



**Pedro Alexandre
Almeida Nunes**

Informatização da Indústria Turística



**Pedro Alexandre
Almeida Nunes**

Informatização da Indústria Turística

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestrado em Engenharia Mecânica, realizada sob orientação científica de José Paulo Oliveira Santos, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri / The jury

Presidente / President

Prof. Doutora Margarida Isabel Cabrita Marques Coelho
Professora Auxiliar da Universidade de Aveiro

Vogais / Committee

Prof. Doutor José Paulo Oliveira Santos
Professor Auxiliar da Universidade de Aveiro (orientador)

Prof. Doutora Ana Maria Pinto de Moura
Professora Auxiliar da Universidade de Aveiro (Arguente Principal)

Agradecimentos / Acknowledgements

Gostaria de agradecer em primeiro lugar à minha mãe, por ter estado sempre presente e me ter ajudado em tudo o que necessitei. Quero agradecer também à minha irmã e ao seu companheiro por me terem apoiado e por terem sido pilares fundamentais ao longo do meu percurso académico. Ao Professor Doutor José Paulo de Oliveira Santos, gostaria também de agradecer, por toda a paciência, acompanhamento e transmissão de conhecimentos no decorrer da realização desta dissertação. Por fim, gostaria de agradecer a todos que de alguma forma me apoiaram durante este percurso.

Palavras-chave

Controlo de Acessos; Base de Dados; Microcontrolador; Internet das Coisas; IoT; Detecção de Ocupação; Algoritmos;

Resumo

Esta dissertação tem como principal objetivo a criação de um sistema que permita automatizar e informatizar processos no setor de alojamento turístico, com vista a reduzir os custos com a receção.

O sistema proposto pretende tornar todo o processo de entrada num espaço alugado totalmente automático. Desta forma a solução proposta engloba uma interface WEB, que pode ser acedida em qualquer parte do mundo através de um *browser*, bem como um módulo de controlo de acessos e um módulo de deteção de presenças que deverão estar presentes em cada um dos espaços a alugar.

O módulo de controlo de acessos é composto por um microcontrolador ESP32 e uma rede de sensores e outros periféricos. O controlo de acessos é feito mediante a introdução de um código válido num teclado alfanumérico situado no exterior do espaço a aceder. Este tem uma senha diferente para cada período de reserva, sendo cada uma delas válida apenas nesse período. Além disso, existe uma senha global que pode ser utilizada pelo proprietário do espaço e pelos funcionários de modo a que estes possam ter acesso a qualquer altura, exceto quando o espaço estiver a ser ocupado por clientes, sendo que a porta possui um mecanismo que a permite trancar mecanicamente a partir do seu interior, garantindo maior privacidade ao seu utilizador. Este módulo comunica com um servidor central através da rede Internet, de modo a inserir os valores dos sensores numa base de dados e a receber informações da mesma, para saber qual a senha de acesso que está em vigor. A rede de sensores utilizada não é intrusiva para o utilizador e serve para que o proprietário possa monitorizar o espaço que esta a alugar. Desta forma este módulo comporta-se como um sistema IoT, pois permite uma monitorização em tempo quase real do ambiente em que está inserido. A plataforma WEB possui um sistema de *login* para clientes e administrador, sendo que qualquer pessoa pode criar uma conta cliente. Desta maneira o cliente pode procurar quartos disponíveis, efetuar a sua reserva e pagar *online* por *PayPal*, uma vez que foi feita a integração desta plataforma na aplicação criada. O administrador pode monitorizar todos os acessos, bem como os valores de todos os sensores, entre outros, na sua página. Assim a interface WEB criada consiste no site do serviço turístico, que comunica com a base de dados do sistema, devidamente normalizada.

Dado que se pretende prescindir dos serviços de funcionários de receção, tornou-se necessário perceber se um dado espaço se encontra ocupado. Nesse âmbito é proposto um módulo adicional para deteção de presenças, baseado na análise sonora de um espaço. Para tal é aplicado o algoritmo FFT ao sinal de um sensor de som ligado a um microcontrolador ESP8266, de modo a ser criado um novo parâmetro baseado nas frequências mais significativas e respetivas amplitudes do sinal. No servidor central são utilizados algoritmos de classificação que determinam a ocupação de um espaço com base no padrão da nova grandeza obtida.

Keywords

Access Control; Database; Internet of Things; IoT; microcontroller; Occupancy Detection; Algorithms;

Abstract

The main goal of this dissertation is the the creation of a system that allows to automate processes in the sector of tourist accommodation, in order to reduce the costs with the reception.

The proposed system aims to make the whole process of entry in a leased space a fully automatic. Thus the proposed solution includes a web interface which can be accessed anywhere in the world through a browser, as well as an access control module and a occupancy detection module that should be present in each one of the spaces to rent.

The access control module is composed by a microcontroller ESP32 and a sensor network and other peripherals. The access control is done by entering a valid code on an alphanumeric keypad located outside the space to be accessed. This has a different code for each period of time, this is, a code is only valid between check-in and check-out dates. Besides that, there is a global password that can be used by the owner of the space and by the employees so that they can access it at any time, except when the space is being occupied by clients. It should be noted that the door has a mechanism that allows it to lock mechanically from the inside, guaranteeing more privacy to users. This module communicates with a central server through the Internet network, in order to send the values of the sensors to a database and to receive information from it, in order to know which access password is valid at that time. The sensor network used is not intrusive to the user and is used by the owner to monitor the space he is renting. In this way this module behaves like an IoT system, because it allows an almost real time monitoring of the environment in which it is inserted.

The WEB platform includes a login system for clients and administrator. Besides that, anyone can create a client account. In this way the customer can search rooms, make his reservation and pay online through PayPal, once the integration of that platform in the created application was made. The administrator can monitor all accesses, as well as the values of all sensors, among others, on his page. Thus the WEB interface created consists of the site of the tourist service, which communicates with the system's database, properly normalized.

As it is intended to dispense the services of reception staff, it became necessary to realize if a given space is occupied. In this context an additional module is proposed for occupancy detection. It is based on the sound analysis of a space. To this end, the FFT algorithm is applied to the signal of a sound sensor connected to an ESP8266 microcontroller, in order to create a new parameter based on the most significant frequencies and respective signal amplitudes. In the central server are used classification algorithms that determine the occupation of a space based on patterns of the new parameter obtained

Conteúdo

1	Introdução	1
1.1	Contexto	1
1.2	Problema a resolver e sua importância	5
1.3	Resultados esperados	6
1.4	Organização da tese	8
2	Estado da arte	11
2.1	Conceitos	11
2.1.1	Controlo de acessos	11
2.1.2	IoT	12
2.1.3	Algoritmos	14
2.1.4	<i>PayPal</i>	19
2.2	Tecnologias	21
2.2.1	Chaves físicas	21
2.2.2	Códigos de barras	22
2.2.3	Códigos QR (<i>Quick Response</i>)	23
2.2.4	RFID (<i>Radio Frequency IDentification</i>)	25
2.2.5	NFC (<i>Near Field Communication</i>)	26
2.2.6	Biometria	29
2.2.7	Teclados numéricos ou alfanuméricos	31
2.3	Soluções académicas propostas por outros	31
2.4	Soluções comerciais atuais	42
3	Proposta de uma nova solução	45
3.1	Requisitos do sistema	46
3.2	Arquitetura da solução proposta	48
3.3	Projeto da base de dados	55
4	Implementação da solução proposta	65
4.1	Arquitetura da solução implementada	65
4.1.1	Comunicação entre fechadura e servidor	67
4.2	<i>Hardware</i>	69
4.2.1	Microcontroladores utilizados	69
4.2.2	Teclado alfanumérico	72
4.2.3	Comunicação com IO <i>expander</i>	73
4.2.4	Módulo RTC	76
4.2.5	Testa com fecho elétrico	79

4.2.6	Sensores utilizados	79
4.2.7	Placas PCB	85
4.3	<i>Software</i>	87
4.3.1	Servidor <i>Web</i>	88
4.3.2	Módulo de controlo de acessos	89
4.3.3	Interface <i>Web</i> desenvolvida	92
4.3.4	Método de pagamento via PayPal	96
4.4	Deteção de presenças	98
4.5	Montagens finais	113
5	Conclusões e trabalho futuro	117
A	Esquemas elétricos das placas PCB	129
B	Páginas da interface <i>Web</i> criada	133
C	Criação de um ambiente de testes <i>Sandbox</i>	143

Lista de Tabelas

3.1	Dados dos clientes	59
3.2	Dados relativos a pagamentos	60
3.3	Dados relativos a cada quarto	60
3.4	Preços por noite	61
3.5	Histórico de reservas	62
3.6	Histórico dos valores dos sensores do ESP32	63
3.7	Histórico dos valores do sensor de som e determinação de ocupação	63
4.1	Mensagens trocadas (formato decimal)	76
4.2	Desempenho dos classificadores	111

Lista de Figuras

1.1	Receitas com exportações (milhares de milhões de dólares) a nível mundial por setor (ano 2015) [2]	2
1.2	Total de chegadas de turistas	3
1.3	Chegadas de turistas em 2016 por continente [3]	3
1.4	Evolução das dormidas em Portugal 2010-2017 [6]	4
1.5	Balança Turística Portuguesa Viagens e Turismo, 2012-2016 [7]	4
1.6	Distribuição das dormidas em 2016 em Portugal por setor	6
1.7	Esquema da solução esperada	7
1.8	Esquema de monitorização de ocupação previsto	8
2.1	Crescimento previsto de sistemas IoT [13]	14
2.2	Exemplo de diferentes curvas de separação	16
2.3	Exemplo de uma árvore de decisão	17
2.4	Esquema de um neurónio de uma RNA [25]	19
2.5	Serviço PDT [28]	21
2.6	Chaves metálicas	21
2.7	Processo de leitura de um código de barras [29]	22
2.8	Exemplo de um código de barras segundo a norma EAN-13	23
2.9	Exemplos de códigos QR	25
2.10	Componentes RFID	26
2.11	Tecnologia NFC integrada em <i>smartphone</i>	27
2.12	Diferentes utilizações da tecnologia NFC	27
2.13	Diferentes formas de identificação por biometria	29
2.14	Teclado numérico para controlo de acessos [44]	31
2.15	Esquemas do modelo proposto por [46]	33
2.16	Modelo de autorização de acessos proposto pelo autor [47]	34
2.17	Implementação do modelo proposto pelo autor [47]	34
2.18	Esquema da solução proposta por [48]	35
2.19	Esquema de ligações [48]	36
2.20	Arquitetura do sistema proposto [50]	37
2.21	Esquema de arquitetura proposta por [51]	38
2.22	Sensores utilizados na solução proposta por [58]	41
2.23	Soluções da empresa "Triplo W"	43
2.24	Soluções da empresa "Chaves do Areeiro"	44
3.1	Mapa de ocupação de um hotel [67]	48
3.2	Esquema da nova solução proposta	50

3.3	Esquema de comunicação para efetuar o registo no sistema	50
3.4	Mapa de interação dos intervenientes para efetuar uma reserva	51
3.5	Mapa de interação dos intervenientes para efetuar o <i>check-in</i>	52
3.6	Esquema do módulo de deteção de ocupação	53
3.7	Esquema dos intervenientes no processo de determinação de ocupação e monitorização do ambiente de um espaço	54
3.8	Mapa de interação dos intervenientes para o proprietário efetuar a monitorização	54
3.9	Listagem de atributos necessários na base de dados	57
3.10	Relação universal para a base de dados criada	58
3.11	Diagrama de dependências funcionais da relação R1	59
3.12	Diagrama de dependências funcionais da relação R2	60
3.13	Diagrama de dependências funcionais da relação R3	61
3.14	Diagrama de dependências funcionais da relação R4	61
3.15	Diagrama de dependências funcionais da relação R5	62
3.16	Atributos relativos a dados dos sensores	63
4.1	Esquema de implementação da solução proposta	66
4.2	Composição da URI [69]	67
4.3	Esquema de comunicação cliente/servidor da solução proposta	68
4.4	Esquema de ligação de periféricos à placa ESP32	71
4.5	Esquema de pinos da placa ESP8266	72
4.6	Teclado alfanumérico utilizado [45]	72
4.7	Esquema de ligações do teclado utilizado [45]	74
4.8	Esquema dos pinos do PCF8574 [76]	74
4.9	Barramento <i>master-slave</i>	75
4.10	Estrutura de mensagem I2C para comunicar com o <i>IO expander</i> [76]	75
4.11	Esquema dos pinos do DS1302 [78]	77
4.12	Esquema dos pinos do DS1302 [79]	78
4.13	Testa convencional e testa com fecho elétrico	79
4.14	Resposta do sensor MG811 [81]	81
4.15	Placa com MG811 e circuito de condicionamento de sinal	82
4.16	Sensor de movimento PIR	83
4.17	Esquema elétrico de um sensor LDR	84
4.18	Princípio de funcionamento do sensor de porta	84
4.19	Esquema de implementação de um sensor de porta	85
4.20	Módulo sensor de som	85
4.21	Módulo a ficar no exterior da divisão	86
4.22	Placa a ficar no interior da divisão	87
4.23	Configuração do XAMPP para envio de emails	89
4.24	Fluxograma de tarefas a efetuar pelo ESP32	91
4.25	Fluxograma de procura de dados na tabela de histórico de reservas	92
4.26	Fluxo de navegação na interface para realizar uma reserva	93
4.27	Fluxo de interações com a API do <i>PayPal</i> [86] [Adaptado]	98
4.28	Redirecionamento para o ambiente <i>SandBox</i>	98
4.29	Representação de uma onda sinusoidal	101
4.30	Aproximação de uma onda quadrada por um somatório de funções sinusoidais	101
4.31	Frequências típicas da voz humana [87]	102

4.32	Intensidade sonora ao longo de um dia	104
4.33	Intensidade sonora ao longo de um dia sem qualquer ocupação	104
4.34	Exemplo de programação de algoritmos a partir de dados conhecidos	105
4.35	Separação de classes através de um hiperplano [24]	107
4.36	Desempenho dos algoritmos criados, quando testados num dia sem qualquer ocupação	111
4.37	Ocupação ao longo de 24h e nível sonoro correspondente	112
4.38	Montagem da solução proposta para controlo de acessos	113
4.39	Placa com módulo 8266, relés e circuito de alimentação	114
4.40	Montagem final do módulo de controlo de acessos	114
4.41	Montagem da solução em ambiente real	115
4.42	Montagem do módulo de deteção de presenças	115
A.1	Esquema elétrico da placa a ficar no exterior	129
A.2	Esquema elétrico da placa a ficar no interior	130
A.3	Alimentação e sensores	131
A.4	Ligações entre ESP32 e restantes periféricos	131
B.1	Página inicial	134
B.2	Página de registo no sistema	135
B.3	Página de <i>login</i> do cliente	135
B.4	Página de busca de quarto	136
B.5	Página de quartos disponíveis	137
B.6	Página para pagamento com <i>PayPal</i>	138
B.7	Página de confirmação de pagamento <i>PayPal</i>	138
B.8	Página de administrador	139
B.9	Página de monitorização do histórico de reservas	139
B.10	Página para alteração de uma reserva	139
B.11	Página para a eliminação de uma reserva	140
B.12	Página para a alteração de preços por noite	140
B.13	Página de monitorização dos dados dos clientes	140
B.14	Páginas de monitorização dos dados dos quartos	141
B.15	Página de monitorização do histórico de acessos	141
B.16	Página de monitorização do histórico de pagamentos	141
B.17	Página de monitorização dos sensores do módulo de controlo de acessos	142
B.18	Página de monitorização do sensor de som	142
B.19	Página de deteção de presenças	142
C.1	Contas <i>Sandbox</i> Criadas	144
C.2	Criação de uma aplicação	145
C.3	Credenciais geradas para comunicar com a API	145
C.4	Definição do comerciante	146
C.5	Definição dos detalhes do produto	146
C.6	Definição das páginas de redirecionamento	147
C.7	Página de redirecionamento onde será efetuado o pagamento	147
C.8	Confirmação de pagamento por parte do vendedor	147
C.9	Confirmação de pagamento por parte do comprador	148

Lista de Acrónimos

API	<i>Application Programming Interface</i>
AVAC	Aquecimento Ventilação e Ar Condicionado
CARET	<i>Classification and REgression Training</i>
CART	<i>Classification And Regression Trees</i>
DDF	Diagrama de Dependências Funcionais
EAN	<i>European Article Number</i>
FFT	<i>Fast Fourier Transform</i>
FNBC	Forma Normal de Boyce Codd
GBM	<i>Gradient Boosting Machine</i>
GPIO	<i>General Purpose Input/Output</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communications</i>
HTML	<i>HiperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
I2C	<i>Inter Integrated Circuit</i>
I2S	<i>Integrated Interchip Sound</i>
INE	Instituto Nacional de Estatística
IoT	<i>Internet Of Things</i>
IP	<i>Internet Protocol</i>
IPN	<i>Instant Payment Notification</i>
LCD	<i>Liquid Crystal Display</i>
LDA	<i>Linear Discrimination Analysis</i>
LDR	Light Dependent Resistor

MAC *Media Access Control*
NFC *Near Field Communication*
PCB Printed Circuit Board
PDT *Payment Data Transfer*
PIB Produto Interno Bruto
PIR *Passive Infra Red*
ppm *partes por milhão*
PWM *Pulse Width Modulation*
QR Quick Response
QR *Quick Response*
RF *Random Forests*
RFID *Radio Frequency IDentification*
RNA Redes Neurais Artificiais
ROM *Read only memory*
RTC *Real Timmer Clock*
SCL *Serial Clock*
SDA *Serial Data*
SMTP *Simple Mail Transfer Protocol*
SPI *Serial Peripheral Interface*
SQL *Structured Query Language*
SRAM *Static Random Access Memory*
SVM *Suport Vector Machine*
TCP *Transmission Control Protocol*
UEXT *Universal Extension Connector*
UPC *Universal Product Code*
UPS *Uninterruptible Power Supply*
URI *Uniform Resource Identifier*
URL *Uniform Resource Locator*
USB *Universal Serial Bus*
VAB Valor Acrescentado Bruto

Capítulo 1

Introdução

Este capítulo pretende apresentar o conjunto de motivações na origem da escolha do tema da informatização da indústria, mais especificamente no ramo do turismo. Esses motivos prendem-se sobretudo com a importância da indústria turística na atual economia e no contexto de aldeia global em que vivemos. Pretende-se nas secções seguintes fazer a contextualização do problema a resolver, bem como expor a sua importância na face à exigência competitiva de todas as indústrias na economia atual, fazendo desde já um prognóstico dos resultados que se esperam obter. Na última secção deste capítulo é apresentada a organização dos restantes capítulos deste documento, dando assim a entender ao leitor em traços gerais todos os assuntos que são abordados, para que este documento possa ser seguido com facilidade.

1.1 Contexto

Nos dias que correm, a deslocação de pessoas é bastante facilitada, muito por causa do elevado desenvolvimento tecnológico que temos ao nosso dispor. É muito fácil viajar para qualquer parte do mundo devido à grande proliferação e eficiência dos meios de transporte. Além disso, o aparecimento e disseminação de redes de comunicação como a *Internet* veio contribuir para que nos tenhamos aproximado do conceito de aldeia global, pois praticamente todos os sistemas estão ligados à rede e podem comunicar entre si, independentemente da distância que os separa. Deste modo é possível realizar um conjunto muito grande de tarefas *online* em qualquer parte do mundo.

A escolha pelo tema de informatização da indústria do turismo surgiu pelo facto de este ser um setor com um peso bastante importante na economia global, com tendência a crescer cada vez mais. Hoje, o volume de negócios do turismo é igual ou mesmo superior ao de algumas grandes fontes de rendimento económico a nível mundial, como são os casos dos produtos alimentícios ou automóveis, tendo-se tornado um dos principais fatores do comércio internacional [1]. Como se pode ver pela figura 1.1 apenas os lucros gerados pela exportação de petróleo e de produtos químicos conseguem ter um peso maior do que os lucros relacionados com o turismo, sendo que estes números se podem alterar, visto que a tendência é o aumento do número de turistas, que fará naturalmente os lucros com este tipo de mercado aumentarem. Contudo o crescimento de outros mercados não é tão previsível, como por exemplo no que toca à exportação de petróleo, uma vez que o aparecimento de novas energias alternativas tende a diminuir o consumo deste tipo de

combustíveis. É assim previsível que o setor turístico alcance futuramente um papel de relevo ainda maior do que aquele que possui atualmente.

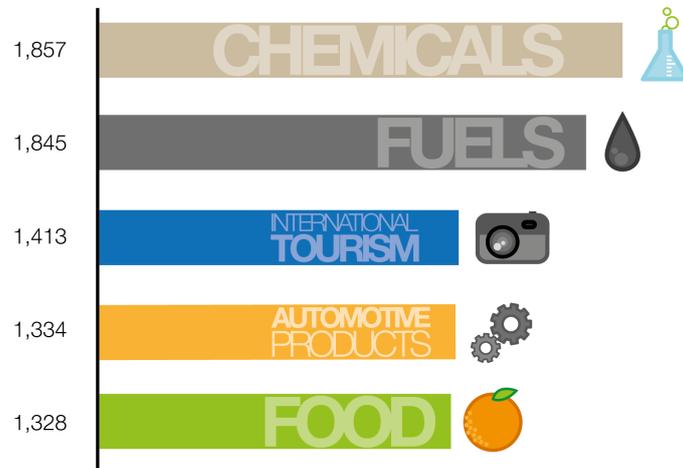
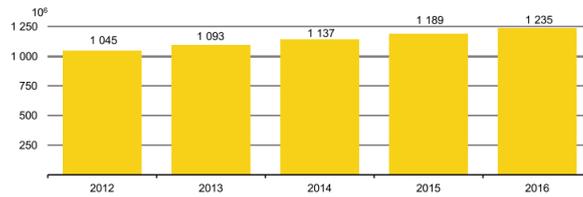


Figura 1.1: Receitas com exportações (milhares de milhões de dólares) a nível mundial por setor (ano 2015) [2]

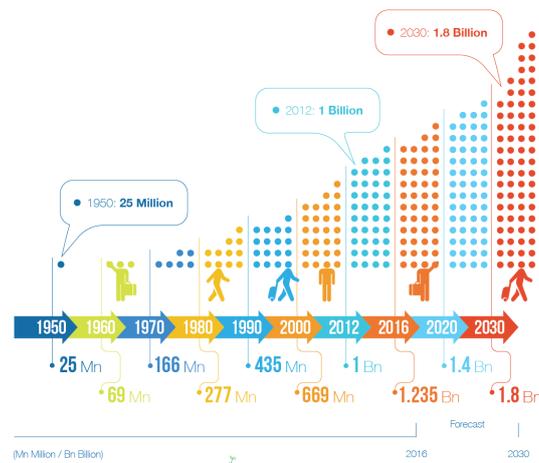
Para perceber a dimensão deste mercado, em 2016 ocorreram 1 235,2 milhões de chegadas de turistas internacionais em todo o Mundo, refletindo um crescimento de 3,9% face ao ano anterior [3]. De acordo com a mesma fonte, nesse mesmo ano as chegadas de turistas internacionais continuaram sua trajetória ascendente pelo sétimo ano consecutivo, registando ainda um crescimento acima da média, como se pode ver na figura 1.2 (a). De facto as previsões para o crescimento desta indústria são bastante animadoras, prevendo-se um total de chegadas a nível mundial de 1,8 mil milhões (figura 1.2 (b)), para o ano de 2030, fazendo com que este setor ganhe ainda mais protagonismo no que diz respeito ao desenvolvimento económico mundial.

No que toca à distribuição do número de chegadas de turistas pelos vários continentes, cerca de metade das chegadas são registadas na Europa, como mostra a figura 1.3, estando assim o nosso país localizado numa das zonas de maior afluência de turistas. De facto o turismo é uma das principais fontes de criação de riqueza do nosso país, sendo que em 2016 este setor gerou cerca de 11,5 mil milhões de euros, o que corresponde a mais de 7% do VAB gerado pelo conjunto da economia nesse ano [4]. De acordo com esta fonte, os turistas gastam em solo nacional cerca de 23 mil milhões de euros, correspondendo a cerca de 12% do PIB nacional. É importante ainda salientar que esse valor é gasto sobretudo em alojamento, transportes e restauração [5], ocupando assim o ramo dos alojamentos uma grande fatia do total das despesas dos turistas.

Para além da importância dos números anteriormente mencionados, a tendência é que este setor continue a crescer em Portugal, tal com acontece mundialmente. A figura 1.4 mostra que tem havido uma tendência de crescimento do número de dormidas ao longo do tempo. Pode verificar-se que a região onde existe o maior número de dormidas é o Algarve seguido de Lisboa, pela região Norte e pela Madeira. Nestas quatro regiões o crescimento do número de turistas sofreu aumento significativo nos últimos anos, sendo que em regiões como o Alentejo ou os Açores, esse crescimento não é tão notório, mas ainda assim a



(a) Número de chegadas de turistas a nível mundial 2012-2016 [3]



(b) Evolução do número de chegadas 1950-2030 (previsão) [2]

Figura 1.2: Total de chegadas de turistas

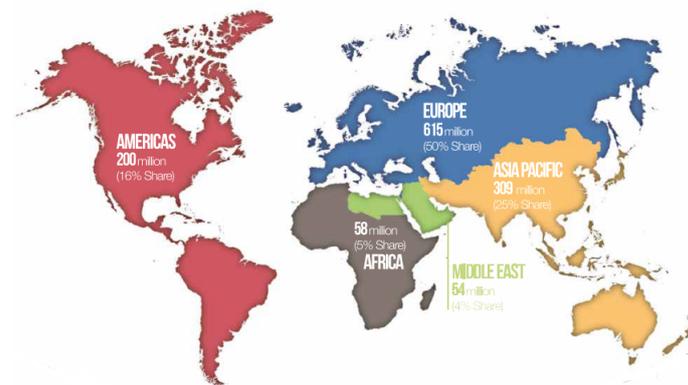


Figura 1.3: Chegadas de turistas em 2016 por continente [3]

tendência é o aumento do número de dormidas. Como resposta a este acréscimo de procura, em 2016 registou-se um aumento da oferta do número de alojamentos, sendo que o setor de alojamento turístico coletivo, na sua totalidade registou um crescimento de 5,2% na capacidade oferecida, aumentando em 5,7% o número de camas face ao ano de 2015 [4].

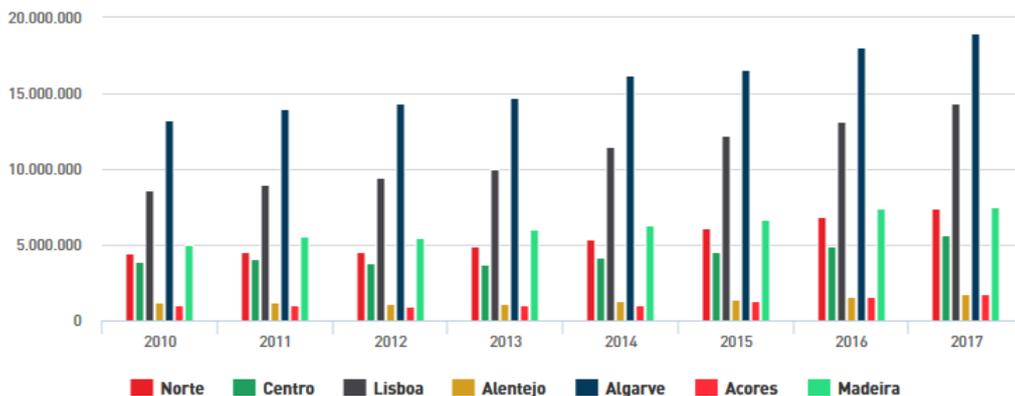


Figura 1.4: Evolução das dormidas em Portugal 2010-2017 [6]

A figura 1.5 representa a balança económica relativa ao turismo em Portugal. Como é visível, regista-se uma tendência clara de aumento das receitas, acompanhado de um aumento menos significativo das despesas, demonstrando assim que uma grande propensão para que se verifique um aumento do saldo desta balança, acentuando a importância deste setor em Portugal.

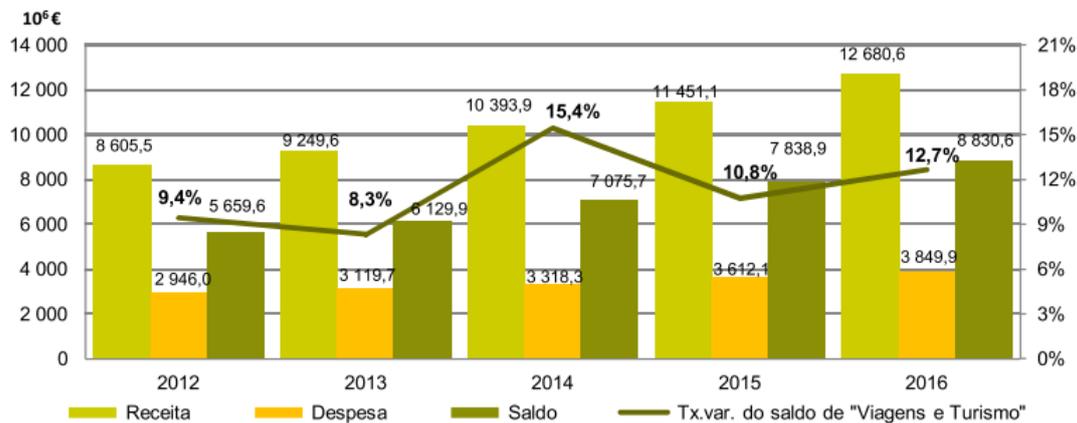


Figura 1.5: Balança Turística Portuguesa Viagens e Turismo, 2012-2016 [7]

No que diz respeito à comparação entre Portugal e outros países no que toca à qualidade do turismo, o nosso país encontrava-se em 2016 no 14^o lugar do *ranking* da *The Travel & Tourism Competitiveness* (subiu um lugar em relação a 2015 [7]), sendo que para essa classificação contribuiu em muito o nível de segurança do nosso país e sobretudo a qualidade das infraestruturas de serviços e turismo [8]. É por isso de esperar que

o investimento neste ramo continue, contribuindo o seu crescimento para a melhoria da economia nacional.

1.2 Problema a resolver e sua importância

A relevância da indústria turística é por demais evidente, quer a nível nacional quer a nível internacional, face aos dados expostos na secção anterior. Devido a essa crescente importância torna-se cada vez mais uma necessidade aumentar a eficiência de todos os processos envolvidos na hospedagem de um cliente. Importa assim mencionar um conjunto de serviços ou etapas fundamentais nesse processo. O primeiro serviço que deve ser disponibilizado é a divulgação *online* do espaço, para que um cliente em qualquer parte do mundo possa verificar as condições do mesmo através de fotografias e possa eventualmente efetuar a sua reserva remotamente através do seu computador, ou qualquer outro dispositivo que tenha um *browser* e ligação à rede *Internet*. Este serviço está amplamente divulgado nos dias que correm, sendo que a maior parte dos serviços de alojamento turístico já têm, ou estão inseridos numa plataforma de reservas *online*, onde os utilizadores podem ver as condições do espaço através de fotografias e efetuar a sua reserva, pelo que a grande falha que ainda existe está associada às fases de *check-in* e *check-out* do cliente.

Atualmente o que acontece é que para o cliente dar entrada no espaço que reservou, necessita de se identificar na receção de modo a que um rececionista introduza os seus dados no sistema (ou verifique os dados introduzidos pelo cliente no momento da reserva) e dessa forma é que é dado acesso ao cliente ao espaço por ele reservado. Normalmente este processo consiste em entregar ao mesmo as chaves do quarto ou casa que alugou.

O facto de haver sempre alguém na receção deste tipo de espaços acarreta custos elevados, especialmente se consideramos que para cobrir as 24 horas de um dia são necessários 3 turnos de 8 horas, o que para além dos custos financeiros associados representa uma diminuição no conforto do cliente, na medida em que este terá de dispor de algum tempo para realizar o seu *check-in*, não podendo dar entrada imediata no espaço por este reservado. É de realçar que os custos com a receção são ainda mais evidentes em pequenos espaços de alojamento, visto que o proprietário do espaço vai pagar por um serviço permanente e o número de clientes é relativamente baixo para fazer face a esse investimento, o que leva a que muitas das vezes o investimento neste tipo de espaços de menor dimensão não seja tão rentável quanto seria desejado. Em alguns estabelecimentos, com vista a reduzir os custos com a receção, as fases de *check-in* e *check-out* apenas se podem realizar entre determinadas horas, pois o serviço de receção não funciona 24 horas por dia obrigando o cliente a deslocar-se ao espaço alugado nesse período, ainda que este pretenda dar entrada no alojamento mais tarde, fazendo com que haja uma clara diminuição do conforto do mesmo. No que diz respeito ao aluguer de casas, o proprietário vê-se obrigado a deslocar-se ao local para que o cliente possa dar entrada no imóvel, tendo de repetir o processo na data de saída.

De acordo com dados do INE, em 2016 a grande fatia das dormidas de turistas em Portugal foi satisfeita pelo ramo da hotelaria (figura 1.6), podendo as fracas percentagens associadas ao turismo no espaço rural e de habitação e ao alojamento local estar associadas a um elevado custo com a receção, que face ao menor número de camas disponíveis deste tipo de alojamentos, tornam este género de negócio menos atrativo, em comparação

com grandes unidades hoteleiras.

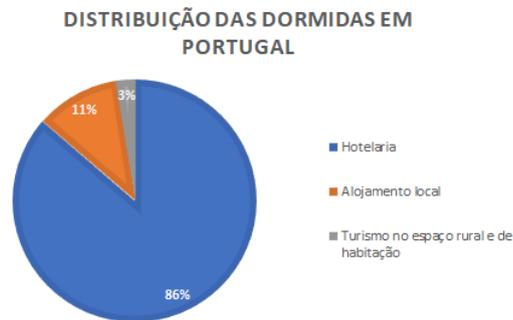


Figura 1.6: Distribuição das dormidas em 2016 em Portugal por setor

Seria assim de grande interesse criar um sistema que automatizasse o processo de *check-in* de forma a que este se tornasse mais eficiente, aliando a esse processo o facto de a reserva continuar a ser feita *online* e agregar-lhe uma forma de pagamento automática, de modo a tornar todo o processo o mais simples e cómodo possível, tanto na perspetiva do cliente, como na do proprietário.

Na perspetiva do proprietário seria também de grande interesse poder monitorizar os acessos realizados, bem como o ambiente envolvente de cada local a alugar, sem que com isso prejudicar o conforto e a privacidade dos utilizadores do espaço. Este tipo de monitorização permitiria ao proprietário ter a noção de se os clientes que efetuaram a reserva estão de facto a usufruir do serviço. Pretende-se que a gestão do espaço de alojamento possa ser feita à distância, prevenindo-se a ausência de um funcionário de receção. Desta maneira é de grande relevo para o proprietário perceber se o espaço que alugou está realmente a ser ocupado, ou se algo de anormal está a acontecer no mesmo, como uma porta que foi esquecida aberta, ou uma lâmpada que foi deixada acesa por engano, entre outros.

1.3 Resultados esperados

Os objetivos desta dissertação passam pela criação de um sistema que permita que as fases de *check-in* e *check-out* do cliente sejam feitas de forma automática. Pretende-se que este possa efetuar a sua reserva numa página *Web* onde poderá também ver as condições do espaço através de fotografias do mesmo e verificar se na data pretendida o espaço se encontra disponível, podendo dessa forma efetuar a sua reserva. A figura 1.7 mostra a sequência de funcionamento que se pretende, desde a escolha e reserva de um quarto, até ao pagamento *online* e entrada no mesmo.

O sistema deverá atribuir um código de acesso ao cliente, apenas válido durante o seu período de estadia, sendo que este se poderá identificar/autenticar no ato de entrada com essa palavra chave, sem necessitar da existência de terceiros para lhe fornecerem uma chave ou qualquer outro objeto que possibilite a entrada no espaço. Deverá ainda ser criada uma senha de acesso, válida em qualquer altura, para que proprietário e funcionários de limpeza possam entrar no espaço sempre com a mesma senha. Desta maneira sempre que houvesse uma reserva estariam válidas duas senhas, a do proprietário/funcionário

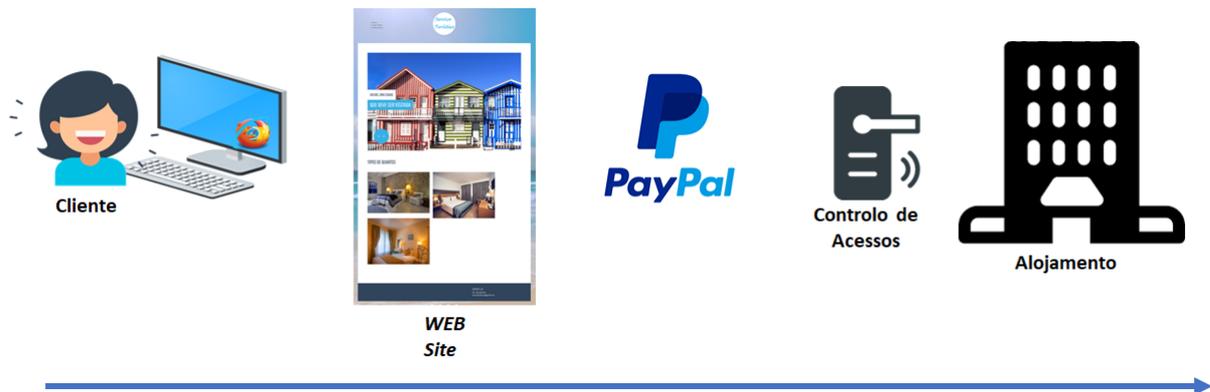


Figura 1.7: Esquema da solução esperada

e a do cliente que efetuou a reserva. O sistema deverá ser capaz de criar um histórico de acessos e tentativas de acesso, distinguindo se o acesso foi concedido pela senha de cliente ou pela do proprietário. Por questões de informatização do processo, todos os dados relevantes para o mesmo deverão ser registados em tabelas de uma base de dados devidamente normalizada.

A porta do espaço a aceder deverá incluir um circuito elétrico controlado por um microcontrolador embutido na mesma, de modo a que seja detetado através de um teclado alfanumérico ligado ao circuito se o código digitado está correto ou não. Este circuito elétrico prevê ainda a existência de um conjunto de sensores que serão utilizados para permitir que o proprietário possa monitorizar o local em questão (figura 1.8), sem com isso por em causa o conforto e privacidade do cliente. É também de interesse poder determinar de uma forma não intrusiva se o espaço que está a ser alugado está ocupado ou vazio. A determinação de ocupação de um espaço poderá permitir uma otimização do serviço de limpeza de quartos, na medida em que o funcionário pode ter conhecimento de quando o quarto está vazio. Estas de informações podem ser de facto muito importantes, pois poderão também permitir saber se o cliente saiu e se esqueceu da porta aberta ou da luz ligada. Este tipo de sistemas de deteção de presenças pode ser também utilizado para aferir se houve invasão do local, comparando datas de *check-in* e *check-out* e horários dos serviços de limpezas com a determinação de ocupação.

De modo a ter uma total independência de terceiros, pretende-se que este sistema esteja ligado a uma plataforma de pagamento *online*, de forma a aumentar a autonomia e a comodidade dos intervenientes no processo. Sendo assim pretende-se que ao fazer a reserva *online*, o cliente possa efetuar o pagamento da mesma maneira via *PayPal*.

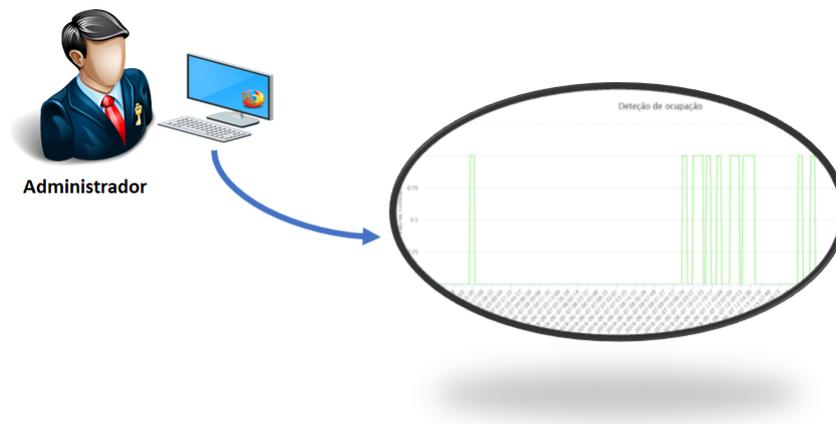


Figura 1.8: Esquema de monitorização de ocupação previsto

1.4 Organização da tese

Este capítulo serve apenas para melhor orientar a leitura deste documento, pretende-se aqui expor a estrutura do mesmo, dando linhas orientadoras acerca dos temas abordados em cada uma das partes. Para além deste capítulo introdutório, esta dissertação é constituída por mais 4 partes, as quais se descrevem sucintamente a seguir:

- **Capítulo 2:** Este capítulo está dividido em quatro secções. A primeira das quais tem um carácter mais teórico e pretende apresentar ao leitor um conjunto de conceitos essenciais para a compreensão de todos os assuntos que serão abordados posteriormente. As secções subsequentes consistem num levantamento de possíveis soluções para resolver o problema identificado na secção 1.3. Esse levantamento passa pela descrição de todas as tecnologias identificadas que podem ser implementadas, bem como de soluções comerciais, académicas ou patentes existentes, de modo a que o sistema apresentado traga vantagens face a outras soluções propostas.
- **Capítulo 3:** O capítulo 3 consiste na proposta de uma solução inovadora face aos problemas apresentados e às soluções já existentes. Aqui é explicada em traços gerais a arquitetura do sistema proposto e maneira como é feita a comunicação entre os diferentes dispositivos, de modo a que se possa ficar no final deste capítulo com uma boa perceção da solução que se pretende implementar.
- **Capítulo 4:** Neste capítulo são expostas todas as etapas da implementação da solução proposta. Pretende-se efetuar uma descrição detalhada de todo o *hardware* utilizado e desenvolvido, bem como do *software* utilizado para que sejam cumpridos todos os requisitos pretendidos.
- **Capítulo 5:** Este último capítulo é dedicado a algumas conclusões acerca de todo o processo que envolveu a criação do protótipo. Aqui são abordadas as melhorias resultantes da implementação deste modelo na indústria turística e são dadas algumas linhas orientadoras de alguns aspetos que poderão ser melhorados num trabalho futuro.

-
- **Apêndice A:** Consiste num conjunto de esquemas elétricos da solução proposta. Serão aqui explicados em detalhe os circuitos que deram origem às placas de circuito impresso realizadas no decorrer desta dissertação.
 - **Apêndice B:** Estão aqui representadas todas as páginas que integram a aplicação *Web* criada. Este apêndice deve ser consultado durante a leitura do capítulo 4.3.4, de modo a que possam ser compreendidas todas as explicações e funcionalidades do site criado.
 - **Apêndice C:** Consiste numa descrição detalhada de todos os passos seguidos para a criação de um ambiente de testes *PayPal Sandbox* que foi utilizado para testar a integração do da plataforma de pagamento no site criado.

Capítulo 2

Estado da arte

Este capítulo apresenta o estado da arte no que diz respeito ao controlo de acessos e à deteção de ocupação de um espaço. Numa fase inicial são abordados conceitos e conhecimentos adquiridos e necessários à compreensão do restante documento. Nas secções seguintes são apresentadas algumas tecnologias e soluções existentes. O principal objetivo deste capítulo é a escolha da melhor tecnologia para resolver o problema em questão, criando uma solução inovadora que possa agregar algumas mais valias de soluções já existentes.

2.1 Conceitos

Esta secção pretende dar a conhecer um conjunto de conceitos pertinentes no desenvolver desta dissertação. Aqui são apresentados os conceitos de controlo de acessos e suas etapas, o conceito IoT, na medida em que se pretende uma solução ligada à rede e que possa ser embutida. Para perceber o funcionamento de algumas soluções apresentadas no âmbito da deteção de ocupação, são descritos alguns algoritmos. Por fim é abordada a utilidade do serviço *PayPal*, como método de pagamento *online*.

2.1.1 Controlo de acessos

O controlo de acessos consiste na restrição de um conteúdo ou de um espaço a determinadas pessoas. No que diz respeito ao controlo de acessos físicos, pode dar-se o exemplo das residências particulares, às quais o acesso está destinado ao seu proprietário ou a outras pessoas que este autorize. Processo semelhante acontece na indústria turística. Um cliente reserva um determinado espaço, mas para poder ter acesso ao mesmo terá de possuir uma credencial de acesso para entrar no mesmo. Normalmente este tipo de controlo é feito mediante uma chave ou um cartão de acesso, que conferem a quem os possuir o direito de aceder a um determinado espaço. Importa referir que pode haver uma certa hierarquia no que diz respeito ao acesso de locais. Por exemplo num hotel, o proprietário terá à partida acesso a todos os locais, sendo que os clientes apenas terão acesso aos espaços por eles reservados durante o seu período de estadia. Pode assim dizer-se que neste caso o dono do espaço tem um estatuto de administrador e os clientes têm estatuto de utilizador.

Contudo o controlo de acessos não se limita apenas a espaços físicos. Este pode restringir um determinado conteúdo. Um exemplo disso é o caso dos computadores, que

podem ter mais do que um utilizador, estando os documentos de cada um dos mesmos protegidos das restantes pessoas, através de credenciais de *login* como nome de utilizador e palavra chave. Outros exemplos de controlo de acessos são a ligação a uma rede *Wi-Fi* com senha e utilizador, ou o acesso a conteúdos como por exemplo o email pessoal.

De forma a que todos os critérios de segurança sejam cumpridos e não haja intrusão de pessoas não autorizadas num determinado espaço ou conteúdo, há algumas etapas que devem ser cumpridas para garantir um controlo de acessos eficiente e sem falhas de segurança. Consideram-se então quatro etapas nessa fase, que são a identificação, a autenticação, a autorização e a auditoria [9], [10], que são expostas em seguida:

- **Identificação:** Partindo do princípio de que se existe um controlo de acessos é por que este é restrito a determinadas entidades, dessa forma é necessário proceder à identificação da pessoa que está a tentar aceder a determinado espaço ou conteúdo. Esta pode ser feita através de um nome de utilizador, de um número de identificação, entre outras.
- **Autenticação:** Depois de feita a identificação, a etapa subsequente consiste em validar a identidade do utilizador. Esta fase assume particular importância em casos em que seja fácil alguém possuir os dados de identificação de outra pessoa, como por exemplo numa conta de email em que várias pessoas possuem dados para a identificação, contudo apenas o proprietário da conta sabe a senha para entrar, funcionando esta como método de autenticação.
- **Autorização:** Determina se determinado espaço ou conteúdo estará à disposição do utilizador. Nesta etapa as credenciais introduzidas nas etapas anteriores vão ser comparadas com os dados existentes no sistema, de modo a aferir se aquele utilizador tem acesso ao conteúdo em questão e se os parâmetros de autenticação são válidos.
- **Auditoria:** Esta fase consiste na monitorização dos vários pedidos efetuados. Desta forma é possível obter uma maior segurança pelo facto de se ter conhecimento de tentativas de acesso não autorizadas. É possível também obter um histórico de acessos, de forma a perceber qual o último utilizador a efetuar o pedido de acesso permitindo desta forma uma avaliação do sistema de controlo de acessos.

2.1.2 IoT

A *Internet* das coisas (IoT) é um conceito tecnológico que foi falado pela primeira vez em 1999 por Kevin Ashton. Segundo este, um sistema IoT consiste em colocar objetos banais do dia a dia conectados à *Internet* e a um conjunto de sensores que lhes permitem agir de modo inteligente. De facto o termo consiste na ideia de fundir o mundo real com o mundo digital. Os seus objetivos são integrar qualquer dispositivo através da rede de computadores, conectando e automatizando uma série de tarefas do quotidiano [11] melhorado a resposta de sistemas em tempo real. Este conceito veio trazer resposta a um problema identificado pelo seu criador, que consistia no facto de os computadores serem cérebros sem sentidos [12], o que fazia com que os computadores apenas soubessem o que lhes era "ensinado", o que era uma grande limitação, pois a quantidade de afluência de informação nos dias que correm é demasiado grande, para ser escrita manualmente. Deste modo se os objetos estiverem ligados a um conjunto de sensores e estiverem a

enviar essa informação através da rede *Internet* é possível obter um grande conjunto de informações em tempo real.

A detecção habilitada pelas tecnologias da rede sem fios aliada a sensores é transversal a muitas áreas da vida moderna, sendo que nos dias que correm, é possível ver este tipo de sistemas em várias áreas [13]. Algumas das quais são evidenciadas em seguida:

- Indústria: Os fabricantes estão a utilizar IoT em duas vertentes distintas. Uma delas é a utilização no interior de fábrica para monitorizar equipamentos e realizar a sua manutenção preditiva/preventiva. No exterior das instalações este tipo de solução é utilizada sobretudo para examinar a maneira como os produtos são usados, permitindo uma recolha de dados que possibilita um ajuste dos produtos para a sua melhoria contínua.
- Transportes: Nos veículos de carga e de transporte público estão a ser utilizadas redes de sensores inteligentes para ajudar a agendar a manutenção e a otimizar consumo de combustível. Com a crescente automatização de veículos particulares e comerciais, prevê-se um grande crescimento na utilização de IoT neste setor.
- Cuidados médicos: Entre as várias aplicações neste ramo, encontram-se monitorizar e solucionar problemas com equipamentos médicos, rastrear implantes, próteses e a implementação em qualquer tipo de equipamento passível de ser vestido, para fornecer dados em tempo real aos médicos.
- Aplicações domésticas: Este tipo de abordagem começa a ser bastante utilizada para ajudar a tornar uma casa inteligente. Entre as aplicações mais utilizadas nesse âmbito encontram-se o controlo e monitorização de eletrodomésticos, sistemas de alerta de intrusão e sistemas de otimização de energia.

Com o elevado número de aplicações onde é possível aplicar a *Internet* das coisas e devido ao facto de a sua utilização contribuir para uma melhoria na qualidade de vida, automatização de tarefas e otimização de processos, o número de equipamentos deste tipo tem vindo a aumentar de forma exponencial nos últimos anos, como mostra a figura 2.1, sendo que se prevê que em 2020 o número total de dispositivos deste tipo chegue aos 50 mil milhões.

Este tipo de sistemas oferece, portanto a capacidade de medir, inferir e compreender indicadores ambientais nos mais diversos espaços. Essa informação pode ser utilizada posteriormente para inferir uma série de novos parâmetros adicionais, ou pode ser processada imediatamente e atuadores podem ser acionados mediante determinados valores, ficando desta maneira o sistema global habilitado a agir em conformidade com as condições ambientais sem que haja qualquer intervenção humana.

A proliferação destes dispositivos numa rede de atuação e comunicação cria a *Internet* das coisas (IoT)[14], em que os sensores e os atuadores se misturam perfeitamente com o ambiente que nos rodeia. Este conceito veio então contribuir para a automatização de muitos processos, possibilitando assim a comunicação entre vários dispositivos que podem ser monitorizados e controlados remotamente a partir de um dispositivo como um computador ou *smartphone*.

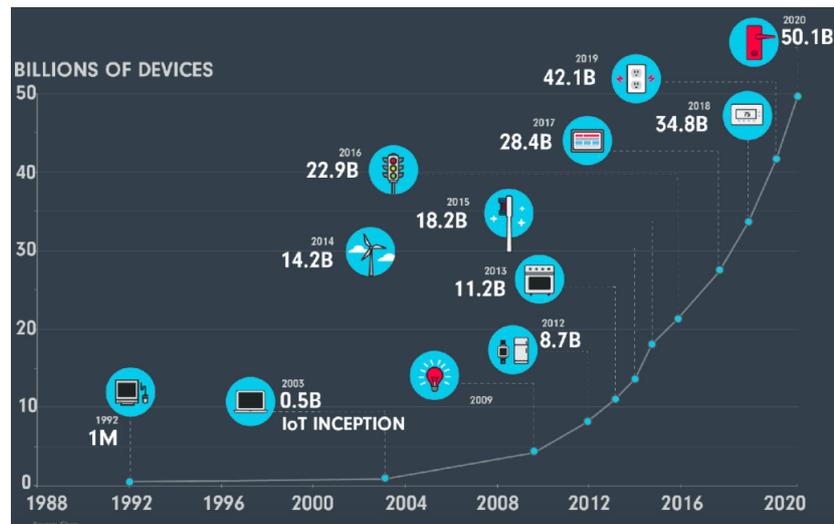


Figura 2.1: Crescimento previsto de sistemas IoT [13]

2.1.3 Algoritmos

Um algoritmo é uma sequência de instruções que diz a um computador o que fazer. Os computadores são constituídos por milhares de milhões de pequenos interruptores chamados transístores, e os algoritmos ligam-nos e desligam-nos milhares de vezes por segundo [15]. O algoritmo mais simples consiste em ligar um interruptor. O estado de um transístor é um *bit* de informação que pode ser "0", ou "1", consoante este esteja desligado ou ligado respetivamente. Este tipo de processos é cada vez mais utilizado para diversos fins, como realizar ações de forma automática, que de outra maneira teriam de ser realizadas por pessoas. Por exemplo, vamos supor que se uma determinada máquina atingir um certo valor de pressão deve ser desligada. Nestes casos em vez de ser o operador a desligar a máquina manualmente, é o próprio sistema que com base no algoritmo de programação responde ao estímulo desligando a máquina, sem necessidade de intervenção humana.

Existem no entanto problemas, cuja resolução passa pela criação de um modelo matemático que possa prever ou responder a determinado comportamento do sistema. Nesses casos é necessário determinar os melhores parâmetros para os modelos matemáticos que descrevem esse comportamento. Tal pode-se tornar uma tarefa bastante complicada, dada a quantidade de dados que pode estar envolvida. Nesses casos poderá ser importante utilizar o conceito de aprendizagem automática ou *machine learning* para construir um algoritmo que realize as ações pretendidas ou que seja capaz de prever determinado comportamento. Neste tipo de abordagem, o algoritmo é construído pelo próprio computador a partir dos dados que lhe são fornecidos, fazendo com que o processo de construção de um algoritmo seja feito de forma bastante mais expedita. Contudo para utilizar este tipo de procedimento é necessário ter um conjunto de dados de entrada, cujos dados de saída sejam conhecidos, para que desta forma o computador possa atribuir os parâmetros necessários ao modelo matemático, de modo a que se obtenha o algoritmo final. Por exemplo, suponhamos que se pretende saber se uma determinada máquina está a funcionar corretamente ou não com base na leitura de um conjunto relativamente grande

de sensores. Se cada um desses sensores por si só não oferecer informação suficiente para que seja retirada uma conclusão, é necessário analisar os diferentes valores em conjunto, o que pode levar a que haja padrões que podem ou não ser facilmente detetáveis para o ser humano, para os quais se determina se a máquina está a funcionar corretamente ou não. Com a aprendizagem automática, é necessário recolher um conjunto de valores para os quais se conheça o estado de funcionamento da máquina. Desta forma, entram no algoritmo parâmetros de entrada e parâmetros de saída correspondentes a cada conjunto de dados de entrada, para que de futuro o algoritmo seja capaz de prever anomalias de funcionamento com base em valores obtidos nessa primeira fase.

Como é de imaginar, a qualidade dos resultados obtidos seguindo esta abordagem está inteiramente dependente dos dados que são introduzidos no computador para a construção do algoritmo. Sendo que existem vários tipos algoritmos que podem ser construídos utilizando a aprendizagem automática, uma maneira de verificar o seu desempenho é testa-lo para os dados que se utilizaram na sua construção e verificar se os dados de saída são os esperados. Contudo este método de avaliação pode não ser o melhor, visto que apenas garante que o algoritmo "aprendeu" corretamente tudo aquilo que lhe foi ensinado, podendo os valores que foram introduzidos não serem suficientemente abrangentes para as possibilidades envolvidas. Convém assim ter um conjunto de dados conhecidos utilizados para ensinar o algoritmo e um outro conjunto que serve para testar a eficácia do mesmo. Caso o desempenho não seja satisfatório, há sempre a possibilidade de introduzir novos dados de aprendizagem, de modo a melhorar o desempenho, tornando assim este tipo de abordagem bastante flexível e fácil de ser melhorada/alterada.

Importa referir que os problemas a serem resolvidos por este tipo de modelos podem ser separados em problemas de regressão ou de classificação. Os primeiros verificam-se quando a resposta que se pretende obter é contínua, por exemplo se for pretendido inferir a massa gorda de um indivíduo indiretamente a partir de um conjunto de dados estamos perante um problema de regressão. Por outro lado se for pretendido determinar um certo valor a partir de parâmetros de entrada, mas esse valor só pode tomar um certo conjunto finito de possibilidades, (por exemplo determinar se um espaço está ocupado ou não com base na informação de diversos sensores) estamos perante um problema de classificação. Desta forma, num problema de classificação, denominam-se de classes o conjunto finito de possíveis parâmetros de saída para um conjunto de parâmetros de entrada. Por exemplo, se o problema for determinar se um espaço está vazio ou ocupado, as classes serão "vazio", e "ocupado", sendo que apenas pode ser atribuída uma classe a uma instância. Assim, uma instância designa um vetor de entrada (conjunto de valores dos vários sensores por exemplo) e cada elemento que compõe uma instância pode ser chamado de descritor.

Para uma melhor compreensão e avaliação do desempenho de diferentes algoritmos, importa compreender alguns problemas matemáticos associados a eles. Estes são chamados na literatura de *overfitting* e *underfitting*. Graficamente, o primeiro problema consiste na criação de uma linha de separação das diferentes classes que separa na totalidade todos os pontos que não pertençam à mesma, o que pode fazer com que o modelo obtido fuja à realidade, não se adaptando a novos dados. Por outro lado o problema de *underfitting* consiste no traçado de uma curva de separação demasiado grosseira face aos dados apresentados, o que faz com que haja um elevado erro associado a este tipo de aproximação. A figura 2.2 mostra à esquerda um caso de *underfitting*, no centro uma curva de aproximação suave e à direita um problema típico de *overfitting*. Tendo em conta

este tipo de problemas, reforça-se a necessidade de testar os algoritmos de aprendizagem automática com dados diferentes daqueles que foram utilizados na sua construção, pois se forem testados os mesmos dados utilizados para "ensinar" o algoritmo, podemos pensar que o modelo obtido funciona na perfeição, quando na verdade apenas está a haver um sobreajuste dos dados, não sendo necessariamente as previsões futuras ajustadas.

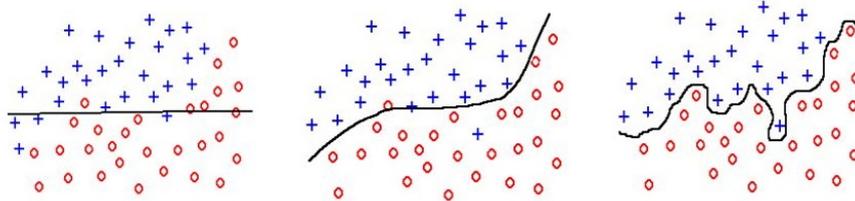


Figura 2.2: Exemplo de diferentes curvas de separação

Árvores de decisão: Este método é também conhecido na literatura como CART (*Classification And Regression Trees*) e consiste em dividir um problema complexo em vários subproblemas simples, sendo essa estratégia aplicada recursivamente [16]. Assim é criado um esquema idêntico a uma árvore, onde em cada nó se encontra uma decisão a ser tomada, cada ramo corresponde a um valor de uma instância e cada folha corresponde uma decisão ou classificação. Por exemplo, a figura 2.3 representa uma árvore de decisão bastante simples que deve ser lida de cima para baixo. Deve primeiro analisar-se a resposta à primeira instância (se está sol, nuvens ou chuva) e realizar o mesmo procedimento até chegar a uma folha com a classificação final, que neste caso contém a decisão de ir ou não jogar ténis.

Este modelo é bastante simples e intuitivo. Contudo quando o objetivo é utilizar uma filosofia de aprendizagem automática, para criar uma árvore de decisão, pode correr-se o risco da existência do problema de *overfitting* [17]. Isso acontece porque na génese deste tipo de algoritmo encontram-se uma espécie de regras que permitem percorrer a árvore de decisão do topo até à sua base. Na filosofia de aprendizagem automática, o computador vai tentar criar regras que se ajustem à totalidade dos dados. Em casos mais complexos, podem ser criadas uma grande quantidade de regras para ajustar os dados, o que leva à construção de uma árvore com dimensões consideráveis e que na maior parte das vezes tem um problema de sobreajuste dos dados. Nessas situações o algoritmo será ótimo quando testado nos dados que foram usados para o treinar ("ensinar"), contudo não se espera um grande desempenho para novos dados.

Florestas aleatórias: Também conhecidas como RF (*Random Forests*), são um tipo de algoritmo que se baseia na criação de um conjunto de árvores de decisão, para efetuar computacionalmente o treino de uma máquina, que deverá tomar uma decisão com base num número elevado de dados que podem ou não estar relacionados entre si. Contudo cada árvore depende de um conjunto de dados aleatórios analisados independentemente [18]. Por exemplo se o pretendido é desligar ou ligar uma máquina com base nos seus valores de temperatura e pressão, segundo esta lógica seria necessário construir uma árvore para cada parâmetro (pressão e temperatura), de modo a que cada árvore dependesse apenas da análise de valores da pressão ou da temperatura, sem olhar aos valores da outra grandeza.

Neste tipo de abordagem o erro de generalização tende a diminuir com o número de

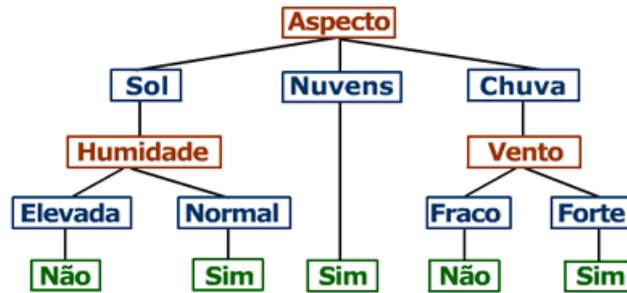


Figura 2.3: Exemplo de uma árvore de decisão

árvores de decisão que são utilizadas, até um certo número, onde ocorre a sua estabilização. O facto de o conjunto de dados utilizados para tomar uma decisão ser aleatório tem um contributo positivo para a generalização do algoritmo [17] fazendo com que este produza melhores resultados depois de efetuada a fase de treino. Os resultados estão assim dependentes do peso atribuído a cada árvore e da correlação existente entre os diversos tipos de dados.

Algoritmos de gradiente: Um dos mais conhecidos na literatura é o GBM (*Gradient Boosting Machine*) consiste num método de aprendizagem automática. Neste método é feita uma primeira aproximação com um modelo mais simples, como por exemplo uma árvore de decisão [19]. Na fase de treino para o qual os resultados são conhecidos é possível melhorar esta solução acrescentando termos adicionais que vão minimizando o residuo (diferença entre o resultado obtido pelo modelo em aprendizagem e o resultado real conhecido). Por exemplo imagine-se que x e y são duas grandezas e se pretende encontrar uma função F , tal que $F(x_n)=y_n$, para um conjunto de dados recolhido. Uma primeira aproximação poderá ser considerar uma reta que pode não ter exatamente a eficácia pretendida. Nesse caso podem ser adicionadas sub-funções simples de modo a diminuir o erro, ficando por exemplo $F(x_n)+h(x_n)=y_n$. De facto podem ser adicionados quantos termos forem necessários, até que o modelo na fase de teste se comporte com a precisão desejada. Estes resultados podem depois ser replicados em ensaios futuros, de modo a garantir a qualidade do modelo. [20]

Maquinas de Vetores de Suporte: Ou SVM (*Support Vector Machine*) é uma técnica robusta de classificação e regressão que maximiza a precisão preditiva de um modelo sem causar *overfitting* dos dados utilizados para o treino. Este método permite a separação de duas classes através de hiperplanos de separação, caso estas duas grandezas sejam linearmente separáveis. Caso não o sejam o algoritmo cria um espaço de alta dimensionalidade de forma a poder separar as grandezas [21].

Para perceber este algoritmo, podemos imaginar um espaço multidimensional, onde existem várias instâncias, que podem ser vistas como o conjunto de parâmetros utilizados de modo a permitir a classificação do conjunto de dados em causa (descritores). Neste tipo de método, a separação em classes é feita à custa de um hiperplano, que terá uma dimensão igual ao número de instâncias menos 1. Isto é, por exemplo, se estiverem a ser utilizados 4 valores de sensores para uma determinada classificação, o hiperplano terá dimensão 3 (3D), se fossem 3 valores diferentes de sensores, o este teria dimensão 2 (2D,reta) e assim sucessivamente. Podemos ver um hiperplano como a zona de separação

entre as duas classes. No caso de as instâncias escolhidas (tipo de dados) serem adequadas, este hiperplano separa as classes com alguma distância dos pontos de amostragem. A classificação de um novo conjunto de dados pode assim ser vista geometricamente como o ângulo que o ponto de amostragem faz com o plano que separa as variáveis. Este tipo de algoritmo pode ser melhorado continuamente, por exemplo removendo os exemplos que estão muito longe dos limites de decisão, ficando assim dependente apenas de um conjunto pequeno dos dados utilizados para "treinar" a máquina, que são designados de vetores de suporte [22].

Algoritmo Bayesiano ou de Naive: Este algoritmo consiste numa forma de classificação que deriva do teorema de Bayes (equação (2.1)) [23] [24]. Esta lei descreve a probabilidade de um evento, conhecendo previamente acontecimentos relacionados com esse evento cuja probabilidade se está a determinar. Desta forma, $P(c)$ é a probabilidade de uma determinada classe ou evento e $P(x)$ é a probabilidade de ocorrência de uma determinada instância para essa classe. $P(c|x)$ é a probabilidade posterior de uma classe, segundo determinada instância, ou seja é a probabilidade de acontecer o evento "c", sabendo que "x" acontece. Da mesma forma $P(x|c)$, pode ser lido como a probabilidade de ocorrer a instância "x" sabendo que ocorreu o evento "c".

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

(2.1)

De acordo com este algoritmo, todos os parâmetros utilizados para a classificação são independentes. Isto é, considera-se que a alteração de um dos parâmetros irá contribuir individualmente para o cálculo da probabilidade de ocorrência de um determinado parâmetro de saída, mesmo que a alteração de um dos parâmetros de entrada esteja dependente da alteração de outros.

Este método de classificação tem inerentes a si inúmeras vantagens, como o facto de ser bastante rápido e de precisar de menos dados para o seu treino, quando comparado com modelos de regressão logística. Além disso em caso de variáveis numéricas, é assumido que estas têm uma distribuição normal, o que é uma suposição forte [23]. Contudo existem algumas desvantagens na utilização deste método. Por exemplo, se for observado um determinado valor para uma variável que não está nos dados de treino (ou que difere bastante dos valores utilizados para treino), a sua probabilidade poderá ser determinada como nula, tendo nesses casos de se utilizar o artifício de obrigar essa probabilidade a tomar o valor de uma constante, para que o teorema de Bayes possa ser utilizado. Além disso, o pressuposto de que todos os descritores são independentes nem sempre é verdade, visto que na maior parte dos casos estes são dependentes uns dos outros.

RNA (Redes Neurais Artificiais): Este tipo de algoritmo consiste num modelo computacional que pretende imitar o cérebro humano. Para isso uma RNA é constituída por vários neurónios interligados. Estes modelos são utilizados em problemas complexos onde não é possível definir uma lista de regras, ou onde o ambiente de onde são recolhidos os dados esteja em constante mudança. Estes algoritmos têm a sua principal utilização na classificação de padrões e previsão.

De modo a perceber melhor este modelo, a figura 2.4 representa esquematicamente um neurónio de uma rede. Essa parte da rede é composta por vários elementos, entre

eles os "x" que são parâmetros de entrada do neurónio em questão, mas que podem ser também vistos como parâmetros de saída de neurónios anteriores. Os "w" são os pesos atribuídos a cada um dos parâmetros. Este tipo de algoritmo também é baseado no treino, com um conjunto de dados conhecidos, pelo que os pesos atribuídos são fruto da experiência adquirida pelo neurónio após a apresentação de um conjunto de dados. Uma combinação linear " U_K " é obtida a partir da soma ponderada de todos os pesos, que depois de passar pelo termo polarizador resulta no parâmetro de saída " Y_K " do neurónio. O termo polarizador pode ser visto como mais um peso, sendo que pode assumir os valores +1 ou -1.

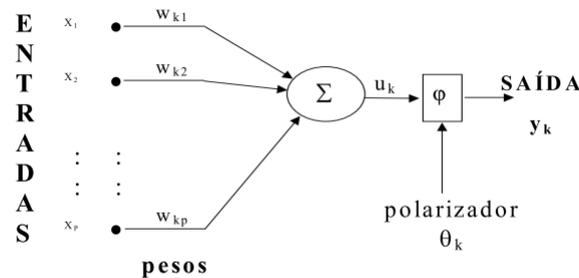


Figura 2.4: Esquema de um neurónio de uma RNA [25]

2.1.4 *PayPal*

Com o desenvolvimento tecnológico que se vivência, fazer compras ou pagar serviços remotamente tem-se tornado uma tendência em crescimento, pelo que tem sido acompanhada pelo aparecimento de várias soluções de pagamento *online*. Esta secção pretende dar a conhecer uma dessas soluções, o *PayPal*, de modo a que se possam perceber as vantagens e o funcionamento deste método de pagamento.

O *PayPal* é um serviço de pagamentos e transferência de dinheiro, criado em 1998, sendo hoje um dos sistemas de comércio online mais reconhecidos mundialmente. Para aceder a este serviço é necessário apenas um endereço de email para servir como identificação e autenticação da conta e um cartão de crédito ou conta bancária para associar aos pagamentos. Uma das vantagens deste serviço é que as informações bancárias ficam armazenadas em segurança no próprio *PayPal*, não sendo reveladas a outras entidades. Com a crescente preocupação com a segurança de dados bancários *online*, esse é um dos motivos que fazem com que este serviço esteja tão popularizado hoje em dia. Em seguida são destacadas algumas das vantagens mais importantes neste sistema [26].

- É um serviço seguro e de confiança, sendo que a taxa de burlas é uma das mais baixas neste tipo de mercado (0,3%).
- É de fácil implementação. A criação de um serviço de pagamento *online* seguro pode ser difícil, pelo que muitas empresas optam pelo *PayPal* como forma de receberem pagamentos. Este fornece os *widgets* e aplicações necessárias ao site e liga-se a uma página que trata de tudo o resto, redirecionando os utilizadores para um servidor seguro.

- Tem uma grande base de potenciais clientes. A maior parte dos consumidores, se não quase a sua totalidade é utilizador da *Internet*, pelo que o número de potenciais clientes é muito elevado.
- Possibilita a utilização de diversas opções de compra, tais como conta pessoal, de estudante ou empresarial, tendo a possibilidade de manter uma conta em diferentes categorias com diferentes privilégios e regras.
- Aceita pagamentos e cartões internacionais. O *Paypal* fornece a opção de pagar e aceitar pagamentos em 23 moedas diferentes, sendo que até aos 100 mil dólares, cobrando uma taxa de 3,9%.
- Guarda um historial de transações. Desta maneira todas as operações de entrada ou saída podem ser consultadas.

Um dos motivos que fazem com que esta plataforma seja cada vez mais utilizada em diversos contextos prende-se pelo facto de disponibilizar um grande número de recursos aos criadores de páginas de comércio ou serviços *online*, que permitem a sua integração como método de pagamento em praticamente qualquer aplicação. Uma dessas ferramentas é o *PayPal Sandbox*, que consiste num ambiente virtual de teste [27]. Este ambiente é autónomo e pretende imitar uma transação real utilizando o *PayPal*, sem que para isso estejam envolvidas contas reais ou quantias monetárias verdadeiras. Este ambiente serve apenas para o criador da plataforma testar se todas as funcionalidades do *PayPal* estão integradas no seu produto da forma que este pretende.

Para efetuar todo o processo de testes, o ambiente disponibilizado é composto por dois sites. Um deles é o da *Sandbox* que serve para acompanhar o histórico de transações e efetuar os pagamentos. O outro é o *site Developer* que serve para a criação de contas fictícias e testes de notificações. Sendo assim para simular todo o processo de pagamentos, o criador da aplicação deverá criar uma conta de produção, para fazer *login* no *site Developer*, bem como uma série de contas de cliente e de vendedor para simular o uso de todos os casos possíveis da aplicação. Após isso todos os erros encontrados nessa fase deverão ser corrigidos até os testes correrem da forma pretendida.

Outra mais valia deste serviço é o facto de incorporar um conjunto de sistemas de notificações que ajudam a monitorizar e automatizar processos. Um desses sistemas é designado por IPN (*Instant Payment Notication*), que consiste num serviço que notifica automaticamente as entidades de eventos correspondentes a transações do *PayPal*, podendo dessa forma essas entidades automatizar funções administrativas, nomeadamente a alteração do estado de pagamento de um determinado serviço ou compra. Outro serviço relacionado com as notificações é o PDT (*Payment Data Transfer*), que permite enviar informações relacionadas com as transações para o comerciante/prestador de serviços.

Como ilustrado na figura 2.5, este serviço pressupõe a existência de 3 intervenientes. O cliente, a plataforma *PayPal* propriamente dita e o *website* desenvolvido pela entidade que está a divulgar o seu produto ou serviço. O processo inicia-se com o pagamento por parte do cliente (1). Este processa-se no servidor *PayPal*, sendo para isso utilizados os botões disponibilizados pela própria plataforma. Segue-se um troca de mensagens entre o *PayPal* e o site desenvolvido, onde o primeiro envia ao segundo o identificador da transação (2), ao que este lhe deve responder enviando o ID da transação e a sua própria identificação (3). Por fim o *PayPal* envia ao site os detalhes da transação (4) que podem ser consultados pelo cliente (5).

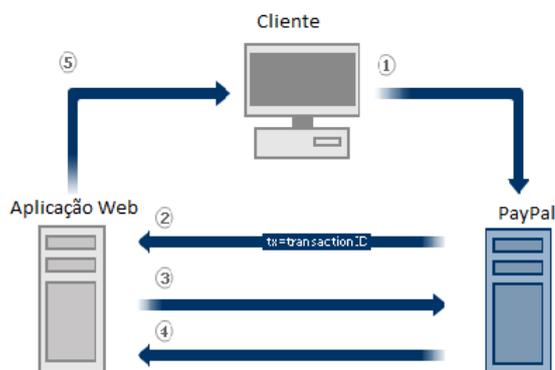


Figura 2.5: Serviço PDT [28]

2.2 Tecnologias

Nesta secção são abordadas e clarificadas algumas das tecnologias existentes que permitem que se efetue controlo de acessos. É de ressaltar que são mencionadas apenas formas de controlo de acessos de espaços físicos, mais precisamente de divisões ou quartos num determinado espaço.

2.2.1 Chaves físicas

Este é um dos métodos mais antigos e também mais utilizados para garantir que um determinado espaço não é acessível a pessoas não autorizadas. Esta tecnologia consiste numa porta com uma fechadura, que pode ser trancada com uma chave comum. Dessa forma apenas o proprietário do local terá acesso ao mesmo, sendo que pode criar várias cópias da sua chave e dessa forma dar acesso ao local a outras pessoas. Esta é uma forma de controlo de acessos bastante simples e robusta, no entanto está sempre dependente de um objeto (chave) que pode ser perdido pelo seu utilizador, pondo em causa a a segurança do espaço. Esta tecnologia é assim pouco flexível, uma vez que a autorização de acesso só pode ser feita mediante entrega da chave e em caso de perda desta ou apropriação indevida da mesma por parte de terceiros, fará com que se tenha de mudar a fechadura da porta.



Figura 2.6: Chaves metálicas

2.2.2 Códigos de barras

Esta tecnologia está amplamente divulgada como um método de identificação de produtos. Atualmente, no supermercados, quase todos os produtos possuem o seu próprio código de barras que os identifica inequivocamente. Isso é bastante importante, pois os registos de entrada e saída dos mesmos são feitos automaticamente e dessa maneira é possível saber em que altura o stock de determinado produto deve ser repostos.

Os códigos de barras consistem na representação gráfica de dados numéricos ou alfanuméricos, podendo para isso possuir uma (1D) ou duas dimensões (2D). A codificação mais usual é a 1D que consiste num conjunto de barras verticais escuras de diferentes larguras. A codificação 2D além de barras possui outros elementos como pontos ou quadrados escuros. O facto de haver uma codificação em vez de os dados estarem impressos diretamente permite uma maior eficácia na identificação, uma vez que números ou letras mal impressos poderiam facilmente induzir em erro o computador.

Um sistema que utilize esta tecnologia prevê a existência de três componentes fundamentais, que são o código de barras, um computador para registo e armazenamento dos dados provenientes desse código (a forma mais usual é utilizar uma base de dados para o efeito) e um leitor capaz de ler os dados e comunicar com as bases de dados do computador.

A leitura de informação nelas contida é feita através de um feixe laser que passa nas barras. A luz é absorvida ou refletida consoante a barra em questão seja escura ou clara. Os dados resultantes da reflexão são transformados em sinais elétricos que irão representar os dados contido num determinado código. Sendo assim, o processo de leitura pode ser dividido em cinco etapas [29], como mostra a figura 2.7. A etapa 1 consiste no varrimento do código pela cabeça do feixe laser. Na etapa 2 é refletida para um célula fotoelétrica, sendo que as áreas claras vão refletir mais luz. De seguida, a célula irá receber a luz refletida (etapa 3) e à medida que o feixe vai percorrendo as listas vão sendo gerados impulsos elétricos pela célula fotoelétrica. Um circuito elétrico presente no *scanner* que emite os feixes de luz transforma esses impulsos num código binário (etapa 4) que será enviado e interpretado pelo computador (etapa 5).

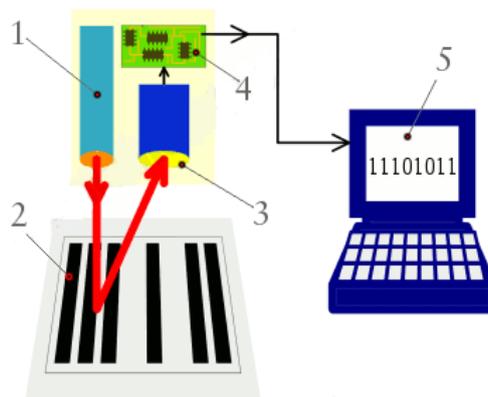


Figura 2.7: Processo de leitura de um código de barras [29]

Devido à proliferação deste tipo de tecnologia, surgiu a necessidade de criar normas específicas para a codificação dos produtos. Atualmente, a norma mais utilizada em todo

o mundo, é a EAN-13 (*European Article Number*) [30], que tal como o nome indica, utiliza 13 algarismos para efetuar a codificação. Segundo esta norma, cada conjunto de algarismos tem um determinado significado. Por exemplo, a figura 2.8 representa um código de barras de acordo com esta norma. Os três primeiros algarismos representam o prefixo do país de registo da empresa do produto. Os algarismos seguintes, que podem ser 3, 4 ou 5, conforme os casos, representam o código do fabricante e os restantes, exceto o último algarismo representam o código do produto. Deste modo todos os produtos mesmo que semelhantes podem ser distinguidos. Por exemplo, duas garrafas de água apesar de serem da semelhantes (mesma marca, mesmo volume, etc.) vão ter códigos de barras ligeiramente diferentes, de modo a serem distinguidas. Importa ainda referir que o último algarismo atua como algarismo de verificação, sendo que é calculado em relação aos outros, de modo a averiguar a validade do código.

Importa referir que existem outras normas de codificação como a UPC (*Universal Product Code*), que é utilizada sobretudo nos Estados Unidos e no Canadá [31] e ao contrário da norma anteriormente referida utiliza apenas 12 dígitos para fazer a codificação, o que limita o número possível de códigos diferentes a ser criados, mas estes têm a vantagem de ser mais fáceis e rápidos de ler. Em Portugal, além da norma EAN-13, é utilizada também a norma GS1 que garante a uniformização da codificação de produtos de peso ou quantidade variáveis [32].

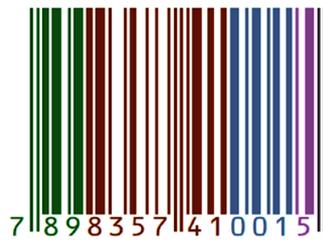


Figura 2.8: Exemplo de um código de barras segundo a norma EAN-13

Esta tecnologia revela-se assim bastante útil no que diz respeito à identificação de algo. No caso concreto do controlo de acessos, esta tecnologia possui uma vantagem em relação às chaves tradicionais. A credencial de acesso (código de barras) não tem de ser entregue pessoalmente por quem quiser autorizar o acesso, pelo que pode ser enviado o código por email por exemplo e o cliente apenas terá de o imprimir. Contudo este método requer uma constante criação de diferentes códigos de barras, para garantir a segurança nos acessos. Tem ainda o inconveniente para o cliente de este ter de apresentar uma imagem com o código para entrar no espaço que reservou.

2.2.3 Códigos QR (*Quick Response*)

Este tipo de tecnologia tem algumas semelhanças com os códigos de barras, pois também armazenam informação na forma de códigos impressos que possuem algumas partes claras e outras partes escuras, sendo que a sua leitura pode ser feita com dispositivos idênticos aos utilizados para ler códigos de barras. No entanto têm uma capacidade de armazenar informação centenas de vezes superior, devido ao facto de serem códigos 2D, isto é armazenam informação em duas direções (figura 2.9 (a)), ao contrário dos códigos

de barras que normalmente só se distinguem pela sua sequência de riscas verticais.

Uma das primeiras utilizações deste tipo de tecnologia foi no âmbito da rastreabilidade de peças no setor da indústria automóvel, mas com a proliferação e sofisticação dos *smartphones*, os códigos QR começaram a ganhar mais importância, sendo utilizados nas telecomunicações, ações de *marketing* e controlo e monitorização de *stock* de produtos [33] [34].

A popularidade desta tecnologia já ultrapassou a do clássico código de barras em algumas áreas e o principal motivo para isso acontecer é que enquanto que um código unidimensional armazena cerca de 20 algarismos, um código QR pode armazenar mais de 7 mil caracteres, tornando o seu uso muito mais apelativo, uma vez que necessitam de menos de um décimo do espaço para codificar a mesma informação [35]. Outra característica que torna esta tecnologia apelativa é o facto de poder não ser necessário um feixe laser para a leitura dos códigos, uma vez que eles podem ser interpretados através de uma imagem tirada com uma câmara e interpretada por um *software* de descodificação.

Além de possuir uma enorme capacidade de armazenamento de informação, esta tecnologia tem também uma elevada robustez no que diz respeito a possíveis erros de interpretação ou perda de informação, estando para isso dividida em 8 zonas distintas [35] (figura 2.9 (b)), cuja função é descrita em seguida.

- **Zona 1:** Estas 3 áreas estão localizadas nos cantos do código e são constituídas por uma matriz de 3x3 quadrados escuros, rodeada de quadrados brancos de espessura de uma unidade que por sua vez é rodeada de quadrados escuros com uma espessura de também uma unidade. Estes padrões encontram-se em 3 dos 4 cantos para permitir identificar a orientação do código, de modo a que este seja bem interpretado pelo *software* de descodificação.
- **Zona 2:** Estes separadores brancos têm uma unidade de largura e servem para um melhor reconhecimento da zona 1, uma vez que a separa de outros dados.
- **Zona 3:** Estas zonas funcionam como uma espécie de escala, visto que a alternância entre unidades brancas e pretas permite identificar a largura de uma unidade.
- **Zona 4:** É uma área idêntica à zona 1, contudo o quadrado central não é uma matriz 3x3. Esta zona serve para alinhar o código, eliminando eventuais distorções causadas pelo ângulo em que a imagem possa ser tirada. Com o aumento do tamanho do código pode existir mais do que um destes padrões de alinhamento.
- **Zona 5:** Esta área ao pé dos separadores (zona 5) dá informações acerca do nível de correção do erro do QR.
- **Zona 6:** Área onde estão os dados de interesse que serão convertido num fluxo de *bits*.
- **Zona 7:** Zona destinada ao armazenamento de dados adicionais que permitem antever e corrigir eventuais erros na codificação dos dados.
- **Zona 8:** Se os *bits* de dados forem múltiplos de 8, esta zona será vazia (não terá qualquer tipo de dados), caso contrário, conterá um número de bits igual ao resto da divisão inteira do total de *bits* de dados por 8.

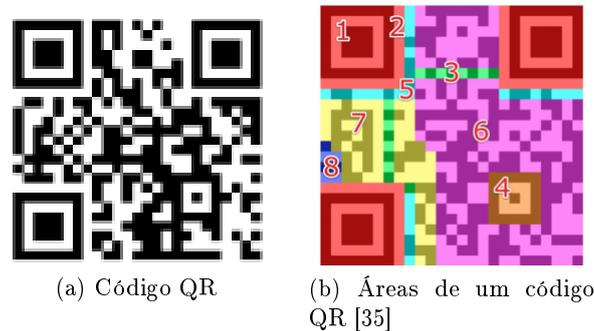


Figura 2.9: Exemplos de códigos QR

Este tipo de solução pode revelar-se bastante interessante, uma vez que permite grandes trocas de informação. No âmbito do controlo de acessos a espaços físicos, esta poderia ser uma solução, uma vez que a identificação do cliente poderia ser feita por um destes códigos [36], que lhe poderia ser enviado, visto que necessitaria apenas de uma imagem do mesmo para fazer a sua autenticação. Contudo seria necessário um algoritmo que gerasse diferentes códigos para que um determinado código de acesso fosse válido apenas num período de tempo (entre as datas de *check-in* e *check-out* de um cliente).

2.2.4 RFID (*Radio Frequency IDentification*)

A tecnologia de identificação por radiofrequência, tal como o nome indica designa qualquer tipo de dispositivo que usa sinais de radiofrequência ou variações do campo eletromagnético para efetuar a comunicação entre os vários elementos do sistema. Este tipo de tecnologia permite a obtenção de informação sem contacto físico (que pode estar armazenada por numa base de dados por exemplo) de um determinado objeto ou pessoa. [37]

Para que todo o processo de obtenção de informação decorra com normalidade são necessários três componentes no sistema [38]. Sendo assim é necessário que exista uma etiqueta (*tag* ou cartão) que possa armazenar toda a informação necessária (figura 2.10 (a)). Este componente pode possuir um *chip* interno que em conjunto com um circuito elétrico permite transmitir informações através de ondas eletromagnéticas por via de uma antena (figura 2.10 (b)). Essa etiqueta necessita que exista um dispositivo que possa ler as informações nela contidas, sendo assim o terceiro elemento é o *software* para interpretar o conteúdo que chega ao leitor de etiquetas, sendo que muitas vezes esse *software* é parte integrante do leitor.

Existem vários tipos de etiquetas RFID. Estas podem ser classificadas segundo o tipo de operações que é possível realizar com as mesmas. Desta forma existem etiquetas apenas de leitura, que já vêm com os seus dados gravados de fábrica no seu *chip* e não podem ser alteradas, permitindo assim apenas a transmissão dos dados que estão gravados de origem. Nesses casos, normalmente os dados gravados servem apenas para identificação de um produto ou pessoa. Existem também etiquetas que possuem duas zonas distintas de memória. Uma delas apenas de leitura, que não poderá ser alterada e outra zona de escrita, onde podem ser colocados e modificados novos dados. [39] [38]

De acordo com a sua forma de ativação as etiquetas podem ser classificadas como

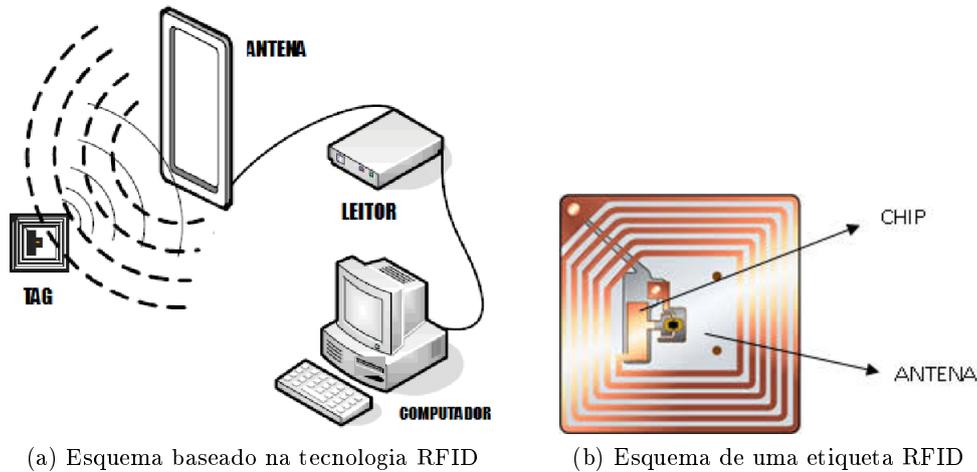


Figura 2.10: Componentes RFID

ativas ou passivas. As *tags* passivas não possuem fonte de energia própria e por isso mesmo têm de entrar num campo eletromagnético gerado por uma antena leitora para serem ativadas e transmitirem ou gravarem informações. As *tags* ativas possuem uma bateria interna que fornece energia para o funcionamento do circuito interno e para enviar dados quando estes são requisitados. Existe ainda um tipo de etiqueta que apesar de possuir alimentação interna, esta serve apenas para alimentar o seu circuito elétrico, sendo que necessita de recorrer a um campo magnético para enviar os dados. O que varia nestes três casos é sobretudo o tempo de utilização e a distância do leitor a que informação pode ser recebida. Em geral as etiquetas passivas têm um alcance mais pequeno, mas um tempo de utilização maior.

No que toca ao controlo de acessos, esta tecnologia é atualmente bastante utilizada, sobretudo em locais bastante frequentados, como universidades ou serviços. A identificação da pessoa é feita por uma etiqueta, que pode ser apenas de leitura. Posteriormente terá de haver um processamento dos dados de modo a proceder a abertura de uma porta controlada eletricamente. Este método é robusto e pode dizer-se que é mais prático do que a utilização de chaves tradicionais. Contudo esta tecnologia também depende da existência de um objeto físico (cartão) que deverá ser entregue ao utilizador de modo a que este se possa posteriormente identificar. Isto acrescido ao facto de serem necessários leitores de etiquetas em cada compartimento, elevando o custo do produto.

2.2.5 NFC (*Near Field Communication*)

Esta é uma tecnologia de comunicação sem fios e proporciona que dois dispositivos que contenham etiquetas NFC troquem informação. Tal como no caso das etiquetas RFID, esta tecnologia utiliza ondas de radiofrequência de curto alcance para estabelecer comunicação entre um etiqueta e um leitor. Desta forma para que haja uma correta comunicação os dois dispositivos devem estar próximos, ou até mesmo em contacto, visto que o máximo alcance para troca de dados se situa à volta dos 10 centímetros. Estes equipamentos são compatíveis com a tecnologia RFID, mas neste caso o mesmo aparelho pode funcionar como *tag* e fornece os seus dados a um outro dispositivo que funciona como leitor ou pode funcionar como um leitor de etiquetas e receber dados de outro

dispositivo [38]. Desta forma os três integrantes de um sistema RFID estão presentes no mesmo aparelho, que possui uma *tag* NFC, um leitor para receber informação proveniente de outras etiquetas e tem integrado *software* que lhe permite interpretar os dados que lhe chegam por intermédio de ondas rádio. Esta tecnologia encontra-se muitas vezes integrada em dispositivos *smartphones* (figura 2.11). [40]



Figura 2.11: Tecnologia NFC integrada em *smartphone*

Atualmente com o desenvolvimentos dos *smartphones*, o uso desta tecnologia tornou-se bastante facilitado, pois estes dispositivos já vêm integrados com etiquetas NFC nos seus cartões, pelo que a sua utilização é bastante simples e intuitiva, sendo atualmente este tipo de dispositivos utilizados para efetuar pagamentos, controlo de acesso, troca de ficheiros entre dispositivos, entre outros (figura 2.12).



Figura 2.12: Diferentes utilizações da tecnologia NFC

Um aparelho que comunique por NFC pode comportar-se como ativo ou passivo, à semelhança das etiquetas RFID. Deste modo podem existir dois dispositivos ativos (que têm a sua própria alimentação) para realizar a troca de mensagens. Nestes casos os dispositivos geram o seu campo eletromagnético alternadamente para que apenas um deles transmita informações num determinado espaço de tempo. Outra possibilidade é a de apenas um dos dispositivos ter a sua própria alimentação (ativo), sendo que o

outro dispositivo necessita de usar o campo eletromagnético do outro dispositivo para lhe enviar informações. Por norma as etiquetas NFC são passivas, sendo que a alimentação das mesmas provém do dispositivo onde estão integradas (bateria do *smartphone*).

No que diz respeito aos modos de operação deste tipo de equipamento, podem mencionar-se três modos diferentes, que são os seguintes: [41]

- **Leitura e gravação:** Este modo tem como base a comunicação passiva e permite a leitura ou alteração de dados existentes num dispositivo NFC. Um exemplo deste tipo de utilização é a leitura de um bilhete de cinema.
- **Peer-to-peer:** Este é um modo para troca bidirecional de informação entre dois dispositivos. Isto é, cada um pode enviar e receber dados. Esta funcionalidade é bastante útil para a troca de ficheiros entre aparelhos por exemplo.
- **Emulação de cartão:** Neste modo o dispositivo NFC pode assumir o papel de um cartão inteligente, sem que o dispositivo leitor note a diferença. Este modo permite ao dispositivo acumular todas as funcionalidades de um cartão inteligente, por exemplo, permite simular um cartão de crédito. Dessa forma possibilita que se façam pagamentos através da aproximação do *smartphone* ao dispositivo leitor.

No que diz respeito às etiquetas propriamente ditas, estas são bastante semelhantes às etiquetas RFID. Ambas possuem uma memória de armazenamento, juntamente com um *chip* conectado a uma antena, para transmissão de ondas eletromagnéticas. A única diferença existente é que as *tags* NFC são formatadas para serem integradas em aparelhos móveis (*smartphones* por exemplo) e são pequenas e baratas o suficiente para serem colocadas em todo o tipo de produtos, como posters de promoção de eventos, bilhetes para eventos, cartões de visita, etc.

Quanto ao tipo de etiquetas, podemos dizer que existem essencialmente quatro tipos diferentes, que são designados de 1 a 4 e cada tipo tem um formato e uma capacidade diferente. Estes tipos de *tag* estão de acordo com a norma ISO 14443A e 14443B, que são as normas *standard* para cartões inteligentes à distância. Por definição, os quatro tipos de etiquetas são os seguintes: [41] [42]

- **Tipo 1:** Estas etiquetas têm a capacidade de serem lidas e de o seu conteúdo poder ser reescrito. Têm uma memória de 96 *bytes*, que é suficiente para armazenar o URL de um *website* ou outro conjunto de pequenos dados. Contudo a capacidade de memória pode ser expandida até 2 *kbyte*. A sua velocidade de comunicação é de 106 kbit/s. Como resultado da grande simplicidade desta *tag*, o seu custo pode ser apelativo para muitas aplicações.
- **Tipo 2:** Esta *tag* tal como a anterior pode ser lida e o seu conteúdo reescrito. Tem uma capacidade de memória de 48 *bytes*, sendo que pode ser expandida até 2 *kbyte* e tem uma velocidade de comunicação de 106 kbit/s.
- **Tipo 3:** Normalmente possui uma capacidade de memória de 2 *kbyte* e uma velocidade de comunicação de 212 kbit/s. Estes tipos de etiqueta são utilizados em aplicações mais complexas, uma vez que o seu custo também é mais elevado.
- **Tipo 4:** Este é o tipo de *tag* mais complexa. Tem uma capacidade de memória de até 32 *kbytes* e a velocidade de comunicação é entre 106 kbit/s and 424 kbit/s.

Este tipo de tecnologia possui um grande número de aplicações e a sua utilização é bastante prática e intuitiva, pelo que apenas é necessário o dispositivo móvel que pode ser facilmente transportado pelo utilizador para qualquer local. Contudo para efetuar um controlo de acessos que pretende ser o mais universal possível, esta pode não ser a melhor solução, visto que atualmente ainda nem todos os *smartphones* possuem este tipo de tecnologia integrada. Além disso, o facto de os dispositivos poderem ser furtados ou perdidos faz com que o nível de segurança destas aplicações não seja o mais indicado.

2.2.6 Biometria

Este tipo de sistema baseia-se no controlo de uma característica física de uma pessoa, de modo a identifica-la inequivocamente. Alguns exemplos de tais características são as impressões digitais, o reconhecimento da íris dos olhos, reconhecimento de voz, entre outras (figura 2.13). Este tipo de método tem como premissa que cada indivíduo é único e pode ser identificado inequivocamente através de um determinada característica.

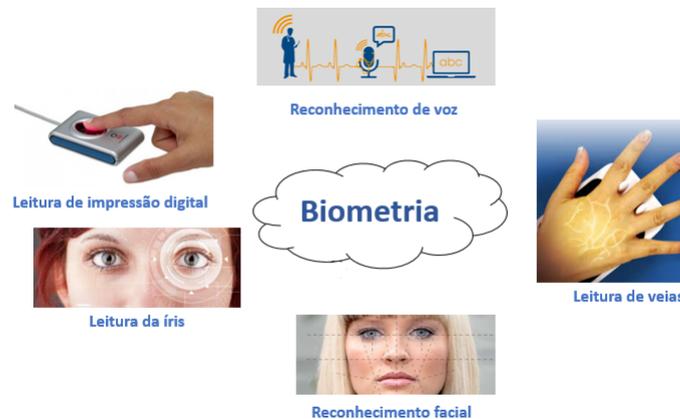


Figura 2.13: Diferentes formas de identificação por biometria

A tecnologia envolvida nos sistemas de reconhecimento biométrico está bastante dependente de qual a característica física do indivíduo que se pretende utilizar para fazer a sua identificação. Não obstante, todos estes processos têm em comum o facto de se poderem dividir em cinco etapas ou fases essenciais, que são a recolha de dados, a transmissão, o processamento de sinal, decisão e armazenamento de dados. Em seguida cada uma destas fases irá ser exposta com mais detalhe. [43]

- **Recolha de dados:** Esta fase consiste na escolha de uma determinada característica diferenciadora e sua medição através de um sensor/aparelho de medição. Os valores lidos pelo sensor deverão ser diferentes de indivíduo para indivíduo, permitindo desta forma a sua identificação. Os dados de um determinado utilizador devem ficar armazenados, para que haja o seu reconhecimento posteriormente, por comparação dos valores lidos nesse momento pelo aparelho de leitura.
- **Armazenamento:** Depois de recolhidos os dados dos utilizadores que se pretende identificar, estes devem ser armazenados em bases de dados para posterior comparação com os dados a serem introduzidos.

- **Transmissão:** Alguns sistemas biométricos recolhem dados numa determinada localização, mas efetuam o seu processamento noutra local/equipamento. Sendo assim é necessário que se dê o seu envio para o local de processamento, de forma adequada. Se estiver envolvida uma grande quantidade de dados, pode ser necessário comprimi-los antes da sua transmissão. Este tipo de processos pode envolver imagens (caso do reconhecimento facial) ou ficheiro de som (reconhecimento por voz), entre outro tipo de ficheiros, pelo que convém que exista um protocolo standard de comunicação implementado no sistema (caso seja necessário efetuar transmissão).
- **Processamento de sinal:** Depois de adquirida e transmitida, uma característica biométrica, é necessário comparar os dados lidos com os dados previamente armazenados. Este processo pode ser dividido em 4 partes que são a segmentação, a extração de um padrão, o controlo de qualidade e correspondência padrão. A segmentação é o ato de encontrar dentro dos dados o padrão que se pretende analisar. Por exemplo no reconhecimento facial esta etapa diz respeito a encontrar os limites (extremos) da face a identificar. Em termos muito gerais podemos afirmar que a segmentação é uma separação grosseira do sinal, enquanto que a extração do padrão consiste num tratamento mais minucioso, onde são eliminadas distorções e ruídos do sinal recebido. Após a extração do padrão, interessa verificar se o sinal recebido é de boa qualidade. Se for verificado que os dados não fazem sentido é possível concluir que o sinal recebido tinha alguma espécie de defeito. Desta maneira seria pedido ao utilizador uma nova leitura. Por fim a amostra tem nesta fase um tamanho muito inferior ao sinal original e já pode ser comparada com os dados existentes nas bases de dados.
- **Decisão:** Após a comparação do sinal recebido com dados que estão armazenados, a etapa final consiste na tomada de decisão, ou seja após recolhidos os dados o sistema irá aferir se o utilizador em causa faz corresponder os seus dados a algum dos que estejam na base de dados. Como os dados amostrados podem não ser exatamente iguais aqueles que estão armazenados, convém estabelecer uma diferença máxima entre padrões adequada, de modo a os dados de um utilizador estejam sempre dentro de determinados valores, sem que este seja confundido com outros.

Este tipo de tecnologia é bastante eficiente no que diz respeito à manutenção da segurança de um dado local, visto que à partida não existe o risco de o utilizador perder as suas credenciais de identificação, ou de estas lhe serem furtadas, o que não se verifica para as tecnologias anteriores. No entanto, para que o processo de reconhecimento ocorra sem falhas, é necessário equipamento de qualidade e *software* com alguma sofisticação para tal. Outro ponto contra esta tecnologia, é o facto de poder ser considerada intrusiva por parte do utilizador. Além disso, no contexto do problema apresentado poderá não ser a melhor solução, visto que se pretende um sistema completamente automático, sem intervenção de terceiros e com esta tecnologia existe sempre a necessidade de introduzir os dados de um determinado utilizador previamente numa base de dados, processo que poderia ser moroso, visto ser necessário recolha dos dados biológicos do cliente, exigindo assim a sua presença nesse processo.

2.2.7 Teclados numéricos ou alfanuméricos

Este tipo de sistemas prevê a existência de um teclado numérico ou alfanumérico junto da porta de acesso (figura 2.14). Este teclado serve basicamente para que o utilizador digite a sua senha de acesso, que lhe foi previamente fornecida. Desta forma este sistema pode realizar os processos de identificação e autenticação de uma vez só, pois uma senha pode identificar inequivocamente um utilizador, se houver mais do que uma pessoa com permissão de acesso e se cada uma tiver uma senha diferente.



Figura 2.14: Teclado numérico para controlo de acessos [44]

Além da porta com circuito elétrico, este sistema deve possuir um sistema para que este possa ler as teclas que foram pressionadas e dessa forma autorizar ou negar o acesso a um determinado local por comparação com as senhas válidas que deverão estar armazenadas em bases de dados. [45] [44]

Este tipo de sistema não requer qualquer tipo de credencial física (não depende de nenhum objeto externo ao sistema), pelo que a autorização de acesso pode ser feita de forma automática, com o envio de uma senha válida ao utilizador que se pretende dar acesso. É possível ainda alterar remotamente as senhas deste tipo de sistema, de modo a ter uma maior segurança, pois caso se pretenda alterar autorizações de acesso, basta modificar o código de validação e fornece-lo apenas a quem se deseja autorizar o acesso.

2.3 Soluções académicas propostas por outros

Com vista a identificar soluções académicas existentes nesta área de estudo, foi feita uma pesquisa em várias plataformas. Esta secção apresenta um conjunto de soluções propostas que podem ser interessantes para o presente trabalho. De uma forma geral podem dividir-se os temas aqui abordados em mecanismos para controlo de acessos e sistemas que permitam a monitorização e determinação da ocupação de um determinado local.

Na figura 2.15 está representada esquematicamente uma solução proposta para o controlo de acessos no setor turístico [46]. Nesta solução o autor pretende garantir as 5 etapas enumerados na figura 2.15 (a).

De um modo geral esta solução dá oportunidade ao proprietário de um determinado espaço de o divulgar *online* com num site para o efeito (etapa 1). Ao mesmo tempo um qualquer cliente estaria habilitado a fazer a sua reserva remotamente (etapa 2) e

ter informação acerca de preços e disponibilidade dos quartos. A reserva teria de ser confirmada pelo proprietário (etapa 3) que teria de contactar o cliente caso a reserva fosse aceite, de modo a que este receba indicações para o pagamento, bem como um código de acesso ao seu quarto.

O registo de reservas (etapa 4) é feito numa base de dados que pode ser controlada pelo proprietário e onde estarão todos os dados necessários para a identificação de cada cliente. O *check-in* (etapa 5) poderia ser feito pelo cliente sem intervenção de outra pessoa, pois este possuiria o código de acesso ao espaço.

A implementação desta solução está esquematizada na figura 2.15 (b) e para a implementar, o autor criou um site, recorrendo a linguagens como HTML, PHP e JavaScript. Este possui diversas páginas destinadas aos possíveis clientes, onde estes poderão encontrar todo o tipo de informação relacionada com o serviço em questão.

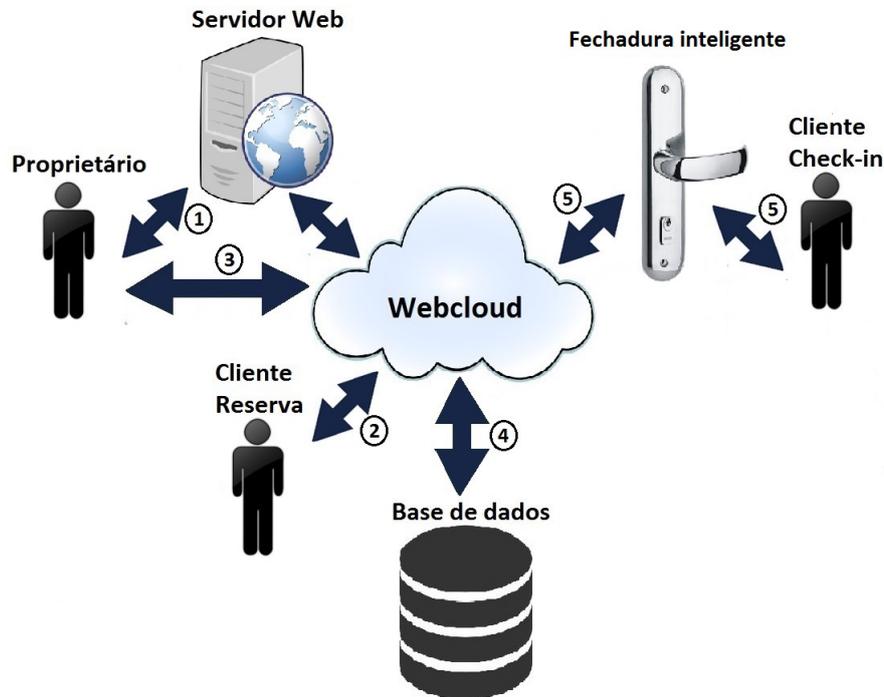
Toda a informação relativa aos clientes, foi armazenada numa base de dados em MySQL. Algumas páginas do site criado trocam informações com a base de dados, de forma a apresentarem a disponibilidade de quartos com base nas reservas que já foram aceites e estão na base de dados. A reserva é feita com o preenchimento de um inquérito presente numa das páginas do site, que será enviado para o email do proprietário após submissão. A aceitação de uma inscrição é feita pelo mesmo, recorrendo para isso a uma página de administrador a que só este tem acesso. Essa página serve apenas para a confirmação de inscrições e para que visualização da ocupação do espaço numa determinada altura. Se a inscrição for aceite, é enviado um email pré-definido ao cliente onde consta o seu código de acesso e indicações relativas ao pagamento do serviço, que será efetuado numa página do site destinada ao efeito, recorrendo ao *PayPal*, que permite pagamento utilizando endereços de email.

A fechadura inteligente implementada nas portas, comunica com a base de dados e com o cliente, de modo a que este se possa identificar autenticar, introduzindo o código que lhe foi previamente atribuído. Para isso foi utilizado o módulo ESP8266, previamente programado, no circuito de uma fechadura elétrica. Este módulo gera uma rede *Wi-Fi* e o cliente deve-se ligar a ela, com recurso a um qualquer dispositivo que seja capaz de aceder à *Internet* e possa ter um *browser*, como um telemóvel, computador ou *tablet*. Depois de conectado à rede, o cliente terá acesso a duas páginas criadas no módulo ESP8266, podendo aí inserir o seu código de acesso que irá ser comparado com o código existente na base de dados para aquela data, podendo assim aceder ao espaço qualquer pessoa que possua a correta credencial de acesso. É com recurso a uma outra página que poderá posteriormente voltar a restringir o acesso ao local (trancar a porta).

Foi proposto um Sistema de Gestão e Controlo de acessos no setor hoteleiro por [47] baseado numa arquitetura centralizada, com um único servidor que comunica com os computadores recorrendo à rede *Internet*. Na figura 2.16 está representada a estrutura de autorização de acessos proposta, em que a identificação de um utilizador é feita com recurso a etiquetas RFID que devem ser lidas por um dispositivo que comunica com o computador por ligação universal (USB). Com base nos dados lidos o responsável poderá autorizar o acesso a um determinado espaço, comunicando com o servidor através da rede *Wi-Fi* utilizando o protocolo TCP/IP.

Para identificar os espaços e perceber se um dado utilizador tem ou não acesso ao mesmo é utilizada a arquitetura da figura 2.17 em que existe um dispositivo intermédio que permite a comunicação das etiquetas RFID e a porta do local em questão, com vista a identificar o espaço e o utilizador. O dispositivo intermédio realiza comunicação com

servidor, de modo a autorizar ou não o acesso.



(a) Modelo conceptual



(b) Implementação do modelo proposto

Figura 2.15: Esquemas do modelo proposto por [46]

Na figura 2.17 está representado esquematicamente o modelo proposto por este autor. Como dispositivo para interação com as etiquetas RFID foi utilizado um conversor (USB 2.0 to serial MCP2200) que comunica com um microcontrolador (PIC18F2520) por I2C. No que diz respeito à validação de acessos, existem dois módulos. O central é composto por um microcontrolador (PIC32MX11FO16B) que em conjunto com a tecnologia *ZigBee* que permite processar todos os pedidos de acesso. Este módulo permite também a comunicação com o servidor comunicando com o mesmo de acordo com o protocolo *ZigBee*, estabelecendo assim uma comunicação sem fios. Existe também em cada porta um dispositivo de identificação do espaço constituído por um microcontrolador (PIC18F2520) e por um relé que possibilita o fecho do circuito de alimentação da fechadura elétrica da porta, caso o acesso seja autorizado, permitindo assim a abertura do trinco elétrico da porta. A conexão do equipamento de comunicação é feita com recurso a um acoplador UEXT (*Universal Extension Connector*) que comunica com o módulo central. Todos os



Figura 2.16: Modelo de autorização de acessos proposto pelo autor [47]

dados necessários ao funcionamento do sistema estão organizados numa base de dados que foi criada em MySQL. Dado o sistema centralizado, essa base encontra-se no servidor que de acordo com os dados desta poderá conceder ou não autorização de acesso a um determinado local.

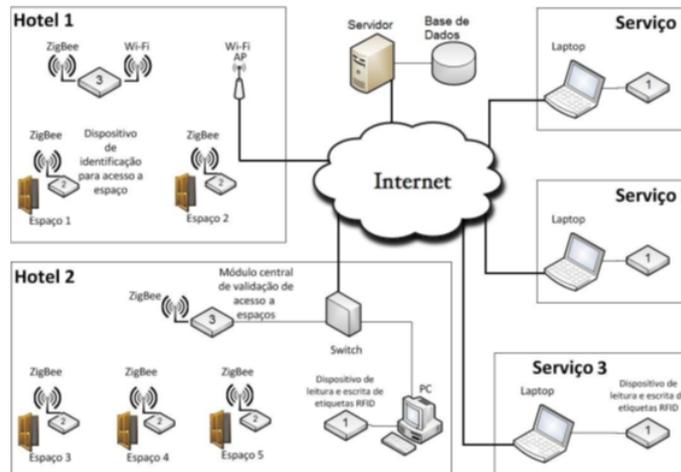


Figura 2.17: Implementação do modelo proposto pelo autor [47]

A figura 2.18 mostra esquematicamente um solução proposta por [48]. Esta baseia-se na criação de um sistema ligado à rede, em que o cliente poderia efetuar a sua reserva *online*, tendo para isso de preencher alguns formulários com dados pessoais. Por sua vez as páginas *Web* criadas comunicam com a base de dados do sistema, introduzindo nas tabelas da base de dados os dados relativos ao cliente. A base de dados também fornece informações às páginas *Web*, verificando se o quarto que se pretende alugar se encontra disponível entre as datas de *check-in* e *check-out* pretendidas. Após efetuado o pedido de reserva, o pagamento é efetuado remotamente, via *PayPal*. Depois de o pagamento ser efetuado, o cliente recebe então de forma automática um email com um código de acesso gerado pelo próprio sistema e inserido na base de dados, para que possa entrar no espaço que reservou. O *website* desenvolvido possui ainda uma página protegida com nome de utilizador e palavra passe, destinada apenas ao administrador, para que este possa monitorizar o estado das reservas.

O sistema instalado junto da porta do estabelecimento é controlado pelo microcontrolador ESP8266 com um módulo *Wi-Fi* integrado, permitindo deste modo que a porta

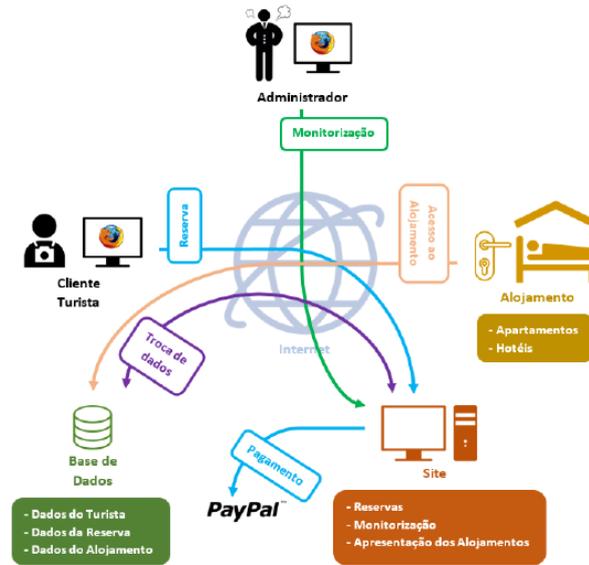


Figura 2.18: Esquema da solução proposta por [48]

possa estar ligada ao servidor central através de *routers*. A figura 2.19 mostra esquematicamente todos os componentes associados à fechadura. Além do microcontrolador, está presente um teclado alfanumérico na parte exterior da porta, para que o cliente possa introduzir a senha que recebeu via email. A leitura do teclado é feita através de um divisor resistivo para dessa maneira poder ser feita a leitura do teclado através dos diferentes valores lidos na entrada analógica do microcontrolador. Através da ligação ao servidor, é possível saber sempre se a senha introduzida é válida ou não, por comparação de informação lida na base de dados.

Existem ainda três LEDs que permitem ao utilizador perceber se a senha que introduziu está correta ou não. O controlo de abertura da porta é feito com recurso a um fecho elétrico que pode ser aberto ou fechado dependendo do estado do relé que é comandado pelo ESP8266.

Este sistema dá resposta a uma grande parte dos problemas identificados anteriormente, contudo, o método de leitura do teclado poderia ser mais eficiente caso se utilizassem entradas digitais do microcontrolador, ou em alternativa um *IO expander* para expandir as entradas/saídas digitais do microcontrolador. Além disso, faz falta nesta solução a criação de uma senha de acesso fixa, de modo a que proprietário e funcionários de limpeza possam utilizar sempre a mesma senha. O facto de apenas ter integrado um sensor de abertura de porta não permite determinar o estado de ocupação de um quarto. Além disso a utilização de apenas um sensor na solução, não permite grande monitorização do ambiente em causa, sendo que seria de interesse a integração de uma rede de sensores que não fosse intrusivo para o cliente e que permitisse ao proprietário monitorizar o ambiente de cada espaço a alugar.

Um método de acesso e identificação baseado na utilização de um dispositivo móvel foi proposto por [49]. Este autor utilizou técnicas de criptografia aliadas com tecnologias de comunicação baseadas nas plataformas *Android* e *Windows Mobile* de modo a utilizar o endereço físico (*MAC address*) de um equipamento como chave de entrada, uma vez que

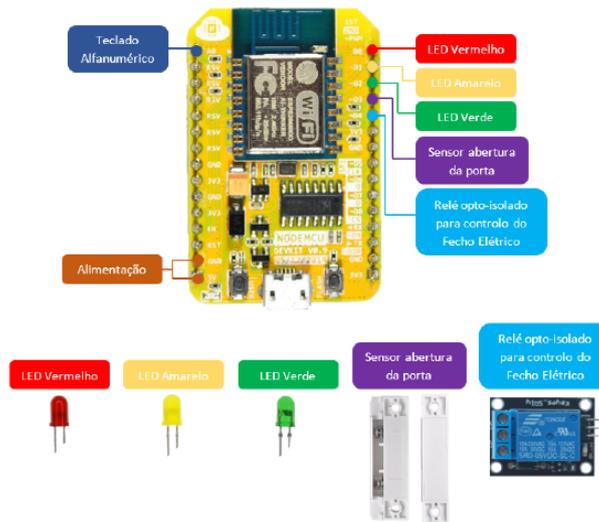


Figura 2.19: Esquema de ligações [48]

este código é único para cada equipamento.

Esta solução prevê que o dispositivo tenha de ser previamente registado por um rececionista utilizando um *website* para esse efeito, onde diversos campos devem ser preenchidos. Posteriormente a autenticação e identificação irá ser feita através de uma comunicação cliente/servidor entre o dispositivo móvel que deverá estar ligado à rede e entre o servidor (autor utilizou o servidor WAMP nesta aplicação).

A vantagem deste método é que os dados para que possa ser concedido o acesso podem ser enviados remotamente, não necessitando o cliente de se deslocar ao local para efetuar o *check-in*. No entanto é necessário um tipo de programação diferente para cada tipo de dispositivo (*Android*, *Windows Mobile*, etc.), o que faz com que a solução seja pouco universal. Além disso o processo de registo não é automático e o facto de ser utilizado um equipamento que está sujeito a ser perdido pelo utilizador, ou a ser-lhe furtado, faz com que a segurança deste tipo de solução não seja a melhor.

Um método de acessos que privilegia a alta segurança foi proposto por [50]. Esta solução é baseada num sistema de IoT e utiliza três modos diferentes de identificação, para garantir a autenticidade do utilizador.

A primeira fase de identificação consiste na leitura da impressão digital, que terá de ser comparada com os dados recolhidos e armazenados na base de dados. Caso o primeiro método seja validado, o utilizador deverá introduzir uma senha num teclado situado perto da porta. Por fim, caso as verificações anteriores sejam aceites, para ser autenticado, o utilizador necessita ainda de responder corretamente a uma pergunta.

A figura 2.20 mostra esquematicamente o modelo proposto, cujo controlo é feito através de um *Raspberry Pi* que tem conectado a si um leitor de impressões digitais e um conjunto de sensores para monitorizar o espaço e garantir a segurança do mesmo, ativando alarmes caso haja alguma inconformidade ou a tentativa de acesso não autorizado. Este sistema permite também acionar a abertura da porta, sendo que estará a comunicar constantemente com servidor IoT, para envio de mensagens ou leitura de dados nas tabelas da base de dados do sistema.

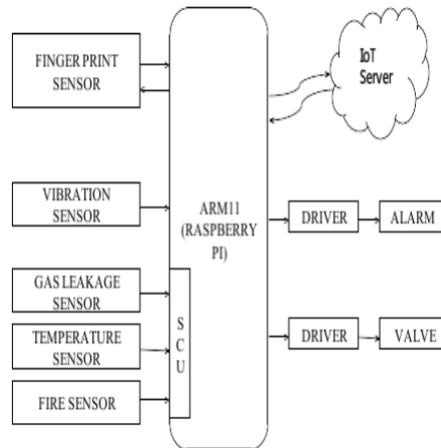


Figura 2.20: Arquitetura do sistema proposto [50]

Esta solução prioriza sobretudo a segurança. Contudo na aplicação pretendida para este trabalho, o controlo feito para identificar o utilizador nesta solução é algo excessivo, pelo que se prevê a utilização de apenas um dos métodos de identificação como forma de simplificar o sistema e poder automatizar o processo.

Um sistema baseado na utilização de um *Raspberry Pi* para processamento de dados e controlo de acessos foi proposto por [51]. Esta solução foi feita para otimizar o controlo de acessos de uma residência, estando para isso o dispositivo ligado à rede *Wi-Fi*. A figura 2.21 mostra a arquitetura deste sistema. A identificação e autenticação do utilizador podem ser feitas de duas formas. Uma delas consiste em digitar uma senha de 5 caracteres que será comparada com a senha correta e em caso de estas serem iguais será autorizado o acesso. Por outro lado, existirá também uma câmara no exterior, junto à porta, que caso a senha seja negada, é tirada uma fotografia ao sujeito que está a tentar ter acesso e é enviada por email ao proprietário. Por sua vez este pode autorizar o acesso remotamente, introduzindo a fotografia na base de dados que irá ser lida pelo *Raspberry Pi*. Desta forma, o controlo de acessos pode ser feito unicamente com base no reconhecimento facial. Mais uma vez caso a imagem capturada não seja identificada nos dados da base de dados, um email é enviado ao proprietário para que este confirme o acesso introduzindo a nova imagem na base de dados, ou então tome outro tipo de decisão caso se trate de um intruso.

O acesso baseado neste sistema é bastante simples, pelo que basta que um indivíduo saiba a senha válida para poder entrar, no entanto o reconhecimento facial requer a existência de uma câmara adequada, bem como *software* de processamento para reconhecimento de padrões. Esta técnica pode ser falível se todo o sistema não estiver devidamente calibrado e na utilização concreta da indústria turística, a utilização deste tipo de solução iria requerer um elevado número de ficheiros na base de dados para a comparação, o que poderia tornar o processo demasiado lento, além disso, o cliente teria de disponibilizar uma fotografia sua, para que a parte do reconhecimento facial pudesse funcionar, tornando o todo o processo mais moroso e intrusivo sendo por isso pouco interessante.

Outras soluções baseadas em códigos foram encontradas, nomeadamente a proposta

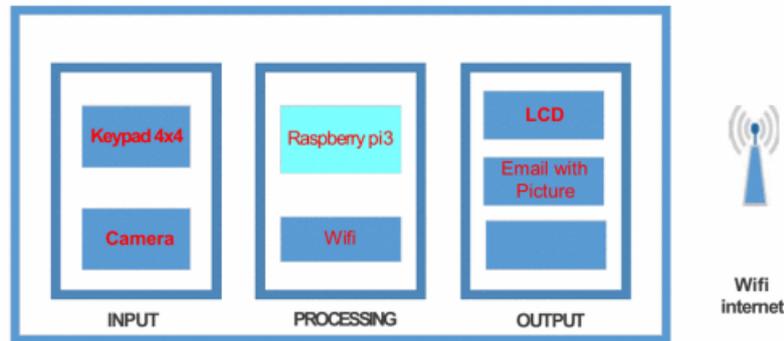


Figura 2.21: Esquema de arquitetura proposta por [51]

feita por [52] que prevê um sistema composto por um microcontrolador PIC 16F877A, um módulo GSM, um LCD, um motor e seus *drivers* de controlo, um alarme e um conjunto de *switches*.

Neste caso o código seria digitado carregando um determinado número de vezes na sequência de *switches*, num certa ordem. Por exemplo se o primeiro número da senha fosse 3, teria de ser pressionado o primeiro botão um número igual de vezes e assim por diante. O ecrã LCD serviria para que o utilizador tivesse a perceção do estado da identificação, enquanto que o motor teria a função de fazer o controlo da abertura da porta, consoante o acesso fosse concedido ou negado. Importa ainda referir que este sistema comunica com o proprietário, enviando-lhe mensagens, recorrendo para isso ao módulo GSM.

Esta solução não é de todo a mais prática, pelo que a utilização de um teclado numérico para digitar as senhas constitui um método mais intuitivo para o utilizador. Também a utilização de rede *Wi-Fi* poderia ser pensada ao invés da utilização de um módulo GSM que requer um modem adicional, bem como um cartão. Na maioria das vezes os espaços já possuem *routers* que permitem a ligação *Wi-Fi*, evitando-se assim a integração de mais componentes no espaço.

No setor turístico, além de ser de interesse um sistema que permita um controlo de acesso do espaço à distância, é também relevante que haja forma de o proprietário perceber se o seu espaço está a ser ocupado ou não, para por exemplo permitir a limpeza do espaço quando este não esteja a ser usufruído. Neste âmbito, existem diversas tecnologias capazes de permitir a identificação e a contagem das pessoas que entram num determinado local.

Uma forma interessante e económica de saber quantas pessoas entraram ou saíram de um local passa pela utilização de um conjunto de sensores junto da porta, que por si só podem não dar grande informação, mas combinando toda a informação extraída dos mesmos, recorrendo para tal a algoritmos apropriados, é possível determinar o estado de ocupação de um local.

Algumas soluções neste âmbito devem ser à partida descartadas, como por exemplo sistemas baseados em visão artificial com o proposto por [53], pois seria demasiado invasivo introduzir câmaras dentro de um quarto. Sistemas que se baseiam na monitorização da utilização de equipamentos (computadores, por exemplo), como o proposto por [54] também não são a melhor solução, pois nada garante a existência desses equipamentos

no quarto destinado a aluguer. Além disso sistemas baseados em etiquetas RFID ativas (como o sistema proposto por [55]) não permitem uma automatização do processo, na medida em que tinham de ser entregues ao cliente na altura do *check-in* essas mesmas etiquetas, e uma eficácia de um sistema destes estaria dependente de o utilizador as carregar sempre consigo e ter o cuidado de as não aproximar de um material metálico. Sendo assim em seguida serão apresentadas algumas soluções que se consideraram de relevo neste âmbito.

Uma das soluções encontradas visa a criação de uma porta inteligente e foi proposta por [56]. Para tal, foram utilizados dois pares de lasers foto-transístores para que dependendo da ordem com que estes fossem acionados, se percebesse se a pessoa a passar por eles estaria a entrar ou a sair de um determinado espaço. Além disso, foi colocado um sensor ultrassónico no topo da porta, para poder aferir a altura da pessoa que estivesse a passar, bem como um tapete capaz de determinar o seu peso. Os sensores de altura e peso estão conectados a arduinos individuais que por sua vez estão conectados com um *Raspberry Pi* instalado na porta. De cada vez que é detetada a passagem num dos lasers, é desencadeada uma interrupção para o *Raspberry Pi* que começa a gravar a leitura dos sensores.

Para atribuição das respetivas identidades, com base nos dados fornecidos pelos sensores foi utilizado o método de classificação de floresta aleatória, para "ensinar" a identidade de cada utilizador. Em alternativa, foi também utilizado o algoritmo SVM e os seus resultados foram comparados.

Esta solução é interessante, pelo que permite contar quantas pessoas entram e saem de um local, com base nos sensores laser, o que é de relevo para este trabalho. Contudo, este tipo de sensor é sensível, pelo que este método poderá não ser suficientemente robusto, além disso no caso em que duas pessoas entrassem no local praticamente ao mesmo tempo, provavelmente só iria ser contabilizada uma entrada. Além disso, a identificação dos utilizadores com base no seu peso e altura não é o melhor método no caso de um serviço turístico, isso porque à partida esses dados não são conhecidos e mesmo que fossem, para efetuar o *check-in* sem ser necessária a presença de um rececionista, é necessário um método muito mais robusto de identificação, de forma a garantir que o acesso apenas é concebido à pessoa que efetuou a reserva para aquele período de tempo.

Existem no entanto outras maneiras de conseguir saber a ocupação de um determinado local. Uma delas foi utilizada para determinar este parâmetro com exatidão, para otimização de um sistema AVAC [57]. Para tal, é utilizado um sensor PIR (Passive Infra Red) , e um *switch*, que são ligados aos pinos GPIO do microcontrolador CC2530. Neste caso é possível saber se a porta se encontra aberta ou fechada, pelo estado do *switch*. Se estiver aberta, à partida haverá pessoas no interior do espaço, pois pressupõe-se que se lá não estiver ninguém, a porta estará fechada. Se a porta estiver fechada, nada garante que a divisão esteja ou não ocupada. Daí a necessidade de utilização do sensor PIR que irá detetar se existem movimentos no interior da sala, determinando se está ou não a ser ocupada. Contudo, estes sensores são bastante sensíveis, pelo que podem ser acionados pela passagem de pessoas junto da porta, mas no exterior. Sendo assim os autores desta solução propõem um algoritmo simples, em que os primeiros seis segundos de impulsos do sensor PIR são ignorados, sendo amostrados os dois segundos seguintes. Essa amostra será comparada com o padrão de não ocupação, para determinar se a divisão está a ser ocupada ou não. Se a sala for determinada como ocupada, o seu estado só mudará eventualmente quando for detetada nova abertura da porta.

Existe um cenário onde uma divisão ocupada poderia ser determinada erradamente como vazia. Tal poderia acontecer quando mais do que uma pessoa estivesse no espaço. Se um dos ocupantes sair e o outro estiver a realizar uma atividade ligeira (escrever ao computador por exemplo), provavelmente o sensor PIR não detetaria nenhuma presença. Para compensar isso, o algoritmo desenvolvido por este autor, prevê que seja ativada uma interrupção no microcontrolador quando uma sala é marcada como fechada e desocupada. Desta forma, ao menor movimento detetado pelo sensor, vai ser amostrado um novo padrão e ser comparado novamente com o estado de não ocupação, para desta forma averiguar se existe alguém no interior da divisão ou não.

Este tipo de solução pressupõem que uma porta aberta significa que a sala está ocupada, o que pode não se verificar. Em particular quando um utilizador sai e se esquece de fechar a porta. Além disso o alcance de sensores PIR é limitado, pelo que numa divisão de grandes dimensões pode-se tornar difícil detetar movimentos, fazendo esta solução pouco universal.

Uma outra solução com vista a otimização de um sistema de AVAC foi proposta por [58]. Neste sistema, o autor optou por tratar a ocupação como um problema de classificação. Este pressuposto é válido, visto que para o caso em estudo as classes possíveis para a ocupação eram conhecidas. Isto é, foram utilizadas duas salas de tipologias diferentes, sabendo que numa delas o número de ocupantes que poderia ter varia entre 0 e 3 pessoas, sendo que na outra esse número varia entre 0 e 9.

Para implementar esta solução foi utilizado um conjunto de sensores (figura 2.22), que inclui sensores de movimento, som, luz, sensores PIR, CO_2 , temperatura, humidade e um sensor de porta (*switch* que indica se a porta está fechada ou aberta). Com vista a treinar um conjunto de dados lidos destes sensores, foram utilizados alguns algoritmos, entre os quais se encontram o algoritmo SVM, árvores de decisão e redes neurais artificiais.

Esta solução não revelou grande eficácia em determinar o número de ocupantes numa divisão [58], contudo os resultados obtidos podem ser utilizados com segurança se o objetivo for apenas determinar se uma divisão se encontra ocupada ou não, independentemente do número de ocupantes que estejam no seu interior. Isso acontece porque é difícil estabelecer vizinhanças com os dados obtidos, por exemplo, os dados para uma ocupação com 5 ou 6 pessoas são bastante semelhantes, pelo que ocorrem erros de classificação devido a esse facto.

Verificou-se também que um dos algoritmos com melhores resultados foi o das árvores de decisão. É importante ainda referir que o número de sensores utilizado é algo elevado, pelo que alguns dos quais podem estar apenas a criar alguma entropia nos dados. De acordo com o autor os dados com mais peso na determinação da ocupação são o nível de CO_2 , o estado da porta (aberta ou fechada) e os valores lidos do sensor de luz.

Um sistema baseado na utilização de um sensor DTH22 (sensor de temperatura e humidade), um sensor de luz e um sensor de CO_2 foi proposto por [59]. Os dados dos sensores são lidos por um microcontrolador arduino. Para saber a correta ocupação do local em estudo, foi utilizada uma câmara juntamente com um *Raspberry Pi*, que envia as imagens obtidas remotamente, com vista a comparar os valores obtidos através dos sensores com os valores reais de ocupação determinados por análise das imagens.

Foram utilizados três conjuntos de dados, sendo que um foi o conjunto de treino e os outros dois conjuntos foram utilizados como teste. Foram utilizados alguns algoritmos ou modelos de classificação diferentes, os quais incluem CART, RF, GBM, entre outros. Para perceber a dependência da qualidade dos resultados com o tipo de dados recebido,

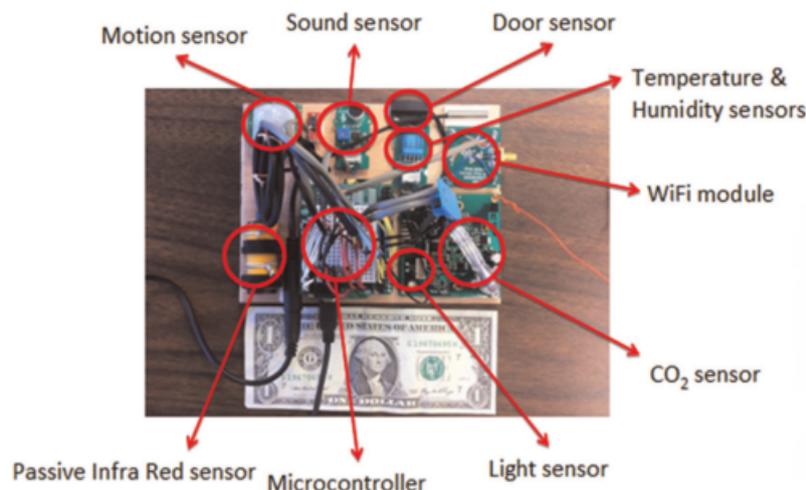


Figura 2.22: Sensores utilizados na solução proposta por [58]

foi feito o teste de utilizar apenas alguns dos dados, com os diferentes tipos de algoritmos, para se perceber qual a combinação que proporciona melhores resultados. O treino dos dados foi feito recorrendo ao pacote de treino e regressão (CARET).

Uma conclusão interessante a tirar desta solução é que o algoritmo de florestas aleatórias consegue produzir resultados muito bons nas amostras de treino. Contudo quando é testado em outras amostras por vezes produz resultados menos satisfatórios em comparação com outros algoritmos. É ainda de salientar que esta solução foi testada apenas numa divisão, com inexistência de luz natural (persianas fechadas), pelo que a replicação dos resultados noutra tipo de divisão pode não acontecer.

Um sistema para determinar a ocupação numa sala naturalmente ventilada e com 6 computadores fixos foi proposto por [60]. O autor propõem a utilização de sensores de temperatura acoplados aos computadores, para perceber o seu padrão de utilização, enquanto que um outro conjunto de sensores seria responsável por recolher dados sobre o clima no interior da sala, medindo parâmetros como humidade, temperatura e intensidade luminosa. Estes dados foram medidos e gravados com sensores comerciais HOBO. Além dos dados mencionados anteriormente foram analisados alguns dados a partir de sensores de qualidade de ar, que fornecem dados acerca da concentração de CO_2 e de compostos orgânicos voláteis no ar. Adicionalmente foram também utilizados três sensores PIR para detetar movimentos. Como ferramenta para a partir dos dados recolhidos determinar a ocupação de um espaço, foi utilizada uma ferramenta do *software Matlab* que permite aplicar o algoritmo de redes neurais artificiais.

Apesar de utilizar uma quantidade considerável de sensores, esta solução não consegue determinar o número de ocupantes com uma precisão superior a 70%, segundo o autor. Outro aspeto negativo é o facto de a solução assumir a existência de computadores na sala e utilizar o seu padrão de utilização para ajudar a determinar a ocupação, pois nem todas as divisões possuem equipamentos deste tipo passíveis de serem monitorizados. Além disso o facto de serem utilizados sobretudo sensores comerciais já calibrados e preparados para gravar os registos, faz com que a solução não seja económica.

Uma solução baseada em eventos determinados a partir da leitura de sensores foi proposta por [61]. São utilizados sensores de som, luz, movimento, CO_2 e humidade relativa. A ocupação de um espaço é determinada a partir de eventos obtidos a partir de cada um dos sensores. Sendo assim cada sensor tem associado a si dois eventos, sendo que cada um está associado a um determinado aumento ou diminuição dos seus valores num determinado espaço de tempo que é diferente para cada sensor.

Este tipo de algoritmo requer um treino intensivo para reconhecer os padrões a partir dos eventos. Além disso os intervalos de tempo de amostragem e a percentagem de variação dos dados dos sensores que definem um evento requerem otimização e poderão divergir quando o sistema for aplicada a outra divisão, tornando a solução pouco universal.

2.4 Soluções comerciais atuais

Esta secção contém um conjunto de solução que existem atualmente no mercado que permitem um controlo de acessos bem como uma monitorização e deteção da ocupação de um dado espaço. Além disso são também descritas algumas patentes existentes e relevantes no caso em estudo.

Uma das soluções existentes e patenteada baseia-se na tecnologia RFID para o controlo de acessos [62]. O sistema proposto é assim composto por uma porta de madeira, um leitor RFID, uma fechadura mecânica e unidade de controlo. Desta forma para obter acesso o utilizador deverá ter em sua posse a etiqueta RFID que autoriza o sistema a abrir a porta. O sistema possui ainda uma UPS (*Uninterruptible Power Supply*) para garantir a alimentação em caso de falha de energia, bem como um sensor que permite saber se a porta se encontra aberta ou fechada.

Esta solução prevê um funcionamento *online*, na medida em que permite a comunicação por mensagens TCP/IP com uma unidade central, onde está alojado o *software* de gestão do sistema que é responsável entre outras coisas pela atualização de bases de dados e dos dados dos utilizadores. Contudo caso haja uma falha na ligação à rede, a unidade de controlo possui uma memória onde ficam armazenados dados, possibilitando o funcionamento *offline* desta solução.

A empresa Triplo W possui atualmente no mercado um sistema de controlo de acessos que permite a utilização de mais do que uma tecnologia [63]. Esse produto trata-se do modelo TW-T16 (figura 2.23 (a)) que traz integrado um teclado *touch* bem como um leitor de cartões RFID. Este dispositivo pode ser programado para efetuar o controlo de acessos apenas por um dos dois métodos (RFID ou senha), ou pode utilizar essas duas tecnologias em simultâneo. A programação dos códigos ou cartões que a fechadura deve aceitar é feita no próprio dispositivo, sendo que para programar é necessária a introdução de um código do sistema, que à partida só o administrador possui.

Este sistema possui algumas desvantagens óbvias, pois funciona *offline*, dificultando dessa forma a alteração das senhas, que obrigaria a que o proprietário do local se deslocasse até à porta, sempre que quisesse alterar a senha de acesso.

A mesma empresa possui também soluções que permitem o controlo de acessos remoto. Para isso têm o módulo GSM RM-100 [64] (figura 2.23 (b)) que tal como o nome indica utiliza rede GSM para trocar mensagens com dispositivos móveis. Este é apenas um módulo adicional que deve ser instalado juntamente com uma fechadura elétrica,

caso esta não exista previamente. A programação do módulo é feita através de uma aplicação móvel disponível para *Android* que possibilita a abertura de porta através de uma chamada efetuada por um dos números de telefone aos quais foi dada autorização pelo administrador, sendo que para isso o utilizador deve possuir também uma aplicação móvel no seu dispositivo.

Este sistema tem o inconveniente de necessitar de um *smartphone* para que o utilizador consiga acesso a um determinado espaço, com a agravante de o seu dispositivo ter obrigatoriamente de possuir um determinado sistema operativo para conseguir ter a aplicação necessária para interagir com o módulo GSM.



Figura 2.23: Soluções da empresa "Tripla W"

A empresa Chaves do Areeiro possui duas soluções que considera apropriadas para o controlo de acesso no setor hoteleiro. Estes sistemas são a fechadura Kaba Oracode [65] e a Kaba Modelo E-790 [66].

O primeiro produto (figura 2.24 (a)) consiste numa fechadura com um teclado integrado, de modo a realizar o controlo de acesso com base numa senha que o cliente deve possuir. A instalação deste sistema é bastante simples, pelo que este não possuiu cabos nem baterias. Possui uma hierarquia de até 8 níveis, o que facilita as autorizações, pois permite criar grupos distintos para funcionários, hóspedes, administradores, etc. No entanto este sistema funciona *offline*, pelo que a automatização de processos pretendida não é possível, pois a programação de senhas do sistema não pode ser feita remotamente.

O segundo produto (figura 2.24 (b)), funciona com base em etiquetas RFID, sendo que após o *check-out* de um cliente a etiqueta a ele associada é cancelada, para segurança do utilizador seguinte. Este produto tem a si associadas as desvantagens já mencionadas à tecnologia RFID. É de salientar que ambos os produtos possuem uma opção de abertura mecânica em caso de falha de energia.



(a) Kaba Oracode [65]



(b) Kaba Modelo E-790 [66]

Figura 2.24: Soluções da empresa "Chaves do Areiro"

Capítulo 3

Proposta de uma nova solução

Este capítulo pretende dar a conhecer detalhadamente ao leitor a solução proposta, que reúne um conjunto de funções encontradas em soluções já existentes, integrando na mesma um conjunto de melhorias face às soluções existentes apresentadas no capítulo anterior. Deste modo o primeiro passo foi escolher uma tecnologia adequada à informatização e automatização do controlo de acessos na indústria do turismo. Pretende-se que o controlo de acessos seja feito sem a intervenção de terceiros, para que o cliente consiga fazer as etapas de *check-in* e *check-out* de forma autónoma, sem que lhe tenha que ser entregues credenciais físicas de acesso por terceiros. Sendo assim as tradicionais chaves de metal são logo descartadas, pois apesar de este ser um método robusto e simples, está dependente da existência de um rececionista para as entregar ao cliente. O mesmo acontece com a tecnologia baseada em etiquetas RFID, cujos cartões têm igualmente de ser entregues ao cliente por alguém, com a desvantagem em relação às chaves metálicas, de que um sistema baseado nesta tecnologia necessita de leitores de etiquetas, contribuindo para um maior custo da solução. Por sua vez, o controlo de acessos baseado na tecnologia de NFC poderia ser uma solução, visto que alguns *smartphones* já estão equipados com este tipo de tecnologia. Contudo o facto de nem todos estarem equipados com etiquetas NFC tornaria a solução pouco universal ao limitar o tipo de utilizadores. Além disso seria necessário o registo prévio do dispositivo no sistema, tornando a solução pouco interessante. O uso de características biométricas também não permite a automatização do processo, visto que os dados biológicos dos clientes teriam de ser previamente recolhidos e inseridos no sistema, podendo o processo ser considerado intrusivo para o cliente, sendo que seriam necessários dispositivos para efetuarem a leitura desses dados. A utilização de códigos de barras ou códigos QR permite a entrega não presencial da credencial de acesso, pois pode ser enviada por email ao cliente um imagem do código com a qual este poderá ter acesso ao espaço. Contudo a geração deste tipo de códigos pode requerer um algoritmo para o processo e além disso os aparelhos de leitura deste tipo de códigos representam um investimento considerável. Uma outra desvantagem é o facto de ser necessário a apresentação de uma uma imagem com o código, cada vez que se queira ter acesso ao local. Posto isto, a tecnologia escolhida foi o teclado alfanumérico, pois a criação automática de senhas é bastante fácil por este processo, sendo que o cliente apenas terá de marcar a senha que recebeu por email para aceder ao espaço. Desta maneira, o teclado e o microcontrolador para leitura do mesmo representam os investimentos adicionais, que são valores inferiores ao custo de leitores de códigos de barras ou de códigos QR, aliando a isso a vantagem de o cliente não ter de carregar consigo

qualquer objeto sempre que deseje aceder ao espaço reservado. Apesar das desvantagens inerentes, a utilização de códigos QR pode ser objeto de um futuro estudo, pois permite a automatização do processo.

3.1 Requisitos do sistema

Após analisadas diversas soluções no capítulo 2, torna-se evidente a falta de uma solução única que cumpra uma série de requisitos no que toca à completa automatização de reservas, controlo de acessos e monitorização de ocupação de um espaço com aplicação na indústria turística. Como foi abordado no capítulo anterior, existem diversas soluções propostas quer no âmbito académico ou comercial, que oferecem algumas das valências pretendidas neste projeto. Contudo, não existe uma solução universal que as englobe a todas. Sendo assim são agora enumeradas todas as especificações que se propõem para esta nova solução.

- **Reserva:** É proposta a criação de um site que permita ao cliente ver todos os quartos disponíveis para determinadas datas de entrada e saída pretendidas e efetuar a sua reserva remotamente, sem sair do conforto do seu lar.
- **Pagamento:** Além de permitir a reserva de um determinado espaço, a interface *Web* deverá também estar associada a uma plataforma de pagamento *online*, para facilitar todo o processo. Sendo assim propõem-se a integração da plataforma *PayPal* para poder realizar os pagamentos remotamente de forma eficiente e segura.
- **Check-in:** Este é um dos pontos fulcrais da solução proposta, pois pretende-se que o processo de *check-in* possa ser realizado de forma automática, sem necessitar da presença de um rececionista que entregue um cartão ou uma chave de acesso ao cliente em questão. Para tal ser possível é necessário escolher uma tecnologia de controlo de acessos adequada. Das abordadas na secção 2.2, a que mais se adequa ao pretendido é o teclado numérico ou alfanumérico, pois permite a criação de um código de acesso diferente para cada reserva, válido apenas no período de estadia da pessoa em questão. Esta tecnologia, juntamente com um módulo de comunicação adequado possibilita a programação automática e à distância de novos códigos. Desta forma, apenas teria de ser enviado um código de acesso ao cliente, não sendo necessário um encontro presencial com o mesmo. Esta tecnologia, permite a geração de códigos aleatórios de uma forma fácil e rápida, o que é uma vantagem face a outras tecnologias, tais como os códigos QR, que necessitam de um algoritmo mais complexo para gerar um código válido.
- **Determinação de ocupação:** Para este requisito é proposta a utilização de um conjunto de sensores, que se considerem não ser intrusivos para o cliente, de modo a que se possa perceber se um determinado espaço está vazio ou ocupado. Como já foi anteriormente referido, tal serviria para otimizar horários de limpeza de quartos, perceber se uma porta foi deixada aberta por engano, ou uma luz foi deixada acesa por lapso. Uma determinação com grande exatidão e em tempo real da ocupação é de grande mais valia, pois permite otimização energética de alguns tipos de sistemas, como equipamentos de AVAC. Em serviços de alojamento com um método de efetuar reservas mais tradicional permite o preenchimento de folhas de

ocupação por parte do proprietário de forma bastante expedita, evitando possíveis erros no preenchimento dessas folhas.

- **Monitorização:** Além de dar a possibilidade de um cliente efetuar a sua reserva, o site criado deverá também permitir acesso a alguma informação, que deverá ser restringida ao administrador. Para tal, este deverá ter uma página protegida com credenciais de email e senha, onde o administrador poderá monitorizar todos os dados relevantes, tais como, ver o histórico de reservas, introduzir ou eliminar uma reserva, ver os preços em vigor e proceder à sua alteração remotamente, bem como ver em qualquer altura, gráficos que mostram os valores lidos nos sensores de um determinado quarto. Deste modo, mesmo que o proprietário não esteja presente, pode ter conhecimento de dados relevantes sobre o ambiente de cada espaço. Deve ainda ter a possibilidade de aferir quais os quartos que se encontram vazios ou ocupados, através da sua área no site.

Torna-se portanto evidente que a solução deverá ser composta por uma aplicação *Web*, que poderá ser consultada por clientes e administrador, sendo que este último deve ter acesso a maior quantidade de informação do que os restantes. Por seu lado, cada quarto deverá ter uma fechadura "inteligente" que consiga comunicar com um servidor central, de modo a saber sempre qual a senha que está em vigor. Deve fazer então, parte do sistema IoT proposto um microcontrolador capaz de comunicar com o servidor e capaz de ler o teclado junto à porta, bem como processar e enviar para a base de dados desse servidor todos os dados recolhidos por um conjunto de sensores. Sendo assim, por uma questão de globalização da solução e porque hoje em dia qualquer espaço turístico possui acesso à *Internet* e está equipado com *routers* de modo a garantir um melhor acesso em todos os locais, foi escolhido o protocolo de comunicação TCP/IP, para garantir uma comunicação entre fechadura e servidor.

Por questões de segurança e para permitir a entrada nos espaços ao proprietário e funcionários, deverá haver uma senha de acesso que seja sempre válida, independentemente do microcontrolador estar ligado à rede ou não, pelo que esta deverá estar na própria memória do dispositivo que controlará a porta e deverá ser apenas do conhecimento do proprietário e funcionários, pois dará acesso ao espaço em qualquer circunstância. Esse requisito possibilitará a abertura da porta, mesmo em casos extremos em que se perca por completo a ligação à rede por tempo indeterminado.

Para diminuir a intervenção humana no processo, deverá ser o próprio sistema a gerar senhas aleatórias de acesso e a notificar via email o cliente que a sua reserva foi efetuada, enviando-lhe nesse mesmo email o código de acesso válido no período para o qual reservou o espaço. A aplicação deverá também perceber se o pagamento efetuado via *PayPal* foi efetuado com sucesso ou não, para apenas notificar o cliente via email e inserir os dados de reserva na base de dados em caso de sucesso.

Este tipo de sistema parte do pressuposto que as reservas são feitas remotamente pelo cliente e o histórico de reservas é guardado em bases de dados automaticamente. Contudo em alguns casos o espaço de serviço turístico funciona de modo mais convencional, isto é, a reserva é feita presencialmente ou por telefone e é o proprietário ou um qualquer funcionário que faz manualmente o registo das reservas, tendo o gestor no final do mês, ao fazer o balanço da contabilidade de conferir as folhas onde esse registo foi feito (semelhante ao da figura 3.1) para verificar quantos dias esteve cada quarto alugado durante o mês. Como são preenchidas manualmente essas folhas podem conter alguns

erros, nomeadamente, algum funcionário poderá ter-se esquecido de preencher uma dada reserva. A pensar nessa situação, surgiu durante a realização desta dissertação a necessidade de modularizar um dos requisitos propostos, que é a deteção de ocupação de um espaço. Desta maneira seria possível aplicar parte da solução em qualquer tipo de serviço de alojamento, independentemente da maneira como seja feito o processo de reserva ou da tecnologia utilizada para controlo de acessos. Assim sendo, o gestor ficaria a saber se em determinado dia o espaço foi alugado ou não.

Este módulo adicional baseia-se num microcontrolador e alguns sensores ligados ao mesmo. Desta maneira seria possível fazer automaticamente um mapa dos dias do mês em que um espaço esteve ocupado, pois com os valores dos sensores será possível determinar em que horas esteve determinado espaço ocupado. Desta maneira torna-se bastante mais fácil para o gestor conferir o balanço da caixa no final do mês, evitando erros de preenchimento dos mapas de ocupação.

De notar que este módulo deverá comunicar também segundo o protocolo TCP/IP (*Internet*) com uma base de dados, e os seus resultados devem ser apresentados de forma gráfica numa página *Web*. O módulo de deteção de presenças está inserido na solução proposta, pelo que é de grande importância para o proprietário perceber se os seus alojamentos estão ocupados ou não, sobretudo quando se pressupõe que na maior parte das vezes este possa estar ausente do local de alojamento, efetuando a gestão e monitorização do espaço remotamente. A decisão de tornar a solução modular prende-se com o facto de que um módulo de deteção de ocupação pode ser de grande relevo noutra tipo de situações, nomeadamente em serviços turísticos que tenham métodos de efetuar as reservas diferentes do proposto. Além disso, pelo levantamento do estado de arte feito neste âmbito, notou-se que a ocupação de um espaço é um parâmetro bastante importante na otimização de gastos de energia com sistemas AVAC, pois a maior parte das soluções propostas tenta resolver esse problema, pelo que as possíveis aplicações deste módulo transcendem a indústria turística, pelo que este poderá ser utilizado em inúmeras circunstâncias, como residências particulares ou qualquer edifício de serviços ou comércio.



Figura 3.1: Mapa de ocupação de um hotel [67]

3.2 Arquitetura da solução proposta

De modo a obter uma solução que cumpra todos os requisitos mencionados anteriormente, é proposta uma solução baseada num sistema IoT com um servidor central, que receberá todas as informações do local e servirá de intermediário para todas as comunicações entre os diversos componentes do sistema proposto. Sendo assim a fechadura

da porta de cada quarto comporta-se como um objeto "inteligente", ou pelo menos um objeto com sentidos, na medida em que estará equipada com um conjunto de sensores que permitirão receber informação em tempo quase real, de modo a que remotamente se possam ter conhecimentos sobre o local de alojamento.

A figura 3.2 mostra em traços gerais a arquitetura pretendida para esta solução. No centro encontra-se o servidor *Web* com o site criado. Para poder ser acessível em qualquer parte do mundo, esse servidor deverá estar ligado à rede global Internet, de modo a ser acessível aos clientes e ao proprietário/administrador através de um *browser* comum.

Todos os dados relevantes, como datas de reserva, preços, dados de clientes, dados de pagamento, entre outros serão armazenados numa base de dados. Para saber qual a senha de acesso válida para o espaço em questão em determinada altura, a fechadura deverá ser capaz de comunicar com a base de dados. Essa comunicação será feita por intermédio do servidor *Web*, na medida em que o sistema composto pela fechadura, microcontrolador e um conjunto de sensores, irá fazer pedidos ao servidor, requisitando uma página *Web* criada para esse efeito, e essa página é que será responsável por inserir na base de dados os valores dos sensores e enviar ao sistema dados relevantes, como datas de reserva e senhas de acesso para cada data.

De modo a que o pagamento da reserva seja efetuado remotamente através de *PayPal*, a aplicação criada deve ser desenvolvida de maneira a permitir a integração deste modo de pagamento, que deverá enviar dados à aplicação acerca de estados do pagamentos, que serão inseridos na base de dados pela própria aplicação.

Esta solução, deverá permitir um elevado grau de automatização de todo o processo. Desta forma, no decorrer deste trabalho, surgiu a necessidade de oferecer ao proprietário a possibilidade de conferir a ocupação do seu estabelecimento, para deste modo ter noção dos acontecimentos que estão a decorrer no espaço que está a alugar. Esta possibilidade adquire especial importância, uma vez que permite um maior controlo e monitorização à distância, o que é essencial, uma vez que com a automatização de todo o processo, passam a ser dispensáveis os serviços de um rececionista, que eventualmente teria também uma tarefa de monitorização e controlo, visto que é o seu papel evitar/controlar acessos indesejados entre outros. Sendo assim tornou-se necessário integrar uma rede de sensores no módulo da fechadura, de modo a perceber se o espaço no seu interior se encontra ocupado ou não. Contudo, como já mencionado anteriormente, alguns espaços de serviço turístico podem optar por ter um sistema de reservas mais conservador, na medida em que em vez de estas serem diretamente inseridas pelo sistema, seria uma pessoa que realizaria manualmente a reserva. Nesses casos, teria de haver o preenchimento de um mapa de ocupação, que no final do mês seria utilizado para verificação da contabilidade. Com vista a evitar lapsos de preenchimento desses mapas, a determinação de ocupação nesses casos pode ser determinante, pois facilita a tarefa do gestor no momento de conferir a contabilidade ao fim de cada mês. Com vista a criar uma solução de deteção de ocupação independente do método utilizado para fazer a reserva ou de controlo de acesso, é proposto um módulo adicional, composto apenas por um microcontrolador e um pequeno conjunto de sensores, de modo a perceber em que dias esteve efetivamente ocupado um determinado espaço, facilitando deste modo a tarefa do gestor. De referir que o módulo adicional poderia ser utilizado independentemente da solução integral. Isto é, é possível aplicar esta solução em espaços que tenham um método de controlo de acessos e de reserva diferentes do proposto. Contudo, quando integrado na solução principal, como mostra a figura 3.2, este módulo permite ao proprietário ter uma ideia muito mais clara

do que se estará a passar no espaço que está a alugar.

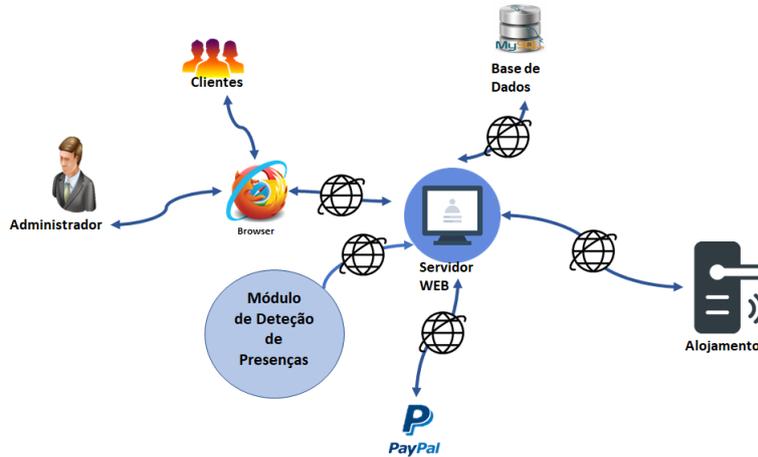


Figura 3.2: Esquema da nova solução proposta

Em seguida, são enumeradas com algum detalhe algumas etapas de funcionamento do sistema, para que se consiga entender exatamente os moldes da solução aqui proposta e de que maneira a que o cliente e administrador podem interagir com a mesma.

Registo no sistema: Antes poder efetuar qualquer reserva no site, o cliente deverá efetuar o seu registo no sistema, preenchendo alguns dos seus dados pessoais, que deverão ser enviados para a base de dados de forma automática. A figura 3.3 mostra a interação entre os vários intervenientes no processo. O cliente deve aceder ao site criado, através do seu *Web browser* e preencher e submeter um formulário de registo com alguns dados pedidos. Após preencher todos os seus dados relevantes, tais como nome, morada, e contactos, a página irá automaticamente adicionar os novos dados numa tabela da base de dados. Após isso, o cliente deve receber um email a confirmar a sua inscrição no sistema. Desse modo, este pode efetuar o seu *login* na página, recorrendo para isso ao seu email como modo de identificação e à inserção de uma senha, escolhida pelo mesmo no momento da reserva como forma de autenticação.

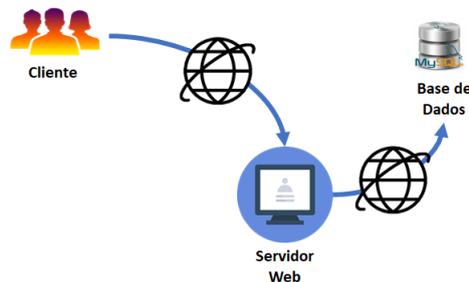


Figura 3.3: Esquema de comunicação para efetuar o registo no sistema

Reserva: Depois de registado no sistema, o cliente já poderá efetuar uma reserva, se assim o desejar. A figura 3.4 mostra esquematicamente como todo o processo decorre. Em primeiro lugar o cliente deve fazer *login* na página a esse fim destinada, preenchendo

o formulário apresentado com o seu email e a senha que este escolheu no momento do registo. Após submissão desses dados, a página *web* irá conectar-se à base de dados, de modo a que esta envie para a aplicação a senha correspondente ao email introduzido. Com essa informação, o site irá comparar a senha válida com a senha introduzida e caso estas sejam iguais, irá redirecionar o cliente para uma nova página. Nesta o cliente poderá escolher que tipo de quarto deseja, bem como as datas de *check-in* e *check-out* desejadas. Mais uma vez, com base nos dados apresentados, a página irá conectar-se à base de dados, para obter informação acerca dos quartos disponíveis entre as datas selecionadas. Em seguida irá ser apresentada uma nova página ao cliente, com todos os quartos que estão disponíveis e preenchem os requisitos por este pretendidos.

Após este processo, o cliente poderá escolher qual o quarto que deseja alugar, sendo que nesse momento a aplicação o deverá redirecionar para uma página de pagamento, enviando os dados relativos ao mesmo (montante de pagamento, quantidade, etc). Nesse momento será apresentada ao cliente uma página do *PayPal* onde este poderá efetuar o seu pagamento *online*. Findo o pagamento, a plataforma de pagamento irá enviar informações acerca do estado do mesmo, que serão enviadas para a base de dados e caso o pagamento tenha sido aceite, é enviado automaticamente um email ao cliente, onde lhe é confirmada a reserva e é fornecida a senha de acesso ao alojamento. Para que a fechadura possa saber quais as senhas de acesso em vigor, a informação relativa à reserva é enviada para a base de dados.

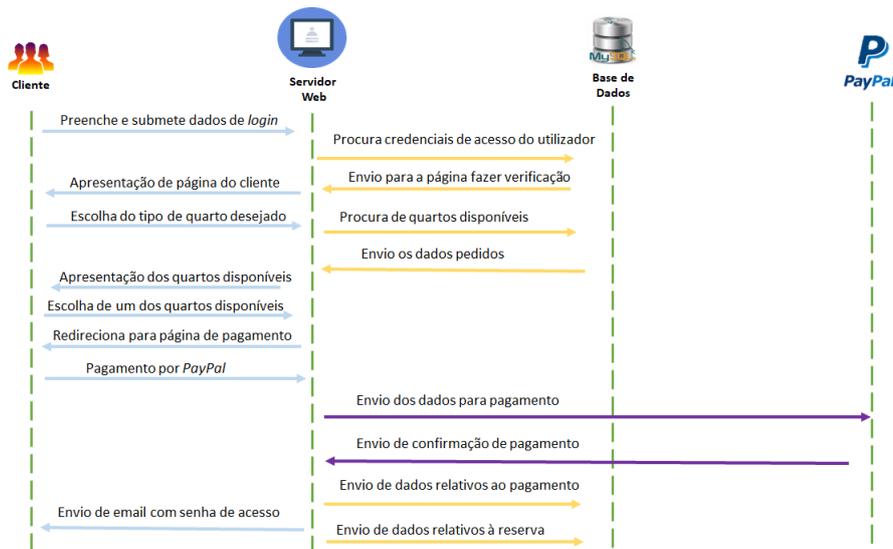


Figura 3.4: Mapa de interação dos intervenientes para efetuar uma reserva

Check-in: No final da fase de reserva, o cliente já terá em sua posse o código de acesso da fechadura válido apenas entre as datas de *check-in* e *check-out* que este escolheu. Posto isto, a figura 3.5 demonstra a interação que existe entre cliente, fechadura, base de dados e servidor *web*. Uma vez junto da porta do local que reservou, o cliente terá apenas de marcar o código de acesso que lhe foi enviado. A fechadura irá então realizar constantemente pedidos ao servidor, requisitando uma página *Web*, que permitirá recolher dados de sensores enviados pela fechadura (que serão inseridos na base de dados). A mensagem deste pedido deverá informar também, qual o quarto que está a realizar

o pedido, para que a aplicação possa conectar-se à base de dados do sistema e enviar para a fechadura a informação correspondente à senha correta naquela altura. Depois de receber essa informação, a fechadura compara a senha inserida com a correta e caso estas sejam iguais, deverá acionar um mecanismo que possibilite a abertura da porta. Caso a senha esteja errada, a porta deverá permanecer fechada.

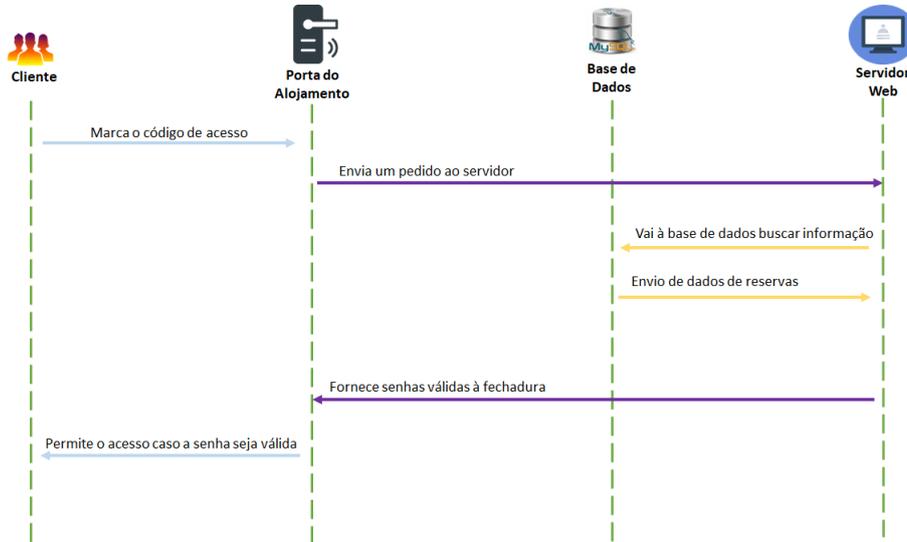


Figura 3.5: Mapa de interação dos intervenientes para efetuar o *check-in*

Módulo adicional para deteção de presenças: Como referido anteriormente, surgiu no decorrer desta dissertação, a necessidade de criar um módulo adicional, que permitisse ao proprietário de um espaço de serviço turístico compreender em que dias esteve efetivamente alugado um determinado espaço. Pretende-se que essa solução possa ser aplicada isoladamente, de modo a evitarem-se erros no preenchimento dos mapas de ocupação. Desta forma no final do mês, seria muito mais fácil para o gestor efetuar a contabilidade.

A figura 3.6 mostra esquematicamente o fluxo de informação proveniente dos vários intervenientes. Prevê-se a existência de um microcontrolador ligado a alguns sensores. Esse controlador deverá ter capacidade para ler os sinais dos mesmos e de os processar, recorrendo para isso a algoritmos. Os sinais processados deverão ser enviados para uma base de dados, para posterior consulta. A determinação de ocupação de um espaço será feita por algoritmos de classificação a partir dos dados processados no microcontrolador, sendo que tanto o sinal processado como a determinação de ocupação são inseridos na base de dados.

De notar que este módulo tem duas vertentes. Uma delas é a sua utilização num espaço com método de reserva tradicional e desse ponto de vista, é útil para auxiliar na gestão, na medida em que permite saber se num determinado dia, o espaço esteve ocupado ou não. Por outro lado, este módulo pode ser utilizado também conjuntamente com o sistema IoT principal que serve para controlo de acessos, na medida em que os dados recolhidos por este módulo permitem ao proprietário um melhor controlo e monitorização do seu espaço de maneira remota.

Como já foi referido anteriormente, pretende-se adotar nesta solução a mesma filosofia

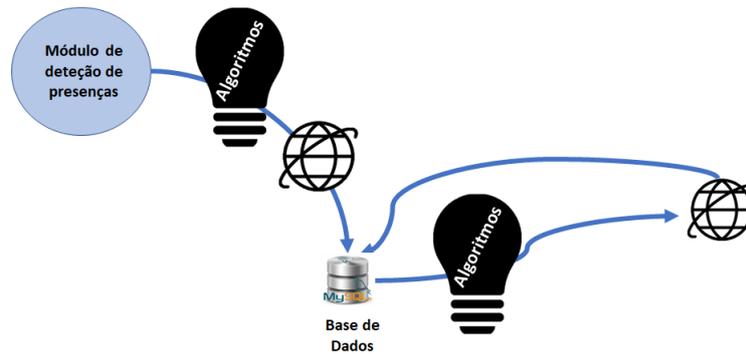


Figura 3.6: Esquema do módulo de detecção de ocupação

abordada no capítulo 2.1.2, de um sistema baseado na *internet* das coisas. Posto isto pretende-se que o módulo da fechadura esteja munido com um número suficiente de sensores que permitam averiguar parâmetros ambientais do sítio em questão, de modo a que haja uma percepção do ambiente que envolve o espaço em questão. Esses dados deverão ser enviados para o servidor *web*, que deverá enviá-los para a base de dados, juntamente com a informação acerca da ocupação do local. Esta decisão deve ser dada por um algoritmo programado em linguagens *web*. A determinação segue uma filosofia de aprendizagem automática (como abordado no capítulo 2.1.3), pelo que devem ser recolhidos alguns dados, para os quais o estado de ocupação seja conhecido, de modo a construir um algoritmo que dê resposta ao problema. De modo a obter uma solução modular, foi utilizado em conjunto com a rede principal de sensores, um módulo adicional, que tal como foi mencionado anteriormente pode funcionar de duas maneiras distintas. Esse módulo engloba o processamento de sinal de um determinado sensor, aplicando-lhe algoritmos, de forma a que os dados enviados por este módulo possam contribuir para a determinação do estado de ocupação de um espaço. Apesar de os algoritmos de detecção de ocupação utilizarem apenas os valores vindos do módulo adicional, os valores da rede de sensores do sistema IoT principal permitem ao proprietário uma ideia muito mais abrangente do ambiente de cada espaço.

A figura 3.7 mostra a interação dos vários intervenientes no processo. A fechadura inteligente irá enviar para o servidor *web* os valores dados pelos sensores, através da rede *internet*. Em conjunto deverá existir um módulo separado, com um conjunto de sensores cujos sinais deverão ser processados e enviados para a base de dados, de modo a que possam ser classificados por um algoritmo, que irá ser responsável por determinar se face aos dados apresentados o espaço se encontra ocupado ou não. O passo final consiste em colocar todos esses dados de ocupação em tabelas da base de dados, para serem consultados sempre que necessário.

Monitorização: Esta função diz respeito sobretudo ao administrador do espaço turístico. A figura 3.8 mostra a interação entre as várias entidades envolvidas no processo. Como se pretende que apenas o administrador tenha acesso aos dados introduzidos no sistema, foi necessário criar uma página, em que apenas este tivesse acesso. Sendo assim, o primeiro passo para monitorizar o sistema será efetuar o *login* com as credenciais de administrador. A página irá à base de dados de modo a ter os dados de acesso corretos para comparação com a senha e email que foram introduzidos. Caso os dados coincidam,

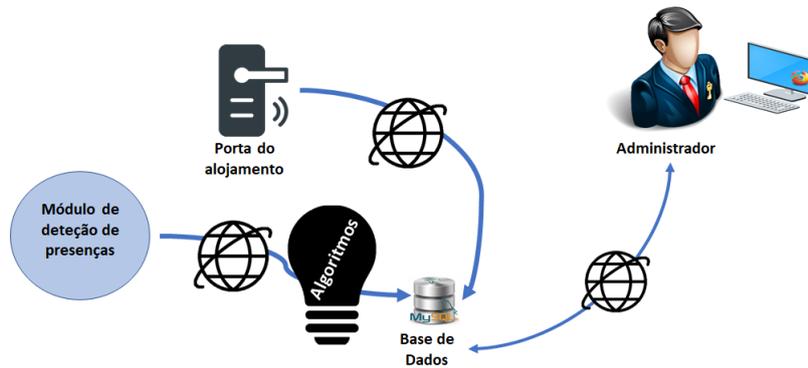


Figura 3.7: Esquema dos intervenientes no processo de determinação de ocupação e monitorização do ambiente de um espaço

o administrador é redirecionado para a sua página, onde poderá escolher quais os dados que deseja ver. Posteriormente a página *web* irá conectar-se às devidas tabelas da base de dados, de modo a que o proprietário seja redirecionado para uma página onde a informação solicitada é mostrada através de gráficos ou tabelas. Deve ser ainda dada ao proprietário a capacidade de poder alterar algumas tabelas da base de dados a partir da página de administrador. Desta forma este poderia cancelar reservas, alterar preços de quartos, remover ou acrescentar uma reserva. Estas funcionalidades facilitariam o acesso ao sistema e a modificação da base de dados, caso haja alguma situação de imprevisibilidade, fazendo com que o sistema possa ser completamente controlado e monitorizado remotamente, em qualquer parte do mundo, desde que se possua um aparelho com um *browser* e ligação à rede *internet*.

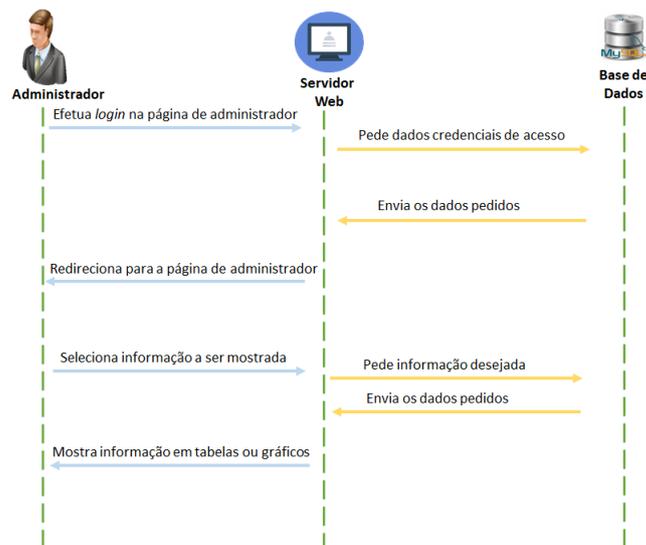


Figura 3.8: Mapa de interação dos intervenientes para o proprietário efetuar a monitorização

3.3 Projeto da base de dados

Uma base de dados consiste num repositório de dados, organizados em tabelas relacionadas entre si. Para gerir e armazenar estas tabelas em disco são utilizados programas denominados de "gestores de bases de dados"[68]. Existe atualmente uma grande quantidade de gestores de bases de dados comerciais, que armazenam em disco os dados de formas diferentes. Contudo, todos ou quase todos eles conhecem e aceitam pedidos realizados por outros programas, desde que esses pedidos estejam de acordo com a linguagem SQL (*Structured Query Language*). Esta linguagem utiliza determinadas instruções, denominadas de *queries*. Desta maneira, praticamente qualquer programa é capaz de consultar, inserir, eliminar ou modificar dados de uma tabela, desde que cumpra com a estrutura da *query* para esse efeito. Nesta dissertação, são as páginas *Web* em PHP as responsáveis por enviar as *queries* para o gestor de bases de dados. Desta forma, estas páginas funcionam como intermediário entre a base de dados e os restantes intervenientes, tais como a fechadura inteligente, o proprietário do espaço a alugar e o cliente.

Cada tabela de uma base de dados pode ser vista como uma relação e cada linha da mesma é designada de registo ou tuplo, sendo que a cada coluna corresponde um atributo. Cada relação é identificada por uma chave primária, que consiste num atributo ou conjunto de atributos que identificam inequivocamente um registo de uma tabela (linha). De uma forma simplista, podemos dizer que a chave primária é uma espécie de B.I de cada registo. Este parâmetro é fundamental, para uma correta manipulação da base de dados. Por exemplo, é necessário identificar claramente um registo, caso se pretenda alterar ou eliminar o mesmo.

Nesta dissertação, a utilização de bases de dados foi fundamental, pois permite que o sistema armazene todos os dados necessários ao seu funcionamento, garantindo um histórico que pode ser consultado a qualquer altura. Posto isto, as principais informações armazenadas na base de dados criada são:

- Dados relativos aos clientes, de modo a que estes estejam identificados no sistema e possam ser contactados caso seja necessário.
- Informações relativas a cada quarto, tais como capacidade, preço, imagens, etc.
- Histórico das reservas efetuadas.
- Histórico dos valores recolhidos por cada sensor.
- Classificação temporal do estado da ocupação dado por três algoritmos diferentes.
- Histórico de dados relativos a pagamentos efetuados via *PayPal*.
- Histórico de acessos e tentativas de acesso a cada espaço.

De modo a projetar a base de dados necessária para a solução proposta neste trabalho, foi necessário atender a alguns requisitos. A base de dados deve ter capacidade para armazenar todos os atributos necessários para a indústria de setor turístico em questão. Todas as tabelas devem estar normalizadas, para ser mais fácil a sua manipulação, pelo que o número de relações deve ser o mais pequeno possível e não deverão existir dados redundantes.

O processo de normalização da base de dados proposta seguiu os seguintes passos:

1. Listagem de todos os atributos necessários para a empresa.
2. Criação de uma relação universal.
3. Divisão em sub-relações, de acordo com a FNBC (Forma Normal de Boyce Codd).
4. Criação das respetivas tabelas.

Os atributos que se consideraram necessários para este trabalho, estão representados na figura 3.9. No que diz respeito ao cliente, é necessário recolher um conjunto de dados sobre o mesmo, como nome completo, morada, número de telefone e endereço email. Estes dados serão obtidos a partir do preenchimento do formulário de registo pelo próprio cliente e são essenciais, na medida em que permitem que haja um contacto por parte da gestão do serviço turístico caso haja alguma eventualidade.

No que diz respeito aos dados de cada quarto, importa ter disponível a informação relativa aos preços por noite, tipo de quarto (suite, individual ou duplo), bem como ter o conhecimento da *password default* de cada um. Este parâmetro consiste numa senha de acesso que estará na própria memória da fechadura e será válida em qualquer altura. Isto permite ao proprietário ter acesso ao espaço sempre que deseja, ou poderá servir também de senha de acesso para os funcionários que tratam da limpeza e manutenção do espaço, para que se evite o fornecimento constante de diferentes senhas aos funcionários.

Convém ainda ter um histórico de reservas, para se saber em que datas esteve cada quarto reservado e qual o cliente que procedeu a sua reserva. Como este sistema oferece uma forma de pagamento *online* via *PayPal*, interessa armazenar um histórico de dados relativos a pagamentos, tais como ID de pagamento, ID do pagador e estado do pagamento. Estes dados são fornecidos pelo próprio *PayPal*, sendo que importa além destes saber qual o cliente que efetuou o pagamento, a data em que o mesmo foi efetuado e o montante envolvido.

É de grande importância ter um histórico de dados relativos aos valores dos sensores, para dessa forma o proprietário poder monitorizar da melhor forma cada espaço. Com vista a ter ainda mais informação acerca do local, convém ter um histórico de acessos ou tentativas de acesso ao local. Desta maneira o proprietário ficaria a saber se alguém entrou, introduzindo a senha correta (distinguindo se foi dado acesso com a senha de cliente ou com a *password default*) e ainda saber se houve tentativas de acesso mal sucedidas (marcação de códigos inválidos).

Para entender a maneira como os vários atributos estão relacionados entre si, foi criado um DDF (Diagrama de Dependências Funcionais) com todos os atributos, criando assim uma relação universal. Essa relação está ilustrada na figura 3.10, sendo que cada seta do diagrama pode ser lida como "existe um e um só".

Cada cliente pode ser identificado inequivocamente pelo seu ID, que consiste num número de cliente e é único para cada identidade. Desta forma para cada ID existe um e um só nome de cliente, morada, email, número de telefone. A senha de *login* é o parâmetro de validação para entrar na página pessoal da aplicação de modo a poder efetuar uma reserva. Na plataforma de reservas convém distinguir o proprietário do administrador, daí ser criado o atributo estatuto, para que apenas o administrador possa aceder à sua página, ficando alguns dados restritos unicamente a essa entidade. Enquanto que o atributo "Estatuto" é preenchido automaticamente com "cliente" no ato do registo, as credenciais de administrador tiveram de ser introduzidas diretamente no gestor de bases de dados *MySQL*.



Figura 3.9: Listagem de atributos necessários na base de dados

No que toca ao histórico de reservas, para identificar inequivocamente um registo, são necessários 3 atributos, a data de *check-in* e *check-out* e o número do quarto (ID Quarto) em questão. Para cada reserva efetuada, existirá apenas um cliente que a efetuou e apenas uma senha de acesso válida para aquele período de tempo.

Para cada quarto existe um e um só "Tipo", uma só "*password default*" e uma só uma só imagem. O atributo "imagem" corresponde à localização em disco de cada uma da imagem que identifica cada quarto na plataforma de reservas. Relativamente ao preço por noite de cada quarto, em alguns casos, esse preço depende da ocupação. Isto é, um quarto duplo pode ser mais barato se for ocupado por apenas uma pessoa. Posto isto, para identificar inequivocamente um preço, são necessários os atributos "ID Quarto" e "Nº de ocupantes". Caso o proprietário opte por praticar os mesmos preços independentemente da ocupação, tal é possível, bastando colocar o mesmo preço na tabela, seja qual for o número de pessoas.

Relativamente aos dados de pagamento, podem ser identificados inequivocamente pelo "ID pagamento" que é atribuído pelo *PayPal*, sendo que para cada compra existe apenas um pagador, uma data de pagamento e um montante envolvido, sendo ainda importante identificar o cliente que efetuou o pagamento e o estado do mesmo. O estado do pagamento indica se este foi ou não aprovado pelo *PayPal*, contudo apenas serão inseridos os dados no sistema caso este tenha sido aprovado, caso contrário é dada indicação ao utilizador que o seu pagamento não foi aprovado.

Por último, importa ter registo de todos os valores dos sensores, bem como registar todos os acessos ou tentativas de acesso a um local, tendo informação de qual o local

onde estão a ser medidos, bem como da data e hora a que a medição foi realizada. Neste caso a data de leitura e o número do quarto identificam inequivocamente um registo. De frisar que o atributo "Data e hora" contém a data em que a leitura foi feita, bem como a hora, tudo no mesmo atributo. Importa referir que os dados do sensor de som não serão enviados ao mesmo tempo dos restantes sensores, pelo que a medição estará a ser feita por outro microcontrolador. Posto isto, para uma melhor manipulação de dados, é de interesse colocar essas leituras em tabelas diferentes. Os atributos "Ocupação SVM", "Ocupação Naive" e "Ocupação DT" são a previsão dos algoritmos de SVM, Naive-Bayes e árvores de decisão para o estado de ocupação de um determinado quarto, como base nos valores do sensor de som. Estes parâmetros e os valores do sensor de som são inseridos ao mesmo tempo na base de dados, não havendo à partida qualquer problema desta informação ficar na mesma tabela.

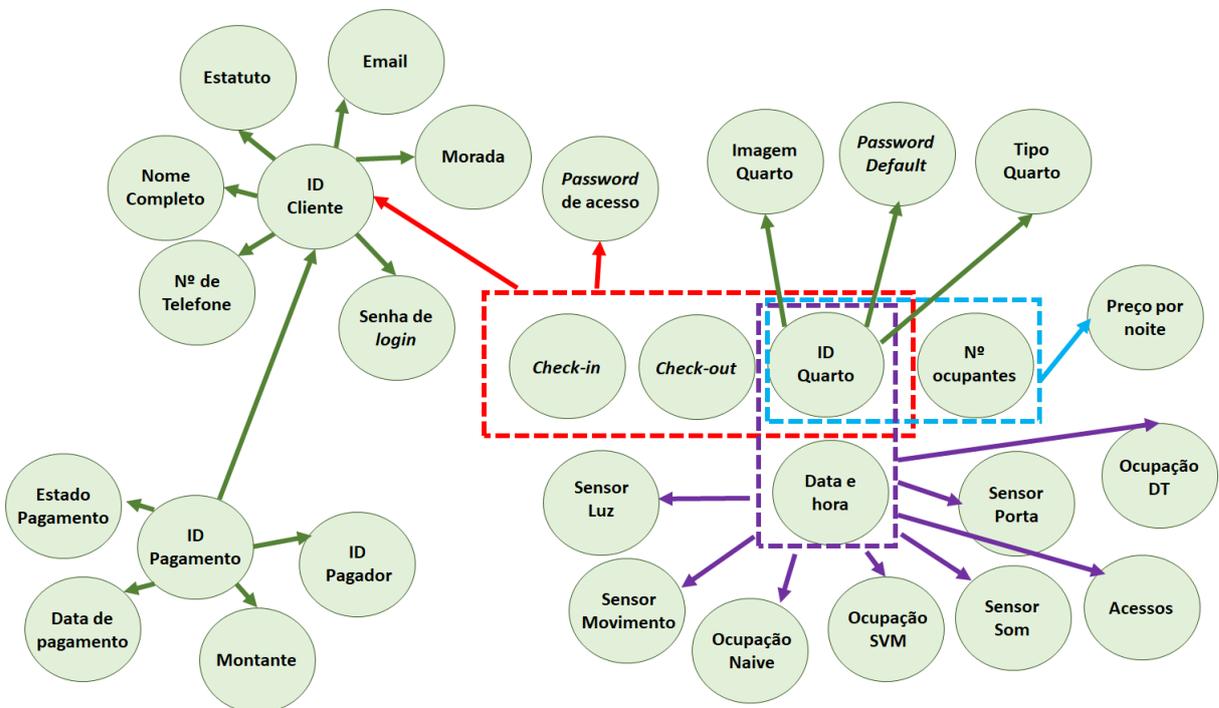


Figura 3.10: Relação universal para a base de dados criada

Como é perceptível pela figura 3.10, o elevado número de atributos faz com que seja praticamente impossível de organiza-los todos na mesma tabela. Posto isto, surge a necessidade de dividir a relação universal em várias relações mais pequenas, de modo a ser mais fácil aceder aos dados pretendidos de forma rápida e expedita. Desta forma, foi utilizada a FNBC (Forma Normal de Boyce Codd) para normalizar as várias relações. Segundo esta norma, uma relação está normalizada se todos os determinantes de uma relação forem chaves candidatas. Entende-se por chave candidata um atributo, ou conjunto de atributos que permitam determinar todos os restantes. Determinante de uma relação é apenas um atributo ou conjunto destes que determina um outro atributo, não identificando necessariamente todos os parâmetros presentes na relação universal.

Na relação universal (R0), cujos atributos estão representados na figura 3.9 temos o

seguinte:

Determinantes:

1. <ID Quarto, *check-in*, *check-out*> 2. <ID Cliente> 3. <ID Quarto> 4. <ID Pagamento> 5. <ID Quarto, Data e hora> 6. <ID Quarto, N^o Ocupantes>

Chave Candidata:

<ID Quarto, *check-in*, *check-out*, ID Pagamento, Data e hora, N^o de ocupantes>

Como é possível verificar, esta relação não está de acordo com a norma FNBC, pois nem todos os determinantes da relação são chaves candidatas. Tornou-se então necessário dividir a relação universal em várias relações mais pequenas que obedecessem a esta norma. Este processo foi feito por iterações, até que se obtivessem relações em que todos os determinantes fossem chaves candidatas. Adianta-se desde já que foram criadas 7 relações normalizadas que deram origem a 7 tabelas diferentes na base de dados.

A primeira relação normalizada a ser encontrada foi a relação R1, cujo DDF se encontra ilustrado na figura 3.11. Para este caso temos o seguinte:

R1: (ID Cliente, Estatuto, Email, Morada, Senha de *login*, N^o de telefone, Nome Completo)

Determinantes:

<ID Cliente>

Chave Candidata:

<ID Cliente>

Todos os determinantes são chaves candidatas, logo a relação está normalizada. Desta relação surgiu a tabela 3.1 que terá um número de linhas igual ao número de clientes registados no sistema.

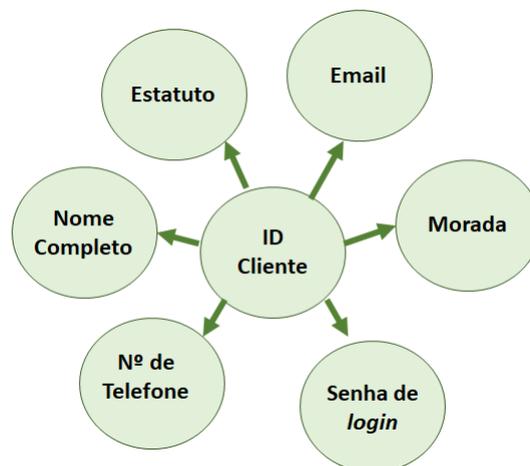


Figura 3.11: Diagrama de dependências funcionais da relação R1

Tabela 3.1: Dados dos clientes

ID cliente	Estatuto	Email	Morada	Senha de <i>login</i>	N ^o Telefone	Nome Completo

O DDF para a segunda relação normalizada está representado na figura 3.12 e nesta situação temos:

R1: (ID Pagamento, ID Cliente, ID Pagador, Montante, Estado Pagamento, Data Pagamento)

Determinantes:

<ID Pagamento>

Chave Candidata:

<ID Pagamento>

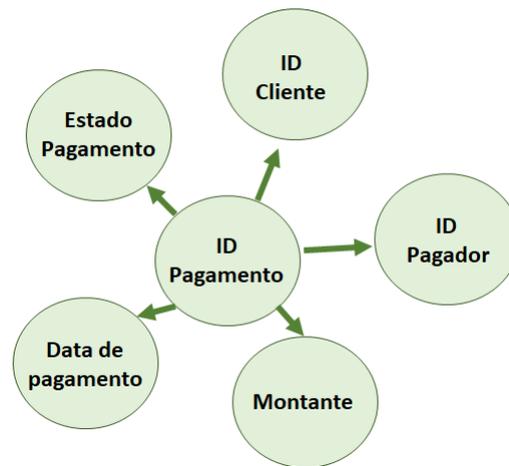


Figura 3.12: Diagrama de dependências funcionais da relação R2

Estando esta relação normalizada, pelos pressupostos já mencionados, obteve-se a tabela 3.2, que terá um número de linhas igual ao número de pagamentos efetuados por *PayPal*.

Tabela 3.2: Dados relativos a pagamentos

ID Pagamento	ID Cliente	ID Pagador	Montante	Estado Pagamento	Data Pagamento

Para a relação R3, representada na figura 3.13, temos:

R3: (ID Quarto, *Password Default*, Imagem Quarto, Tipo Quarto)

Determinantes:

<ID Quarto>

Chave Candidata:

<ID Quarto>

Daqui resulta a tabela normalizada 3.3.

Tabela 3.3: Dados relativos a cada quarto

ID Quarto	<i>Password Default</i>	Tipo Quarto	Imagem Quarto

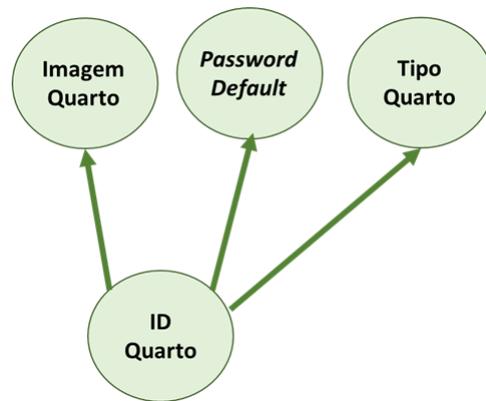


Figura 3.13: Diagrama de dependências funcionais da relação R3

Como o preço por noite de cada quarto pode depender do número de ocupantes, houve a necessidade de criar a relação R4, representada na figura 3.14.

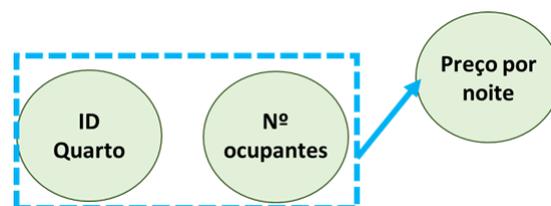


Figura 3.14: Diagrama de dependências funcionais da relação R4

Daqui se retira:

R4: (ID Quarto, N^o Ocupantes, Preço por noite)

Determinantes:

<ID Quarto, N^o Ocupantes>

Chave Candidata:

<ID Quarto, N^o Ocupantes>

Obtendo-se a tabela 3.4.

Tabela 3.4: Preços por noite

ID Quarto	N ^o Ocupantes	Preço por noite

Relativamente ao histórico de reservas, foi criada a relação normalizada R5, representada na figura 3.15.

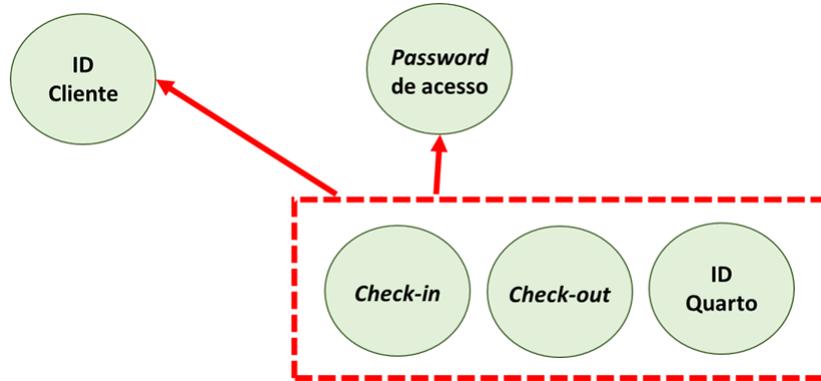


Figura 3.15: Diagrama de dependências funcionais da relação R5

Os parâmetros desta relação são os seguintes:

R3: (ID Quarto, *Check-in*, *Check-out*, ID Cliente, *Password de acesso*)

Determinantes:

<ID Quarto, *Check-in*, *Check-out*>

Chave Candidata:

<ID Quarto, *Check-in*, *Check-out*>

De onde se criou a tabela 3.5

Tabela 3.5: Histórico de reservas

ID Quarto	<i>Check-in</i>	<i>Check-out</i>	<i>Password de acesso</i>	ID cliente

Por último, falta apenas criar relações normalizadas para os valores obtidos pelos sensores. Houve a necessidade de criar relações diferentes para este tipo de dados, visto que eles são provenientes de microcontroladores diferentes. Posto isto criaram-se as relações R6 e R7 representadas na figura 3.16.

Para a relação R6 temos:

R6: (ID Quarto, Data e hora, Sensor Porta, Sensor Movimento, Sensor Luz, Acessos)

Determinantes:

<ID Quarto, Data e hora>

Chave Candidata:

<ID Quarto, Data e hora>

Daqui resultou a tabela 3.6. De notar que o atributo "Acessos" poderá tomar quatro valores diferentes. Pode ter o valor "Sem acessos", se entre cada pedido do ESP ao servidor não houver nenhuma tentativa de acesso. Caso tenha sido concedido acesso desde a última ligação à base de dados, o valor pode ser "Acesso cliente", caso o acesso tenha sido dado por uma senha normal ou "Acesso default" caso o acesso tenha sido dado

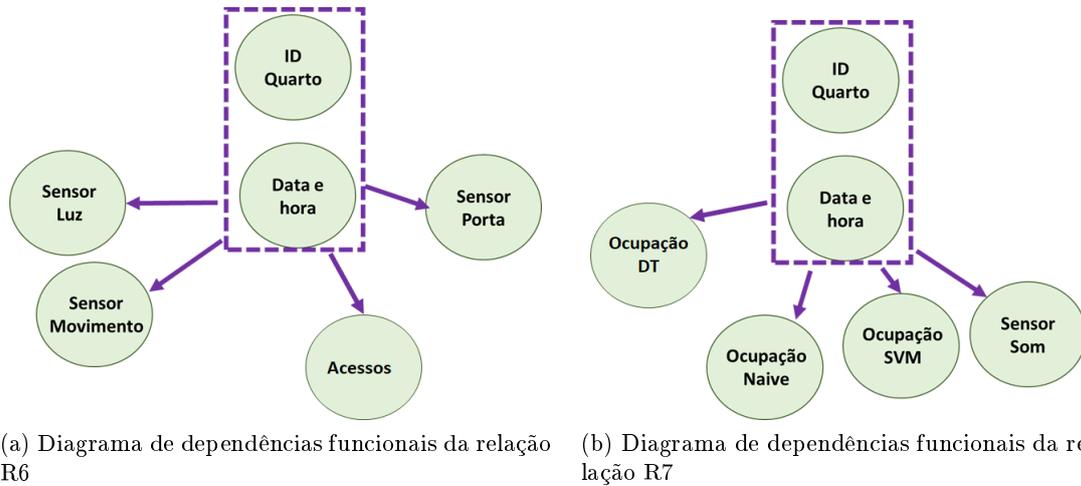


Figura 3.16: Atributos relativos a dados dos sensores

com a marcação da *password default*. Caso tenha havido uma tentativa não sucedida de acesso, este atributo terá o valor de "Acesso negado".

Tabela 3.6: Histórico dos valores dos sensores do ESP32

ID Quarto	Data e hora	Sensor Movimento	Sensor Porta	Sensor Luz	Acessos

Enquanto que para a relação R7:

R7: (ID Quarto, Data e hora, Sensor Som, Ocupação Naive, Ocupação SVM)

Determinantes:

<ID Quarto, Data e hora>

Chave Candidata:

<ID Quarto, Data e hora>

De onde resultou a tabela 3.7.

Tabela 3.7: Histórico dos valores do sensor de som e determinação de ocupação

ID Quarto	Data e hora	Sensor Som	Ocupação Naive	Ocupação SVM	Ocupação DT

Em suma, as 7 tabelas criadas e respectivas colunas de cada uma foram:

- Histórico de Reservas (ID Cliente, *Check-in*, *Check-out*, ID Quarto, *password de acesso*).
- Dados de cliente (ID Cliente, Morada, Senha de *login*, Nome completo, N^o telefone, Email, Estatuto).
- Dados dos Quartos (ID Quarto, Imagem de quarto, *Password default*, Tipo de quarto).

- Preços (ID Quarto, Nº de ocupantes, Preço por noite).
- Histórico de sensores ESP32 (ID Quarto, Data e hora, Sensor porta, Sensor luz, Sensor movimento, Acessos).
- Histórico do sensor de som (ID Quarto, Data e hora, Sensor som, Ocupação SVM, Ocupação Naive, Ocupação DT).
- Histórico módulo de detecção de ocupação (ID Quarto, Data e hora, Sensor som, Ocupação SVM, Ocupação Naive, Ocupação DT).

Capítulo 4

Implementação da solução proposta

Este capítulo pretende dar a conhecer ao leitor todas as etapas necessárias para a implementação da solução proposta no capítulo anterior. Sendo assim, para uma melhor compreensão, esta parte está dividida em 4 partes essenciais:

- *Arquitetura*: Nesta secção é apresentada detalhadamente a solução proposta. São analisados os principais componentes da mesma, bem como os meios de comunicação entre os equipamentos.
- *Hardware*: Aqui são descritos detalhadamente todos os componentes integrantes da solução. São analisados os diagramas elétricos do sistema proposto e é descrito o funcionamento dos sensores utilizados, bem como qualquer outro componente relevante.
- *Software*: Nesta parte é explicado o funcionamento da interface *web* desenvolvida, bem como dos programas dos microcontroladores utilizados. É ainda abordada com detalhe a base de dados criada para armazenar toda a informação necessária.
- *Deteção de presenças*: Nesta parte é exposto em detalhe o método utilizado para realizar a deteção de ocupação de um espaço. São detalhados os algoritmos de aprendizagem automática utilizados e é feita uma análise crítica aos resultados obtidos utilizando o módulo proposto.
- *Montagens finais*: São mostradas as montagens dos vários módulos propostos e sua implementação em ambiente real.

4.1 Arquitetura da solução implementada

Conforme mencionado no capítulo anterior, existe uma parte da solução que se pretende modular. Nesse sentido, foram utilizados dois microcontroladores diferentes, o ESP8266 e o ESP32. Ambos os equipamentos possuem uma *shield Wi-Fi*, pelo que é possível liga-los a uma rede sem fios de modo a que estes comuniquem com

outros equipamentos. Para que tal aconteça, é necessário que exista no local um ou mais *routers/acces points*, para que se estabeleça ligação à rede.

A figura 4.1 mostra esquematicamente os principais componentes deste sistema, bem como a interação entre eles. O sistema IoT é baseado numa arquitetura central, com apenas um servidor, onde estarão armazenadas todas as informações provenientes dos microcontroladores, numa base de dados e onde estará também a interface *web* desenvolvida. Este sistema de arquitetura central, foi pensado de modo a que um serviço turístico tenha em cada um dos seus espaços a alugar uma fechadura inteligente, composta por um microcontrolador ESP32, que tem ligados a si sensores de movimento, luz e um sensor de porta, de modo a que possa fornecer informações sobre o ambiente de cada espaço. Este dispositivo será também responsável pelo controlo de acessos propriamente dito. Para isso tem conectado a si um teclado alfanumérico que servirá para um determinado cliente possa inserir o seu código de acesso. Além disso a abertura da porta será feita com base na ativação de um relé, por parte do microcontrolador, que permitirá abrir ou fechar o fecho elétrico da porta. Para a determinação de ocupação, faz parte da solução um módulo adicional composto por um microcontrolador ESP8266 e por um sensor de som, cujos sinais, depois de devidamente processados serão enviados para uma base de dados, periodicamente.

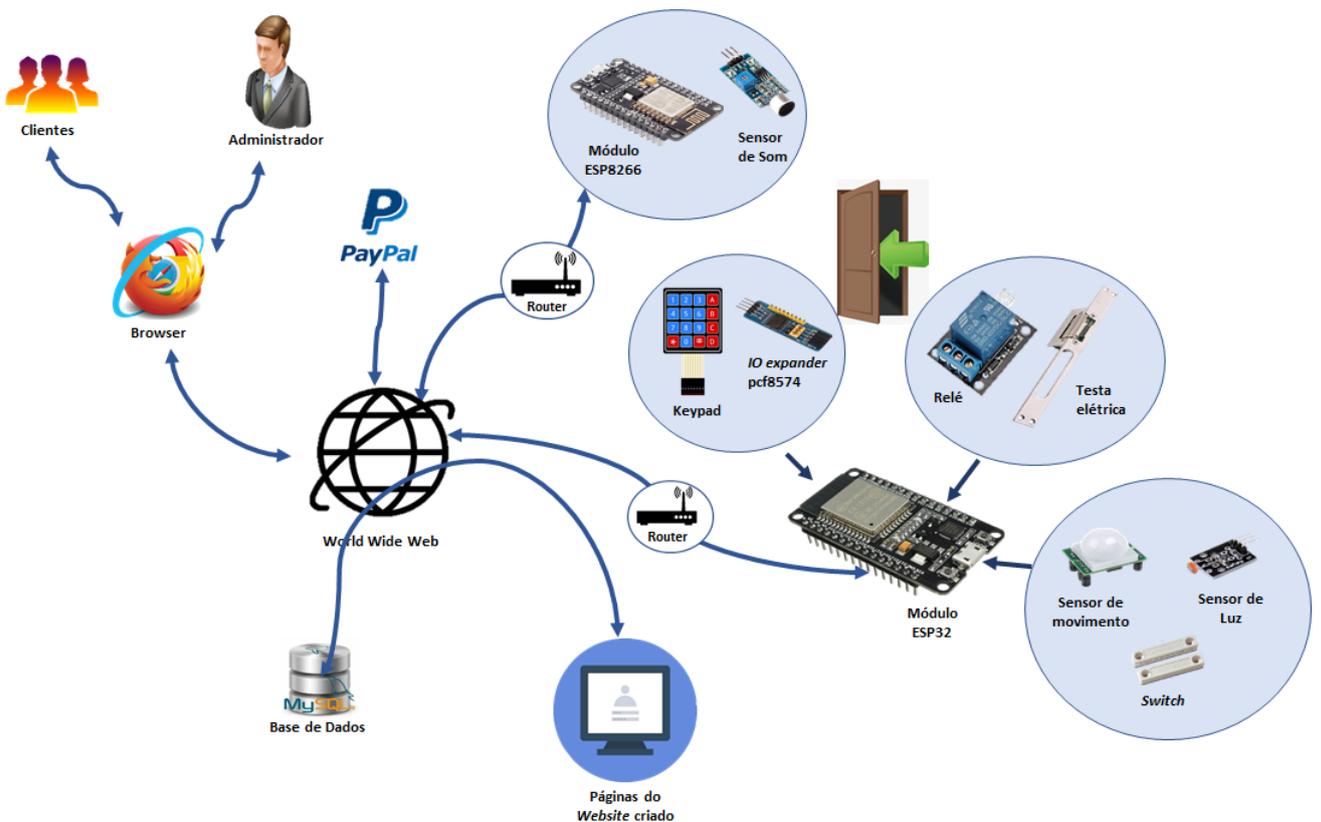


Figura 4.1: Esquema de implementação da solução proposta

4.1.1 Comunicação entre fechadura e servidor

O pretendido para o sistema é que este funcionasse com uma comunicação sem fios, com uma ligação à rede *internet*, para que a comunicação com o sistema pudesse ser feita em qualquer local. Posto isto poderiam ter sido utilizadas redes GSM ou GPRS, tal como utilizado nas soluções propostas por [52] e [64]. Este tipo de comunicação é utilizada pelas operadoras móveis, para permitir o acesso à *internet* a partir de um telemóvel. Para tal é necessário um cartão SIM de uma operadora móvel, acarretando para além dos custos de aquisição desse cartão, custos associados ao consumo de dados móveis pelo sistema. Uma outra alternativa seria a utilização de rede *Wi-Fi*, como nas soluções propostas por [46] e por [48]. Para aceder à *internet* a partir desta rede, são necessários *routers/access points* para permitir que dispositivos geograficamente próximos destes se possam ligar em rede. Atualmente, a maior parte dos serviços da indústria turística já possui nas suas instalações esses dispositivos, para permitir o acesso à *internet* em todo o espaço, pelo que se considerou esta alternativa a mais viável para estabelecer comunicação entre os diversos dispositivos, pois permite a instalação do sistema, sem que seja necessário adquirir cartões de telemóvel (SIM), ou ficar sujeito às tarifas de faturação dessa operadora móvel.

Como já referido, esta solução segue uma arquitetura cliente/servidor, com um servidor central. De um modo geral existem vários clientes, visto todos os microcontroladores desempenham esse papel, sendo que o papel de servidor é desempenhado por um computador. Para uma correta comunicação entre estes equipamentos, foi utilizado o protocolo HTTP, que utiliza o protocolo TCP/IP para troca de mensagens.

Este protocolo define um conjunto de interações entre as aplicações clientes e as aplicações servidoras, bem como a sintaxe de um conjunto de mensagens que todos os clientes e servidores HTTP reconhecem e sabem processar [69]. De facto, a grande vantagem deste protocolo, face a outros já existentes para troca de ficheiros, é que este veio criar uma forma bastante fácil e intuitiva de identificar e localizar um documento. Essa forma é baseada na utilização do URI (*Uniform Resource Identifier*). De um modo geral, podemos dizer que o URI é uma mensagem com a localização de um ficheiro no computador de destino. Para ter toda a informação acerca da localização do ficheiro, essa mensagem deverá conter também o nome do computador (endereço de IP), bem como a porta do computador de destino que se está a aceder. A figura 4.2 mostra a estrutura de que de uma URI, sendo que no início da mensagem terá de ser explícito o protocolo que está a ser utilizado (HTTP, neste caso).

Protocolo://computador[:tepport number]/localização do documento dentro do computador de destino
[? Paramentos do pedido] [# ancora]

Figura 4.2: Composição da URI [69]

A figura 4.3 mostra como é feita a comunicação entre clientes e servidor. Estando os microcontroladores a atuar como aplicação cliente, são eles quem iniciam a

comunicação, requisitando ao servidor um ficheiro contido no mesmo. Para que isso aconteça, os clientes devem estar ligados ao servidor numa porta virtual (ligação TCP/IP). Depois disso qualquer cliente que esteja ligado ao servidor poderá enviar uma mensagem HTTP do tipo GET (1). Esta mensagem requisita um determinado ficheiro localizado no servidor (2). Sendo que esse ficheiro tem a extensão php, ao ser chamado pelo cliente, este irá executar uma série de tarefas:

- Recolha de dados: A troca de informação entre cliente e servidor não se processa num único sentido, na medida em que na composição da URI, é possível incluir o envio de dados pelo método GET no php, deste modo, é possível obter dados dos microcontroladores.
- Processamento e armazenamento de dados: Após obter os dados vindos da aplicação cliente, o ficheiro chamado poderá também ter a função de processar dados, recorrendo a algoritmos. Isso acontece na determinação de ocupação, os dados chegam ao ficheiro requisitado e são processados por algoritmos que retornam o estado de ocupação de um local, sendo que esses dados serão depois armazenados em bases de dados(3).
- Envio de dados ao cliente: Após receber os dados provenientes do cliente, o ficheiro requisitado irá conectar-se à base de dados (4) e enviar para o cliente a resposta (5), que consiste num ficheiro que contém os dados das reservas para o quarto do cliente *Web* em questão. Dessa forma, o ESP32 saberá sempre quais as senhas válidas.

De notar que o fluxo bi-direcional de informação acontece com os dois microcontroladores, pois o servidor responde ao cliente com uma cópia do ficheiro requerido. Contudo, no caso do ESP8266 a resposta serve apenas para informar que o pedido foi efetuado, pois este microcontrolador não requer qualquer informação adicional do servidor para desempenhar a sua função. Importa ainda referir que após a receção da resposta por parte do cliente, a ligação TCP/IP é imediatamente terminada.



Figura 4.3: Esquema de comunicação cliente/servidor da solução proposta

4.2 *Hardware*

4.2.1 Microcontroladores utilizados

Tal como já foi anteriormente mencionado, foram utilizados dois microcontroladores diferentes, o ESP32 e o ESP8266. Um dos principais motivos que levou à escolha destes dispositivos foi a sua reduzida dimensão, aliada ao facto de possuírem uma *shield Wi-Fi* integrada, possibilitando a sua ligação ao servidor através da rede *Wi-Fi*, segundo o protocolo IEEE 802.11. No que diz respeito ao ESP32, este tem as seguintes características técnicas: [70]

- *Wi-Fi*: Conexão *Wi-Fi* 802.11 (2.4 GHz), até 150 Mbps
- Bluetooth: BLE (*Bluetooth Low Energy*)
- Processador: Tensilica Xtensa Dual-Core 32-bit LX6 microprocessador de até 240 MHz (o mais comum são 160 MHz);
- ROM: 448 KB
- SRAM: 520 KB
- Pinos (GPIO): Inclui ADCs (Conversor Analógico para Digital de 12 bit), DACs (Conversor Digital para Analógico), I2C, UART (Recetor / Transmissor Assíncrono Universal) , SPI (Interface Periférica Serial), I2S (*Integrated Interchip Sound*), e PWM (Pulse Width Modulation).
- Tensão de alimentação: Cerca de 3,3V com possibilidade de ser alimentado por 5V devido ao seu regulador de tensão interno.
- Corrente: Consumo de corrente máximo de 240 mA [71].
- Programação: É compatível com o Arduino IDE, um ambiente de programação para arduinos, que possui algumas bibliotecas, facilitando assim o trabalho do programador.

Este dispositivo disponibiliza um elevado número de pinos que podem ser utilizados como entradas/saídas digitais ou analógicas, possibilitando que esteja ligado a este dispositivo uma grande rede de sensores e outros dispositivos, pelo que nos dias que correm esta é uma placa bastante utilizada para projetos IoT. Esta, tal como o ESP8266 traz ainda embutida uma porta micro USB, de modo a interagir com um computador, sendo a comunicação realizada por USB-Serial, de acordo com o protocolo de comunicação RS232, de modo o programa seja transferido para o microcontrolador. A porta micro USB pode ainda servir para alimentação da placa com 5V, se esta estiver ligada um dispositivo que lhe forneça corrente, como por exemplo um transformador que forneça 5V através de uma porta USB ou a uma *powerbank*. Para garantir uma alimentação correta, faz parte da placa um regulador de tensão que converte os 5V fornecidos em 3,3V, que é a tensão disponibilizada na maior parte dos pinos (existem alguns pinos que fornecem 5V). Além de todas as características técnicas que possui, esta placa tem ainda como vantagem o seu preço bastante apelativo, sendo que uma destas unidades custa cerca de 5 euros [72].

Na figura 4.4 estão representados os periféricos conectados ao ESP32 na solução proposta. Estes têm as seguintes funções:

- Relé opto-isolado: Através deste módulo, é realizado o controlo de abertura do fecho elétrico da porta, por uma saída digital da placa. O facto de ser opto-isolado, protege o microcontrolador das elevadas correntes e tensões necessárias para a abertura da porta (cerca de 12V e 1A).

- LED vermelho: É ativado quando deteta que foi introduzida uma senha de acesso, mas esta era inválida. Serve de indicador visual para o utilizador.

- LED amarelo: Serve para sinalizar que foi detetada uma tecla. Desta forma um cliente terá a perceção que a tecla que marcou foi reconhecida, pelo piscar deste LED.

- LED verde: Tal como os anteriores, é utilizado para dar uma indicação visual ao utilizador, neste caso de que a senha que introduziu foi aceite.

- Sensor de movimento: Ligado a uma saída digital do microcontrolador, este sensor indica se foi ou não detetado algum movimento no interior de uma divisão (nas proximidades do mesmo).

- Sensor de luz: Sensor analógico, que permite saber o nível de luminosidade de um local.

- Sensor de porta: Dá indicação sobre o estado da porta (aberta ou fechada).

- Módulo RTC 1302: Este módulo utiliza 3 pinos para a comunicação com o microcontrolador e serve essencialmente para indicar a data e hora atuais ao mesmo, mesmo que o sistema por alguma razão fique *offline*.

- PCF8574: Este periférico é um *IO expander* e serve essencialmente para efetuar a comunicação entre o teclado alfanumérico e o ESP32. Desta maneira são utilizados apenas dois pinos para realizar a comunicação, ao invés de utilizar 8 entradas digitais do microcontrolador.

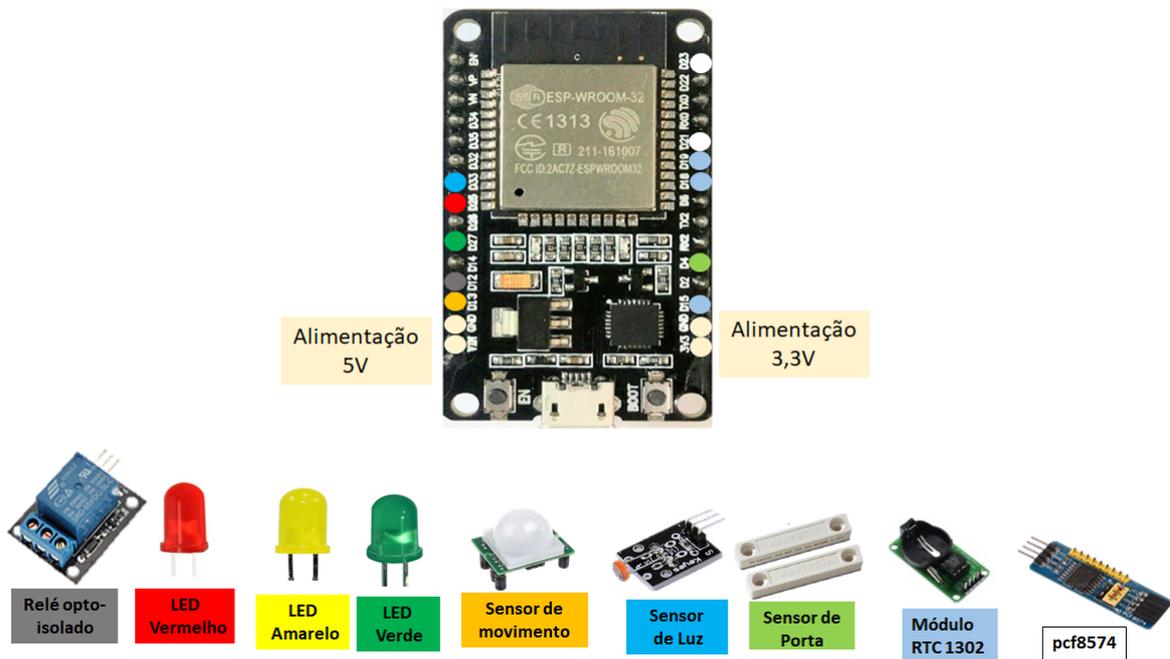


Figura 4.4: Esquema de ligação de periféricos à placa ESP32

No que diz respeito ao módulo adicional para detecção de ocupação, este é composto por um microcontrolador ESP8266 e por um sensor de som, como mostra a figura 4.5. A placa utilizada possui as seguintes características técnicas: [73]

- Pinos: Tem conectores GPIO, barramentos I2C, SPI, UART, entrada ADC, saída PWM e sensor interno de temperatura
- Processador: Standard LX3 da Tensilica, opera em 80MHz, com possibilidade de operar em 160MHz.
- RAM:32KBytes para instruções e 96KBytes para dados.
- ROM:64KBytes.
- Memória flash: SPI Winbond W25Q40BVNIG de 512KBytes.
- Programação: Compatível com Arduino IDE.
- ADC de 10 bit.

Este microcontrolador possui à partida características técnicas inferiores às do ESP32, contudo, este módulo tem apenas um periférico conectado a si, pelo que é necessária apenas uma entrada analógica. Além disso como as necessidades de processamento serão também menores, não houve necessidade de utilizar uma nova placa ESP32, por isso a decisão recaiu por utilizar uma placa de menor complexidade e mais económica (custo aproximado de 3 euros [74]).

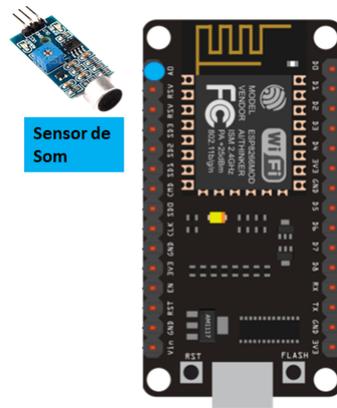


Figura 4.5: Esquema de pinos da placa ESP8266

4.2.2 Teclado alfanumérico

Como referido no capítulo anterior, no que diz respeito à escolha de uma tecnologia para o controlo de acessos, optou-se pela utilização de um teclado alfanumérico, como mostra a figura 4.6. Este dispositivo possui as seguintes características: [45]

- *Design* ultra fino.
- Possui um suporte adesivo.
- Excelente relação preço/desempenho.
- Interface fácil adequada para qualquer microcontrolador.
- Opera a um máximo de 24 VDC, 30 mA.
- Interface de 8 pinos para interagir com a matriz 4x4.



Figura 4.6: Teclado alfanumérico utilizado [45]

A grande versatilidade deste componente, aliada ao seu baixo custo e às suas dimensões, faz com que este possa ser aplicado em projetos de controlo de acessos e em sistemas embutidos. No que diz respeito ao seu funcionamento, este é composto por

uma matriz 4x4, cujos elementos se encontram interligados por 8 fios condutores, como mostra a figura 4.7. Deste modo os fios numerados de 1 a 4 ligam eletricamente as colunas da matriz, enquanto os restantes ligam as linhas da mesma. Uma maneira de o microcontrolador interpretar qual das teclas foi pressionada consiste em ligar cada um dos pinos da matriz a entradas digitais do mesmo, de modo a que este possa obter a leitura dos 8 pinos da matriz. Desta forma se as entradas estivessem puxadas a 3,3V por uma resistência *pull-up* (ou estivessem definidas como uma entrada *pull-up* do microcontrolador), sempre que fosse pressionada uma tecla, duas das entradas digitais mudariam o seu valor. Desse forma seria possível identificar de que linha e coluna viria o impulso, identificando inequivocamente a tecla pressionada.

Esta abordagem tem porém um inconveniente, que é o facto de serem necessários 8 pinos disponíveis do microcontrolador, limitando desta forma o número de periféricos que poderiam estar ligados ao ESP32. Outra forma de realizar a leitura do teclado seria utilizar um divisor resistivo, à semelhança do que propôs o autor [48] na sua solução, abordada no capítulo 2.3. Desta maneira seria necessário apenas um GPIO do microcontrolador, de modo a efetuar a leitura do sinal analógico vindo do divisor resistivo, que teria um intervalo de valores diferente consoante a tecla que fosse pressionada. Contudo, segundo o próprio autor desta solução, esta não é a maneira mais robusta de efetuar a leitura da tecla pressionada, pois a existência de ruído eletromagnético faz com que possam existir erros de leitura do teclado. Posto isto, a estratégia passou pela utilização de um *IO expander*, o PCF8574. Este dispositivo é de uma maneira sucinta uma forma de aumentar o número de GPIO digitais de um microcontrolador, pois possui 8 pinos extra que podem ser utilizados como entrada ou saída digital. Por sua vez, o PCF8574 é capaz de comunicar com o microcontrolador seguindo o protocolo I2C, necessitando para isso apenas de 2 pinos do ESP32. Desta forma é possível efetuar a leitura das teclas, de uma forma robusta, utilizando um pequeno número de pinos do microcontrolador.

4.2.3 Comunicação com *IO expander*

A leitura da senha introduzida no teclado, é feita à custa de 3 intervenientes essenciais, o ESP32, o *IO expander* e o próprio teclado. Sendo que o PCF8574 funciona como um prolongamento dos pinos do microcontrolador, existe a necessidade de comunicação entre estes dois componentes. A troca de mensagens serve essencialmente para o microcontrolador perceber o estado dos pinos do *IO expander* e para poder controlar o estado dos mesmos, uma vez que estes funcionam como entrada e como saída digital ao mesmo tempo.

Essa troca é feita segundo o protocolo de comunicação I2C. Segundo o mesmo, são necessários dois condutores para ligar dois ou mais dispositivos. Um condutor é utilizado para transmitir o sinal de relógio e é denominado de SCL (*Serial Clock*) e o segundo condutor, conhecido como SDA (*Serial Data*) permite uma troca bidirecional de dados entre os dispositivos [75].

A figura 4.8 mostra os pinos do PCF8574. Os GPIO de P0 a P7 são aqueles que funcionam como prolongamento dos pinos do ESP32 e é a eles que ficam ligados os 8 fios do teclado alfanumérico. Os pinos GND e VCC são de alimentação e deverão

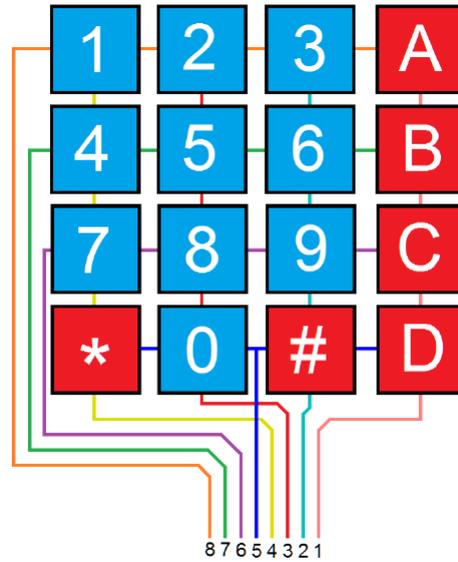


Figura 4.7: Esquema de ligações do teclado utilizado [45]

estar ligados respetivamente aos 0V do microcontrolador e aos 3,3V do mesmo. Os pinos A0, A1 e A2, têm uma grande importância, na medida em que permitem definir o endereço I2C deste dispositivo. Os pinos SDA e SCL do *IO expander* são conectados a entradas I2C do ESP32, de modo a que estes dispositivos troquem mensagens I2C entre si.

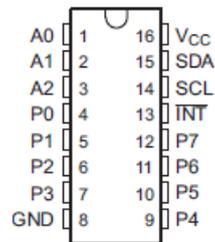


Figura 4.8: Esquema dos pinos do PCF8574 [76]

A comunicação I2C é do tipo *master-slave*, podendo estar vários dispositivo *slaves* ligados a um *master* por barramento, como mostra a figura 4.9. Sempre que não estiverem a ser utilizadas, as linhas de dados e de relógio devem ter uma tensão igual e positiva, daí estarem puxadas a 5V na figura 4.9 com resistências *pull up*.

O dispositivo com o papel de *master* (ESP32, neste caso) é responsável por gerar o sinal de relógio, bem como de controlar a linha de dados para enviar ordens e dados a outros dispositivos. Cada dispositivo tem um endereço de 7 *bits*, de modo a que as mensagens e ordens do *master* cheguem apenas ao dispositivo pretendido. Caso o ESP32 requisite determinados dados ao PCF8574, depois de fazer o pedido, este terá de manter o sinal de relógio ativo o tempo suficiente para receber uma

resposta do *slave*. Durante esse tempo será o *slave* a controlar a linha de dados.

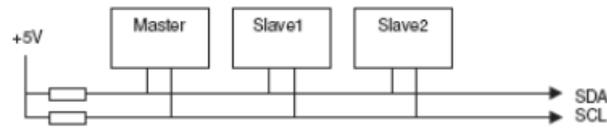


Figura 4.9: Barramento *master-slave*

De acordo com as regras deste protocolo, a tensão aplicada à linha de dados, SDA, só pode ser alterada se a linha de sinal, SCL tiver uma tensão nula. Caso tal aconteça, é sinal de que uma mensagem vai começar ou que a mensagem que estava a ser enviada terminou. Isto é, se a tensão da linha de dados passar a zero enquanto o sinal de relógio tiver uma tensão positiva, tal identifica o início de uma mensagem (*start bit*). Por outro lado, se a linha de dados passar a 5V enquanto o sinal de relógio tiver uma tensão positiva, isso indica que a mensagem terminou (*stop bit*).

Para uma correta troca de informação entre o PCF8574 e o ESP32, as mensagens I2C têm de obedecer a uma certa estrutura de mensagem (figura 4.10). Para que os dispositivos se apercebam que uma mensagem irá ser transmitida, o primeiro campo da mensagem deve ser um *start bit*, seguido de um *byte* com o endereço I2C do dispositivo *slave*. Este *byte* depende das ligações dos pinos A0, A1 e A2 do PCF8574. Neste caso optou-se por uma conexão dos mesmos aos 0V, sendo que dessa forma, os *bits* correspondentes, na figura 4.10 serão iguais a zero. O último *bit* desse *byte* contém a instrução a realizar (escrita ou leitura) numa das posições de memória do *slave*. Depois deste envio o dispositivo deve enviar um sinal, designado de *acknowledge* para que o dispositivo *master* saiba que a informação foi recebida. Só depois dessa resposta é que o *slave* envia um *byte* de dados, esperando o respetivo *acknowledge* para confirmar a receção dos dados. A mensagem termina quando for enviado o *bit* de fim de mensagem (P ou *stop bit*).

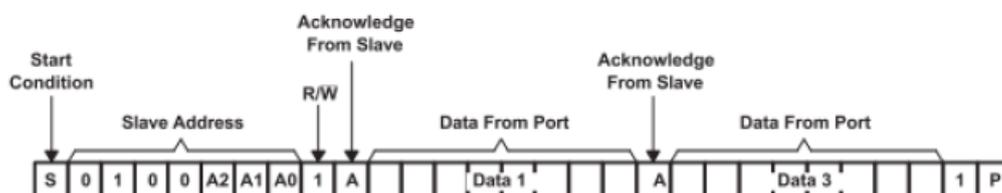


Figura 4.10: Estrutura de mensagem I2C para comunicar com o *IO expander* [76]

Como os pinos do *IO expander* são entradas e saídas digitais ao mesmo tempo, optou-se por colocar esses pinos com um valor que interessasse para perceber qual das colunas ou linhas tinha sido ativa. Dessa forma o microcontrolador escreve um *byte* que contém o valor de cada uma das saídas e o *IO expander* responde enviado um *byte* com o estado das suas entradas, que pode ser diferente do que foi inicialmente introduzido, caso seja pressionada alguma tecla. Sendo assim,

inicialmente é enviado para o *slave* mensagem: '0b11110000'. Esta mensagem irá modificar o estado dos pinos P0 a P7, como se estes fossem saídas digitais e o microcontrolador estará a receber continuamente uma mensagem igual à que enviou, exceto se uma tecla for carregada. Se tal acontecer um dos *bits* a '1' da mensagem irá passar a '0' e dessa forma é possível saber em que coluna se encontra. A tabela 4.1 mostra as mensagens recebidas, dependendo da coluna e da linha a que pertença a tecla pressionada (a contar da esquerda para a direita e de cima para baixo). Assim que for detetado que se carregou numa tecla, é enviado um byte diferente ao *slave*, que será '0b00001111', sendo lido imediatamente uma mensagem que desta vez será diferente da enviada, pois estará a ser pressionada uma tecla e consoante um dos 4 bits a '1' vier a zero, é possível determinar a linha da tecla, como mostra a tabela 4.1 (contagem de cima para baixo), ficando esta assim inequivocamente identificada. É de realçar que esta troca de informação se realiza de forma bastante rápida, pelo que para detetar que foi pressionada uma tecla o tempo que é necessário pressionar a mesma é bastante pequeno. Depois de identificar a tecla é escrita novamente no pcf8574 a mensagem '0b11110000', sendo que um *array* vai sendo preenchido e quando for detetado que foi inserido um número de caracteres iguais aos da palavra chave, a senha introduzida irá ser comparada com as senhas válidas naquele momento.

Por forma a facilitar a programação da troca de mensagens destes dispositivos, foram utilizadas funções da biblioteca *Wire* [77], disponíveis para o ambiente de programação Arduino IDE.

Tabela 4.1: Mensagens trocadas (formato decimal)

Mensagem enviada	Mensagem recebida	Coluna/Linha
240	240	nenhuma tecla
240	224	coluna 1
240	208	coluna 2
240	176	coluna 3
240	112	coluna 4
15	14	linha 1
15	13	linha 2
15	12	linha 3
15	7	linha 4

4.2.4 Módulo RTC

Como foi mencionado anteriormente, a comunicação entre o servidor central e os microcontroladores é feita através da troca de ficheiros, segundo o protocolo HTTP. No caso do ESP32, responsável pelo controlo de acessos e leitura das senhas introduzidas, é fundamental que esse dispositivo receba no ficheiro pedido ao servidor, informações sobre a data e hora atuais, de modo a que esta seja comparada com as datas de *check-in* e *check-out* também vindas no mesmo ficheiro pedido ao servidor com vista a determinar a validade da senha introduzida. Sendo assim a fiabilidade deste sistema está muito dependente da qualidade da ligação à rede, pois se

esta falhar, o ESP32 deixa de receber informação, podendo ficar com informações desatualizadas.

Com vista a tornar todo o projeto mais robusto, optou-se pela utilização de um RTC (*Real Timmer Clock*). Este consiste num pequeno dispositivo com memória, ao qual está conectado um oscilador. Desta forma é possível, lendo as posições de memória corretas do RTC, obter a data e a hora atuais.

Assim sendo, de cada vez que é feito um pedido por parte do ESP32 ao servidor, o ficheiro PHP pedido, irá à base de dados buscar para aquele quarto, um conjunto de três datas de *check-in*, datas de *check-out* e respetivas senhas (válidas apenas entre aquele período) mais próximas da data atual. Estes dados irão ser recebidos pelo microcontrolador, que os irá guardar. Desta forma caso haja por qualquer motivo uma falha de ligação à rede prolongada, o microcontrolador terá informação suficiente para comparar corretamente as senhas, pois sendo a informação da data e hora atual dada por *hardware*, isso significa que mesmo que a ligação ao *access point* ou ao servidor falhe durante algum tempo, o ESP32 continua a ter na sua memória 3 senhas, validando a sua autenticidade consoante a data atual esteja entre os dias de entrada e saída no espaço alugado.

Nesta aplicação optou-se então pela utilização do RTC DS1302, que possui as seguintes características: [78]

- Processa segundos, minutos, horas, dias da semana, dias do mês, meses e anos (de 2000 a 2099).
- Tem 56 *bytes* de SRAM.
- Tensão: 3,3 a 5,5V.
- Possui circuito de deteção de falha de energia.
- Consome menos de 500 nA no modo bateria com oscilador em funcionamento.

Na figura 4.11 está representado a distribuição de pinos deste dispositivo. Nos pinos x1 e x2 deve estar conectado um oscilador de 32,768 kHz para um correto funcionamento. Os pinos GND, Vcc1 e Vcc2 são os pinos de alimentação. O GND é o terra da alimentação e o terminal positivo da alimentação será o que tiver maior tensão entre Vcc1 e Vcc2, sendo que o de menor tensão está desenhado para a existência de uma bateria, dessa forma, caso haja corrente no circuito principal de alimentação, o RTC é alimentado pelo mesmo e a bateria carrega. Caso este falhe, entrará em ação a bateria, para que se evite uma paragem do relógio por falta de corrente. Os restantes pinos servem para comunicar por SPI (*Serial Peripheral Interface*) com o microcontrolador, pelo que estão conectados a GPIOs do mesmo para esse efeito.

O protocolo SPI permite a troca de dados entre equipamentos, através de dois fios condutores. Desta forma, o pino *Serial Data Output* de um dispositivo liga ao pino *Serial Data Input do outro* e vice-versa. Os sinais elétricos enviados e recebidos são digitais, têm uma tensão nula ou positiva (3,3V neste caso pois é a tensão de operação do ESP32). Este protocolo prevê também a existência de um terceiro condutor, onde um sinal digital, designado de sinal de relógio (*clock*) com uma frequência conhecida, permite aos dois dispositivos sincronizar o instante de tempo

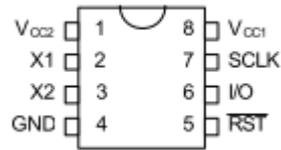


Figura 4.11: Esquema dos pinos do DS1302 [78]

em que o emissor aplica uma tensão no condutor e o instante de tempo em que o recetor deve amostrar esse sinal digital. Este é o tipo de comunicação utilizada neste caso concreto e designa-se de comunicação síncrona. [79]

À semelhança do protocolo I2C, este prevê uma comunicação do tipo *master-slave*, onde vários é possível ligar mais do que um *slave* ao mesmo *master*, por barramento, como mostra a figura 4.12. Nesse caso, apenas dois dispositivos poderão estar ativos ao mesmo tempo, competindo ao *master* a função de escolher o dispositivo com que quer comunicar, ativando o pino *slave select*. Além disso, o dispositivo *master* (neste caso o ESP32) gera o sinal de relógio e controla os instantes em que este está ligado ou desligado. O *slave* (DS1302 neste caso) pode apenas receber ou enviar dados, se e enquanto o ESP32 ativar o sinal de relógio. Como o DS1302 apenas responde a mensagens do ESP32, é necessário enviar algo, ainda que sejam *bits* nulos para esse dispositivo, para este enviar os seus dados (informações de data e hora). Por cada *bit* enviado ao *slave* ele responde com outro, no mesmo instante, em sincronismo, tratando-se assim de um dialogo *full duplex*.

De modo a facilitar esta comunicação no ambiente de programação Arduino IDE, foi utilizada uma biblioteca criada especificamente para este RTC, chamada *virtuabotixRTC* [80], que se revelou um método bastante mais fácil de interagir com este periférico.

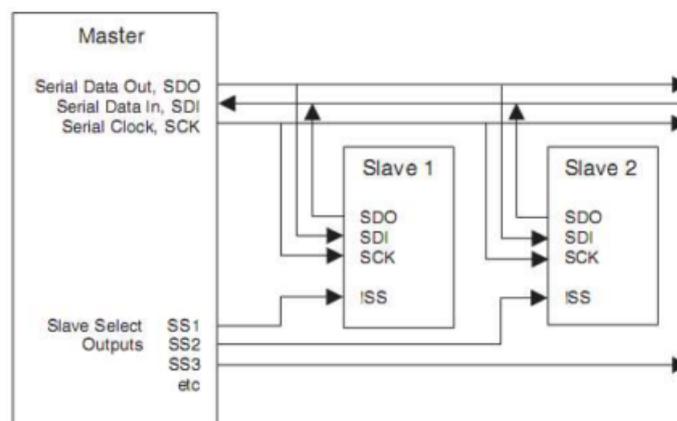


Figura 4.12: Esquema dos pinos do DS1302 [79]

4.2.5 Testa com fecho elétrico

A solução para controlo de acessos proposta, pretende ser de baixo custo e de fácil implementação. Dessa forma, para que o controlo de abertura da porta seja feito de maneira mais simples possível e de modo a não fazer modificações acentuadas nas instalações de serviço turístico, é proposta a utilização de testas com fecho elétrico. Na figura 4.13 está representada à esquerda uma fechadura com uma testa convencional e à direita uma fechadura com fecho elétrico.

Este tipo de testa é alimentada com 12V, sendo que para o seu controlo foi utilizado um relé que é atracado com os 3,3V de uma saída digital do ESP32. A testa é alimentada com 12V e consome cerca de 1A, pelo que, de modo a proteger o microcontrolador e restante circuito elétrico de correntes tão elevadas, foi utilizado um relé opto-isolado para acionar a passagem dos 12V para a testa. Desta maneira o controlo da abertura da porta é feito ativando ou desativando uma saída digital do microcontrolador.

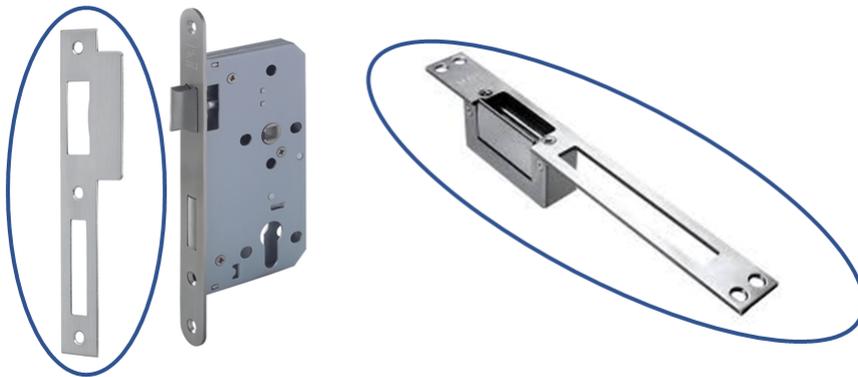


Figura 4.13: Testa convencional e testa com fecho elétrico

4.2.6 Sensores utilizados

Além de todos os periféricos utilizados, para garantir um controlo de acessos completamente automatizado e robusto, houve a necessidade de integrar no sistema um conjunto de sensores. Com a utilização desta solução, a gestão de um serviço turístico poderia ser feita remotamente. Dessa maneira, o gestor não necessitaria de estar no local, nem tão pouco haveria a necessidade de um rececionista. Sendo assim o proprietário ou gestor do serviço tem a necessidade de saber remotamente informações relevantes acerca da utilização do seu estabelecimento. Algumas das informações mais relevantes, passam por saber se cada espaço esteve efetivamente ocupado, se alguma das portas foi deixada aberta por descuido, ou se uma lâmpada foi deixada acesa por engano.

Ao longo da realização desta dissertação, como já referido anteriormente, chegou-se à conclusão que saber se uma determinada divisão esteve ocupada ou não, poderia ser uma tarefa que poderia ser interessante em qualquer serviço turístico, sendo

que a aplicabilidade deste tipo de soluções transcende o setor do turismo, podendo ser aplicado em qualquer área de serviços, ou mesmo em ambiente residencial para otimização de sistemas AVAC, por exemplo. Foi nessa medida que foi proposto um módulo adicional que pretende fazer a deteção de presenças com base na análise do ruído de um determinado espaço. Dessa forma, o módulo composto pelo ESP8266 e pelo sensor de som poderia ser utilizado em qualquer espaço, mesmo que o controlo de acessos fosse feito de modo mais tradicional.

Em seguida são expostos os sensores utilizados quer no módulo principal que é responsável também por controlar os acessos, quer no módulo adicional para deteção de presenças. Além de ser explicado o seu modo de funcionamento, é evidenciada a sua relevância para a solução proposta. Além disso outros sensores não integrados na solução, mas testados serão também abordados, sendo discutida a razão pela qual se optou pela sua não integração.

De salientar que os sensores experimentados vão de encontro aos que foram utilizados nos capítulos 2.3 e 2.4. Sendo que o pretendido seria monitorizar o estado de ocupação de uma divisão, bem como ter informações adicionais sobre o ambiente em que se encontra o protótipo proposto, sem com isto colocar em causa a privacidade e o conforto do utilizador final.

Desta forma, foi testada a utilização de sensores de movimento, som, luz, CO_2 , e de um sensor de porta. Sensores de temperatura e humidade foram à partida descartados, uma vez que condições de temperatura poderiam variar bastante com a localização do serviço, área de envidraçados do local, entre outros, como a existência de sistemas de aquecimento ou ar condicionado, pelo que à partida este tipo de sensores não oferece informação muito relevante acerca do estado de ocupação do espaço, nem tão pouco as informações deste são de grande relevo para o proprietário do espaço em questão.

Sendo assim o primeiro sensor que se julgou ser de grande importância, pois à partida permitiria determinar se um espaço estaria ocupado ou não foi um sensor de CO_2 . Neste âmbito foi testada a utilização do sensor MG811. Este consiste numa célula que fornece uma saída na faixa de 250-330 mV, para concentrações de CO_2 entre 400 e 10000 ppm como mostra a figura 4.14 [81]. O gráfico da mesma figura mostra ainda que a tensão de saída e a concentração de CO_2 têm uma relação logarítmica, de acordo com a equação 4.1. Onde V_s designa a tensão de saída do sensor, V_0 é a tensão de saída, para uma concentração de 400 ppm, CCO_2 refere-se às várias concentrações de dióxido de carbono que se podem medir e ΔV_s é a diferença entre a tensão de saída do sensor quando a concentração é de 400 ppm e a tensão do mesmo quando a concentração é de 1000 ppm.

$$V_s = V_0 + \Delta V_s / (\log_{10}(400) - \log_{10}(1000)) * (\log_{10}(CCO_2) - \log_{10}(400))$$

(4.1)

Este sensor possui uma amplitude de sinal bastante baixa, por isso mesmo, foi usada uma placa que contém um circuito de condicionamento de sinal, como a que

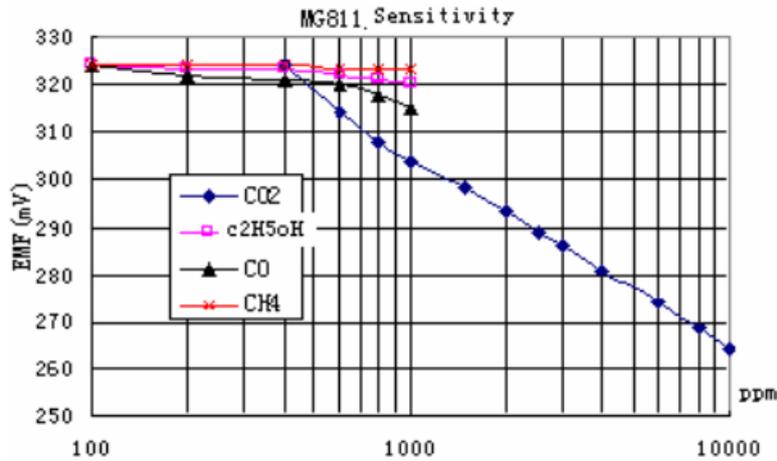


Figura 4.14: Resposta do sensor MG811 [81]

se mostra na figura 4.15. Desta forma, as tensões de saída da placa seriam afetadas por um ganho do amplificador, que pode ser regulado manualmente através de um potenciômetro presente no módulo. Tal faz com que estas tomem valores mais significativos nas gamas entre os 0 e 1V que são as tensões de operação das entradas analógicas microcontroladores utilizados.

O objetivo principal com a utilização deste sensor era a determinação de ocupação de um espaço, pois previa-se que um espaço ocupado viesse a ter níveis de concentração de CO_2 superiores a um espaço vazio. Contudo, verificou-se, deixando o sensor em funcionamento num laboratório do DEM (Departamento de Engenharia Mecânica) alguns dias, que os valores obtidos por este sensor não variavam significativamente com a ocupação. Uma causa apontada para tal, poderia ter sido o facto de que quando o laboratório onde ocorreram testes, sempre que estava ocupado, a porta do mesmo estava aberta, podendo esse facto justificar o facto de os valores serem semelhantes para uma sala ocupada e para uma sala vazia. Sendo assim experimentou-se fechar a porta com ocupantes dentro da sala. O que se concluiu foi que os valores de concentração dados pelo sensor apenas aumentavam ligeiramente passada meia hora de a sala estar fechada e ocupada. Uma determinação de ocupação baseada na concentração de gases seria portanto um método pouco preciso, uma vez que os valores demoram algum tempo a alterar-se. Podiam esses valores juntamente com o sensor de porta determinar a ocupação (pois os valores do MG811 revelaram-se dependentes do estado de abertura da porta). Contudo, a resposta seria sempre lenta, pois para o sensor de gás se "aperceber" da presença humana teria de decorrer cerca de meia hora. Além do mais outros fatores poderiam influenciar esses valores, como uma janela aberta, ou a existência de um sistema de ventilação na divisão.

Aos problemas mencionados anteriormente, acresce o facto de este sensor ter um custo elevado, face aos restantes componentes do protótipo [82]. Sendo assim, a decisão foi não integrar este componente na solução final.

Uma maneira de verificar se um espaço está ocupado consiste em detetar movi-



Figura 4.15: Placa com MG811 e circuito de condicionamento de sinal

mento no interior desse mesmo espaço, posto isto, decidi utilizar-se o sensor de movimento PIR representado na figura 4.16. Este módulo contém um sensor piroelétrico que deteta níveis de radiação infravermelha. Todos os objetos emitem alguma radiação de baixo nível e quanto mais quente algo é, mais radiação emite. Neste tipo de sensores de movimento, o detetor de radiação é dividido em duas partes, pois o que interessa nestes casos é medir a variação da radiação e não saber exatamente os níveis da mesma [83]. Sendo assim, quando não houver qualquer tipo de movimento, as duas metades irão medir a mesma radiação, que é a proveniente das paredes e restante ambiente que rodeia o sensor, mas quando um corpo quente se aproxima do sensor, uma das metades irá receber mais radiação do que a outra e quando essa pessoa se afasta será a outra metade a receber mais radiação. Dessa maneira é possível detetar movimentos.

Este módulo pode ser alimentado tanto pelos 3,3V do microcontrolador como pelos 5V do mesmo e tem um ângulo de detecção de até cerca de 100° . A sua sensibilidade e o seu tempo de restabelecimento (tempo até poder detetar movimentos depois de uma primeira detecção) são ajustáveis, através de potenciômetros incluídos no mesmo. Possui 3 pinos, dois deles são utilizados para alimentação e o outro é o sinal de saída, que será de 0V caso não seja detetado movimento, ou 3,3V caso algo seja detetado.

Este tipo de sensor fornece informações importantes ao proprietário de um estabelecimento turístico, pois à partida se for detetado movimento, significa que a divisão em causa está a ser ocupada. No entanto, nem sempre será detetado movimento, uma vez que o alcance destes dispositivos é limitado, pelo que se a divisão tiver dimensões consideráveis, é possível que parte dos movimentos não sejam detetados. A conclusão com a utilização deste sensor, foi de que ele por si só é incapaz de permitir saber se um espaço se encontra ocupado ou não, contudo, fornece informações importantes, pelo que se optou por utilizar este sensor na solução final.

Um outro parâmetro ambiental, que pode ser interessante de obter, é o nível de luz numa determinada divisão. Este atributo pode não dar um grande contributo para a determinação de ocupação, uma vez que o nível de luz de um local está muito dependente da sua localização, tamanho de envidraçados, entre outros. Contudo

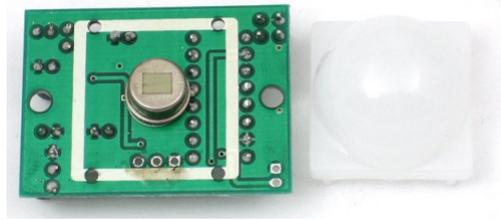


Figura 4.16: Sensor de movimento PIR

o conhecimento dos valores de um sensor de luz, ao longo do tempo pode dar informações bastante interessantes. Por exemplo, se o nível de intensidade luminosa for alto a uma hora em que não existe iluminação natural (meia noite por exemplo), isso indica que existe iluminação artificial ligada dentro da divisão (uma lâmpada por exemplo), o que poderá dar indicação de que alguém se encontrou naquela divisão. Por outro lado, se por exemplo o nível de luminosidade de uma divisão for elevado durante toda a noite, tal poderá ser um indicador de que alguém se esqueceu de uma luz ou da televisão acesas, dando assim ao proprietário uma percepção daquilo que se possa estar a passar nas suas instalações.

Posto isto, decidiu-se integrar na solução um sensor LDR (*Light Dependent Resistor*). Tal como o nome indica, este sensor consiste numa resistência que varia com a intensidade luminosa. De modo a obter um valor que varia consoante a luminosidade, esta resistência sensível à luz é colocada em série com uma de valor conhecido, como mostra a figura 4.17. Desta maneira, o circuito comporta-se como um divisor resistivo. Quando está escuro, a foto-resistência tem um valor muito alto, cerca de $10\text{M}\Omega$, sendo a tensão V_{out} muito baixa, próxima de zero. Por sua vez quando existe alguma luminosidade, a resistência variável diminui, verificando-se um valor de V_{out} de acordo com a equação 4.2, em que V_{out} vai variar entre 0V e a tensão de alimentação (5V neste caso), permitindo desta forma a leitura deste sensor por uma entrada analógica do ESP32.

$$V_{out} = \frac{5 * R1}{R1 + LDR}$$

(4.2)

Um dos aspetos de grande importância consiste em saber se a porta de um determinado quarto se encontra aberta ou fechada. Mais do que ajudar na determinação de ocupação de um espaço, este tipo de sensor permite aumentar a segurança do local, visto que se a porta estiver aberta durante muito tempo, ou numa altura em que não existem reservas, tal poderá ser um indicador de que alguém deixou a porta aberta por lapso, ou de que o espaço foi invadido por pessoas não autorizadas. Este tipo de indicador ganha especial relevância, se tivermos em conta que o proprietário gere o espaço remotamente, deslocando-se ao local poucas vezes.



Figura 4.17: Esquema elétrico de um sensor LDR

Sendo assim, um dos sensores utilizados juntamente com o ESP32, foi um sensor de porta. Este sensor consiste em duas metades, uma delas com uma ampola e outra com um íman. A figura 4.19 mostra o princípio de funcionamento do mesmo. Uma das extremidades da ampola estará ligada aos 0V do ESP32 enquanto que a outra está ligada a uma entrada analógica do mesmo puxada a 3,3V, pois foi definida no código como sendo uma entrada *pull-up*. Desta maneira, sempre que o íman e a ampola estiverem próximos, a ampola abre o circuito, ficando a entrada do microcontrolador a 3,3V. Por outro lado, se as duas metades estiverem afastadas, o circuito fecha devido ao campo magnético e a entrada digital lê o valor de 0V.

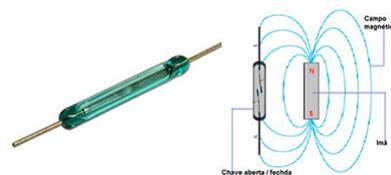


Figura 4.18: Princípio de funcionamento do sensor de porta

Na solução implementada, uma das partes estará fixa no aro da porta, enquanto que a outra estará presa à porta, como mostra a figura 4.19. Desta maneira sempre que a porta estiver aberta, as peças estarão separadas e quando estiver fechada, esta juntam-se, fechando o circuito.

Todos os sensores anteriormente mencionados permitem ao proprietário ter uma noção do que se passa no seu estabelecimento, de modo a que este consiga monitorizar remotamente o seu espaço, contudo continuava a fazer falta um método que permitisse determinar o estado de ocupação de cada quarto e que fosse o mais universal possível, podendo ser utilizado em qualquer divisão independentemente da sua localização ou outro fator qualquer. Posto isto, o módulo de deteção de presenças é constituído pelo ESP8266 e por um sensor de som.

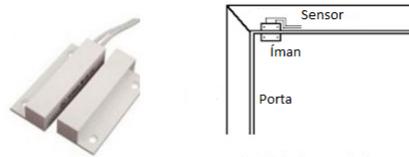


Figura 4.19: Esquema de implementação de um sensor de porta

Num espaço ocupado por seres humanos, é inevitável que hajam barulhos/ruídos, provocados pela voz dos mesmos, ou por aparelhos eletrônicos como televisões, telemóveis, etc. Desta forma, pretendia-se que analisando e tratando os sinais do sensor de maneira adequada, fosse possível determinar se um espaço estava ocupado ou não.

Posto isto foi utilizado o sensor de som mostrado na figura 4.20. Este é alimentado pelos 5 V do ESP8266 e o seu pino de *output* é ligado à entrada analógica do mesmo. De ressaltar que essa entrada analógica apenas esta a medir a pressão sonora, ou seja, o valor de saída é tanto maior quanto maior for a pressão sonora. Graficamente é possível obter uma curva deste parâmetro ao longo do tempo. Sendo assim, é possível decompor uma qualquer curva num somatório de sinusoides, com diferentes frequências e amplitudes segundo a série de Fourier. Para tal, foi utilizada a biblioteca *arduinoFFT* do ambiente de programação arduino IDE, para decompor o sinal vindo do sensor num somatório de sinusoides. A técnica utilizada para processar o sinal e detetar a presença humana é detalhada mais adiante neste documento, no capítulo 4.4.



Figura 4.20: Módulo sensor de som

4.2.7 Placas PCB

Para o desenvolvimento desta dissertação, de modo a aproximar a solução proposta de um prototipo final, foram desenvolvidas placas PCB (placas de circuito impresso). Para tal, foi utilizado o *software Eagle* da "AutoDesk", que permite projetar este tipo de componentes de forma intuitiva, devido ao número de bibliotecas disponíveis para esta interface.

Pretende-se que parte da solução fique do lado exterior da porta, para que o cliente possa marcar o seu código no teclado para o efeito. Contudo por uma questão de segurança, seria preferível que o microcontrolador e toda a parte da alimentação

ficasse no interior do espaço cujo acesso se pretende controlar, pois caso contrário seria possível forçar a entrada, manipulando a alimentação do trinco elétrico. Além disso, é de conveniência que os sensores também se encontrem dentro da divisão a monitorizar, para assim fornecerem ao proprietário dados relevantes do ambiente envolvente de um determinado espaço.

Desta maneira, a parte de controlo de acessos foi dividida em dois módulos, uma para ficar no exterior (figura 4.21) e outra no interior do espaço (figura 4.22). A interligação dos dois é feita à custa de um cabo *ethernet*, compactando assim os fios necessário para a ligação.

O módulo exterior contém apenas o *IO expander* juntamente com o teclado, para que o código possa ser marcado no exterior, juntamente os LEDs para que o utilizador tenha uma indicação visual das ações que está a realizar ao pressionar as teclas.

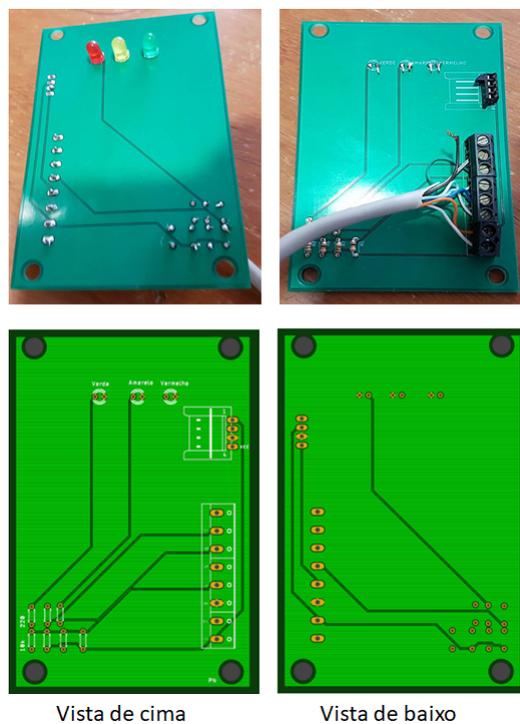


Figura 4.21: Módulo a ficar no exterior da divisão

O módulo a ficar no interior da divisão contém todo o circuito de alimentação, bem como os sensores mencionados anteriormente. A alimentação é conseguida a partir de um transformador de 12V que alimenta diretamente o trinco com fecho elétrico. Para alimentar os sensores e o ESP32 essa voltagem é muito elevada, pois estes dispositivos toleram uma alimentação de até 5V no máximo. Sendo assim e de forma a separar a alimentação dos sensores da alimentação do microcontrolador, utilizaram-se dois reguladores de tensão o LM7805, que permitem passar os 12V vindos do transformador para os 5V necessários. Além da alimentação proveniente do transformador, a placa exterior possui também uma ligação a uma bateria UPS,

de modo a garantir uma alimentação do sistema, caso haja falha de energia. Sendo assim, essa bateria carrega enquanto o transformador fornecer energia e caso este falhe, o circuito será alimentado pela bateria.

Para o módulo adicional de deteção de ocupação não foi desenhada qualquer placa, uma vez que o sistema já é bastante compacto pois apenas tem um periférico ligado (sensor de som), sendo o ESP8266 alimentado através da sua entrada micro USB por um transformador de 5V.

Todos os esquemas elétricos referentes às placas encontram-se descritos no apêndice A, para uma melhor compreensão.

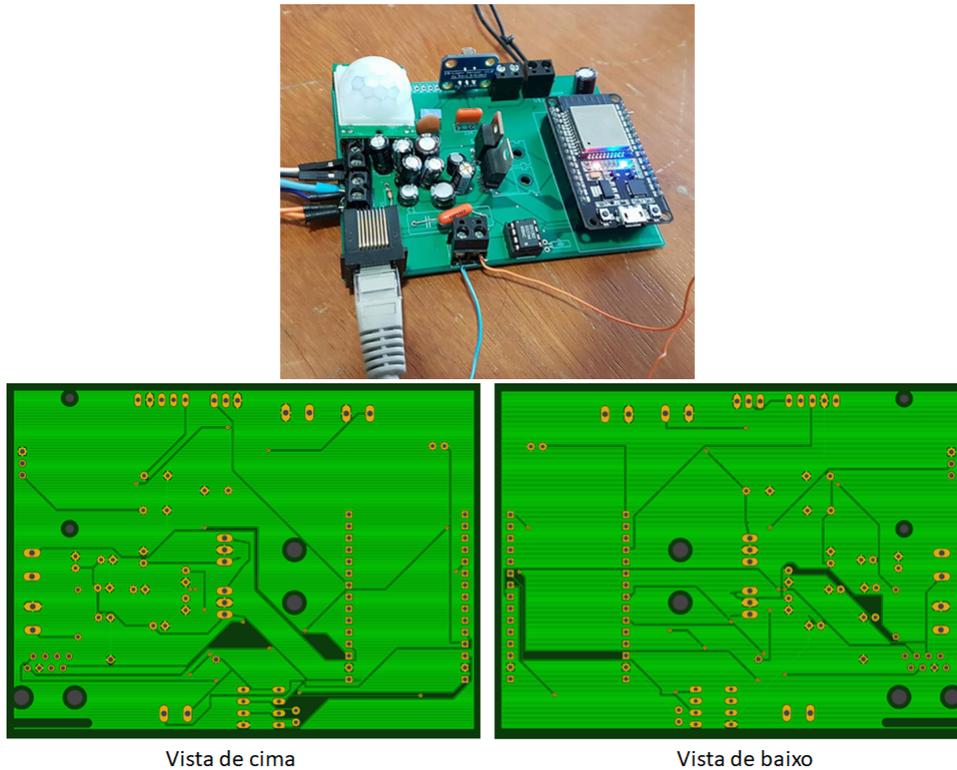


Figura 4.22: Placa a ficar no interior da divisão

4.3 Software

Esta secção é dedicada a apresentar de forma detalhada todo o *software* utilizado na implementação e a programação envolvida na solução proposta. Aqui são abordados os seguintes tópicos:

- Servidor *Web* (Apache): São expostas todas as funcionalidades do modelo de comunicação cliente/servidor utilizado.
- Módulo de controlo de acessos: Aqui é exposta a maneira como foi elaborado o programa do microcontrolador ESP32 utilizado para o controlo de acessos,

de forma a que fique claro o funcionamento desta parte da solução.

- Interface *Web*: São descritas todas as funcionalidades da interface criada e a maneira como clientes e proprietário podem interagir com a mesma.
- Método de pagamento via *PayPal*: É descrita detalhadamente a forma como foi integrada esta plataforma de pagamento na interface *Web* criada, de maneira a que a forma como é feita a comunicação entre cada entidade fique clara.

4.3.1 Servidor *Web*

A comunicação nesta aplicação é do tipo cliente/servidor. O servidor receberá os pedidos HTTP de todos os clientes *Web*, conforme descrito no capítulo 4.1.1. Deste modo um servidor consiste num computador, com uma aplicação que lhe permita receber e enviar a resposta de todos os pedidos recebidos. De forma a realizar os testes necessários para garantir um bom funcionamento de todo o sistema, foi utilizado como servidor um computador do DEM, com um pacote de aplicações denominado *XAMPP*. Este pacote de *software* tem integrada uma aplicação denominada de servidor "Apache" que permite ao computador em questão receber e responder a pedidos feitos por aplicações clientes *Web* numa porta TCP/IP. Neste caso específico, foi utilizada a porta 81 para receber ligações TCP/IP, sendo que para comunicação com a base de dados, foi utilizado o *MySQL server* que é um servidor de bases de dados e aceita ligações na porta 3306 do mesmo computador. Sendo assim, além de servidor *Web* o computador em questão desempenha também o papel de servidor de bases de dados.

Além de assegurar uma correta comunicação com os microcontroladores de cada alojamento, o servidor Apache tem ainda a função de alojar as páginas *Web* desenvolvidas com recurso às linguagens de programação HTML, *javascript*, PHP, entre outras, de modo a que o cliente possa efetuar a sua reserva remotamente e os dados sejam inseridos na respetiva base de dados do sistema. Para que o servidor "Apache" consiga encontrar os ficheiros pedidos pela aplicação cliente (*browser* ou microcontrolador), esses ficheiros devem estar no diretório "C:\xampp\htdocs".

De acordo com os requisitos pretendidos para este trabalho, há a necessidade de enviar emails ao cliente. Essa necessidade existe em dois momentos distintos. Um deles é na altura do registo do mesmo no sistema, em que se prevê a confirmação do registo via email. A segunda situação prevista é o momento após a realização da reserva e confirmação de pagamento, pois é imperioso informar o cliente da sua senha de acesso. De forma a tornar o processo o mais rápido e eficiente possível, torna-se necessário que o envio destas mensagens seja feito de forma automática. O PHP tem uma função chamada "mail", que permite o envio automático de emails de forma rápida e intuitiva. Contudo, para que esta função realize o seu propósito, é necessário configurar alguns ficheiros do XAMPP, nomeadamente o ficheiro "php.ini" e o ficheiro "sendmail.ini". A figura 4.23 mostra as alterações que tiveram de ser feitas nestes ficheiros.

Para completar todo o processo, foi criada uma conta de email no "Gmail", de onde as mensagens pudessem ser enviadas. Este endereço destina-se também a receber

mensagens com possíveis dúvidas dos clientes. As alterações efetuadas nos ficheiros de configuração servem essencialmente para definir o servidor SMTP (*Simple Mail Transfer Protocol*) e a conta de email da qual se querem enviar mensagens. De referir que o campo "*auth_password*" é preenchido com a senha de *login* da conta de email criada. O servidor SMTP funciona como uma espécie de estação dos correios, pois está encarregue de receber as mensagens de email, que neste caso são definidas previamente num ficheiro com extensão PHP e posteriormente proceder ao seu reencaminhamento para o endereço correto, que virá contido na mensagem. Esta conexão é feita através da porta 587.



```
php.ini
smtp_server=smtp.gmail.com
smtp_port=587
smtp_ssl=auto

error_logfile=error.log

debug_logfile=debug.log

auth_username=servicoturisticoua@gmail.com
auth_password=

pop3_server=
pop3_username=
pop3_password=

force_sender=servicoturisticoua@gmail.com

sendmail.ini
[mail function]
sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"
mail.add_x_header=Off
```

Figura 4.23: Configuração do XAMPP para envio de emails

4.3.2 Módulo de controlo de acessos

O módulo de controlo de acessos tem como principal função detetar a senha introduzida por um cliente no ato de entrada no espaço, para poder abrir o trinco elétrico, caso a senha em questão seja válida. Mas além disso, o microcontrolador ESP32 realiza um conjunto mais extenso de tarefas, que vão desde a leitura de sensores, interação com o RTC e comunicação com o servidor. Posto isto, na figura 4.24 está representado o fluxograma de funcionamento do ESP32. Este tem um conjunto de tarefas que são realizadas apenas uma vez, sendo essas tarefas designadas no fluxograma de *Setup* inicial. Nesta fase realizam-se as seguintes tarefas:

- Definição das variáveis globais do projeto.
- Configuração dos parâmetros para comunicação I2C com o PCF8574, nomeadamente definição dos pinos de dados (SDA) e de relógio (SCL) e definição da frequência de relógio.
- Configuração do pinos para comunicação SPI com o RTC.
- Definição dos pinos onde estarão ligados os sensores como entradas analógicas ou digitais, conforme o tipo de sensor.

- Definição dos pinos de acionamento do relé para abertura da porta e dos LEDs como saídas digitais.
- Conexão à rede *Wi-Fi*.

Como já referido, esta rotina apenas se realiza uma vez, no início do programa. Quanto às restantes, são feitas ciclicamente. Quando se chega à ultima tarefa no fluxograma, este volta ao início, efetuando a leitura dos pinos do *IO expander* (identificada com o número 1), como mostra a figura 4.24.

Estando várias tarefas a ser realizadas, o programa foi feito de modo a que se priorizasse a leitura do teclado. Desta forma importa perceber se alguma tecla foi pressionada e caso tenha sido, o LED amarelo acende-se até que a tecla deixe de ser pressionada, dando indicação ao utilizador que a tecla que marcou foi reconhecida. De seguida é necessário guardar o seu valor num *array* e incrementar o contador de teclas pressionadas. Desta maneira se alguma tecla tiver sido pressionada ou se o contador de teclas for maior do que zero, as rotinas mais morosas como o pedido de ligação ao servidor não serão realizadas, permitindo uma leitura mais rápida da senha que esteja a ser introduzida. Contudo importa definir um tempo máximo para o utilizador premir a tecla seguinte assim que o contador de teclas seja incrementado pelo menos uma vez. Caso alguém por engano carregue apenas numa tecla, sem ser sua intenção marcar uma senha válida, o código ficaria indefinidamente a tentar ler a seguinte tecla. Sendo assim definiu-se um tempo máximo de 20 segundos entre a marcação de cada tecla. Caso esse tempo seja excedido, o contador de teclas é posto a zero e o LED vermelho pisca, dando indicação ao utilizador que excedeu o tempo de marcação.

Caso o contador de teclas esteja a zero e não estiver ser pressionada nenhuma tecla, o microcontrolador verifica o estado de ligação à rede *Wi-Fi*. Caso se encontre desligado, este irá tentar restabelecer a ligação e verificar de novo se esta se encontra estabelecida. Em caso afirmativo, o ESP32 irá ler os valores dos sensores de movimento, porta e luz e irá enviar um pedido HTTP ao servidor. Nesse pedido, irá ser requisitado um ficheiro PHP, sendo que lhe serão passadas informações pelo método "GET". Estas são os valores dos sensores e o número do quarto que corresponde o microcontrolador em questão e informação acerca de acessos ou tentativas de acesso entre pedidos ao servidor. Por sua vez, o servidor irá enviar uma resposta de onde consta a hora e data atual. Além disso, irá ser enviada uma cópia do conteúdo do ficheiro requisitado.

O ficheiro em questão irá fazer a ligação à base de dados, efetuando para cada registo da tabela de histórico de reservas o procedimento ilustrado pelo fluxograma da figura 4.25. Desta maneira, irão ser enviado para o microcontrolador os dados de *check-in*, *check-out* e senha de acesso de 3 registos diferentes, sendo que da maneira que o processo de recolha de dados foi efetuado, será enviada a senha em vigor na data do pedido ao servidor e as próximas duas senhas das duas reservas seguintes e respetivas datas de entrada e saída. Desta maneira o sistema de acessos consegue ser eficaz mesmo no caso de haver uma falha prolongada de ligação à rede, pois os dados recebidos ficam armazenados na memória do ESP32.

As senhas de acesso foram definidas como sendo um conjunto de 5 caracteres. Sendo assim, sempre que tenham sido pressionadas 5 teclas num espaço de tempo inferior a 20 segundos entre cada uma, significa que foi feita uma tentativa de acesso. Nessa altura o ESP32 irá ter armazenadas todos os caracteres pressionados. Posto isto irá fazer a comparação da senha inserida com as que possui armazenadas em memória.

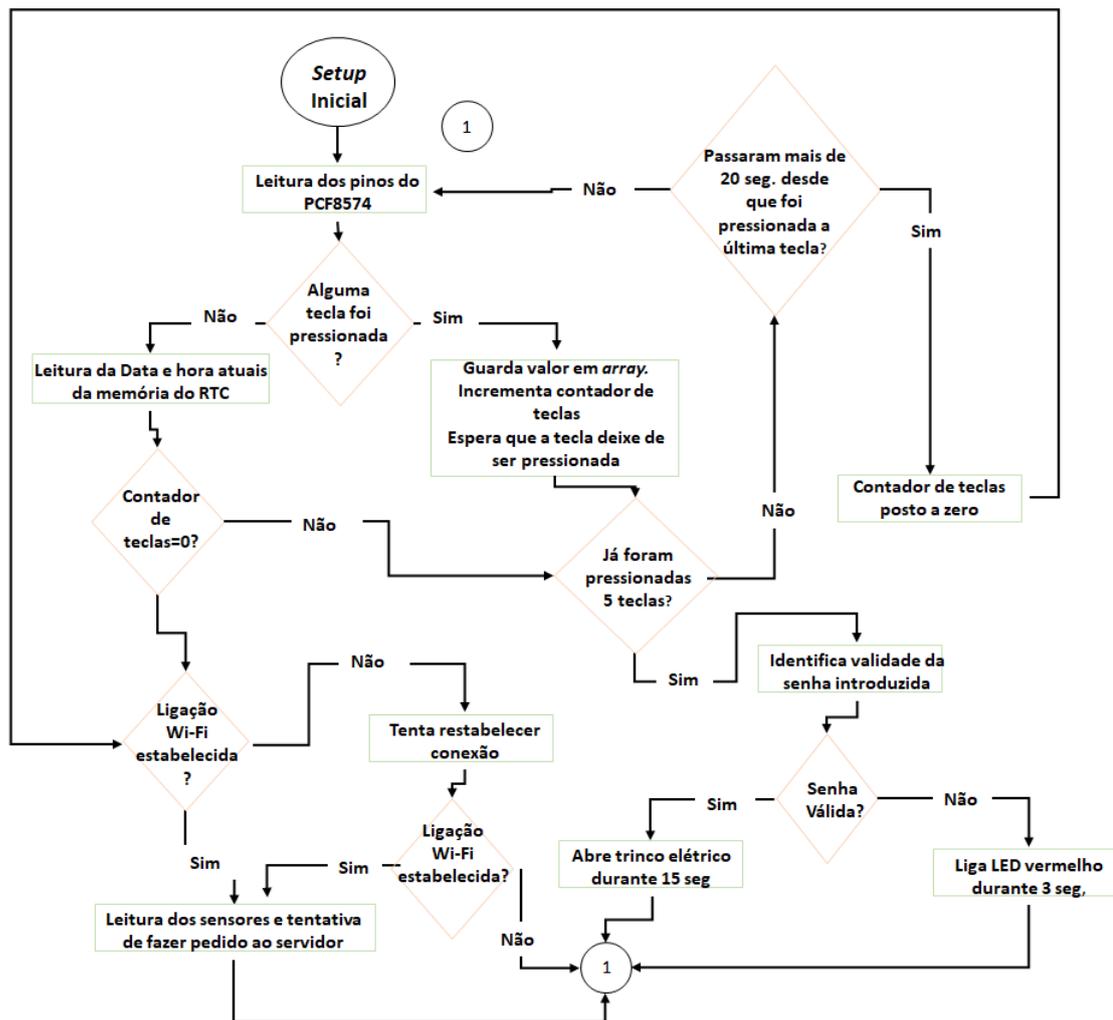


Figura 4.24: Fluxograma de tarefas a efetuar pelo ESP32

Sendo assim, caso a senha introduzida corresponda à *password default*, será dado acesso imediato, abrindo o fecho elétrico da porta durante 15 segundos e acendendo o LED verde. Caso a senha introduzida corresponda a umas das 3 recebidas do servidor, irão ser comparadas as datas de *check-in* e *check-out* com a data atual, para aferir se aquela senha é válida. Caso se confirme a validade da senha, é aberto o fecho da porta e aceso o LED verde, caso contrário, é aceso o LED vermelho durante 3 segundos e o contador de teclas posto a zero.

De notar que a hora e data atuais podem ser obtidos de duas maneiras, ou através da mensagem vinda do servidor ou através do RTC. Sempre que se consiga estabelecer uma ligação ao servidor, o RTC é atualizado com a hora e data provenientes do mesmo, caso contrário, o mesmo irá funcionar como um relógio, pelo que atualiza a data e hora da sua memória de em tempo real.

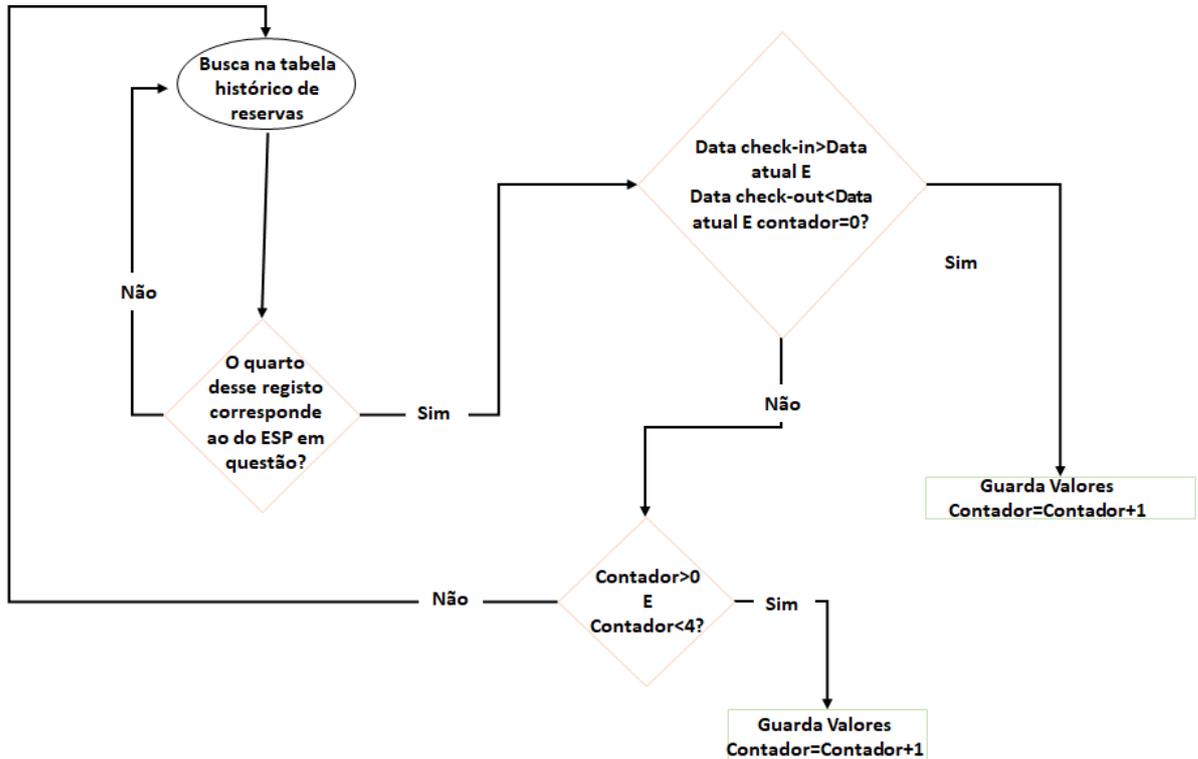


Figura 4.25: Fluxograma de procura de dados na tabela de histórico de reservas

4.3.3 Interface *Web* desenvolvida

Uma automatização completa no setor da indústria turística deve oferecer aos clientes a possibilidade de efetuarem a sua reserva automaticamente em qualquer parte do mundo. Sendo que o objetivo central desta dissertação é fazer com que o processo de entrada num alojamento de serviço turístico se processe de forma completamente automática, o que engloba a fase de reserva do espaço, pagamento do mesmo e por fim, o *check-in* e *check-out* automático, sem a intervenção de terceiros, tornou-se necessário criar uma plataforma *Web* que permita ao cliente efetuar a sua reserva. Na figura 4.26 estão representadas as etapas de registo no sistema e fases necessárias para efetuar uma reserva.

A interface *Web* criada pretende cumprir vários objetivos, entre os quais se destacam os seguintes:

- Divulgar o espaço, recorrendo para tal à utilização de imagens apelativas do mesmo, ou do ambiente que o rodeia.
- Oferecer ao cliente ter informações acerca de preços e quartos disponíveis.
- Recolher informações sobre o cliente, para inserir no sistema e contactar posteriormente caso seja necessário.
- Permitir ao cliente efetuar a reserva de um quarto disponível.

- Possibilitar forma de pagamento *online*.
- Permitir ao proprietário monitorizar cada um dos alojamentos que está a alugar.
- Dar a possibilidade ao proprietário de alterar reservas, ou visualizar qualquer dado relevante da base de dados remotamente.



Figura 4.26: Fluxo de navegação na interface para realizar uma reserva

Para um seguimento das explicações que irão ser feitas acerca das páginas criadas, recomenda-se a consulta das imagens das mesmas, presentes no apêndice B deste documento. A página inicial (figura B.1) possui imagens apelativas do meio envolvente ao local de alojamento, bem como algumas imagens de quartos. Além de pretender chamar a atenção do cliente, esta página possui ligação para outras três páginas. Uma delas é a página de registo no sistema, que pode ser consultada clicando na palavra *sign in*. Esta página possui um formulário com os dados de cliente (figura B.2) e serve para o inscrever no sistema. Desta forma, sempre que este realizar uma reserva poderá ser identificado e contactado caso seja necessário. O formulário desta página possui algumas verificações feitas com funções em *JavaScript*. Deste modo pretendem-se evitar possíveis enganos na marcação dos dados. Algumas das verificações consistem em averiguar se o email introduzido não tem espaços, se possui os símbolos "@" e ".". Verifica também se o número de telefone introduzido possui unicamente símbolos numéricos e se contém pelo menos 9 algarismos, entre outros. Caso os dados não cumpram as verificações impostas, é mostrada uma imagem de alerta ao cliente e o processo de registo no sistema não se concretiza. Caso contrário, é mostrada uma mensagem na página a informar o cliente que se encontra inscrito no sistema e é-lhe enviado um email a confirmar a inscrição. Nessa altura os dados inseridos pelo cliente já estarão na tabela de dados de cliente da

base de dados e este poderá efetuar o seu *login* na página principal. Como forma de evitar a duplicação de dados de cliente, o atributo email foi definido na tabela da base de dados como único, pelo que o mesmo email pode estar presente apenas num registo.

Existem dois tipos de *login* diferente nesta aplicação. O *login* como cliente ou como administrador. As páginas para tal efeito são bastante semelhantes nos dois casos, pois possuem apenas um formulário em que é pedido o email e a senha de acesso (figura B.3). Desta maneira é possível realizar o processo de identificação e autenticação de um indivíduo.

A página de *login* do cliente poderá ser encontrada clicando em "*Login Cliente*" ou carregando nas hiperligações das imagens dos quartos. Caso o email inserido se encontre na tabela de dados de clientes e a senha introduzida seja a correspondente para esse registo, é permitido o acesso ao sistema, onde o cliente poderá procurar os quartos disponíveis de acordo com a sua preferência (figura B.4). Desta forma este poderá escolher o tipo de quarto que procura (suite, individual ou duplo), quantos ocupantes irá ter o referido quarto e para restringir a busca, caso seja do interesse do utilizador, definir um preço máximo a pagar por noite. Quando submetida a busca por quarto, será apresentada uma nova página ao cliente, contendo todos os quartos disponíveis que cumpram os requisitos estabelecidos (figura B.5). Aqui o cliente pode escolher qual o quarto que deseja efetivamente alugar. Para isso terá apenas de carregar no ícone do carrinho de compras e será imediatamente redirecionado para o ambiente *PayPal* (figura B.6) onde deverá iniciar sessão com uma conta, de modo a confirmar o pagamento (figura B.7). Após a confirmação de pagamento, o cliente será redirecionado para uma nova página, onde lhe será dada a indicação se o seu pagamento foi bem sucedido ou não. Caso o pagamento tenha sido bem sucedido, é gerada uma senha de acesso aleatória com 5 caracteres (só possui os caracteres presentes no teclado alfanumérico) e é automaticamente enviado um email ao cliente a informar que a reserva foi bem sucedida, juntamente com a senha de acesso que este deve utilizar durante o seu período de estadia. Nesse momento são preenchidas com mais um registo as tabelas de histórico de reservas e de dados de pagamentos. Desta forma, os únicos dados das tabelas da base de dados que devem ser previamente preenchidos pelo proprietário são os dados relativos a cada quarto, que devem ser inseridos manualmente na base de dados. Contudo esses são à partida dados estáticos (à exceção do preço por noite que poderá ser alterado), pelo que esse preenchimento só terá de ser efetuado uma única vez. De notar que na realização deste trabalho se pressupôs que o serviço de alojamento turístico pudesse ter vários quartos do mesmo tipo (individuais, duplos ou suites) e que mesmo sendo do mesmo tipo, os vários quartos podem diferir entre si, pelo que cada um é identificado por um número único (ID). No momento da reserva o cliente pode escolher qualquer quarto livre, desde que esteja de acordo com os requisitos pretendidos.

No que diz respeito ao administrador, para efetuar o *login* na sua área pessoal, deverá carregar na hiperligação da palavra "*Administrador*" e será reencaminhado para uma página parecida à ilustrada na figura B.3. O processo é idêntico ao de *login* de utilizador, mas neste caso além de ser verificado o email e senha de acesso, será também verificado o campo de estatuto da base de dados e só será dado acesso à área de administrados se o email e senha introduzidos corresponderem a um registo em que o atributo "estatuto" esteja definido como "administrador", caso contrário o acesso é negado. É importante notar que o administrador também possui os privilégios de um cliente, na medida em que se este fizer o *login* na área de cliente poderá efetuar uma reserva no seu

próprio estabelecimento.

Uma vez dado o acesso à área de administrador (figura B.8), este tem um conjunto de ações de controlo/monitorização do sistema que pode realizar, que irão ser abordadas em seguida.

- Monitorização do histórico de reservas: É dada a possibilidade ao administrador de monitorizar todas as reservas efetuadas, pelo que este pode escolher um quarto específico e as datas entre as quais deseja que lhe sejam apresentadas reservas. Após a submissão deste formulário, é apresentada uma tabela (figura B.9) com o histórico de reservas para o quarto selecionada e cujas datas de estadia estão entre as datas previamente selecionadas. O facto de se incluir um formulário antes da apresentação dos dados é feito porque, tendo em conta que a tabela de histórico pode ter um número de registos considerável, desta maneira torna-se mais fácil para o administrador consultar apenas uma parte da tabela.
- Controlo de reservas: Uma reserva pode ter de ser eliminada, por exemplo um cliente comunicar ao proprietário que deseja ver a sua reserva cancelada por um qualquer motivo. Posto isto é essencial que uma reserva possa ser eliminada remotamente pelo proprietário, devendo este identifica-la inequivocamente pelo atributos que compõem a chave candidata da relação da tabela (figura B.11). Desta forma na área de cliente, o quarto que estaria anteriormente indisponível passa automaticamente a estar vago. Além disso, caso os dados de uma reserva tenham de ser alterados, tal operação também deve poder ser feita remotamente pelo proprietário. Assim este poderia alterar datas de *check-in* e *check-out* de uma reserva, bem como o quarto a ser reservado, o cliente e qual a senha de acesso para aquele período de estadia (figura B.10).
- Alteração de preços: Permite ao proprietário alterar os preços por noite de cada um dos quartos que deseja alugar. Para isso terá apenas de escolher o quarto ao qual deseja alterar o preço, bem como o número de ocupantes (figura B.12), visto que existe a possibilidade de praticar diferentes preços para o mesmo quarto, consoante o número de ocupantes do mesmo.
- Monitorização de tabelas diversas: É dada também a possibilidade ao proprietário deste ver remotamente o conteúdo de diversas tabelas da base de dados, nomeadamente a de dados de cliente (figura B.13), dados dos quartos e preços em vigor (figura B.14). Este tipo de dados permite saber quantos clientes estão inscritos no sistema, quais os preços em vigor e qual a tipologia de cada quarto, bem como a *password default* de cada um. Este último atributo mencionado deverá ser inserido manualmente, pelo que essas senhas estarão na memória interna do ESP32 e não está previsto que este envie esses dados para o servidor, visto estes serem fixos. É possível ainda visualizar um histórico de pagamentos (figura B.16) e um histórico de acessos ou tentativas de acesso (figura B.15), de forma a que o proprietário tenha dados suficientes para perceber se o seu espaço está a ser utilizado.
- Monitorização do ambiente de cada espaço: Existem dois módulos a enviar constantemente dados relativos a leituras de vários sensores. Torna-se por isso interessante que esses dados possam ser vistos a qualquer altura de forma gráfica. Posto isto criaram-se gráficos *Highcharts5*, de modo a que se possa ter uma perspetiva do

ambiente de cada divisão (figuras B.18 e B.17). De modo a visualizar apenas uma parte dos dados é pedido o preenchimento de um formulário, no qual se definem o quarto cujos parâmetros se querem consultar e as datas entre as quais se desejam ver os valores. De notar que no caso dos sensores lidos pelo ESP32, houve uma normalização dos valores do sensor de luz, de modo a que estes estejam compreendidos entre "0"(escuridão) e "1"(luminosidade máxima). No caso dos sensores digitais, o sensor de porta a "1"indica que esta se encontra fechada e "0"que se encontra aberta enquanto que "1"no sensor de movimento significa que houve deteção de movimento e "0"que não houve qualquer deteção.

- Monitorização de ocupação: À semelhança do requisito anterior, também esta informação é apresentada de forma gráfica e indica o estado de ocupação de um quarto, num determinado período de tempo, com base em diferentes algoritmos utilizados para deteção de presenças com base na leitura do sensor de som. Nos gráficos pode ler-se o valor "0", caso a divisão seja determinada como vazia ou "1", caso seja determinada como ocupada (figura B.19).

A interface foi criada de forma a que todas as ações sejam realizadas de forma mais automática possível. De facto, de todos os dados relevantes ao funcionamento do sistema, apenas as credenciais de administrador e a tabela de dados dos quartos têm de ser preenchidas manualmente. Contudo estas informações mantêm-se à partida ao longo do tempo, pelo que esse trabalho terá de ser feito apenas uma vez.

4.3.4 Método de pagamento via PayPal

Para uma completa automatização do processo de reservas e *check-in* tornou-se necessário agregar ao projeto um método de pagamento que fosse simples, rápido seguro e intuitivo. Posto isto, optou-se por integrar na interface criada o *PayPal* como forma de pagamento. Esta opção foi tomada, tendo em vista todas as vantagens mencionadas no capítulo 2.1.4.

Desta forma tornou-se necessário interagir com o *PayPal* a partir da interface *Web* criada. Para realizar todos os testes necessários para aferir o bom funcionamento do sistema, foi utilizada uma ferramenta de testes designada de *PayPal Sandbox*, que consiste num ambiente de simulação, em que se podem criar contas de vendedor e comprador fictícias, de modo a poder aferir que todo o processo de pagamento se realiza da forma esperada.

Para facilitar a interação como os serviços da plataforma de pagamento, esta disponibiliza aos programadores uma API (*Application Programming Interface*). Esta é uma ferramenta que permite integrar vários sistemas de forma bastante segura, facilitando a troca de informação entre estes. De uma forma simplista, uma API é uma ponte que liga diferentes aplicações, ainda que para o utilizador comum. Esta ligação é invisível para o utilizador comum e traz diversas vantagens do ponto de vista de implementação, tais como: [84]

- Segurança: A forma como uma API proporciona a integração faz com que a troca de informação seja muito segura, pois esta cria uma espécie de barreira onde apenas um conjunto muito específico de informações está disponível.

- Redução do volume de dados: Pelo facto de uma API vincular apenas um tipo de informação, restringindo o acesso a outras, há uma restrição da manipulação de dados, que possibilita um melhor aproveitamento de informações. Isto faz com que o sistema não fique ocupado com dados desnecessários.
- Auditoria: Permite saber quem teve acesso, quando e o que consultou em determinada API, possibilitando uma perspetiva detalhada quanto ao fluxo de dados através da mesma.

Sendo assim, a ligação entre a interface *Web* criada e o *PayPal* foi feita à custa de uma API. A figura 4.27 mostra esquematicamente o fluxo de informação entre o utilizador, o site criado, a API e o *PayPal*.

O processo inicia-se quando o cliente seleciona a opção de pagamento na interface criada. O processo *SetExpressCheckout* consiste em configurar os parâmetros da transação, tais como valor dos itens, taxas adicionais, quantidades, despesas de transporte, entre outras. Após o envio dessa informação, a API responde com um *TOKEN* que redireciona o cliente para o ambiente *PayPal*, ou para o ambiente *Sandbox* (figura 4.28), caso o processo esteja em fase de testes. Tal como foi mencionado anteriormente, o processo é realizado de forma a que o cliente nem sequer note a existência da API.

Uma vez redirecionado, o cliente terá de entrar na sua conta *PayPal* e autorizar o pagamento. Após o processo de autorização, o cliente irá ser redirecionado novamente para a interface *Web* criada, para uma página definida previamente. Seguidamente, irá haver uma troca de informação entre a API e a interface, residindo aqui uma das vantagens da utilização deste método de implementação. A operação *GetExpressCheckoutDetails* poderá ser feita em qualquer altura, como mostra a figura 4.27 e permite conhecer detalhes do cliente e da transação. A operação Pagamento é a mais importante deste fluxo de informação, pois só através da mesma é que uma transação se transforma em pagamento efetivo e o valor pago pelo cliente irá para a conta do vendedor.

A página da interface para onde o cliente é redirecionado após efetuar o pagamento, também é responsável por inserir na base de dados os detalhes relativos à transação, obtidos na operação *GetExpressCheckoutDetails*. Além disso, essa mesma página deve inserir os dados relativos à reserva efetuada, na base de dados, caso o pagamento seja efetuado, criando assim uma senha de acesso aleatória. Por último um email deve ser enviado ao proprietário a confirmar o pagamento e a facultar a senha de acesso válida nas datas para as quais reservou o espaço.

Para interagir com a API, foi utilizado um *Kit* de desenvolvimento de *software* designado de *PayPal SDK* para PHP [85]. Desta forma a programação e o envio de pedidos para a API foi bastante facilitado. De modo a poder compreender todos os passos necessários para a criação de um ambiente de testes *Sandbox*, recomenda-se a consulta do apêndice C.

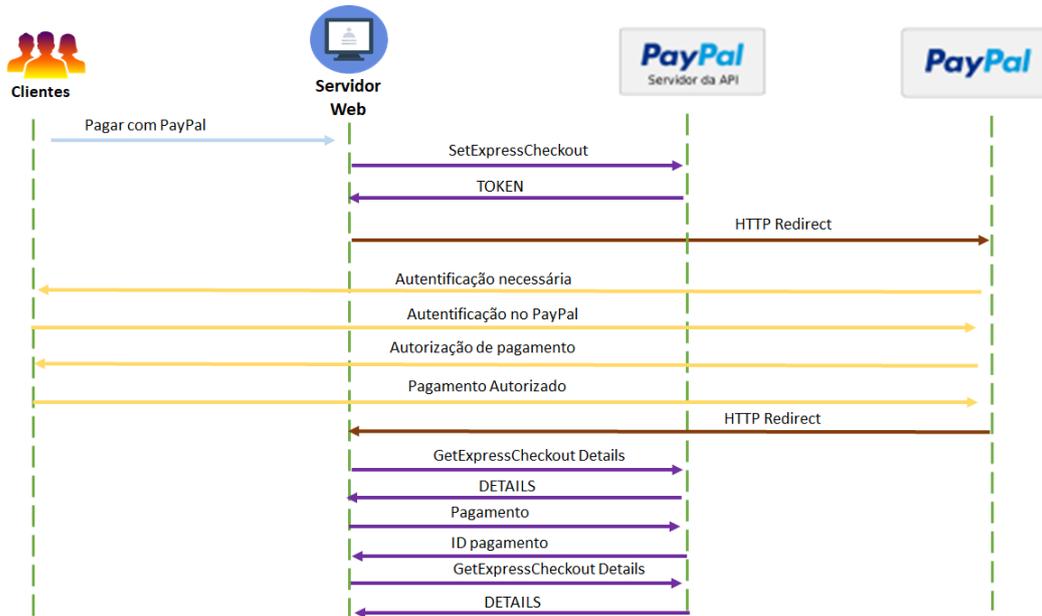


Figura 4.27: Fluxo de interações com a API do *PayPal* [86] [Adaptado]

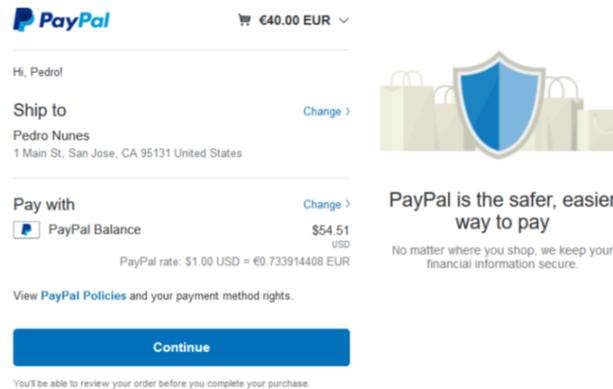


Figura 4.28: Redirecionamento para o ambiente *SandBox*

4.4 Detecção de presenças

Como foi visto no capítulo 2.3, existem numerosas soluções propostas para a detecção de ocupação de um espaço. O pretendido nesta dissertação é a criação de uma forma de saber se uma determinada divisão se encontra ocupada ou não, recorrendo para tal a um microcontrolador e a um conjunto de sensores que não intrusivos para o cliente e que mantivessem baixo o custo da solução.

A utilização de um conjunto de sensores, com a aplicação de algoritmos apropriados foi o que se estudou com vista a conseguir os objetivos, tendo em conta que o proposto é tratar a ocupação como um problema de classificação, isto é, uma determinada divisão pode assumir apenas um de dois estados, "vazia" ou "ocupada". Posto isto a escolha

de sensores e algoritmos de aprendizagem automática foi feita com o intuito de tornar a solução o mais universal possível, de modo a que pudesse ser replicada em qualquer divisão, sem para isso alterar os dados de treino dos algoritmos. A opção de utilizar *machine learning* ou aprendizagem automática deve-se ao facto de inicialmente estar prevista a utilização de um grande número de sensores, com uma grande quantidade de dados associada. Este facto associado ao estado de arte encontrado nesta área ser baseado nesta filosofia, levou a que a opção fosse a utilização deste tipo de algoritmos.

Analisando uma vez mais as soluções abordadas no capítulo 2.3 para a determinação de ocupação, temos que algumas das que foram abordadas não cumprem o requisito da universalidade que se pretende neste caso. A solução proposta por [57] funciona apenas em alguns ambientes, na medida em que a deteção de movimentos no interior de uma divisão depende bastante das dimensões da mesma, dado o limitado alcance dos sensores PIR. A solução proposta por [59] foi testada num ambiente bastante específico, pois o laboratório utilizado para fazer os testes não possui luz natural, pelo que um sensor de luz nessa situação se torna um indicador bastante forte, o que não acontece noutros locais com iluminação natural. A solução proposta por [58] pareceu ser a que mais se assemelhava ao pretendido. Apesar de não conseguir determinar o número de ocupantes de uma divisão, este autor conseguiu segundo o mesmo, determinar se um determinado espaço está ou não ocupado com grande exatidão. Além disso a variedade de sensores utilizados permite que a solução seja global o suficiente para ser aplicada em qualquer divisão.

Tendo como referência o trabalho proposto por [58] a primeira tentativa de determinar a ocupação, durante esta dissertação, consistia na utilização de sensores de CO₂, de luz, de movimento e de porta ligados ao ESP32. Pretendia-se com os valores lidos dos diferentes periféricos distinguir em que situações estaria ou não ocupado um espaço. Numa fase inicial de testes, recolheram-se valores destes quatro sensores e armazenaram-se na base de dados, de modo a que se pudesse ter uma ideia da maneira como a ocupação de um local influenciava os valores. Nessa fase de testes observaram-se algumas coisas importantes. A primeira das quais, foi que devido ao espaço de testes (laboratório do DEM) ter uma área envidraçada grande, na maior parte do dia o sensor de luz estaria com valores de luminosidade alta. O pretendido com este tipo de sensor era perceber se uma lâmpada estaria ligada, associando uma luz artificial à presença humana no local. Contudo apenas seria possível garantir tal facto se não houvesse luz natural, o que não se verifica para a maioria dos locais. Além disso uma lâmpada acesa não significa obrigatoriamente que um espaço esteja ocupado, pode querer apenas dizer que alguém se esqueceu de a desligar quando deixou a divisão vazia.

O local de testes possui uma área considerável, pelo que o conjunto composto pelo ESP32 e pelos sensores, nessa fase de testes foi colocado em cima de uma das mesas do laboratório. Notou-se que devido ao facto de o espaço ser de grande dimensão, na maior parte das vezes os movimentos dos ocupantes não eram detetados por se encontrarem longe do sensor, ou por existirem objetos entre o mesmo e os ocupantes. Contudo, considerou-se que os dados lidos pelo por este periférico pudessem ser interessantes, visto que como era possível colocar os sensores no interior de uma divisão, logo sempre que fosse detetado movimento pelo PIR, significava que esse espaço estaria ocupado.

Como os valores do sensor de luz e de movimento nem sempre se alteravam com a ocupação de um espaço, esperava-se o os valores do sensor de CO₂ pudesse de alguma forma contribuir de uma maneira mais decisiva para uma correta classificação. Contudo,

observou-se durante a fase de testes que os valores deste sensor não diferiam, estando o espaço ocupado ou não. Pensou-se que tal fenómeno se devesse ao facto de que sempre que o laboratório onde foram realizados os testes estava ocupado, a porta do mesmo estava aberta, podendo a ventilação natural devida à abertura da porta justificar o não aumento dos níveis de CO₂ medidos pelo sensor. Posto isto realizaram-se novos testes no mesmo laboratório, mas com a porta fechada. Nestes notou-se que estando uma pessoa relativamente perto do sensor, o nível de CO₂ medido por este aumenta ligeiramente, mas só passados 30 minutos. Mesmo tendo uma pessoa próxima, o tempo que decorreu até que os valores do sensor aumentassem foi demasiado grande, o que fez com que se analisasse a utilidade do mesmo, pois a sua utilização não dava grande contributo para a classificação que se pretendia, além disso o facto de o espaço poder ter sistemas de ventilação não natural, ou de a divisão ter janelas abertas, diminuem ainda mais a eficácia do dispositivo. Outro aspeto negativo é o facto de ter um preço excessivo para uma aplicação que se pretende de baixo custo. Todos estes motivos fizeram com que se abandonasse a sua utilização.

Os sensores de porta, movimento e luz dão de facto algumas indicações acerca do espaço onde estão inseridos, contudo para alcançar o objetivo de classificar o estado de ocupação de uma divisão, são necessários parâmetros mais gerais, que possam garantir que um espaço está ocupado, independentemente de condições ambientais variáveis, como a dimensão do espaço em análise, hora do dia ou localização geográfica do mesmo. Posto isto parâmetros como temperatura ou humidade foram descartados, pois estão dependentes de parâmetros que variam ao longo do tempo e de região para região.

Um parâmetro que está diretamente ligado à ocupação é a existência de som na divisão. A voz do ser humano tem características que a distingue dos demais sons ou ruídos. Além disso quando um local está ocupado, é normal existirem ruídos no seu interior, que de outra forma não existiriam, tais como arrastar de cadeiras, barulho de dispositivos como televisão, computador, entre outros. Dadas essas condições, resolveu-se utilizar um sensor de som com vista a cumprir o objetivo de classificar um local quanto à presença humana no interior do mesmo.

O sensor de som utilizado (figura 4.20) fornece na entrada analógica de um microcontrolador um valor proporcional à pressão sonora exercida num ponto, sendo que a sensibilidade do mesmo pode ser ajustado com um potenciômetro que o mesmo tem integrado. Apenas com esse parâmetro torna-se difícil distinguir os sons típicos de um local ocupado de um local vazio, pelo que foi necessário um processamento do sinal recebido, antes de este ser analisado pelos algoritmos de classificação.

De modo ter um processamento de sinal mais eficaz e a poder utilizar esta solução sem o módulo de controlo de acessos, por razões já mencionadas, utilizou-se o ESP8266 para processar e enviar para a base de dados os sinais vindos do sensor de som.

Ao nível do processamento feito no ESP8266 importa perceber a importância de distinguir sons provocados pela presença humana, como o falar, arrastar objetos no interior da divisão, entre outros, de sons que em nada têm a ver com a presença de alguém naquela divisão, como pessoas a conversar do lado de fora da mesma, carros a passar na rua, o vento, etc.

Os fenómenos sonoros estão associados à vibração de partículas materiais, que se propaga através de ondas longitudinais. Estas são frequentemente designadas por ondas sonoras e podem ser caracterizadas por diversas grandezas (figura 4.29) como a amplitude (distância vertical entre um ponto médio da onda e um ponto máximo ou mínimo),

o comprimento de onda (distância entre dois pontos no mesmo estágio de vibração) e frequência (número de vezes que a onda se repete por segundo, é medida em Hz). Sabendo a frequência de onda de um determinado som seria possível distinguir diferentes tipos de sons e sabendo a sua amplitude seria possível saber se a intensidade corresponde a um som no interior ou exterior da divisão. Contudo o que se obtém do sensor de som não são ondas sinusoidais, mas sim ondas quadradas, em que a amplitude é tanto maior quanto maior for a intensidade do som.

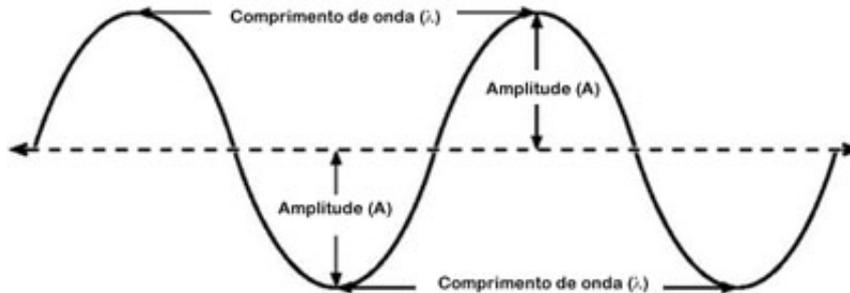


Figura 4.29: Representação de uma onda sinusoidal

De modo a poder tratar o sinal recebido pelo sensor e a perceber quais as frequências e amplitudes de um determinado som, houve a necessidade de recorrer à série de Fourier, para aproximar o sinal do sensor por um somatório de ondas sinusoidais, com frequências e amplitudes diferentes, como mostra a equação 4.3, onde $F(t)$ é a função que se pretende aproximar, a_0 , a_n e b_n dependem da função original e são as amplitudes de cada onda que compõe o somatório. A variável L define metade do período da função original.

$$F(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n * \cos(\frac{n\pi t}{L}) + b_n * \sin(\frac{n\pi t}{L})] \quad (4.3)$$

Desta forma quanto maior o número de parcelas do somatório utilizadas para aproximar a função original, mais correta será a aproximação feita. A figura 4.30 mostra uma onda quadrada a ser aproximada sucessivamente por um maior número de sinusoides. Enquanto que o sinal mais à esquerda que é aproximado apenas por uma parcela do somatório tem um desvio considerável da função inicial, a representação mais à direita mostra uma aproximação feita com quatro parcelas, que já se ajusta bastante melhor à função original.

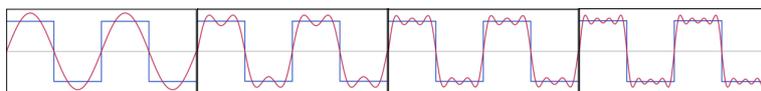


Figura 4.30: Aproximação de uma onda quadrada por um somatório de funções sinusoidais

Para aproximar o sinal recebido da maneira que ilustra a figura 4.30 foi utilizada a biblioteca "arduinoFFT" disponível para o ambiente de programação utilizado. Com esta é possível utilizar o algoritmo FFT (Fast Fourier Transform). Este calcula a série de Fourier, reconstruindo um sinal (mesmo que não seja periódico) de forma bastante precisa pela adição de sinais sinusoidais com frequências e amplitudes distintas, sendo que quantas mais sinusoides forem utilizadas, mais parecido com o sinal original ficará a aproximação.

Para a utilização do algoritmo FFT é necessário ter algum cuidado a definir algumas grandezas que são de grande importância para um melhor desempenho do mesmo na aplicação pretendida, com por exemplo:

- Número de amostras: Consiste no número de leituras seguidas que devem ser feitas da entrada analógica do ESP8266. Estas irão formar o sinal de entrada que deve ser aproximado por sinusoides. O número escolhido deve ser uma potência de base 2. Um número mais elevado de amostras torna o algoritmo mais lento, contudo permite uma melhor resolução dos resultados.
- Frequência de amostragem: De acordo com o teorema da amostragem de Nyquist Shannon, para se poder amostrar um sinal, a frequência de amostragem precisa ser pelo menos o dobro da frequência do sinal que se está a tentar detetar. Por outras palavras, o algoritmo só será capaz de detetar frequências até metade da frequência de amostragem.
- Período de amostragem: É unicamente dependente da frequência de amostragem, sendo o inverso desta grandeza.

Os sons produzidos pela fala de um ser humano estão compreendidos entre os 100 e os cerca de 3000 Hz, como mostra a figura 4.31. De modo a analisar grande parte das frequências ilustradas, utilizou-se uma frequência de amostragem de 4 kHz, de maneira a conseguir detetar frequências até 2 kHz, desta forma podem ser detetados sons, como o de pessoas a falar e outros barulhos dentro da divisão. Como o ESP8266 está responsável apenas pelo processamento de sinal do sensor de som e pelo envio de dados para a base de dados, escolheu-se um número de 1024 amostras, de modo a ter uma boa resolução.

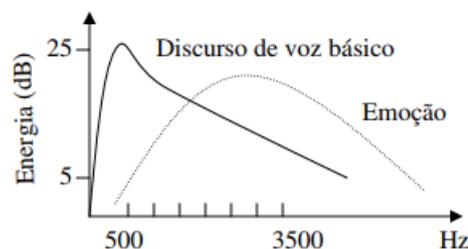


Figura 4.31: Frequências típicas da voz humana [87]

Processando o sinal de entrada com o algoritmo FFT é obtido para cada conjunto de amostras um conjunto de sinusoides com frequências e amplitudes diferentes. Com base

a simplificar o sinal resultante, foram recolhidas as 5 sinusóides mais significativas, isto é, de entre os pares de frequências e amplitudes obtidos selecionaram-se os as 5 ondas com maior amplitude.

Numa primeira fase, para averiguar se realmente seria possível distinguir um espaço ocupado de um espaço vazio a partir dos harmónicos mais significativos de um sinal, fez-se o seguinte processamento no ESP8266, antes de enviar os dados para a base de dados:

- Recolheram-se 1024 amostras de sinal, com uma frequência de 4 kHz.
- Eliminou-se algum do ruído, filtrando o sinal.
- Aplicou-se o algoritmo FFT.
- Guardaram-se as cinco frequências mais significativas e respetiva amplitude, na gama entre os 70 Hz e os 1900 Hz. Desta maneira ruídos de fundo como vento, carros a passar na rua, entre outros seriam atenuados.
- De modo a uma melhor manipulação dos dados, criou-se uma variável nova, que se designou de "intensidade sonora". O uso desta expressão pode ser abuso de linguagem, mas serve apenas para uma mais fácil explicação do procedimento adotado. Sejam as frequências mais significativas recolhidas de $H_1, H_2... H_n$, com $n=5$ e sejam as respetivas amplitudes designadas de $A_1, A_2... A_n$. A intensidade sonora (I_s) foi então calculada segundo a equação 4.4.

$$I_s = \sum_{n=1}^5 [A_n * H_n]$$

(4.4)

- Repetiu-se o processo 100 vezes e no final foi calculada a média de todas as intensidades sonoras obtidas. Como os valores obtidos eram de ordens de grandeza bastante elevadas, houve a necessidade de dividir o valor final por uma constante (10^5), de modo a melhor manipulação dos valores.

Após o processamento descrito anteriormente, a média das várias medições foi enviada para a base de dados. De modo a avaliar a utilidade dos dados criados para a deteção de presenças, numa primeira fase, foram feitos gráficos da intensidade sonora (média de 100 medições) ao longo do tempo, para perceber se os valores deste parâmetro variavam conforme o estado de ocupação do local. Pela análise gráfica dos dados foi possível verificar que estes eram bastante diferentes, dependendo da ocupação do local. A figura 4.32 mostra os dados recolhidos durante um dia inteiro. As áreas marcadas a vermelho na figura representam períodos de ocupação enquanto que as áreas a verde representam intervalos de tempo em que o espaço de testes esteve vazio.

A análise gráfica permite perceber se o processamento e manipulação de dados foi adequado com vista a cumprir os objetivos, contudo, tornou-se necessário criar um modelo matemático de classificação dos dados obtidos, de modo a determinar o estado de

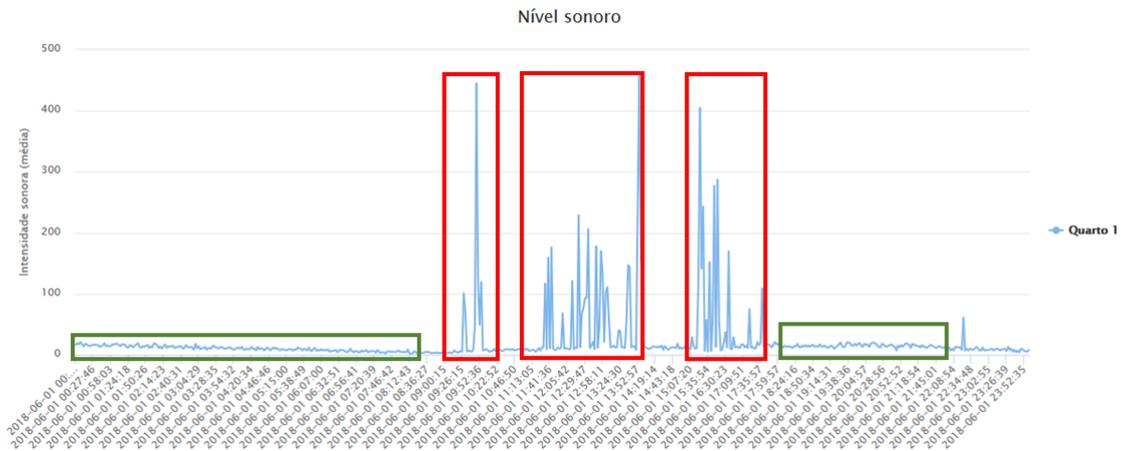


Figura 4.32: Intensidade sonora ao longo de um dia

ocupação de uma divisão. Esta necessidade decorre do facto de nem sempre ser fácil identificar visualmente o estado de ocupação de um local, principalmente se os dados a analisar não tiverem diferentes estados associados. Por exemplo, a figura 4.33 mostra os níveis medidos durante um sábado, em que o laboratório esteve vazio e com a janela aberta. Olhando apenas para este período, sem ter conhecimento de dados antigos, por exemplo, os ilustrados na figura 4.32, torna-se difícil afirmar se as medições correspondem a um período de ocupação ou de ausência de pessoas no local. Contudo, comparando os gráficos, é possível ver que os valores máximos para uma divisão vazia são bastante abaixo, em comparação com um local ocupado por pessoas. Desta forma, houve a necessidade de criar um algoritmo que classificasse automaticamente os dados, de modo a que a pessoa que monitoriza o local não tenha de ter conhecimentos prévios de quais os valores para os quais o espaço está ocupado.

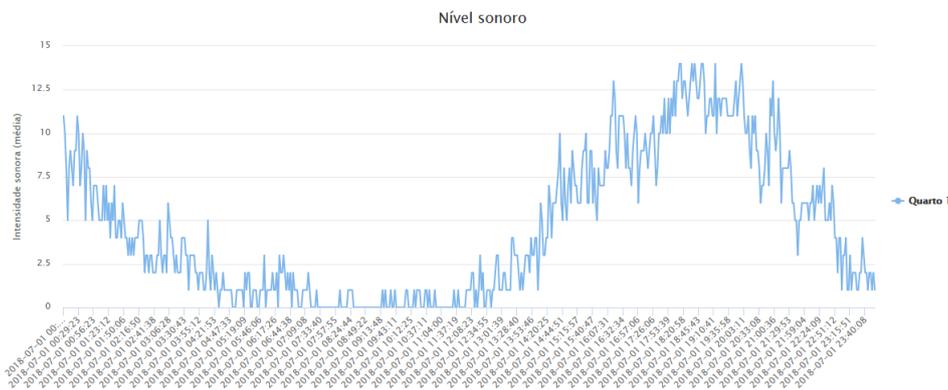


Figura 4.33: Intensidade sonora ao longo de um dia sem qualquer ocupação

Como descrito anteriormente, de início tentaram-se utilizar diferentes sensores, de modo a criar um algoritmo com base na filosofia da aprendizagem automática, cujos

parâmetros de entrada fossem os diferentes valores normalizados dos mesmos, seguindo uma abordagem idêntica à proposta por [58]. Contudo, o que estava em análise eram dados de apenas um sensor. A análise com base em apenas um valor torna-se difícil, pois mesmo em períodos de ocupação existem momentos de menor intensidade de som. Posto isto para uma classificação mais correta seria necessário analisar um padrão de amostras recolhidas e não apenas o último valor, procedimento idêntico ao realizado proposto por [57], na medida em que este autor analisou padrões de um único sensor (sensor de movimento) para classificar a ocupação de um local.

Para analisar um padrão tornou-se necessário definir o número de amostras que deviam formar esse mesmo padrão. Com base na análise visual dos gráficos obtidos, estimou-se que cinco medições seriam um valor adequado para classificar se num determinado momento os dados obtidos correspondem a um espaço vazio ou ocupado.

Para a criação do algoritmo que melhor se assemelhasse à realidade dos resultados que foram sendo recolhidos, foram testados três métodos de classificação diferentes, as árvores de decisão, a classificação Bayesiana e o algoritmo de SVM. Para a utilização de uma abordagem de aprendizagem automática, foi utilizada a biblioteca "PHP ML", que permite a criação de algoritmos de uma maneira intuitiva em linguagens *Web* (PHP). A figura 4.34 exemplifica a utilização desta biblioteca. A primeira etapa consiste em requisitar um conjunto de ficheiro com código fonte que são criados no momento de instalação da biblioteca. Seguem-se os passos normais de construção de um algoritmo baseado em aprendizagem automática, ou seja este é treinado com dados de exemplo, sendo que são introduzidos nesta fase os *inputs* e os *outputs* conhecidos para o conjunto de dados de entrada. Após essa fase o algoritmo é construído e pode classificar novos dados.

```

38
39 use Phpml\Exception\InvalidArgumentException;
40 use Phpml\Exception\InvalidOperationException;
41 use Phpml\Exception\LibsvmCommandException;
42 use Phpml\SupportVectorMachine\Kernel;
43 use Phpml\SupportVectorMachine\SupportVectorMachine;
44 use Phpml\SupportVectorMachine\Type;
45 use PHPUnit\Framework\TestCase;
46 use Phpml\Classification\NaiveBayes;
47 use Phpml\ModelManager;
48
49 $samples = [[552,85,251,7,6],[19,26,166,145,6,6],[78,10,11,11,53],[162,16,12,358,1928],[241,12,11,11,11],[14,11,12,12,12],
50 [11,12,12,12,12],[12,12,12,13,12],[81,24,11,11,12],[11,12,13,12,69],[3627,6554,6794,2483],[3888,16,11,19,13],
51 [13,131,22,12,24],[100,25,52,12,17],[17,12,13,83,13],[149,11,12,12,12],[20,10,10,10,6],[6,8,8,10,7],[6,6,6,8,8],
52 [8,6,65,9,7],[7,8,7,21,7],[8,10,709,199,63],[6,159,5,223,9],[691,41,22,63,44],[7,108,7,6,139],[11,10,121,246,23],
53 [4,10,426,650,475],[10,426,650,475,78],[267,20,20,9,176],[10,1270,4661,4859,9624],[3161,2593,899,806,176],
54 [474,214,353,58,22],[131,185,320,305,910],[295,318,138,137,44],[44,47,331,9,415],[60,68,71,71,11,11],[71,71,11,12,256],
55 [10,273,4,154,8],[10,707,11,97,12],[2740,6022,4489,376],[13,113,10,10,13],[13,10,12,11,10],[44,1,0,1,0],
56 [143,10,177,9,68],[7,9,10,12,56],[56,12,11,8,8],[42,9,6,99,12],[171,9,11,8,5],[100,14,14,596,9],[34,97,3,4,6],[61,57,6,8,4]];
57 $labels = ['ocupado','ocupado','ocupado','ocupado','vazio','vazio','vazio','vazio','vazio','vazio','ocupado','ocupado','ocupado','ocupado',
58 'vazio','vazio','vazio','vazio','vazio','vazio','vazio','vazio','vazio','ocupado','ocupado','ocupado','ocupado','ocupado',
59 'ocupado','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado',
60 'ocupado','ocupado','vazio','vazio','vazio','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado','ocupado'];
61
62 $classifier1 = new SupportVectorMachine(Type::C_SVC, Kernel::LINEAR, 100.0);
63 $classifier1->train($samples, $labels);
64 $SVM=$classifier1->predict([$valores[0],$valores[1],$valores[2],$valores[3],$valores[4]]);
65 $classifier2 = new NaiveBayes();
66 $classifier2->train($samples, $labels);
67 $Naive=$classifier2->predict([$valores[0],$valores[1],$valores[2],$valores[3],$valores[4]]);

```

Figura 4.34: Exemplo de programação de algoritmos a partir de dados conhecidos

A criação de uma árvore de decisão é feita por um processo indutivo, sendo que a maioria dos métodos de construção das mesmas começam o processo de cima para baixo, ou seja, desde o primeiro nó até à classificação final. Para tal, utilizam os dados

de treino fornecidos e com base nestes, procuram iterativamente o melhor atributo que separa as classes para realizar a ramificação da árvore e recursivamente processam os subproblemas resultantes de cada ramo. Desta maneira, sempre que todas as instâncias (conjunto de amostras) restantes pertencem à mesma classe, o nó é terminal e pode efetuar a classificação. Se durante o processo iterativo a lista de atributos (neste caso um atributo é um dos 5 valores médios que compõem o padrão) chegar a estar vazia num momento em que todas as instâncias ainda não pertençam à mesma classe, estas são classificadas de acordo com a classe mais frequente no conjunto das instâncias. Por outras palavras, temos um conjunto de dados de treino recolhido durante o período de testes. Cada instância corresponde a um padrão de 5 leituras que deve ser classificado. Na fase de treino, a classificação para cada instância é conhecida e iterativamente o algoritmo de aprendizagem automática irá tentar criar regras que dividam as instâncias nas respetivas classificações, criando para isso ramificações, de modo a construir a árvore recursivamente. No presente problema, existem apenas duas classes "ocupado" e "vazio", que determinam o estado de um espaço.

Por sua vez, a classificação Bayesiana tem na sua génese um processo estatístico, baseado no teorema de Bayes (equação 2.1). Em problemas de classificação interessa determinar $P(c|x)$, ou seja, a probabilidade da classe "c" dada a instância "x". Os dados de treino são utilizados para determinar as probabilidades $P(c)$ e $P(x)$. Desta forma de acordo com o teorema de Bayes é possível calcular a probabilidade posterior para cada classe. Isto é, é calculada a probabilidade de uma instância (conjunto de 5 leituras do sensor de som) pertencer à classe "ocupado", bem como a probabilidade de esta pertencer à classe "vazio". A probabilidade que for maior determina a classe atribuída pelo algoritmo. Como $P(x)$ (probabilidade de uma dada instância) é constante para todas as classes, podemos simplificar a equação 2.1 e obter a equação 4.5.

$$P(c|x) \propto P(x|c) * P(c) \quad (4.5)$$

A probabilidade de uma dada classe é dada pelo quociente entre o número de dados de treino que pertence a essa classe e o número total de dados utilizados para treino do algoritmo. Uma instância é, neste caso um vetor com 5 medições (descritores). Para reduzir o custo computacional, considera-se que os descritores são independentes, o que neste caso é um pressuposto aceitável. Como as grandezas utilizadas são contínuas, $P(x|c)$ pode ser calculada partindo do pressuposto que assume uma distribuição Gaussiana de acordo com a equação 4.6. Onde μ_{ci} e σ_{ci} são, respetivamente, a média e o desvio padrão, calculados considerando os valores de um determinado descritor A_j das instâncias que pertencem à classe c_i .

$$P(x_j|c_i) = (1/\sqrt{2\pi}\sigma_{ci}) \frac{e^{-(x_j-\mu_{ci})^2}}{2\sigma_{ci}^2} \quad (4.6)$$

Um regressão linear pode ser utilizada quando a tarefa é a classificação de dados numéricos com é o caso. A ideia fundamental deste tipo de técnica é expressar uma classe como uma combinação linear de atributos da forma que indica a equação 4.7, onde

c é a classe, $\alpha_1, \alpha_2, \dots, \alpha_n$ são os valores dos atributos ($n=5$, para esta aplicação, pois cada instância tem 5 atributos) e w_1, w_2, \dots, w_n são os pesos de cada atributo.

$$c = w_0 + w_1\alpha_1 + w_2\alpha_2 + \dots + w_n\alpha_n \quad (4.7)$$

Apesar da sua grande simplicidade e grande aplicabilidade na área da estatística, a grande desvantagem da regressão linear é que caso os limites entre classes não sejam lineares, a tarefa de classificação torna-se pouco precisa com estes modelos. Posto isto, o algoritmo de SVM utiliza de forma mais robusta modelos lineares, mesmo para classes não linearmente separáveis [24].

Para se entender melhor o conceito de linearmente separável, podemos atentar no exemplo da figura 4.35 onde existem duas classes "+" e "o". Neste caso as duas grandezas podem ser separadas geometricamente por uma reta, por isso são linearmente separáveis. O algoritmo de SVM visa a criação de um hiperplano que separa linearmente as grandezas, com máxima margem de separação. Neste caso o hiperplano é uma reta, pois existem apenas dois atributos (x_1 e x_2), contudo, caso haja mais atributos a dimensão do hiperplano vai ser maior, por exemplo, no caso específico dos padrões de som, existem 5 atributos, logo o hiperplano vai ter uma dimensão de $5-1=4$. Às instâncias mais próximas do hiperplano dá-se o nome de vetores de suporte e durante o treino são estes vetores que vão adquirir maior importância, pois estarão mais perto da linha de separação de classes.

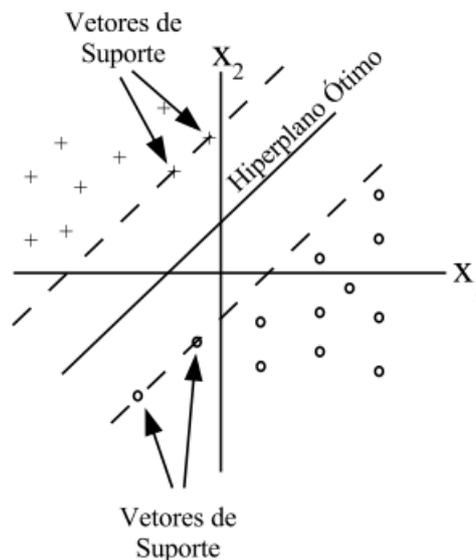


Figura 4.35: Separação de classes através de um hiperplano [24]

Se considerarmos um conjunto de treino definido por x_i, c_i , onde x_i é o vetor de atributos e c_i a classe da instância i . Como existem duas classes, caso estas sejam linearmente separáveis, podemos distingui-las fazendo $c_i = +1$ ou $c_i = -1$, consoante a classe seja "ocupado" ou "vazio". Desta maneira a equação de separação pode ser escrita de acordo com a equação 4.8, onde " w " é uma matriz de pesos e " x " é um vetor com os

atributos de cada instância.

$$w^T x + b = 0 \quad (4.8)$$

Desta forma, os dados de treino são utilizados para determinar computacionalmente os valores de "w" e de "b", de modo a que seja encontrado um hiperplano ótimo. No caso em que estes não sejam linearmente separáveis, é utilizado um mapeamento não linear que transforma o espaço amostral num novo espaço, onde as instâncias são linearmente separáveis [24].

A aprendizagem automática funciona como um construtor automático de algoritmos, calculando os parâmetros necessários através de um conjunto de dados fornecidos (instâncias), juntamente com a classe conhecida de cada um. Numa primeira fase, com vista a uma primeira aproximação, foram recolhidas 62 instâncias para períodos em que a ocupação era conhecida. Importa notar que nem sempre que o laboratório onde foram efetuados os testes se encontrava ocupado, os valores de intensidade sonora obtidos era diferentes de uma sala vazia. Tal deve-se ao facto de que por vezes um determinado local tem uma ocupação de carácter silencioso, por exemplo pessoas a fazer trabalho de escritório em silêncio não irão fazer com que os níveis de ruído se alterem. Nessas situações não é expectável que o algoritmo criado classifique corretamente os dados. Por essa razão, no conjunto de todos os dados recolhidos, evitou-se o uso desse tipo de instâncias, que podem ser designadas de *outliers*.

Uma primeira forma de perceber se os algoritmos se estão a adaptar bem aos dados de treino, é realizar uma previsão com esse conjunto de dados. Esta análise apenas fornece uma primeira impressão do grau de ajustamento do algoritmo, sendo que muitas vezes um bom ajuste neste tipo de dados é sinónimo de *overfitting* e não de um bom ajuste aos dados. A tabela 4.2 mostra a exatidão de cada algoritmo quando foi testado para os dados de treino, calculada pelo quociente do número de previsões certas, pelo número de previsões total.

Para uma primeira fase, as três abordagens revelam um bom ajustamento aos dados de teste, contudo era necessário testar os algoritmos criados a partir dos dados de treino em novas amostras. Posto isto, ao mesmo tempo que eram enviados para a base de dados novos valores vindos do ESP8266, eram executados os três algoritmos, classificando um padrão composto pelas 5 últimas leituras de uma determinada divisão.

Verificou-se, que apesar de ter uma grande exatidão nos dados de treino, à árvore de decisão não se adaptava muito bem a novos dados, fornecendo classificações que não correspondiam minimamente à realidade. O algoritmo de SVM produzia melhores resultados, contudo em algumas instâncias, sobretudo aquelas que se encontravam mais próximas do limiar de separação, produzia resultados insatisfatórios. O algoritmo Bayesiano classificava corretamente sempre que um espaço estivesse vazio, contudo, quando se encontrava ocupado, as previsões tinham algumas falhas ainda.

Nesta altura, com base nos resultados obtidos, podia-se concluir que muito provavelmente a árvore de decisão estaria com um problema de sobreajustamento dos dados de treino, pois era detetada ocupação ainda que os valores do sensor de som não justificassem tal classificação. Quanto aos restantes algoritmos, tudo levava a crer que poderiam melhorar o seu desempenho para novos valores se fossem utilizados mais dados para treino. Posto isto, acrescentaram-se novos dados, para os quais os algoritmos haviam falhado, com vista a melhorar a performance dos mesmos. Desta maneira, aos dados

anteriormente recolhidos, foram acrescentadas novas amostras, perfazendo um total de 110 instâncias. De seguida foram testados novamente os algoritmos nas 110 amostras de treino.

A terceira coluna da tabela 4.2 mostra os resultados obtidos. A árvore de decisão voltou a classificar corretamente todas as instâncias, o que não quer dizer forçosamente que tenha um bom desempenho para novos dados. Face ao 1^o treino tanto os classificadores de SVM e Bayesiano obtiveram piores resultados. No caso do algoritmo Bayesiano, como as instâncias acrescentadas para formar o novo conjunto de treino foram sobretudo amostras para as quais o algoritmo havia falhado numa fase de observação, esperava-se, que mesmo tendo uma menor percentagem de acertos na fase de treino, este se adaptasse melhor aos novos dados a serem lidos, visto que, quando testado no primeiro grupo de dados de treino (62 instâncias), este novo modelo produziu os mesmos resultados que o antigo. Quanto ao algoritmo de SVM, foi assumido que as instâncias eram linearmente separáveis, logo, o facto de o seu desempenho ter diminuído no novo teste deve-se ao facto de terem sido utilizados dados mais próximos do hiperplano de separação, o que impossibilitou a criação de um hiperplano que separasse todas as instâncias de treino. Contudo, quando testado para o primeiro grupo de treino, este novo modelo manteve o mesmo desempenho do que o modelo gerado com menos dados, sendo expectável que para novos dados o algoritmo tivesse melhor eficácia, visto terem sido utilizados mais dados que poderiam ser vetores de suporte.

Como forma de aferir com maior certeza a qualidade dos algoritmos criados, foi comparada a ocupação real durante um dia (24h) com os resultados das classificações obtidas por cada um dos modelos obtidos. Importa referir que não é de todo expectável uma eficácia de 100%, pois em casos em que o laboratório onde foram efetuados os testes esteja ocupado, mas os seus ocupantes em silêncio, é expectável que o modelo falhe. Deste modo deve ser feita uma análise crítica tendo em conta esse facto.

A exatidão de classificação durante um dia inteiro em que se realizou a experiência, encontra-se na quarta coluna da tabela 4.2. O algoritmo Bayesiano e de SVM produziram resultados semelhantes ao da última fase de treino, enquanto que a eficácia da árvore de decisão baixou claramente face aos testes em amostras de treino. Apesar disso, foi o algoritmo com melhor percentagem de acertos, 83,11 % das classificações feitas por este algoritmo ao longo do dia foram corretas.

Além de se olhar para os números apresentados, é de interesse fazer uma análise gráfica dos acontecimentos. A figura 4.37 mostra graficamente a ocupação determinada por cada um destes modelos e a ocupação real do dia em questão. Para se ter uma ideia mais abrangente importa ainda comparar estes resultados com os valores obtidos pelo sensor de som no mesmo período de tempo. A ocupação real corresponde à ocupação do laboratório no DEM, num dia normal de trabalho, em que o primeiro ocupante da sala chegou cerca das 9h e 36min, estando a respetiva sala sempre ocupada até às 19h e 22min. Ao longo desse período de ocupação, houve intervalos de tempo em que os níveis de intensidade sonora foram baixos, devido ao facto de que os ocupantes estavam sobretudo a fazer trabalho de escrita, havendo por isso períodos de grande silêncio.

O algoritmo Bayesiano sempre que detetou ocupação, a sala estava de facto ocupada, contudo era expectável, que face aos valores do sensor de som, este detetasse mais vezes a presença humana, revelando um melhor ajuste dos dados. O classificador SVM, apesar de não ter sido o modelo com melhor percentagem de acertos, ajustou-se bem aos dados lidos pelo sensor de som, não sofrendo de qualquer problema de *overfitting* dos dados de

treino, pois a percentagem de acertos nos dados de treino e em novos dados foi idêntica, além disso, não houve nenhuma falsa ocupação determinada, pois sempre que a sala era determinada como ocupada, tal correspondia à realidade. A árvore de decisão, apesar de ter sido o algoritmo com melhores resultados demonstrou um ligeiro sobreajuste de dados, pois à 1h e 27min determinou um período de ocupação, o que não corresponde a realidade, nem os dados de intensidade sonora nesse instante fazem com que se justifique tal classificação. Para perceber se os modelos criados estavam de facto com problemas de *overfitting*, testaram-se os mesmos num dia em que não houve qualquer ocupante no laboratório. Os dados recolhidos pelo sensor são os da figura 4.33 e os resultados obtidos das classificações estão presentes na figura 4.36. Como se pode verificar apenas a árvore de decisão detetou períodos de ocupação no espaço de tempo em que a sala esteve vazia, mesmo que os valores obtidos pelo sensor não justificassem tal classificação, demonstrando assim mais uma vez um problema de sobreajuste, não sendo o modelo mais confiável, apesar de no primeiro teste ter sido o modelo com melhor precisão.

Apesar de não ser o algoritmo com melhor percentagem de acerto, o classificador SVM é o que oferece mais garantias, pois não deteta falsas ocupações, tendo-se adaptado bastante bem aos dados introduzidos. Contudo o facto de haver períodos de ocupação, em que os ocupantes da divisão se encontram em silêncio faz com que este tipo de modelos não tenha um desempenho melhor face à ocupação real. Contudo para a aplicação que se propões que é permitir ao proprietário ter uma visão geral da ocupação de espaços e permitir perceber em que dias esteve um quarto alugado, a solução proposta tem um desempenho satisfatório.

O problema da árvore de decisão de se ajustar demasiado aos dados de treino pode estar relacionado com o facto de os dados utilizados serem contínuos, pelo que à partida se houvesse uma divisão dos mesmos por grupos, esse problema ficaria resolvido. O mesmo serve para o classificador Bayesiano, pois como é baseado no cálculo de probabilidades, a divisão dos dados por classes poderia aumentar a eficácia preditiva deste modelo. Contudo, o máximo que se conseguiria seria uma eficácia preditiva a rondar os valores obtidos pelo algoritmo de SVM, sem sobreajustamento de dados. Isso acontece, porque os níveis de intensidade sonora não são suficientes para classificar com maior eficácia o estado de ocupação de uma divisão, fazendo falta outro parâmetro ambiental para a obtenção de melhores resultados.

Ainda assim, estes resultados podem ser replicados para qualquer outra divisão, visto que mesmo numa divisão de grande dimensão, os valores de intensidade sonora aumentam caso haja alguém a falar no seu interior, ainda que longe do sensor. Além disso, dado o processamento de dados feito no microcontrolador, sons exteriores à divisão são suficientemente atenuados para não serem confundidos com intensidade sonora decorrente de ocupação do local, demonstrando que o som foi um bom parâmetro ambiental para a classificação que se pretendia, contudo, para melhores resultados seriam necessários mais parâmetros.

Tabela 4.2: Desempenho dos classificadores

Algoritmo	Exatidão do 1º treino (%)	Exatidão do 2º treino (%)	Exatidão teste 24h (%)
Classificação Bayesiana	90,32	67,27	70,36
Árvore de decisão	100	100	83,11
SVM	95,16	82,73	77,15



Figura 4.36: Desempenho dos algoritmos criados, quando testados num dia sem qualquer ocupação

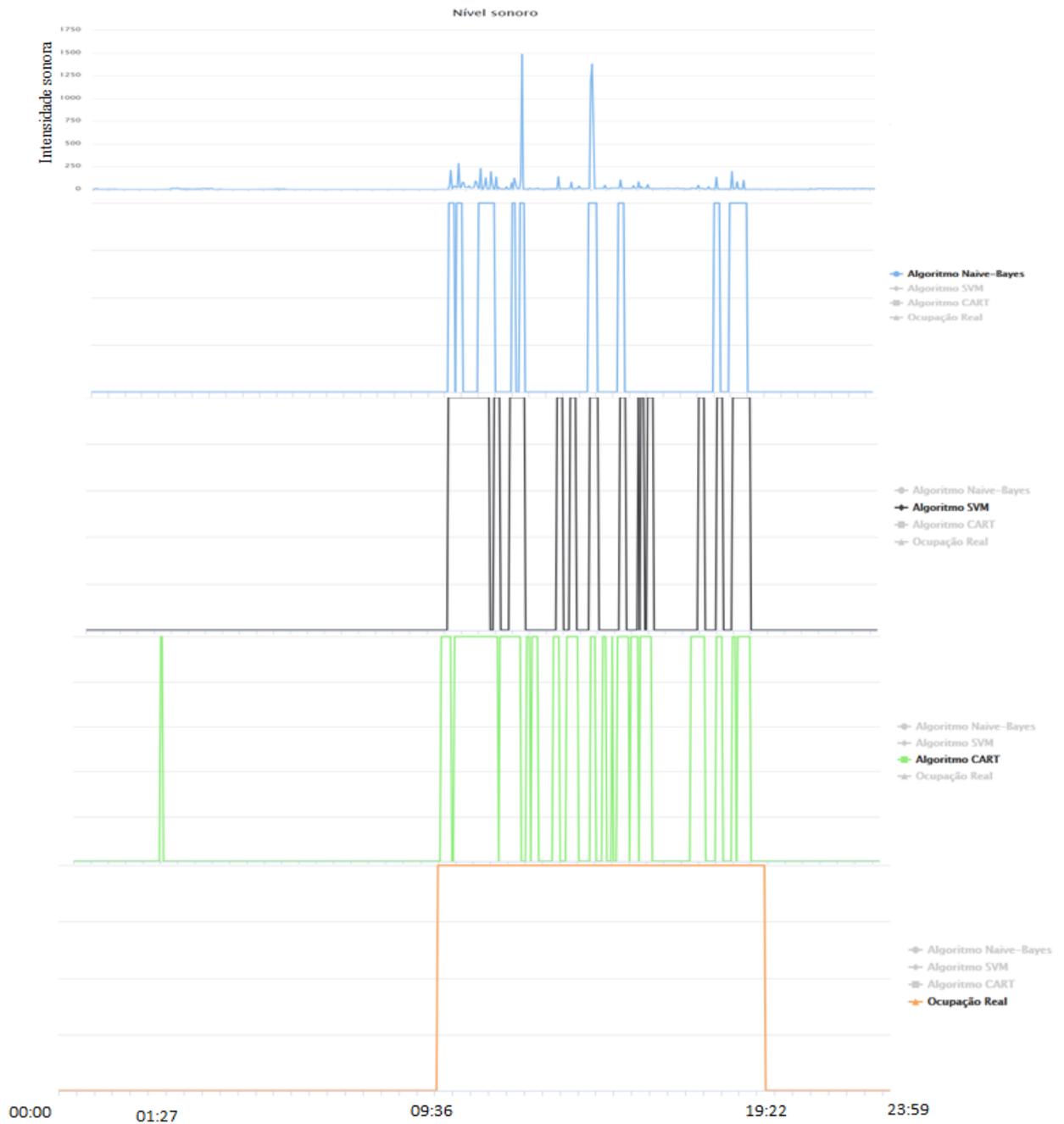


Figura 4.37: Ocupação ao longo de 24h e nível sonoro correspondente

4.5 Montagens finais

Após o desenvolvimento de todo o *software* e *hardware* necessário para a criação de uma solução inovadora, houve a necessidade de realizar a montagem de todos os componentes, de forma a testar o seu funcionamento e robustez. A figura 4.38 mostra a montagem dos diversos componentes do módulo de controlo de acessos. Neste consta um módulo relé opto-isolado, que fica no interior do espaço a ocupar, bem como a placa PCB desenvolvida para acomodar os sensores e circuito de alimentação. No exterior deve ficar o módulo que contém os LEDs indicativos e o teclado alfanumérico. A alimentação é disponibilizada a partir de um transformador 12V e a testa com fecho elétrico deve ficar montado na porta de acesso.

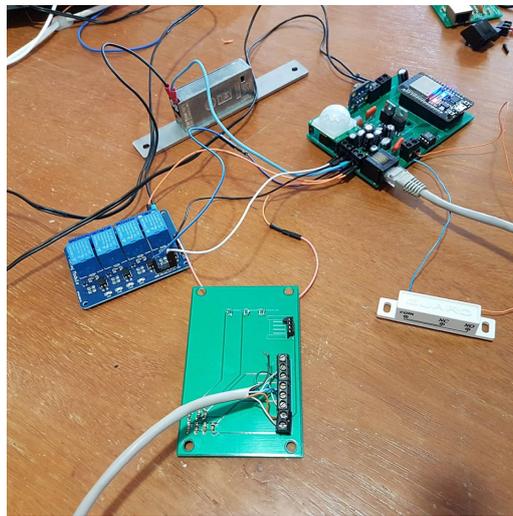


Figura 4.38: Montagem da solução proposta para controlo de acessos

Apesar de todos os componentes se encontrarem montados numa placa de circuito impresso, aumentando a robustez da montagem, a solução ainda não se aproximava o suficiente de um protótipo para a sua implementação em ambiente real. Posto isto, após pesquisa, foi encontrada no mercado uma placa de desenvolvimento (figura 4.39) que contém um microcontrolador ESP8266 embutido, bem como um circuito que permite a alimentação a partir de 12V e 2 relés que permitem a passagem dessa mesma corrente e estão atracados a saídas digitais do ESP8266. Este módulo tem um preço bastante acessível, pois custa cerca de 5 euros [88] e traz ainda uma pequena caixa que permite encapsular o módulo a ficar no interior. Posto isto, é proposta uma alternativa para o controlo de acessos, baseada nesta placa. Esta nova solução contempla apenas o controlo de acessos propriamente dito, pelo que não tem conectado nenhum sensor. Este facto, apesar de contribuir para uma diminuição dos parâmetros ambientais que são monitorizados, torna o controlo de acessos propriamente dito mais rápido, pois os pedidos que o ESP8266 faz ao servidor podem ser feitos com uma cadência menor, aumentando a rapidez do sistema.

A figura 4.40 mostra a montagem de todos os componentes do sistema. Como se pode ver, o módulo a ficar no interior da divisão é bastante mais compacto e além disso já se encontra encapsulado. O módulo do teclado foi também encapsulado numa

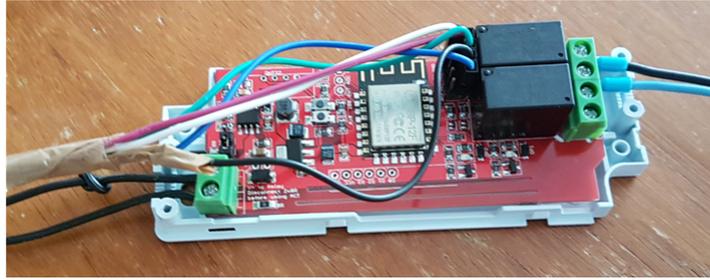


Figura 4.39: Placa com módulo 8266, relés e circuito de alimentação

caixa apropriada, que possui uma saída para os fios de ligação entre os módulos e um envidraçado, de modo a que o utilizador consiga ver os LEDs de indicação a acenderem. Para esta solução, foi utilizado um teclado idêntico ao mencionado na secção 4.2.2, mas com menos uma coluna na matriz de teclas, para estar de acordo com a caixa em que foi encapsulado. Assim, o teclado fino está colocado entre a placa PCB desenvolvida e a caixa exterior. Desta forma, sempre que o utilizador pressiona uma tecla, irá exercer pressão no teclado colocado no interior que está ligado ao PCF8574, que por sua vez está ligado ao ESP8266 no interior da divisão.

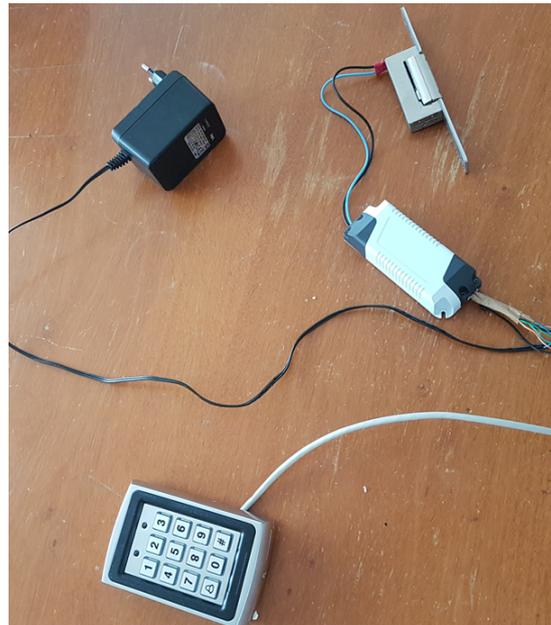


Figura 4.40: Montagem final do módulo de controlo de acessos

Sendo assim, foram reunidas todas as condições para a implementação em ambiente real da solução, que está representada na figura 4.41. Nela se podem ver o módulo interior (canto superior esquerdo), o teclado no exterior da divisão (canto superior direito), o transformador de 12V que alimenta todo o sistema (canto inferior esquerdo) e a porta equipada com a testa com fecho elétrico (canto inferior direito). Desta maneira é possível afirmar que foi criado um protótipo pronto a implementar em ambiente real.



Figura 4.41: Montagem da solução em ambiente real

O módulo de deteção de ocupação é por si só bastante compacto, pelo que é composto apenas por um ESP8266 e um sensor de som, pelo que para o tornar mais robusto e pronto a implementar em qualquer local, este foi encapsulado numa caixa apropriada, como mostra a figura 4.42, sendo a sua alimentação assegurada por um transformador de 5V ligado na porta micro USB do microcontrolador. Desta maneira este módulo pode ser colocado em qualquer local, desde que esteja alimentado e consiga captar sinal *Wi-Fi*.

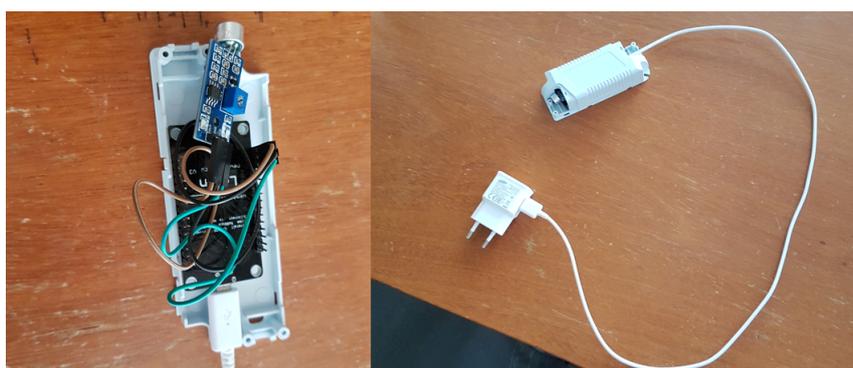


Figura 4.42: Montagem do módulo de deteção de presenças

Capítulo 5

Conclusões e trabalho futuro

A automatização está hoje em dia presente em quase todas as áreas, pois permite uma otimização dos processos de modo a que estes sejam mais rápidos e possam gerar maior lucro. No que toca à indústria turística, continua a haver a necessidade de existência de funcionários na receção, pois tarda em ser comercializado um sistema que permita que o acesso a um espaço alugado possa ser efetuado de forma automática, sem a interferência de terceiros no processo. Esse facto, faz com que o investimento em pequenos negócios de alojamento possa não ser tão rentável como seria de esperar, pois o trabalho de receção exige muitas vezes que se façam 3 turnos de 8 horas de trabalho para cobrir as 24h de um dia. Como os pequenos serviços de alojamento possuem menos capacidade, normalmente, não conseguem ser um negócio com lucros tão apelativos como seria de esperar, pois o investimento em funcionários faz com que o espaço tenha de estar praticamente sempre lotado para que o retorno seja tão significativo com se pretende.

As soluções comerciais existentes nesta área não fornecem todos os requisitos que se pretendem num método de controlo de acessos que visa ser totalmente automático. Posto isto, o objetivo principal deste trabalho é a criação de uma solução de baixo custo baseada na utilização de um sistema embutido controlado por um microcontrolador para automatizar processos na indústria turística. Assim, para a construção do modelo proposto foram gastos cerca de 20 euros com o módulo de controlo de acessos exterior, o que inclui a caixa metálica que envolve o teclado (solução mais cara), sendo que utilizando outro tipo de caixa para o teclado, o custo possa diminuir substancialmente. O módulo interior de controlo de acessos (microcontrolador) custou cerca de 5 euros. No que diz respeito ao módulo de deteção de presenças, foram gastos cerca de 6 euros. A estes custos acresce-se o custo com a alimentação dos dispositivos, pelo que os transformadores de 5V e 12V respetivamente utilizados para deteção de presenças e controlo de acessos custaram em conjunto cerca de 10 euros. Para a implementação foi necessária a utilização de um computador que desempenhasse o papel de servidor. Caso o estabelecimento que pretenda optar pela solução proposta não possua este equipamento, este será outro investimento a ter de ser realizado, sendo que será necessário apenas um computador independentemente do número de quartos, enquanto que os módulos de controlo de acessos e de deteção de presença deverão estar presentes em cada quarto, representando um investimento proporcional ao número de quartos.

A solução proposta oferece um método totalmente automático do processo de alojamento de clientes, que vai desde a reserva do espaço até ao *check-in* e *check-out*. A interface *Web* criada permite que um cliente efetue a sua reserva em qualquer parte do

planeta, inclusive junto da porta do estabelecimento, bastando para isso que este tenha um dispositivo com ligação à Internet e um *browser*. O pagamento também é feito de forma automática, pois foi integrada na aplicação a plataforma *PayPal* através de uma API. Os testes realizados utilizando o *PayPal Sandbox* mostraram um funcionamento correto desta interação, proporcionando assim ao cliente uma maneira fácil e segura de pagamento *online*. Apesar dos ensaios de pagamento terem sido feitos num ambiente de testes, as alterações a efetuar para passar para um ambiente real são muito poucas, bastando mudar as credenciais de comunicação com a API para credenciais de uma conta *PayPal* de comerciante.

No que diz respeito ao *hardware*, foi feita a montagem de todos os componentes nas placas de circuito impresso desenhadas. Desta forma, foi possível testar a robustez de funcionamento dos mesmos. Para aferir o desempenho do sistema, o protótipo foi posto em funcionamento durante vários dias e revelou um comportamento robusto. Houve mesmo a realização de várias simulações, tendo-se feito a reserva e pagamento no site criado e em seguida testou-se se o código fornecido permitia abrir a porta, tendo-se verificado um correto funcionamento de todo o sistema, pois a senha obtida com a reserva apenas permite o acesso no período para o qual foi realizado o aluguer. Para os testes efetuados, foi utilizado um computador presente no laboratório do DEM, para desempenhar o papel de servidor. Este encontra-se ligado à rede da UA (Universidade de Aveiro), pelo que algumas aplicações são bloqueadas pelos serviços informáticos. Sendo assim a função de envio automático de emails é bloqueada. Contudo a mesma aplicação foi testada noutro computador ligado a uma rede diferente da UA, verificando-se um correto funcionamento da mesma.

Além de permitir o controlo de acessos, era imperativo que os valores dos sensores fossem enviados para uma base de dados de modo a serem visualizados pelo proprietário. Nesse sentido, a interface *Web* possui páginas de monitorização acessíveis ao proprietário que lhe permitem ver gráficos dos valores dos sensores ao longo do tempo. Os testes de comunicação com a base de dados e servidor foram bem sucedidos. No que diz respeito aos sensores propriamente ditos, fornecem indicações importantes do ambiente de cada quarto, no entanto, na montagem final o sensor de movimento ficou perto de uma fonte de calor (regulador de tensão LM 7805), ficando os valores do mesmo comprometidos devido ao seu princípio de funcionamento baseado em raios infravermelhos.

De modo a permitir que funcionários e proprietário possam aceder em qualquer altura a qualquer divisão, foi criada uma senha válida que se encontra na memória do microcontrolador. Deste modo o pessoal que trabalha no serviço turístico não necessitaria de andar constantemente a decorar novas senhas. Por motivos de segurança e monitorização do local, todos os acessos são registados automaticamente na base de dados, permitindo ao proprietário saber se o acesso foi feito com a senha de cliente ou com a senha geral. Da mesma maneira todas as tentativas de acesso não autorizadas são também registadas (tentativas de acesso com senhas inválidas), permitindo uma completa monitorização do espaço.

O facto de a gestão do espaço turístico poder ser feita à distância e se pretender dispensar os serviços de associados à receção, fez com que se tornasse importante perceber se um espaço estava ocupado. Pretendia-se inicialmente que fosse possível também otimizar serviços de limpeza, pois seria possível saber quando um espaço estaria vazio. Como se percebeu a potencialidade de aplicação da solução noutras áreas, criou-se um módulo com o intuito de que este funcionasse como um detetor de ocupação de uma

divisão.

A determinação de ocupação proposta com base no som, nem sempre deteta a presença de pessoas, visto que uma divisão pode ser ocupada e as pessoas no seu interior estarem a realizar uma atividade que não produza qualquer som, como ler ou escrever. Sendo assim o módulo não cumpre o objetivo de realizar uma classificação precisa em tempo quase real, contudo os modelos matemáticos propostos, sobretudo o algoritmo SVM produzem resultados satisfatórios face aos valores obtidos pelo sensor. De facto a determinação de ocupação não é mais precisa, porque tal requer a introdução de outro parâmetro ambiental que ajude na classificação, pois como se verificou, nem sempre os níveis sonoros variam com a ocupação (no caso de os ocupantes estarem em silêncio). Concluiu-se ainda que sempre que era detetada ocupação, encontrava-se de facto alguém na divisão, ou tinha estado há muito pouco tempo, o que permite ao proprietário ter uma ideia geral da ocupação de um quarto ao longo do tempo. A utilização deste tipo de solução parece ser também viável no caso de uma indústria de alojamento turístico cujas reservas não sejam feitas de modo automático, pois permite saber em que dias esteve cada quarto alugado, evitando possíveis falhas de preenchimento de mapas de presença e facilitando a tarefa do gestor na contabilidade ao final do mês.

Dos vários testes efetuados, foi possível concluir que o algoritmo de FFT e todo o processamento de sinal aplicado ao sinal do sensor de som permite a criação de um parâmetro (intensidade sonora) que permite distinguir sons no interior de uma divisão dos sons exteriores, como pessoas a conversar no exterior ou carros a passar na rua. Isso faz com que apesar de não ser possível determinar o estado de ocupação em tempo real apenas com este indicador ambiental, se possa ter um grande confiança no modelo matemático SVM, sempre que este determina uma divisão como ocupada. Quando este a classifica como vazia é que surgem algumas falhas pois isso pode querer apenas dizer que os ocupantes no interior da divisão estão em silêncio. Semelhantes conclusões se podem tirar para o modelo Bayesiano, ainda que este tenha percentagens de acerto bastante abaixo do anterior algoritmo. O modelo de árvores de decisão revelou problemas de sobreajustamento de dados, apesar de ter revelado grandes percentagens de acerto, contudo o algoritmo mais robusto é o de SVM, pois as árvores de decisão em alguns casos classificam uma divisão ocupada, sem que esta esteja ou os níveis sonoros indiquem tal. Os modelos Bayesiano e das árvores de decisão poderiam apresentar contudo melhores resultados se os dados contínuos fossem agrupados por classes. Tal requereria um trabalho de divisão dos dados em vários grupos, o que é desnecessário, visto que o máximo que se conseguiria era aproximar os resultados destes aos obtidos pelo algoritmo de SVM, devido a falta de mais indicadores ambientais. Assim, apenas com um descritor conseguiu-se criar um módulo bastante compacto, de baixo custo e de fácil instalação que permite de uma maneira geral ao proprietário ter noção de períodos em que o seu espaço esteve ocupado, embora para uma informação mais detalhada, os algoritmos necessitem de mais descritores ambientais.

A otimização dos horários de limpeza com base na determinação de ocupação não se revela uma tarefa possível com a utilização do módulo proposto, contudo este não é um problema que necessite de grande intervenção, visto que o funcionário de limpeza poderá sempre confirmar a presença de alguém batendo à porta do respetivo quarto. Tal facto não invalida a importância do módulo proposto, visto que quando integrado em conjunto com a solução principal permite um maior grau de monitorização do espaço pelo proprietário, na medida em que permite perceber se o espaço alugado está de facto

a ser ocupado. Em estabelecimentos em que a fase de reserva não é feita unicamente de maneira automática, o módulo em questão pode ser utilizado para aferir os dias em que o espaço esteve alugado, evitando possíveis erros no preenchimento de folhas de ocupação.

De uma maneira geral, os objetivos propostos neste trabalho foram cumpridos, existem no entanto algumas melhorias que podem ser feitas, tendo em vista uma continuidade deste trabalho que passam por:

- **Melhoria da interface do teclado:** Foram utilizados LEDs para dar indicações visuais ao cliente, para que este se aperceba de quando foi detetada uma tecla, se a senha que marcou estava correta ou não. Foram utilizadas 3 saídas digitais para esse fim. O protótipo ficaria mais apelativo com a integração de um LCD com interface I2C. Desta forma seriam utilizados apenas dois GPIOs do microcontrolador, que poderiam ser os mesmos da comunicação com *IO expander* (ligação em barramento com endereços I2C diferentes) e o cliente poderia ter uma noção do processo de forma mais apelativa.
- **Tornar a solução mais modular:** A solução proposta consiste em 2 módulos, um para controlo de acessos e monitorização e outro para deteção de ocupação, contudo a hipótese de juntar toda a rede de sensores num módulo e utilizar o módulo de controlo de acessos apenas para a leitura do teclado alfanumérico e comunicação com a base de dados, pode tornar o sistema global mais rápido, na medida em que o módulo de controlo de acessos não necessitaria de efetuar ligação com o servidor com uma cadência tão grande.
- **Utilização de mais parâmetros para determinar a ocupação:** O modelo proposto para deteção de ocupação é suficiente para a aplicação em causa, contudo a determinação de ocupação de uma forma precisa ao longo do tempo tem uma aplicabilidade em quase todos os locais de serviços e habitação, pois esse é um parâmetro muito importante na otimização de consumos com sistemas AVAC, por essa razão pode ser de interesse utilizar valores de mais sensores para criar um algoritmo mais preciso.

Bibliografia

- [1] UNWTO. Why tourism? | World Tourism Organization UNWTO. [Accessed on: 2018-02-14]; 2015. Available from: <http://www2.unwto.org/content/why-tourism>.
- [2] Infographics | World Tourism Organization UNWTO. [Accessed on: 2018-02-14]; Available from: <http://media.unwto.org/content/infographics>.
- [3] UNWTO. UNWTO Annual Report 2016; 2017. Available from: <http://cf.cdn.unwto.org/sites/all/files/pdf/annual{ }report{ }2016{ }web{ }0.pdf>.
- [4] INE. Estatísticas do Turismo 2016; 2017. Available from: <https://www.ine.pt/xportal/xmain?xpid=INE{ }xpgid=ine{ }publicacoes{ }PUBLICACOESpub{ }boui=277048338{ }PUBLICACOESStema=55581{ }PUBLICACOESmodo=2>.
- [5] Economia Online. Turismo responde por 7% da economia nacional. [Accessed on: 2018-02-14]; 2017. Available from: <https://eco.pt/2017/12/07/turismo-responde-por-7-da-economia-nacional/>.
- [6] on: 2018-02-14] TA. BI - Dormidas e Hóspedes por tipologia 2010|2015; 2016. Available from: <http://travelbi.turismodeportugal.pt/pt-pt/BI/paginas/default.aspx?datasetId=C2832151DAA84CCE95CED397C1B459AF>.
- [7] Financeiros M. Evolução do Turismo Nacional; 2016. Available from: <http://www.bancobpi.pt/nocachecontent/conn/UCM/uuid/dDocName:PR{ }WCS01{ }UCM01031805>.
- [8] Schwab PK, Martin C, Samans R, Moavenzadeh J, Drzeniek-Hanouz M. The Travel & Tourism Competitiveness Report 2017: Paving the way for a more sustainable and inclusive future. The Travel & Tourism Competitiveness; 2017. Available from: <http://www3.weforum.org/docs/WEF{ }TTCR{ }2017{ }web{ }0401.pdfhttp://ev.am/sites/default/files/WEF{ }TTCR{ }2017.pdf{ }0Ahttps://www.weforum.org/reports/the-travel-tourism-competitiveness-report-2017>.
- [9] Carvalho PI. Sistema de monitorização de veículos de mercadorias [Tese de Mestrado]. Universidade de Aveiro; 2014. Available from: <http://hdl.handle.net/10773/15726>.
- [10] Moreira PAT. Gestão de controlo de acessos [Tese de Mestrado]. FEUP- Faculdade de Engenharia do Porto; 2009. Available from: <https://repositorio-aberto.up.pt/handle/10216/57899>.

- [11] Deidmar GLC, Sobreira DdS, de Lima WD. Internet das Coisas na Educação. *Revista Tecnologias em Projeção*. 2018;8(2):67–78.
- [12] Gabbai A. Kevin Ashton Describes "the Internet of Things". *SMITHSONIAN MAGAZINE*. 2015; Available from: <https://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/http://www.smithsonianmag.com/innovation/kevin-ashton-describes-the-internet-of-things-180953749/?no-ist>.
- [13] Editors S. Investing in EHealth. June; 2010. Available from: <https://unifiedinbox.com/investing-in-iot/>.
- [14] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013 sep;29(7):1645–1660. Available from: <https://www.sciencedirect.com/science/article/pii/S0167739X13000241>.
- [15] Domingos P. *A Revolução do Algoritmo Mestre*. 5th ed. Manuscrito, editor; 2018.
- [16] Gama J. *Árvores de Decisão Sumario Árvores de Decisão*; 2002. Available from: http://www.dcc.fc.up.pt/~ines/aulas/MIM/arvores{}_de{}_decisao.pdf.
- [17] Tin Kam Ho. Random decision forests. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. vol. 1; p. 278–282. Available from: <http://ieeexplore.ieee.org/document/598994/>.
- [18] Flaxman AD, Vahdatpour A, Green S, James SL, Murray CJ, Population Health Metrics Research Consortium (PHMRC). Random forests for verbal autopsy analysis: multisite validation study using clinical diagnostic gold standards. *Population health metrics*. 2011 aug;9(1):29. Available from: <https://link.springer.com/content/pdf/10.1023%7D2FA%7D3A1010933404324.pdfhttp://www.ncbi.nlm.nih.gov/pubmed/21816105http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3160922>.
- [19] Friedman JH. *A Gradient Boosting Machine*;
- [20] Eddelbuettel D. *A Gentle Introduction to Rcpp*. In: *Seamless R and C++ Integration with Rcpp*; 2013. p. 3–18. Available from: http://www.chengli.io/tutorials/gradient{}_boosting.pdfhttp://link.springer.com/10.1007/978-1-4614-6868-4{}_1.
- [21] Holland S, Griffiths R, Woodman M. Avoiding object misconceptions. *ACM SIGCSE Bulletin*. 1997 mar;29(1):131–134. Available from: http://seer.ufrgs.br/index.php/rita/article/viewArticle/rita{}_v14{}_n2{}_p43-67http://portal.acm.org/citation.cfm?doid=268085.268132.
- [22] Shmilovici A. Support Vector Machines. In: *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US; 2010. p. 231–247. Available from: http://link.springer.com/10.1007/978-0-387-09823-4{}_12.

- [23] 6 passos fáceis para aprender o algoritmo Naive Bayes. [Acceded on: 2018-04-16];. Available from: <https://www.vooo.pro/insights/6>.
- [24] Mersmann LHDC. Classificação Probabilística Baseada em Análise de Padrões [Tese de Doutorado]. Universidade Federal Fluminense; 2007.
- [25] Goulart DA, Tacla MA, Marback PMF, Solé D, Paranhos A, Perez HB, et al. Redes neurais artificiais aplicadas no estudo de questionário de varredura para conjuntivite alérgica em escolares. Arquivos Brasileiros de Oftalmologia. 2006;69(5):707–713. Available from: <http://www2.ica.ele.puc-rio.br/Downloads/33/ICA-introdu{ç}{~}{a}oRNs.pdf>.
- [26] Guias Teóricas | Vantagens e Desvantagens do Paypal - NOS. [Acceded on: 2018-03-03];. Available from: <http://www.nos.pt/empresas/repositorio-informacao/criar-uma-empresa/guias-teoricos/Pages/vantagens-desvantagens-paypal.aspx>.
- [27] PayPal. PayPal Developer. [Acceded on: 2018-03-03]; 2018. Available from: <https://www.paypal-brasil.com.br/desenvolvedores/tutorial/testando-as-apis-paypal-no-sandbox/https://developer.paypal.com/docs/integration/direct/express-checkout/integration-jsv4/>.
- [28] Paypal developers. [Acceded on: 2018-02-27];. Available from: <https://developer.paypal.com/docs/classic/paypal-payments-standard/integration-guide/paymentdatatransfer/>.
- [29] How do barcodes and barcode scanners work? - Explain that Stuff. [Acceded on: 2018-02-20];. Available from: <http://www.explainthatstuff.com/barcodescanners.html>.
- [30] Rodrigues LGP, Heidemann LA, De Carli E, da Silveira LGM. Decodificando o código de barras. Física na Escola. 2011;12(2):24–7. Available from: <http://www.lume.ufrgs.br/bitstream/handle/10183/116459/000966276.pdf?sequence=1http://www.sbfisica.org.br/fne/Vol12/Num2/a07.pdfhttp://www.lume.ufrgs.br/bitstream/handle/10183/116459/000966276.pdf?sequence=1%}0Ahttp://www.sbfisica.org.br/fne/Vol12/Num2/a>.
- [31] Códigos de barras UPC-A - Simbologia nos códigos de barras. [Acceded on: 2018-05-10];. Available from: <https://www.cognex.com/pt-br/resources/symbologies/1-d-linear-barcodes/upc-a-barcodes>.
- [32] GS1. Norma GS1;. Available from: <http://syncpt.gs1pt.org>.
- [33] Gao J, Kulkarni V, Ranavat H, Chang L, Mei H. A 2D barcode-based mobile payment system. In: 3rd International Conference on Multimedia and Ubiquitous Engineering, MUE 2009. IEEE; 2009. p. 320–329. Available from: <http://ieeexplore.ieee.org/document/5318908/>.
- [34] Kieseberg P, Leithner M, Mulazzani M, Munroe L, Schrittwieser S, Sinha M, et al. QR code security. In: Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia - MoMM '10. IEEE; 2010.

- p. 430. Available from: <http://ieeexplore.ieee.org/document/4698844/http://portal.acm.org/citation.cfm?doid=1971519.1971593>.
- [35] Kieseberg P, Leithner M, Mulazzani M, Munroe L, Schrittwieser S, Sinha M, et al. QR code security. In: Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia - MoMM '10; 2010. p. 430. Available from: http://delivery.acm.org/10.1145/1980000/1971593/p430-kieseberg.pdf?ip=193.137.168.32&id=1971593&acc=ACTIVESERVICE&key=2E5699D25B4FE09E.861C198C983DE13B.4D4702B0C3E38B35.4D4702B0C3E38B35&{}_{}_acm{}_{}_={}_533fbb9759d9dfbd78c200df6ab61d94http://portal.ac.
- [36] Huang PC, Chang CC, Li YH, Liu Y. Efficient access control system based on aesthetic QR code. Personal and Ubiquitous Computing. 2018 feb;22(1):81–91. Available from: <http://link.springer.com/10.1007/s00779-017-1089-y>.
- [37] Berz EL. Funcionamento do RFID em crachás inteligentes - dissertação; 2011. Available from: http://repositorio.pucrs.br/dspace/bitstream/10923/1551/1/000430794-Texto%7BCompleto-0.pdfhttp://tede.pucrs.br/tde{}_busca/arquivo.php?codArquivo=3393.
- [38] Finkenzerler K. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and near-Field Communication. Wiley; 2010.
- [39] Margaret Rouse [Acceded on: 2018-02-21]. What is RFID (radio frequency identification)? - Definition from WhatIs.com; 2017. Available from: <http://www.radio-electronics.com/info/wireless/radio-frequency-identification-rfid/technology-tutorial-basics.phphttp://internetofthingsagenda.techtarget.com/definition/RFID-radio-frequency-identification>.
- [40] VIBHOR S, Gusain P, PRASHANT K. Near Field Communication. Advances in Intelligent Systems Research. 2013;2013(Cac2s):342–345.
- [41] O que é NFC (Near Field Communication)?. [Acceded on: 2018-02-22]; Available from: <https://www.infowester.com/nfc.phphttp://www.infowester.com/nfc.php>.
- [42] Poole I. NFC Tags and Tag Types. [Acceded on: 2018-02-22]; Available from: <http://www.radio-electronics.com/info/wireless/nfc/near-field-communications-tags-types.php>.
- [43] Wayman J, Jain AK, Maltoni D, Maio D. An Introduction to Biometric Authentication Systems. In: Biometric Systems; 2005. p. 1–20. Available from: https://link.springer.com/content/pdf/10.1007/1-84628-064-8{}_1.pdfhttp://www.springerlink.com/content/m1647w5773136432/fulltext.pdf.
- [44] Standalone Keypad Access Control User Manual; Available from: <https://www.allsecurityequipment.com/manuals/Stand-Alone-Access-Controller-SS-TS2000.pdf>.

- [45] Parallax. 4x4 Matrix Membrane Keypad datasheet; 2011. 916.
- [46] Pimentel DRC. Controlo de acessos no sector turístico [Tese de Mestrado]. Universidade de Aveiro; 2015.
- [47] Carmo J. Gestão e Controlo de Acessos no Setor Hoteleiro [Tese de Mestrado]. Universidade de Aveiro; 2014. Available from: <http://hdl.handle.net/10773/13832>.
- [48] Calçado SFJ. Automatização e controlo de reservas na indústria hoteleira [Tese de Mestrado]. Universidade de Aveiro; 2017.
- [49] Teh TY, Lee YS, Cheah ZY, Chin JJ. IBI-Mobile Authentication: A Prototype to Facilitate Access Control Using Identity-Based Identification on Mobile Smart Devices. *Wireless Personal Communications*. 2017 may;94(1):127–144. Available from: <http://link.springer.com/10.1007/s11277-016-3320-y>.
- [50] Sowjanya G, Nagaraju S. Design and implementation of door access control and security system based on IOT. In: 2016 International Conference on Inventive Computation Technologies (ICICT). vol. 2. IEEE; 2016. p. 1–4. Available from: <http://ieeexplore.ieee.org/document/7824850/>.
- [51] Hussein NA, Al Mansoori I. Smart Door System for Home Security Using Raspberry pi3. In: 2017 International Conference on Computer and Applications, ICCA 2017. IEEE; 2017. p. 395–399. Available from: <http://ieeexplore.ieee.org/document/8079785/>.
- [52] Ibrahim A, Paravath A, Aswin PK, Iqbal SM, Abdulla SU. GSM based digital door lock security system. In: Proceedings of 2015 IEEE International Conference on Power, Instrumentation, Control and Computing, PICC 2015. IEEE; 2016. p. 1–6. Available from: <http://ieeexplore.ieee.org/document/7455796/>.
- [53] Tomastik R, Lin Y, Banaszuk A. Video-based estimation of building occupancy during emergency egress. In: Proceedings of the American Control Conference. IEEE; 2008. p. 894–901. Available from: <http://ieeexplore.ieee.org/document/4586606/>.
- [54] Melfi R, Rosenblum B, Nordman B, Christensen K. Measuring building occupancy using existing network infrastructure. In: 2011 International Green Computing Conference and Workshops, IGCC 2011. IEEE; 2011. p. 1–8. Available from: <http://ieeexplore.ieee.org/document/6008560/>.
- [55] Li N, Calis G, Becerik-Gerber B. Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations. *Automation in Construction*. 2012 jul;24:89–99. Available from: <https://www.sciencedirect.com/science/article/pii/S0926580512000283?via=IISD>.
- [56] Chil Prakash V, Prakash AK, Arote U, Munigala V, Ramamritham K. Demo Abstract. In: Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems - e-Energy '15. New York, New York, USA: ACM Press; 2015. p. 191–192. Available from: <http://delivery.acm.org/10>.

- 1145/2780000/2770938/p191-chil-prakash.pdf?ip=193.137.168.32&id=2770938&acc=ACTIVESERVICE&key=2E5699D25B4FE09E.861C198C983DE13B.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=833710702&CFTOKEN=82872484&{}_{}_acm{}_{}_=1511651629{}_287f6ce4763987http://dl.acm.org/citation.cfm?doid=2768510.2770938.
- [57] Agarwal Y, Balaji B, Gupta R, Lyles J, Wei M, Weng T. Occupancy-driven energy management for smart building automation. In: Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10. New York, New York, USA: ACM Press; 2010. p. 1. Available from: <http://portal.acm.org/citation.cfm?doid=1878431.1878433>.
- [58] Yang Z, Li N, Becerik-Gerber B, Orosz M. A systematic approach to occupancy modeling in ambient sensor-rich buildings. SIMULATION. 2014 aug;90(8):960–977. Available from: <http://journals.sagepub.com/doi/10.1177/0037549713489918>.
- [59] Candanedo LM, Feldheim V. Accurate occupancy detection of an office room from light, temperature, humidity and CO₂ measurements using statistical learning models. Energy and Buildings. 2016 jan;112:28–39. Available from: <http://dx.doi.org/10.1016/j.enbuild.2015.11.071><http://linkinghub.elsevier.com/retrieve/pii/S0378778815304357>.
- [60] Ekwevugbe T, Brown N, Pakka V, Fan D. Real-time building occupancy sensing using neural-network based sensor network. In: 2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST). IEEE; 2013. p. 114–119. Available from: <http://ieeexplore.ieee.org/document/6611339/>.
- [61] Doukas H, Patlitzianas KD, Iatropoulos K, Psarras J. Intelligent building energy management system using rule sets. In: Building and Environment. vol. 42. Eleventh International IBPSA Conference; 2009. p. 3562–3569. Available from: <http://linkinghub.elsevier.com/retrieve/pii/S036013230600312X>.
- [62] Marques CAP, Rodrigues LFG. Sistema de Controlo de Acessos Integrado e Online. [Acceded on: 2018-02-28]; 2007. Available from: https://worldwide.espacenet.com/publicationDetails/biblio?DB=EPDOC&II=0&ND=3&adjacent=true&locale=pt{}_pt&FT=D&date=20090420&CC=PT&NR=103864A&KC=Ahttp://servicosonline.inpi.pt/pesquisas/GetSintesePDF?nord=2196931<https://worldwide.espacenet.com/publicatio>.
- [63] TW-T16. [Acceded on: 2018-02-27]; Available from: https://www.triplow.pt/loja/index.php?route=product/product/download/?download{}_id=167&product{}_id=3521.
- [64] GSM RM-100. [Acceced on: 2018-02-27]; Available from: https://www.triplow.pt/loja/controlo-acessos/controlo-acesso-telemovei?product{}_id=3200.
- [65] Kaba Oracode | Chaves do Areeiro. [Acceced on: 2018-02-27]; Available from: <http://www.chavesareeiro.pt/Produtos/SistemasElectr{ó}nicosedeControlodeAcessos/Soluç{~}{o}esparaHotelaria/KabaOracode.aspx>.

- [66] Kaba Modelo E-790 | Chaves do Areeiro. [Acceded on: 2018-02-27]; Available from: <http://www.chavesareeiro.pt/Produtos/SistemasElectr{ó}nicosedeControlodeAcessos/Solu{ç}{~}{o}}esparaHotelaria/KabaModeloE-790.aspx>.
- [67] Sistema de Gestão Hoteleira. [Acceded on: 2018-05-08]; Available from: <https://admhotel.com/>.
- [68] Santos JP. Apontamentos de Informática Industrial, bases de dados; 2015.
- [69] Santos JP. Apontamentos teóricos, Informática Industrial 2015/2016- Protocolo HTTP; 2015. p. 87–96.
- [70] DOIT ESP32 DEVKIT V1 Board. [Acceded on: 2018-05-15]; Available from: <https://makeradvisor.com/tools/esp32-dev-board-wi-fi-bluetooth/>.
- [71] Conhecendo o ESP32. [Acceded on: 2018-05-16]; Available from: <https://jgamblog.wordpress.com/2018/02/05/conhecendo-o-esp32-introducao-01/{%}0Ahttps://jgamblog.wordpress.com/2018/02/05/conhecendo-o-esp32-introducao-01/>.
- [72] ESP 32S Placa de Desenvolvimento. [Acceded on: 2018-05-15]; Available from: <https://pt.aliexpress.com/item/ESP32-ESP-32-ESP32S-ESP-32S-CP2102-Wireless-WiFi-Bluetooth-Development-Board-Micro-USB-Dual-Core/32867696371.htmlhttps://pt.aliexpress.com/item/ESP32-ESP-32-ESP32S-ESP-32S-CP2102-Wireless-WiFi-Bluetooth-Devel>.
- [73] Curvello A. Apresentando o módulo ESP8266 . [Acceded on: 2018-05-17]; 2015. Available from: <https://www.embarcados.com.br/modulo-esp8266/>.
- [74] Fonseca P. Internet das Coisas. Computerworld. 2014; Novembro. Available from: <https://pt.aliexpress.com/item/NodeMcu-Lua-WIFI-Internet-of-Things-development-board-based-ESP8266-module/32448461056.html?spm=a2g03.search0104.3.43.48d13103Dij2ke{&}ws{ }ab{ }test=searchweb0{ }0,searchweb201602{ }1{ }10152{ }10151{ }10065{ }10344/{ }10068http://www.computer>.
- [75] Santos JP. Apontamentos teóricos TAC, I2C. 2011;p. 134–150.
- [76] Circuits I. pcf8574 datasheet; 2001.
- [77] Biblioteca Wire. [Acceded on: 2018-02-09]; Available from: <https://www.arduino.cc/en/Reference/Wire>.
- [78] Dallas. RTC 1302 Datasheet;
- [79] Santos JP. Apontamentos teóricos TAC-SPI; 2018.
- [80] Biblioteca virtuoabotixRTC. [Acceded on: 2018-02-10]; Available from: <https://github.com/msparks/arduino-ds1302>.

- [81] Olimex. MG811: CO2 Sensor. [Acceded on: 2018-04-20]; Available from: www.olimex.cl/pdf/C02b.pdf.
- [82] Sensor De Dióxido De Carbono MG811. [Acceced on: 2018-05-21]; Available from: <https://pt.aliexpress.com/item/MG811-C02-Carbon-Dioxide-Gas-Sensor-Module-Detector-with-Analog-Signal-Output/32465511639.html?spm=a2g03.search0104.3.63.7b9942b71v0QAv{&}ws{ }ab{ }test=searchweb0{ }0,searchweb201602{ }1{ }10152{ }10151>.
- [83] System AL. PIR motion sensor datasheet; 2009. Available from: <http://www.google.com/patents/US7579595>.
- [84] O que é API? Entenda de uma maneira simples. [Acceced on: 2018-05-29]; Available from: <http://blog.vertigo.com.br/o-que-e-api-entenda-de-uma-maneira-simples/>.
- [85] PayPal PHP SDK. [Acceced on: 2018-05-29]; Available from: <https://paypal.github.io/PayPal-PHP-SDK/>.
- [86] PayPal. PayPal Developer. [Acceced on: 2018-05-29]; 2018. Available from: <https://developer.paypal.com/docs/integration/direct/express-checkout/integration-jsv4/>.
- [87] An F, Compara M. Reconhecimento do sinal de voz; Available from: <http://radio.lx.it.pt/STG/STG0405acetatos4.pdf>.
- [88] Wifi IoT Relay Board VDC Based on ESP8266 - ElectroDragon. [Acceced on:2018-06-30]; Available from: <http://www.electrodragon.com/product/wifi-iot-relay-board-vdc-based-esp8266/>.

Apêndice A

Esquemas elétricos das placas PCB

Tal como mencionado anteriormente, houve a necessidade criar nesta dissertação duas placas de circuito impresso distintas. Esta decisão foi tomada por questões de segurança e porque para a monitorização do espaço era indispensável que os sensores utilizados se encontrassem no interior do mesmo, ao passo que o teclado alfanumérico teria de estar obrigatoriamente no interior do alojamento.

Sendo assim, as figuras A.1 e A.2 mostram os esquemas elétricos das duas placas desenvolvidas. A placa do exterior contempla os LEDs de sinalização, bem como o encaixe para o *IO expander* PCF8574. Em teoria, tanto os pinos para comunicação I2C do ESP32, como os do PCF8574 possuem resistências de *pull-up* nos pinos de dados e de relógio. Contudo, verificou-se no decorrer dos testes efetuados, que adicionando resistências internas de *pull-up* a eficiência da comunicação aumenta.

A ligação entre as placas é feita à custa de um cabo *ethernet*. Enquanto que na placa interior foi projetada de forma a que exista uma ficha RJ45 para ligação do cabo, na placa exterior optou-se pela utilização de outro tipo de fichas, por uma questão de compactação da placa, pois dessa forma seria mais fácil encapsula-la e ao teclado numa caixa apropriada.

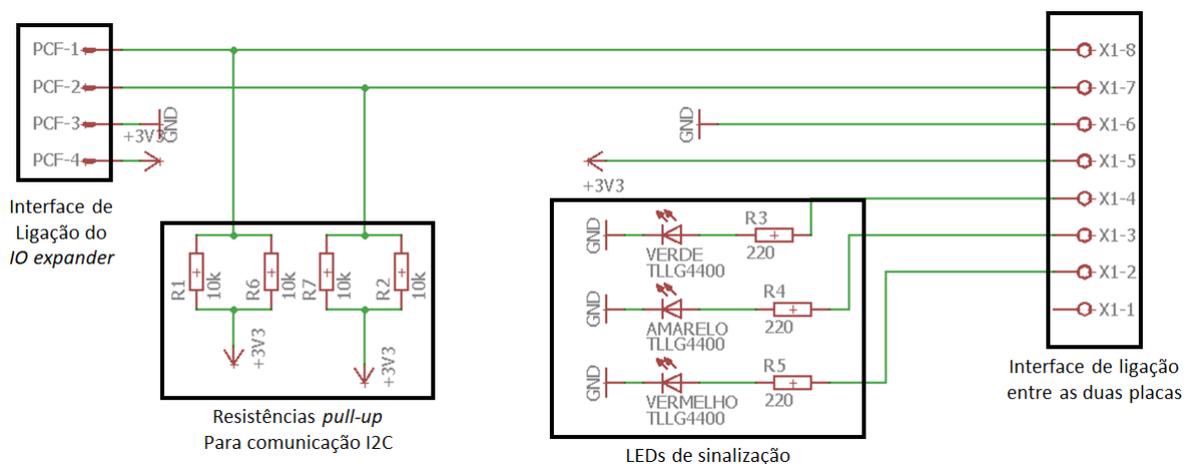


Figura A.1: Esquema elétrico da placa a ficar no exterior

A placa a ficar no interior do alojamento tem conectada a si toda a parte da alimentação do sistema e serve também de ligação de todos os sensores e outros periféricos.

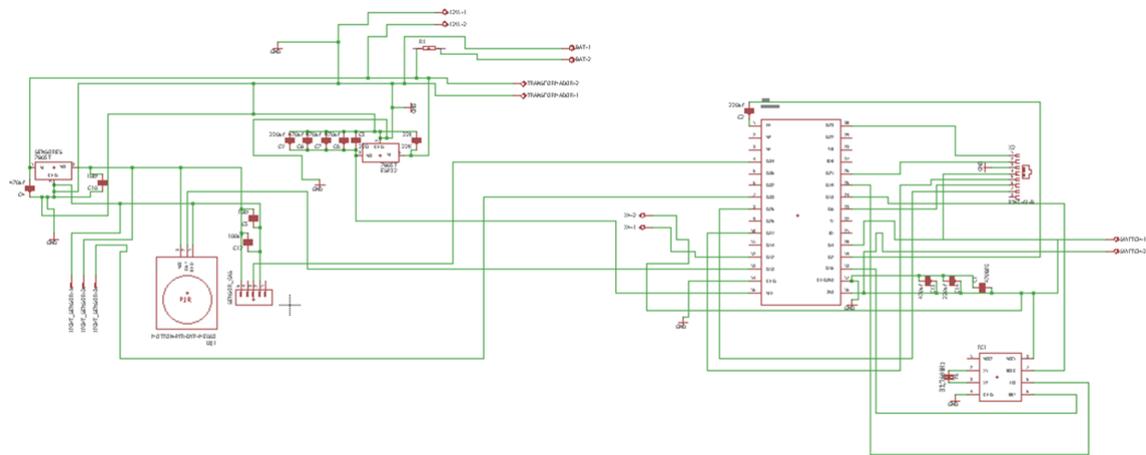


Figura A.2: Esquema elétrico da placa a ficar no interior

A figura A.3 mostra o esquema de alimentação proposto, a partir de um transformador de 12V, que é utilizado para alimentar o fecho elétrico. A alimentação dos sensores foi separada da do ESP32, na medida em que foram utilizados 2 reguladores de tensão LM7805. Dessa maneira a tensão de alimentação é mais estável e permite a utilização de apenas uma fonte de corrente (transformador). Foi ainda desenhada uma ficha, que serve para alimentação de uma bateria, pelo transformador. Assim caso haja uma falha de energia, todo o sistema pode continuar a ser alimentado pelo circuito UPS.

De notar apesar de não ser utilizado na versão final do projeto, foi projetado um encaixe para o sensor de CO_2 , pois aquando do desenho da placa previa-se a sua utilização, que veio a ser descartada por motivos citados anteriormente.

A figura A.4 mostra as fichas que servem para ligar o módulo relé opto-isolado à placa, atracando o mesmo a uma saída digital do ESP32. Foram projetados também encaixes para o RTC e para o ESP32, como seria de esperar.

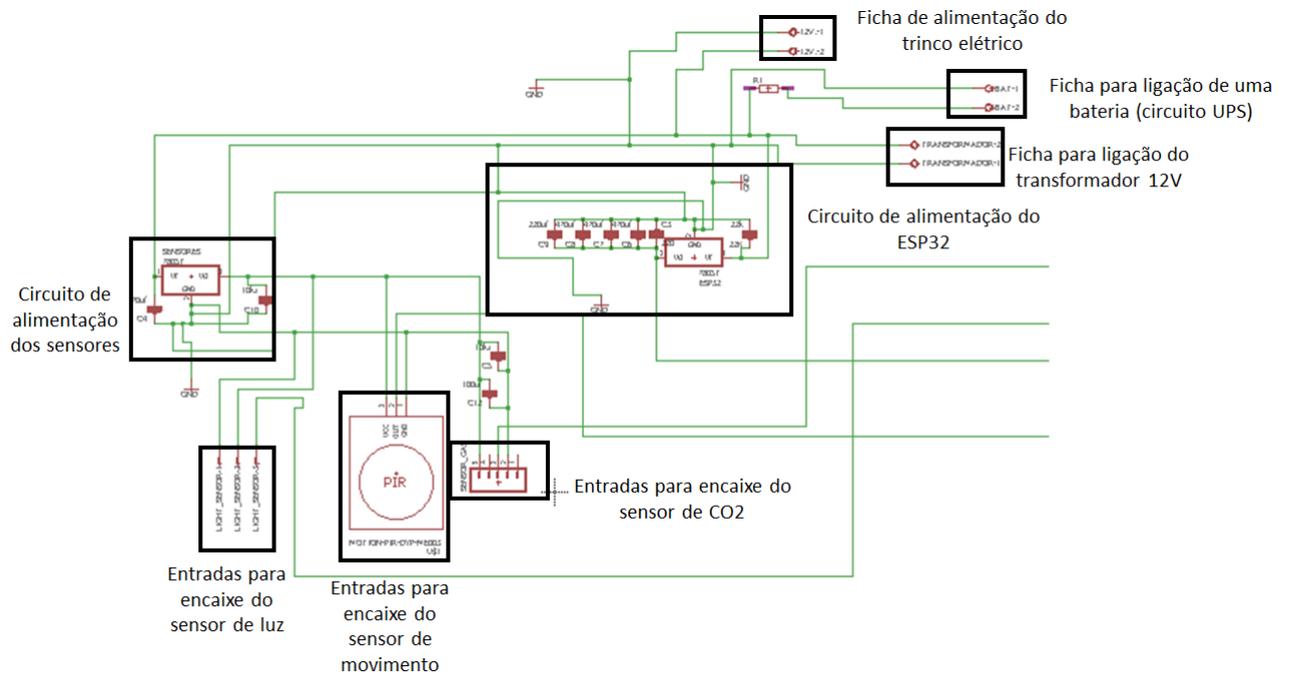


Figura A.3: Alimentação e sensores

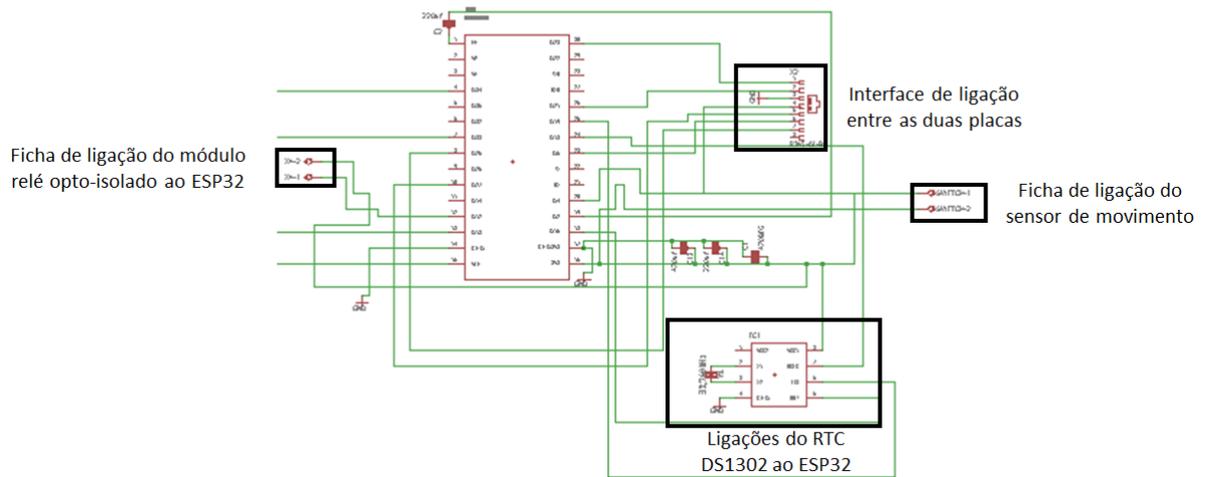


Figura A.4: Ligações entre ESP32 e restantes periféricos

Apêndice B

Páginas da interface *Web* criada

Nesta secção são expostas todas as páginas que compõem a interface *Web* criada. Recomenda-se a consulta de todas as imagens presentes, aquando da leitura do capítulo 4.3.3, para uma boa compreensão do documento.

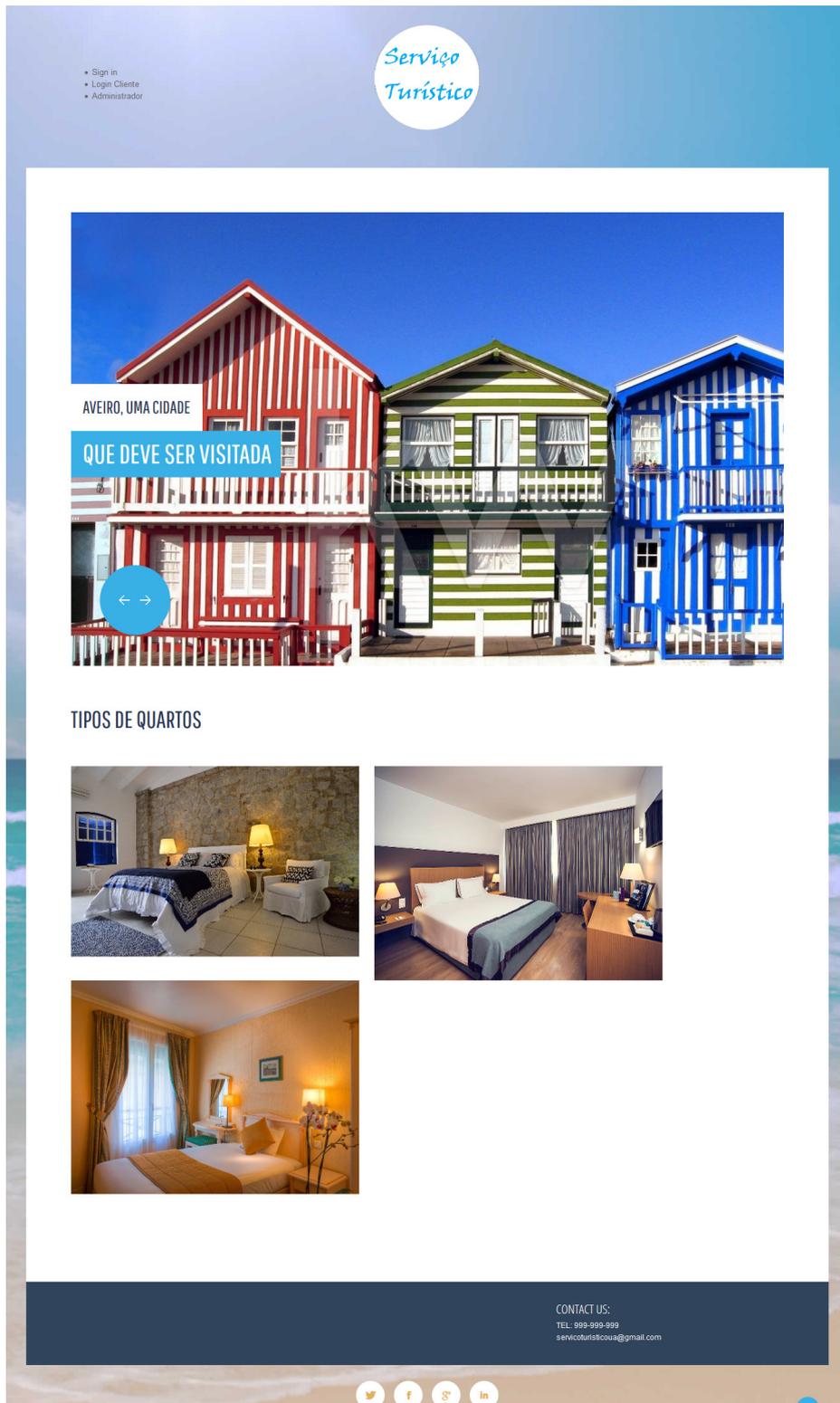
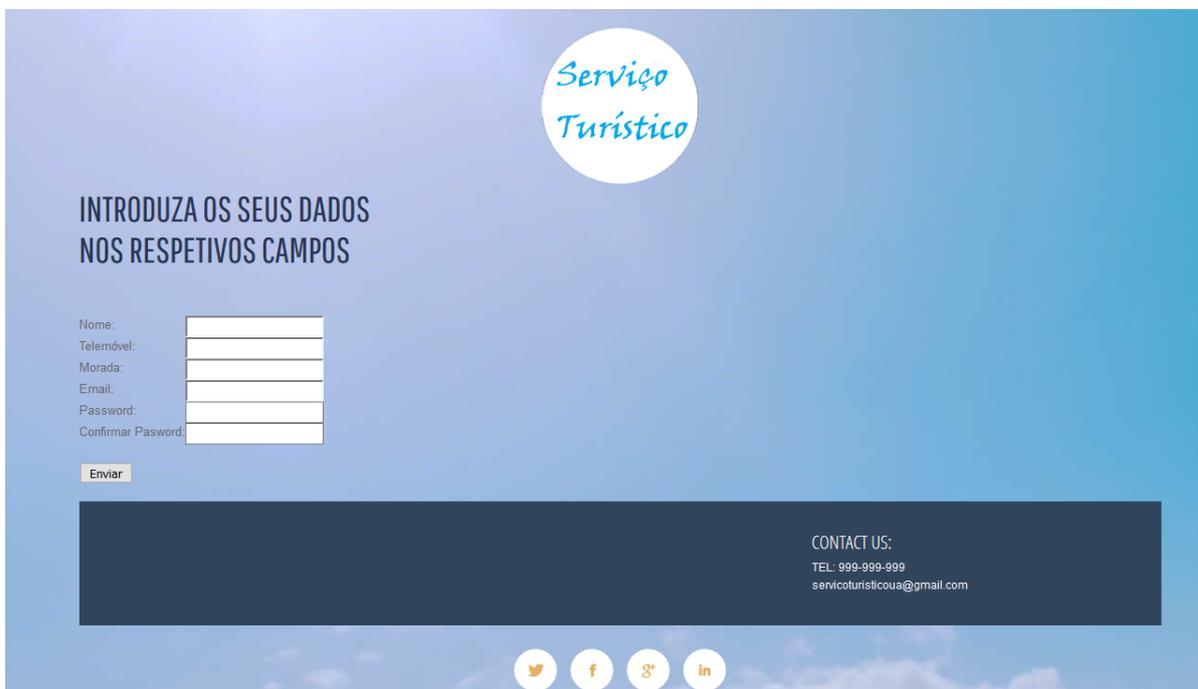
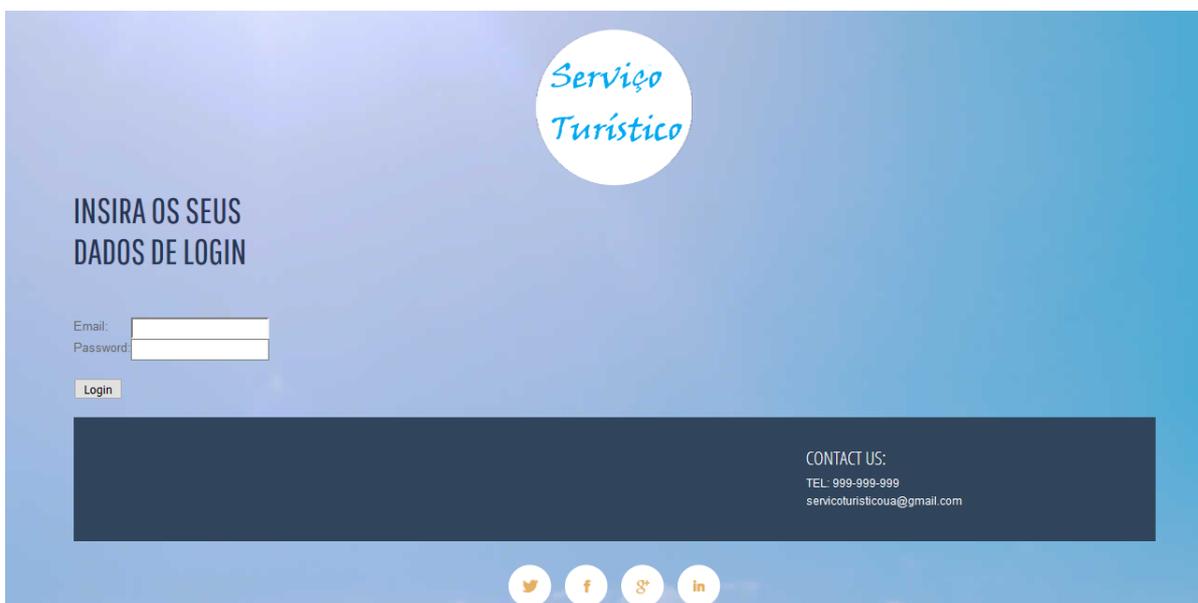


Figura B.1: Página inicial



The registration page features a blue gradient background with a white circular logo at the top center containing the text "Serviço Turístico". Below the logo, the heading "INTRODUZA OS SEUS DADOS NOS RESPETIVOS CAMPOS" is displayed. The registration form includes five input fields: "Nome:", "Telemóvel:", "Morada:", "Email:", and "Password:", followed by a "Confirmar Password:" field. A "Enviar" button is positioned below the fields. At the bottom right, a dark blue footer contains the text "CONTACT US:", "TEL: 999-999-999", and "servicoturisticoua@gmail.com". Social media icons for Twitter, Facebook, Google+, and LinkedIn are located at the bottom center.

Figura B.2: Página de registo no sistema



The client login page features a blue gradient background with a white circular logo at the top center containing the text "Serviço Turístico". Below the logo, the heading "INSIRA OS SEUS DADOS DE LOGIN" is displayed. The login form includes two input fields: "Email:" and "Password:". A "Login" button is positioned below the fields. At the bottom right, a dark blue footer contains the text "CONTACT US:", "TEL: 999-999-999", and "servicoturisticoua@gmail.com". Social media icons for Twitter, Facebook, Google+, and LinkedIn are located at the bottom center.

Figura B.3: Página de *login* do cliente

The screenshot displays a web interface for 'Serviço Turístico'. At the top center, there is a circular logo with the text 'Serviço Turístico'. Below the logo, the page is divided into two main sections: 'QUARTOS' (Rooms) and 'BUSCA DE QUARTO' (Room Search).

QUARTOS

- Quarto individual**: DESDE 40 € ATÉ 80 € (Image of a single bed room)
- Quarto duplo**: DESDE 100€ ATÉ 160€ (Image of a double bed room)
- Quarto duplo**: DESDE 100€ ATÉ 160€ (Image of a double bed room with a balcony)
- Suite**: DESDE 170€ ATÉ 250€ (Image of a suite with a large bed and view)

BUSCA DE QUARTO

Form fields for searching:

- Tipo de quarto: Individual (dropdown)
- Número de ocupantes: 1 (dropdown)
- Preço Máximo: Sem limite (dropdown)
- Data de check in: mm / dd / yyyy (text input)
- Data de check out: mm / dd / yyyy (text input)
- Procurar (button)

At the bottom right, contact information is provided:

CONTACT US:
TEL: 999-999-999
servicoturisticoua@gmail.com

At the bottom center, there are social media icons for Twitter, Facebook, Google+, and LinkedIn. A blue circular arrow icon is located at the bottom right corner of the page.

Figura B.4: Página de busca de quarto

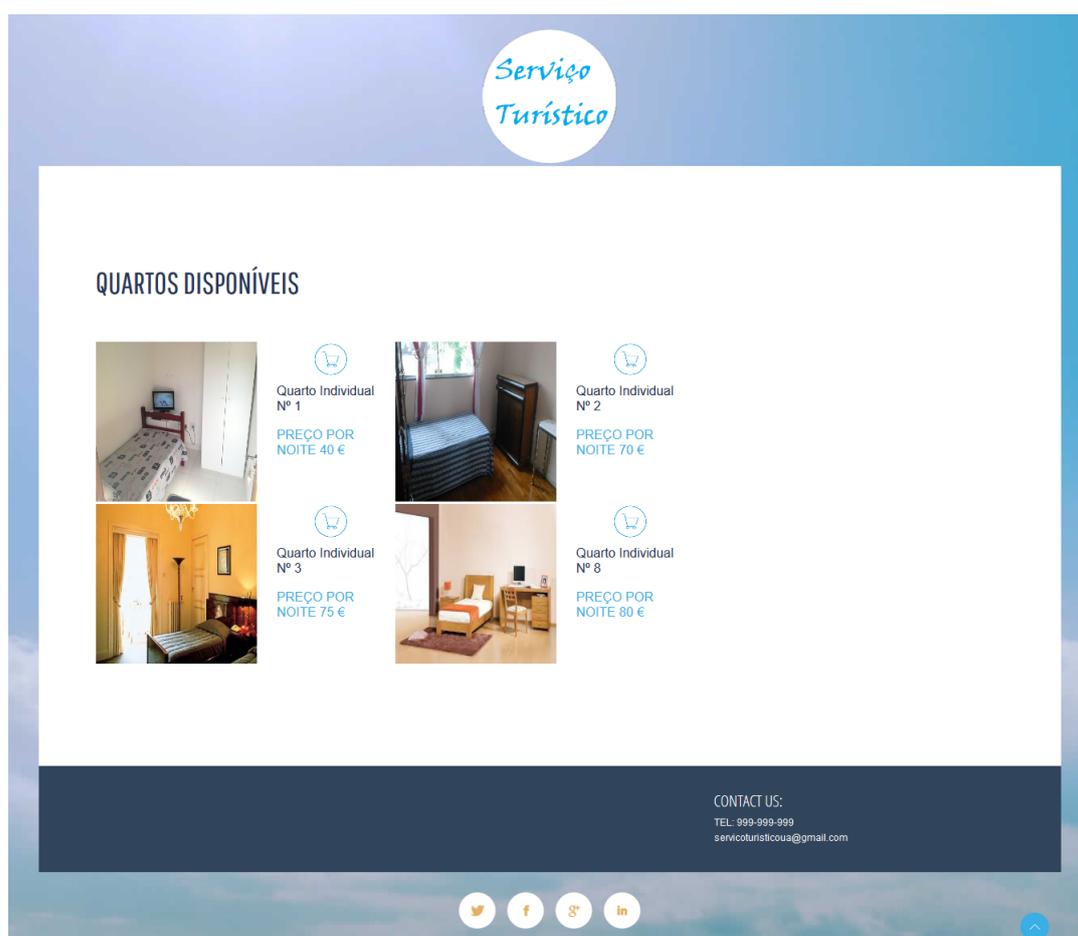
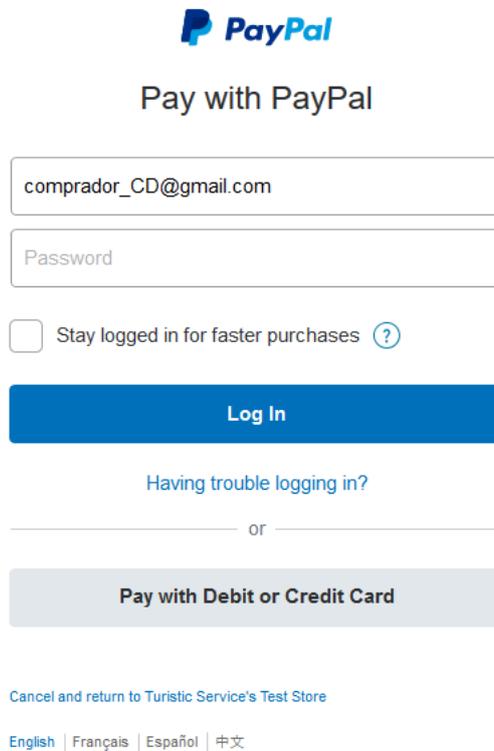
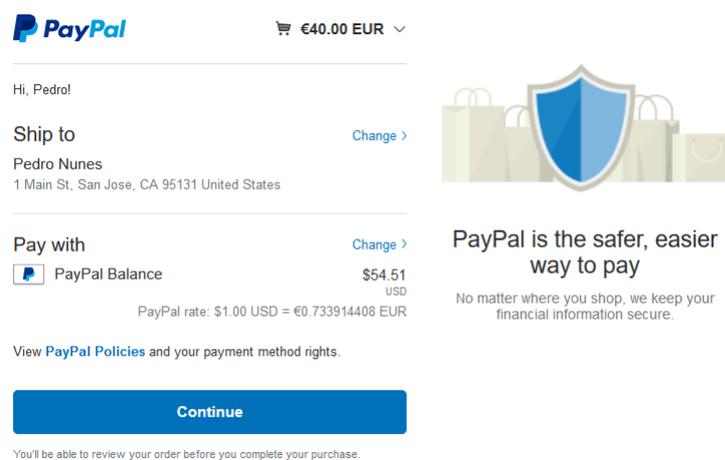


Figura B.5: Página de quartos disponíveis



The image shows the PayPal login interface. At the top center is the PayPal logo. Below it is the heading "Pay with PayPal". There are two input fields: the first contains the email address "comprador_CD@gmail.com" and the second is labeled "Password". Below the password field is a checkbox labeled "Stay logged in for faster purchases" with a question mark icon. A blue "Log In" button is positioned below the checkbox. Underneath the button is a link "Having trouble logging in?". A horizontal line with the word "or" in the center separates this section from a grey button labeled "Pay with Debit or Credit Card". At the bottom of the page, there is a link "Cancel and return to Turistic Service's Test Store" and a language selection menu with options: "English", "Français", "Español", and "中文".

Figura B.6: Página para pagamento com *PayPal*

The image shows the PayPal payment confirmation page. At the top left is the PayPal logo. To its right is a shopping cart icon with the text "€40.00 EUR" and a dropdown arrow. Below the logo, the text "Hi, Pedro!" is displayed. The "Ship to" section shows the name "Pedro Nunes" and the address "1 Main St, San Jose, CA 95131 United States", with a "Change >" link. The "Pay with" section shows "PayPal Balance" with a "Change >" link, the amount "\$54.51 USD", and the conversion rate "PayPal rate: \$1.00 USD = €0.733914408 EUR". Below this is a link "View PayPal Policies and your payment method rights." and a blue "Continue" button. To the right of the payment details is an illustration of shopping bags and a shield, with the text "PayPal is the safer, easier way to pay" and "No matter where you shop, we keep your financial information secure." At the bottom, a small note states "You'll be able to review your order before you complete your purchase."

Figura B.7: Página de confirmação de pagamento *PayPal*

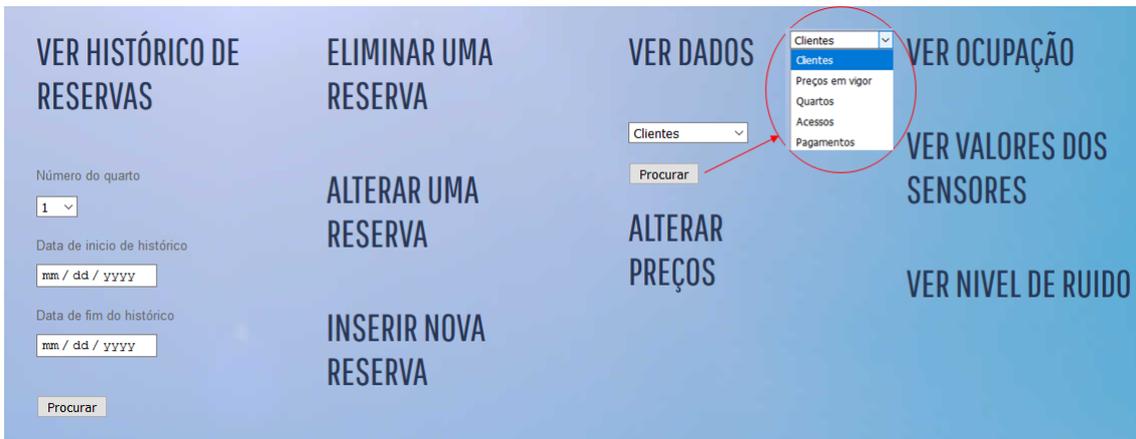


Figura B.8: Página de administrador

HISTORICO DE RESERVAS QUARTO 1

Check-in	Check-out	Cliente	Password
2018-03-01	2018-03-03	1	1234A
2018-03-01	2018-03-16	11	111111
2018-03-01	2018-03-17	9	999999
2018-03-05	2018-03-06	1	45454
2018-04-01	2018-04-30	1	12345
2018-05-18	2018-05-19	4	12345
2018-05-19	2018-05-20	1	5D854
2018-05-21	2018-05-22	1	6A53C
2018-05-22	2018-05-23	1	11111
2018-05-28	2018-05-29	1	A81BD
2018-05-29	2018-05-30	1	1553A
2018-05-30	2018-06-01	1	11111
2018-06-29	2018-06-30	1	**DC2

Figura B.9: Página de monitorização do histórico de reservas

RESERVA A ALTERAR NOVOS DADOS DA RESERVA

Número do quarto:

Check_in:

Check_out:

cliente (inserir ID_cliente):

password (inserir nova password):

Número do novo quarto:

Nova data de Check_in:

Nova data de Check_out:

Figura B.10: Página para alteração de uma reserva



RESERVA A ELIMINAR

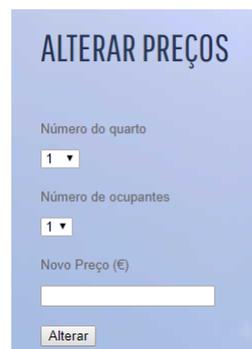
Número do quarto
1 ▾

Check_in
mm / dd / yyyy

Check_out
mm / dd / yyyy

Eliminar

Figura B.11: Página para a eliminação de uma reserva



ALTERAR PREÇOS

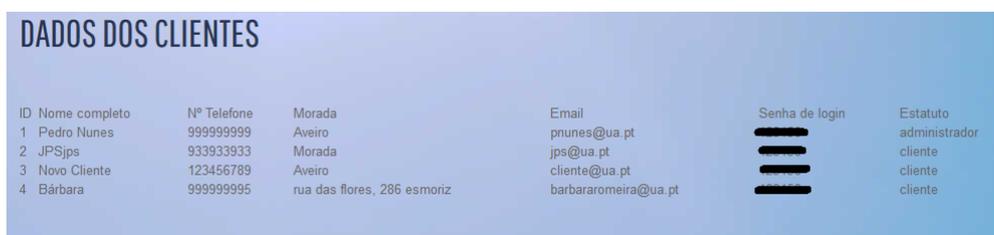
Número do quarto
1 ▾

Número de ocupantes
1 ▾

Novo Preço (€)

Alterar

Figura B.12: Página para a alteração de preços por noite



ID	Nome completo	Nº Telefone	Morada	Email	Senha de login	Estatuto
1	Pedro Nunes	999999999	Aveiro	pnunes@ua.pt	██████████	administrador
2	JPSjps	933933933	Morada	jps@ua.pt	██████████	cliente
3	Novo Cliente	123456789	Aveiro	cliente@ua.pt	██████████	cliente
4	Bárbara	999999995	rua das flores, 286 esmoriz	barbararomeira@ua.pt	██████████	cliente

Figura B.13: Página de monitorização dos dados dos clientes

DADOS DOS QUARTOS

Número do quarto	Ocupantes	Preço (€)
1	1	40
2	1	70
3	1	75
4	1	100
4	2	150
5	1	110
5	2	160
6	1	100
6	2	150
7	1	100
7	2	150
8	1	80
9	1	100
9	2	150
10	1	170
10	2	200
10	3	250

DADOS DOS QUARTOS

Número do quarto	Tipologia de quarto	Password Default
1	Individual	123A*
2	Individual	
3	Individual	
4	Duplo	11111
5	Duplo	11111
6	Duplo	
7	Duplo	11111
8	Individual	11111
9	Duplo	11111
10	Suite	11111

Figura B.14: Páginas de monitorização dos dados dos quartos

HISTÓRICO DE ACESSOS

Quarto	Data/hora	Acesso
1	2018-06-03 12:37:43	Acesso_negado
1	2018-06-03 13:16:30	Acesso_negado
1	2018-06-03 13:24:42	Acesso_cliente
1	2018-06-03 13:24:47	Acesso_cliente
1	2018-06-03 13:24:53	Acesso_cliente
1	2018-06-03 13:32:35	Acesso_cliente
1	2018-06-03 13:35:59	Acesso_cliente
1	2018-06-03 13:36:44	Acesso_default
1	2018-06-03 13:54:03	Acesso_negado

Figura B.15: Página de monitorização do histórico de acessos

HISTÓRICO DE PAGAMENTOS

ID pagamento	ID Cliente	ID pagador	Montante	Estado	Data de pagamento
PAY-3B823377A4253344WLMBO5SY	1	5UBK5FTW7B5PW	40	approved	2018-05-21
PAY-42K14413CH5326936LL6GGXI	1	5UBK5FTW7B5PW	55	approved	2018-05-16
PAY-4A179433GX6425515LMIWUDQ	1	5UBK5FTW7B5PW	40	approved	2018-06-01
PAY-4EG050511M1768143LL7QGFI	1	5UBK5FTW7B5PW	40	approved	2018-05-18
PAY-56556398WV6758840LMGCH5A	1	5UBK5FTW7B5PW	40	approved	2018-05-28
PAY-6W4743086E218561TLL6UQOI	2	5UBK5FTW7B5PW	55	approved	2018-05-17
PAY-8R140083GY3330131LMGVVDGY	1	5UBK5FTW7B5PW	40	approved	2018-05-29
PAY-9JF66835CS799851BLL7N4CA	4	5UBK5FTW7B5PW	40	approved	2018-05-18
PAY-9MF63397XN055190CLL6GNRQ	2	5UBK5FTW7B5PW	55	approved	2018-05-16

Figura B.16: Página de monitorização do histórico de pagamentos



Figura B.17: Página de monitorização dos sensores do módulo de controlo de acessos

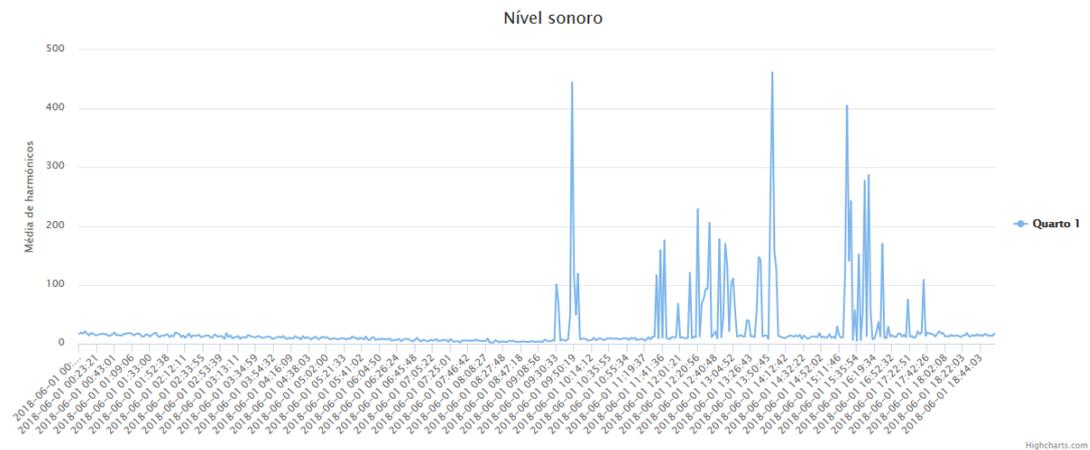


Figura B.18: Página de monitorização do sensor de som

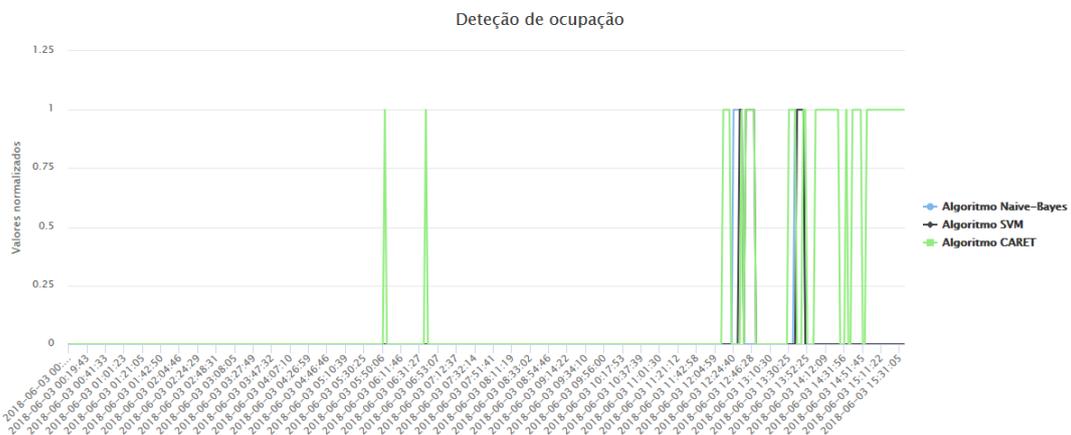


Figura B.19: Página de deteção de presenças

Apêndice C

Criação de um ambiente de testes *Sandbox*

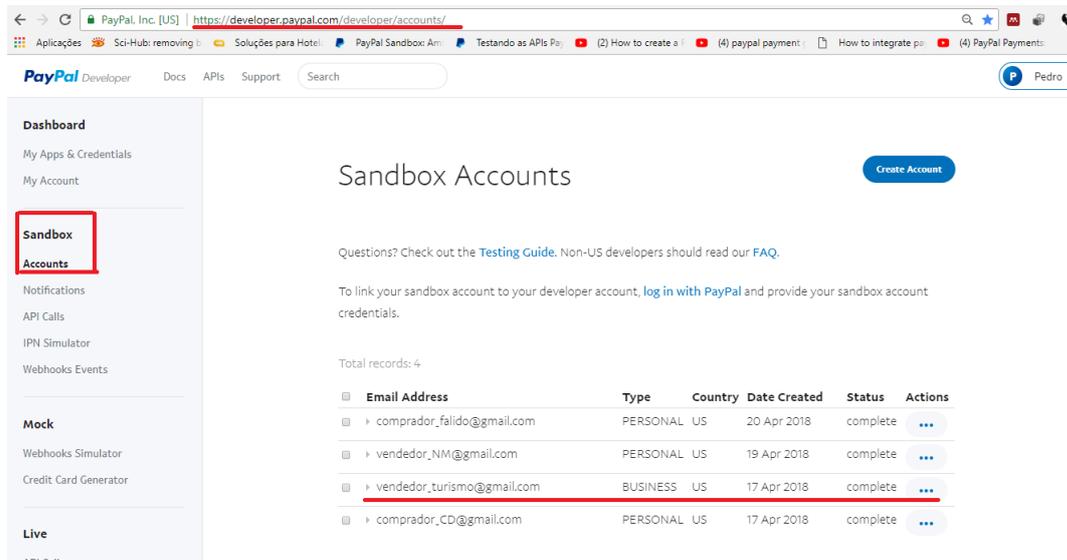
Para criar um ambiente controlado onde se possa simular uma transferência com a ferramenta *PayPal Sandbox* foi necessário seguir um conjunto de procedimentos que irão ser expostos em seguida.

A primeira coisa a ser feita é a criação de uma conta *PayPal* comum. Para esta dissertação foi criada uma conta com o mesmo email que foi utilizado para enviar mensagens automáticas aos clientes. Seguidamente deverá ser feito login na plataforma *developer* do *PayPal*. Este site foi criado especialmente para programadores e aí é possível criar contas *Sandbox* e gerar credenciais para comunicar com as APIs do *PayPal*.

Uma vez na plataforma de desenvolvimento é necessário criar várias contas *Sandbox*, no separador *Sandbox-accounts*. Para simular uma transferência são necessárias pelo menos duas contas. Uma de comprador e outra de vendedor. No entanto convém criar várias contas de comprador diferentes, de modo a garantir que todas as diferentes situações são testadas. A figura C.1 mostra as contas criadas para testes. Uma delas é uma conta de comerciante e é para onde o montante da transferência será efetuado. As restantes dizem respeito a diferentes tipos de compradores, um comprador com saldo na conta *PayPal*, outro sem saldo na conta *PayPal*, mas com conta bancária associada de onde poderá sair o montante da transferência e um outro cliente, sem qualquer método de pagamento possível (sem saldo e sem conta bancária associada).

Após a criação das contas de teste, é necessária a criação de uma aplicação. Esta irá gerar credenciais que serão utilizadas no código PHP para comunicar com a API. Estas identificam a identidade à qual irá ser feito o pagamento, pelo que para criar uma aplicação no separador *My Apps & Credentials* é necessário criar de antemão uma conta de vendedor. A figura C.2 ilustra a criação de uma aplicação. Na figura C.3 estão representadas as credenciais de identificação que permitem a utilização de APIs no processo de comunicação com o *PayPal*.

O ambiente de testes fica desta forma pronto, bastando em seguida criar um código em PHP que comunique com a API do *PayPal*. A primeira coisa a fazer no código, é incluir os ficheiros do kit de desenvolvimento *PayPal SDK* para PHP e identificar a identidade comerciante através das credenciais geradas na aplicação criada na plataforma *developers*, com mostra a figura C.4. Em seguida é necessário definir uma série de detalhes, como o nome do produto que se irá comercializar, preço, quantidades, entre outros, como mostra

Figura C.1: Contas *Sandbox* Criadas

a figura C.5. Por fim é necessário definir um URI de redirecionamento, após o pagamento ser efetuado (figura C.6). Essa página será responsável por recolher dados vindos da API, tais como o ID da pessoa que efetuou o pagamento, ID da compra, entre outros. Será também esta página a responsável por transformar a transação em pagamento, transferindo o montante respectivo da conta do cliente para a conta do comerciante (figura C.7).

Em qualquer momento é possível fazer *login* no site *Paypal Sandbox* e confirmar que as transferências foram realizadas. A figura C.8 mostra o comprovativo que é recebido na conta do vendedor e a figura C.9 mostra o comprovativo recebido na conta do cliente.

A integração do *PayPal* foi feita num ambiente de testes, contudo para passar desse ambiente para um método de pagamento real, basta gerar uma aplicação *PayPal* ao invés de uma aplicação *Sandbox*. Para tal é necessário criar previamente uma conta *PayPal* de vendedor. No que diz respeito ao código PHP, bastará modificar as credenciais de identificação para as da nova aplicação e em vez de ser direcionada para o *Sandbox*, o cliente será redirecionado para o *PayPal*.

The screenshot shows the PayPal Developer 'Create App' form. On the left is a navigation sidebar with sections: **Dashboard** (My Apps & Credentials, My Account), **Sandbox** (Accounts, Notifications, API Calls, IPN Simulator, Webhooks Events), and **Mock** (Webhooks Simulator, Credit Card Generator). The main content area is titled 'Application Details' and contains an 'App Name' input field, a 'Sandbox developer account' dropdown menu (set to 'vendedor_turismo@gmail.com (US)'), a reminder text, and a 'Create App' button.

Figura C.2: Criação de uma aplicação

The screenshot shows the 'teste_paypal' application details page. The app name is 'teste_paypal' and it is currently in 'Sandbox' mode. The 'App display name' is 'teste_paypal'. A 'SANDBOX API CREDENTIALS' section is expanded, showing the 'Sandbox account' as 'vendedor_turismo@gmail.com', the 'Client ID' as 'ARAbD6ipoKC-qVKit1uGtdG6lfrWLOkkKnmrEEyMscO7HWI72R25Nt9ivGllLpchaxzWaBLloQlhXD', and the 'Secret' as 'Show'.

Figura C.3: Credenciais geradas para comunicar com a API

```

<?php
use PayPal\Api\Payer;
use PayPal\Api\Item;
use PayPal\Api\ItemList;
use PayPal\Api\Details;
use PayPal\Api\Amount;
use PayPal\Api\Transaction;
use PayPal\Api\RedirectUrls;
use PayPal\Api\Payment;

require 'vendor/autoload.php';

define('SITE_URL', 'http://localhost');

$paypal = new \PayPal\Rest\ApiContext (
new \PayPal\Auth\OAuthTokenCredential (
"ARAbD6ipoKC-qVKjtluGtdG6lfrWlLokkKnmrEEyMsc07HWI72R2Snt9jvG1ILpcghaxzWaBLJoQJhXD"
,
"E1bItWzTsFdl8mfOpjTjY2UuSVb8Myq-FiNe2wkVC6nvoOWXqdkB_5NuytLTXR2qmvWEgJw1xxfiPpYd"
)
);

```

Figura C.4: Definição do comerciante

```

32 $product='Aluguer de Quarto';
33 $price=$_POST['price'];
34 $quantidade=$_POST['quantity'];
35 $shipping=0.00;
36 $total=$price*$quantidade+$shipping;
37 $payer= new Payer ();
38 $payer->setPaymentMethod('paypal');
39 $item= new Item ();
40 $item->setName ($product)
41     ->setCurrency ('EUR')
42     ->setQuantity ($quantidade)
43     ->setPrice ($price);
44 $itemList=new ItemList ();
45 $itemList->setItems ([$item]);
46 $details= new Details ();
47 $details->setShipping ($shipping)
48     ->setSubtotal ($price*$quantidade);
49 $amount= new Amount ();
50 $amount->setCurrency ('EUR')
51     ->setTotal ($total)
52     ->setDetails ($details);
53 $transaction=new Transaction ();
54 $transaction->setAmount ($amount)
55     ->setItemList ($itemList)
56     ->setDescription ('Pagamento de serviço de aluguer de quartos')
57     ->setInvoiceNumber (uniqid());

```

Figura C.5: Definição dos detalhes do produto

```

$redirectUrls=new RedirectUrls();
$redirectUrls->setReturnUrl(SITE_URL . '/sitehotel/pay.php?success=true
    ->setCancelUrl(SITE_URL . 'sitehotel/pay.php?success=false');
$payment= new Payment();
$payment->setIntent('sale');
$payment->setPayer($payer);
$payment->setRedirectUrls($redirectUrls);
$payment->setTransactions([$transaction]);
try{
    $payment->create($paypal);
} catch (Exception $e) {
    die($e);
}

$approvalUrl=$payment->getApprovalLink();
echo $approvalUrl;
header("Location: {$approvalUrl}");

```

Figura C.6: Definição das páginas de redirecionamento

```

$paymentId=$_GET['paymentId'];
$payerId=$_GET['PayerID'];
$montante=$_GET['montante'];
$payment=Payment::get($paymentId,$paypal);

$execute= new PaymentExecution();
$execute->setPayerId($payerId);

```

Figura C.7: Página de redirecionamento onde será efetuado o pagamento

Order details	Quantity	Price	Subtotal
Aluguer de Quarto	1	€40.00 EUR	€40.00 EUR
Purchase Total			€40.00 EUR
Payment details			
Purchase Total	€40.00 EUR		
Sales Tax	€0.00 EUR		
Shipping Amount	€0.00 EUR		
Handling Amount	€0.00 EUR		
Insurance Amount	€0.00 EUR		
Gross Amount	€40.00 EUR		
PayPal Fee	-€1.51 EUR		
Net Amount	€38.49 EUR		

[Issue a refund](#)

Figura C.8: Confirmação de pagamento por parte do vendedor

Completed >

MAY 29	Turistic Service's Test Store Payment	- €40.00 EUR
-----------	--	-----------------

Paid with PayPal balance	Seller info Turistic Service's Test Store vendedor_turismo@gmail.com
Exchange rate \$54.51 USD = €40.00 EUR 1 USD = 0,73 EUR	Invoice ID 5b0d519742817
Ship to Pedro Nunes 1 Main St San Jose, CA 95131 United States	Purchase details Aluguer de Quarto €40.00 EUR
Transaction ID 4j533156AN839464L	Total €40.00 EUR

Figura C.9: Confirmação de pagamento por parte do comprador