

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Desenvolvimento de Sistema de Aquisição e Interface IoT para Sensor Integrado de Presença e Humidade em Cama Médica

Ricardo José Nunes Organista

Mestrado em Engenharia Eletrotécnica e de Computadores

Orientador: Miguel Fernando Paiva Velhote Correia

29 de março de 2024

Resumo

Num mundo em que a tecnologia desempenha um papel cada vez mais crucial, a sua integração nos cuidados de saúde torna-se imperativa. Atualmente na realidade portuguesa, os hospitais públicos estão constantemente sobrelotados, sendo necessário um esforço considerável por parte da equipa médica para chegar a todos os doentes. A implementação de sistemas inteligentes pode otimizar o trabalho dos profissionais de saúde, proporcionando cuidados mais eficientes aos pacientes e aliviando o excesso de trabalho sobre as equipas médicas.

A área dos têxteis inteligentes apresenta um vasto potencial a explorar. Com os avanços na eletrónica, especialmente na conceção de componentes pequenos e impercetíveis, essa realidade certamente se integrará cada vez mais na vida das pessoas. No ambiente hospitalar, destaca-se a sua aplicação na monitorização dos sinais vitais dos pacientes.

O estudo foi proposto pela Invacare, líder mundial na produção e distribuição de produtos médicos inovadores. Consistiu na caracterização do funcionamento de um sensor sob a forma de um lençol. Este sensor foi projetado para identificar a presença e ausência de pessoas, além de monitorizar os níveis de humidade, que quando diferentes de zero indicam uma possível libertação de líquidos pelos pacientes, como sangue, urina, entre outros. O objetivo foi integrar o lençol num ecossistema de soluções Invacare, utilizando a Internet das Coisas para permitir a comunicação em tempo real entre o lençol e outros dispositivos. A perspetiva futura, envolve a integração do sistema criado, com um conjunto mais vasto de sensores.

Após a caracterização, foi desenvolvido um circuito capaz de processar o sinal proveniente do sensor, utilizando um microcontrolador ESP32 para definir estados e enviar dados para uma plataforma em nuvem. Essa plataforma contém as informações a serem transmitidas aos cuidadores em tempo real.

De seguida, foi criada uma aplicação para smartphone que apresenta em tempo real o estado de presença e de humidade do lençol. O sistema alerta um cuidador caso o paciente esteja ausente da cama ou os níveis de humidade sejam superiores a 10%. Na fase final, foi projetada uma caixa para incorporar o circuito, representando o que pode vir a ser uma solução final no que diz respeito à estética e apresentação do produto. Esta solução foi concebida não apenas para hospitais, mas também para lares e ambientes domésticos, sendo aplicável em todas as situações que envolvem a monitorização de pessoas acamadas, uma vez que, o lençol não incomoda o utilizador e o circuito fica numa unidade externa ao sensor.

A validação do sistema desenvolvido, foi efetuada através da realização de testes que pretendiam simular situações reais, tais como um paciente a dormir, ou a ocorrência de perda de fluidos. Os testes foram realizados por um período significativo de tempo, de forma a ser o mais fiel possível da realidade. Os resultados obtidos destacaram a estabilidade e consistência do sistema, evidenciando a sua aplicabilidade no contexto dos cuidados de saúde.

O trabalho desenvolvido, pretende ser uma contribuição para a área dos têxteis inteligentes nos cuidados médicos, destacando-se pelo desenvolvimento de uma solução inovadora e versátil.

Palavras chave: Aplicação para Smartphone, Cuidadores, IoT, Monitorização, Sensor, Têxteis inteligentes

Abstract

In a world where technology plays an increasingly crucial role, its integration into healthcare becomes imperative. Currently, in the Portuguese reality, public hospitals are constantly overcrowded, requiring a considerable effort from the medical team to reach all patients. The implementation of intelligent systems can optimize the work of healthcare professionals, providing more efficient care to patients and alleviating the workload on medical teams.

The field of smart textiles presents a vast potential to be explored. With advancements in electronics, particularly in the design of small and imperceptible components, this reality will undoubtedly integrate more deeply into people's lives. In the hospital environment, its application in monitoring patients' vital signs stands out.

The study was proposed by Invacare, a global leader in the production and distribution of innovative medical products. It involved characterizing the operation of a sensor in the form of a bedsheet. This sensor was designed to identify the presence and absence of individuals, as well as monitor humidity levels, which, when different from zero, indicate a potential release of liquids by patients, such as blood, urine, among others. The goal was to integrate the bedsheet into an Invacare solutions ecosystem, using the Internet of Things to enable real-time communication between the bedsheet and other devices. The future perspective involves integrating the created system with a broader set of sensors.

Following the characterization, a circuit capable of processing signals from the sensor was developed, using an ESP32 microcontroller to define states and send data to a cloud platform. This platform contains information to be transmitted to caregivers in real-time.

Subsequently, a smartphone application was created, presenting real-time information on the bedsheet's presence and humidity status. The system alerts a caregiver if the patient is absent from bed or if humidity levels exceed 10%. In the final phase, a box was designed to incorporate the circuit, representing a potential final solution in terms of aesthetics and product presentation. This solution was conceived not only for hospitals but also for homes and domestic environments, applicable in all situations involving the monitoring of bedridden individuals, as the bedsheet does not disturb the user, and the circuit remains external to the sensor.

The validation of the developed system was carried out through tests simulating real situations, such as a patient sleeping or the occurrence of fluid loss. The tests were conducted over a significant period to be as faithful as possible to reality. The results highlighted the stability and consistency of the system, demonstrating its applicability in the context of healthcare.

The developed work aims to contribute to the field of smart textiles in healthcare, standing out for the development of an innovative and versatile solution.

Keywords: Caregivers, IoT, Monitoring, Sensor, Smart Textiles, Smartphone Application

Agradecimentos

Ao longo da realização do projeto documentado na presente dissertação, vivenciei uma série de desafios e conquistas. Sem dúvida, foi um processo de aprendizagem marcado por muitas ideias, algumas que levaram ao avanço e outras que fizeram perceber que o caminho não era por ali. A escrita deste documento representa o culminar de um longo ciclo de estudos, cujo percurso se torna muito complicado sem a contribuição de pessoas que, de diversas formas, colaboraram para a sua finalização.

Assim sendo, gostaria de expressar os meus sinceros agradecimentos:

Ao professor Miguel Velhote, o meu orientador académico, pela sua compreensão, e que com o seu conhecimento e experiência foi crucial no desenvolvimento do projeto proposto e na clarificação de aspetos científicos.

Ao professor Henrique Gonçalves, o meu coorientador, por me ter proposto o tema e introduzido ao mundo dos têxteis inteligentes e da engenharia nos cuidados de saúde. A sua disponibilidade constante, pronta resposta no esclarecimento das dúvidas que iam surgindo e ajuda na impressão 3D foram fundamentais.

Aos meus pais, que tornaram possível todo o meu percurso académico, pela força que me deram ao longo da minha vida, pela paciência e por sempre me apoiarem nas minhas decisões.

Por último, a todos os amigos e pessoas que contribuíram de forma direta ou indireta para o meu desenvolvimento pessoal e profissional, cruzando-se neste longo caminho académico que culminou neste documento de dissertação.

*“Empty your mind, be formless, shapeless, like water.
Now you put water into a cup, it becomes the cup.
You put water into a bottle, it becomes the bottle.
You put it in a teapot, it becomes the teapot.
Now, water can flow or it can crash.
Be water, my friend.”*

Bruce Lee

Conteúdo

1	Introdução	1
1.1	Enquadramento e motivação	1
1.2	Objetivos	2
1.3	Estrutura da Dissertação	3
2	Têxteis inteligentes	5
2.1	Definição	5
2.2	Funcionalidades de um têxtil inteligente	6
2.3	Sensores têxteis de pressão	7
2.4	Estudos existentes	8
2.5	Protótipos de têxteis inteligentes	12
3	Fundamentos	17
3.1	Lençol <i>TEXTIBLE Wisbi</i>	17
3.2	<i>Internet of Things</i>	19
3.2.1	Modelos de comunicação da <i>Internet of Things</i>	19
3.2.2	Desafios da <i>Internet of Things</i>	22
3.3	Cadeia de medição	23
3.3.1	Circuito Integrado 555	23
3.3.2	Condensador plano	27
3.4	Microcontroladores	29
3.4.1	<i>ESP-WROOM-32</i>	30
3.5	Wi-Fi	31
3.5.1	Modos Wi-Fi no ESP32	32
3.5.2	<i>Wi-Fi Manager</i> no ESP32	33
3.6	<i>Network Time Protocol</i>	34
3.7	Firebase	35
3.7.1	<i>Authentication</i>	35
3.7.2	<i>Cloud firestore</i>	35
3.8	Flutter	36
3.8.1	Linguagem de programação Dart	37
4	Arquitetura	39
4.1	Visão Global	39
4.2	Sensor <i>TEXTIBLE Wisbi</i>	41
4.2.1	Comportamento do sistema no estado seco	42
4.2.2	Comportamento do sistema no estado húmido	42
4.3	Circuito para aquisição do sinal	43

4.4	ESP32 - Processamento e tratamento dos dados	45
4.4.1	Determinação da capacitância	45
4.4.2	Definição dos estados	46
4.4.3	Fluxograma do funcionamento do sistema	48
4.5	<i>Firebase</i>	49
4.5.1	<i>Cloud Firestore</i>	50
4.5.2	Escrita de dados na <i>Cloud Firestore</i>	51
4.5.3	<i>Firebase Authentication</i>	53
5	Interface IoT	55
5.1	Visão Geral	55
5.2	Menu de Registo	57
5.2.1	Conexão à Internet	59
5.3	Calibração do sistema	60
5.4	Menu de login	63
5.5	<i>Dashboard</i>	64
5.6	Alertas	65
6	Apresentação de resultados	69
6.1	Deteção de presença e ausência	69
6.1.1	Deteção da desconexão do sistema	72
6.2	Deteção de humidade	73
7	Implementação e Desenvolvimento do Protótipo	75
7.1	Conector	76
7.2	Invólucro	76
7.3	Desenvolvimento e implementação da PCB	77
8	Conclusões e Trabalho Futuro	79
8.1	Conclusões	79
8.2	Trabalho Futuro	81
A	Interface de utilizador	83
A.1	Menus da aplicação	83
A.2	<i>Wi-Fi Manager</i>	84
B	Implementação e Desenvolvimento do Protótipo	85
B.1	Conector do lençol	85
B.2	Processo de criação do protótipo	85
	Referências	87

Lista de Figuras

2.1	Sensor têxtil capacitivo apresentado por Mayer <i>et al.</i>	7
2.2	<i>VitalJacket Baby</i>	12
2.3	Nike + iPod Sports Kit	12
2.4	ANGEL	13
2.5	Tecnologia <i>Stappone</i>	13
2.6	<i>NanoStim</i>	14
2.7	Almofada ergonómica PPS4	14
2.8	<i>MuscleBAN Strap</i>	14
3.1	Filamento incorporado no lençol <i>TEXIBLE Wisbi</i>	18
3.2	Modelo de comunicação: <i>Device-to-Cloud</i>	20
3.3	Modelo de comunicação: <i>Device-to-Gateway</i>	21
3.4	Diagrama de blocos geral de uma cadeia de medição	23
3.5	Diagrama de blocos do circuito integrado 555	23
3.6	Funcionamento dos comparadores	24
3.7	Terminais do circuito integrado 555	24
3.8	Modo de operação astável do CI 555	25
3.9	Condensador plano	27
3.10	Campo elétrico no dielétrico	28
3.11	Diagrama de blocos de um microcontrolador	29
3.12	Terminais da placa de desenvolvimento ESP-WROOM-32	31
3.13	ESP32: Modo <i>Access Point</i>	32
3.14	ESP32: Página inicial do Wi-Fi Manager	33
3.15	Arquitetura: Network Time Protocol	34
3.16	Estrutura dos dados na <i>Cloud Firestore</i>	36
3.17	Arquitetura geral	37
4.1	Arquitetura global do sistema	40
4.2	Filamentos incorporados no lençol - Perspetiva traseira	41
4.3	Circuito de aquisição do sinal	43
4.4	Função para a leitura da capacitância pelo ESP32	45
4.5	Diagrama de máquina de estados do lençol	47
4.6	Fluxograma geral do sistema desenvolvido	48
4.7	Arquitetura da base de dados do sistema desenvolvido [Fonte: Cloud Firestore]	50
4.8	Diagrama de atividades do processo de escrita de dados no <i>Firebase</i>	51
4.9	Fluxograma da função responsável por atualizar o estado de presença	52
4.10	Autenticação dos utilizadores [Fonte: Firebase Authentication]	53
5.1	Arquitetura geral da aplicação desenvolvida para <i>smartphone</i>	55

5.2	Diagrama de componentes da aplicação	56
5.3	Diagrama de caso de uso geral	57
5.4	Diagrama de atividades do processo de registo	58
5.5	Menu de registo	58
5.6	Diagrama de atividades do processo de ligação à rede Wi-Fi	59
5.7	Diagrama de caso de uso do <i>Wi-Fi Manager</i>	60
5.8	Menu de calibração da aplicação	61
5.9	Funções do ESP32 no processo de calibração	61
5.10	Diagrama de sequência do processo de calibração	62
5.11	Menu de login da aplicação	63
5.12	Diagrama de atividades do processo de login	63
5.13	Menu <i>Dashboard</i>	64
5.14	Diagrama de atividades - Menu <i>Dashboard</i>	64
5.15	<i>Dashboard</i> - Alerta de ausência	65
5.16	Diagrama de sequência do processo de envio de notificações	66
5.17	Diagrama de sequência: Mecanismo de alerta de sistema desligado	67
5.18	Diagrama de sequência: Desconexão do transmissor	68
5.19	Interface IoT para deteção de presença e humidade	68
6.1	Capacitância: Lençol seco e ausente	69
6.2	Capacitância: Lençol seco e presente	70
6.3	Capacitância: Lençol seco nos dois estados	71
6.4	Diferentes estados detetados pelo lençol	72
6.5	Capacitância: Lençol húmido	73
7.1	Circuito de testes	75
7.2	Conectores macho do lençol	76
7.3	Conector desenvolvido	76
7.4	Medidas da base da caixa	76
7.5	Projeto da caixa a ser implementada	76
7.6	Aspeto final do Transmissor	77
7.7	Invólucro: Botão On/Off e suporte para pilha 9 V	77
7.8	Esquema da implementação do circuito numa PCB	77
7.9	Vista superior da PCB	78
7.10	Vista inferior da PCB	78
7.11	Processo de soldagem das <i>pistas</i> da PCB	78
7.12	Integração de circuito final na caixa	78
A.1	Menus da aplicação para smartphone	83
A.2	Processo de ligação à rede Wi-Fi por parte do utilizador	84
B.1	Medidas do conector do lençol	85
B.2	Processo de desenvolvimento do invólucro e conector	85

Lista de Tabelas

2.1	Resultados do estudo	10
3.1	Tabela de Verdade para um Flip-Flop SR	24
3.2	Permissividade relativa de alguns materiais	28
3.3	Resumo de Características do ESP-WROOM-32	31
4.1	Permissividade relativa de alguns tecidos não condutores	42

Abreviaturas e Símbolos

ADC	Analog to Digital Converter
ALU	Arithmetic Logic Unit
AOT	Ahead-of-time
BaaS	Backend-as-a-Service
BLE	Bluetooth Low Energy
CI	Circuito Integrado
CPU	Central Processing Unit
CU	Control Unit
DAC	Digital to Analog Converter
DDoS	Distributed Denial-of-Service
EEPROM	Electrically Erasable Programmable Read-Only Memory
EPROM	Erasable Programmable Read-Only Memory
GPIO	General Purpose Input Output
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
JIT	Just-in-time
M2M	Machine-to-Machine
OMS	Organização Mundial de Saúde
PCB	Printed Circuit Board
RAM	Random Access Memory
ROM	Read-Only Memory
RFID	Radio-Frequency Identification
SDK	Software Development Kit
SoC	System-on-chip
SQL	Structured Query Language
UI	User Interface
UML	Unified Modeling Language
WEB	World Wide Web

Capítulo 1

Introdução

No panorama atual da saúde, tendo em conta o papel fundamental que os cuidadores desempenham, a evolução tecnológica tem sido crucial na melhoria da qualidade de vida dos pacientes. Em particular, os dispositivos médicos surgem como ferramentas essenciais para proporcionar cuidados personalizados e eficazes.

O contexto específico do presente estudo, diz respeito ao desenvolvimento de um sistema inovador para a monitorização de todas as pessoas que requeiram cuidados especiais. Este, propõe-se abordar a necessidade e conceção de um sistema embebido de monitorização de acamados, capaz de detetar a saída do paciente da cama, e quando presente, monitorizar o nível de humidade, gerando alarmes sempre que um comportamento anormal seja detetado, como o paciente levantar-se ou libertar qualquer tipo de fluído.

Ao longo deste documento, serão explorados detalhadamente os objetivos específicos do sistema, o desenvolvimento de hardware e software necessários e a criação de uma plataforma digital para interface e comunicação. Através desta abordagem, almeja-se contribuir para o avanço da tecnologia aplicada à saúde, proporcionando soluções inovadoras que beneficiem tanto pacientes quanto cuidadores.

1.1 Enquadramento e motivação

O crescente desafio de melhorar a qualidade de vida da população idosa e o aumento dos casos de demência, tem elevado a demanda por soluções tecnológicas na área dos cuidados de saúde. De acordo com a Organização Mundial de Saúde, em 2020 mil milhões de pessoas tinham mais de 60 anos, número que duplicará até 2050. Já o número de pessoas com 80 ou mais anos, por sua vez, deve triplicar no mesmo período, chegando aos 426 milhões [1]. É essencial agir agora para garantir que essas pessoas possam viver da forma mais saudável possível, pelo maior tempo possível.

À medida que a sociedade envelhece, estima-se que o número de pessoas com demência em todo o mundo aumente de 55 milhões em 2019 para 139 milhões em 2050 [2]. Neste contexto, a desorientação de pacientes, sintoma frequente associado à demência, representa uma preocupação

significativa para os cuidadores, uma vez que, estes utentes podem a qualquer momento levantar-se das suas camas "desnorteados" sem saber para onde ir, o que exige uma resposta eficaz para garantir a segurança dos pacientes e a tranquilidade dos familiares. Outro problema frequente é a incontinência urinária, ou por outras palavras, a perda involuntária de urina, que afeta sobretudo pessoas com 60 ou mais anos. De acordo com diretrizes da OMS a prevalência da incontinência urinária relatada em estudos populacionais varia de 9.9% a 36.1% e é o dobro em mulheres mais velhas do que nos homens [3]. Para além de pessoas com demência, é necessário assegurar cuidados às pessoas com mobilidade reduzida, que por vezes têm de fazer as suas necessidades parcial ou totalmente acamadas, neste contexto, a incontinência urinária e fecal torna-se uma preocupação significativa para este tipo de pacientes, uma vez que, certos componentes da urina e fezes podem atacar a pele e causar danos, além de que, o contacto contínuo deve ser evitado, existindo uma mudança rápida dos lençóis que compõem a cama.

O trabalho desenvolvido e documentado nesta dissertação, pretende ser uma resposta para todas as problemáticas expostas acima, baseando-se na criação de um sistema embebido *Internet Of Things* (IoT), que monitoriza o estado de presença e de humidade na cama de um doente, alertando o cuidador em situação anormal. Foi proposto pela *Invacare*, uma empresa líder na produção e distribuição de produtos médicos de apoio domiciliário e de cuidados continuados.

1.2 Objetivos

A presente dissertação, tem como objetivo principal o desenvolvimento de um sistema embebido que monitorize a presença ou ausência de pacientes acamados, assim como, os níveis de humidade associados. Este sistema pode ser utilizado em vários contextos e setores, por exemplo: para uso profissional, no apoio aos enfermeiros nos hospitais ou em lares e, por outro lado, para uso doméstico, ajudando os cuidadores de pessoas idosas. Atualmente, existe um kit chamado *TEXIBLE Wisbi* desenvolvido por uma empresa austríaca especializada no design, desenvolvimento e produção de soluções têxteis inteligentes, chamada *TEXIBLE* [4]. O kit atual possibilita a realização dessa monitorização, no entanto, a proposta do sistema pela *Invacare* visa reconfigurar o projeto, eliminando a necessidade do transmissor desenvolvido pela empresa Austríaca. A intenção é integrar o produto num ecossistema de soluções *Invacare* e, ao mesmo tempo, avançar em direção a uma solução integrada com outros sensores presentes na cama.

O sistema desenvolvido, deve ser capaz de detetar variações do estado de presença do paciente e, alertar o cuidador quando o mesmo estiver ausente ou a humidade no lençol for superior a 10%. Para isso, será necessário também o desenvolvimento de uma aplicação para ser utilizada pelo cuidador, de forma a que, este tenha acesso em tempo real ao estado da cama do paciente.

Esta dissertação, visa facilitar a vida de todos os envolvidos e aumentar o nível de monitorização de acamados.

1.3 Estrutura da Dissertação

Para além da introdução, esta dissertação contém mais 7 capítulos. No capítulo 2, é descrita a definição, funcionalidades e aplicações de têxteis inteligentes e são apresentados artigos científicos que abordam o seu uso nos cuidados de saúde. O capítulo 3, contém os fundamentos teóricos que sustentam as tecnologias utilizadas ao longo deste projeto. No capítulo 4 é abordada a arquitetura do projeto desenvolvido. O capítulo 5 explica os detalhes da interface IoT utilizada. No capítulo 6 apresentam-se resultados do presente projeto. No capítulo 7 é descrito o processo de transição do protótipo inicial, originalmente implementado numa *breadboard*, para uma solução mais próxima do que será um produto final, envolvendo o projeto e montagem de uma PCB (*Printed Circuit Board*). Por último, no capítulo 8 são apresentadas as conclusões do estudo realizado e feita uma referência ao trabalho que poderá vir a ser desenvolvido no futuro.

Capítulo 2

Têxteis inteligentes

Neste capítulo, é apresentada uma revisão da literatura sobre os têxteis inteligentes, as suas funcionalidades e aplicações, a noção de sensores de pressão flexíveis e algumas tecnologias de apoio aos cuidadores disponíveis atualmente.

2.1 Definição

A evolução da qualidade de vida e as crescentes expectativas dos consumidores, têm promovido uma transformação significativa no setor têxtil. Nesse contexto, as roupas, frequentemente consideradas como uma segunda pele dos seres humanos, tornam-se não apenas elementos comuns, mas também candidatas ideais para a integração de materiais inteligentes, integrando tecnologia, onde as fibras e tecidos são reinventados, dando origem aos *smart textiles*.

O termo têxteis inteligentes ou *smart textiles*, deriva do conceito de materiais inteligentes, que foi apresentado pela primeira vez em 1989 no Japão, pelo Professor Toshiyoshi Takagi [5]. À medida que surge mais literatura de investigação e protótipos de produtos, existem várias definições para têxteis inteligentes, mas todas são unânimes e estão de acordo com Xiaoming Tao, cuja publicação *Smart Fibres, Fabrics and Clothing*, é uma referência na área dos *Smart Textiles* e refere que os têxteis inteligentes são definidos como: "os materiais e estruturas que sentem e reagem às condições ambientais ou estímulos, nomeadamente mecânicos, térmicos, químicos, elétricos, magnéticos ou de outras fontes" [6].

Os têxteis inteligentes podem ser divididos em 3 grupos [7]:

- Têxteis inteligentes passivos: capazes apenas de detetar o ambiente/utilizador, com base em sensores;
- Têxteis inteligentes ativos: deteção reativa a estímulos do ambiente, integrando uma função de atuador e um dispositivo de deteção, onde se enquadram os têxteis com memória de forma;
- Têxteis muito inteligentes: capazes de sentir, reagir e adaptar o seu comportamento às circunstâncias do ambiente.

O nível de complexidade é crescente, ou seja, os têxteis inteligentes passivos, apresentam o nível de complexidade mais baixo, uma vez que, funcionam basicamente como sensores, os inteligentes ativos, que para além de conseguirem sentir os estímulos, produzem também uma reação ao estímulo e por último os têxteis muito inteligentes, que para além de combinarem as duas funcionalidades anteriormente referidas, são capazes de se adaptar ao meio ambiente.

Para um material ser inteligente, deve pertencer a um sistema, constituído por um sensor, um atuador e uma unidade de processamento externa, responsável pela comunicação. Devem ser tomados cuidados, para que, as pessoas não sintam fios ou outros componentes eletrónicos a tocar no seu corpo [8]

2.2 Funcionalidades de um têxtil inteligente

Um têxtil inteligente, ou e-têxtil apresenta cinco funções principais, nomeadamente: sensorização, processamento de dados, atuação, armazenamento e comunicação, as quais se sistematizam nos pontos seguintes [9]:

1. Os sensores, são dispositivos que recebem um estímulo, como calor, pressão entre outros e em resposta, emitem um sinal que seja capaz de ser convertido e interpretado por outros dispositivos. Normalmente o sinal é de natureza elétrica.
2. A componente de processamento de dados é uma função facultativa, apenas requerida se o material for ativo. Até ao momento, ainda requer a utilização de componentes eletrónicos, no entanto, esforços estão a ser feitos para integrar estes componentes ativos nas fibras, visando à miniaturização e flexibilidade.
3. Os atuadores, por sua vez, respondem a estímulos originados pelos sensores e, após o processamento de dados, realizam ações como movimento ou libertação de substâncias. Materiais com memória de forma, como ligas e polímeros, têm aqui um papel preponderante.
4. Quanto à capacidade de armazenamento, esta é essencial para o armazenamento de dados ou energia, especialmente para alimentar os sensores, o processamento de dados e a atuação. Estratégias de gestão eficiente de energia, incluem a combinação adequada de fontes de energia e capacidade de armazenamento, sendo que, já existem soluções que produzem energia a partir do calor corporal, através da ação mecânica das roupas ou por radiação solar.
5. Por fim, a comunicação nos têxteis inteligentes assume diversas formas, pode acontecer entre o utilizador e o produto têxtil e no sentido inverso. Fibras óticas, fios condutores e tecnologias de tela flexível são algumas das abordagens utilizadas para facilitar esta comunicação.

A integração da característica inteligente no material têxtil pode ser realizada de várias formas, incluindo a utilização de diferentes tipos de fibras, a integração de fios condutores através de

tecelagem, o uso de tintas condutoras, entre outras abordagens que podem passar pela utilização de uma unidade externa ao material têxtil.

O conceito de *smart textiles* tende a evoluir para que todo o sistema seja composto por materiais têxteis. A principal dificuldade em atingir esse nível é obter um produto final maleável, flexível, com boa resistência mecânica, condutividade elétrica e ainda ser resistente à água [10].

2.3 Sensores têxteis de pressão

Os sensores têxteis de pressão, ou sensores de pressão flexíveis são dispositivos projetados para medir e detetar, variações na pressão exercida sobre uma superfície flexível. Associada à evolução da sociedade, existem vários novos tipos de dispositivos em desenvolvimento, que se baseiam em sensores de pressão flexíveis e têm aplicações na área automóvel, saúde entre outras [11]. Existem diferentes métodos com características próprias, capazes de converter estímulos táteis em sinais elétricos, os quais são:

- **Piezo-resistividade:** Os sensores piezo-resistivos, apresentam uma variação na resistência elétrica quando são submetidos a uma deformação mecânica. São adequados para o uso em grandes intervalos de pressão, tornando possível que grandes deformações sejam medidas de forma confiável [12].
- **Piezo-eletricidade:** refere-se às cargas elétricas geradas em certos tipos de materiais sólidos, como cristais e cerâmicas específicas, em resposta a tensões mecânicas aplicadas. Devido à sua alta sensibilidade e rápido tempo de resposta, os sensores piezo-elétricos são amplamente utilizados na deteção de pressões dinâmicas, como vibrações sonoras e deslizamentos.
- **Capacitância:** A capacitância é a capacidade de armazenar carga elétrica. Num condensador de placas paralelas, a capacitância é afetada pela área das placas e pela sua distância, conforme detalhado na secção 3.3.2. Quando é aplicada uma pressão, as placas movem-se e a capacitância varia. Esta variação é geralmente da ordem de vários picofaradays, e pode ser utilizada para controlar a frequência de um oscilador.

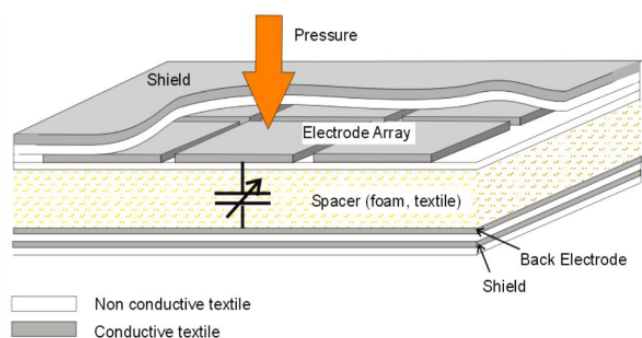


Figura 2.1 – Sensor têxtil capacitivo apresentado por Mayer *et al.* [13]

Meyer *et al.* (2006) [13], apresentaram um sensor de pressão capacitivo têxtil puro, que consiste numa estrutura básica de três camadas, formando uma capacitância com um dielétrico não condutor sensível à pressão, conforme ilustrado na figura 2.1. O sensor descrito, é composto por duas camadas exteriores de tecido condutivo, separadas por uma espuma espaçadora elástica, apelidada de "espaçador compressível". Este espaçador, quando é comprimido ou alterado, permite ao sensor medir uma variedade de pressões. Este design, foi aplicado na deteção da atividade muscular de um braço humano, sendo fixado nos músculos *bíceps* e *tríceps* através de uma banda elástica.

Durante o teste, que envolvia o levantamento de um peso de 2 kg, observou-se um aumento na pressão exercida pelos músculos sobre a banda elástica, demonstrando a eficácia do sensor em detetar variações de pressão associadas à atividade muscular.

2.4 Estudos existentes

Existem várias tecnologias para cuidados de enfermagem, incluindo robôs para acalmar pacientes com demência, óculos de realidade aumentada e colchões inteligentes que tentam prevenir/minimizar o aparecimento de úlceras de pressão.

Um estudo alemão: "*Expectations of new technologies in nursing care among hospital patients in Germany – an interview study*" levado a cabo pelo Departamento de Orientação ao Paciente e Educação em Saúde do Instituto de Epidemiologia da Escola de Medicina de Hannover e pelo Grupo de Pesquisa em Medicina Digital da Universidade de Bielefeld na Alemanha, visou investigar atitudes, expectativas, preocupações e efeitos da implementação de novas tecnologias de assistência aos cuidados de enfermagem, sob a perspetiva dos pacientes. O estudo consistiu na realização de entrevistas a 17 pacientes com idades compreendidas entre os 23 e os 87 anos de uma ala de cirurgia de traumatologia de um hospital universitário na Alemanha, onde foram apresentadas oito tecnologias diferentes de cuidados de enfermagem e os pacientes foram consultados sobre os impactos positivos e negativos que estas teriam.

As tecnologias que constituíram o estudo para além do *Texible Wisbi* foram [14]:

- *Bedside Terminal*: Sistema multimédia, colocado próximo da cama e que permite entretenimento, por exemplo ouvir música e a comunicação das necessidades dos pacientes à equipa médica;
- *Pflegebrille - Nursing Glasses*: Óculos eletrónicos de realidade aumentada para orientação prática, permitindo que os cuidadores realizem tarefas sem usar as mãos, através de movimentos de cabeça ou comandos de voz;
- *Ekamove*: Colchão inteligente que reposiciona pacientes para prevenir úlceras de pressão;
- *JustoCat*: Robô em forma de gato para acalmar pacientes com demência, mimetizando interações com animais reais;
- *Lea*: Andarilho com *tablet* que suporta rotas de caminhadas e possui alarmes de queda;

- *Relay*: Robô autônomo para transporte seguro de medicamentos ou amostras em ambientes hospitalares.
- *Carbon Hand*: Consiste numa luva robótica usada sobre a mão que suporta os movimentos naturais, como agarrar um objeto, fazendo com que a utilização dos músculos e tendões do utilizador seja reduzido.

Os resultados do estudo alemão, revelam que as pessoas com mais estudos colocavam algumas questões relacionadas com falhas na tecnologia, no entanto todos foram unânimes quanto à influência positiva da utilização destes equipamentos nos cuidados dos pacientes.

Na tabela 2.1, inserida na página seguinte, podem ser vistos mais detalhadamente os resultados do estudo da universidade de Bielefeld.

Tabela 2.1 – Resultados do estudo

	Aspetos negativos	Aspetos positivos
Impacto nos pacientes	• Implicações emocionais ou sociais negativas (perda de autonomia); • Preocupações relativamente a pacientes idosos, devido à falta de experiência com tecnologia);	• Melhoria da saúde e bem-estar; • Melhoria da experiência do paciente no hospital (maior conforto);
Impacto nos cuidadores	• Substituição de enfermeiros por máquinas; • Medo de perda de empregos; • Aumento da pressão de trabalho devido à operação com máquinas novas e novas tarefas de manutenção;	• Menor esforço físico exigido; • Melhor organização e gestão do tempo; • Foco maior no cuidado ao paciente; • Adaptação a desafios emergentes;
Impacto nos cuidados de enfermagem	• Redução de postos de trabalho; • Menos contacto entre enfermeiros e pacientes;	• Aumento da qualidade de prestação de cuidados aos pacientes; • Bem-estar da equipa médica que se sente valorizada pela preocupação dada aos seus esforços físicos;

Considerando o estudo, e apesar de ainda existirem alguns pontos negativos em relação ao uso de tecnologia na indústria hospitalar, é necessário ter em atenção, que as pessoas alvo tinham maioritariamente idades entre os 50 e os 59 anos, o que pode fazer com que tenham algumas opiniões céticas quanto ao uso da tecnologia. Focando no sistema desenvolvido, que como dito anteriormente é uma reestruturação da tecnologia *Texible Wisbi*, convém lembrar que será uma mais valia no caso de pessoas com incontinência urinária ou fecal, uma vez que, de acordo com dados do *Journal of Wound, Ostomy and Continence Nursing*, que realizou um estudo para avaliar o impacto da incontinência em danos na pele: 42.5% das pessoas com incontinência tem danos na pele, como pele avermelhada e dermatite [15], das quais cerca de 20% são pessoas residentes em lares [16].

Outro estudo apresentado em 2020 na décima terceira Conferência Internacional de Informática em Saúde, uma conferência onde médicos e investigadores se reúnem para apresentar os últimos avanços da informática aplicada à saúde, destaca a crescente prevalência e os cuidados associados à incontinência em lares de idosos. A pesquisa foi levada a cabo em dois lares na Bélgica, e ao longo de três semanas, foram testados quatro produtos que mediam parâmetros relevantes para inferir sobre a incontinência em idosos. Os mais relevantes foram o lençol *Texible WISBI* e o *DFree*, um dispositivo colocado na região do osso púbico, que deteta o tamanho da bexiga utilizando uma tecnologia não invasiva de ultrassom [17]. Relativamente à utilização, o sensor DFree foi tolerável principalmente nos primeiros dias, dependendo da mobilidade do residente. Já o uso do lençol inteligente passou praticamente despercebido e foi considerado mais fácil de usar pelos profissionais de saúde. Quanto aos resultados, o lençol mostrou-se capaz de detetar humidade e foi o mais preciso para detetar os períodos de descanso durante a noite, ou seja, analisando os resultados era perceptível quando o doente se levantava ou não.

Num outro estudo, D. Ghosh *et al.* propuseram um dispositivo inteligente para monitorizar o nível de solução salina em terapias intravenosas, destacando a importância da vigilância para evitar acidentes graves. O sistema utiliza sensores e tecnologias de Internet das Coisas, incluindo um sensor de carga e o microcontrolador ESP32. O sensor converte o peso da garrafa em tensão, enquanto o ESP32 gera mensagens com base na tensão recebida, utilizando o protocolo MQTT-S para comunicação, uma vez que, é um protocolo leve projetado para dispositivos de consumo ultra baixo e de baixo custo [18].

Concluindo, e de uma forma simplista, as implementações de IoT nos cuidados de saúde baseiam-se na existência de sensores, que enviam dados para um microcontrolador responsável por colocar esses dados numa página da Web via Wi-Fi.

2.5 Protótipos de têxteis inteligentes

Para adicionar um sistema embebido a um têxtil, é particularmente importante ter atenção a 3 fatores, os quais: o stress mecânico a que o circuito estará sujeito, o stress químico devido a lavagens, libertação de suor e até mesmo a chuva e por último as condições térmicas, quando por exemplo se passa o tecido a ferro. Nesta secção são ilustrados alguns protótipos no âmbito dos têxteis inteligentes.

VitalJacket Baby

A *biodevices*, é uma empresa portuguesa, que desenvolve e comercializa soluções de Engenharia Biomédica para o apoio ao diagnóstico médico. Desenvolveram o *VitalJacket Baby*, um *body* para bebés que monitoriza o coração, sendo o primeiro sistema de eletrocardiograma ambulatório desenvolvido para bebés. Permite ao bebé brincar e dormir confortavelmente como se de um *body* normal se tratasse [19].



Figura 2.2 – VitalJacket Baby [19]

Nike e iPod Sports Kit

Em 2006, a *Apple* em parceria com a *Nike* lançou o *Nike + iPod Sports kit*, um sistema desenvolvido para monitorizar e registar dados de atividades físicas, principalmente corridas. O sistema consistia num pequeno sensor que podia ser colocado na sapatilha e um recetor que se conectava ao *iPod*. O sensor registava informações como distância percorrida, tempo, ritmo e calorias queimadas durante a atividade física. Esses dados eram transmitidos sem fio para o *iPod*, onde o utilizador poderia acompanhar o seu desempenho em tempo real.



Figura 2.3 – Nike + iPod Sports Kit [20]

ANGEL

Para além do lençol *Texible Wisbi*, a empresa *TEXIBLE* desenvolve outros têxteis inteligentes, tais como a *ANGEL*. Esta representa uma inovação em roupa de trabalho inteligente, projetada para situações de emergência. Equipada com eletrónica avançada e tecidos confortáveis, esta roupa possui elétrodos integrados capazes de detetar acidentes elétricos ou quedas. Em caso de acidente, a *ANGEL* ativa automaticamente uma cadeia de resgate individual, alertando uma lista previamente definida de contactos de emergência.



Figura 2.4 – ANGEL [21]

Stappone

O *Stappone*, também concebido pela *TEXIBLE*, utiliza sensores de pressão têxtil integrados em palmilhas. É desenhado para uso diário e fornece dados detalhados sobre a distribuição e intensidade da pressão exercida em cada ponto do pé. As palmilhas captam a pressão em tempo real, permitindo a identificação de possíveis anomalias na marcha do utilizador. Os dados são depois transmitidos por *bluetooth* para um computador ou *tablet*, facilitando a análise e o acompanhamento do padrão de marcha, para aplicações em saúde, reabilitação e até mesmo desporto.

Este sistema está ilustrado na figura 2.5.

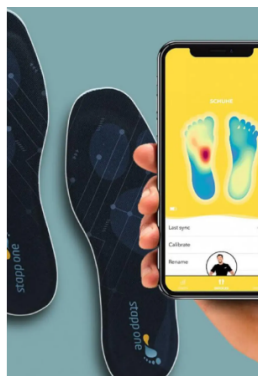


Figura 2.5 – Tecnologia Stappone [21]

NanoStim

O *NanoStim* é um protótipo resultante de um projeto de investigação português, levado a cabo por um consórcio de seis instituições, como o grupo Impetus, a Universidade do Minho entre outros. Este permite a reabilitação de lesões musculares à distância, através de eletroestimulação. Os biosinais são enviados através de algoritmos de inteligência artificial.



Figura 2.6 – NanoStim [22]

SMART-HEALTH-4-ALL

O *SMART-HEALTH-4-ALL* foi um projeto que agregou mais de 30 empresas. Iniciou-se em 2020 e terminou em 2023. Tinha como objetivo desenvolver produtos inovadores nas áreas da saúde digital e das tecnologias médicas. No âmbito deste projeto, destaca-se o desenvolvimento do PPS4, que consiste numa almofada ergonómica capaz de alterar a sua forma, ajudando o utilizador a melhorar o posicionamento da cabeça durante o sono. Destaca-se também o *MuscleBAN Strap*, que permite realizar exercícios de fisioterapia de um modo autónomo e remoto, em casa do paciente [23].

Estes dois produtos foram desenvolvidos pelo CITEVE, um centro tecnológico sediado em Famalicão responsável por apoiar o desenvolvimento das capacidades técnicas e tecnológicas das indústrias têxtil e do vestuário, através da difusão da inovação [24].

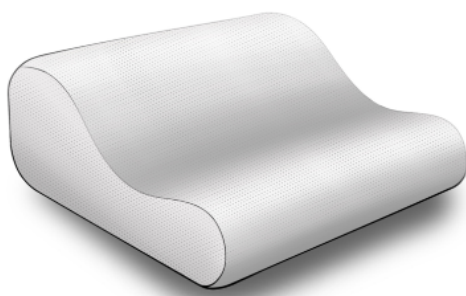


Figura 2.7 – Almofada ergonómica PPS4 [25]



Figura 2.8 – MuscleBAN Strap [25]

De uma forma geral, todos os protótipos apresentados contribuem de forma direta ou indireta para os cuidados médicos do utilizador. O *ANGEL* é o mais diferente dos protótipos apresentados, no sentido em que não efetua monitorização de sinais vitais, mas deteta acidentes elétricos e avisa uma lista de contactos predefinida o mais rápido possível.

Concluindo, destaca-se a empresa *Biodevices* e o CITEVE, uma vez que, são duas entidades portuguesas que contribuem para o avanço da tecnologia utilizada nos têxteis inteligentes.

Capítulo 3

Fundamentos

Neste capítulo, apresentam-se os principais conceitos relacionados às escolhas tecnológicas adotadas na solução proposta, tendo em conta os objetivos específicos do projeto.

3.1 Lençol *TEXIBLE Wisbi*

Como já mencionado neste documento, a *Texible* desenvolveu uma solução inovadora para o cuidado ativo de pessoas acamadas, denominada *TEXIBLE Wisbi*. Este produto consiste num lençol inteligente, capaz de detetar automaticamente a presença de humidade ou fluidos corporais, bem como o movimento do paciente ao sair da cama, alertando o cuidador em qualquer um destes casos. Este sistema de alerta automático é particularmente útil em ambientes de cuidados médicos, oferecendo uma monitorização contínua dos pacientes.

A seguir, são apresentadas as funcionalidades do lençol *TEXIBLE Wisbi*, conforme descrito na brochura [26] da *TEXIBLE*:

- Conforto máximo devido à sua composição em dupla flanela de algodão rugoso.
- Embora possa ficar molhado, é projetado para evitar vazamentos. Isso significa, que protege eficazmente o colchão e o lençol subjacente contra a humidade.
- Possibilidade de lavagem como um têxtil comum, sendo recomendável uma lavagem a 95 °C.
- É respirável, graças a uma composição multi camadas e a uma membrana respirável.
- Comparativamente a outros e-têxteis, apresenta uma espessura consideravelmente mais fina.

Os sensores utilizados num têxtil inteligente, para além de terem as propriedades de um sensor convencional, possuem características particulares como a dimensão, para serem incorporados nos têxteis. No caso do *Texible Wisbi*, não existem propriamente sensores integrados no lençol, mas antes, uma disposição inteligente de materiais condutores [27] que, através da variação de propriedades como a resistência e capacitância, torna possível detetar humidade e presença de

pessoas. Esta inovação, transforma o tecido num sensor por si só, oferecendo uma solução discreta e confortável para os pacientes.

O *TEXIBLE Wisbi* é colocado por baixo do lençol inferior da cama. Este está conectado a um transmissor, que contém o circuito de condicionamento e processamento do sinal, responsável por enviar um aviso via WLAN para uma aplicação de *smartphone*, caso seja detetada a saída da cama ou a presença de fluidos corporais.

Na aplicação, o cuidador pode verificar em tempo real o estado do lençol inteligente ou receber alertas sobre quaisquer situações anormais.

As especificações técnicas relevantes são:

- **Tamanho:** 64 x 100 cm.
- **Capacidade de absorção de água:** 700 ml/m².
- **Ciclos de lavagem recomendados:** 100 ciclos.
- **Alimentação:** 3 Pilhas AA pequenas.
- **Conexão Wlan:** 2.4 GHz 802.11b/g/n - WPA/WPA2.

A figura 3.1 ilustra a face traseira do lençol inteligente, a qual fica em contacto com a cama. Observem-se os filamentos que o constituem.



Figura 3.1 – Filamento incorporado no lençol *TEXIBLE Wisbi*

3.2 Internet of Things

O termo *Internet of Things* foi utilizado pela primeira vez em 1999 pelo pioneiro britânico em tecnologia, Kevin Ashton, para descrever um sistema no qual objetos do mundo físico poderiam ser conectados à Internet através de sensores [28]. Mais especificamente, Kevin referiu as vantagens de colocar etiquetas de identificação por radiofrequência (RFID) nos objetos de cadeias de abastecimento, com a finalidade de contar e rastrear os bens, sem necessidade de intervenção humana. Embora a definição de IoT ainda não esteja uniformizada em todo o mundo, dada a sua vasta amplitude de significados e diversas aplicações, este documento adotará a definição proposta pela *ORACLE*, uma multinacional de tecnologia especializada no desenvolvimento e comercialização de hardware e software. Segundo a *ORACLE*, a *Internet of Things* descreve a rede de objetos físicos incorporados com sensores, software e outras tecnologias, com o objetivo de conectar e trocar dados com outros dispositivos e sistemas através da Internet [29].

A ideia de integrar sensores e inteligência em objetos comuns, é discutida desde as décadas de 1980 e 1990. Apesar de existirem alguns projetos que remontam a essa época, o desenvolvimento foi lento inicialmente devido às limitações tecnológicas, uma vez que, os componentes eletrônicos eram demasiado grandes e não havia uma forma eficiente para os objetos se comunicarem. A mudança aconteceu quando os processadores se tornaram suficientemente acessíveis e eficientes em termos energéticos para serem usados em larga escala, tornando viável a conexão de bilhões de dispositivos. A introdução das etiquetas RFID, uma tecnologia sem fios e de baixo consumo, foi um avanço significativo, além disso, a adoção do *Internet Protocol version 6* (IPv6), foi crucial para fornecer um número suficiente de endereços IP para todos os dispositivos no mundo [30]. Esse desenvolvimento foi essencial para o crescimento exponencial da Internet das Coisas (IoT).

Inicialmente, a IoT foi pensada para um contexto industrial, onde a sua aplicação é conhecida como *Machine-to-Machine* (M2M), destinada à troca automática de dados entre dispositivos. Assim, as organizações podiam acompanhar e gerir ativos, inventários, frotas de transporte entre outros [31]. Hoje em dia, a Internet das Coisas tornou-se um termo popular para descrever cenários nos quais a conexão com a Internet e a capacidade de computação se estendem a uma variedade de objetos do quotidiano, como por exemplo, uma lâmpada que pode ser ligada através de uma aplicação para *smartphone*, ou um ar condicionado que possa ser inicializado e parametrizado à distância.

Em grande escala, a Internet das Coisas é utilizada em automação residencial, por exemplo no controlo e monitorização dos sistemas eletrônicos, elétricos e mecânicos utilizados em diversos tipos de edifícios [32], ou até mesmo nas *Smart Cities*.

3.2.1 Modelos de comunicação da Internet of Things

Tecnicamente, é útil perceber como os dispositivos integrados num sistema IoT se conectam e comunicam. De acordo com um documento escrito em 2015, pela *Internet Architecture Board* (IAB) [33] existem 4 modelos de comunicação comuns, usados por dispositivos IoT, os quais são:

Device-to-Device: O modelo de comunicação dispositivo-para-dispositivo representa dois ou mais dispositivos que se conectam e comunicam diretamente um com o outro, através de vários tipos de redes sem fios. Muitas vezes, são utilizados protocolos como *Bluetooth*, *Z-Wave* ou *ZigBee* para estabelecer comunicações diretas dispositivo-a-dispositivo. Este modelo de comunicação é normalmente utilizado em aplicações como sistemas de automação residencial, nas quais os dispositivos geralmente utilizam pequenos pacotes de dados para comunicarem com baixos requisitos de transmissão de dados.

Device-to-Cloud: No modelo de comunicação dispositivo-para-nuvem, o dispositivo IoT conecta-se diretamente a um serviço de nuvem da Internet, como por exemplo, um provedor de serviço de aplicação, para trocar dados e mensagens de controlo. Esta abordagem frequentemente tira vantagem de mecanismos de comunicação existentes, como o Wi-Fi, para estabelecer uma conexão entre o dispositivo e a Internet, que finalmente se conecta ao serviço de nuvem. Este modelo é adotado por alguns dispositivos populares, como a tecnologia *Samsung SmartTV*, que utiliza a conexão à Internet para transmitir informações de visualização do utilizador para análise pela *Samsung* e ainda para habilitar os recursos interativos de reconhecimento de voz da TV.

Para uma melhor compreensão, a figura 3.2 ilustra o modelo *Device-to-Cloud*.

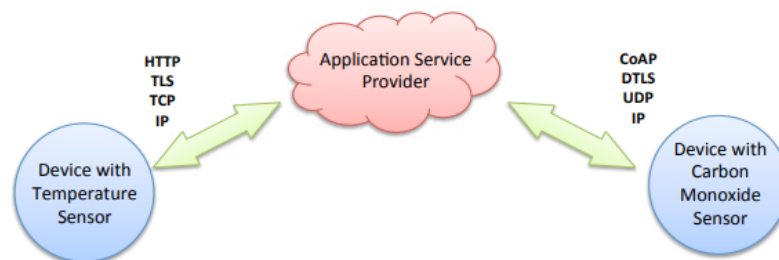


Figura 3.2 – Modelo de comunicação: *Device-to-Cloud* [34]

Device-to-Gateway: Neste modelo, o dispositivo IoT conecta-se ao serviço em nuvem, através de um *Local Gateway*, o que significa que existe um intermediário entre o dispositivo e o serviço em nuvem, muitas vezes, este intermediário é um *smartphone*, que executa uma aplicação responsável por comunicar e transmitir os dados de um dispositivo IoT local para a nuvem, visto que, existem dispositivos que não têm a capacidade de se conectar diretamente à nuvem. Um exemplo bastante comum nos dias de hoje, é o uso de pulseiras para monitorização de exercício físico, onde o *smartphone* é a *gateway*, responsável por receber os dados via *Bluetooth* e posteriormente uma aplicação transmite as informações para a Internet através do Wi-Fi.

A figura 3.3 mostra o modelo *Device-to-Gateway*.

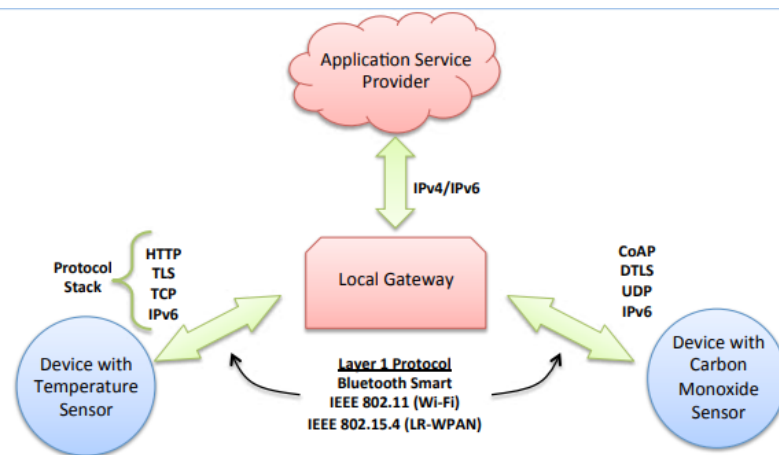


Figura 3.3 – Modelo de comunicação: *Device-to-Gateway* [34]

Back-End Data-Sharing: O modelo de partilha de dados no *back-end* refere-se a uma arquitetura de comunicação que permite aos utilizadores exportar e analisar dados de objetos inteligentes de um serviço em nuvem em combinação com dados de outras fontes. Um exemplo prático para melhor percepção é: uma quinta equipada com sensores conectados que monitorizam as características do solo. Esses dados são enviados para o Servidor A, onde uma aplicação de agricultura inteligente está em execução. No servidor, os dados são processados e contextualizados, resultando na geração de recomendações e indicadores valiosos para otimizar as práticas agrícolas na fazenda. Para agregar mais valor aos dados dos sensores, o Servidor A comunica com os Servidores B e C, recolhendo dados meteorológicos, histórico da plantação e características de cada tipo de planta fornecidos pelas aplicações. Com base nesses dados, é possível saber qual o melhor momento para fazer a colheita, aumentando consideravelmente a produção [35].

Na maioria das aplicações existentes é adotado mais do que um modelo de comunicação. O projeto desenvolvido no âmbito da dissertação, encaixa-se predominantemente no modelo *Device-to-Cloud*, na medida em que o lençol inteligente envia dados para uma plataforma de nuvem, como será explicado posteriormente. O fluxo de dados é simples e direto e os intervenientes são o sensor, a plataforma de nuvem e a aplicação para *smartphone*. Por outro lado, pode-se dizer que existem também traços do modelo *Device-to-Gateway*, uma vez que, o sensor só consegue enviar dados para a nuvem porque tem um microcontrolador associado, responsável por estabelecer ligação via Wi-Fi com a internet, comportando-se como uma *Local Gateway*.

3.2.2 Desafios da *Internet of Things*

A implementação da IoT na área da saúde trouxe inúmeras vantagens para a monitorização dos pacientes. No entanto, essa integração não está isenta de riscos, sobretudo no que concerne à segurança e privacidade dos dados recolhidos pelos sensores. Estes recolhem dados que muitas vezes são bastante sensíveis, como informações provenientes do ambiente doméstico, por exemplo, através de sistemas inteligentes como a *Amazon Alexa* ou o *Google Home*. É importante garantir que esses dispositivos cumprem com princípios básicos de segurança, como a criptografia dos dados.

No contexto da segurança, várias ameaças são identificadas, incluindo ataques físicos que afetam diretamente o hardware. Um exemplo claro é o ato de desconectar intencionalmente um dispositivo, interrompendo assim a sua operação normal. Existem também ataques de rede, sendo o *Distributed Denial-of-Service* (DDoS) um exemplo comum, este tipo de ataque visa sobrecarregar servidores, serviços ou redes com tráfego excessivo, tornando recursos essenciais indisponíveis [36]. À medida que mais dispositivos são conectados à Internet, aumenta a probabilidade de surgirem potenciais vulnerabilidades de segurança. Dispositivos IoT mal protegidos, podem servir como pontos de entrada para ciberataques, permitindo que pessoas mal-intencionadas reprogramem um dispositivo ou o utilizem de forma inadequada. A gravidade disso é evidente, especialmente em dispositivos críticos como semáforos ou *pacemakers*, que devem funcionar sem margem para erros.

Relativamente à privacidade dos dados, esta é uma preocupação central, que exige proteções robustas para os dados pessoais dos utilizadores. Em muitos contextos jurídicos, é necessária autorização legal para processar dados pessoais, garantindo assim a proteção contra acessos não autorizados e o uso indevido de informações [37]. A combinação de dados aparentemente inofensivos provenientes de sensores, pode fornecer um retrato completo do utilizador. Por exemplo, uma escova de dentes com ligação à Internet pode transmitir dados inofensivos sobre hábitos de escovagem, mas, quando combinada com informações do frigorífico e do dispositivo de monitorização de atividade física, cria um retrato detalhado dos hábitos da pessoa em questão. Isso pode representar um problema de privacidade para aqueles que desconhecem como as informações recolhidas são utilizadas [34].

Os dispositivos IoT, devem ser concebidos com alguns serviços básicos de segurança, tais como: autorização, autenticação, confidencialidade, disponibilidade, integridade e não repúdio [38]. Neste documento, destacam-se a autenticação e a confidencialidade por se considerar serem os mais relevantes. A autenticação, essencialmente, consiste na verificação da identidade do utilizador antes do acesso aos dados ser concedido.

3.3 Cadeia de medição

Uma cadeia de medição é constituída por um conjunto de elementos, que são devidamente associados de forma a medir uma determinada grandeza. Em geral, uma cadeia de medição é composta por um transdutor, responsável por converter uma grandeza física num sinal elétrico, por circuitos de condicionamento adequados, que podem incluir pontes de medição, amplificadores, filtros e outros dispositivos, projetados para amplificar e filtrar o sinal, garantindo que ele esteja nas condições ideais para ser convertido de analógico para digital pelo conversor A/D [39].

A figura 3.4 ilustra o processo descrito.



Figura 3.4 – Diagrama de blocos geral de uma cadeia de medição

No sistema que foi desenvolvido, o sensor adotado é o próprio *TEXIBLE Wisbi*. O circuito responsável pela aquisição do sinal é composto por um circuito integrado 555, amplamente reconhecido como um timer confiável, além de alguns condensadores e uma resistência. Para realizar o processamento do sinal, foi incorporada a placa de desenvolvimento ESP-WROOM-32. Nesta secção, será explicado o funcionamento dos componentes que compõem o circuito de aquisição e processamento de sinal, desenvolvido no contexto deste projeto.

3.3.1 Circuito Integrado 555

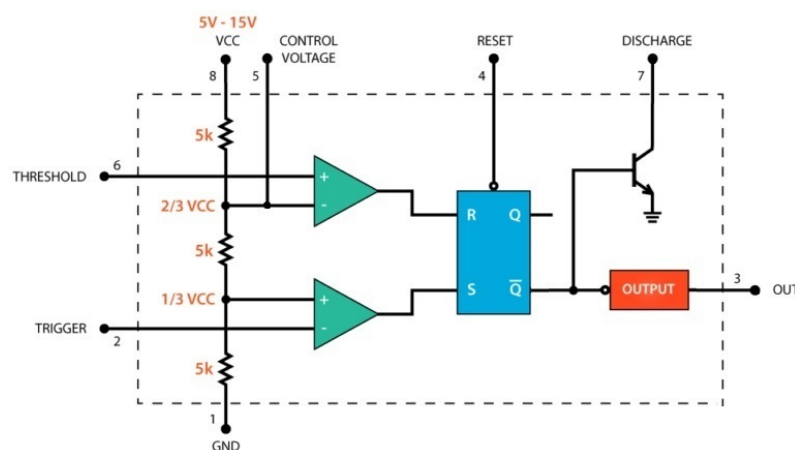


Figura 3.5 – Diagrama de blocos do circuito integrado 555 [40]

O 555 é um circuito integrado (CI), desenvolvido inicialmente para funcionar como oscilador e *timer*. Foi projetado por Hans Camenzind em 1971 e atualmente ainda é utilizado em várias

aplicações devido à sua versatilidade, nomeadamente quando se trata de temporizações, como por exemplo quando uma lâmpada liga por um determinado período de tempo quando alguém passa.

Na figura 3.5 está ilustrado o diagrama de blocos deste componente. Analisando a figura da esquerda para a direita, este é composto por um divisor de tensão, dois comparadores, um flip-flop SR, um transistor de descarga e o bloco de saída. O divisor de tensão é composto por três resistências de $5\text{ k}\Omega$, as quais criam duas tensões de referência em $\frac{1}{3}$ e $\frac{2}{3}$ da tensão fornecida que, conforme recomendado pelo fabricante tipicamente pode variar entre 5 a 15 V [40].

Quanto aos comparadores, um comparador é um elemento do circuito, cuja saída será nível lógico *HIGH* ou V_{CC} se $V_+ > V_-$ e será nível lógico *LOW* ou 0 V se $V_+ \leq V_-$ [41], tal como exemplificado na figura 3.6.

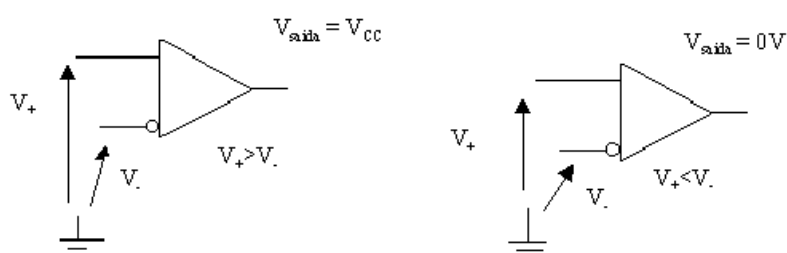


Figura 3.6 – Funcionamento dos comparadores [41]

A tabela de verdade do flip-flop SR pode ser consultada na tabela 3.1.

Tabela 3.1 – Tabela de Verdade para um Flip-Flop SR

Inputs		Output
S_n	R_n	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	?

O CI 555 utilizado, possui oito terminais, sendo dispostos quatro de cada lado. A identificação dos terminais está representada na figura 3.7.

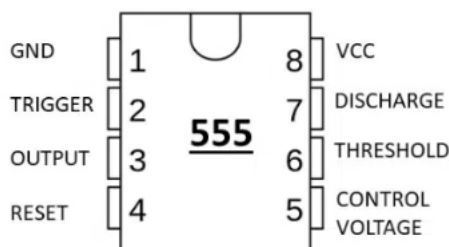


Figura 3.7 – Terminais do circuito integrado 555 [42]

1. **GND:** Representa a tensão de referência de 0 V.
2. **TRIGGER:** Quando a tensão neste terminal é inferior a $\frac{1}{3}V_{CC}$, a saída toma o valor lógico *HIGH*.
3. **OUTPUT:** Representa a saída do CI, cuja amplitude do sinal é cerca de 1.5 V inferior a V_{CC} .
4. **RESET:** Reinicializa todo o circuito, quando a tensão no terminal é inferior a 0.7 V.
5. **CONTROL VOLTAGE:** Regula a sensibilidade do CI.
6. **TRESHOLD:** Quando a tensão neste terminal é superior a $\frac{2}{3}V_{CC}$, a saída toma o valor lógico *LOW*.
7. **DISCHARGE:** Tem a função de fazer a descarga do condensador.
8. **V_{cc} :** Tensão de alimentação.

O circuito integrado 555 possui três modos de operação, que são: Biestável, Monoestável e Astável.

No modo biestável como o próprio nome indica, a saída fica estável no estado *LOW* e *HIGH*. Por exemplo, quando o terminal *TRIGGER* estiver no estado *LOW*, a resposta na saída será o nível lógico *HIGH*, voltando ao nível *LOW*, através do terminal de *RESET*.

No modo monoestável existe apenas um estado estável, que é normalmente o nível lógico 0, ou seja, após um determinado tempo a saída do CI voltará sempre ao estado *LOW*. Quando o terminal *TRIGGER* vai para nível lógico 0, a saída do 555 vai para o nível lógico *HIGH*, voltando ao nível lógico 0, algum tempo depois. Esta configuração é muito utilizada nos sistemas em que uma luz acende por um determinado tempo, quando alguém é detetado.

Por último, no modo astável não existe nenhum estado estável. Este modo terá uma explicação mais detalhada, uma vez que, foi o modo utilizado no desenvolvimento do projeto.

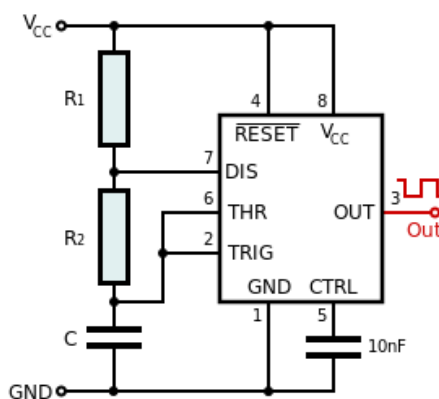


Figura 3.8 – Modo de operação astável do CI 555 [43]

A figura 3.8 ilustra um circuito utilizado no modo de operação astável do 555. Esta configuração é utilizada quando é necessário alternar entre os estados ligar/desligar, como por exemplo piscar LEDs.

Considerando que o condensador C está inicialmente descarregado e $V_C = V_2 = V_6$, o circuito tem o seguinte comportamento [44]:

1. Inicialmente o condensador C começa a carregar através de R_1 e R_2 .
 - Nesta situação, a tensão aos terminais *THR* e *TRIG* é próxima de 0 e portanto inferior a $\frac{1}{3}V_{CC}$, fazendo com que o terminal S do flip-flop seja ativado e consequentemente a saída do 555 toma o nível lógico *HIGH*.
2. Quando o condensador atinge uma tensão superior a $\frac{1}{3}V_{CC}$, o flip-flop fica com 0 em *Set* e *Reset* e portanto o estado mantém-se em *HIGH*
3. O condensador continua a carregar até que a tensão atinge um valor um pouco superior a $\frac{2}{3}V_{CC}$.
 - Quando isto acontece, o comparador ligado ao terminal de *Reset* do flip-flop é ativado, fazendo com que o transistor de descarga seja ativado e a saída *OUT* do 555 mude para o nível lógico *LOW*.
4. Por fim, com o transistor ativo, o condensador descarrega até ficar com uma tensão um pouco inferior a $\frac{1}{3}V_{CC}$.
 - Quando isto acontece, o terminal S do flip-flop é acionado e a saída do 555 muda para o nível alto novamente, o transistor de descarga é desativado e o condensador, começa a carregar novamente

Este ciclo repete-se, dando origem a uma onda quadrada na saída do 555. A resistência R_2 determina o tempo de carga e descarga, enquanto que R_1 influencia apenas o tempo de carga do condensador.

Alguns estudos já efetuados no âmbito dos sensores capacitivos utilizados em têxteis inteligentes, mostram que a utilização de um circuito de aquisição composto por um microcontrolador e um circuito integrado 555 se revela eficaz.

Por exemplo, no artigo *Design and Fabrication of a Capacitance Based Wearable Pressure Sensor Using E-textiles* de 2015, foi utilizado um sensor de pressão baseado em uma configuração de camadas composta pela utilização de neopreno, um material dielétrico compressível colocado entre duas camadas de tecido condutivo [45]. Um circuito de aquisição, utilizando um Arduino e um circuito integrado 555, revelou-se eficaz na obtenção do sinal proveniente do sensor. Outro exemplo é em *Design, development and application of a real-time capacitive sensor for automatically measuring liquid level*, publicado em 2019. Nesse estudo, um sensor capacitivo foi utilizado para a medição do nível de líquido por imersão existente num tanque [46]. O sistema utilizou um CI 555 e um microcontrolador PIC18F4520 para aquisição do sinal e também se revelou eficaz.

3.3.2 Condensador plano

O condensador plano é constituído por duas placas condutoras planas e paralelas entre si com uma dada área A , separadas a uma distância d por um material isolante conhecido como dielétrico. Quando o condensador é submetido a uma tensão começa a carregar, acumulando cargas positivas $+Q$ numa das placas ou armaduras e cargas negativas na outra $-Q$. A eletrização das placas provoca a existência de um campo elétrico E , uniforme na região central, mas não uniforme na região periférica devido ao efeito de bordo, que será desprezado [47]. A figura 3.9 representa o modelo de um condensador plano.

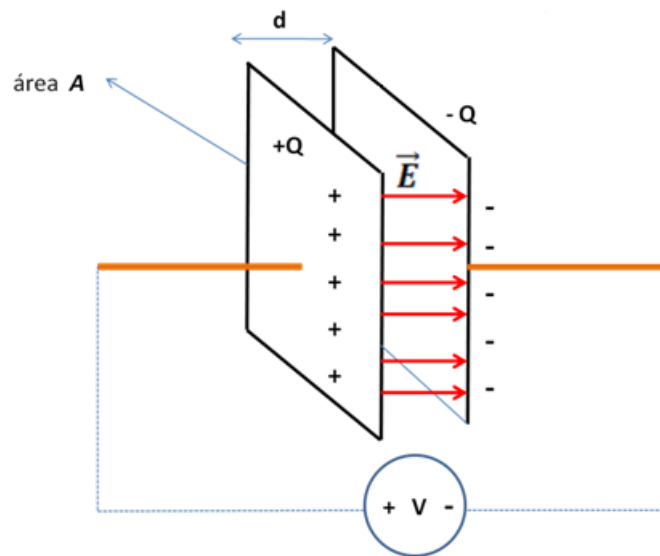


Figura 3.9 – Condensador plano - Figura adaptada de [48]

O campo elétrico E no interior de um condensador plano é dado pela seguinte fórmula:

$$E = \frac{V}{d} \quad (3.1)$$

sendo E o campo elétrico estabelecido, V a diferença de potencial e d a distância entre as placas condutoras.

Quando se insere um material isolante (dielétrico) entre os condutores de um condensador carregado, são formados dipolos que polarizam o material e resultam num campo elétrico induzido no interior que se opõe ao campo elétrico original. Isso provoca uma diminuição na intensidade do campo elétrico entre as placas, por outras palavras diminui a diferença de potencial entre as placas, como se pode comprovar através da equação 3.2.

$$C = \frac{Q}{V} \quad (3.2)$$

onde C é a capacitância, Q a carga elétrica no condensador e V a diferença de potencial.

A figura 3.10 pretende ilustrar isso mesmo.

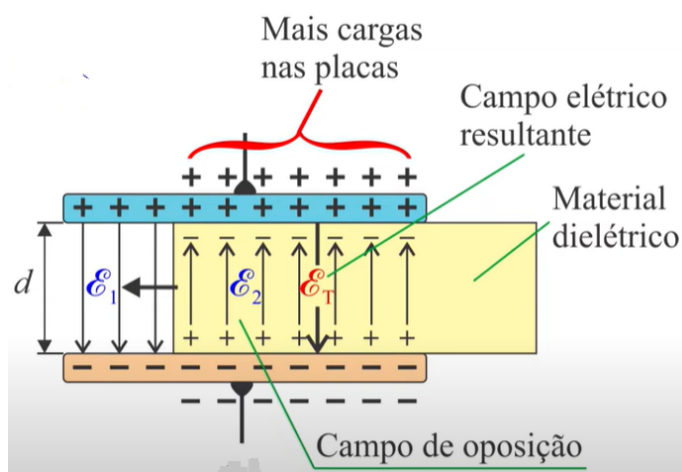


Figura 3.10 – Campo elétrico no dielétrico [49]

O aumento da capacidade do condensador com a introdução de um dielétrico depende da natureza do dielétrico, que é caracterizada pela sua permissividade elétrica [50] e relaciona-se com a capacitância através da fórmula:

$$C = \frac{\epsilon_r \cdot \epsilon_0 \cdot A}{d} \quad [51], \quad (3.3)$$

onde C é a capacitância, ϵ_r é a permissividade relativa do material entre as placas, ϵ_0 é a permissividade do vácuo, A é a área das placas, e d é a distância entre as placas.

A tabela 3.2 contém a permissividade relativa, um valor adimensional, de alguns materiais:

Tabela 3.2 – Permissividade relativa de alguns materiais [52]

Material	ϵ_r
Ar (seco)	1.00059
Mylar	3.2
Borracha de neoprene	6.7
Nylon	3.4
Papel	3.7
Papel impregnado com parafina	3.5
Poliestireno	2.56
Porcelana	6
Teflon	2.1
Vácuo	1.00000
Água	80

O vácuo é o elemento com a menor permissividade relativa, ou seja, é o material com a menor capacidade de alterar ou influenciar um campo elétrico aplicado, sendo usado como referência para comparar as propriedades dielétricas dos outros materiais.

3.4 Microcontroladores

Um microcontrolador é um circuito integrado, projetado para controlar tarefas específicas num sistema embarcado. Tipicamente inclui um processador, memória e periféricos de entrada/saída (E/S), assemelhando-se a um pequeno computador.

Inicialmente as funcionalidades de um microcontrolador resumiam-se a periféricos de entrada e saída (I/O) e foram agregando a cada nova versão ou produto, *Random Access Memory* (memória RAM), *Erasable Programmable Read-Only Memory* (memória EPROM), circuito oscilador *clock*, interfaces de comunicação e mais recentemente interfaces de rede como *Ethernet* e *Wi-Fi* [53].

A figura 3.11 representa o diagrama de blocos de um microcontrolador.

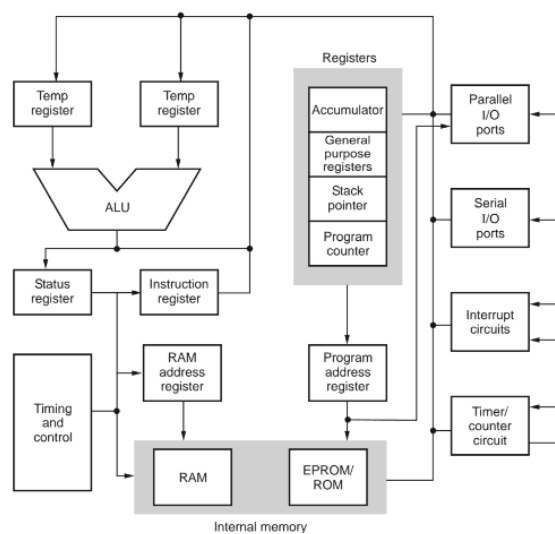


Figura 3.11 – Diagrama de blocos de um microcontrolador [54]

Os elementos principais de um microcontrolador são:

- **Central Processing Unit (CPU):** responsável por controlar o fluxo de informações do microcontrolador. A CPU é composta por duas partes principais: a Arithmetic Logic Unit (ALU) e a Control Unit (CU). A ALU conduz todas as operações aritméticas e lógicas, enquanto a CU auxilia na execução das instruções do processador [55].
- **Memória RAM:** é uma forma de memória volátil, ou seja, armazena informações apenas quando a energia está ligada. Esta, permite que o microcontrolador processe muitos bits de informação simultaneamente.
- **Read-Only Memory (ROM):** é a memória não volátil usada para armazenar o programa ou *firmware* do dispositivo. Ao contrário da memória RAM, a memória ROM mantém o seu conteúdo mesmo quando a energia é desligada.
- **Portas de entrada/saída (E/S):** interfaces através das quais um microcontrolador comunica com o mundo exterior. Essas portas, permitem que o microcontrolador leia sinais de sensores ou dispositivos externos e envie sinais para controlar outros dispositivos.

- **Oscilador Interno:** um componente essencial que gera um sinal de *clock*, determinando a taxa na qual o microcontrolador executa as suas operações [56].
- **Memória EEPROM:** é um tipo de memória não volátil, utilizada para armazenar dados que precisem de ser retidos mesmo que a energia seja desligada.

Existem diversos fabricantes de microcontroladores e sistemas embarcados que se destacam pela diversidade de soluções oferecidas. Cada um contribui com abordagens distintas para atender às diversas necessidades dos programadores e do mundo em geral. O Arduino Uno, baseado no microcontrolador AVR ATmega328P originalmente fabricado pela Atmel, que em 2016 foi comprada pela Microchip Technology é conhecido pela sua abordagem amigável e acessível, tornando-se uma escolha popular em projetos acadêmicos num nível inicial. Por outro lado, o ESP32, desenvolvido pela Espressif Systems, destaca-se pela incorporação de conectividade Wi-Fi e Bluetooth, conferindo-lhe uma posição estratégica em aplicações de Internet das Coisas. A família STM32, da STMicroelectronics, oferece uma ampla gama de microcontroladores de 32 bits, reconhecidos pela sua potência computacional e versatilidade em diversas aplicações industriais e de consumo [57]. Para além destes, saliente-se também o Raspberry Pi, que embora seja uma placa de desenvolvimento mais abrangente, com um sistema completo de computação embarcada, também é digno de menção, especialmente quando se procura uma solução que vai além das capacidades tradicionais de um microcontrolador.

3.4.1 *ESP-WROOM-32*

O ESP32 é um microcontrolador SoC (system-on-chip) de baixo custo e eficiente energeticamente, desenvolvido pela Espressif Systems. Tem conquistado uma posição de destaque no mundo dos sistemas embebidos devido ao seu alto nível de desempenho e versatilidade, sendo ideal para projetos de IoT. Este documento focar-se-á na versão *ESP-WROOM-32*, que foi a primeira placa de desenvolvimento do módulo ESP32 publicamente disponível e criada pela Espressif [58].

Esta, possui uma tensão de operação de 2.2 V a 3.6 V e consome no máximo 260 mA no modo ativo. A placa pode ser alimentada através de um conector USB Type-C, ou através do pino Vin com uma alimentação regulada de 5.0 V.

Quanto aos terminais, existem 30 pinos no total, dos quais 25 são GPIO's, ou seja General Purpose Input/Output, que são pinos que podem ser configurados como entradas/saídas digitais. Dos 25 GPIO's 9 são capacitivos, ou seja, detetam mudanças na capacitância como por exemplo um dedo humano. Existem ainda dois Analogue to Digital Converter (ADC's) de 12 bits, que suportam medições em 15 canais e são responsáveis por converter sinais analógicos em digitais e dois Digital Analog Converter (DAC's) de 8 bits para converter sinais digitais em analógicos [59].

A figura 3.12 apresenta de uma forma abrangente, os terminais da placa de desenvolvimento utilizada no projeto desenvolvido.

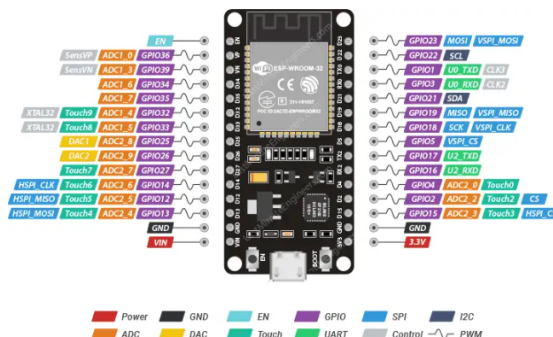


Figura 3.12 – Terminais da placa de desenvolvimento ESP-WROOM-32 [60]

A tabela 3.3 resume as características da placa de desenvolvimento *ESP-WROOM-32*:

Tabela 3.3 – Resumo de Características do ESP-WROOM-32

Característica	Descrição
Microprocessador	Tensilica Xtensa Dual-Core LX6 de 32 bits, com uma frequência de clock entre 80-240 MHz
Consumo de Energia	Ultrabaixo, cerca de 5 μ A no modo <i>Deep-sleep</i>
Conectividade	Wi-Fi 802.11b/g/n, Bluetooth Low Energy (BLE) e 4.2
Periféricos	Diversos integrados, como toque capacitivo, ADCs, DACs, UART, SPI, I2C, PWM
Memória	520 kB de SRAM, 448 kB de ROM e 4 MB de memória Flash
Temperatura de Operação	-40°C - +125°C
Linguagens de Programação	C, C++, MicroPython, entre outras

3.5 Wi-Fi

A sigla "Wi-Fi" não possui um significado específico em termos técnicos. O termo surgiu em 1999 como uma forma mais fácil de descrever o padrão sem fios IEEE 802.11, por intermédio da empresa de marketing *Interbrand*, contratada pela *Wi-Fi Alliance* [61].

O *Wi-Fi* é um protocolo de comunicações sem fio, que utiliza ondas de rádio e entrega pacotes de dados com uma frequência entre 2.4, 5 e 6 GHz, que permite ligar vários dispositivos entre si como: *laptops*, *smartphones*, televisões entre outros, desde que estejam dentro do alcance da rede,

evitando a necessidade de existirem ligações físicas. O IEEE (Institute of Electrical and Electronics Engineers) desenvolveu vários padrões dentro da família 802.11 para comunicações sem fio. Os padrões mais comuns incluem 802.11b, 802.11a, 802.11g, 802.11n, 802.11ac e 802.11ax. Cada padrão introduziu melhorias na velocidade, alcance e eficiência da rede Wi-Fi. Algumas das suas vantagens são a facilidade de instalação, mobilidade e baixo custo de manutenção, no entanto, apresenta algumas desvantagens comparativamente a redes com fios, como menor segurança, velocidade de transmissão mais baixa e maior vulnerabilidade a ameaças externas [62].

A arquitetura 802.11 possui dois modos, o modo de infraestrutura e o modo Ad-Hoc. No modo de infraestrutura, os dispositivos da rede comunicam entre si através de um ponto de acesso que normalmente é o router. Por outro lado, as redes Ad-Hoc não requerem um ponto de acesso centralizado. Em vez disso, os dispositivos comunicam diretamente entre si.

3.5.1 Modos Wi-Fi no ESP32

O ESP32 possui três modos distintos de operação Wi-Fi, são eles o *Station Mode* (STA), o *Access Point Mode* (AP) e o *Access point and Hotspot Mode*. No primeiro, o ESP32 conecta-se a uma rede Wi-Fi através de um router, recebendo um endereço IP exclusivo, tal como os *smartphones* ou qualquer outro dispositivo quando são ligados a uma rede Wi-Fi.

No modo Access Point (AP), o ESP32 cria a sua própria rede local Wi-Fi, com um nome e um IP. Basicamente, atua como um hotspot, no qual podem ser conectados outros dispositivos. Esta funcionalidade é útil, quando por exemplo existem vários dispositivos ESP32 a comunicar entre si [?]. Ao contrário de um router Wi-Fi, o ESP32 não possui uma interface para uma rede com fios, sendo por isso, este modo de operação também denominado Soft Access Point (Soft-AP). É importante dizer que, apenas podem estar conectados 5 dispositivos ao mesmo tempo [63] e que apesar de ser criada uma rede Wi-Fi, não significa que o microcontrolador está conectado à Internet.

Por último, no *Access point and Hotspot Mode*, o ESP32 pode ser configurado simultaneamente como estação Wi-Fi e ponto de acesso.

A figura 3.13 ilustra o modo ponto de acesso.



Figura 3.13 – ESP32: Modo *Access Point* [64]

3.5.2 Wi-Fi Manager no ESP32

Inicialmente, quando se criavam scripts para o ESP32 se conectar a uma rede Wi-Fi, era necessário introduzir manualmente no código, o nome e a palavra passe (SSID e password) da rede que se pretendia aceder. Isso não era muito prático, uma vez que, caso a rede mudasse, era necessário reprogramar o microcontrolador com os novos dados [65].

O *Wi-Fi Manager* tornou possível conectar a placa ESP32 a diferentes pontos de acesso sem a necessidade de estar sempre a introduzir os dados da rede e a fazer upload do script para a placa. Toda a configuração da rede Wi-Fi é feita através de uma interface web.

A figura 3.14 apresenta o layout da página inicial do *Wi-Fi Manager*.



Figura 3.14 – ESP32: Página inicial do Wi-Fi Manager

O *Wi-Fi Manager* segue os seguintes passos:

1. Quando o ESP32 é iniciado, verifica se existem credenciais já gravadas para alguma das redes apresentadas.
2. Se as credenciais Wi-Fi corresponderem à rede Wi-Fi pretendida, é realizado login e estabelecida uma conexão Wi-Fi. Assumindo que na primeira vez que o ESP32 é inicializado, não existem credenciais gravadas para qualquer rede encontrada, a biblioteca WiFi Manager define o ESP32 como um ponto de acesso automaticamente. Um ponto de acesso é onde o ESP32 fornece a sua própria rede Wi-Fi para outros dispositivos se conectarem.
3. De seguida, qualquer dispositivo com Wi-Fi que possua um navegador, pode conectar-se ao ponto de acesso criado, onde constam todas as redes disponíveis e introduzir a password da rede pretendida. Após isso, o ESP32 reinicia.
4. Se a password for inserida corretamente, as informações são guardadas permanentemente no ESP32 e este, entra no modo estação, tentando conectar-se à rede Wi-Fi desejada.
5. Por último, se a conexão for estabelecida com sucesso, significa que o ESP32 tem ligação à Internet. Caso contrário, o microcontrolador entra novamente no modo ponto de acesso para uma nova tentativa de conexão com outra senha.

À semelhança de um *smartphone* ou tablet, existe um menu que apresenta todas as redes Wi-Fi existentes ao alcance do ESP32 e basta colocar a password da rede que se pretende aceder. Essas informações são gravadas na memória não volátil ou EEPROM, permitindo que a informação seja mantida, mesmo quando o microcontrolador é desligado [66].

3.6 Network Time Protocol

O Network Time Protocol (NTP) foi desenvolvido em 1981 por David Mills. É um protocolo utilizado para sincronizar relógios de sistemas computacionais, como servidores, routers entre outros, a partir de referências de tempo confiáveis. A sua principal finalidade é garantir que todos os dispositivos numa rede tenham uma referência de tempo comum e precisa.

A figura 3.15 ilustra a arquitetura geral do protocolo NTP.

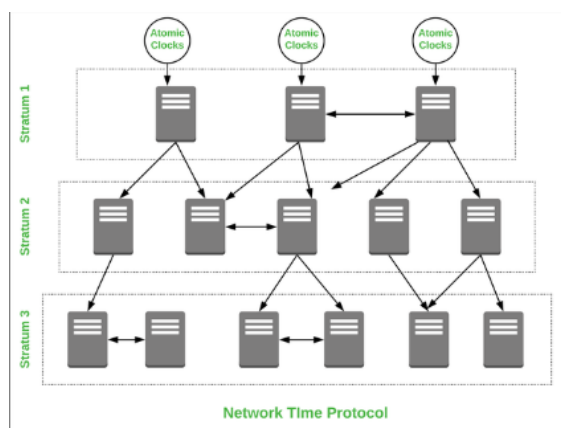


Figura 3.15 – Arquitetura: Network Time Protocol [67]

Os servidores NTP têm uma arquitetura hierárquica, dividida em camadas também designadas por estratos. No topo da hierarquia, no estrato 0 está a referência primária de tempo, que geralmente é um relógio atômico ou um recetor GPS. Os servidores de estrato 2 sincronizam com os servidores de estrato 1, os do estrato 3 com os servidores de estrato 2 e assim sucessivamente, criando uma cadeia de sincronização descendente [68]. Teoricamente, quanto mais perto da raiz, ou seja, do estrato 0, maior a exatidão do tempo.

Concluindo, o protocolo NTP é fundamental para manter a consistência temporal em ambientes computacionais, permitindo que sistemas distribuídos operem de forma coordenada.

3.7 Firebase

O Firebase é uma ferramenta de *Backend-as-a-Service* (BaaS), lançada em abril de 2012. Devido às suas capacidades avançadas e tecnológicas, como a sua base de dados em tempo real, atraiu o interesse da Google e foi adquirido em 2014. A estrutura do Firebase é subdividida em três categorias principais: construção, melhoria e expansão de aplicações [69].

Em termos mais simples, o Firebase é uma plataforma da Google desenvolvida para facilitar a criação de aplicações móveis. Esta plataforma oferece uma variedade de serviços e ferramentas, tais como autenticação de utilizadores com o *Authentication*, armazenamento de dados em tempo real através da *Cloud Firestore*, hospedagem na nuvem e análise de desempenho. Esta abordagem permite que os programadores se concentrem na experiência do utilizador e na lógica da aplicação, eliminando a necessidade de criar software específico para cada aspeto do *backend*. Assim, o Firebase proporciona uma experiência de desenvolvimento simplificada, permitindo uma atenção mais direcionada para a lógica da aplicação e a experiência do utilizador [70].

Para suportar o desenvolvimento, o Firebase disponibiliza Software Development Kits (SDKs) para diversas plataformas, incluindo Android, iOS e web. Estes são conjuntos de ferramentas, fundamentais para os *developers* construírem aplicações Firebase de forma eficiente e eficaz.

3.7.1 Authentication

O *Firebase Authentication* é uma ferramenta do Firebase, que como o próprio nome indica, desempenha um papel crucial no processo de autenticação dos utilizadores. O seu objetivo principal, é oferecer um método de autenticação e identificação seguro, melhorando a experiência de login e facilitando a integração dos utilizadores [71]. Foi desenvolvido pela mesma equipa que criou o Login do Google, o Smart Lock e o Gestor de senhas do Chrome e suporta autenticação através de vários métodos como: contas de e-mail/senha, números de telefone e alguns provedores como o Google, Twitter, Facebook e GitHub [72].

Quando um utilizador se autentica, as suas informações contém um ID único (UID) distinto para cada provedor, que permanece inalterado para o mesmo utilizador. Esse código é utilizado para identificar o utilizador e determinar as áreas do Firebase às quais ele tem acesso.

3.7.2 Cloud firestore

A *Cloud Firestore* é uma base de dados NoSQL dinâmica e hospedada na nuvem, que representa um serviço essencial do Firebase, uma vez que, permite armazenar, sincronizar e consultar dados de aplicações móveis e web em tempo real [73].

As características relevantes a mencionar são:

- Permite o armazenamento e sincronização de dados em tempo real.
- Foi projetado para ser escalável [74], ou seja, tem a capacidade de lidar eficientemente com o crescimento do volume de dados e do número de utilizadores em simultâneo sem que isso afete negativamente o desempenho ou a estabilidade do sistema.

- É flexível, permite adicionar novos campos aos documentos, modificar a estrutura dos dados ou realizar ajustes na organização dos documentos, sem a necessidade de recriar a base de dados.
- Suporte offline, permitindo que os dados sejam acedidos e modificados mesmo sem conexão com a Internet. As alterações feitas offline são armazenadas em cache e, ao se reconectar, são automaticamente sincronizadas.
- Relativamente à segurança, existem regras que permitem controlar operações de leitura e escrita nos ficheiros, o que proporciona um controlo de acesso aos dados com base na identidade dos utilizadores.

Destaca-se o facto de que, ao contrário de uma base de dados SQL, na *Cloud Firestore* não existem tabelas ou linhas, em vez disso, os dados são armazenados em formato JSON e organizados por coleções e documentos [75]. As coleções funcionam como estruturas organizacionais dentro das quais existem documentos que contêm campos com informação.

Esta estrutura está ilustrada na figura 3.16.

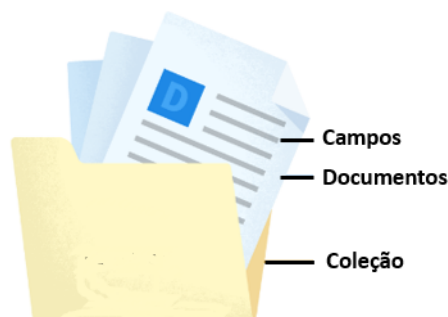


Figura 3.16 – Estrutura dos dados na *Cloud Firestore* - Figura adaptada de [76]

A utilização em conjunto do *Firebase Authentication* e da *Cloud Firestore*, permite o desenvolvimento de apps de forma simplificada, fazendo com que, várias empresas conhecidas como *The New York Times*, *duolingo*, *The Economist*, entre outras utilizem o Firebase para implementar as suas próprias aplicações.

3.8 Flutter

O Flutter é uma *framework* de código aberto desenvolvida pela Google em 2015. Destaca-se como uma poderosa ferramenta para o desenvolvimento de aplicações móveis nativas, pois permite a criação de aplicações para uma variedade de plataformas, incluindo iOS, Android, Web, Windows, MacOS e Linux, através de um único código fonte, o que simplifica o desenvolvimento

de aplicações multiplataforma [77]. Os elementos fundamentais da *framework* são os widgets, responsáveis por moldar a interface do utilizador e incluem desde simples botões até *layouts* complexos. Existem Widgets de *layout*, que posicionam outros widgets, e de interface, que representam elementos interativos, como botões e demais elementos com os quais o utilizador interage.

De acordo com Eric Windmill, o Flutter proporciona todos os elementos essenciais para a construção de aplicações, incluindo mecanismos de renderização, componentes de interface do utilizador, estruturas de teste, entre outros recursos [78].

As aplicações desenvolvidas no Flutter são escritas na linguagem de programação Dart, e ambientes de desenvolvimento robustos como o VSCode ou o Android Studio são amplamente utilizados para trabalhar eficientemente com esta framework, que já conquistou a confiança de grandes empresas, incluindo a BMW, eBay e Toyota.

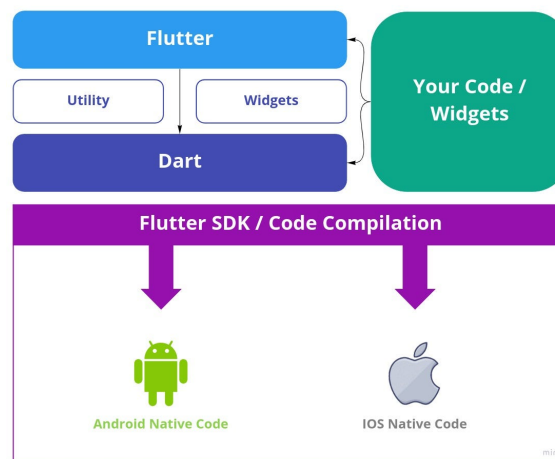


Figura 3.17 – Arquitetura geral [79]

3.8.1 Linguagem de programação Dart

O Dart é uma linguagem de programação orientada a objetos, apresentada pela Google na conferência GOTO em 2011. Inicialmente, propunha-se como uma alternativa ao JavaScript, uma linguagem amplamente consolidada, o que levou a alguma hesitação por parte dos programadores. No entanto, graças ao surgimento do Flutter o uso da linguagem Dart ganhou um impulso significativo [80].

Possui algumas características que a tornam tão importante quando utilizado em conjunto com o Flutter, como a capacidade de compilação just-in-time (JIT) e ahead-of-time (AOT). A compilação JIT permite a existência do *hot reload* ou carregamento dinâmico, possibilitando a visualização instantânea na aplicação, dos efeitos das alterações no código durante o desenvolvimento. Por sua vez, a compilação AOT transforma o Dart em código nativo eficiente [81]. Para além disso, a linguagem adota uma sintaxe baseada em C, o que facilita a transição para os programadores familiarizados com linguagens orientadas a objetos.

Em síntese, a utilização do Flutter e Dart no desenvolvimento de aplicações móveis destaca-se pela abordagem multiplataforma, o que permite a criação de aplicações para dispositivos iOS e Android através de um único código fonte.

Capítulo 4

Arquitetura

Neste capítulo apresenta-se o sistema desenvolvido ao longo do período da dissertação, assim como, as tecnologias que foram utilizadas. Inicia-se com uma visão global da arquitetura, destacando os principais requisitos e funcionalidades do sistema e depois são abordadas características técnicas do sensor, métodos de aquisição e condicionamento do sinal e desenvolvimento da interface IoT.

4.1 Visão Global

Como já referido anteriormente, a empresa *Invacare* forneceu o lençol *TEXIBLE Wisbi* e propôs a criação de um sistema de aquisição do sinal e processamento da informação, para que não seja necessária a utilização do transmissor originalmente fornecido pela empresa austríaca. Para alcançar esse objetivo foram estabelecidos os seguintes requisitos funcionais:

- Deve ser capaz de detetar a presença/ausência de pessoas.
- Deve ser capaz de detetar o nível de humidade na cama.
- Deve providenciar o estado do lençol em tempo real.
- Deve ser capaz de criar alertas em tempo real em caso de saída não autorizada, ou níveis de humidade elevados.
- Desenvolvimento de uma aplicação para *Smartphone*, intuitiva para o cuidador.
- O sistema deve incluir uma opção de calibração manual por parte dos utilizadores, assegurando precisão nas medições e adaptabilidade a diferentes contextos de utilização, nomeadamente diferentes colchões.

Quanto aos requisitos não funcionais:

- Não utilizar sensores externos de medição de presença ou humidade para além do próprio lençol.

- O sistema deve assegurar a atualização dos seus estados e, se necessário, emitir um alerta num tempo de resposta inferior a 5 segundos.
- Implementação de medidas de segurança para proteger os dados do utilizador e garantir a integridade do sistema.

Considerando todos os requisitos anteriormente mencionados e tendo em conta que o lençol, com características a serem detalhadas posteriormente, atua como o sensor para a deteção de presença e humidade, propõe-se a arquitetura ilustrada na figura 4.1:

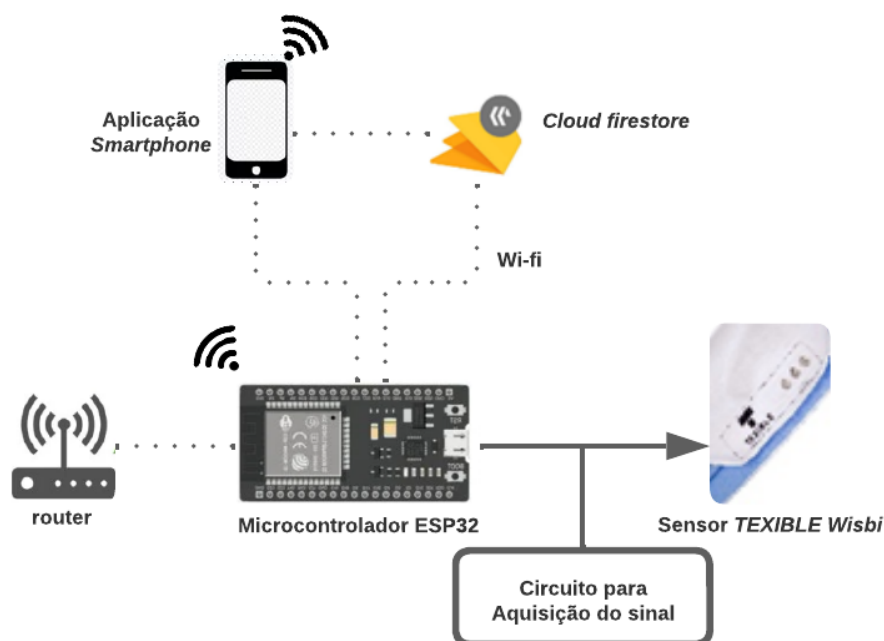


Figura 4.1 – Arquitetura global do sistema

Da figura, reitera-se que é necessário que o sistema inclua:

1. Um circuito para a aquisição da informação proveniente do lençol, projetado para gerar um sinal cujas características variem em resposta à presença ou ausência de pessoas na cama, assim como aos níveis de humidade.
2. Um dispositivo computacional central, responsável por processar e gerir as informações provenientes do circuito de aquisição, este tem um papel crucial, uma vez que, após a análise dos dados recebidos envia para um armazenamento em nuvem, o estado do lençol.
3. Um armazenamento em nuvem, que irá gerir a autenticação dos utilizadores e registar os dados provenientes da unidade computacional central, para no fundo servir de elo de comunicação entre o microcontrolador e a aplicação para *Smartphone*.

4. Uma interface de utilizador, que verifica em tempo real as informações relativas à presença e humidade no armazenamento em nuvem e as fornece de forma intuitiva ao cuidador, através de um *dashboard* na aplicação.

4.2 Sensor *TEXTIBLE Wisbi*

O lençol *TEXTIBLE Wisbi* é o ponto de contacto entre a cama e o paciente, sendo colocado por baixo do lençol inferior. Tem incorporados filamentos internos impercetíveis, que ligam numa das suas extremidades a 3 terminais estrategicamente posicionados, que serão conectados ao sistema embebido desenvolvido para transmitir o sinal.

A figura 4.2 ilustra a parte traseira do lençol, ou seja, aquela que fica em contacto com o colchão e na qual é possível observarem-se os filamentos incorporados no têxtil.

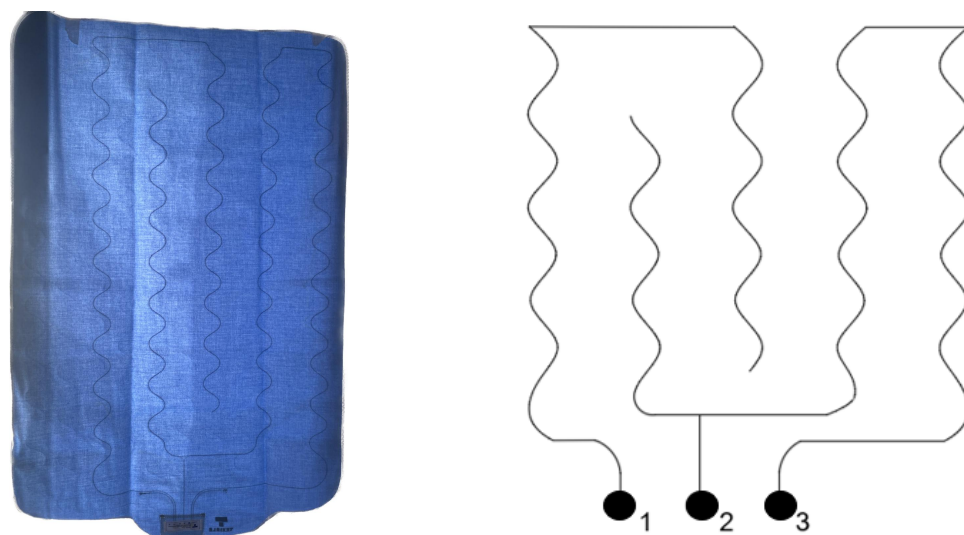


Figura 4.2 – Filamentos incorporados no lençol - Perspetiva traseira

As figuras acima têm como objetivo representar a distribuição dos filamentos ao longo do lençol. À esquerda, ilustra-se uma representação real, enquanto que, à direita se observa uma representação esquemática. Na figura da direita, é possível notar que do terminal 1 emerge um filamento que não se conecta a nada na extremidade oposta, indicando assim a presença de um circuito aberto entre os terminais 1 e 2. Por outro lado, os terminais 2 e 3 estabelecem um circuito fechado e o circuito comporta-se como uma resistência. Optou-se por não aprofundar essa característica, uma vez que, após diversas tentativas para identificar um padrão na variação da resistência em resposta à presença de pessoas, essa abordagem revelou-se ineficaz, pois não foi possível identificar um padrão consistente na variação de resistência com a presença de pessoas.

Nesta configuração, entre os terminais 1 e 2, o comportamento do circuito assemelha-se ao de um condensador plano, onde os filamentos seriam as placas condutoras e o tecido seria o dielétrico.

Visualizando o lençol como um condensador plano, é importante ter em consideração a fórmula da capacitância apresentada em 4.1, que foi já explicada no capítulo anterior.

$$C = \frac{\epsilon_r \cdot \epsilon_0 \cdot A}{d} \quad [51], \quad (4.1)$$

Pode concluir-se, que a capacitância é afetada pela permissividade do dielétrico, pela distância entre as placas e ainda pela área das mesmas.

4.2.1 Comportamento do sistema no estado seco

Quando não existe nenhum doente sobre o lençol, a capacitância depende apenas do tecido do lençol. No geral, os têxteis apresentam uma constante dielétrica muito baixa, pois são materiais muito porosos. Como exemplo, a tabela 4.1 mostra as propriedades dielétricas de tecidos normais, não condutores.

Tabela 4.1 – Permissividade relativa de alguns tecidos não condutores [82]

Tecido não condutor	ϵ_r
Cordura®	1,90
Algodão	1,60
100% Poliéster	1,90
Tecido Quartzel®	1,95
Cordura/ Lycra®	1,50

Destaque para o algodão que constitui 67% do tecido do lençol e o poliéster que constitui 30%. O poliuretano, apesar de não estar representado na tabela, constitui os restantes 3%.

Pode-se então concluir, que a capacitância no estado seco e ausente se deve sobretudo à permissividade do algodão, a qual é 1.60.

Por outro lado, quando existe um acamado, a constante dielétrica do tecido é alterada, pois a flexão e o alongamento do tecido influencia a sua permissividade e espessura [82]. Por outras palavras, quando um corpo está sobre o lençol, existe uma distensão e por outro lado, quando se levanta existe uma flexão no tecido. Isto, de acordo com Rita *et al* [82] influencia a permissividade, elevando-a.

Através da equação anterior conclui-se portanto, que se a permissividade elétrica aumenta, a capacitância também, pois são diretamente proporcionais.

4.2.2 Comportamento do sistema no estado húmido

A capacidade das fibras do tecido para absorver humidade, deve ser considerada na caracterização do comportamento dielétrico dos têxteis, uma vez que, esta altera as propriedades dos tecidos.

A água tem uma constante dielétrica de cerca de 80 como já apresentado na tabela 3.2 do capítulo 3. Embora este valor varie, pois depende da temperatura e salinidade, a água possui uma constante dielétrica muito maior e mais estável do que os têxteis, cuja constante dielétrica é normalmente entre 1 e 2, devido à sua alta porosidade. Portanto, quando a água é absorvida pelas

fibras têxteis, ela altera as propriedades eletromagnéticas do tecido, aumentando a sua constante dielétrica [82]. Dado que, a permissividade relativa da água é 80, um valor muito superior ao têxtil, é normal que quando o lençol estiver húmido a capacitância tome valores bastante superiores comparativamente ao estado do lençol seco.

É ainda importante dizer que, entre os terminais 1 e 2 existe uma resistência da ordem dos megaohms quando o lençol está seco. Quando molhado, devido ao aumento da condutividade do tecido, a resistência diminui bastante, sendo da ordem dos ohms.

Concluindo, apesar de não existir acesso a dados técnicos do sensor, foi efetuada uma extensa pesquisa e vários testes experimentais, de forma a perceber como poderia ser o funcionamento do lençol. Foram realizadas um conjunto inicial de medidas de resistência, capacitância e indutância nos diferentes terminais para caracterização do sensor. Com base nesses testes, foi implementado um circuito para medir as variações, como é explicado na secção seguinte.

4.3 Circuito para aquisição do sinal

Para adquirir o sinal do lençol, implementou-se o circuito ilustrado na figura 4.3.

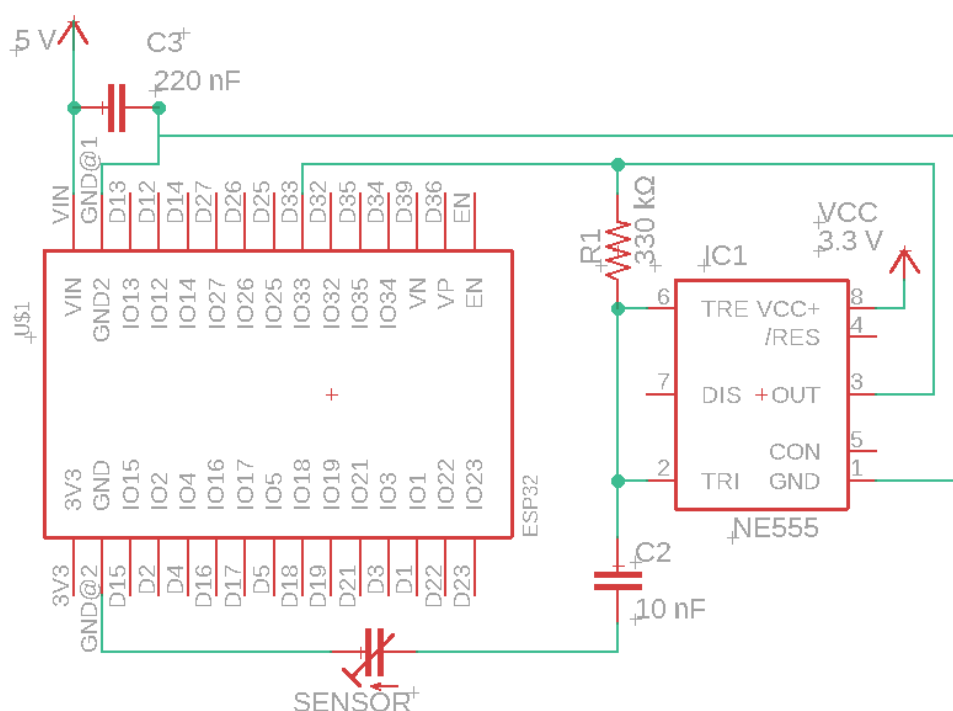


Figura 4.3 – Circuito de aquisição do sinal

Este circuito é dividido em dois blocos distintos: o andar oscilador, onde se introduz o Circuito Integrado NE555 e o andar de processamento da informação, executado pelo microcontrolador ESP32.

Como mencionado anteriormente, os terminais 1 e 2 do lençol desempenham o papel de um condensador plano e, as variações decorrentes da mudança na ocupação do lençol e da humidade, alteram a sua capacitância. Após diversas tentativas com outras técnicas menos eficazes, constatou-se que um circuito oscilador é capaz de atender aos objetivos desejados, traduzindo as alterações físicas em sinais elétricos mensuráveis. Para esse fim, utilizou-se o CI NE555, conhecido pela sua fiabilidade.

Analisando o circuito, o condensador C3 de 220 nF, colocado entre o Vin e o GND é um condensador de desacoplamento, ou seja, tem a função de estabilizar a tensão de entrada. No contexto do projeto, configurou-se o NE555 no modo astável, onde o pino 2 (*Trigger*) está ligado ao pino 6 (*Threshold*). Nessa configuração, a frequência de oscilação do CI é determinada pela capacitância do lençol e pela resistência R1. Com uma resistência de 330 kΩ e um condensador de 10 nF em série com o lençol, a fórmula 4.2 deduzida do *datasheet* do NE555, define a capacitância do circuito.

$$C = \frac{1.44}{(R_A + 2R_B)f} \quad (4.2)$$

onde C é a capacitância, R_A e R_B são resistências e f é a frequência de oscilação.

Observe-se que quanto maior a capacitância, menor a frequência de oscilação e vice-versa. Além disso, o condensador em série com o lençol limita a capacitância máxima do oscilador, atingida quando o lençol está molhado.

Apesar de existirem quatro estados possíveis, os quais são: Lençol desocupado e seco, lençol ocupado e seco, lençol ocupado e húmido e lençol desocupado e húmido, apenas se vai considerar os três primeiros. O estado desocupado e húmido não é relevante, uma vez que a finalidade do dispositivo é avisar o cuidador uma de duas situações:

1. O paciente ausentou-se.
2. O paciente está molhado.

Em qualquer uma delas o cuidador deve-se deslocar de imediato até ao paciente. Assim sendo, o estado desocupado e húmido não é relevante, pois o facto de estar molhado já é condição suficiente para alertar o cuidador. Uma vez o sensor molhado, este terá que ser seco, ou substituído por outro, antes de poder ser colocado na cama novamente.

Em repouso, sem a presença de uma pessoa, a capacitância é menor, resultando em uma frequência de oscilação maior. Quando uma pessoa se deita, a capacitância aumenta, diminuindo a frequência de oscilação. Do mesmo modo, a presença de humidade altera as propriedades dielétricas do lençol aumentando consideravelmente a capacitância e, consequentemente, diminuindo a frequência de oscilação.

É crucial notar que, no contexto deste projeto, os valores reais da capacitância não são relevantes, uma vez que, apenas interessam definir os limites que permitem detetar a transição do estado do lençol desocupado para ocupado e, quando ocupado, a transição do estado seco para húmido. O

limite entre ausência e presença, é estabelecido aquando da calibração do sistema, como explicado no capítulo 6.

Por fim, a saída do CI é conectada à *General Purpose Input/Output* (GPIO) 33 do ESP32. Os dados recolhidos são então interpretados pelo microcontrolador, que determina o estado do sistema com base na presença ou ausência de pessoas e no nível de humidade. O método específico pelo qual os dados são interpretados e processados, desde a aquisição da capacitância até à definição do estado correspondente, será detalhado na secção 4.4.

4.4 ESP32 - Processamento e tratamento dos dados

Nesta secção, serão abordados os aspetos relacionados exclusivamente ao microcontrolador ESP32, destacando-se a forma como é determinada a capacitância, a definição dos estados e por fim apresenta-se um fluxograma geral do código. O foco será estritamente no processamento interno de informações pelo ESP32, proporcionando uma visão detalhada da interpretação dos sinais do sensor capacitivo e da lógica que orienta a transição entre os diferentes estados do lençol inteligente. Por enquanto, excluem-se os aspetos referentes à comunicação com a aplicação externa.

4.4.1 Determinação da capacitância

Após a análise do funcionamento do lençol *TEXIBLE Wisbi* e do circuito de aquisição do sinal do sensor nas secções anteriores, esta subsecção explica o processamento desses dados.

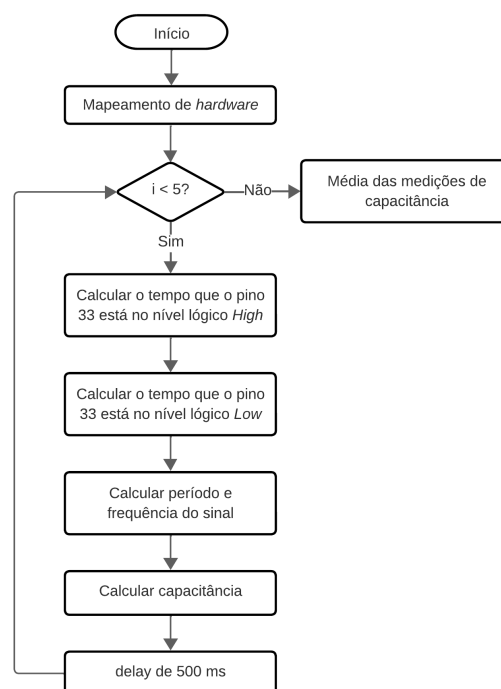


Figura 4.4 – Função para a leitura da capacitância pelo ESP32

A figura 4.4 ilustra a função responsável pela leitura da capacitância, processando o sinal proveniente da GPIO33. Inicialmente, realiza-se o mapeamento do *hardware*, ou seja, declara-se a variável `freq_osc` associada ao GPIO 33, definido como um pino de entrada que irá receber o sinal proveniente do CI 555, caracterizado por uma onda quadrada. Essa forma de onda é crucial para medir a frequência de oscilação do sinal, a qual está diretamente relacionada com a capacitância do sensor. Para além disso, são definidas as constantes auxiliares RA e RB, como 0 e 330 k Ω , respetivamente. Neste circuito, a resistência RA não existe fisicamente, mas uma vez que a fórmula 4.2 contém duas resistências, optou-se por representá-la no código pelo valor 0.

A medição da capacitância é realizada pelo microcontrolador através de um método amostral, envolvendo cinco medições distintas com um intervalo de 500 milissegundos entre cada uma. Essa abordagem visa obter uma média estável e representativa da capacitância do sensor, contribuindo para a precisão das leituras.

A função tem a assinatura: `cap(float resA, float resB)`, cujos parâmetros de entrada são as resistências RA e RB e executa as seguintes operações:

1. Utiliza a função `pulseIn()` para medir o tempo em μ s, no qual o sinal está em nível lógico alto e armazena esse valor em `H_us`.
2. De forma similar, mede o tempo no qual o sinal está em nível lógico baixo e armazena-o em `L_us`.
3. Calcula o período total `T_us` como a soma de `H_us` e `L_us`.
4. A partir do período, obtém-se a frequência do sinal, pois como se sabe, a frequência é o inverso do período.
5. Finalmente, utiliza-se a frequência para calcular a capacitância em cada iteração pela fórmula 4.2, acumulando o resultado numa variável total.

Após as iterações, a função retorna a média das medições de capacitância, dividindo o total acumulado pelo número de amostras que são 5. A leitura de capacitância é crucial para a determinação dos estados do lençol, uma vez que, permite ao microcontrolador interpretar se há presença ou ausência de pessoas e o nível de humidade, com base nas variações de capacitância detetadas.

A definição dos estados será explicada na subsecção seguinte.

4.4.2 Definição dos estados

A utilização do lençol inteligente segue uma sequência lógica que começa com o lençol inicialmente seco e sem ninguém. Posteriormente, um paciente deita-se, e o estado muda para seco com alguém presente. Finalmente, se algum fluido for derramado, como urina ou suor, o lençol passa a estar húmido com alguém presente. Neste último estado, é crucial que um cuidador intervenha rapidamente para trocar e lavar o lençol, minimizando desconfortos ou problemas maiores para o paciente, portanto, a transição para um estado de lençol húmido e desocupado não é considerada

relevante para o funcionamento do sistema. Considera-se que neste caso, o lençol é desconectado para ser submetido a uma lavagem.

Para o adequado funcionamento do sistema e de forma a que este cumpra todos os objetivos aos quais se propõe, são essenciais três estados principais: lençol seco e ausente, lençol seco e presente, e lençol húmido e presente. Na figura 4.5 está ilustrado o diagrama de máquina de estados do lençol.

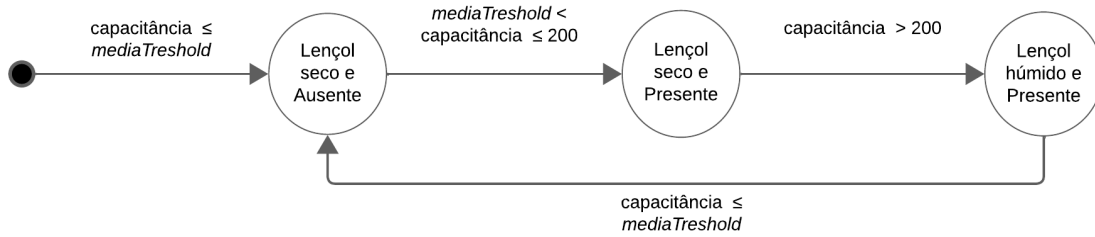


Figura 4.5 – Diagrama de máquina de estados do lençol

As transições entre os estados são controladas por variáveis definidas no código do microcontrolador, as quais são: a *capacitância* que é o valor medido em tempo real da capacitância nos terminais do sensor, a *absenceThreshold* que é o valor de referência da capacitância quando o lençol está seco e desocupado, obtido aquando da calibração inicial do sistema e a *presenceThreshold* é o valor de referência para quando o lençol está seco e ocupado, também determinado durante a calibração. Detalhes adicionais sobre a calibração serão abordados no capítulo 5 e 6.

Como explica em detalhe o capítulo 6, o limite foi definido através do cálculo da média aritmética entre as referências, quando o lençol está seco desocupado e ocupado, de acordo com a fórmula:

$$\text{mediaThreshold} = \frac{\text{absenceThreshold} + \text{presenceThreshold}}{2} - 5 \quad (4.3)$$

No estado inicial tem-se que para capacitâncias inferiores ou iguais a *mediaThreshold*, o lençol é considerado seco e desocupado. Quando alguém se deita, a capacitância aumenta e portanto estabeleceu-se que, quando esta é superior a *mediaThreshold* e inferior ou igual a 200 pF, o lençol se encontra seco e ocupado. A constante 200 pF é o limite estabelecido para considerar a fronteira entre o estado seco e molhado. Por último, quando o sensor é molhado, a capacitância aumenta de forma exponencial, pelo que se definiu que quando esta for superior a 200 pF, o lençol apresenta-se húmido e ocupado. Como dito anteriormente, a passagem do estado húmido ocupado, para húmido desocupado, não é relevante, uma vez que, se depreende que neste caso o lençol foi desligado do sistema para ser submetido a lavagem.

A constante 200 pF foi estabelecida com base na realização de testes empíricos, nos quais ficou evidente que seria um limite bem definido para as transições entre o estado seco e húmido. Por exemplo, considere-se o seguinte cenário hipotético: a calibração inicial, determina que o valor de referência para o estado seco e desocupado é de 60 pF e para o estado seco e ocupado é de 100

pF. A detecção de ausência ocorre quando o microcontrolador lê valores de capacitância inferiores ou iguais a 75 pF, de acordo com a expressão 4.3. Esta fórmula proporciona uma margem de segurança de 15 pF relativamente à referência de ausência medida, pois $75 \text{ pF} - 60 \text{ pF} = 15 \text{ pF}$.

Este intervalo considera as oscilações normais do sistema, sendo suficientemente largo para garantir fiabilidade. A presença, por sua vez, é detetada quando o valor é superior a 75 pF, e inferior ou igual a 200 pF. Acima de 200 pF, o sistema interpreta a condição como húmido e ocupado. Esta abordagem assegura uma transição suave entre os estados, minimizando o risco de interpretações erradas.

4.4.3 Fluxograma do funcionamento do sistema

O fluxograma representado na figura 4.6, proporciona uma visão abrangente das operações realizadas pelo microcontrolador.

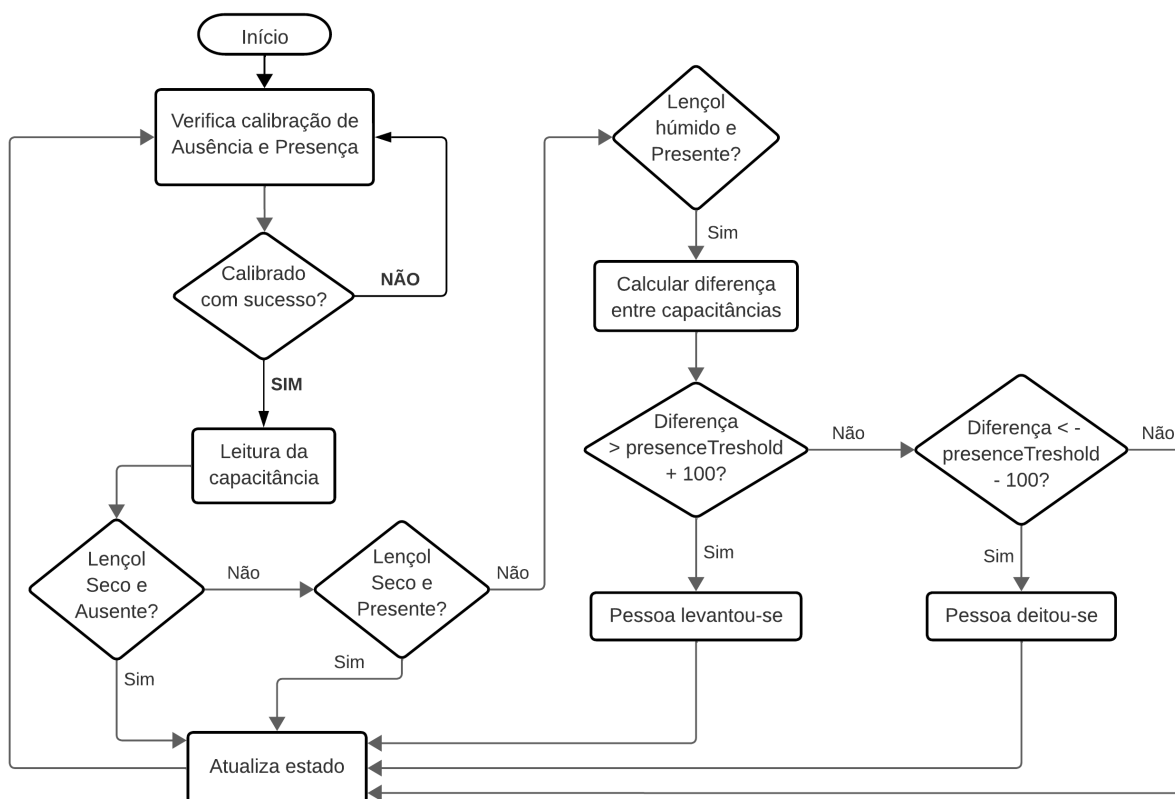


Figura 4.6 – Fluxograma geral do sistema desenvolvido

O sistema apenas funciona mediante uma calibração prévia bem-sucedida. Inicialmente, o microcontrolador executa uma verificação das calibrações de ausência e presença e, somente após a conclusão satisfatória desta etapa é que o sistema avança para a medição da capacitância em tempo real nos terminais do sensor. Com base no valor obtido, o sistema avalia primeiramente

se o lençol está seco e desocupado e, em caso positivo, o estado é atualizado, caso contrário, é investigada a possibilidade de o lençol se encontrar seco, mas ocupado, atualizando o estado se necessário. Caso nenhum dos estados anteriores se verifique, o sistema avança para a condição de o lençol estar húmido e ocupado. Assume-se que o lençol só se torna húmido na presença do utilizador, devido a eventos como perda de urina ou suor, o que indica a existência de alguém sobre ele.

Finalmente, se o estado húmido e ocupado for confirmado, o sistema calcula a variação da capacitância em relação à medição anterior, através da fórmula: $diferença = capacitância\ anterior - capacitância\ atual$. Uma variação positiva significativa, definida como superior a $presenceThreshold + 100$, indica que a pessoa se levantou, isto acontece, porque quando o utilizador se levanta do lençol existe uma redução abrupta da capacitância, fazendo com que a diferença seja positiva, pois neste caso, tem-se que o valor atual da capacitância é menor que o valor anterior. Por outro lado, se alguém se deitar no lençol já húmido, a capacitância aumentará significativamente, resultando numa variação negativa. Para evitar falsas deteções atribuídas a flutuações normais de medição, estabeleceu-se que a deteção de alguém se deitar sobre o lençol molhado, ocorre quando a variação da capacitância é inferior a $-presenceThreshold - 100$. Posteriormente o estado é atualizado e o processo é repetido, garantindo uma monitorização contínua.

O fluxograma dá uma ideia geral de como o sistema está concebido, mas nas secções seguintes mais detalhes serão fornecidos.

4.5 *Firestore*

O *Firestore*, em particular a *Cloud Firestore* foi a plataforma escolhida para servir de interface de comunicação entre o microcontrolador e a aplicação para *Smartphone*. Esta, desempenha um papel crucial na transmissão e gestão de dados, assegurando o correto funcionamento do sistema desenvolvido.

É relevante destacar que, o *Firestore* tem vários planos de adesão, sendo que foi adotado o *Plano Spark*, o plano gratuito, que numa fase inicial do projeto cumpre com todas as exigências requeridas. Este plano, apesar de gratuito, oferece bastantes recursos, sendo que os relevantes para o sistema desenvolvido são [83]:

- Capacidade de ter até 50 mil utilizadores diferentes, autenticados.
- 1 GB de dados armazenados.
- 20 mil gravações em documentos, por dia.
- 50 mil leituras por dia.

A escolha pelo plano gratuito foi motivada pela necessidade de uma solução eficaz e economicamente viável.

4.5.1 Cloud Firestore

Relativamente ao sistema desenvolvido, o microcontrolador envia os dados para a *Cloud Firestore*, e a aplicação concebida para *Smartphone* lê os valores correspondentes. Por outras palavras, a *Cloud* é o elo de comunicação, no qual o ESP32 e a aplicação efetuam operações de leitura e escrita de dados. Destaca-se o facto de que, ao contrário de uma base de dados SQL, na *Cloud Firestore* não existem tabelas ou linhas, em vez disso, os dados são armazenados em documentos, e estes, por sua vez, são organizados em coleções [75].

Essencialmente, as coleções funcionam como estruturas organizacionais, dentro das quais existem documentos, os quais contêm cada um, campos com informação.

Para este projeto, a arquitetura definida está representada na figura 4.7.

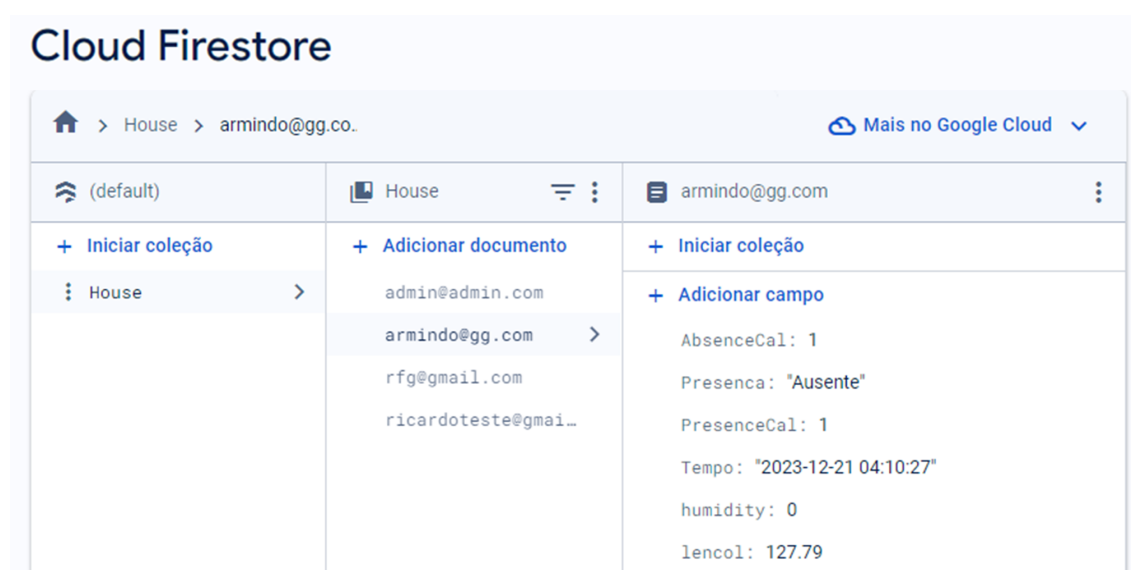


Figura 4.7 – Arquitetura da base de dados do sistema desenvolvido [Fonte: Cloud Firestore]

Existe uma coleção principal denominada *House*, composta por documentos individuais identificados pelos e-mails dos utilizadores registados.

Cada documento contém os seguintes campos:

- *AbsenceCal*: *flag* que indica se o lençol está calibrado relativamente à ausência.
- *Presenca*: uma string que indica o estado *Ausente* ou *Presente* do paciente.
- *PresenceCal*: *flag* que indica o estado da calibração de presença.
- *Tempo*: variável que indica a data e hora atual.
- *humidity*: armazena em percentagem, o valor da humidade atual.
- *lencol*: armazena o valor em tempo real da capacitância do lençol.

É de realçar a importância da utilização dos endereços de e-mail como identificadores únicos dos documentos no *Firebase* (títulos dos documentos). Esta estratégia é crucial para distinguir claramente os utilizadores, assegurando uma comunicação unívoca e segura entre cada microcontrolador ESP32 e a plataforma *Firebase*. Ao vincular cada conjunto de dados ao e-mail do respetivo utilizador, garante-se que a informação é acessível exclusivamente pelo utilizador correspondente, ficando esta depois disponível no *dashboard* da aplicação.

4.5.2 Escrita de dados na *Cloud Firestore*

Após uma autenticação de utilizador bem sucedida, é necessário disponibilizar os dados de presença e humidade no *Dashboard* da aplicação, para isso, estes necessitam de ser escritos na *Cloud Firestore*. A presente subsecção, detalha os passos inerentes a este processo, que é fundamental para o bom funcionamento do sistema.

Todos os dados necessários para o correto funcionamento da aplicação, são escritos pelo microcontrolador na *Cloud Firestore*. Cada campo do documento, é controlado por uma função específica, responsável pela sua atualização ou criação.

A sequência de passos para a criação de documentos e atualização de variáveis, no *Firebase*, estão representados na figura 4.8.

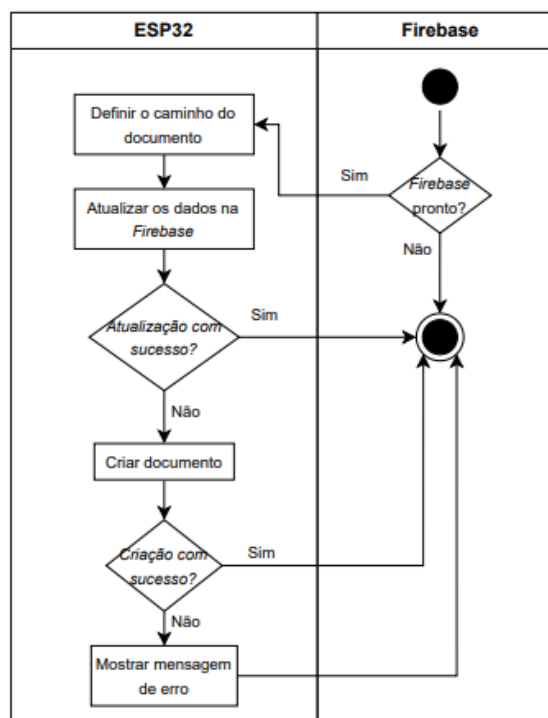


Figura 4.8 – Diagrama de atividades do processo de escrita de dados no *Firebase*

O diagrama ilustra o fluxo lógico das operações de atualização ou criação de registos no *Firebase*. Esse processo é análogo em todas as funções envolvidas, diferenciando-se apenas pelo tipo de dado que recebem e escrevem na base de dados.

Para entender melhor esta implementação, a figura 4.9 apresenta o fluxograma da função *firestorePresenceUpdate(String Body)*, responsável por definir a presença ou ausência do paciente, baseada no estado do sistema.

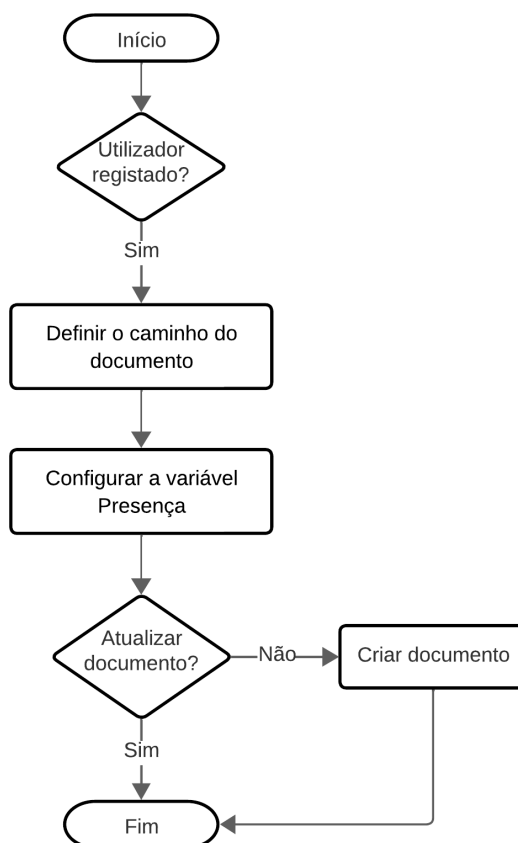


Figura 4.9 – Fluxograma da função responsável por atualizar o estado de presença

A seguir, enfatizam-se os pontos-chave:

1. Primeiramente, é verificada se a conexão com o *Firebase* está ativa através do método *Firebase.ready()*, que retornará erro, caso por exemplo a autenticação não tenha sido realizada com sucesso. Este passo é crucial para garantir que a operação de escrita na base de dados só aconteça se existir uma conexão estável.
2. Um objeto da classe *FirebaseJson*, chamado *content*, é instanciado para conter os dados a serem enviados. Neste caso, os dados são provenientes da variável *Body*, o parâmetro de entrada da função. É uma string, que dependendo do valor da capacitância, será definida como *Ausente* ou *Presente*. Esta atribuição é realizada pela linha de código *content.set("fields/Presenca/stringValue", Body)*.
3. O caminho do documento na *Cloud Firestore* é estabelecido concatenando a string "House/", que é o nome da coleção, com o e-mail do utilizador, armazenado na variável *epromEmail*.

Esta operação define o local exato na base de dados, onde a variável *Presence* será armazenada ou atualizada.

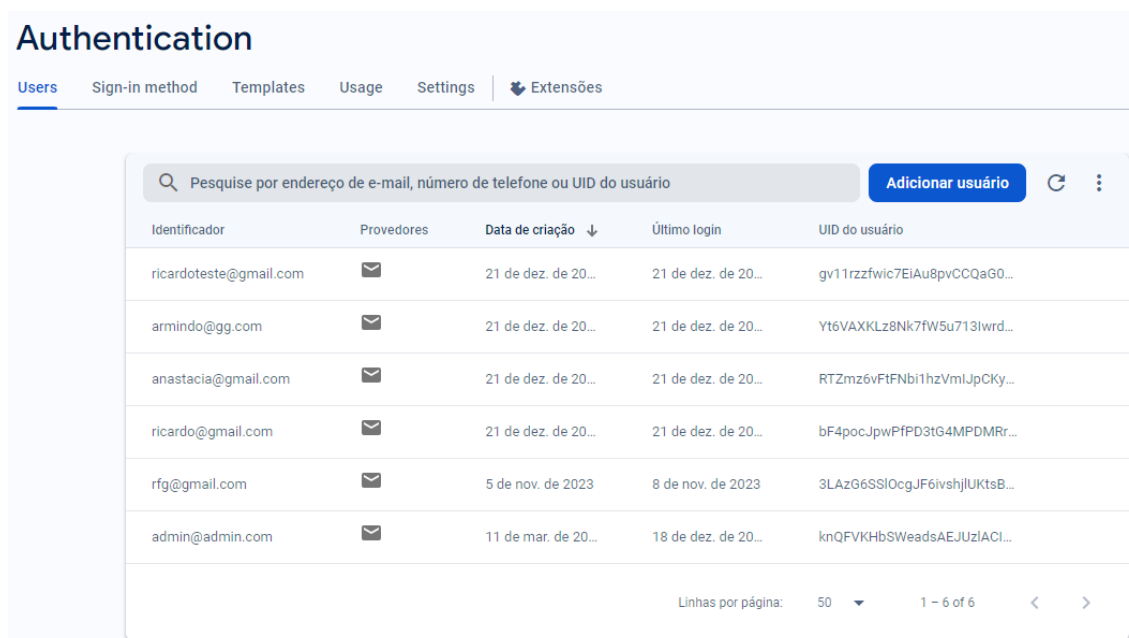
4. A função tenta primeiro atualizar um documento existente com o novo estado de presença usando o método *Firebase.Firestore.patchDocument*. Se esta operação for bem-sucedida, a função é encerrada imediatamente, indicando que a atualização foi efetuada com êxito.
5. Caso o documento não exista, por exemplo, num primeiro login do utilizador. Um novo documento é criado com os dados fornecidos, utilizando o método *Firebase.Firestore.createDocument*.

Concluindo, este processo envolve inicialmente a verificação da existência do registo do utilizador, seguida da atualização ou criação do documento na base de dados conforme a necessidade, assegurando assim a correta gestão das informações.

4.5.3 *Firebase Authentication*

No sistema desenvolvido, a *Authentication* tem um papel fundamental, pois, garante que apenas utilizadores registados e devidamente autenticados, tenham acesso aos seus dados específicos armazenados na *Cloud Firestore*. Optou-se pelo método de autenticação baseado em e-mail e senha, uma vez que, é um formato familiar e amplamente adotado em diversas plataformas online, facilitando assim o processo de registo.

A figura 4.10 ilustra a estrutura da *Firebase Authentication* do sistema desenvolvido.



The screenshot shows the 'Users' tab in the Firebase Authentication console. It features a search bar at the top with the placeholder text 'Pesquise por endereço de e-mail, número de telefone ou UID do usuário'. To the right of the search bar is a blue button labeled 'Adicionar usuário'. Below the search bar is a table with the following columns: 'Identificador', 'Provedores', 'Data de criação', 'Último login', and 'UID do usuário'. The table contains six rows of user data. At the bottom right of the table, there is a pagination control showing 'Linhas por página: 50' and '1 - 6 of 6'.

Identificador	Provedores	Data de criação	Último login	UID do usuário
ricardoteste@gmail.com	✉	21 de dez. de 20...	21 de dez. de 20...	gv11rzzfwic7EiAu8pvCCQaG0...
armindo@gg.com	✉	21 de dez. de 20...	21 de dez. de 20...	Yt6VAXKLz8Nk7fW5u713lwrD...
anastacia@gmail.com	✉	21 de dez. de 20...	21 de dez. de 20...	RTZmz6vFtFNbi1hzVmlJpCKy...
ricardo@gmail.com	✉	21 de dez. de 20...	21 de dez. de 20...	bF4pocJpwPfpD3tG4MPDMRr...
rfg@gmail.com	✉	5 de nov. de 2023	8 de nov. de 2023	3LAzG6SSIOcgJF6ivshjIUKtsB...
admin@admin.com	✉	11 de mar. de 20...	18 de dez. de 20...	knQFVKHbSWeadsAEJUzIACI...

Figura 4.10 – Autenticação dos utilizadores [Fonte: Firebase Authentication]

Observe-se que, existem 5 colunas:

- *identificador*: corresponde ao e-mail do utilizador registado.
- *Data de criação*: indica a data de registo do utilizador.
- *Último login*: mostra a data do último login.
- *UID do utilizador*: um identificador único e inalterável atribuído a cada utilizador, mantendo-se constante independentemente do número de registos feitos pelo mesmo.

A utilização desta ferramenta, no contexto do projeto desenvolvido é efetuada da seguinte forma: Quando um cuidador se regista na aplicação, é adicionada uma nova entrada à tabela da imagem. Posteriormente, sempre que um utilizador tente efetuar login, o sistema verifica todas as entradas da tabela e, caso o e-mail introduzido corresponda a alguma entrada existente, o login é realizado com sucesso, caso contrário, o acesso ao *dashboard* da aplicação é negado. Cada linha da tabela corresponde a um utilizador diferente.

Concluindo, a utilização estratégica de e-mails como identificadores únicos, em conjunto com as funcionalidades robustas da *Cloud Firestore* e o sistema de autenticação do *Firebase*, asseguram uma transmissão de dados precisa e segura. Isso possibilita que cada dispositivo ESP32 comunique de forma inequívoca com o seu lençol correspondente, mesmo em cenários onde existem utilizadores a interagir simultaneamente com a aplicação. Desta forma, assegura-se a integridade dos dados transmitidos, eliminando os erros de comunicação entre os dispositivos e a *Cloud*, garantindo assim a fiabilidade e a eficácia do sistema em ambientes com múltiplos utilizadores.

Capítulo 5

Interface IoT

Ao longo do documento, foram sendo feitas referências a uma aplicação para *smartphone*, que será utilizada pelo cuidador com a finalidade de mostrar em tempo real o estado de ausência/presença e percentagem de humidade da cama do paciente. Este capítulo explica em detalhe o processo de desenvolvimento da app, recorrendo para isso a diagramas UML (Unified Modeling Language).

5.1 Visão Geral

Numa solução a ser comercializada, terá de existir para além do sistema desenvolvido, um guia rápido da aplicação, para que os utilizadores não tenham dúvidas quanto ao seu uso. No âmbito da dissertação, a solução foi pensada essencialmente num contexto de utilização doméstica, na medida em que existe um paciente para um cuidador.

A figura 5.1 ilustra um fluxograma acerca da arquitetura geral da aplicação.

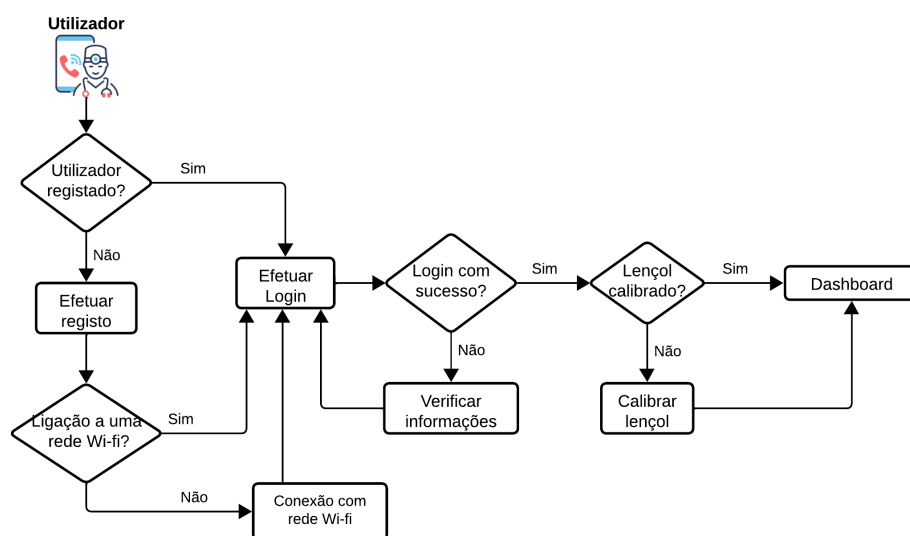


Figura 5.1 – Arquitetura geral da aplicação desenvolvida para *smartphone*

Aquando da primeira utilização é necessário que o utilizador ou cuidador ligue o sistema e se registre na aplicação, tal e qual como em qualquer outra plataforma como os serviços de email ou redes sociais. De seguida, é necessário ligar-se à rede criada pelo "lençol inteligente". Esta talvez seja a parte complicada, uma vez que, os utilizadores comuns podem não estar tão familiarizados com este tipo de ações. Depois do registo feito, e do sistema já estar conectado a uma rede Wi-Fi, é necessário efetuar o login com os dados fornecidos aquando do registo. Uma verificação da calibração do sistema será efetuada e caso o sistema não esteja calibrado será necessário a sua calibração. Por fim, se o sistema estiver calibrado e o login for realizado com sucesso, é apresentado o *dashboard* com as informações do paciente em tempo real.

A arquitetura geral pode ser dividida em quatro áreas principais, as quais são: o registo, que inclui a ligação com a rede Wi-Fi, o login, a calibração e o *dashboard*. Estas funcionalidades serão explicadas em detalhe nas secções seguintes.

De modo a perceber os principais componentes que compõem a aplicação e as relações que existem entre eles, a figura 5.2 apresenta o diagrama de componentes da aplicação.

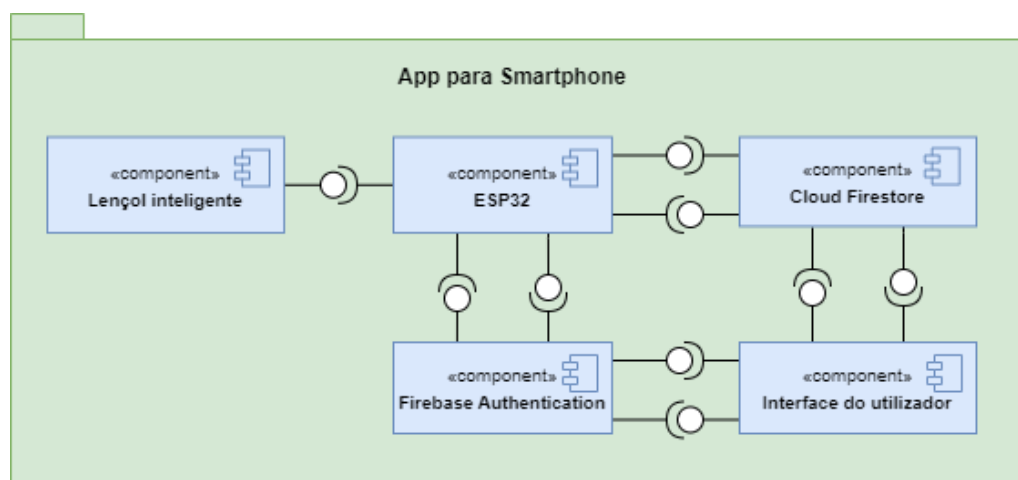


Figura 5.2 – Diagrama de componentes da aplicação

Sistematizando a figura, pode-se inferir que o lençol inteligente fornece informações ao microcontrolador. Este por sua vez, e de igual modo à interface de utilizador, comunicam com a *Cloud Firestore* e com a *Firebase Authentication* em dois sentidos, ou seja, fornecem dados e também requerem alguns. Por exemplo, o microcontrolador envia à *Cloud Firestore* os estados de presença/ausência e também requer dados relativos a *flags* de calibração como veremos ao longo das próximas secções.

Por último, e para contextualizar as ações do cuidador, apresenta-se um diagrama de caso de uso geral tendo em conta a ótica do utilizador. Este é o interveniente responsável por efetuar o registo da primeira vez que utilizar o lençol inteligente, estabelecer a conexão a uma rede Wi-Fi para que o ESP32 efetue ações de leitura/escrita no *Firebase*, calibrar o sistema se necessário e por último fazer login para aceder ao *dashboard* e ver o estado da cama do paciente.

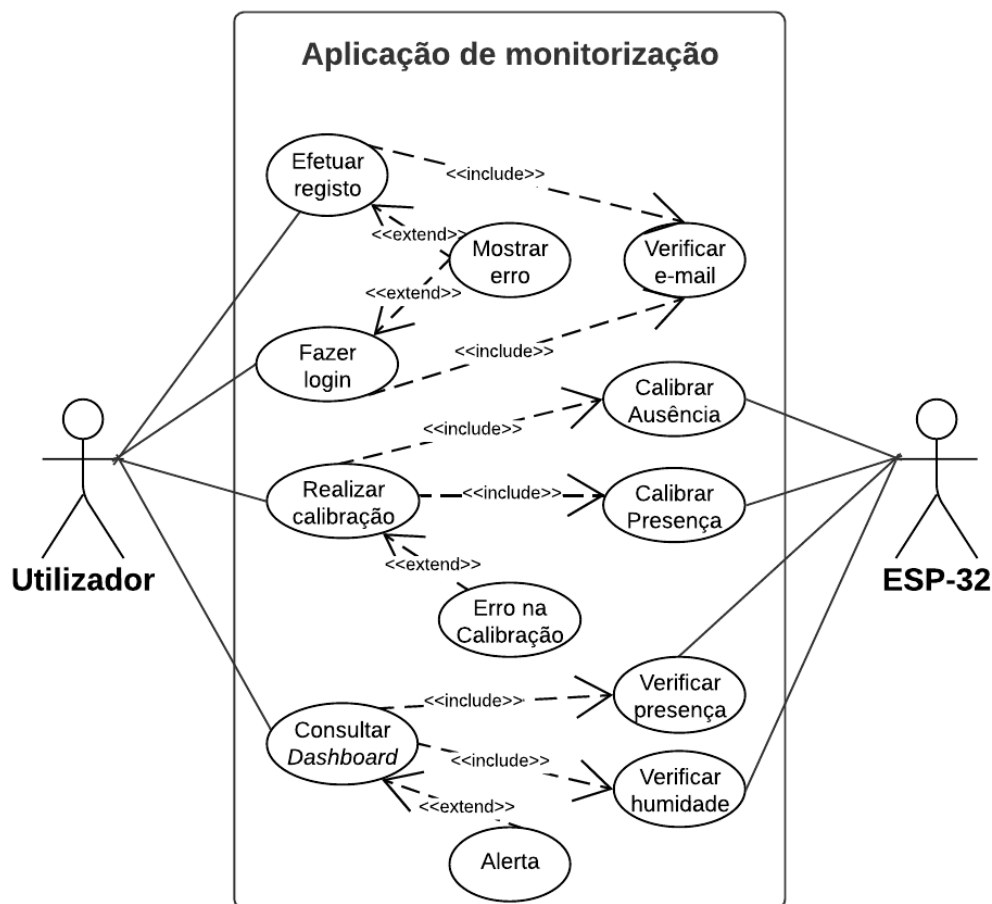


Figura 5.3 – Diagrama de caso de uso geral

5.2 Menu de Registo

O primeiro passo, depois do lençol estar já sobre a cama e pronto a ser utilizado, é efetuar o registo do utilizador na aplicação. Para isso, é necessário fornecer um e-mail, password e verificar esta última.

Essencialmente, depois da aplicação fazer internamente algumas verificações como: se o formato de e-mail é válido ou se as passwords coincidem e, assumindo que os dados estão corretos é adicionada na *Firebase Authentication* uma entrada com o e-mail registado. Ao mesmo tempo, é atribuído pela plataforma um *Unique Identifier* que identifica inequivocamente o utilizador registado e, para além disso, é criado na *Cloud Firestore* um novo documento, cujo nome é o e-mail do registo. Este mecanismo é muito importante, uma vez que, permite distinguir inequivocamente todos os utilizadores através do seu e-mail, evitando que quando dois cuidadores estão a aceder ao dashboard do seu lençol, corram o risco de ver informações de outro lençol/doente.

Por fim, depois do documento ser criado e o registo estar completo, é feita a transição para o menu de login.

A figura 5.4 mostra as ações do *back-end* do sistema durante o processo de registo.

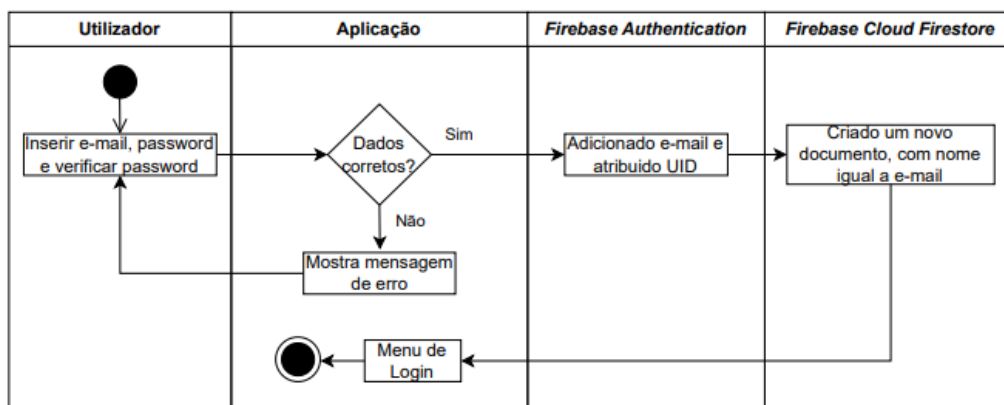


Figura 5.4 – Diagrama de atividades do processo de registo

A figura 5.5 ilustra o menu de registo da aplicação.



Figura 5.5 – Menu de registo

5.2.1 Conexão à Internet

Depois do utilizador efetuar o registo na aplicação, é necessário estabelecer a conexão entre o microcontrolador e a Internet, pois só assim existirá comunicação entre o ESP32 e o *Firebase*, o que é necessário para o funcionamento do sistema.

Para isso, o microcontrolador cria a sua própria rede Wi-Fi local, cujo nome é *AutoConnectAP*, funcionando como um ponto de acesso sem fios, e permitindo que outros dispositivos se conectem a ele. Quando isto acontece diz-se que o ESP32 se encontra no modo ponto de acesso. *AutoConnectAP*. Note-se que, esta rede local criada não providencia acesso à Internet.

O utilizador terá de abrir a lista de redes Wi-Fi no seu *smartphone* e entrar na referida rede criada. Automaticamente, o *Wi-Fi Manager* abre no ip 192.168.4.1 e são apresentadas ao cuidador a lista de redes Wi-Fi próximas. Neste passo, o utilizador terá de escolher a rede que pretende, e colocar a password tal como se tivesse a configurar a rede Wi-Fi de casa. Para além da password da rede terá de fornecer o e-mail e a password com a qual se registou na aplicação.

Por fim, e caso os dados fornecidos estejam corretos, o microcontrolador terá acesso à Internet e entra no modo estação, uma vez que, funciona como um cliente de uma rede sem fios.

A figura 5.6 sistematiza o processo de ligação à rede Wi-Fi.

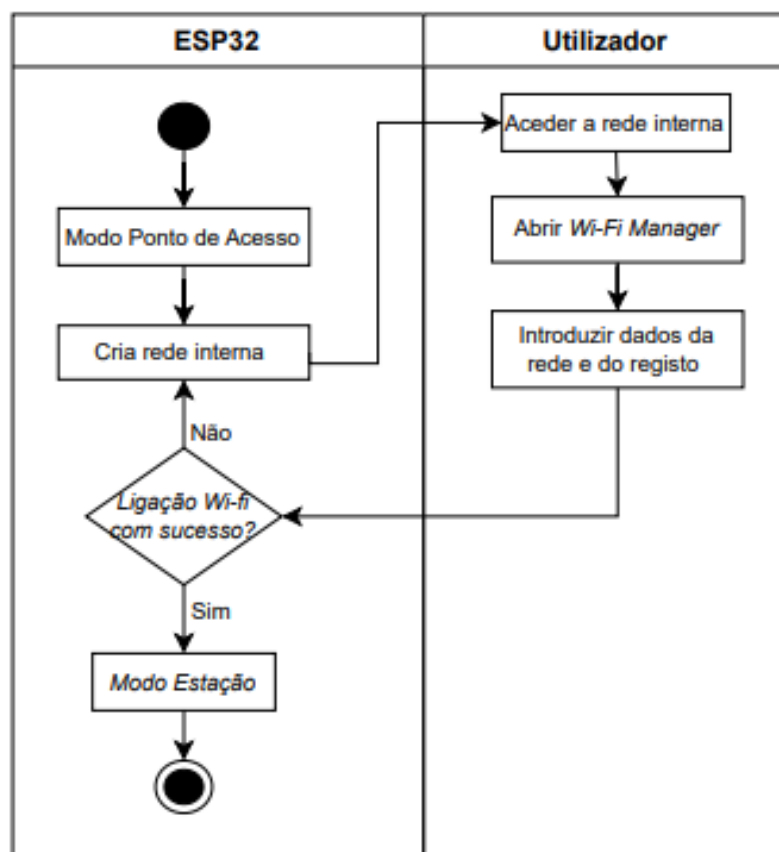


Figura 5.6 – Diagrama de atividades do processo de ligação à rede Wi-Fi

Tendo em conta a ótica do utilizador, a figura 5.7 realça as principais interações do mesmo, que basicamente incluem apenas inserir a password da rede Wi-Fi e fornecer os dados do registo.

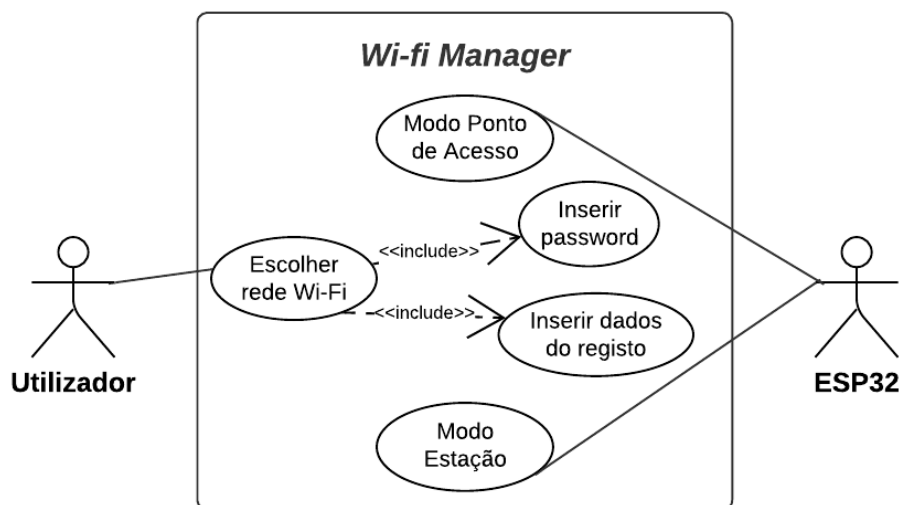


Figura 5.7 – Diagrama de caso de uso do *Wi-Fi Manager*

As imagens da interface do utilizador, que ilustram os passos para estabelecer a conexão do sistema com a Internet, podem ser consultadas na figura A.2 do anexo A

5.3 Calibração do sistema

A calibração do sistema é fundamental para o correto funcionamento do lençol inteligente, esta etapa determina se os estados de presença/ausência são determinados corretamente, providenciando uma informação fidedigna aos cuidadores. Note-se que, a calibração apenas é feita para o estado de presença/ausência e em nada interfere com a deteção de humidade.

Tendo em conta que o lençol inteligente pode ser utilizado em várias condições de utilização, nomeadamente, diferentes colchões e pessoas com diferentes pesos, é necessário proceder à calibração do sistema aquando da primeira utilização. Neste caso, a própria aplicação exige a calibração antes do utilizador poder avançar para o *dashboard*. A estabilidade dos valores lidos pelo sistema durante a calibração de ausência é elevada, sendo pouco provável que, durante a operação normal do sistema no estado de ausência, esses valores apresentem divergências significativas. Caso o sistema identifique uma discrepância de 10 pF nos valores medidos quando o lençol se encontra desocupado, em comparação com os valores obtidos durante a calibração de ausência, automaticamente solicitará uma nova calibração se essa diferença não for suficiente para ser detetada presença.

Como já referido, a calibração é requerida automaticamente aquando da primeira utilização, mas na eventualidade de ser necessário recalibrar o sistema, o utilizador pode aceder ao menu de calibração através do menu de registo. Primeiramente é calibrada a ausência e para isso, o lençol deve estar conectado ao transmissor e não existir nada nem ninguém sobre o mesmo. Quando a

aplicação comunicar que a calibração de ausência foi efetuada com sucesso, é necessário calibrar a presença. Para isso, deita-se o paciente no lençol e aguarda-se até novo alerta de sucesso de calibração, desta vez, de presença.

A figura 5.8 ilustra o menu de calibração.

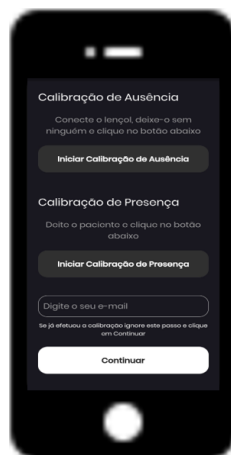


Figura 5.8 – Menu de calibração da aplicação

Na imagem, é possível observar um campo designado para inserir o e-mail utilizado durante o processo de registo. Este é necessário, pois permite que a aplicação envie as *flags* de calibração para o documento específico na *Cloud Firestore*, onde serão posteriormente lidas pelo microcontrolador. Note-se que no sistema desenvolvido, o e-mail funciona como o identificador do lençol.

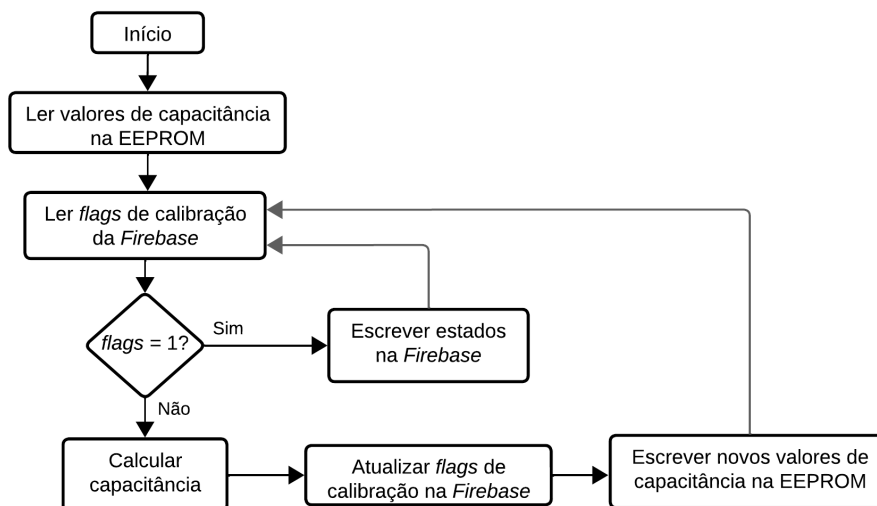


Figura 5.9 – Funções do ESP32 no processo de calibração

A figura 5.9 sistematiza as ações do microcontrolador no processo de calibração. Este é desencadeado pelo utilizador e controlado por duas *flags* distintas: *AbsenceCal* e *PresenceCal*. A primeira refere-se à ausência, enquanto a segunda está associada à presença. Quando o cuidador carrega no botão de calibração na aplicação, esta define a *flag* correspondente para 0 na *Cloud*

Firestore. Por outro lado, o ESP32 quando verifica que o valor é 0, mede valores de capacitância durante um intervalo de 5 segundos e calcula a média dos valores obtidos. De seguida, a *flag* é definida para 1 na *Cloud Firestore*, indicando à aplicação que o sistema foi calibrado. Os valores de capacitância provenientes da calibração e recolhidos pelo ESP32 são armazenados na memória não volátil EEPROM, dado que, são necessários para o microcontrolador estabelecer os estados de presença e ausência. Uma vez gravados na EEPROM, mesmo que o sistema seja desligado, permanecem preservados.

Caso passem mais de 17 segundos, desde que a *flag* de calibração é colocada a 0, e esta ainda não tenha transitado para 1, o utilizador recebe um aviso de falha na calibração. Nesse cenário, terá de verificar a conexão com a rede Wi-Fi e o e-mail introduzido. O tempo 17 segundos, foi determinado com base em testes empíricos. Este tempo compreende o período máximo desde o qual, o microcontrolador identifica que a *flag* está definida como 0, calcula a média dos valores de capacitância durante pelo menos 5 segundos e coloca a *flag* a 1 na *Cloud Firestore*. Em situações desfavoráveis, esse ciclo pode levar cerca de 10 segundos, portanto, a escolha de 17 segundos foi estabelecida, proporcionando uma margem adicional de 7, o que inclui algum atraso que possa existir na comunicação.

Para concluir, a figura 5.10 ilustra a troca de informações entre os principais intervenientes no processo de calibração, que para além do utilizador incluem a aplicação, a *Cloud Firestore* e o microcontrolador.

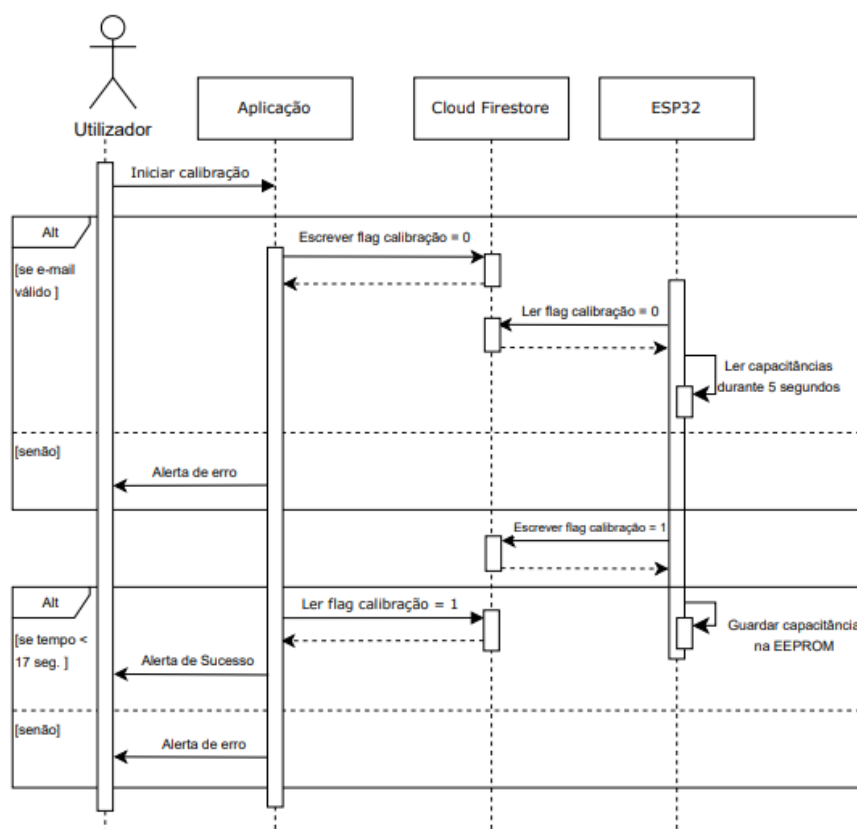


Figura 5.10 – Diagrama de sequência do processo de calibração

5.4 Menu de login

O menu de login, tem o aspeto da figura 5.11.



Figura 5.11 – Menu de login da aplicação

Depois do registo, o utilizador é solicitado a efetuar login na aplicação. Essa etapa é crucial para que o sistema possa identificar o cuidador e conceda o acesso ao *Dashboard*, onde poderá consultar em tempo real o estado da cama do paciente, caso reúna as condições necessárias, nomeadamente, o utilizador estar registado e o sistema calibrado.

No menu de login, o utilizador terá de fornecer o e-mail com o qual se registou e a respetiva password. Para além disso, será necessário preencher também o seu nome e género, para que depois no *Dashboard*, o cuidador saiba o nome do doente.

Na prática, o que o código faz é verificar se existe alguma entrada com o e-mail introduzido na *Firebase Authentication*. Esta etapa é essencial, pois garante que apenas os utilizadores autenticados têm acesso ao *Dashboard*, contribuindo para a segurança e integridade do sistema. Já para detetar se o sistema está ou não calibrado, analisa na *Cloud Firestore* o estado das *flags* de calibração. Caso ambas estejam com o valor 1, significa que o sistema está calibrado.

O diagrama de atividades do processo de login encontra-se na figura 5.12

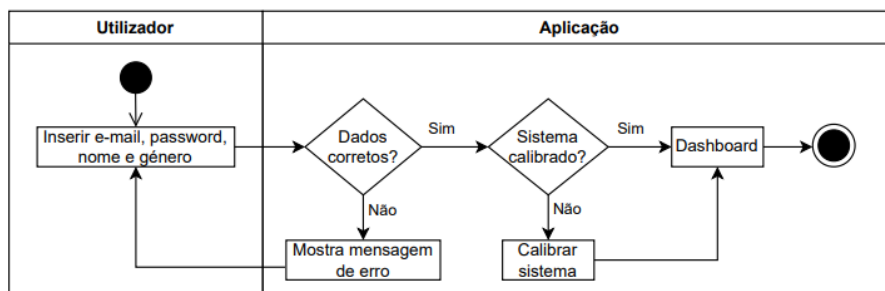


Figura 5.12 – Diagrama de atividades do processo de login

5.5 Dashboard

O último menu da aplicação é o *dashboard*, ilustrado na figura 5.13.

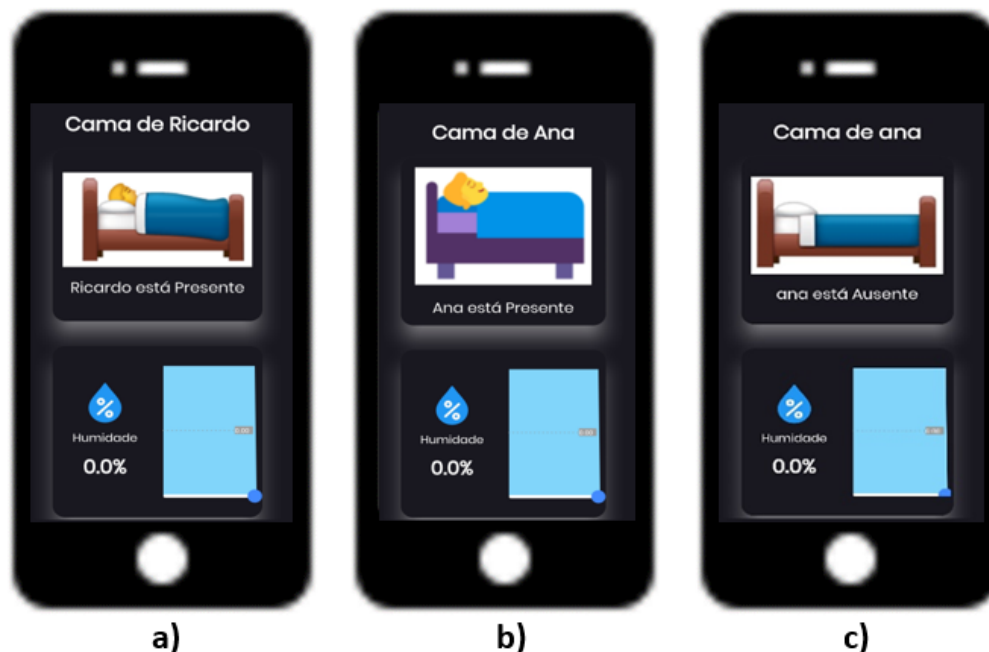


Figura 5.13 – Menu *Dashboard*

Este menu desempenha um papel crucial para os cuidadores, fornecendo acesso em tempo real ao estado da cama do paciente, especialmente no que se refere à presença e humidade. O *Dashboard* é dividido em duas secções: a parte superior, que exhibe o estado de ausência juntamente com o nome do paciente. Se o paciente for homem e estiver presente, a figura a) é apresentada, caso seja uma mulher, a figura b) é ilustrada. Quando o paciente está ausente, aparecerá a figura c).

Na parte inferior do menu, os cuidadores podem verificar a percentagem de humidade, acompanhada por um gráfico que representa uma curva, indicando variações na sua percentagem. No exemplo da imagem, a curva não é observada, pois a humidade é de 0%.

A figura 5.14 sistematiza os processos que acontecem no *back-end*.

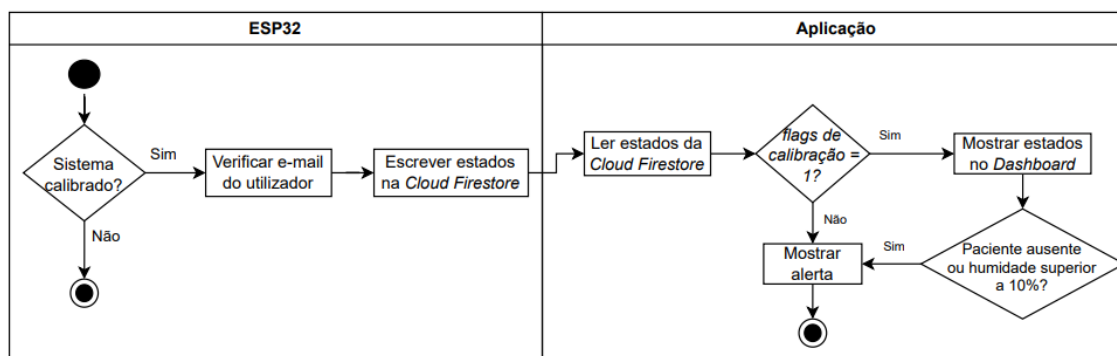


Figura 5.14 – Diagrama de atividades - Menu *Dashboard*

Essencialmente, a aplicação verifica se o utilizador é elegível para consultar o *dashboard* e, caso o sistema esteja calibrado os estados são mostrados. No caso de o sistema não estar calibrado, a aplicação solicita a realização da calibração.

Se o paciente não estiver na cama ou por qualquer motivo existir libertação de líquidos, o cuidador recebe alertas a cada minuto, mesmo quando a aplicação está em segundo plano. Esta funcionalidade será detalhada na secção seguinte.

5.6 Alertas

Os alertas desempenham um papel crucial na operação da aplicação, abrangendo não apenas mensagens de erro resultantes de dados incorretos inseridos pelo utilizador, como um e-mail inválido, mas também avisos essenciais para o funcionamento adequado do sistema. Exemplos desses alertas incluem a notificação de ausência do paciente ou da humidade ser superior a 10%, a situação de uma eventual desconexão entre o lençol e o transmissor, ou ainda, este se encontrar desligado.

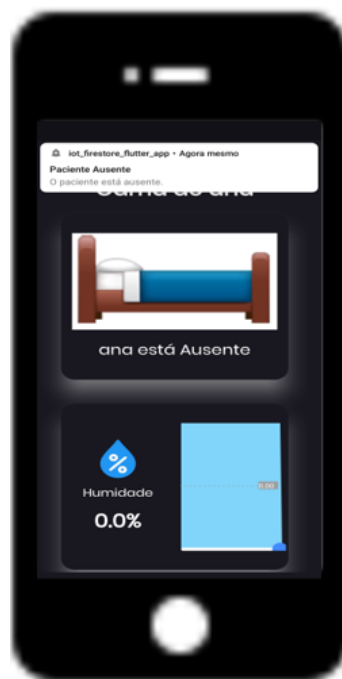


Figura 5.15 – *Dashboard* - Alerta de ausência

A figura 5.15 ilustra o *dashsboard* com uma notificação *push* no cenário em que o acamado está ausente. Este tipo de notificações caracterizam-se por aparecerem no topo do ecrã ou no painel, caso o *smartphone* esteja bloqueado.

Para a implementação das notificações foi utilizada a biblioteca *flutter_local_notifications*, que proporciona funcionalidades relevantes no contexto deste projeto, como agendar e personalizar notificações. Com o propósito de alertar o cuidador acerca da ausência do paciente ou de uma humidade superior a 10%, é realizada primeiramente a inicialização do serviço de notificações.

De seguida, a aplicação verifica na *Cloud Firestore* as variáveis relativas à presença e humidade, caso detete ausência ou humidade elevada alerta o cuidador através de uma notificação *push*.

Após o primeiro alerta, é inicializado um timer de 1 minuto e caso se continue a verificar ausência ou humidade, outra notificação é enviada. A figura 5.16 esquematiza este procedimento através de um diagrama de sequência.

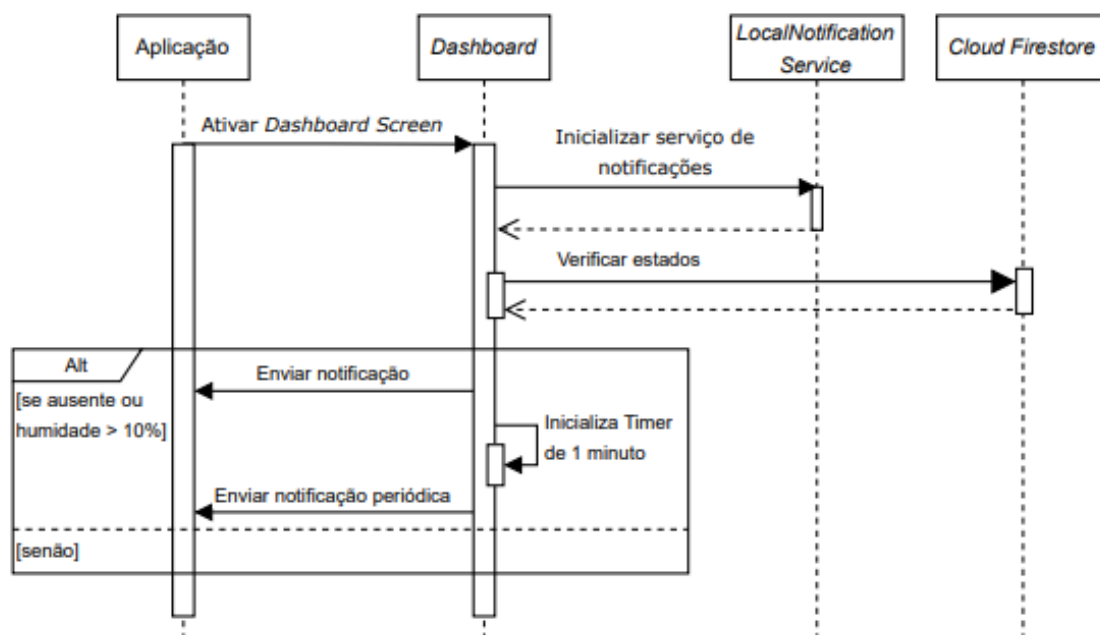


Figura 5.16 – Diagrama de sequência do processo de envio de notificações

Se por lapso o cuidador não ligar o sistema, ou na eventualidade de existir alguma anomalia e o mesmo deixe de funcionar, ou até mesmo pelo simples facto da bateria acabar, é imperativo o cuidador ser informado de que o transmissor se encontra desligado.

Para isto, implementou-se o seguinte mecanismo: o microcontrolador lê a hora atual de um servidor *Network Time Protocol*, cujo fuso horário corresponde ao da Europa Ocidental, onde está incluído Portugal. De seguida, envia a hora atual para a *Cloud Firestore*. A aplicação, por sua vez, lê a hora atual do servidor NTP e a hora escrita pelo ESP32 na *Cloud Firestore*. Se a diferença entre esses tempos for superior a 20 segundos, significa que, por qualquer motivo o microcontrolador ou a aplicação não comunicaram com o *Firebase*. O valor de 20 segundos foi definido com base em testes empíricos, onde se verificou que abrangia qualquer atraso que pudesse existir na comunicação. Para além disso, é um tempo máximo bastante aceitável para alertar o cuidador em caso do sistema se encontrar desligado.

É importante clarificar, que apesar do servidor NTP ler a hora portuguesa, isso não influencia caso o lençol inteligente seja utilizado noutro país. O que importa é o microcontrolador e a aplicação lerem os dados de tempo de um servidor, cujo fuso horário é o mesmo.

A figura 5.17 apresenta o diagrama de sequência para este género de alerta.

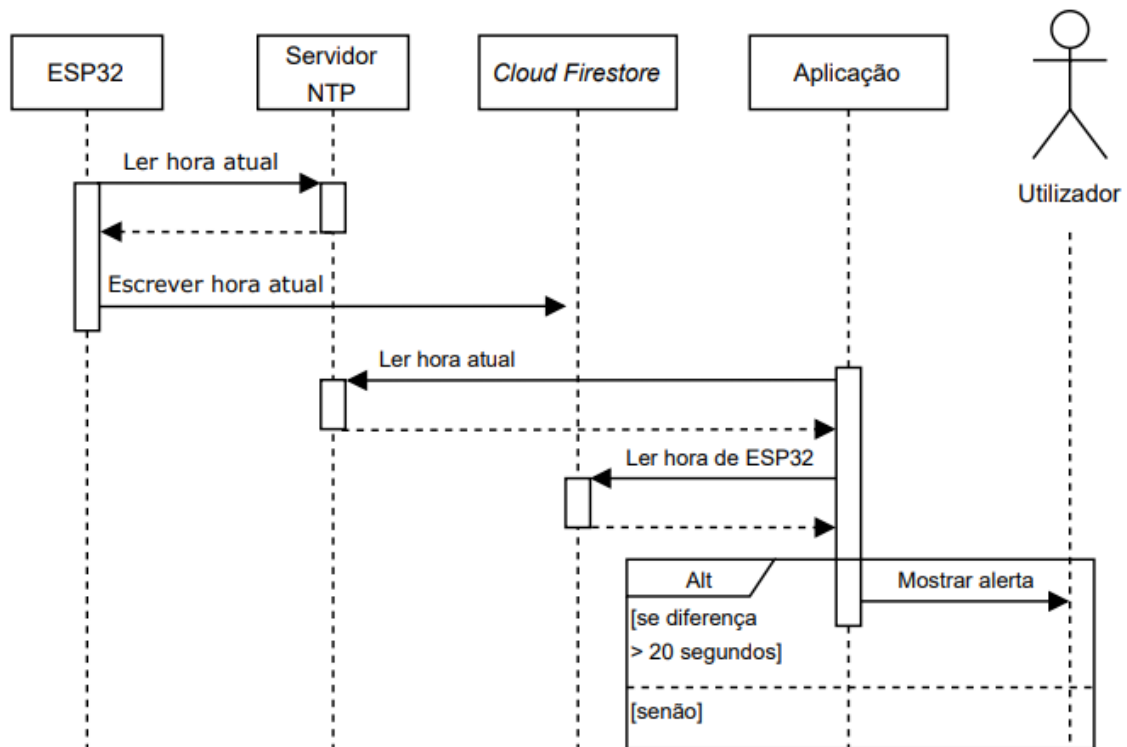


Figura 5.17 – Diagrama de sequência: Mecanismo de alerta de sistema desligado

O sistema não alerta apenas o cuidador, caso o transmissor esteja desligado, mas também identifica se por qualquer motivo, o conector entre o transmissor e o lençol está desconectado. Essa detecção é crucial, uma vez que, o sistema não funcionará sem o conector no lençol.

A detecção ocorre quando a aplicação lê um valor de capacitância da variável *lencol* da *Cloud Firestore* inferior a 40 pF. Este limiar foi estabelecido após testes empíricos, onde se observou que, quando o conector não está ligado, a capacitância medida pelo sistema é de 40 pF. Esse valor existe, porque apesar do conector não estar ligado, existe um condensador de 10 nF ligado ao microcontrolador, conforme já apresentado na figura 4.3 da secção 4.3.

A figura 5.18 ilustra o processo de detecção da desconexão do transmissor.

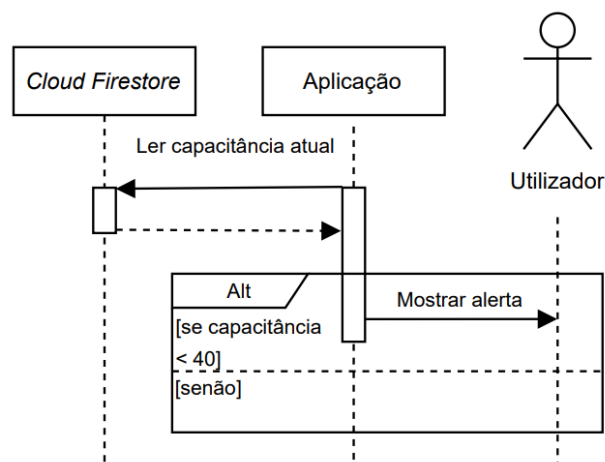


Figura 5.18 – Diagrama de sequência: Desconexão do transmissor

Concluindo, e de forma a sistematizar as noções apresentadas ao longo deste capítulo, a figura 5.19 resume o processo desde a autenticação até à escrita dos estados no *Firebase*.

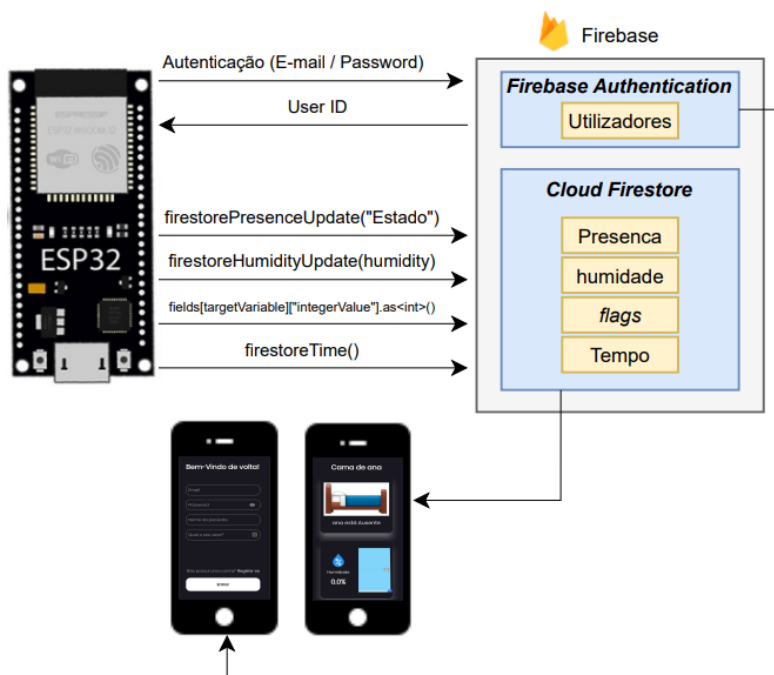


Figura 5.19 – Interface IoT para detecção de presença e humidade - Figura adaptada de [84]

Capítulo 6

Apresentação de resultados

Este capítulo, apresenta os resultados obtidos a partir da implementação e teste do lençol inteligente desenvolvido. Os dados apresentados refletem o desempenho do sistema em condições reais de presença ou ausência de pessoas e humidade, fornecendo dados cruciais sobre a sua fiabilidade e aplicabilidade prática.

6.1 Detecção de presença e ausência

Primeiramente, foram realizados testes com o lençol seco, para determinar a presença e ausência de acamados no lençol. O método adotado para testar o dispositivo e avaliar o seu desempenho, envolveu a recolha de dados do microcontrolador relativamente à capacitância medida. Os dados foram registados quando o lençol estava vazio, para detetar a ausência de pessoas, e durante a noite, quando uma pessoa estava deitada, para detetar a presença. Os resultados dessas medições foram enviados para uma folha do *Google Sheets* e estão ilustrados nas imagens [6.1](#) e [6.2](#).

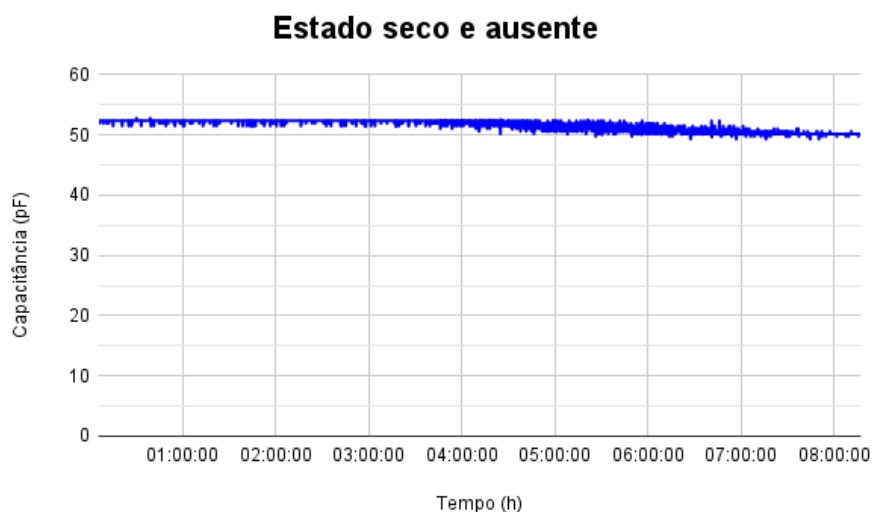


Figura 6.1 – Capacitância: Lençol seco e ausente

A figura 6.1, apresenta a variação da capacitância do lençol ao longo do tempo, quando este está seco e sem a presença de alguém. Este teste foi realizado ao longo de aproximadamente 8 horas e permite inferir que os valores recolhidos na ausência de pessoas são bastante constantes. A média da capacitância situou-se nos 50.9 pF, variando de um mínimo de 49.3 pF a um máximo de 52.4 pF. O desvio padrão obtido foi de 0.76 pF, indicando uma consistência significativa, considerando os 1573 dados recolhidos, que foram registados em intervalos de 10 em 10 segundos.



Figura 6.2 – Capacitância: Lençol seco e presente

Quanto à presença de pessoas, os valores recolhidos permitem concluir que, quando alguém está presente no lençol e este fica sujeito a uma determinada pressão, a capacitância medida não é constante, apresentando uma maior variação relativamente à deteção de ausência. Este comportamento seria de esperar, uma vez que, na presença de um corpo, movimentos, mesmo que mínimos, são inevitáveis. Os valores obtidos durante os testes de presença revelam uma capacitância média de 79.3 pF, com um desvio padrão de 4.6 pF. O valor mínimo foi 65.9 pF e máximo 90.8 pF. É fundamental destacar que o valor específico medido em ambos os estados não é crucial, pois pode variar de acordo com fatores como o tipo de colchão utilizado, ou o peso da pessoa. O importante é existir um limite bem definido, entre o estado de ausência e presença, pelo que, é fundamental existir uma diferença significativa entre o valor medido nos dois estados.

Para garantir uma deteção confiável em diversos contextos de uso, como por exemplo diferentes colchões, o menu de calibração é de extrema importância. É através dos valores obtidos no processo de calibração de ausência e presença que os limites são definidos. A calibração, no contexto do lençol inteligente, nada mais é, do que o microcontrolador medir a capacitância e efetuar uma média durante 5 segundos em dois cenários distintos: quando o lençol está seco e vazio, e quando está seco e com alguém presente. Os valores obtidos durante o processo de calibração são armazenados na EEPROM, e o limite para distinguir a ausência da presença é calculado. Para além disso, o facto de existir a funcionalidade de calibração do lençol, permite atualizar constantemente

a fiabilidade do sistema.

A figura 6.3 incorpora num único gráfico as informações das últimas duas imagens, proporcionando uma compreensão visual do limite entre os estados.



Figura 6.3 – Capacitância: Lençol seco nos dois estados

Na figura, a linha verde ilustra os valores medidos no estado de presença com o lençol seco, enquanto a linha azul representa o estado ausente. A linha vermelha ilustra qual seria o *threshold* neste caso. Para se obter o *threshold* é feito o seguinte:

- Calibração de ausência, que envolve calcular a média da capacitância durante 5 segundos, quando o lençol está seco e sem ninguém.
- Calibração de presença, que envolve calcular a média da capacitância durante 5 segundos, quando existe alguém no lençol.
- Calcular a média das duas médias anteriores e subtrair 5.

Neste caso os valores médios de capacitância na ausência e presença são 50.9 pF e 79.3 pF respetivamente, resultando em uma média de 65.1 pF. Depois subtrai-se 5, ficando o limite nos 60.1 pF. Esta subtração é realizada para garantir uma margem maior na deteção de presença, considerando que é o estado onde os valores medidos são menos constantes. Qualquer valor igual ou superior a 60.1 pF seria entendido pelo microcontrolador como presença, e valores inferiores, como ausência.

Estes resultados oferecem uma visão detalhada da resposta do lençol inteligente, estabelecendo uma base sólida para a compreensão das suas capacidades e limitações.

6.1.1 Detecção da desconexão do sistema

A figura 6.4 ilustra a variação entre os diferentes estados possíveis no lençol inteligente quando este está seco.

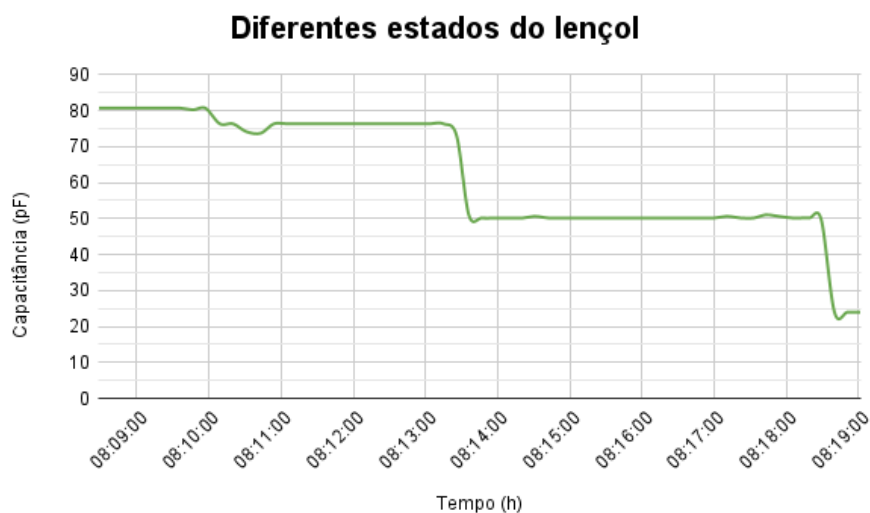


Figura 6.4 – Diferentes estados detetados pelo lençol

Inicialmente a capacitância encontra-se nos 80 pF, o que indica a presença de um acamado e depois desce para cerca de 50 pF, permitindo concluir que o paciente se levantou. Por fim, existe uma queda abrupta para os 20 pF, o que indica que o transmissor foi desconectado, pois, como dito anteriormente neste documento, uma capacitância inferior a 40 pF significa uma desconexão do transmissor.

6.2 Detecção de humidade

Para analisar a resposta do sistema na presença de humidade, o teste consistiu em colocar água no lençol e analisar os dados durante o processo de absorção e secagem de água pelo lençol. O gráfico da figura 6.5 apresenta os valores de capacitância recolhidos.

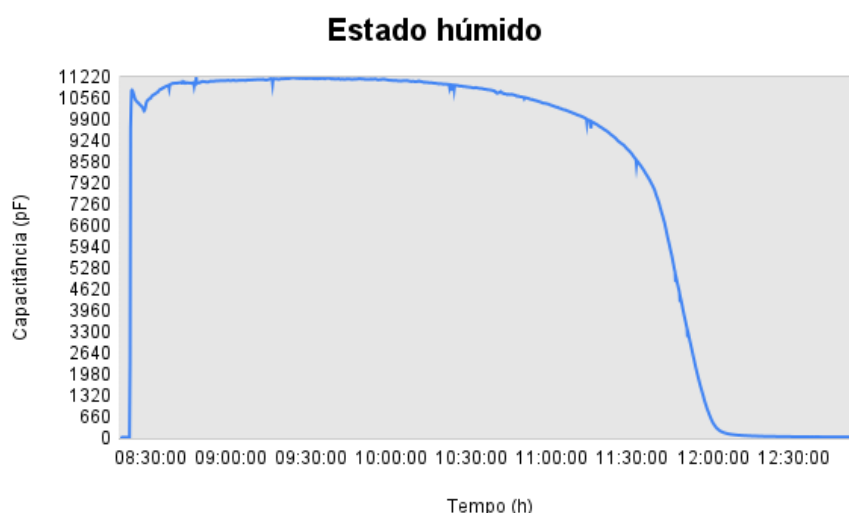


Figura 6.5 – Capacitância: Lençol húmido

Conclui-se que, a análise da capacitância proporciona uma visão eficaz do comportamento do lençol em relação ao processo de absorção e secagem de água. Através do aumento de capacitância, é possível inferir que o lençol está a absorver água, atingindo depois um valor máximo antes de começar a diminuir. A diminuição da capacitância indica que o lençol está em fase de secagem. A observação da figura indica que o lençol esteve em processo de secagem das 10:00h às 12:00h, totalizando um período de 2 horas.

Na conceção do lençol inteligente, pressupõe-se que um alerta a indicar o estado húmido, resultará numa célere substituição da cama do doente, do lençol e da própria roupa do paciente. A rápida intervenção é crucial, uma vez que, não é expectável que um paciente permaneça numa cama molhada. Por isso mesmo, o estado de ausência quando o lençol está molhado não foi considerado durante o desenvolvimento deste sistema e apesar de poder ser detetado, podem existir algumas falhas.

Para a determinação da capacitância e baseado em testes empíricos, onde o lençol era molhado com diferentes quantidades de água, foi definido um limite mínimo de 200 pF e máximo de 11700 pF. O ponto 200 pF foi estabelecido, por ser um valor que se pode afirmar com segurança que existe água no sistema e não é proveniente apenas da ocupação da cama. Já o ponto máximo foi definido, após observar que a partir de 11700 pF, o aumento da quantidade de água no lençol não se reflete mais no aumento da capacitância. Isto acontece devido ao condensador de 10 nF em série com o lençol, que limita a capacitância total do circuito.

De seguida é efetuada uma interpolação linear, mapeando 200 pF para 0% de humidade e 11700 pF para 100%. Os pontos intermédios são obtidos por interpolação, fornecendo uma representação contínua da humidade do lençol ao longo da sua variação de capacitância.

Contrariamente à deteção de presença, não é necessária uma calibração específica para a humidade, pois os limites são constantes e predefinidos, como explicado anteriormente.

Capítulo 7

Implementação e Desenvolvimento do Protótipo

Este capítulo descreve o processo de transição do circuito de testes, inicialmente implementado numa *breadboard*, para uma solução mais próxima do que poderá vir a ser um produto final. Isso envolveu a integração do circuito numa *Printed Circuit board* (PCB) dentro de um invólucro, proporcionando proteção física e melhorando a estética.

A figura 7.1 ilustra o circuito implementado na *breadboard* e que foi de extrema importância na realização dos vários testes.

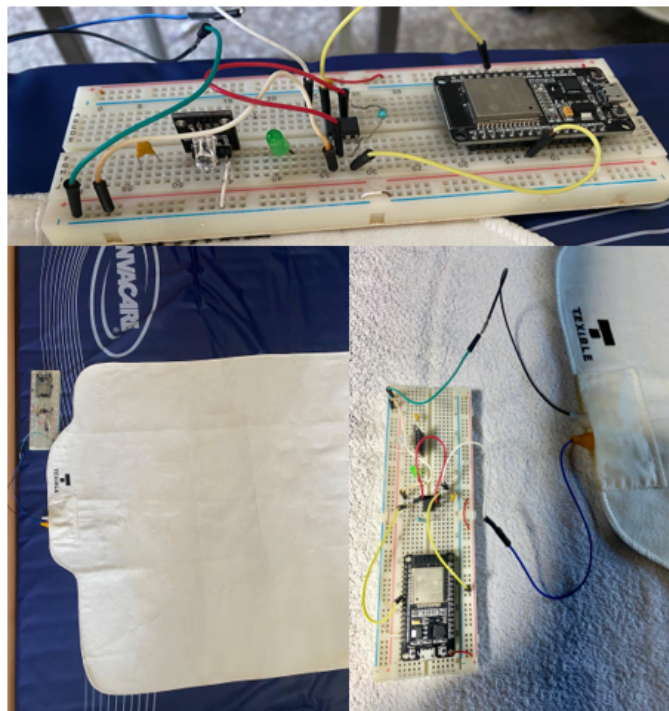


Figura 7.1 – Circuito de testes

7.1 Conector

Inicialmente, foi essencial encontrar uma solução para a conexão entre o lençol e o circuito desenvolvido. Uma vez que, os conectores do lençol são pinos do tipo macho de botões de pressão de retrosaria, projetou-se um conector que incluísse botões de pressão fêmea que encaixassem no lençol, tal como ilustrado nas figuras 7.2 e 7.3.



Figura 7.2 – Conectores macho do lençol [85]



Figura 7.3 – Conector desenvolvido

As medidas do conector podem ser verificadas na figura B.1 do anexo B.

7.2 Invólucro

Depois do conector, foi necessário pensar numa caixa para incluir o circuito. Este processo foi feito com o apoio da *Invacare*, que se disponibilizou para realizar a impressão de uma caixa 3D depois de fornecidas as medidas. Para isso, teve-se em conta a dimensão da PCB a ser incluída e da bateria para alimentar o circuito.

As figuras 7.4 e 7.5 ilustram a fase de projeto da caixa.

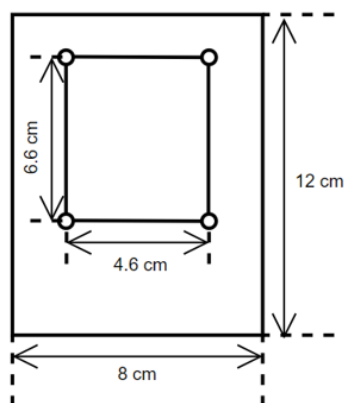


Figura 7.4 – Medidas da base da caixa

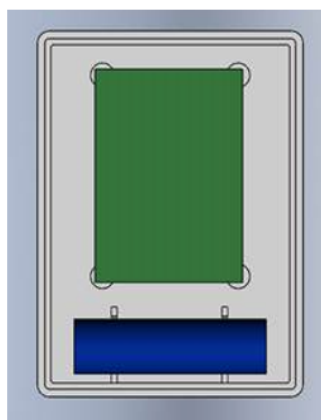


Figura 7.5 – Projeto da caixa a ser implementada

A base da caixa contém 4 suportes para fixar a PCB através de 4 parafusos. Para além da placa de circuito impresso, foi inserido na caixa um botão para ligar/desligar o sistema e adaptado um suporte para uma pilha de 9 V. As figuras 7.6 e 7.7 mostram o aspeto final do invólucro.



Figura 7.6 – Aspeto final do Transmissor

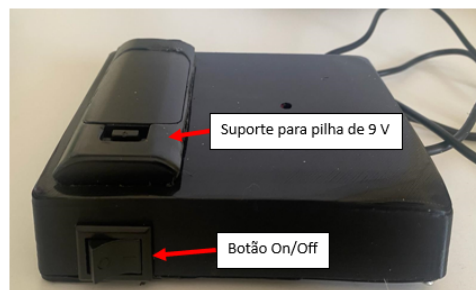


Figura 7.7 – Invólucro: Botão On/Off e suporte para pilha 9 V

Na parte central existe um led vermelho, que acende quando o transmissor está ligado.

7.3 Desenvolvimento e implementação da PCB

Para transitar da placa de teste para uma solução final, foi necessário transferir o circuito para uma PCB. Inicialmente, o circuito de aquisição de sinal foi implementado no software Eagle, conforme previamente apresentado neste documento.

A figura 7.8 apresenta o circuito sob a forma de esquemático numa PCB.

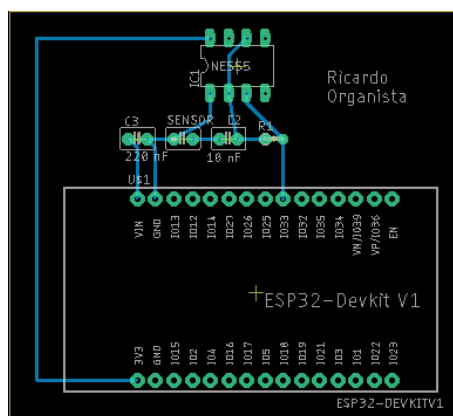


Figura 7.8 – Esquema da implementação do circuito numa PCB

As linhas a azul, representam as *pistas* da PCB, localizadas na parte inferior da placa. Estas foram pensadas de forma a facilitar a integração dos componentes nas placas, evitando sobreposições de caminhos. Uma vez que, como o circuito é relativamente pequeno, os caminhos das *pistas* foram efetuados manualmente através de processo de soldagem. A placa PCB escolhida possui dimensões de 5 cm x 7 cm.

As figuras 7.9 e 7.10, apresentam as vistas superior e inferior da placa, respectivamente:

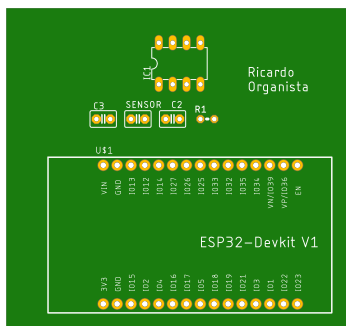


Figura 7.9 – Vista superior da PCB

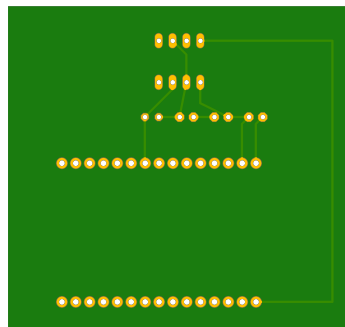


Figura 7.10 – Vista inferior da PCB

Ambas as figuras são relevantes para perceber o circuito implementado, no entanto, a vista inferior é extremamente útil porque contém os caminhos tal e qual como devem ser soldados. Após o projeto da PCB estar finalizado, passou-se à integração dos componentes na placa e à respetiva solda.

As figuras 7.11 e 7.12 ilustram o processo de soldagem e o circuito final, fixado através de 4 parafusos à base da caixa.

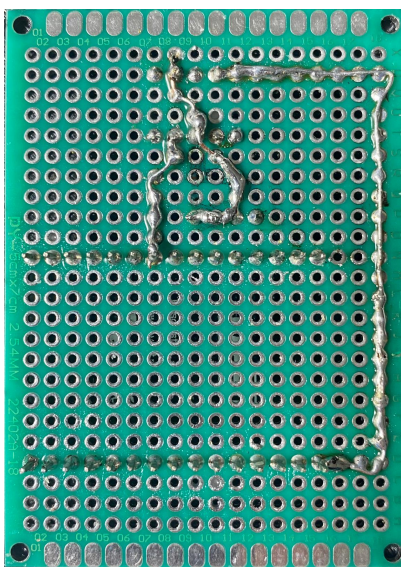


Figura 7.11 – Processo de soldagem das *pistas* da PCB

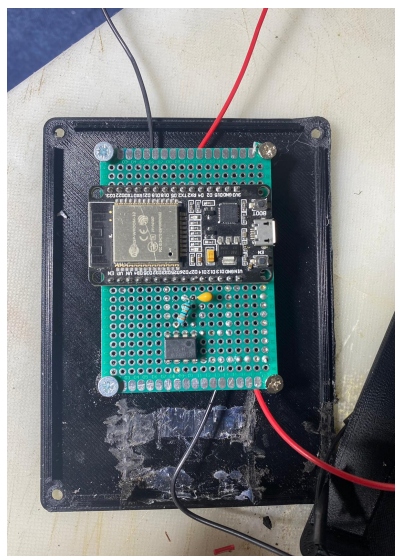


Figura 7.12 – Integração de circuito final na caixa

A figura B.2 do anexo B contém mais detalhes do processo de criação do protótipo.

Capítulo 8

Conclusões e Trabalho Futuro

8.1 Conclusões

Os avanços na eletrônica e instrumentação serão essenciais para a evolução e difusão dos têxteis inteligentes. Claro que, tal só será possível com uma estreita cooperação entre as várias áreas de estudo e uma das soluções é o desenvolvimento de componentes mais pequenos passíveis de serem integrados facilmente em têxteis, assim como, a utilização de técnicas que permitam incluir os componentes de forma a que o têxtil seja funcional e não incomode o utilizador.

Apesar de já existirem vários produtos que se enquadram nos têxteis inteligentes e são comercializados, existe constantemente o desenvolvimento de novas soluções por parte de grupos de investigação. No entanto, os têxteis inteligentes ainda não são muito publicitados nem utilizados no quotidiano das pessoas.

O principal objetivo deste projeto, foi desenvolver uma solução capaz de adquirir o sinal proveniente de um sensor sob a forma de um lençol e alertar os cuidadores caso o acamado se ausentasse ou o nível de humidade devido a libertação de fluídos fosse superior a 10%. O primeiro passo foi caracterizar o sensor, isto é, perceber o seu funcionamento. Este processo demorou bastante tempo, uma vez que, não existia acesso a nenhuma informação relativa ao funcionamento do mesmo. Inicialmente, pensava-se que as alterações físicas no lençol, como por exemplo alguém se deitar, se traduziria numa variação de resistência e utilizou-se mecanismos como pontes de *wheatstone* e uma combinação de divisores de tensão com um amplificador de instrumentação para quantificar essa variação de resistência, tal revelou-se ineficaz para a deteção de presença. Relativamente à humidade, esta sim era detetável utilizando um divisor de tensão, no qual uma das resistências era o próprio sensor, pois o aumento de líquidos traduz-se numa variação de tensão que era quantificável pelo ADC do ESP32. No entanto, não foi esta a abordagem adotada.

Após se ter verificado que a deteção de presença não influenciava a variação da resistência e depois de uma pesquisa mais detalhada, concluiu-se que uma abordagem possível poderia ser a deteção de variações na capacitância do sensor. Para isso, implementou-se um circuito oscilador com o CI 555 no modo astável e finalmente foi possível fazer a distinção entre o estado ausente e presente com sucesso, pois, quando é exercida pressão no lençol, a capacitância aumenta.

Relativamente à humidade, percebeu-se que esta também se traduzia em variações na capacidade, isto é, quanto maior a humidade, maior a capacidade. A distinção entre o estado de presença com o lençol seco e o estado de presença com o lençol molhado, é possível, pois quando o lençol está molhado a capacidade apresenta valores significativamente mais elevados comparativamente a quando o lençol está seco. A utilização de apenas um circuito para a deteção de pressão e humidade, apresenta a limitação de que por vezes quando o lençol está húmido, não é detetado quando a pessoa se deita ou levanta. No entanto, uma vez que, o objetivo é o cuidador se deslocar ao doente, assim que o lençol fique molhado para trocar a cama e a roupa do acamado, tal limitação não se considerou preocupante, uma vez que, com um circuito mais complexo os custos de implementação seriam maiores, o que se refletiria no preço final do produto.

A caracterização do sensor e aquisição do sinal foi apenas a primeira etapa de um longo projeto. Seguiu-se o desenvolvimento do software a ser colocado no ESP32, responsável por: processar o sinal proveniente do circuito, definir o estado do lençol e comunicar com uma plataforma hospedada na nuvem, de forma a enviar os dados de presença ou ausência, o valor percentual da humidade e o estado de algumas *flags* de comunicação. A plataforma escolhida, devido à facilidade de integração em tempo real e escalabilidade automática foi o *Firebase*, uma plataforma desenvolvida pela Google, que visa simplificar o processo de criação de uma aplicação.

De seguida, foi necessário desenvolver uma aplicação onde os cuidadores têm disponível em tempo real o estado dos pacientes. Para além disso, também notifica os cuidadores em caso de alarme. A aplicação foi desenvolvida em linguagem Dart na *framework* Flutter. Esta escolha, deveu-se à eficiência do Flutter em criar interfaces de utilizador consistentes em diversas plataformas, como iOS e Android, usando o mesmo código-fonte, o que economiza tempo e recursos de desenvolvimento.

Por fim, de forma a criar um protótipo do que seria uma solução final, procedeu-se à impressão 3D de uma caixa onde se colocou o circuito desenvolvido, assim como, de um conector para ligar o sistema ao lençol. Para isso, foi necessário efetuar um projeto prévio no software Eagle, o que conferiu alguma experiência no design de PCB's.

As principais dificuldades sentidas foram na caracterização do sensor como já referido, e principalmente no facto de ter existido um grande estudo e familiarização com todas as tecnologias utilizadas no desenvolvimento do projeto, o que exigiu bastante tempo. Para além disso, a criação da aplicação também exigiu bastante esforço, uma vez que, nunca tinha existido contacto com a criação de aplicações, muito menos em linguagem dart que é utilizada inclusive no mundo profissional.

Para além disso, foi um desafio perceber como o uso concorrente deste sistema podia ser conseguido, assim como a utilização do *Wi-Fi Manager*, que permite que os utilizadores conectem o microcontrolador à Internet, como se de um telemóvel se tratasse.

Os resultados obtidos mediante a realização de testes que reproduziram fielmente as situações da vida real, revelaram a estabilidade do sistema em diferentes cenários. Esses incluíram o lençol no estado seco e desocupado, seco e ocupado durante uma noite de sono, e, por último, o lençol húmido, obtido pela adição de água. Além disso, a existência da funcionalidade de calibração do

sistema permite uma revalidação constante, sempre que necessário, assegurando a sua fiabilidade em diversas situações de uso. Conclui-se, portanto, que o sistema é fiável e apresenta aplicabilidade nos contextos de cuidados de saúde.

8.2 Trabalho Futuro

Um possível trabalho futuro, é integrar o sistema desenvolvido neste projeto num sistema mais abrangente, onde em apenas um *dashboard*, seja possível ver os estados de presença e humidade de vários pacientes de um hospital, por exemplo por quarto ou por piso. Um sistema assim, deixaria de fazer sentido utilizar em ambiente doméstico, pelo que seria mais direcionado exclusivamente para a indústria hospitalar.

Outro dos trabalhos futuros, poderá envolver a adaptação do têxtil inteligente não apenas para a monitorização de pacientes acamados, mas também para a monitorização de animais de companhia mais envelhecidos. Isso permitiria aos donos saber o estado do animal, expandindo assim as possibilidades de aplicação do sistema.

Anexo A

Interface de utilizador

A.1 Menus da aplicação

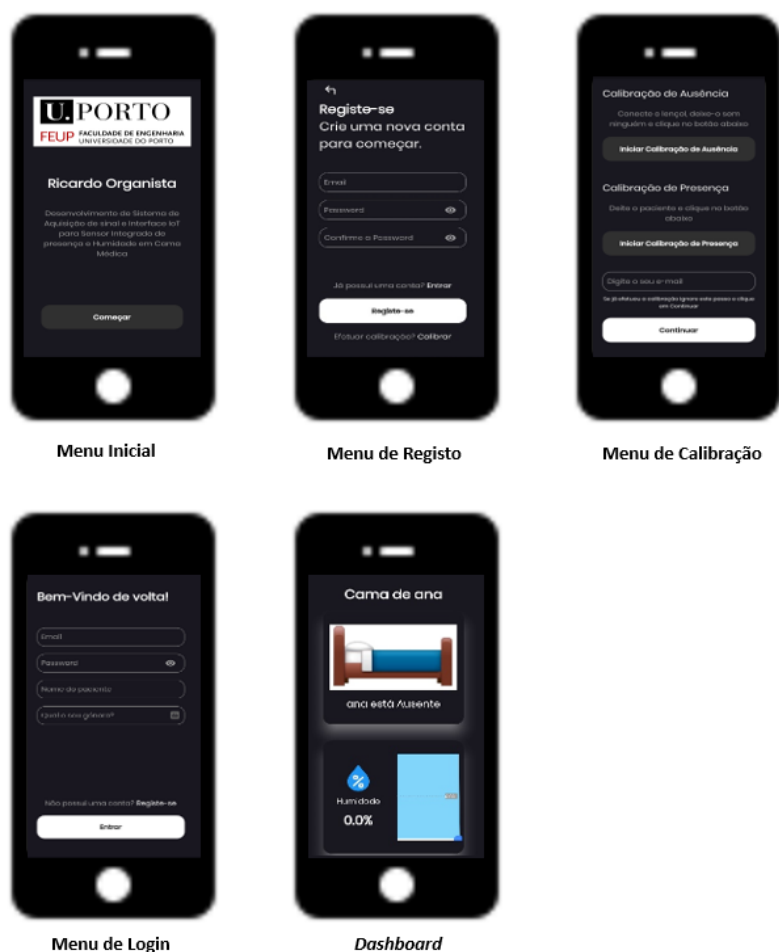


Figura A.1 – Menus da aplicação para smartphone

A.2 Wi-Fi Manager

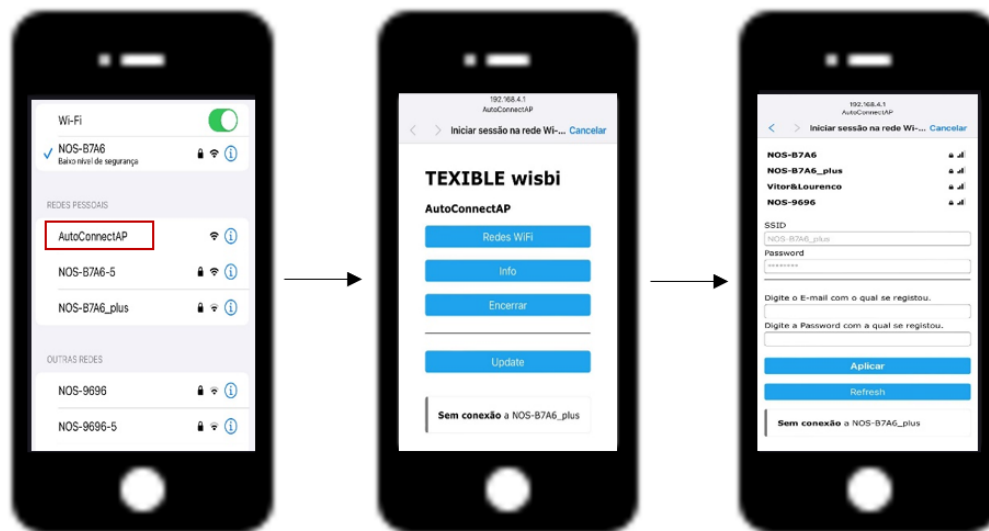


Figura A.2 – Processo de ligação à rede Wi-Fi por parte do utilizador

Anexo B

Implementação e Desenvolvimento do Protótipo

B.1 Conector do lençol

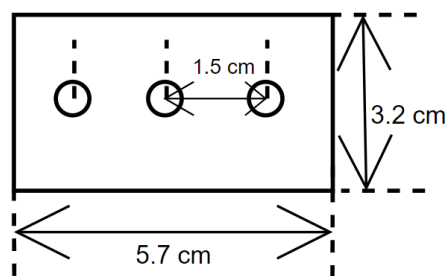


Figura B.1 – Medidas do conector do lençol

B.2 Processo de criação do protótipo

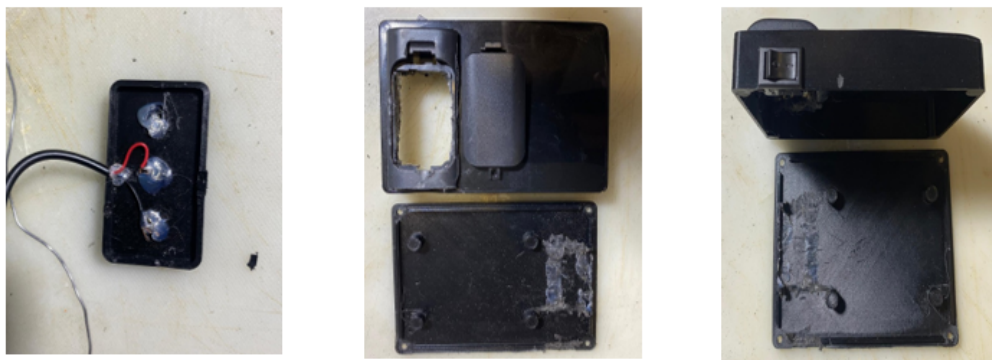


Figura B.2 – Processo de desenvolvimento do invólucro e conector

Referências

- [1] World Health Organization. Ageing and health, outubro 2022. Disponível em <https://www.who.int/news-room/fact-sheets/detail/ageing-and-health>, acessado a última vez em 20/novembro/2023.
- [2] World Health Organization. Dementia, janeiro 2021. Disponível em <https://www.who.int/news-room/facts-in-pictures/detail/dementia>, acessado a última vez em 20/novembro/2023.
- [3] World Health Organization. Evidence profile: urinary incontinence - integrated care for older people. *Guidelines on community-level interventions to manage declines in intrinsic capacity*, página 1, janeiro 2017. Disponível em <https://www.who.int/publications/i/item/WHO-MCA-17.06.08>.
- [4] TEXIBLE. About us. Disponível em <https://www.texible.com/about-us/>, acessado a última vez em 22/novembro/2023.
- [5] TESTEX. Classificação e aplicação de tecidos têxteis inteligentes. Disponível em <https://www.testextextile.com/pt/classifica%C3%A7%C3%A3o-e-aplica%C3%A7%C3%A3o-de-tecidos-de-tecido-inteligente/>, acessado a última vez em 23/novembro/2023.
- [6] Liena Van Langenhove. *Smart textiles for medicine and healthcare*. Woodhead Publishing, First edição, 2007.
- [7] Matteo Stoppa e Alessandro Chiolerio. Wearable electronics and smart textiles: A critical review. julho 2014. Disponível em <https://doi.org/10.3390/s140711957>.
- [8] E. R. Post e M. Orth. Smart fabric, or "wearable clothing". *Digest of Papers. First International Symposium on Wearable Computers*, páginas 167–168, 1997. Disponível em <https://doi.org/10.1109/ISWC.1997.629937>.
- [9] Lieva Van Langenhove e Carla Hertleer. Smart clothing: a new life. *International Journal of Clothing Science and Technology*, 16:64–68, 2004. Disponível em <https://www.emerald.com/insight/content/doi/10.1108/09556220410520360/full/html>.
- [10] Alexandre Ferreira, Fernando Ferreira, e Fernando Oliveira. Têxteis inteligentes—uma breve revisão da literatura. *REDIGE*, 5, 01 2014.
- [11] Fenlan Xu, Xiuyan Li, Yue Shi, Luhai Li, Wei Wang, Liang He, e Ruping Liu. Recent developments for flexible pressure sensors: A review. *Micromachines*, 9(11), 2018. Disponível em <https://doi.org/10.3390/mi9110580>.

- [12] Chong-an Di e Daoben Zhu Yaping Zang, Fengjiao Zhang. Advances of flexible pressure sensors toward artificial intelligence and health care applications. *Royal Society of Chemistry*, páginas 64–68, 2014. Disponível em <https://doi.org/10.1039/C4MH00147H>.
- [13] Jan Meyer, Paul Lukowicz, e Gerhard Troster. Textile pressure sensor for muscle activity and motion detection. páginas 69 – 72, 11 2006. Disponível em <http://doi.org/10.1109/ISWC.2006.286346>.
- [14] U-V. Albrecht, R. Klawunn, e M-L. Dierks. Expectations of new technologies in nursing care among hospital patients in germany - an interview study. *Frontiers in psychology*, páginas 1–3, 2023. Disponível em <https://doi.org/10.3389/fpsyg.2023.1227852>.
- [15] J. Junkin e J.L. Selekof. Prevalence of incontinence and associated skin injury in the acute care inpatient. *Journal of Wound, Ostomy and Continence Nursing*, páginas 260–269, maio 2007. Disponível em <https://doi.org/10.1097/01.WON.0000270820.91694.1f>.
- [16] K. Y. Woo. Management of moisture-associated skin damage: A scoping review. *Adv Skin Wound Care*, página 3, novembro 2017. Disponível em <https://doi.org/10.1097/01.ASW.0000525627.54569.da>.
- [17] H. Strauven, I D’Haeseleer, K. T’Jonck, H. Hallez, V Abeele, P. Crombez, e B. Vanrumste. Towards an ambient support system for continence management in nursing homes: An exploratory study. *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - HEALTHINF*, páginas 438–446, 01 2020. Disponível em <https://doi.org/10.5220/0008963404380446>.
- [18] D. Ghosh, A. Agrawal, N. Prakash, e P. Goyal. Smart saline level monitoring system using esp32 and mqtt-s. Em *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, páginas 1–5, 2018. Disponível em <https://doi.org/10.1109/HealthCom.2018.8531172>.
- [19] Biodevices. Vitaljacket baby. Disponível em <https://www.biodevices.pt/pt/vitaljacket-baby/>, acessado a última vez em 30/novembro/2023.
- [20] Apple. Nike+ipod. Disponível em <https://www.apple.com/ca/ipod/nike/>, acessado a última vez em 30/novembro/2023.
- [21] TEXIBLE. Our projects. Disponível em <https://www.texible.com/projects/>, acessado a última vez em 02/dezembro/2023.
- [22] Portugal têxtil. Nanostim, 2023. Disponível em <https://portugaltexil.com/nanostim-apresenta-prototipo/>, acessado a última vez em 02/dezembro/2023.
- [23] CITEVE. Citeve divulga projetos da área da saúde na medica. Disponível em https://www.citeve.pt/inteligencia_tecnologica/o_que_ha_de_novo/citeve_divulga_projetos_da_area_da_saude_na_medica-dabf6c9c, acessado a última vez em 03/dezembro/2023.
- [24] CITEVE. Sobre o citeve. Disponível em <https://www.citeve.pt/citeve/confianca#Sobre>, acessado a última vez em 03/dezembro/2023.

- [25] CITEVE. Smart-health-4-all. Disponível em https://www.citeve.pt/conteudos/artigos/projeto_mobilizador-95b37e59, acessado a última vez em 03/dezembro/2023.
- [26] TEXIBLE. Wisbi - melhor suporte de cuidados. Disponível em https://idealsafe.pt/wp-content/uploads/2020/06/TEXIBLE-Wisbi-HOME_Brochure_PT.pdf.
- [27] TEXIBLE. Sensors. Disponível em <https://www.texible.com/smart-textiles/sensors/>, acessado a última vez em 03/dezembro/2023.
- [28] Kevin Ashton. That ‘internet of things’ thing, 2009. RFID JOURNAL, disponível em <https://www.itrco.jp/libraries/RFIDjournal-That%20Internet%20of%20Things%20Thing.pdf>.
- [29] ORACLE. What is iot. Disponível em <https://www.oracle.com/pt/internet-of-things/what-is-iot/>, acessado a última vez em 03/dezembro/2023.
- [30] Steve Ranger. What is the history of the internet of things, 2020. Disponível em <http://tinyurl.com/3kj7tchb>, acessado a última vez em 05/dezembro/2023.
- [31] k. Sohraby, D. Minoli, B. Occhiogrosso, e W. Wang. A review of wireless and satellite-based m2m/iot services in support of smart grids. *Mobile Netw Appl*, página 1, agosto 2018. Disponível em <https://doi.org/10.1007/s11036-017-0955-1>.
- [32] S. Dey, A. Roy, e S. Das. Home automation using internet of thing. *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, páginas 1–5, 2016. Disponível em <https://doi.org/10.1109/UEMCON.2016.7777826>.
- [33] Internet Architecture Board. Architectural considerations in smart object networking. páginas 03–09, março 2015. Disponível em <https://www.rfc-editor.org/rfc/rfc7452.txt>.
- [34] Karen Rose, Scott Eldridge, e Lyman Chapin. The internet of things: An overview. *The internet society (ISOC)*, 80:14–22, 2015.
- [35] EMBARCADOS. Modelos de comunicação para iot, 2016. Disponível em <https://embarcados.com.br/modelos-de-comunicacao-para-iot/>, acessado a última vez em 05/dezembro/2023.
- [36] Sweta Bhattacharya, Muhammad Aqeel, e Fahad Ali. A review of security and privacy concerns in the internet of things (iot). páginas 07–11, setembro 2022. Disponível em <https://doi.org/10.1155/2022/5724168>.
- [37] Kateryna Mokliakova, Gautam Srivastava, e Uttam Ghosh et.al. Privacy issues in smart iot for healthcare and industry. páginas 311–315, fevereiro 2022. Disponível em https://doi.org/10.1007/978-3-030-81473-1_15.
- [38] Sumit Singh Dhanda, Brahmjit Singh, e Poonam Jindal. Lightweight cryptography: A solution to secure iot. páginas 2–5, janeiro 2020. Disponível em <https://doi.org/10.1007/s11277-020-07134-3>.
- [39] Aurélio Campilho. *Instrumentação Electrónica - Métodos e Técnicas de Medição*. FEUP edições, primeira edição, 2013.

- [40] Dejan. 555 timer ic – working principle, block diagram, circuit schematics, 2022. Disponível em <https://howtomechatronics.com/how-it-works/555-timer-ic-working-principle-block-diagram-circuit-schematics/>, acessado a última vez em 08/dezembro/2023.
- [41] Endigital. Circuito integrado 555 como multivibrador astável. Disponível em <https://endigital.orgfree.com/sequencial/CI555.htm>, acessado a última vez em 08/dezembro/2023.
- [42] Manual da Eletronica. Ci 555, características e aplicações. Disponível em <https://www.manualdaeletronica.com.br/ci-555-caracteristicas-aplicacoes/>, acessado a última vez em 08/dezembro/2023.
- [43] Wikimedia Commons. Circuit diagram of a standard 555 astable circuit., 2006. Disponível em https://commons.wikimedia.org/wiki/File:555_Astable_Diagram.svg, acessado a última vez em 08/dezembro/2023.
- [44] Fábio Guimarães. Ci 555 – o que é e como funciona, 2017. Disponível em <https://mundoprojetado.com.br/ci-555-o-que-e-e-como-funciona/>, acessado a última vez em 08/dezembro/2023.
- [45] A. Arogbonlo, C. Usma, A.Z. Kouzani, e I. Gibson. Design and fabrication of a capacitance based wearable pressure sensor using e-textiles. *Procedia Technology*, páginas 270–275, agosto 2015. Disponível em <https://doi.org/10.1016/j.protcy.2015.07.043>.
- [46] R.T. Bento, R.W.O. Silva, L.A. Dias, A.F. Filho, e A.J.C. Pitta. Design, development and application of a real-time capacitive sensor for automatically measuring liquid level. *SN Applied Sciences*, páginas 01–06, junho 2019. Disponível em <https://doi.org/10.1007/s42452-019-0770-3>.
- [47] Electricity Magnetism. O campo elétrico no interior de um capacitor de placas paralelas. Disponível em <https://www.electricity-magnetism.org/pt-br/qual-e-o-campo-eletrico-dentro-de-um-capacitor-de-placas-paralelas/>, acessado a última vez em 27/dezembro/2023.
- [48] Condensadores: Materiais dielétricos, permissividade e rigidez dielétrica. Disponível em <https://onlinestores.onsale2024.ru/content?c=capacitor+placas+paralelas&id=9>, acessado a última vez em 27/dezembro/2023.
- [49] Ciência Elétrica. Condensador. princípios e aplicações, 2022. Disponível em <https://youtu.be/lvmlPHX6FQs?t=325>, acessado a última vez em 27/dezembro/2023.
- [50] M. Ferreira e J.A. Moreira. Capacidade e condensadores. *Revista de Ciência Elementar*, páginas 01–02, 2014. Disponível em https://www.fc.up.pt/pessoas/jfgomes/pdf/vol_2_num_2_64_art_capacidadeCondensadores.pdf.
- [51] EEPower. Application of dielectric materials to capacitors. Disponível em <https://eepower.com/capacitor-guide/fundamentals/dielectric-materials/>, acessado a última vez em 27/dezembro/2023.
- [52] John W. Jewett e Raymond A. Serway. *Physics for Scientists and Engineers with Modern Physics*. Sétima edição, 2007.

- [53] S. de Oliveira. *Internet das coisas com ESP8266, Arduino e Raspberry Pi*. Novatec, First edição, 2017.
- [54] A. P. Godse e Dr. D. A. Godse. *Microcontrollers*. Technical Publications, First edição, 2020.
- [55] Loomia Technologies Inc. What is a microcontroller?, 2017. Disponível em <https://www.loomia.com/blog/what-is-a-microcontroller>, acessado a última vez em 10/dezembro/2023.
- [56] TESCA. What is a microcontroller, types, applications, and how does it work?, 2021. Disponível em <https://www.tescaglobal.com/blog/what-is-a-microcontrollers-and-how-does-it-work/>, acessado a última vez em 10/dezembro/2023.
- [57] ST life.augmented. Stm32 32-bit arm cortex mcus. Disponível em <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>, acessado a última vez em 12/dezembro/2023.
- [58] Wikipedia. Esp32 - printed circuit boards. Disponível em <https://en.wikipedia.org/wiki/ESP32>, acessado a última vez em 14/dezembro/2023.
- [59] Last Minute Engineers. Esp-wroom-32 module. Disponível em <https://lastminuteengineers.com/getting-started-with-esp32/>, acessado a última vez em 14/dezembro/2023.
- [60] Last Minute Engineers. Esp32 pinout reference. Disponível em <https://lastminuteengineers.com/esp32-pinout-reference/>, acessado a última vez em 14/dezembro/2023.
- [61] Matthew Sparkes. What does wi-fi stand for? Disponível em <https://www.newscientist.com/question/what-does-wi-fi-stand-for/>, acessado a última vez em 16/dezembro/2023.
- [62] ELSYS. What does wi-fi stand for? Disponível em <https://elsys.com/blog/prefere-wi-fi-ou-cabo-saiba-as-principais-vantagens-de-cada/>, acessado a última vez em 16/dezembro/2023.
- [63] Last Minute Engineers. Create a simple esp32 web server in arduino ide. Disponível em <https://lastminuteengineers.com/creating-esp32-web-server-arduino-ide/>, acessado a última vez em 16/dezembro/2023.
- [64] ElectronicWings. Esp32 wi-fi basics getting started. Disponível em <https://www.electronicwings.com/esp32/esp32-wi-fi-basics-getting-started>, acessado a última vez em 16/dezembro/2023.
- [65] Microdigisoft. Esp32 with wifimanager to manage ssid and password, 2022. Disponível em <https://microdigisoft.com/esp32-with-wifimanager-to-manage-ssid-and-password-no-hard-coding/#>, acessado a última vez em 19/dezembro/2023.

- [66] Arduino e Cia. Como usar o wifimanager com esp32 e esp8266, 2023. Disponível em <https://www.arduinoecia.com.br/como-usar-o-wifimanager-com-esp32-e-esp8266/>, acessado a última vez em 19/dezembro/2023.
- [67] GeeksforGeeks. Network time protocol (ntp), 2022. Disponível em <https://www.geeksforgeeks.org/network-time-protocol-ntp/>, acessado a última vez em 19/dezembro/2023.
- [68] IP Encyclopedia. Ntp, 2021. Disponível em <https://info.support.huawei.com/info-finder/encyclopedia/en/NTP.html>, acessado a última vez em 19/dezembro/2023.
- [69] Ashok Kumar S. *Mastering Firebase for Android Development*. Packt Publishing Ltd., Primeira edição, 2018.
- [70] Doug Stevenson. What is firebase? the complete story, abridged, 2018. Disponível em <https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0>, acessado a última vez em 02/janeiro/2024.
- [71] Login simples e em várias plataformas. Disponível em <https://firebase.google.com/products/auth?hl=pt>, acessado a última vez em 02/janeiro/2024.
- [72] Firebase. Firebase authentication, 2018. Disponível em <https://firebase.google.com/docs/auth>, acessado a última vez em 02/janeiro/2024.
- [73] Cloud Firestore. Store and sync app data at global scale. Disponível em <https://firebase.google.com/products/firestore>, acessado a última vez em 04/janeiro/2024.
- [74] Cloud firestore. Disponível em <https://firebase.google.com/docs/firestore?hl=pt>, acessado a última vez em 04/janeiro/2024.
- [75] Modelo de dados do cloud firestore. Disponível em <https://firebase.google.com/docs/firestore/data-model?hl=pt>, acessado a última vez em 04/janeiro/2024.
- [76] Jose R F Junior. Cloud firestore -> json, 2018. Disponível em <https://www.linkedin.com/pulse/cloud-firestore-jos%C3%A9-r-f-junior/?originalSubdomain=pt>, acessado a última vez em 10/janeiro/2024.
- [77] AWS. O que é o flutter? Disponível em <https://aws.amazon.com/pt/what-is/flutter/>, acessado a última vez em 10/janeiro/2024.
- [78] Eric Windmill. *Flutter in Action*. Manning, Primeira edição, 2020.
- [79] Medium. How flutter dart code gets compiled to native apps, 2022. Disponível em <https://pinedamg.medium.com/how-flutter-dart-code-gets-compiled-to-native-apps-c4612ea0ef0e>, acessado a última vez em 10/janeiro/2024.
- [80] Thomas Bailey e Alessandro Biessek. *Flutter for Beginners*. Packt Publishing Ltd., Segunda edição, 2021.

- [81] Dart. An approachable, portable, and productive language for high-quality apps on any platform. Disponível em <https://dart.dev/>, acessado a última vez em 10/janeiro/2024.
- [82] Rita Salvado, Caroline Loss, Ricardo Gonçalves, e Pedro Pinho. Textile materials for the design of wearable antennas: a survey. páginas 03–07, novembro 2012. Disponível em <https://doi.org/10.3390/s121115841>.
- [83] Firebase. Planos de preços. Disponível em <https://firebase.google.com/pricing?hl=pt>, acessado a última vez em 03/janeiro/2024.
- [84] Random Nerd Tutorials. Esp32: Firebase web app to display sensor readings. Disponível em <https://randomnerdtutorials.com/esp32-esp8266-firebase-web-app-sensor/>, acessado a última vez em 18/janeiro/2024.
- [85] Home Care Innovation. Wisbi plus – smart waterproof mattress protector with sensors socket receiver. Disponível em <https://homecareinnovation.com/en/product/wisbi-plus-smart-waterproof-mattress-protector-with-sensors-socket-receiver>, acessado a última vez em 20/janeiro/2024.