.2	Your caption here . . . . .	71
D.3	Your caption here . . . . .	73
D.4	Your caption here . . . . .	75
F.1	Casos de utilização: Login/Logout . . . . .	79
F.2	Casos de utilização: Login/Logout . . . . .	79



# Acrónimos

<b>NTC</b>	Negative Temperature Coefficient
<b>API</b>	Application Programming Interface
<b>HTML</b>	HyperText Markup Language
<b>HTTP</b>	HyperText Transfer Protocol
<b>IDE</b>	Integrated Development Environment
<b>LDR</b>	Light Dependent Resistor
<b>IoT</b>	Internet of Things
<b>SM</b>	Sensor Module
<b>CM</b>	Controller Module
<b>CSV</b>	Comma-Separated Values
<b>SGBD</b>	Sistema de Gestão de Base de Dados
<b>URL</b>	Uniform Resource Locator
<b>CSS</b>	Cascading Style Sheets
<b>SDLC</b>	Systems Development Life Cycle
<b>WWW</b>	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>

HyperText Markup Language (HTML)

\* O gênero salicornia HTML inclui cerca de 117 espécies, sendo Salicornia herbacea, Salicornia bigelovii, Salicornia europea, Salicornia prostata, Salicornia mmosissima e Salicornia virginica aquelas com maior ocorrência. [1] [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 aquários 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 este 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

A presente dissertação está dividida em 7 capítulos: ...

O primeiro capítulo descreve e enfatiza a importância  
De seguida, no Estado de Arte, é  
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.



# Conceito de IoT no cultivo da Salicórnia

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[1].

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  $\beta$ -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 confeção 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.

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.

## 2.4 Evolução tecnológica: o IoT

Antes de descrever a importância e o conceito de Internet of Things (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 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 2.3 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 [15].

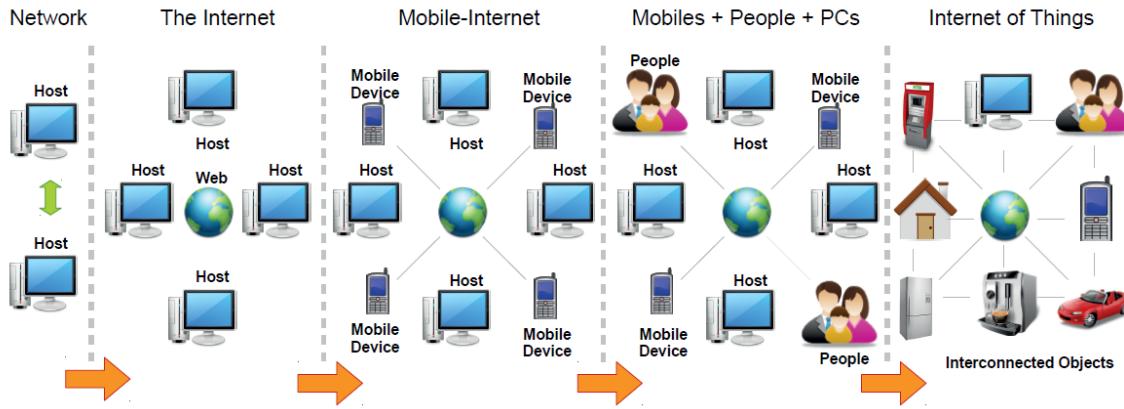


Figura 2.3: Evolução da internet em cinco fases (Adaptado de [15])

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 [16]. 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.

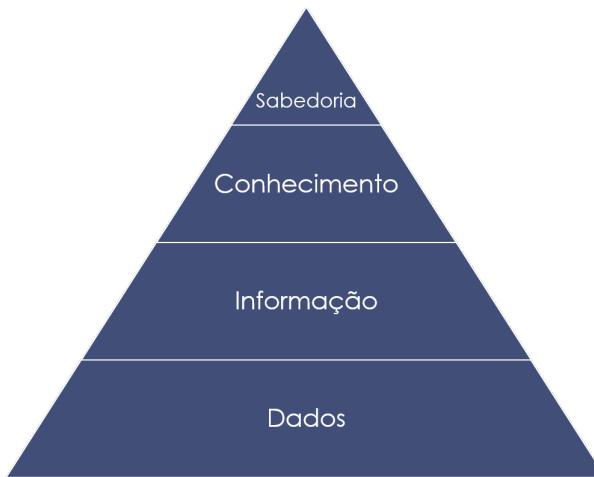


Figura 2.4: Pirâmide do conhecimento: modelo DIKW

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.

**2.4.1 Vantagens****2.4.2 Desvantagens****2.5 Considerações finais**

# 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. Sistema de Gestão de Base de Dados (SGBD))

## 3.1 Sistema de gestão de base de dados

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

### 3.1.1 MySQL

### 3.1.2 SQL server

### 3.1.3 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, isto é, 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* [17].

### 3.1.4 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].

## 3.2 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

### 3.2.1 ASP.net

### 3.2.2 Flask

### 3.2.3 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.

### 3.2.4 Conclusões e solução adotada

### **3.3 Desenvolvimento mobile**

#### **3.3.1 Plataformas nativas**

#### **3.3.2 Multi-plataforma**

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

#### **3.3.3 Conclusões e solução adotada**

## 3.4 REST Frameworks

### 3.4.1 Flask-RESTful

### 3.4.2 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 .

### 3.4.3 Conclusões e solução adotada

com autenticação via token

app mobile microcontroladores -*i* controller modulers

documentação com swagger

## 3.5 Micro-controladores

Para o cenário anteriormente descrito foram utilizados dois micro-controladores bastante comuns no mercado: um Arduino Nano e um Raspberry Pi 3. Neste contexto, assume-se que o Arduino Nano é considerado um Sensor Module (SM) que possui um conjunto de sensores enquanto que o Raspberry Pi 3 é um Controller Module (CM) que recebe os dados provenientes do SM enviando-os para o servidor.

### 3.5.1 Arduino Nano

O Arduino é fruto da evolução de um projeto italiano desenvolvido no ano de 2005, cujo objetivo foi 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 com elevada facilidade utilização]. O Arduino é utilizado para projetos especialmente no contexto do IoT e da robótica educativa. Neste micro-controlador, podem ser estendidos vários módulos, dependendo da tarefa que se quer que seja executada.

O Arduino possui um conjunto de pinos que podem ser programados para funcionarem como entradas ou saídas fazendo com que o Arduino interaja com o meio externo para os mais diversos fins. Para além dos pinos de I/O existem pinos de alimentação que Fornecem diversos valores de tensão que podem ser utilizados para transmitir energia elétrica aos diferentes componentes de um projeto.

Na figura 3.1 e 3.2 apresenta-se uma imagem do arduino utilizado e a identificação dos diferentes pinos existentes, respectivamente. Na tabela 3.1 encontram-se as principais características desta versão do Arduino.

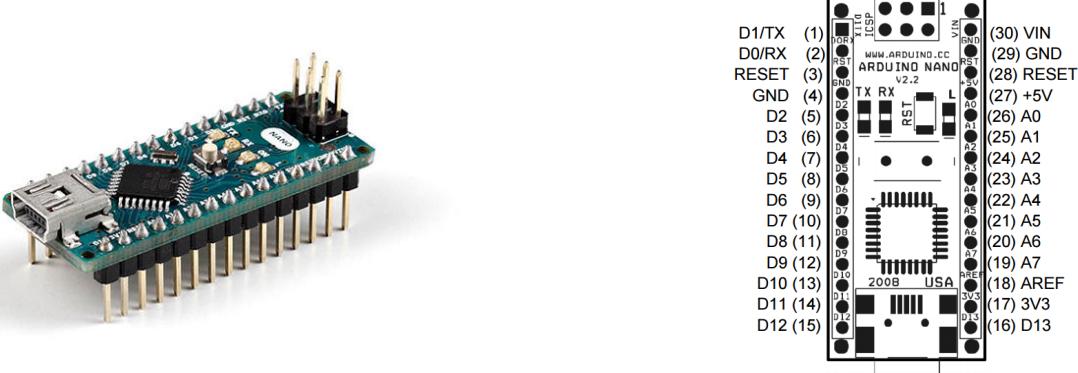


Figura 3.1: Arduin Nano

Figura 3.2: Identificação dos pinos no Arduino Nano

Microcontrolador	ATmega328
Tensão de operação	5V
Tensão de entrada	7-12V
Portas digitais	14 (6 podem ser usadas como PWM)
Portas analógicas	8
Corrente nos pinos I/O	40mA
Memória Flash	32KB (2KB usado no bootloader)
Memória RAM (SRAM)	2KB
EEPROM	1KB
Velocidade do Clock	16MHz
Dimensões	45 x 18mm
LED Interno	Pino digital 13
Ligaçāo USB	Ligaçāo ao computador e alimentação

Tabela 3.1: Características do sensor TTC 104

### 3.5.2 Raspberry Pi

O Raspberry Pi (figura 3.3) é considerado um micro-computador do tamanho de um cartão de crédito que possui um conjunto de *hardware* integrado que tal como Arduino possibilita uma interação com o meio exterior. O principal objetivo deste poderoso componente consistiu em promover o ensino da ciência da computação em escolas de ensino básico. O Raspberry Pi foi desenvolvido no Reino Unido pela *Raspberry Pi Foundation*.



Figura 3.3: Raspberry Pi 3

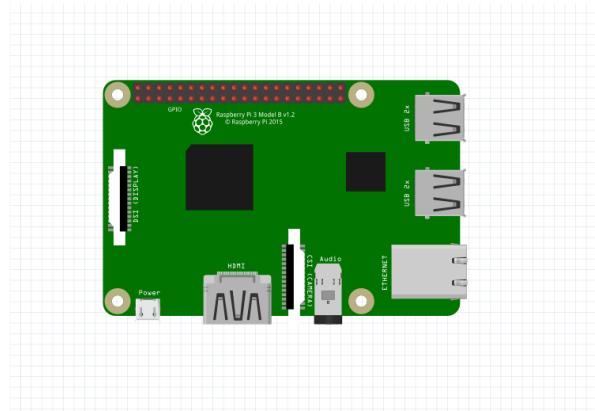


Figura 3.4: Identificação dos principais componentes no Raspberry Pi 3

---

## 3.6 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.

### 3.6.1 Sensor de salinidade

### 3.6.2 Sensor de temperatura

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

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

	<b>Raspberry Pi 3 Model B</b>	<b>Raspberry Pi 2 Model B 1.2</b>
<b>Processor Chipset</b>	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
<b>Processor Speed</b>	QUAD Core @1.2 GHz	QUAD Core @900 MHz
<b>RAM</b>	1GB SDRAM @ 400 MHz	1GB SDRAM @ 400 MHz
<b>Storage</b>	MicroSD	MicroSD
<b>USB 2.0</b>	4x USB Ports	4x USB Ports
<b>Max Power Draw/voltage</b>	2.5A @ 5V	1.8A @ 5V
<b>GPIO</b>	40 pin	40 pin
<b>Ethernet Port</b>	Yes	Yes
<b>WiFi</b>	Built in (802.11n)	No
<b>Bluetooth LE</b>	Built in (4.1)	No

Tabela 3.2: Comparação entre versão 2 e 3 do Raspberry Pi

### 3.6.3 Sensor de luminosidade

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.

## 3.7 Tecnologias de comunicação

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.

### 3.7.1 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.

### 3.7.2 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.

### 3.7.3 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).

### 3.7.4 Bluetooth

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

### 3.7.5 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 802.11. A Rede IEEE possui como principal característica transmitir sinal sem fio através de ondas!

### 3.7.6 Comparação de tecnologias de comunicação

## 3.8 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.

### 3.8.1 Multi-monitorização de estufas agrícolas

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

### 3.8.2 Agroopar

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

### 3.8.3 outras que vale a pena para comparacao..

[18]

# Sistema de controlo e monitorização: arquitetura e modelação

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, especificidades e respectiva arquitetura. É também descrito como os elementos interagem entre si. É também apresentado todo o processo de modelação do sistema tendo por base os requisitos adquiridos pelo cliente.

## 4.1 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.

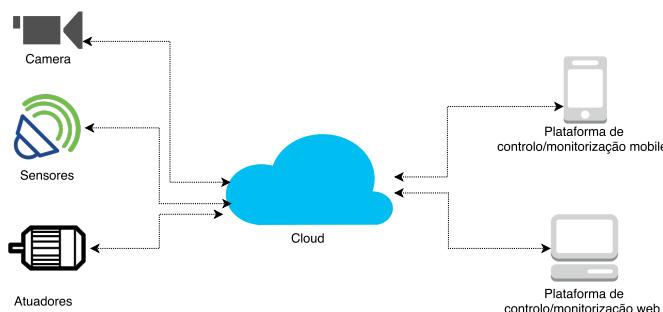


Figura 4.1: Ilustração principais componentes

O esquema da figura 4.1 ilustra todos os componentes de um modo geral e as diferentes plataformas com que o cliente pode interagir.

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.

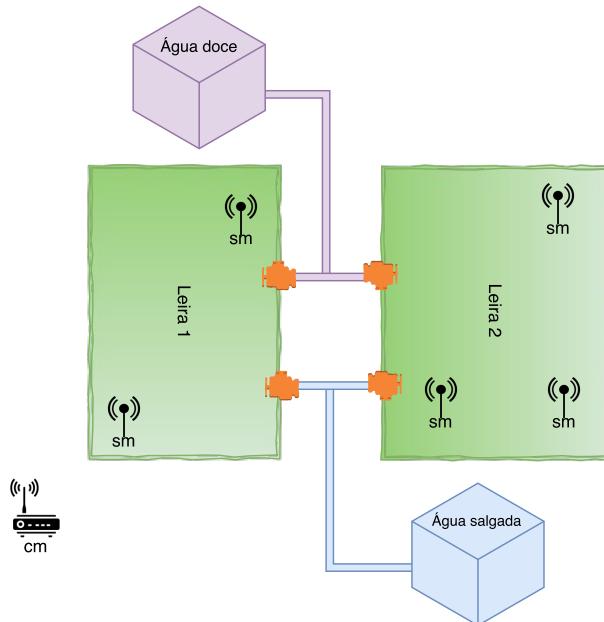


Figura 4.2: Ilustração de uma ”quinta” onde se produz salicornia

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.

## 4.2 Componentes

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

- **Sensor Module:** 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:** 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 4.3, 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.

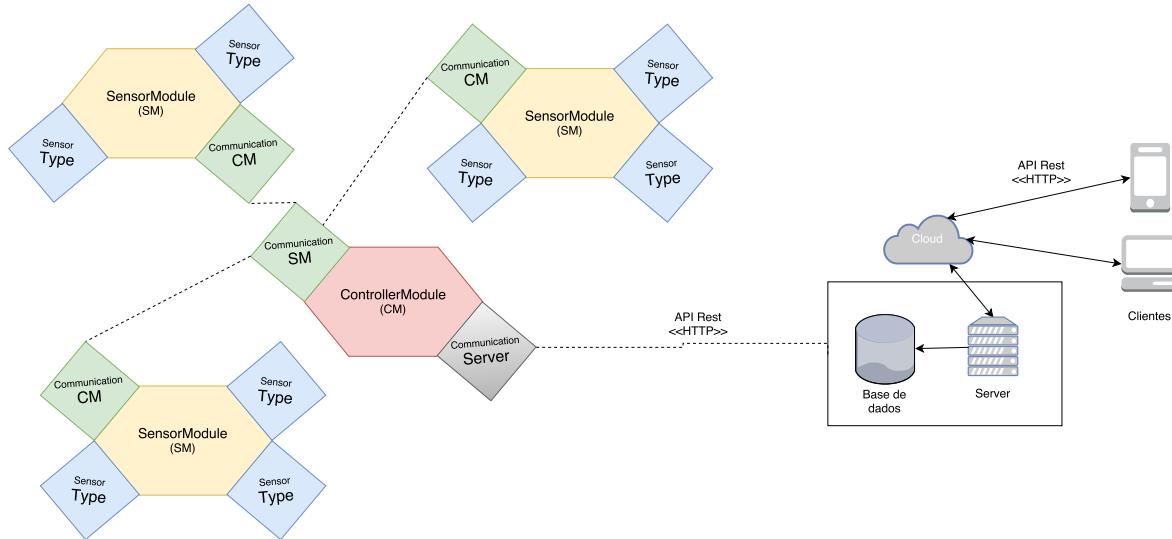


Figura 4.3: Pirâmide do conhecimento: modelo DIKW

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

#### 4.2.1 Controller Module

nome bateria memoria comunicacao localizacao

#### 4.2.2 Sensor Module

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

### 4.3 Análise de requisitos

Durante o desenvolvimento um software pressupõe-se que os seus intervenientes sigam determinadas metodologia para o seu programa possa revolucionar a vida de um grupo em específico ou até mesmo da sociedade.

O ciclo de vida do desenvolvimento de um software, também conhecido como Systems Development Life Cycle (SDLC), é composto genericamente por quatro fase principais: concepção, projeto, criação e a implementação. Antes do SDLC, o processo de desenvolvimento do software foi tomado como atividade informal sem regras e padrões formais. Isso poderá levar a vários problemas, tais como o atraso no desenvolvimento, aumento de custos e baixa qualidade do software criado.

O ciclo de vida do desenvolvimento de um software dá o padrão necessário e as etapas para o desenvolvimento de um software com qualidade. Existem numeros modelos e visões para o ciclo de vida de desenvolvimento, o que a seguir descrevo foi considerado por Munish Kaur [2]. A figura 4.4 mostra as várias etapas que compõe este modelos e que seguidamente são descritas.

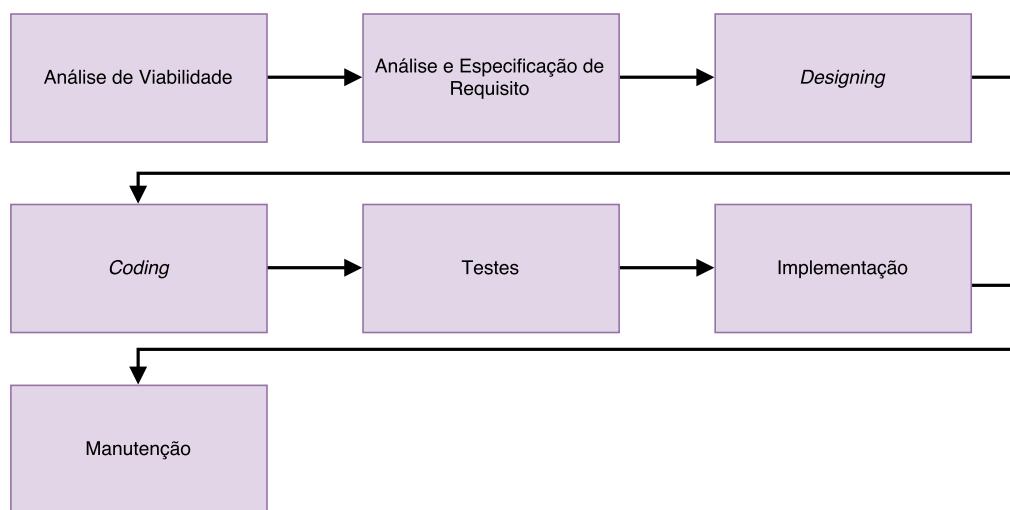


Figura 4.4: Fase de desenvolvimento de um software (Adaptado de [2])

- **Análise de Viabilidade:** Nesta fase, a viabilidade do projeto está sendo avaliada em termos de dados de entrada, dados de saída, processamento necessário para transformar a entrada para a produção necessária, análise de custo e planejamento do projeto. Ele também inclui a viabilidade técnica em termos de software, hardware e pessoas qualificadas. No final do estudo de viabilidade é gerado o relatório de viabilidade.
- **Analise e especificação de requisitos:** nesta fase são recolhidos e analisados os

requisitos necessário para a elaboração do *software*. Pretende-se que no final desta fase sejam conhecidos os vários requisitos do software e seja também criado um documento que os especifique.

- **Designing:** consiste na tradução dos requisitos especificado na fase anterior para a estrutura lógica. Pretende-se que seja elaborado um documento de especificação do *design*.
- **Coding:** o programação real é elaborada nesta fase. O documento de *design* é traduzido para o código-fonte numa determinada linguagem de forma a que possa ser executado.
- **Testes:** o código-fonte gerado na fase anterior é testado usando vários cenários de testes. São usadas várias técnicas de teste para avaliar a correção e validação do *software*.
- **Implementação:** o software desenvolvido é implementado para que possa ser disponibilizado ao utilizador para uso real. Pretende-se que o utilizado do sistema possa reportar erros ou problemas quando encontrados.
- **Manutenção:** o software poderá sofrer alterações que possam solucionar problemas que tenham ocorrido. Esta fase é responsável pela pós-implementação e manutenção do software para o seu bom funcionamento.

A forma como um software realiza as tarefas para as quais foi desenvolvido determina a eficiência de execução do mesmo.

O levantamento de requisitos é umas das partes mais importantes do processo que resultará no desenvolvimento de um determinado sistema. Entender aquilo que o cliente deseja ou o que o cliente acredita que precisa e as regras do negócio ou processos do negócio. Isso é o fator determinante que move essa importante função que faz parte da Engenharia de Software(Engenharia de requisitos).

#### 4.3.1 Requisitos funcionais

Os requisitos funcionais descrevem os critérios que devem ser usados para avaliar as funções específicas ou os comportamentos de um determinado sistema. Seguidamente são apresentados os requisitos funcionais propostos pelo cliente no contexto deste projeto para as duas plataformas disponíveis: web (dashboard) e mobile.

##### Aplicação mobile (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 no sistema, embora tenha que estar obrigatoriamente associado a uma empresa.
- O utilizador comum só terá acesso à sua área reserva após a validação por parte da 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 com as suas características
  - Tipo de comunicação a um controller modules que permite a comunicação como servidor
  - Sensor modules a um determinado controller modules e as suas características
  - Um ou vários tipos de comunicação de um Sensor moduel
  - Um ou vários sensores a um sensor module em que cada sensor é de um determinado tipo
- Visualizar graficamente os dados de cada sensor para um determinado SM.
- Visualizar em modo tabular (dataset) os dados de cada sensor para um determinado SM.
- Em cada uma das visualizações anteriormente descritas, pretende-se que seja possível uma filtragem por data
- O sistema permitirá a exportação dos dados de um determinado SM em formato Comma-Separated Values (CSV).
- Consultar a documentação da API
- Consultar o token de autenticação da API

- Alterar configurações do utilizador

### Aplicação mobile

- A interface da aplicação mobile deve permitir que o utilizador, seja ele qual for, entre ou faça *login* no sistema.
- A interface da aplicação mobile deve permitir que o utilizador, seja ele qual for, saia ou faça *logout* no sistema.
- Visualizar graficamente os dados de cada sensor para um determinado SM.
- Receber alarmes quando um determinado valor lido está fora do estipulado.

#### 4.3.2 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

## 4.4 Modelação

### 4.4.1 Entidades envolventes

No contexto do sistema descrito existem três entidades distintas que são importantes descrever:

- **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 gere todos os general users que se possam associar à sua empresa. 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.

#### 4.4.2 Casos de uso

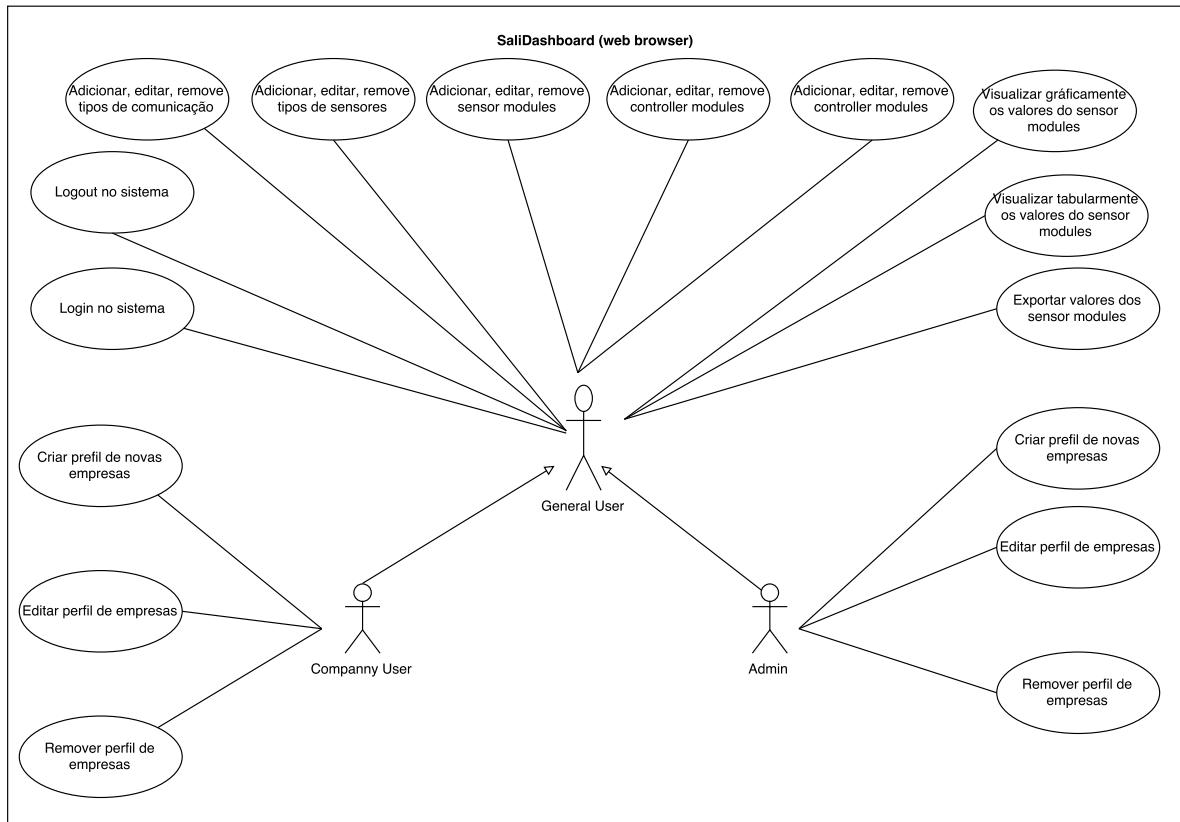


Figura 4.5: Pirâmide do conhecimento: modelo DIKW

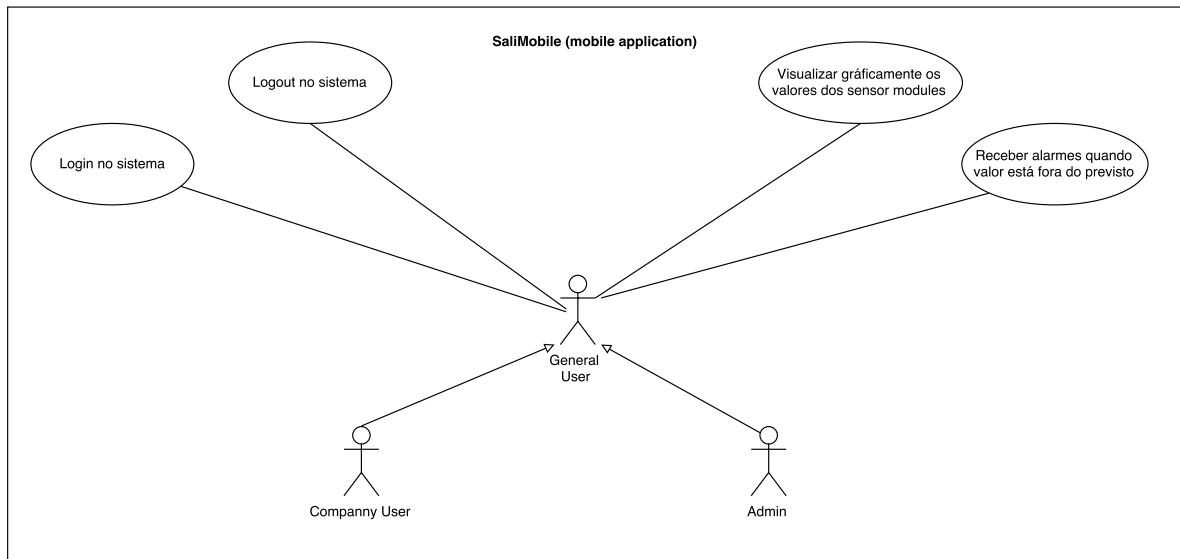


Figura 4.6: Pirâmide do conhecimento: modelo DIKW

### 4.4.3 Modelo de dados

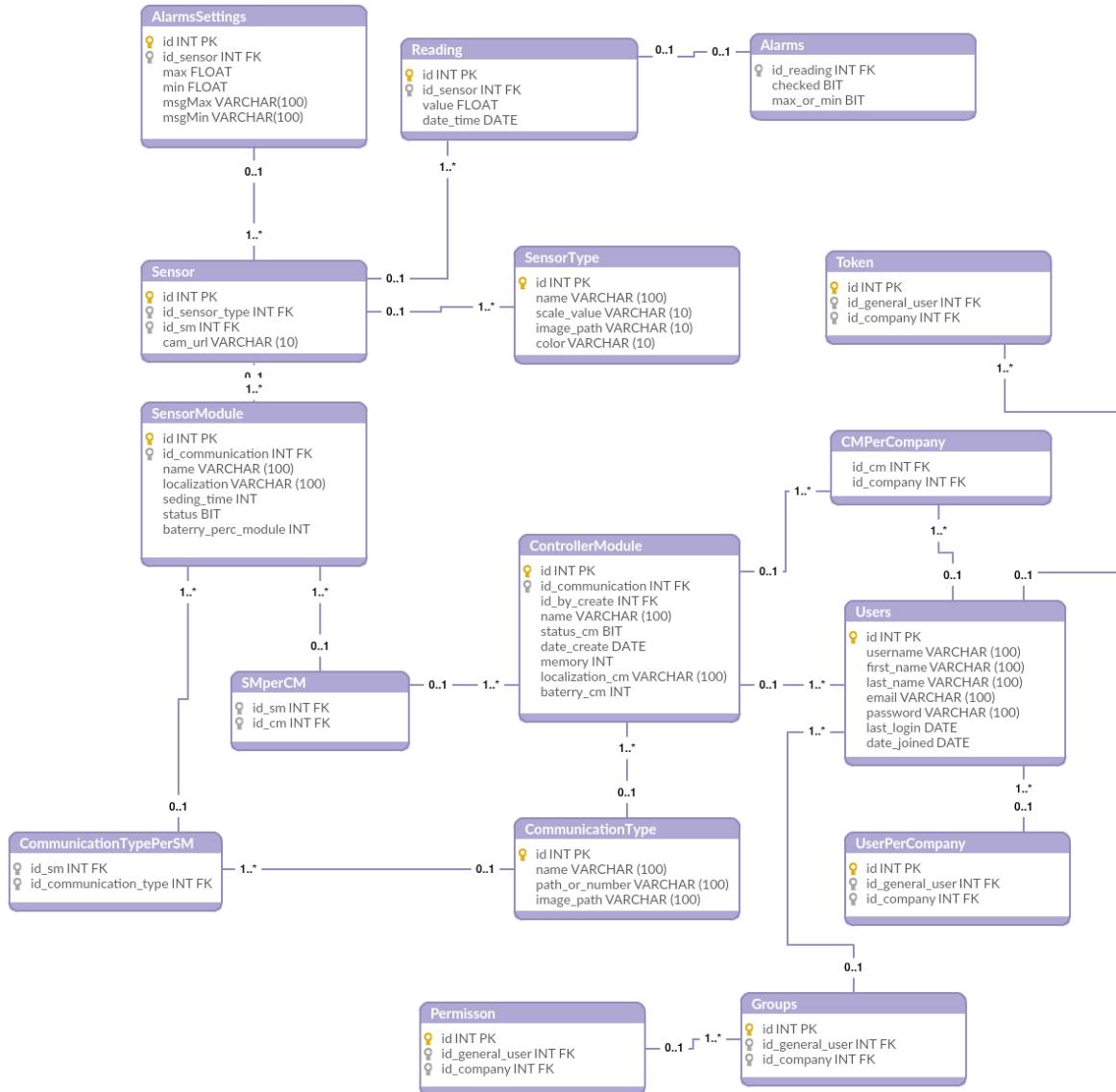


Figura 4.7: Pirâmide do conhecimento: modelo DIKW

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

Tabela 4.1: My caption

## 4.5 Arquitetura lógica

Normalmente este tipo de arquitetura é composto por: camada de apresentação, camada de lógica de negócio e camada de acesso a dados.

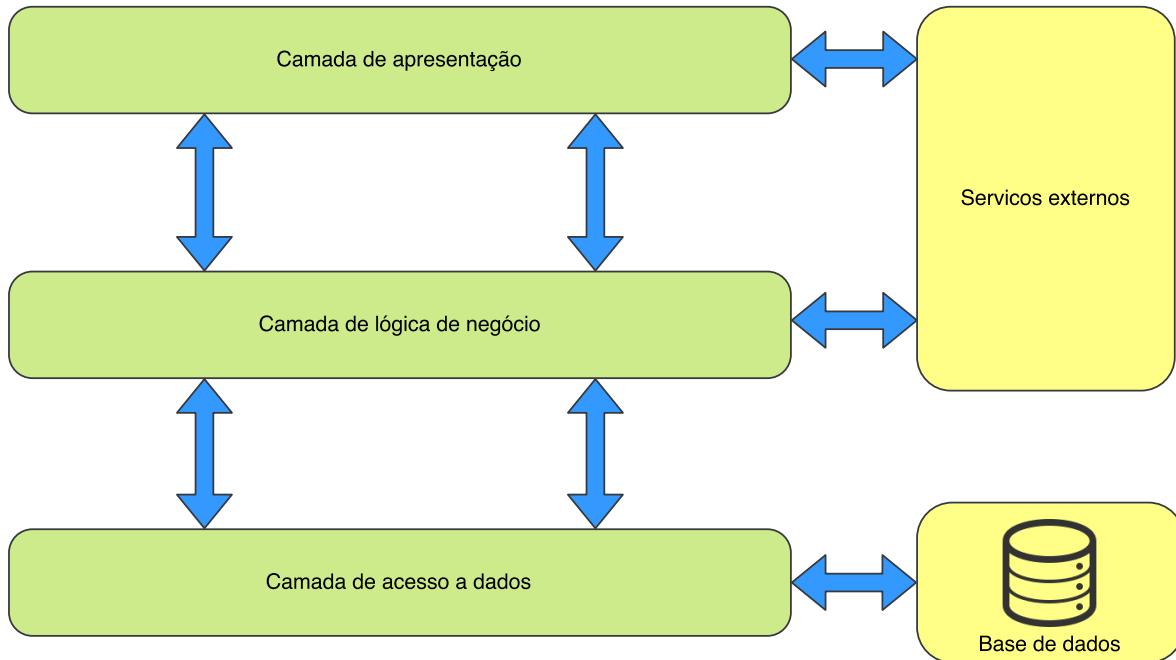


Figura 4.8: Arquitetura lógica

### 4.5.1 Camada de apresentação

A camada de apresentação é responsável pela comunicação entre os utilizadores e a aplicação, sendo ela web ou mobile, exibindo informações aos utilizadores, abrangendo uma interface que permite solicitações ao sistema. Esta camada tem uma relação de dependência com a camada de lógica de negócio e tira partido do acesso a serviços de informação externos, que fornecem diversas funcionalidades.

A interface do utilizador foi desenvolvida em HTML e Cascading Style Sheets (CSS), fazendo uso de jQuery e Javascript.

- HTML é a linguagem padrão usada para estruturar e apresentar conteúdos na web. Neste caso, foi utilizado HTML5, a quinta versão do HTML.
- CSS é uma linguagem usada para descrever a apresentação de conteúdo escrito em uma marcação Como HTML.
- Javascript é a linguagem de programação para páginas da web.

- **JQuery:** JQuery é uma biblioteca Javascript que simplifica a programação Javascript

#### 4.5.2 Camada de lógica de negócio

#### 4.5.3 Camada de acesso a dados

Nesta camada deverão estar presentes as funcionalidades de criação, edição, remoção ou simples de visualização dos dados, sendo responsável pelas operações de persistência e consulta de dados solicitadas pela camada de lógica de negócios.

### 4.6 Arquitetura física

#### 4.6.1 Sistema de informação

##### 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 4.2: My caption

Autenticação

Documentação automática

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

**4.6.2 Aplicação web**

**4.6.3 Aplicação mobile**

#### 4.6.4 Simulação em hardware

Após a desenvolvimento da API descrita no capítulo anterior, pretendeu-se simular o sistema num contexto real. Para tal, planeou-se encontrar hardware que encaixasse no contexto deste projeto. Foram utilizados dois micro-controladores e alguns sensores. Neste capítulo será descrito cada um deles e o processo de desenvolvimento da respetiva simulação.

## Sensores utilizados

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. Todos os sensores que se apresentam encontram-se ligados a um Arduino nano.

### 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 4.10 encontra-se o esquema de ligação deste componente e na tabela 4.3 as propriedades principais.

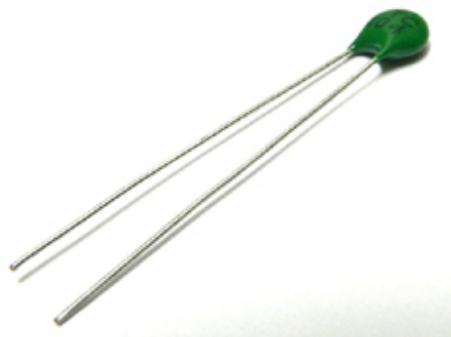


Figura 4.9: TTC 104 NTC

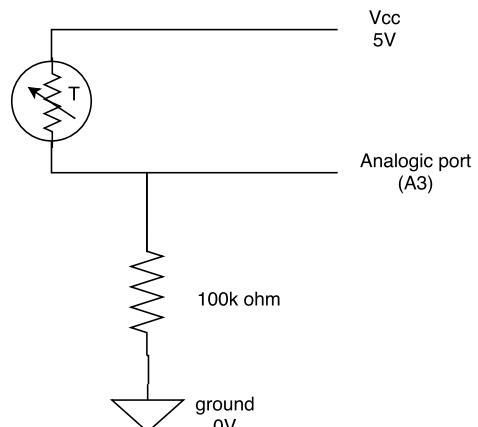


Figura 4.10: Esquema eletrotécnico da ligação do sensor de temperatura

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

Tabela 4.3: Características do sensor TTC 104 [19]

### Luminosidade

Para simular a luminosidade incidente foi utilizado um sensor do tipo foto-resistência. Este sensor, também conhecido como Light Dependent Resistor (LDR), não é mais do que uma resistência variável cujo o seu valor varia conforme a intensidade da luz que incide sobre ele i.e. à medida que a intensidade da luz aumenta, a sua resistência diminui. Este sensor tem múltiplas aplicações, entre as quais se destaca a monitorização solar, indicador da posição do sol (up/down), alarmes anti-roubo, alarme para abertura/fecho de portas entre outras.

Como vimos na secção X do capítulo do Estado de Arte é um sensor de baixo custo e bastante fácil de utilização. Na figura 4.12 encontra-se o esquema de ligação do componente e na tabela 4.5 são apresentadas as principais características do sensor utilizado.

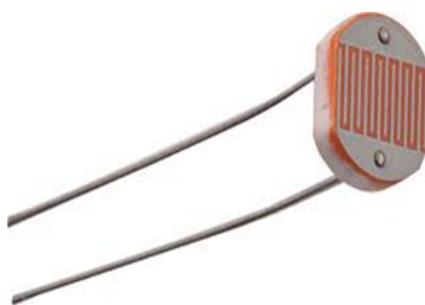


Figura 4.11: Sensor foto-resistência GL5528

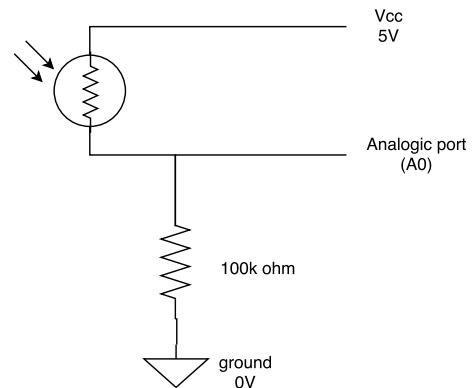


Figura 4.12: Esquema eletrotécnico da ligação do sensor de luminosidade

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)
Material	Carbono
Preço	0.22 €/unidade

Tabela 4.4: Características do sensor GL5528 [20]

### Sensor de nível líquido

Este sensor não é mais do que um interruptor que é ativo sempre que um determinado líquido ultrapassa o mesmo. Sempre que algum líquido atingir o pedaço de plástico este irá subir ativando assim o circuito. Na figura 4.14 encontra-se o esquema da ligação deste sensor.



Figura 4.13: *Water Level Switch Liquid Level Sensor Plastic Ball Float*

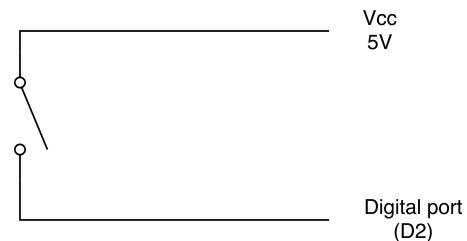


Figura 4.14: Esquema eletrotécnico da ligação do sensor de nível líquido

### Simulador de válvula para transferências de águas

Para a simulação de uma válvula que permitirá as transferências de água doce e salgada foi utilizado um simples *led*. Este possibilita facilmente identificar através da ativação do *led* se a válvula se encontra ativa ou não.



Figura 4.15: Led simples.

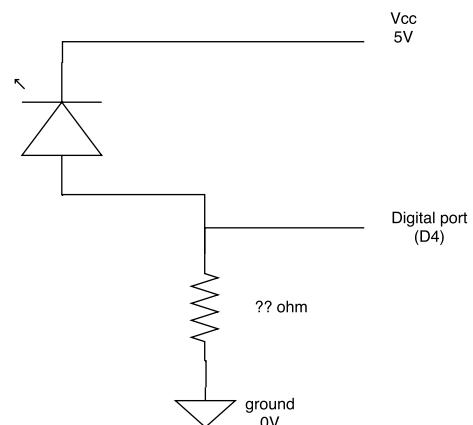


Figura 4.16: Esquema eletrotécnico da ligação do led

## Comunicação

Nesta secção, será apresentado o tipo de comunicação para este cenário de simulação. Pretendeu-se que cada um dos módulo ficassem isolados entre si, o que implicou o estudo e respetiva escolha de algumas tecnologias de comunicações sem fio. Neste caso, foram escolhidas as seguintes:

- **Bluetooth:** utilizado para a comunicação entre o Arduino (SM) e o Raspberry Pi 3 (CM). No Arduino foi utilizado um módulo Bluetooth HC-06 e no Raspberry Pi 3 foi utilizado o seu próprio módulo interno.
- **Wifi:** utilizado para a comunicação entre o Raspberry Pi 3 (CM) e o servidor.

O esquema da figura 4.17 pretende esquematizar os tipos de comunicação envolvidos nesta simulação para cada um dos componentes.

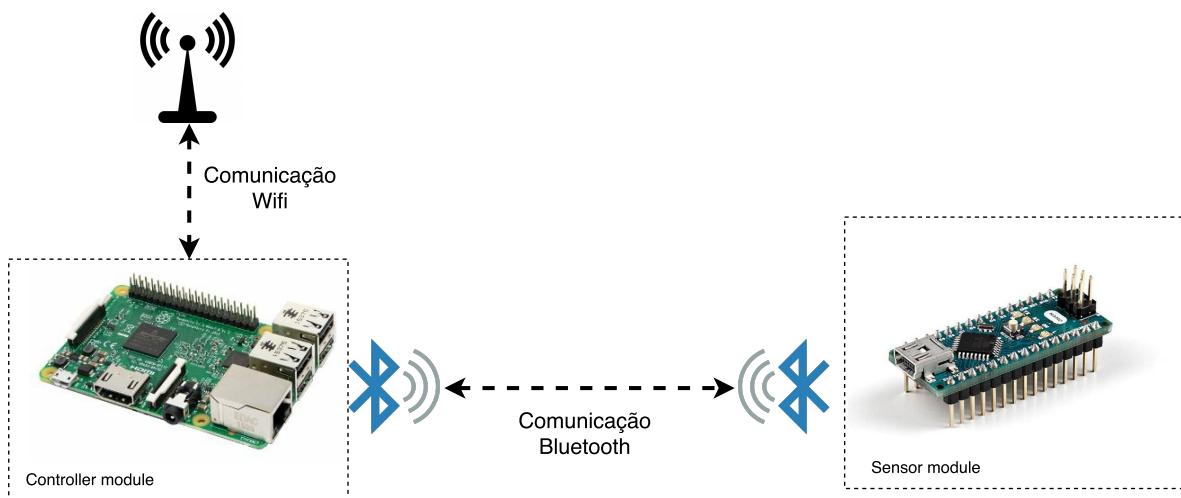


Figura 4.17: Arquitetura lógica

### Módulo bluetooth HC-06

Este módulo bluetooth HC-05 oferece uma forma fácil e barata de comunicação com seu projeto Arduino. Diferente do modelo HC-06, suporta tanto o modo mestre como escravo, além de ter uma fácil configuração.

Em sua placa existe um regulador de tensão e você poderá alimentar com 3.3 a 5v, bem como um LED que indica se o módulo está pareado com outro dispositivo. Possui alcance de até 10m.

é mais uma forma simples e barata de enviar e receber informações remotamente.



Figura 4.18: Flower one.

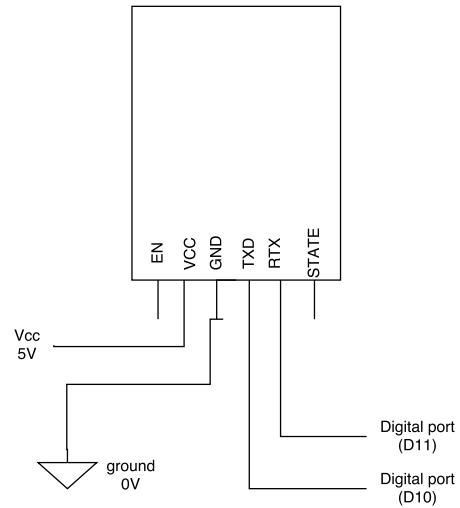


Figura 4.19: Esquema eletrotécnico da ligação do módulo bluetooth

Diâmetro	5mm
Protocolo Bluetooth	v2.0+EDR
Frequência	2,4GHz Banda ISM
Segurança	Autentificação e Encriptação
Tensão	3,3v (2,7-4.2v)
Alcance	10m
Dimensões	26,9 x 13 x 2,2mm
Peso	9,6g
Preço	32€/unidade

Tabela 4.5: Características do sensor GL5528 [20]

## 4.7 Diagrama de componentes

## 4.8 Considerações finais



# 5

# Implementação

Neste capítulo será apresentado ....

kkkkkkkkkkDurante o desenvolvimento um software pressupõe-se que os seus intervenientes sigam determinadas metodologia para o seu programa possa revolucionar a vida de um grupo em específico ou até mesmo da sociedade. Durante o desenvolvimento um software pressupõe-se que os seus intervenientes sigam determinadas metodologia para o seu programa possa revolucionar a vida de um grupo em específico ou até mesmo da sociedade. kkkkkkkkkk

## 5.1 Sistema de informação

### 5.1.1 Aplicação web

### 5.1.2 Aplicação mobile

### 5.1.3 *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 bitsRAM 2GB

## 5.2 Simulação em hardware

Nesta secção pretende-se explicar a implementação a nível de *software* no contexto desta simulação para cada um dos micro-controladores.

### 5.2.1 Arduino

No que diz respeito ao Arduino Nano (SM), numa fase inicial, procedeu-se à ligação dos diversos componentes anteriormente apresentados na *breadboard* tal como se encontra apresentado no Anexo G. Para auxiliar o desenvolvimento de *software* foi utilizada a versão 1.8.1 do Integrated Development Environment (IDE) do próprio Arduino<sup>1</sup>.

Seguidamente apresenta-se a implementação necessária a nível de sensores e de comunicação.

### 5.2.2 Sensores

Foram desenvolvidos os seguintes métodos que permitem aceder aos valores lidos de cada um dos sensores. Para além disso, foi criado um método que permite alterar o estado da válvula para transferência de água.

- `int readTemperature(int port)`: é efetuada uma leitura no porto analógico. Após a leitura este é convertido para °C (graus Celsius)
- `long readLuminosity(int port)`:
- `int readWaterValve(int port)`: é efetuada uma leitura no porto digital através do método `digitalRead`.
- `int readWaterLevel(int port)`: é efetuada uma leitura no porto digital através do método `digitalRead`.
- `void setWaterValve(int port, int state)`: se a variável `state` for 1 então o porto é colocado a HIGH (1) através do método `digitalWrite`, caso contrário é colocado a LOW (0)

Inicialmente procedeu-se à leitura de cada sensor de forma individual de modo a garantir o seu total funcionamento. Sempre que é feita um pedido de leitura dos sensores pelo CM os valores são enviados com o seguinte formato:

`<temperatura>;<nível_água>;<luminosidade>;<estado_válvula>` (5.1)

---

<sup>1</sup><https://www.arduino.cc/en/Main/Software>

### 5.2.3 Comunicação

Numa primeira fase procedeu-se à comunicação entre o SM e CM através de porta série. Seguidamente resolveu-se incorporar o módulo bluetooth de modo a tornar cada módulo independente. De módulo de interagir com o módulo bluetooth utilizou-se o package `SoftwareSerial.h` disponível no Arduino. Decidiu-se que caso o módulo bluetooth recebesse valores de 0 a 2 tinha diferentes comportamentos:

- **0:** ativação (ligar) da válvula;
- **1:** desativação (desligar) da válvula;
- **2:** recebe dados obtidos pelos sensores no formato definido em (5.1)

Antes de proceder à implementação de envio e receção de dados por bluetooth no Raspberry Pi 3 optou-se por testar este mecanismo através de uma aplicação Android existente na *Play Store* chamada de *Bluetooth Terminal HC-05*<sup>2</sup>

#### Raspberry Pi

##### Comunicação

Como é possível observar na figura 4.17, para a comunicação no Raspberry Pi (CM) entre o Arduino (SM) foi utilizado o modulo interno de bluetooth 4.1 que este incorpora no seu hardware. Para tal, foi desenvolvido um *script* em Python que permite o seguinte:

1. Verificar dispositivos bluetooth disponíveis
2. Estabelecer conexão com módulo HC-06 através de um socket para comunicação utilizando para isso o package `socket` do python.
3. Aceder à API para verificar estado da válvula de admissão de águas e enviá-lo através do socket utilizando o método `send()`
4. No caso se ser enviado o dígito 1 a válvula será aberta, enquanto que se for enviado o dígito 2 a válvula é fechada.
5. No caso de ser enviado o dígito 2, o socket ficará à espera de receber os dados lidos pelos sensores, utilizando para isso o método `recv()`
6. Após receber os dados lidos, é efetuado algum processamento para que os dados sejam enviados através da API.

---

<sup>2</sup><https://play.google.com/store/apps/details?id=project.bluetoothterminal>

7. Todos os pontos 3 a 6 são repetidos com um atraso igual ao seding time definido o sensor module na dashboard.

Para permitir o acesso aos recursos do sistema Bluetooth foi utilizada uma extensão (*package*) do Python denominado de *pybluez*<sup>3</sup>.

#### 5.2.4 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? nao havia orçamento..

### 5.3 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 etc).

Neste capítulo é descrita a tecnologia de processamento de imagem utilizada tal como o algoritmo disponibilizado pela mesma. Apresenta-se a implementação deste mecanismo e os testes necessários.

#### 5.3.1 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 5.1). 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*.

---

<sup>3</sup><https://github.com/karulis/pybluez>

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 [21].

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[21].



Figura 5.1: Logótipo OpenCV

### 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 5.2) permitirá a aquisição de imagem e servirá também como *controller module* ao sistema de aquisição de dados.

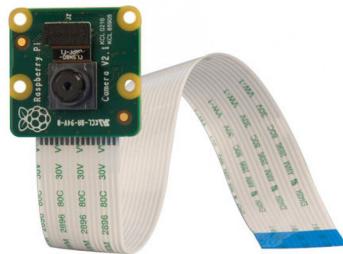


Figura 5.2: Raspberry Pi Camera Board V2 8MP 1080p

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 câmara Raspberry Pi<sup>4</sup>, 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.

### 5.3.2 Algoritmos 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/>

---

<sup>4</sup><http://picamera.readthedocs.io/en/release-1.13/>

Dataset:

é usado um detector HOG juntamente com um classificador linear SVM  
parametros do método detectMultiScale do opencv

- `img`: parâmetro obrigatório.
- `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`.

### 5.3.3 Testes

Foram considerados 4 frames de imagens .... e no apêndice X

### 5.3.4 Implementação

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.

## 5.4 Considerações finais



# 6

## Resultados

- 6.1 Aplicação web
- 6.2 Aplicação mobile
- 6.3 Testes API
- 6.4 Simulação em hardware
- 6.5 Sistema de deteção de intrusos



# 7

## **Conclusão e trabalho futuro**

**7.1 Conclusão**

**7.2 Trabalho futuro**

**7.3 Considerações finais**



# Bibliografia

- [1] 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.
- [2] P. T. Hiep, H. Noi, V. Nam, N. H. Hoang, H. Noi, and V. Nam, “A Review of Open Source Software Development Life Cycle Models,” *International Journal of Multimedia and Ubiquitous Engineering*, vol. 9, no. 5, pp. 391–402, 2014. [Online]. Available: <http://dx.doi.org/10.14257/ijseia.2014.8.3.38>
- [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] T. Our, “Resume : Context Aware Computing for The Internet of Things : A Survey Article 2013,” pp. 1–5, 2013.
- [16] 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>
- [17] The PostgreSQL Global Development Group, “PostgreSQL: About,” 2012. [Online]. Available: <https://www.postgresql.org/about/> [Accessed: 2017-05-29]
- [18] 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>
- [19] “Datasheet, NTC Thermistor TTC05 Series, Disc Type for Temperature Sensing/Compensation.” [Online]. Available: <http://extra-parts.com/datasheets/TTC.pdf>
- [20] L. LIDA OPTICAL&ELECTRONIC CO., “Datasheet, CdS Photoconductive cells, GL5528,” p. 1. [Online]. Available: <https://pi.gate.ac.uk/pages/airpi-files/PD0001.pdf> [Accessed: 2017-05-24]

- [21] 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:

<b>winStride</b>	<b>padding</b>	<b>scale</b>	<b>detection (number)</b>	<b>execution time (seg)</b>
(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

<b>winStride</b>	<b>padding</b>	<b>scale</b>	<b>detection (number)</b>	<b>execution time (seg)</b>
(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

<b>winStride</b>	<b>padding</b>	<b>scale</b>	<b>detection (number)</b>	<b>execution time (seg)</b>
(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

<b>winStride</b>	<b>padding</b>	<b>scale</b>	<b>detection (number)</b>	<b>execution time (seg)</b>
(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

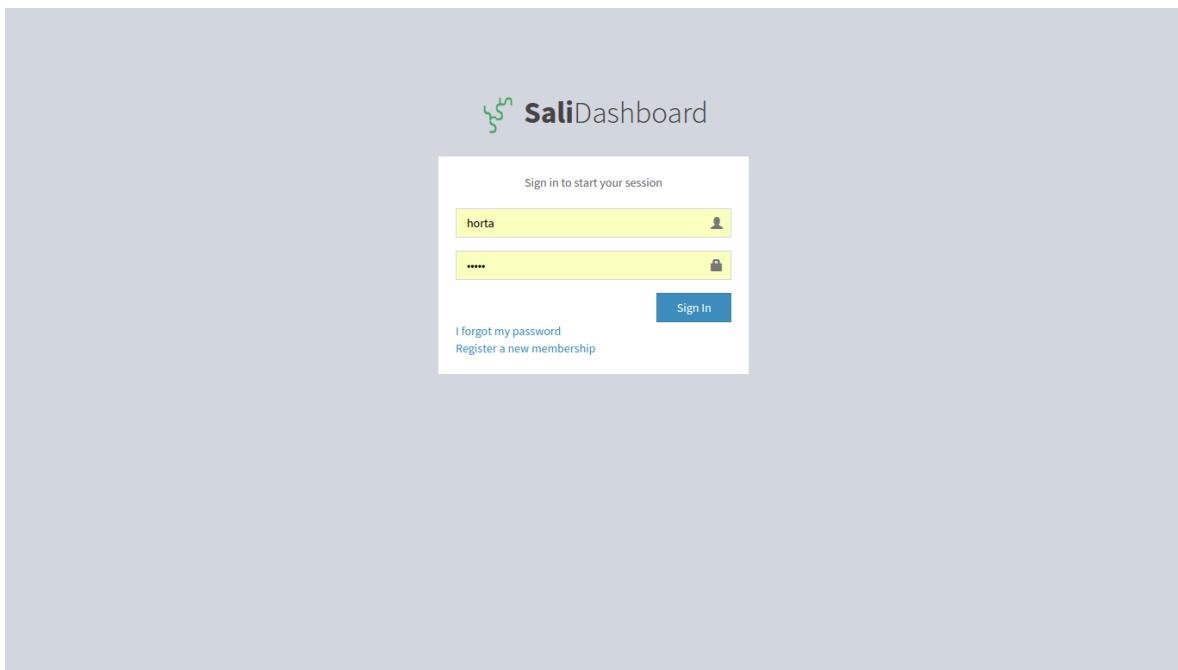


Figura E.1: Pirâmide do conhecimento: modelo DIKW

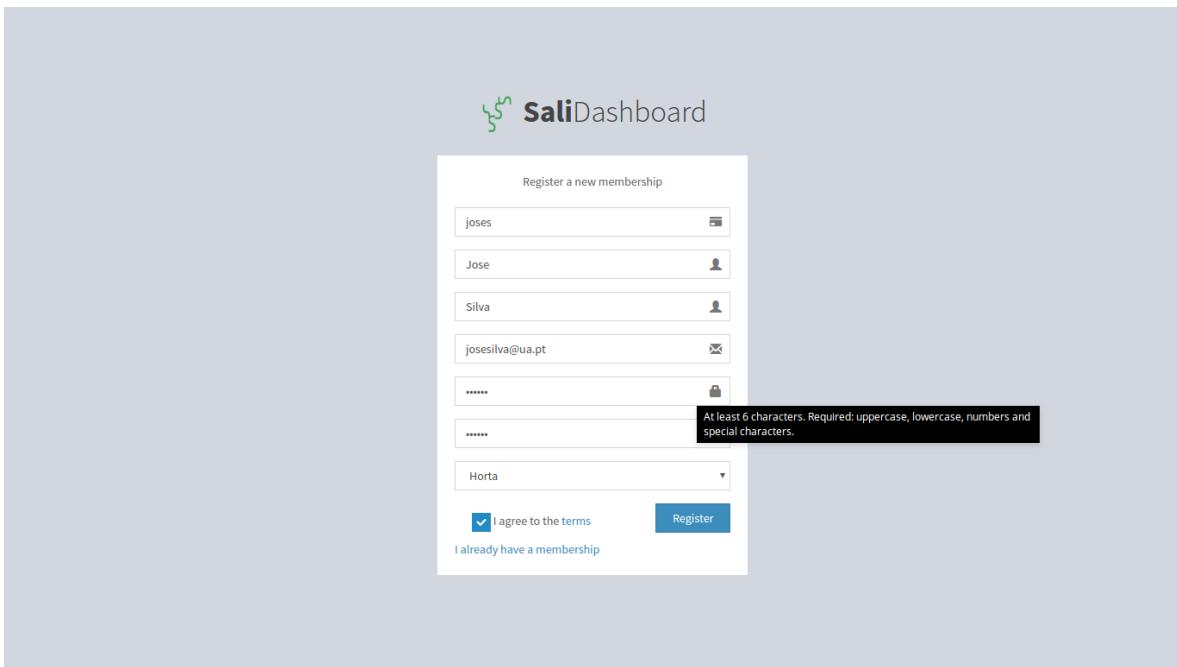


Figura E.2: Pirâmide do conhecimento: modelo DIKW

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

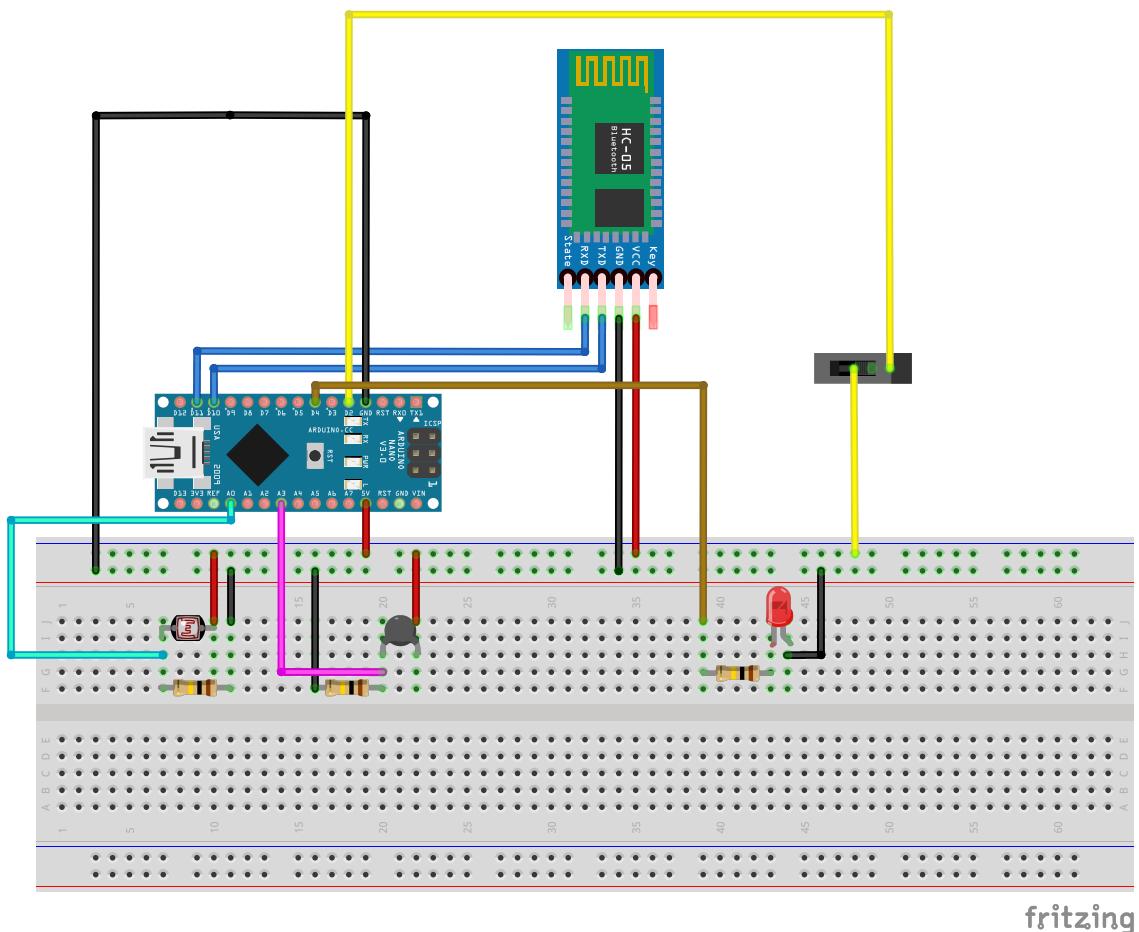


Figura G.1: Pirâmide do conhecimento: modelo DIKW

