



**POLITECNICO  
SETUBAL**

ANA EMÍLIA  
SEABRA CORREIA

**Desenvolvimento de um Sistema  
Didático Baseado em Raspberry PI  
para o Ensino da Domótica**



Dissertação tese de mestrado  
Mestrado em Engenharia Eletrotécnica e de  
Computadores  
Escola Superior de Tecnologia de Setúbal

**ORIENTADOR**

Professor Doutor Victor Manuel de Carvalho  
Fernão Pires

**CO- ORIENTADOR**

Professor Especialista Sérgio Fernando Pereira  
Delgado de Sousa

ANA EMÍLIA  
SEABRA CORREIA

**Desenvolvimento de um Sistema  
Didático Baseado em Raspberry PI  
para o Ensino da Domótica**

**JÚRI**

*Presidente:* Professor Daniel José Medronho Foito,  
ESTSetúbal/IPS

*Orientador:* Professor Victor Manuel de Carvalho Fernão  
Pires, ESTSetúbal/IPS

*Vogal:* Professor Joaquim José Rodrigues Monteiro, ISEL/IPL



## RESUMO

A presente dissertação tem por objetivo o desenvolvimento de um sistema didático baseado em tecnologia IoT, utilizando a plataforma Raspberry Pi para apoio ao ensino da domótica em contextos de formação técnica e superior. O protótipo concebido integra sensores de temperatura e humidade (DHT22), luminosidade (LDR) e movimento (PIR), bem como atuadores destinados ao controlo de iluminação e estores, possibilitando a simulação de ambientes inteligentes

A arquitetura proposta assenta numa abordagem modular e de baixo custo, recorrendo a programação em Python, comunicação via GPIO e interface web desenvolvida em Flask. Esta combinação permite aos utilizadores monitorizar condições ambientais e interagir com os dispositivos em tempo real, assegurando simultaneamente funcionalidades de automatização, supervisão remota e envio de notificações.

O protótipo concebido constitui um recurso educativo capaz de apoiar metodologias de aprendizagem ativa, nomeadamente a aprendizagem baseada em projetos, promovendo a aquisição de competências nucleares na domótica e nas tecnologias digitais aplicadas a edifícios inteligentes. A dissertação, por sua vez, delineia propostas de melhoria e perspetivas de trabalho futuro orientadas para o alargamento das funcionalidades do sistema e para o reforço da sua maturidade técnica e valor formativo.

**Palavras-chave:** Domótica; IoT; Raspberry Pi; Casas Inteligentes; Plataforma Educativa.

## **ABSTRACT**

This dissertation presents the development of an educational system based on IoT technologies using the Raspberry Pi platform to support the teaching of home automation in technical and higher education contexts. The proposed prototype integrates temperature and humidity sensors (DHT22), a light sensor (LDR), and a motion sensor (PIR), along with actuators for lighting and shutter control, enabling the simulation of smart home environments.

The system architecture follows a modular and low-cost approach, employing Python programming, GPIO-based communication, and a web interface developed using Flask. This setup allows users to monitor environmental conditions and interact with devices in real time, providing automation features, remote supervision, and automatic alerts.

The developed prototype constitutes an effective educational tool that supports active, project-based learning methodologies, fostering essential competencies in home automation and digital technologies applied to smart buildings. The dissertation also outlines improvement recommendations and future work directions aimed at expanding the platform and incorporating additional functionalities, further enhancing its educational value.

**Keywords:** Home Automation; IoT; Raspberry Pi; Smart Homes; Educational Systems.

## LISTA DE FIGURAS

- Figura 1 – Evolução da Domótica
- Figura 2 – Pirâmide de Automação
- Figura 3 – Tecnologia X10
- Figura 4 – Tecnologia LonWorks
- Figura 5 – Esquema de princípio Z-Wave
- Figura 6 – Funcionamento da tecnologia EnOcean
- Figura 7 – Esquema de funcionamento BACnet
- Figura 8 – Representação do protocolo ZigBee
- Figura 9 – Raspberry Pi 3 B+
- Figura 10- Diagrama de blocos com componentes e fluxo de dados
- Figura 11 – Diagrama geral do sistema domótico
- Figura 12 – Diagrama de blocos da arquitetura funcional do sistema domótico
- Figura 13 – Fluxograma da lógica de programação em Python
- Figura 14 – Controlo de estores e luzes no sistema real
- Figura 15 – Detecção de intruso
- Figura 16 – Monitorização das condições meteorológicas internas
- Figura 17 – Gráficos de consumo energético diário
- Figura 18 – Gráficos de consumo energético mensal
- Figura 19 – Representação da Casa Inteligente



## LISTA DE SIGLAS E ACRÓNIMOS

**IOT**- ing. “Internet of Things” – pt. “Internet das Coisas”;

**Gateway** – equipamento utilizado numa rede informática

**SOC**- ing. “System On a Chip” – pt “Sistema Operativo num circuito integrado”

**OSI**- ing.” Open Systems Interconnection”- pt “Ligação de Sistema aberto”

**PLC**- ing. “programmable logic controller”- pt “Programador lógico controlado

**ISO** – ing .” International Organization for Standardization” – pt “Organização Internacional para a Estandardização”



# INDÍCE

<b>CAPÍTULO 1- INTRODUÇÃO</b> .....	12
1.1- Preâmbulo .....	12
1.2- Conceito de domótica .....	12
1.3- IOT- Internet das Coisas .....	15
1.4- Comunicação .....	16
1.5- Objetivo .....	17
1.6- Estrutura da tese .....	18
<b>CAPÍTULO 2- INTRODUÇÃO AO ENSINO ATRAVÉS DA DOMÓTICA</b> .....	19
2.1- Tecnologia X10 .....	21
2.2- Tecnologia CEBus .....	22
2.3- Tecnologia LonWorks .....	22
2.4 - Tecnologia EIB/KNX .....	23
2.5 – Tecnologia ZWAVE .....	24
2.6 – Tecnologia EnOcean .....	25
2.7. - Tecnologia BACnet .....	26
2.8. – Tecnologia ZigBee .....	27
2.9. - Internet das coisas e microprocessadores .....	27
<b>CAPÍTULO 3 - DESENVOLVIMENTO E IMPLEMENTAÇÃO DO PROTÓTIPO</b> .....	30
3.1.- Objetivo Geral .....	30
3.2- Arquitetura do Sistema e Diagrama de Blocos .....	30
3.2.1. Sensores .....	31
3.2.2. Unidade Central de Controlo – Raspberry Pi .....	31
3.2.3. Atuadores .....	31
3.2.4. Interface Web .....	31
3.2.5. Fluxo de Comunicação .....	31
3.4. Apresentação Geral do Protótipo .....	33
3.5. Arquitetura Funcional do Sistema .....	34
3.6. Programação do Raspberry Pi .....	35
3.5. Comunicação e Monitorização .....	37
3.6. Considerações Técnicas e Didáticas .....	37
<b>CAPÍTULO 4 – ENSAIOS E TESTES NO PROTÓTIPO</b> .....	38

<b>4.1. Ensaio a efetuar no protótipo .....</b>	<b>38</b>
<b>4.2. Preparação do Ambiente de Testes .....</b>	<b>38</b>
<b>4.3. Validação Pedagógica do Protótipo .....</b>	<b>39</b>
<b>4.4. Considerações Finais .....</b>	<b>39</b>
<b>CAPÍTULO 5 - CONCLUSÃO .....</b>	<b>45</b>
<b>5.1 Conclusões Finais.....</b>	<b>45</b>
<b>Bibliografia.....</b>	<b>48</b>
<b>Anexos .....</b>	<b>50</b>

# **CAPÍTULO 1- INTRODUÇÃO**

## **1.1- Preâmbulo**

A presente dissertação tem como objetivo a domótica e a sua aplicação como uma forma inteligente e eficiente de controlo de habitações, através da gestão inteligente dos recursos de uma forma automática. Para isso alia os benefícios dos equipamentos eletrónicos aos equipamentos informáticos, trazendo assim conforto, segurança e conectividade às habitações.

A sociedade atual encontra-se em forte crescimento tecnológico, pelo que, a tendência é que os consumos energéticos sejam cada vez maiores, uma vez que a energia elétrica é necessária para a grande maioria das atividades domésticas, comerciais e industriais. Assim, otimizar os consumos energéticos das habitações através da utilização de sistemas de controlo inteligentes, não só permite redução dos consumos, como irá aumentar o conforto dos utilizadores. De referir que para além dos ganhos económicos, esta solução também permite melhoria a nível ambiental e de conforto.

## **1.2- Conceito de domótica**

A Domótica é uma tecnologia relativamente recente e poderá ser responsável pela gestão de todos os recursos habitacionais. Este termo nasceu da fusão da palavra “Domus”, que significa casa, com a palavra “Robótica”, que está ligada ao ato de automatizar, isto é, realizar ações de forma automática. Este conceito de automação surge da vontade de encontrar soluções que forneçam resposta à necessidade do homem de querer realizar o mínimo esforço nas suas atividades diárias e rotineiras, pelo que, a domótica, para além de introduzir conforto e melhoria de vida aos seus utilizadores, introduz ainda novos conceitos, tais como a comunicação e segurança. A utilização dos sistemas de domótica vai de acordo com as necessidades do consumidor ou poderá funcionar através de um modo mais automatizado.

A Domótica é uma forma inteligente de controlo de edifícios, através da gestão dos recursos de uma forma automática ou manual. Para isso alia os benefícios dos equipamentos eletrónicos aos equipamentos informáticos, trazendo assim conforto e segurança às habitações.

A rápida evolução tecnológica das últimas décadas dos sistemas de informação, originou uma sociedade mais exigente em relação ao seu conforto, segurança, gestão de recursos e comunicação. Este desenvolvimento tecnológico veio fortalecer o desvanecimento do edifício estático, fazendo com que o edifício se tornasse dinâmico, tornado deste modo através da conjugação da automação com os sistemas de informação ganhasse uma maior ênfase. [1]

A domótica nos últimos anos tem vindo a expandir-se devido ao grande conforto que

traz aos seus utilizadores, têm surgido diversas empresas especializadas neste tipo de tecnologia, como pode ser observado na figura 1.

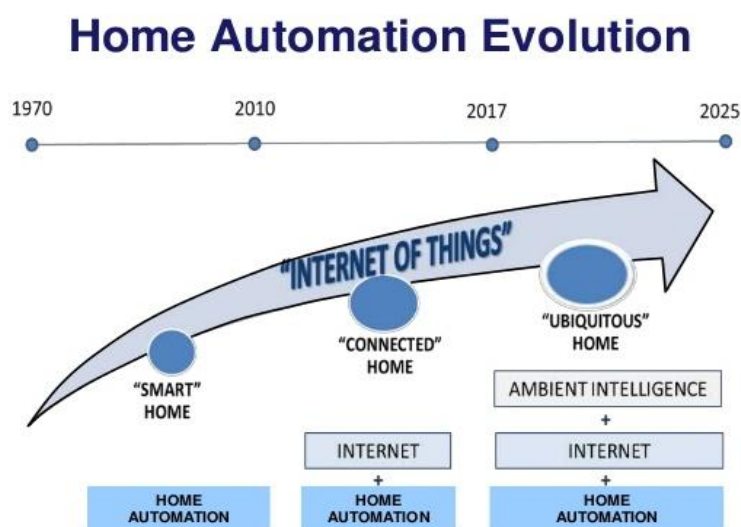


Figura 1 - Evolução da Domótica (adaptação de fonte " Smart Home IOT")

A domótica incorpora sistemas de informação e comunicação em todo o edifício com controlo automatizado, monitorização e gestão de todos os distintos subsistemas ou serviços do edifício, de forma ótima e integrada.

Desta forma, local ou remotamente, é possível monitorizar e controlar totalmente o edifício e informar do seu estado atual em relação a todos os sistemas e, mais importante, comunicar a ocorrência de qualquer situação de alarme, permitindo a sua correção, na maior parte dos casos remotamente. Existem diversos objetivos na domótica, com destaque para os seguintes:

- Elevado nível de conforto e comodidade indo ao encontro às necessidades individuais de cada cliente;
- Realizar de forma integrada e otimizada ações de manutenção, segurança e comunicações.
- Gestão de energia e monitorização de consumos, fortemente explorados pela gestão técnica;
- Evolução e flexibilidade da instalação.

O funcionamento dos sistemas domóticos depende das preferências e necessidades do utilizador, pelo que , existem situações em que a utilização poderá ser passiva onde o utilizador tem uma ação direta e local, por exemplo: interruptores, botões de

pressão, telecomandos, etc, ou então poderão ser sistemas mais avançados e independentes que funcionam através de sensores e atuadores que comunicam entre si, podendo ser programados conforme as necessidades (Climatização – Controlo da temperatura, Estores elétricos – abertura/Fecho dependente da luminosidade).

Através da domótica é possível garantir os seguintes benefícios:

- Segurança – Videovigilância, (Ex: possível fazer com que a habitação pareça estar ocupada quando na realidade não está);

Económico – Gestão eficiente por ex. controlo do sistema de iluminação (Ex. dimmers, sensores de movimento, sensores crepusculares);

- Conforto – Permite o utilizador ter as suas tarefas diárias facilitadas;

No decorrer do último século, devido a um forte desenvolvimento tecnológico, a ideia de edifício estático desapareceu, emergindo na sua linha evolutiva o conceito de edifício dinâmico, em que a tecnologia se conjuga com a informação residente, otimizando as capacidades do edifício na oferta de serviços.

Nesta linha de pensamento distinguem-se 3 períodos:

- Até 1900 – Os edifícios são caracterizados por uma estrutura fixa, em que os principais materiais usados são a pedra e a madeira, e os serviços prestados são reduzidos ao básico, tais como distribuição de águas, esgotos e iluminação.

- Entre 1900 e 1945 – Aparecimento de estruturas em cimento armado e de equipamentos elétricos mais aperfeiçoados, destacando-se o uso generalizado de eletrobombas e motores elétricos.

- Após 1945 até à atualidade – A nível estrutural não se verificam grandes alterações, nas habitações. Pelo contrário nos edifícios fornecedores de serviços, assiste-se ao aparecimento de estruturas dinâmicas, tais como paredes falsas, locais de trabalho em espaço aberto (open space), etc. A nível tecnológico os sistemas de telecomunicações e processamento de informação são cada vez mais avançados

Os requisitos da sociedade atual, a nível de conforto doméstico e avanço da tecnologia industrial, devidos a extraordinária progressão da informática, conduzem a que as soluções devam ser flexíveis, adaptáveis a novas tecnologias e requisitos. Todo o descrito anteriormente deu origem ao Edifício Inteligente.

No entanto este conceito engloba um conjunto muito variado de áreas ligadas à arquitetura de exteriores e interiores, automação, controlo e gestão de edifícios, telecomunicações e infraestruturas de suporte, informática, ecologia, entre muitas outras.

O que se pode denotar é que da inter-relação entre estas diferentes áreas, podem surgir locais de trabalho mais confortáveis, económicos, seguros, ergonómicos e

estimulantes, propiciando formas de interação que levem à prestação de serviços mais eficazes e produtivos.

Mas para se chegar ao ponto atual, foi necessário passar por algumas fases. Aquilo que há alguns anos parecia impensável é hoje uma realidade atingida, graças ao desenvolvimento da microeletrônica, uma área da eletrônica, que nos trouxe um tremendo desenvolvimento tecnológico. Steve Wozniak (co-fundador da Apple Computer), há 25 anos disse que “a generalização do PC conduziria à Automação Doméstica”.

O objetivo de aplicar a filosofia de edifício inteligente às casas de habitação não se encontra tão longe de ser atingido, sendo que com o aparecimento de novos sistemas modulares, a indústria começou a expandir-se e a tentar fazer chegar ao mercado soluções a preços mais acessíveis. Atualmente a Domótica já não é apenas um conceito abstrato, mas uma realidade que se vai tornando uma necessidade.

### **1.3- IOT- Internet das Coisas**

A Internet das Coisas, também conhecida pelo acrónimo IoT, compreende todos os aparelhos e objetos que se encontram habilitados a estarem permanentemente ligados à Internet, sendo capazes de se identificar na rede e de comunicar entre si, e sendo possível alterarem o seu estado através daquele meio, com ou sem o envolvimento ativo do ser humano. Por outro lado têm a capacidade para recolher uma vasta quantidade de informação sobre o que os rodeia. Estes dispositivos podem ser encontrados em veículos, luzes de trânsito, eletrodomésticos, câmaras de vigilância, detetores de condições ambientais, sensores de presença, e dispositivos médicos.

O objetivo de todos estes dispositivos e sobretudo da grande quantidade de dados que são extraídos e resultantes da respetiva interação dos mesmos através da Internet, é utilizar esta informação para benefício do utilizador particular, e/ou da sociedade em geral.

O conceito de Internet das Coisas não é tão recente quanto se pensa, embora aplicado com mais intensidade, nos dias de hoje dado que, começou a ser desenvolvido em 1999 por Gevin Ashton, no Laboratório de Auto-ID do Instituto de Tecnologia de Massachusetts (MIT). As primeiras tecnologias que surgiram foram as dos sensores e identificação por radiofrequência. Entretanto, o conceito já evoluiu para além do que era desenvolvido tradicionalmente, nomeadamente de Máquina a Máquina ou M2M. Assim, passam a ser ligações avançadas de dispositivos, sistemas e serviços que embarcam numa ampla variedade de protocolos, domínios e aplicações.

Em 2020 existem aproximadamente 26 bilhões de dispositivos ligados à internet. Estima-se que, cada indivíduo esteja cercado por 1.000 ou 5.000 objetos com as

características IoT. Este impacto é visível quer na esfera pessoal ou profissional. Sendo que esta última influencia diretamente o mercado e os seus negócios. [5]

As aplicações e dispositivos IoT estão cada vez mais presentes no quotidiano, sendo possível dividi-los em duas grandes áreas: os de quotidiano e os de mercado industrial e /ou infraestruturas.

Nas habitações, esta tecnologia centra-se no conceito amplo da automação, que pode ir desde o sistema de iluminação até ao de segurança. O IOT gera alguns benefícios de onde se destaca a economia dos gastos domésticos, como por exemplo, a conta de luz. Caso ninguém esteja num determinado compartimento, o sistema IoT será capaz de desligar a luz automaticamente, o que evita desperdícios. O mesmo pode dizer-se dos frigoríficos que, ao serem configurados, conseguem informar-nos do prazo de validade dos produtos e, mais uma vez, evitam desperdícios. Ainda no âmbito doméstico, também existem dispositivos da Internet das Coisas capazes de auxiliar e dar assistência a pessoas com deficiência e idosos.

No que diz respeito à gestão da manutenção, a IoT aplica-se aos sensores e tecnologias NFC que podem ser instalados nos equipamentos, ou em qualquer outra área em que exista esta necessidade. A principal vantagem é a possibilidade de realizar atividades como a manutenção preventiva em ativos de alta criticidade. Outros benefícios são: Redução dos custos; Otimização dos processos; Monitorização remota e em tempo real; Aumento da vida útil dos ativos (via manutenção preditiva); Assertividade e segurança nos dados internos para análise de produtividade.

#### **1.4- Comunicação**

Para existir comunicação, é necessário um emissor, uma mensagem e um recetor. Um protocolo de comunicação descreve formalmente o formato da mensagem a ser transmitida e as respetivas regras de transmissão relativamente à sintaxe, semântica e sincronização da comunicação.

Em 1984 foi desenvolvido pela Internacional *Organization for Standardization* (ISO) uma norma de comunicações entre redes de computadores distribuído por sete camadas, (física, lógica, rede, transporte, sessão, apresentação e aplicação), nas quais é possível realizar comunicação entre máquinas distintas e definir diretivas genéricas para a elaboração de redes de computadores independente da tecnologia utilizada, sejam essas redes de curta, média ou longa distância.

Este modelo tem o nome de *Open Systems Interconnection* (OSI). Neste modelo as três primeiras camadas são a base para definição das características da rede de dados, não existindo interação com o utilizador.

A camada física, tem por função transmitir corretamente um bit através do meio físico de interligação, sendo as características de propagação do sinal muito importantes neste nível. A camada de rede, em oposição à camada lógica, considera a rede como uma entidade lógica que interliga máquinas independentemente das redes físicas a

que estão ligadas. Este nível é também responsável por suportar os mecanismos necessários ao encaminhamento dos pacotes de dados entre quaisquer sistemas que a ele se encontram ligados, independentemente da rede física. A camada de transporte é responsável pela transmissão de mensagens que permite a comunicação entre utilizadores finais e o sistema de comunicação. A camada de apresentação permite compatibilizar a informação entre sistemas diferentes uma vez que a representação de dados relativa a cada sistema, depende de diversos fatores como: arquitetura dos processadores, unidades aritméticas, linguagens, etc. Por último, a camada de aplicação engloba as funcionalidades relativas às aplicações.

Os protocolos mais comuns usados em domótica apenas possuem as camadas física, lógica e de rede passando de imediato para a camada de aplicação.

Na figura 2 apresenta-se a descrição de alguns protocolos com aplicabilidade domótica.

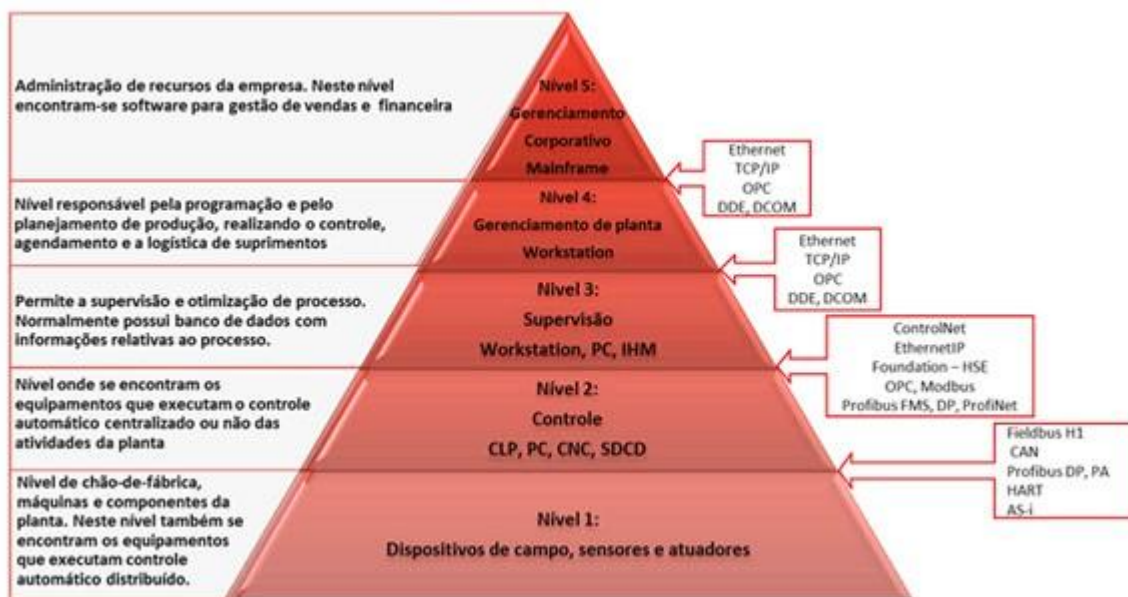


Figura 2 – Pirâmide [2] ( <https://www.empoweredautomation.com/what-is-the-five-layer-automation-pyramid?>)

## 1.5- Objetivo

Esta dissertação apresenta o desenvolvimento de um sistema domótico baseado em IoT, utilizando a plataforma Raspberry Pi para o controlo inteligente de uma moradia unifamiliar. O objetivo principal consistiu na criação de um protótipo funcional capaz de gerir iluminação, estores e condições ambientais de forma remota e automatizada.

Além da vertente técnica, este trabalho foi concebido com forte ênfase educacional, tendo em vista a sua aplicação em ambientes de ensino técnico e superior. A

simplicidade da arquitetura, a programação acessível em Python e a utilização de componentes de baixo custo tornam o projeto ideal para fins pedagógicos, contribuindo para a aprendizagem prática de conceitos como automação, eletrônica, programação e integração de sistemas IoT.

A utilização do sistema no ensino, demonstraram elevada fiabilidade e eficiência, avaliação do protótipo demonstra um elevado nível de fiabilidade, confirmado pela estabilidade contínua do sistema durante os testes funcionais. Os sensores (DHT22, LDR, PIR e reed switch) apresentaram leituras consistentes validando também a aplicabilidade didática do protótipo em contextos de formação técnica e engenharia. No que concerne a eficiência, a execução do código demonstrou baixo uso de CPU e memória, assegurando processamento rápido e eficaz mesmo com múltiplas leituras simultâneas.

A arquitetura modular e o uso de protocolos eficientes (MQTT/HTTP) permitiram escalabilidade, facilitando a integração de novos sensores e atuadores sem reestruturação significativa. A interface desenvolvida demonstrou ergonomia e rapidez de interação, contribuindo para uma experiência de utilização eficiente e intuitiva.

## **1.6- Estrutura da tese**

O **primeiro capítulo** apresenta uma breve introdução à domótica.

O **segundo capítulo** pretende descrever de um modo sucinto a introdução ao ensino da domótica, bem como os fabricantes existentes, os softwares mais utilizados e hardware necessário.

O **terceiro capítulo** centra-se no desenvolvimento e implementação do protótipo, com ênfase na montagem do mesmo e programação

O **quarto capítulo** centra-se na análise dos resultados, verificando se os mesmos vão de encontro ao pressuposto inicial.

O **quinto capítulo** apresenta as conclusões e considerações finais obtidas após o desenvolvimento da presente dissertação. Será apresentada a possibilidade de desenvolvimentos futuros.

# **CAPÍTULO 2- INTRODUÇÃO AO ENSINO ATRAVÉS DA DOMÓTICA**

## **2.1- Preâmbulo**

A domótica, enquanto tecnologia de automação aplicada a edifícios, evoluiu para além do seu papel funcional em residências inteligentes, afirmando-se como uma poderosa ferramenta de ensino nas áreas da eletrónica, automação, programação e tecnologias de informação. O seu potencial educativo reside na capacidade de transformar conceitos teóricos em experiências práticas, proporcionando aos alunos um ambiente de aprendizagem realista, dinâmico e interdisciplinar.

Integrar a domótica no ensino permite explorar sistemas automatizados de forma tangível, utilizando sensores, atuadores, microcontroladores e interfaces gráficas. Esta abordagem facilita a compreensão de temas complexos como redes de comunicação, lógica de controlo, gestão de energia, sustentabilidade e Internet das Coisas (IoT), estimulando o pensamento crítico e a resolução de problemas.

A utilização de plataformas abertas e de baixo custo, como o Raspberry Pi e o Arduino, aliada a linguagens de programação acessíveis como Python, torna a implementação de sistemas domóticos viável em contextos educativos com recursos limitados. Além disso, permite que os estudantes projetem, montem e programem os seus próprios protótipos, incentivando a autonomia, a criatividade e o trabalho colaborativo.

Do ponto de vista pedagógico, o ensino com base na domótica aproxima-se de metodologias ativas, como a aprendizagem baseada em projetos (PBL – Project-Based Learning), promovendo uma ligação direta entre teoria e prática. A resolução de desafios reais relacionados com conforto, segurança ou eficiência energética estimula o envolvimento dos alunos e reforça a aplicabilidade dos conhecimentos adquiridos.

Neste contexto, a domótica não deve ser vista apenas como uma tecnologia de conforto, mas como uma plataforma educativa estratégica que prepara os futuros técnicos e engenheiros para os desafios da Indústria 4.0 e da sociedade digital.

A evolução da domótica ao longo das últimas décadas não só transformou o modo como habitamos os espaços, como também abriu novas oportunidades no contexto educativo. A automatização residencial, outrora vista como uma tecnologia de luxo, passou a ser uma ferramenta de ensino poderosa para desenvolver competências em áreas como eletrónica, automação, programação, redes de comunicação e Internet das Coisas (IoT).

Neste capítulo, propõe-se uma introdução aos principais conceitos, normas e tecnologias associadas à domótica, com o objetivo de contextualizar a sua aplicação em projetos educacionais. A domótica, ao integrar sensores, atuadores, controladores e interfaces gráficas, permite aos alunos experimentar diretamente os princípios da

engenharia e da computação num ambiente prático e motivador. O uso de plataformas acessíveis, como o Raspberry Pi e sensores de baixo custo, reforça o seu potencial pedagógico, facilitando a criação de protótipos funcionais que ilustram o funcionamento de sistemas inteligentes de controlo.

Para além do seu valor tecnológico, a domótica oferece um cenário multidisciplinar ideal para a aprendizagem ativa, colaborativa e baseada em problemas. Ao explorar diferentes tecnologias – como KNX, ZigBee, Z-Wave ou EnOcean – os estudantes são desafiados a analisar, comparar e implementar soluções de automação, compreendendo não apenas os aspetos técnicos, mas também as implicações energéticas, ambientais e sociais dos sistemas que projetam.

Assim, este capítulo fornece o alicerce teórico necessário para a compreensão das diferentes tecnologias de domótica, que servirão de base para o desenvolvimento do protótipo educativo apresentado nos capítulos seguintes.

Inicialmente, os sistemas domóticos foram dominados por produtores proprietários, ou seja, produtos que pertenciam a uma entidade/marca, não existindo normas (standards), para a criação destes sistemas. Com a evolução da tecnologia, foram surgindo movimentos de uniformização, com objetivo central de permitir a compatibilidade entre produtos de diferentes fabricantes, deste modo foi possível promover reduções de custos, permitindo deste modo, uma expansão célere destes sistemas. Convém realçar que a escolha de produtos de certos fabricantes pode acarretar alguns riscos, porque este fornecedor pode abandonar o negócio e caso surja uma avaria, poderá ser difícil substituir o equipamento, o que diretamente representa um risco associado ao investimento efetuado, além de que a tecnologia poderá ficar obsoleta.

Em contrapartida, o facto de existirem equipamentos normalizados, que são fabricados por diversas empresas, faz com que os equipamentos sejam compatíveis independente da marca, minimizando assim o risco associado ao investimento.

Os sistemas de comunicação que dominam o mercado são os seguintes: X10, CEBus, EIB/KNX e, LonWorks, sendo que atualmente o sistema de comunicação mais adotado é o KNX, por ser uma tecnologia mais madura, normalizada a nível internacional e com uma vasta implantação no mercado [6]

No início, o modo de controlar os pontos de utilização das moradias era do tipo industrial, em que todas as entradas e saídas eram aglomeradas no quadro elétrico e controladas por um *programmable logic controller* (PLC). Atualmente, existem vários protocolos de *bus* que através de diferentes topologias e meios de distribuição nos permitem controlar de modo mais ou menos descentralizado a maioria dos equipamentos da habitação.

Os protocolos baseados em standards internacionais não específicos de um só fabricante e com um software de programação comum são designados de protocolos

abertos, como por exemplo: Zigbee ;modbus, Zwave, X10, DALI, EnOcean, BacNet e Modbus.

No mercado da domótica, e dada a sua dimensão, estão disponíveis um conjunto de protocolos abertos e também proprietários, como o LONworks.

## 2.1- Tecnologia X10

A tecnologia X10 foi desenvolvida nos anos 70, tendo sido o seu protocolo normalizado em 1997. A grande vantagem desta tecnologia é o seu baixo preço e grande variedade de equipamentos que tornam o X10 como um dos protocolos de domótica mais divulgado no mundo. Este sistema permite facilmente implementar um sistema domótico numa instalação já existente (Figura 3).

O X10 usa a rede de distribuição de energia elétrica como meio de comunicação entre dispositivos, sendo esta a sua maior vantagem, pois dispensa o uso de cabos adicionais. Uma desvantagem prende-se com o facto de a comunicação ser relativamente lenta, não permitindo ativação simultânea de diversos dispositivos e tendo também um número limitado de funções.

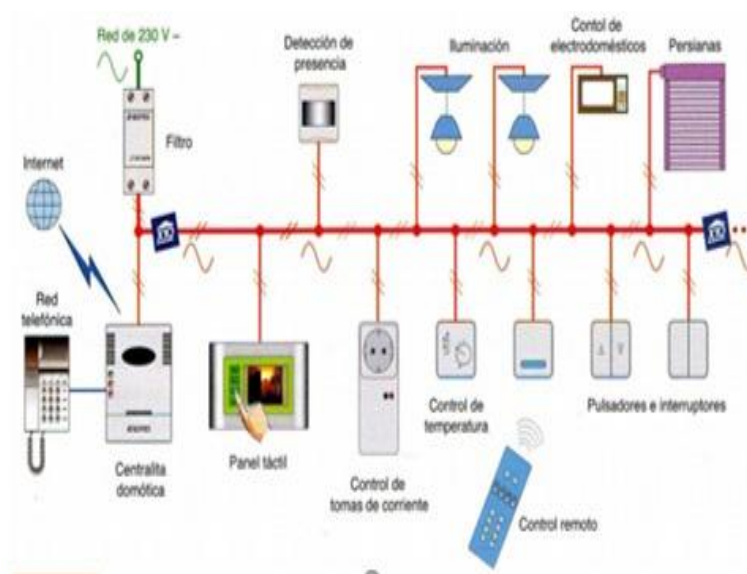


Figura 3 - Tecnologia X10 ( " Smarter Home Automation")

O princípio de funcionamento passa pela instalação de um atuador junto ao equipamento que é acionado num qualquer outro ponto da instalação elétrica ligado à rede com um recetor de rádio frequências (RF), isto é, o utilizador através de um comando RF clica no botão correspondente ao aparelho a controlar.

Todos os atuadores presentes na instalação da habitação recebem um endereço único, constituído por um código de habitação (de A a P) e um código de equipamento (de 1 a 16), existindo uma limitação de 256 equipamentos. Esta simplicidade de

endereçamento torna-se num ponto fraco da solução, porque no caso de edifícios de apartamentos, pode surgir a interferência entre instalações.

Atualmente é possível efetuar aplicações mais complexas com o protocolo X10, sendo que existem no mercado interfaces RS232 que permitem a ligação a um computador e assim criar funções lógicas ou horárias e também ligar a alguns sistemas de alarme ou climatização.

## **2.2- Tecnologia CEBus**

Relativamente ao CEBus, este é um protocolo muito poderoso e complexo que foi criado em 1984 e normalizado em 1995. Este protocolo utiliza as camadas Física, Ligação de Dados, Rede e Aplicação do modelo OSI. Apresenta ainda algumas semelhanças com outras tecnologias nomeadamente o EIB/KNX. Apesar de ser capaz de suportar vários meios de comunicação, como rede elétrica, cabo coaxial, infravermelhos, fibra ótica, cabo de cobre de par entrançado e radiofrequência, a tecnologia CEBus apresenta custos elevados e possui uma cota de mercado bastante baixa. Isto deve-se em parte, à concorrência com o X10, que possui preços bastante mais acessíveis e tem uma larga divulgação.

## **2.3- Tecnologia LonWorks**

A tecnologia LonWorks foi desenvolvida por uma empresa norte-americana, a Echelon Corp., no ano de 1993. Esta tecnologia é constituída por um circuito integrado VLSI chamado Neuron, que é um microcontrolador que pode ser programado em linguagem C, orientada a eventos e objetos, inclui memória RAM e ROM, e, principalmente, suporte ao protocolo de comunicação LonTalk.

Os Neuron Chips, são microcontroladores de pequena dimensão e com preço acessível face à concorrência, para a ligação à rede, sendo necessário incluir apenas um transceptor (Transmissor-Recetor) adequado, tornando viável que a tecnologia seja integrada ao dispositivo sem aumento de tamanho. O protocolo LonTalk, foi aberto aos fabricantes, no final da década de 90, permitiu que o fabrico de equipamentos para esta tecnologia pudesse ser realizado por diversos fabricantes. O protocolo LonTalk é a base da comunicação em rede da tecnologia LonWorks. O LonTalk é constituído por seis camadas distintas: Física, Enlace, Rede, Transporte, Sessão e Apresentação.

Dos protocolos proprietários, o LON é o que tem maior relevância, não só por ser dos mais antigos, mas pelas funcionalidades, fiabilidade e número de instalações em funcionamento existentes.

O LON foi inicialmente batizado com o nome de LONtalk, e evoluiu para um standard internacional, sendo atualmente um protocolo aberto. Este protocolo é suportado em cabo de bus, powerline, rádio frequência e ethernet. A estrutura da tecnologia Lonworks é apresentada na figura 4.

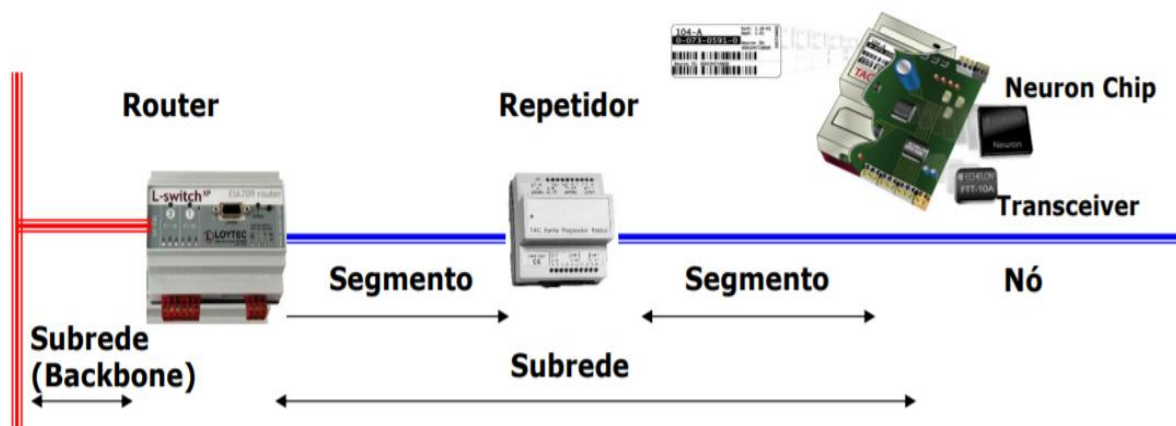


Figura 4- LonWorks (“Smarter Home Automation”)

## 2.4 - Tecnologia EIB/KNX

A tecnologia KONNEX (KNX) foi criada a 14 de Abril de 1999, através da junção dos sistemas EIB, Batibus e EHS. Inicialmente esta tecnologia tinha como principal objetivo a criação de uma única norma europeia para a automação de edifícios.

Esta tecnologia também conhecida como KNX, é um protocolo reconhecido mundialmente e utilizado para aplicações de controlo de edifícios na área da domótica, abrangendo um campo tão vasto como: iluminação e controlo de estores; sistemas de segurança, aquecimento, ventilação, ar condicionado, vigilância, alarme, controlo de água; gestão energética, medição inteligente, bem como eletrodomésticos, áudio/vídeo.

A comodidade, eficiência e segurança dos sistemas domóticos levam a uma cada vez maior procura e avanço tecnológico dos mesmos.

O KNX apresenta um elevado custo de hardware e software. Contudo, esta é uma tecnologia totalmente aberta e garante um funcionamento com produtos de diferentes marcas e modelos desde que sejam certificados e identificados pelo logótipo KNX. Devido ao facto de contemplar vários meios de comunicação (rede elétrica, par entrelaçado, rádio frequência, infravermelhos e Ethernet) e de existir a possibilidade de incluir gateways para outros meios, torna-se um sistema flexível. A taxa de transferência deste sistema varia consoante o meio de comunicação utilizado: na rede elétrica situa-se nos 1200 bps e esta taxa aumenta quando se utiliza rádio frequência e par entrelaçado Este sistema é bastante fiável, mas tem a desvantagem da sua implementação ter, normalmente, um custo elevado.

De acordo com a associação KNX as principais vantagens do protocolo são, para além da standardização:

- A certificação de todos os produtos de modo a incluir o seu logo; A qualidade elevada dos mesmos ao ser obrigatório responder à norma ISO 9001;
- Um único software para parametrização (ETS -Engineering Tool Software );
- A possibilidade de ser utilizado como o único protocolo num edifício;
- A interoperacionalidade com os outros protocolos;
- A independência em relação ao hardware e software;
- O funcionamento em diferente meios físicos.

O modo de parametrização, através do software próprio, ETS, permite definir um elevado número de funções para as instalações, mas exige formação especializada e tem um custo significativo.

A associação KNX, ciente desta realidade lançou este ano uma versão mais simplificada, o ETS Inside, que com um custo muito inferior e uma interface mais intuitiva permite que as instalações possam ser parametrizadas com maior rapidez. No entanto apresenta um menor número de funcionalidades

## **2.5 – Tecnologia ZWAVE**

O protocolo Z-Wave define-se como um protocolo interoperável, sem fios, baseado em RF para controlar e monitorizar aplicações em ambientes residenciais e pequeno comércio. Apesar do objetivo de controlar e monitorizar pequenas aplicações ser atingido, não são possíveis aplicações de vídeo ou imagem. A frequência de transmissão é de 868.42MHz para o sistema europeu e de 908MHz para o americano, permitindo a sua utilização livre. A utilização destas frequências evita a interferência com os 2.4GHz dos equipamentos Wi-fi e Bluetooth, no entanto, cai na frequência de alguns equipamentos sem fios existentes nas habitações, como por exemplo os telefones sem fios. Uma característica do protocolo é que todos os equipamentos são repetidores de sinal, o que permite atingir todas as áreas das habitações desde que cada um dos dispositivos esteja ao alcance de outro. O limite de equipamentos por cada rede é de 232, mas podem ser acopladas várias redes, garantido assim que se aglomeram todos os equipamentos de uma moradia. Cada uma das redes tem o seu Network ID (32bits) e cada dispositivo tem o seu Node ID (8bits).

Para iniciar uma instalação, é necessário atribuir endereços a cada equipamento e efetuar o emparelhamento com as respetivas funções. Este é um processo simples que é realizado utilizando apenas as teclas presentes nos dispositivos. A alimentação dos equipamentos é garantida por uma pilha interna e pela tensão da rede à qual estão ligados. A figura 5 pretende ilustrar o funcionamento.





Figura 6– EnOcean (“EnOceanAlliance.org”)

## 2.7. - Tecnologia BACnet

Building automation and controls network (BACnet), desenvolvido pela ASHRAE, começou a ser definido em 1987, teve o seu primeiro standard em 1995 e foi criada a ISO16484-5 em 2003. O propósito inicial do BACnet foi agregar os diferentes sistemas utilizados na América do Norte de modo a poder controlar toda uma instalação a partir de um único posto de trabalho.

As áreas de aplicação deste protocolo abrangem o aquecimento, ventilação e ar condicionado, detecção de incêndio, controlo de iluminação, intrusão, elevadores e sistemas de visualização. As razões de sucesso deste protocolo passam pela arquitetura adaptável, o modelo ser facilmente alterado, não depender de tecnologia atual e quantidade de fabricantes presentes no protocolo e na associação. A figura 7 ilustra o funcionamento deste sistema.

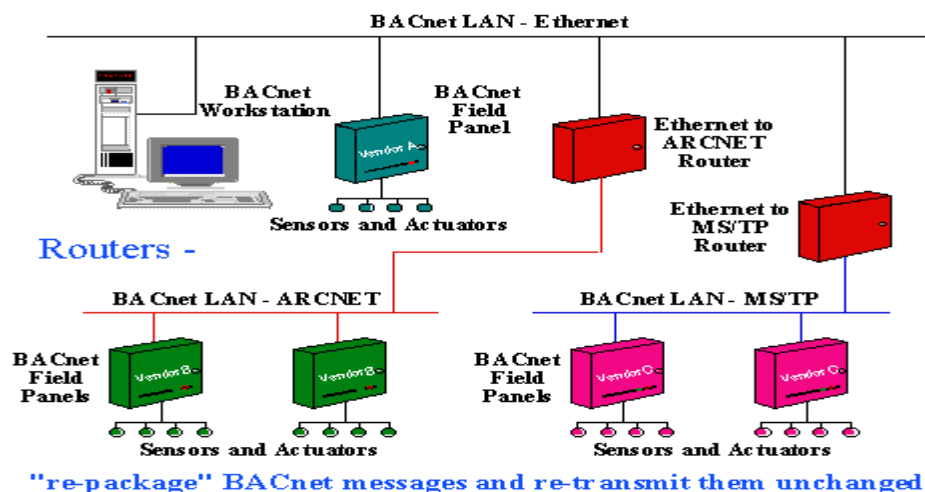


Figura 7– Esquema de funcionamento BaCnet, (BACnet.org)

## 2.8. – Tecnologia ZigBee

O protocolo ZigBee, foi criado com base em especificações que pretendiam limitar o custo por dispositivo e simplificar a sua aplicação quando comparado com as restantes tecnologias wireless. O alcance dos dispositivos está restringido de 10 a 100 metros em linha de vista, na gama de frequências de 868 MHz no espaço europeu. A taxa de transmissão de dados é de 20Kbit/s, que é suficiente para as funções simples de comando dos equipamentos domésticos. A topologia utilizada é do tipo malha, que permite de uma forma simples ultrapassar o alcance reduzido de cada dispositivo. Este protocolo é ideal para remodelações e expansões em habitações, quando o fator preço é decisivo.



Figura 8 – ZigBee, [8] (Connectivity Standards Alliance )

## 2.9. - Internet das coisas e microprocessadores

Microcontroladores são microprocessadores orientados para controlo de equipamentos e estão presentes na maior parte dos sistemas eletrónicos do nosso quotidiano.

Um microcontrolador é um computador que possui um processador (normalmente até 32 bits), memória de programa (Flash), de dados (RAM) e dispositivos de entrada e saída de informação sendo por isso autónomo. Ao invés dos computadores pessoais que estão preparados para executar uma grande variedade de operações, os microcontroladores têm uma utilização específica, no entanto apresentam as seguintes vantagens: Inclusão de diversos periféricos; Inclusão de memória de dados e de programa no mesmo circuito integrado e possibilidade de o programador ter acesso direto aos pinos do circuito;

A escolha de um microcontrolador deve ter em linha de conta a funcionalidade pretendida, o custo, consumo de energia e software necessário ao desenvolvimento das aplicações. Uma vez que a domótica necessita de um elevado número de dispositivos que vão necessitar de microcontroladores, o custo deve ser reduzido, e deve ser contabilizado na escolha a expansibilidade futura do sistema.

As arquiteturas Advanced RISC Machine (ARM), Microprocessor without Interlocked Pipeline Stages (MIPS) e Atmel AVR são as mais utilizadas. No caso da domótica, cujo fulcro são sensores e atuadores, a velocidade não é a característica mais importante.

No caso de uma habitação inteligente, cujo supervisor controla vários nós, são necessários mais recursos, pelo que se optou pelo Raspberry pi 3 como solução domótica de baixo custo, mas com capacidade de expansão e possibilidade de integração com outros sistemas. Ao ser utilizado um SOC de baixo custo, é possível implementar sistemas em rede com um elevado número de dispositivos abrindo caminho para o adequado suporte à criação de ambientes inteligentes.

O System On Chip (SOC) engloba todos os componentes de um computador, ou qualquer outro sistema eletrónico num circuito integrado (chip). Geralmente contém: funções digitais, analógicas, mistas e de Radiofrequência – RF. A sua aplicação principal é na área de sistemas embebidos.

As características de um SOC assemelham-se às de um microcontrolador. Os microcontroladores possuem geralmente, RAM inferior a 100 K (apenas poucos KBytes), e são sistemas de chip único. SOC é designado para utilização em processadores mais potentes, capazes de executarem programas como o Windows ou o Linux, nas quais necessitam de memórias externas (flash, RAM) para funcionarem, e que são usados com vários periféricos acoplados. A grande aplicabilidade dos sistemas SOC centra-se no fator de integração de componentes no chip para reduzir os custos de fabricação e tornar disponíveis sistemas mais compactos. Quando não é possível para construir uma determinada aplicação SoC, a alternativa é o sistema de encapsulamento (System in package (SiP), em inglês), abrange vários chips num único encapsulamento.

O Raspberry Pi é uma placa de desenvolvimento baseada num SOC de baixo custo e que tem o tamanho de um cartão de crédito desenvolvido no Reino Unido pela Fundação Raspberry Pi. Para a sua utilização basta ligar um teclado e um rato, e de seguida ligar tudo a um monitor. A função primordial do Raspberry Pi é oferecer uma alternativa barata, prática e acessível para que os indivíduos de todas as idades desenvolvam as capacidades da computação facilitando a aprendizagem de programação em linguagens como Scratch e Python. Apesar do tamanho diminuto, o Raspberry Pi permite também navegação na internet, reprodução de conteúdos multimédia, criação de conteúdos em forma de texto, imagens e jogos.

O Raspberry Pi surgiu em 2006, baseado no microcontrolador Atmel ATmega644. Os esquemas e layout de PCB foram disponibilizados ao público pela Fundação trustee Eben Upton , que reuniu professores, acadêmicos e admiradores de programação para criar um computador que motivasse as crianças a desenvolverem algo criativo. O computador é inspirado no Acorn BBC Micro de 1981, sendo que o Modelo A, Modelo B e Modelo B+ são referências aos modelos originais de escolaridade do computador BBC Micro britânico, desenvolvido pela Acorn Computers. A primeira versão de arquitetura ARM do Raspberry Pi foi montada num pacote do mesmo tamanho de uma memória stick USB, possuía uma porta USB numa extremidade e uma porta HDMI na outra.

O **Raspberry Pi 3 modelo B+** é o um produto da família **Raspberry Pi**. O coração da PCB é o processador Broadcom BCM2837B0 de 4 núcleos que opera a **1,4 GHz**. O sistema é montado numa versão com um dissipador de calor incorporado, para que a refrigeração do Raspberry seja fácil. A característica que o diferencia dos modelos anteriores é a introdução do WiFi de banda dupla, que permite a comunicação com redes que operam na frequência de 2,4 GHz ou 5 GHz (IEEE 802.11.b/g/n/ac), esta versão também suporta o Bluetooth 4.2.



*Figura 9 – Raspberry Pi 3 + (Fundação Raspberry Pi)*

O **modelo B+** da **Raspberry Pi 3** , apresentado na figura 9, também incorpora o Gigabit Ethernet, contudo a comunicação ainda é limitada pela velocidade máxima do USB 2.0. Este modelo é permite alimentar o equipamento através do **PoE** (Power over Ethernet) sendo, contudo, necessário um HAT especial.

## CAPÍTULO 3 - DESENVOLVIMENTO E IMPLEMENTAÇÃO DO PROTÓTIPO

O Capítulo 3 da presente dissertação descreve de forma detalhada o processo de conceção, desenvolvimento e implementação de um protótipo funcional de um sistema domótico, com foco simultâneo na vertente técnica e pedagógica.

### 3.1.- Objetivo Geral

O principal objetivo deste capítulo foi desenvolver um sistema domótico baseado em tecnologia IoT (Internet of Things), recorrendo à placa Raspberry Pi como unidade de controlo central, permitindo simular o funcionamento de uma habitação inteligente com controlo automatizado de iluminação, estores e climatização. Este protótipo foi idealizado para aplicação em ambientes de ensino técnico e superior, permitindo aos alunos uma aprendizagem prática e aplicada dos conceitos teóricos.

### 3.2- Arquitetura do Sistema e Diagrama de Blocos

Para representar de forma clara a estrutura funcional do sistema, foi elaborado um **diagrama de blocos** (Figura 10), que ilustra os principais componentes e o fluxo de dados entre eles. Este tipo de diagrama é essencial para a compreensão sistémica do projeto, pois mostra como os diferentes subsistemas interagem entre si, desde a recolha de dados até à atuação sobre os dispositivos da casa.

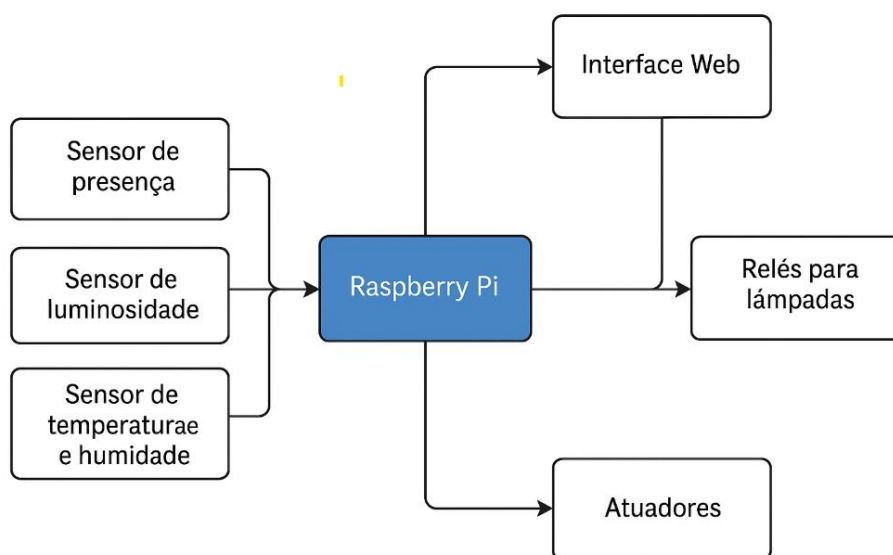


Figura 10– Diagrama de blocos com componentes e o fluxo de dados

A arquitetura funcional está dividida nos seguintes blocos principais:

### **3.2.1. Sensores**

Os sensores utilizados (PIR, LDR e DHT22) são responsáveis pela recolha de dados do ambiente:

- Sensor de presença (PIR): deteta movimento em áreas monitorizadas.
- Sensor de luminosidade (LDR): mede a intensidade da luz ambiente.
- Sensor de temperatura e humidade (DHT22): recolhe dados ambientais para conforto térmico.

### **3.2.2. Unidade Central de Controlo – Raspberry Pi**

O Raspberry Pi 3 modelo B+ atua como o “cérebro” do sistema. Está programado em Python e executa a lógica de controlo, recebendo dados dos sensores, processando essas informações e ativando os atuadores conforme as condições pré-definidas.

Esta unidade também hospeda o servidor web desenvolvido com Flask, que permite o controlo e monitorização remota via interface gráfica acessível através de qualquer navegador.

### **3.2.3. Atuadores**

São os dispositivos que realizam ações físicas:

- Relés para lâmpadas
- Motores para estores
- Saídas digitais para sistema de climatização.

### **3.2.4. Interface Web**

Desenvolvida em ambiente local, esta interface permite ao utilizador:

- Monitorizar dados em tempo real (temperatura, humidade, luz, presença)
- Controlar remotamente os dispositivos (luz, estores)
- Receber notificações via e-mail (por exemplo, deteção de presença)

### **3.2.5. Fluxo de Comunicação**

O fluxo de funcionamento é cíclico e baseado em eventos:

1. Os sensores recolhem dados do ambiente;
2. O Raspberry Pi interpreta os dados e executa ações definidas na lógica de programação;
3. Os atuadores respondem fisicamente às instruções;
4. A interface exhibe os dados ao utilizador e permite intervenção manual, se necessário.

Este fluxo está representado num fluxograma de programação (Figura 12), onde se observam as decisões lógicas implementadas no código Python.

### 3.3. Importância Didática

Além do valor técnico, o protótipo foi desenvolvido com um intuito pedagógico claro: proporcionar aos alunos uma plataforma acessível, modular e extensível para a aprendizagem dos seguintes conceitos:

- Fundamentos de eletrónica digital (sensores e atuadores)
- Programação de sistemas embebidos (Python, GPIO)
- Noções de IoT e redes de comunicação
- Integração de hardware com software
- Conceção e montagem de sistemas domóticos

Nos anexos desta dissertação apresentam-se três guias de laboratório possíveis para utilização do protótipo

Apesar dos bons exemplos existentes, ainda deixa espaço para:

- Mais cursos integrados em escolas secundárias ou ensino profissional — facilitando o ingresso jovem na área;
- Atualização constante face às novas tecnologias (IoT, IoE, IA, domótica baseada em software aberto, integração com energia renovável, smart-grid, “smart home/cloud”);
- Maior oferta de cursos modulares e flexíveis (online, híbrido, formação contínua) para técnicos que já trabalham;
- Integração pedagógica: não só no ensino técnico, mas utilizando a domótica como ferramenta educativa para outras disciplinas (sustentabilidade, conforto, design de edifícios, energia, informática, etc.).

### 3.4. Apps de controlo / domótica adaptáveis ao ensino

São aplicações concebidas para controlo de sistemas domóticos “reais” (ou de maquetes), e que podem servir de base para laboratórios, projetos ou demonstrações em contexto educativo. As mais utilizadas são:

- **Homey for Education** — uma iniciativa pensada especialmente para escolas/instituições, onde se pode usar a plataforma de domótica da Homey como ferramenta educativa. Permite aos alunos aprender programação, automações, integração de dispositivos IoT e lógica de controlo real.
- **OpenHAB** — software open-source de automação residencial, largamente utilizado como hub de domótica, compatível com diversos protocolos (Zigbee, Z-Wave, Wi-Fi, etc.), e que pode ser instalado em hardware como Raspberry Pi. Permite configurar regras, controlos, interfaces de visualização e controlo — excelente para uso educativo.

Tendo por base a literatura da área e nas iniciativas atuais, Existem três grandes linhas de desenvolvimento nesta área:

- Camada prática real / laboratório — com apps e sistemas (como Homey, OpenHAB, apps de controlo via smartphone) adaptáveis a projetos reais ou maquetes.
- Camada de iniciação / programação educativa — plataformas visuais ou kits de robótica/IoT (VIPLE, Makeblock, etc.) que permitem ensinar lógica, sensores/atuadores e conceitos de automação de forma lúdica ou introdutória.
- Camada teórico-prática / curso ou e-learning — cursos, tutoriais, plataformas online e ambientes de aprendizagem guiada, que combinam teoria, prática e uso de apps/interfaces para ensinar domótica, IoT e automação

A app desenhada pode combinar as três camadas pode oferecer uma abordagem rica e eficaz: desde a sensibilização teórica até a implementação prática e projeto final.

### 3.4. Apresentação Geral do Protótipo

Este capítulo descreve o protótipo que foi desenvolvido com o objetivo de simular uma moradia unifamiliar inteligente, controlada através de tecnologia IoT (Internet of Things), tendo como unidade central o microcomputador Raspberry Pi. O sistema permite monitorizar e controlar funções como iluminação, climatização, estores, alarmística e sensores diversos. A Figura 10 apresenta o diagrama de blocos desenvolvido com base na planta representada no Anexo 1.

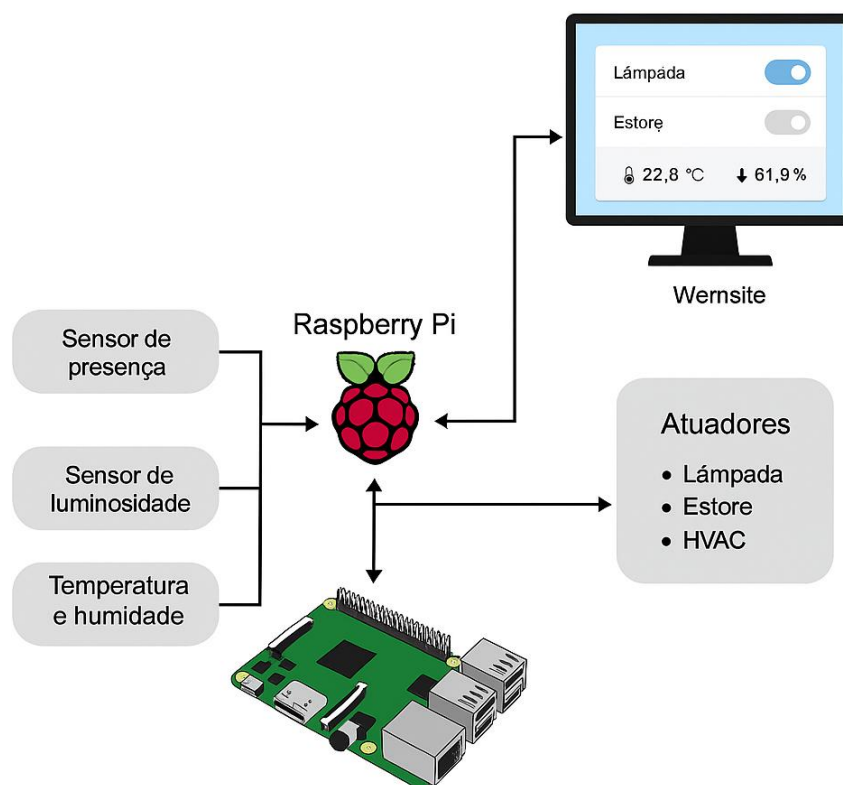


Figura 11– Diagrama geral do sistema domótico

O diagrama apresentado na Figura 11 representa o funcionamento central do sistema, com base na utilização de um Raspberry Pi como unidade de controlo. Os componentes representados são:

- Sensores, divididos nas seguintes áreas:
  - a) Sensor de presença (PIR): Deteta movimento de pessoas em ambientes internos.
  - b) Sensor de luminosidade (LDR): Mede a intensidade da luz ambiente, controlando estores e iluminação;
  - c) Sensor de temperatura e humidade (DHT22): Fornece dados ambientais para conforto e climatização.
- Raspberry: Atua como controlador central. Recebe os dados dos sensores e envia comandos aos atuadores.
- Atuadores, onde lâmpada liga/desliga conforme deteção ou manualmente; Estore: sobe/desce com base na luminosidade; HVAC: sistema de aquecimento, ventilação e ar condicionado
- Interface Web: Exibe os valores dos sensores e permite ao utilizador comandar os dispositivos remotamente, via browser.

O fluxo de funcionamento do sistema segue uma sequência lógica de prioridades. Numa primeira fase, os sensores efetuam a recolha dos dados provenientes do ambiente, fornecendo informações sobre as variáveis monitorizadas, em seguida, o Raspberry Pi processa os sinais recebidos e determina as ações a executar com base na lógica de controlo previamente definida. Posteriormente, os atuadores são ativados de acordo com estas decisões, assegurando a resposta física do sistema. Por fim, a interface web apresenta os dados em tempo real, permitindo ao utilizador monitorizar e interagir manualmente com o sistema sempre que necessário.

### 3.5. Arquitetura Funcional do Sistema

Conforme referido, o sistema foi desenvolvido com base num Raspberry Pi 3+, responsável por gerir sensores, atuadores e interface de utilizador. Na Figura 12 encontra-se o diagrama de blocos da arquitetura principal:

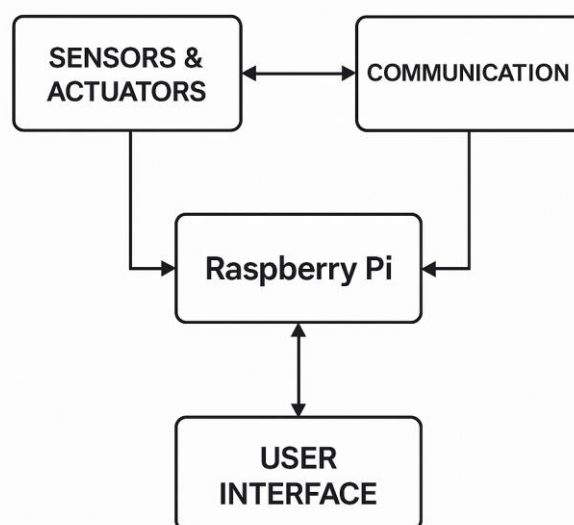


Figura 12– Diagrama de Blocos da Arquitetura Funcional do Sistema Domótico

O diagrama da fig. 12 apresenta a arquitetura funcional do sistema baseado em Raspberry Pi, com os seguintes blocos:

- Sensores e atuadores: Compreende todos os dispositivos físicos de entrada (sensores PIR, LDR, DHT22) e saída (relés, LEDs, motores de estores). São ligados diretamente aos pinos GPIO do Raspberry Pi. Enviam sinais analógicos ou digitais ao controlador e recebem comandos de ativação.
- Comunicação: Refere-se aos canais de comunicação utilizados para troca de dados entre os sensores/atuadores e o Raspberry Pi. Inclui o barramento GPIO, I2C ou protocolos de rede (Wi-Fi) se os sensores forem remotos. Também pode incluir MQTT, HTTP ou WebSocket em versões mais avançadas.
- RASPBERRY PI: É o cérebro do sistema, responsável pelo processamento dos dados dos sensores, execução da lógica de controlo, e envio de comandos para atuadores. Executa scripts em Python e hospeda o servidor web (via Flask). Atua como gateway local ou como nó intermediário em redes IoT maiores.
- Interface do utilizador: Interface gráfica acessível via navegador web. Permite ao utilizador visualizar valores de sensores, ativar manualmente os dispositivos, configurar parâmetros e monitorizar o sistema. É servida pelo Raspberry Pi através da rede local ou internet.

O fluxo de funcionamento do sistema inicia-se com a interação do utilizador através da interface, a partir da qual são enviados os comandos e solicitações. A interface comunica com o Raspberry Pi, que atua como unidade central de controlo, onde os dados recebidos são processados e executados segundo a lógica programada. O Raspberry Pi aciona os sensores e atuadores ligados às portas GPIO, promovendo a resposta física correspondente às instruções do utilizador ou às condições detetadas. Os sensores, por sua vez, atualizam continuamente os valores recolhidos do ambiente, garantindo o feedback necessário ao sistema. Este processo estabelece uma comunicação permanente e bidirecional entre todos os blocos funcionais, assegurando um controlo em tempo real, automatizado e eficiente.

### **3.6. Programação do Raspberry Pi**

O software foi desenvolvido em Python 3, utilizando bibliotecas como RPi.GPIO e GPIO Zero. As principais funcionalidades incluem leitura de sensores, controlo de atuadores, criação de cenários automáticos e notificações por e-mail. Também foi implementado um painel de controlo remoto com Flask. A Figura 13 apresenta o fluxo lógico da programação.

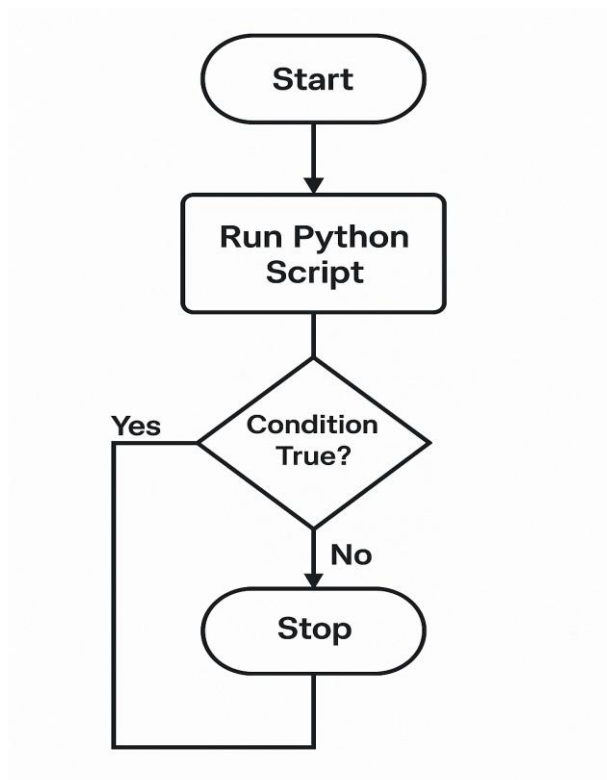


Figura 13 – Fluxograma da Lógica de Programação em Python

A imagem mostra um fluxograma simplificado do funcionamento do sistema domótico com base em scripts Python. Este fluxograma representa a lógica básica de execução de um script que controla sensores e atuadores conectados ao Raspberry Pi.

As etapas do fluxograma são as seguintes:

1. Start: Início do processo. Pode representar o boot do Raspberry Pi ou o acionamento manual do sistema;
2. Run Python Script: O script principal é executado, contém instruções para:
  - Ler sensores (temperatura, luz, movimento);
  - Processar os dados;
  - Acionar dispositivos conforme a lógica definida;
3. Condition True? O sistema avalia se uma condição predefinida é satisfeita, por ex. “Movimento detetado?” ou “Luminosidade < limite?”. Pode envolver múltiplos sensores ou variáveis lógicas;
4. Yes → Loop: Se a condição for verdadeira, o sistema continua em execução e atualiza a interface e/ou aciona atuadores;
5. No → Stop: Se a condição for falsa (por exemplo, fim de tempo de execução ou erro) o script encerra a execução e pode reiniciar ou aguardar nova instrução.

Para a aplicação no Projeto Domótico, este tipo de lógica de programação é utilizado em:

- main.py, que roda continuamente e lê sensores
- Scripts de teste com ciclos while True que monitorizam eventos
- Processos automáticos de segurança, como detecção de intrusos ou controlo de iluminação.

A programação foi feita em **Python**, com suporte das bibliotecas:

- RPi.GPIO – controlo de pinos
- Flask – servidor web local
- Adafruit\_DHT – leitura do sensor DHT22

### **3.5. Comunicação e Monitorização**

O sistema permite notificações por e-mail e monitorização remota em tempo real através de um painel web. Esta funcionalidade melhora a segurança e a supervisão do sistema domótico, mesmo à distância.

No que concerne às Camadas Funcionais são utilizadas as seguintes:

- Camada Física: sensores/atuadores conectados via GPIO
- Camada Lógica: scripts Python para decisão e controlo
- Camada Interface: painel de controlo em web browser

### **3.6. Considerações Técnicas e Didáticas**

Este protótipo demonstra conceitos fundamentais de domótica e IoT, sendo uma ferramenta eficaz para o ensino técnico-profissional. A modularidade e baixo custo tornam-no ideal para formação em áreas como automação e robótica, abordando tópicos como integração de sensores, onde os alunos podem montar, testar e modificar o sistema em laboratório, explorando os princípios de IoT, circuitos digitais, e aplicações web.

## CAPÍTULO 4 – ENSAIOS E TESTES NO PROTÓTIPO

Neste capítulo pretende-se demonstrar que o sistema proposto permite a implementação de um conjunto alargado de atividades laboratoriais, essenciais para que os formandos adquiram competências práticas no domínio da domótica e das tecnologias digitais aplicadas a edifícios inteligentes.

O protótipo foi concebido de forma modular, integrando sensores, atuadores, uma plataforma Raspberry Pi e uma interface web, possibilitando a realização de testes de leitura, controlo, automatização e integração em rede. A abordagem experimental adotada permitiu não só verificar a capacidade operacional do sistema, como também avaliar o seu potencial enquanto ferramenta pedagógica para cursos técnicos e tecnológicos.

### 4.1. Ensaios a efetuar no protótipo

Para que o protótipo esteja apto a ser utilizado na lecionação de conteúdos relacionados com IoT e domótica é necessário que se verifiquem as seguintes condições:

- Validar o funcionamento individual e integrado dos sensores e atuadores.
- Avaliar a estabilidade da comunicação entre os módulos do sistema.
- Testar a interface de controlo desenvolvida em Flask e a sua usabilidade.
- Verificar o desempenho do Raspberry Pi em situações de carga real.
- Demonstrar a aplicabilidade do protótipo em atividades práticas de formação.
- Identificar eventuais limitações e oportunidades de melhoria.

A realização destes ensaios é fundamental para comprovar que o protótipo pode ser utilizado de forma eficaz em contextos educativos, proporcionando aos formandos contacto direto com tecnologias de automação residencial baseadas em IoT.

### 4.2. Preparação do Ambiente de Testes

Para a execução dos testes é necessário montado um ambiente controlado composto por: **Raspberry Pi 3 B+** com sistema operativo Raspberry Pi OS; Sensores integrados: DHT22 (temperatura e humidade), LDR (luminosidade), PIR (movimento); Atuadores: LED e módulo relé para simulação de iluminação, motor para simulação do controlo de estores, Interface web construída em Flask, com comunicação via GPIO, Scripts em Python responsáveis pela lógica de interação entre sensores, atuadores e utilizador.

O ambiente de testes foi configurado de modo a replicar cenários reais de habitação inteligente, permitindo observar o comportamento do sistema perante estímulos externos controlados.

### **4.3. Validação Pedagógica do Protótipo**

Para além da validação funcional, foi realizada uma análise orientada ao seu uso como recurso didático. O protótipo tem potencial para integrar atividades práticas de formação, como leitura de sensores, programação, lógica de controlo e análise de dados. Pode ser aplicado em módulos de sistemas domóticos, eletrónica, IoT, automação e redes de comunicação. Tem capacidade de suportar metodologias de aprendizagem ativa, tais como:

- aprendizagem baseada em projetos (PBL),
- aprendizagem experimental,
- aprendizagem por problemas.

As funcionalidades desenvolvidas permitem que os formandos tenham contacto direto com aplicações reais da automação residencial, reforçando o carácter prático e tecnológico da formação.

O sistema demonstrou capacidade para operar como plataforma educativa completa, replicando cenários reais de domótica e possibilitando a aquisição de competências essenciais na área.

### **4.4. Considerações Finais**

O sistema é suficientemente robusto para a utilização em contextos de formação, permitindo aos formandos aprender de forma prática conceitos de IoT, automação residencial e integração de sistemas.

As observações efetuadas durante esta fase forneceram ainda elementos importantes para futuras melhorias, que serão apresentadas no capítulo seguinte. No sistema de iluminação a temporização programada em Python assegura o desligar automático após o intervalo configurado. Esta funcionalidade é eficaz na redução de consumos energéticos, potenciando uma gestão racional de iluminação tal como recomendada em sistemas inteligentes de controlo energético. Adicionalmente, verificou-se que a integração com a interface web permite ao utilizador ignorar temporariamente a lógica automática e efetuar o comando manual da iluminação, garantindo flexibilidade operacional — um aspeto valioso no contexto educativo, permitindo trabalhar diferentes modos de controlo.

No que concerne aos estores automáticos, sistema de estores possibilita a capacidade de adaptação às variações da luminosidade natural. Através do sensor LDR é possível monitorizar com precisão a intensidade. A lógica de controlo

implementada prevê não apenas a atuação imediata do motor, mas também mecanismos de proteção contra acionamentos repetitivos, garantindo maior durabilidade dos componentes. Do ponto de vista energético, o sistema pode contribuir para otimizar o conforto térmico e luminoso da habitação simulada, oferecendo uma aplicação prática de conceitos de eficiência energética – aspecto de grande valor no contexto de ensino técnico-profissional.

A monitorização ambiental realizada pelo sensor DHT22 apresenta leituras e coerentes com os cenários utilizados. A taxa de atualização das medições revelou-se adequada à finalidade do sistema, possibilitando uma supervisão contínua e precisa das condições térmicas e higrométricas. A função de registo de dados (logging) mostra-se essencial, permitindo a criação de históricos para análise posterior — uma funcionalidade de grande pertinência no ensino de sistemas inteligentes e análise de dados IoT.

O sistema integra uma interface web que permite o controlo remoto das cargas elétricas associadas, como iluminação, estores e dispositivos auxiliares. Através desta interface, o utilizador pode ativar ou desativar cada carga de forma individual, diretamente a partir de um computador pessoal (PC) ou de qualquer dispositivo com acesso à rede local. A comunicação entre o utilizador e o Raspberry Pi é estabelecida via servidor Flask, garantindo uma resposta rápida e fiável aos comandos executados. A visualização no PC apresenta uma disposição gráfica intuitiva, onde são exibidos em tempo real os estados das cargas e os valores provenientes dos sensores, proporcionando uma gestão centralizada e eficiente do sistema doméstico.

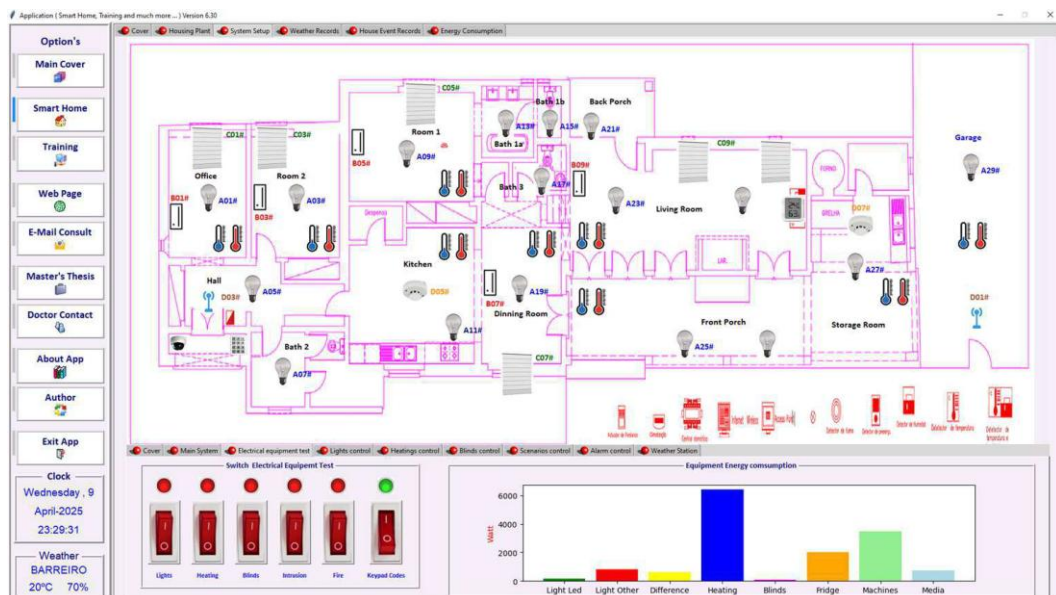


Figura 14 – Controlo de estores e luzes no sistema real

A Figura 14 ilustra a interface de controlo implementada no sistema real, onde é possível observar o painel principal desenvolvido em Flask. Esta interface permite ao

utilizador acionar remotamente os estores e o sistema de iluminação da moradia, de forma individual ou simultânea, através de botões interativos. Cada comando é transmitido via rede local ao Raspberry Pi, que executa a ação correspondente, atualizando de imediato o estado dos dispositivos. A interface apresenta ainda indicadores visuais que confirmam o estado atual de cada carga (ligado/desligado), garantindo um controlo intuitivo, fiável e acessível a partir de qualquer computador ou dispositivo móvel.

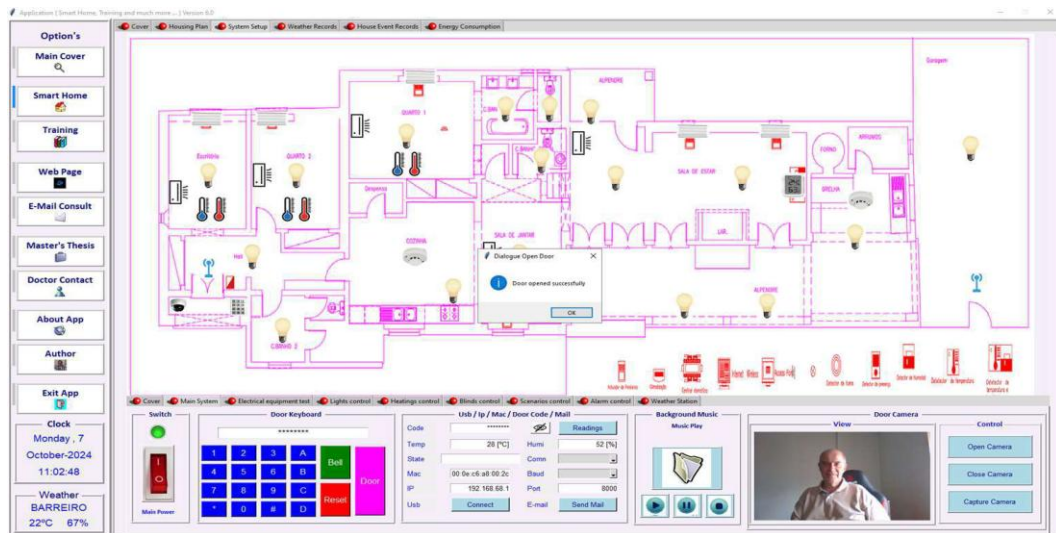


Figura 15 – Detecção de Intruso

Na Figura 15 é demonstrada a funcionalidade de segurança do sistema através da deteção de intrusão. O sensor PIR (Passive Infrared Sensor) capta o movimento no ambiente e envia o sinal ao Raspberry Pi, que ativa de forma automática o alarme sonoro (buzzer) e envia simultaneamente uma notificação por e-mail ao utilizador. Este teste permitiu validar o correto funcionamento da rotina de segurança, confirmando que a deteção ocorre em tempo real e que o sistema é capaz de reagir de forma autónoma, sem necessidade de intervenção manual.

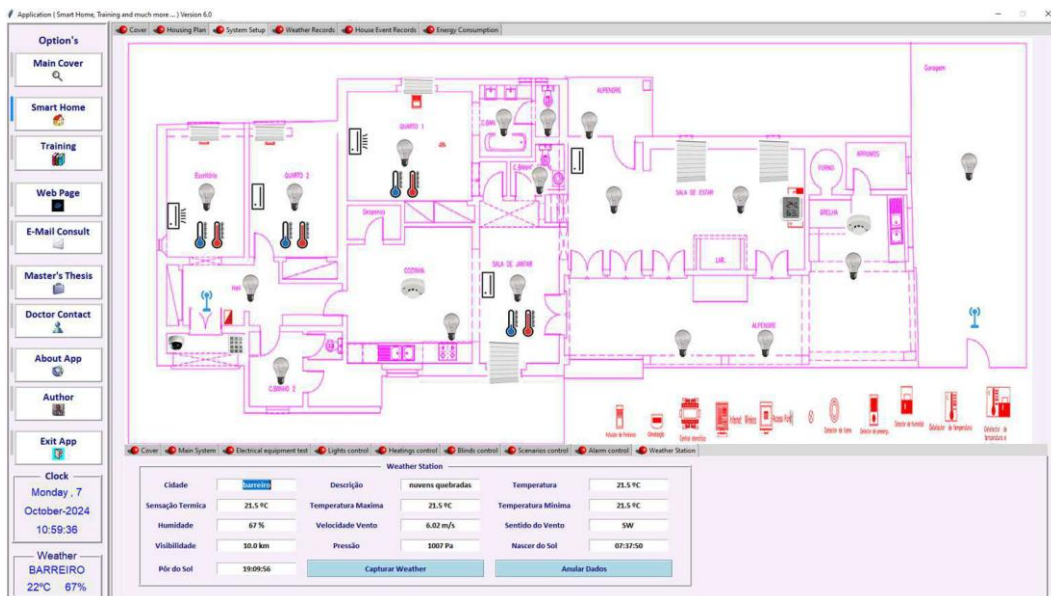


Figura 16 – Monitorização das condições meteorológicas

A Figura 16 apresenta o ecrã de monitorização das condições meteorológicas internas, obtidas através do sensor DHT22. A interface exibe, em tempo real, os valores de temperatura e humidade recolhidos pelo sensor, possibilitando ao utilizador acompanhar as variações ambientais do espaço. Estes dados são atualizados automaticamente e armazenados para análise posterior, permitindo a criação de históricos e o estudo do comportamento térmico do ambiente ao longo do tempo. Esta funcionalidade demonstra a integração entre hardware e software e a capacidade do sistema em fornecer informação contínua e precisa.

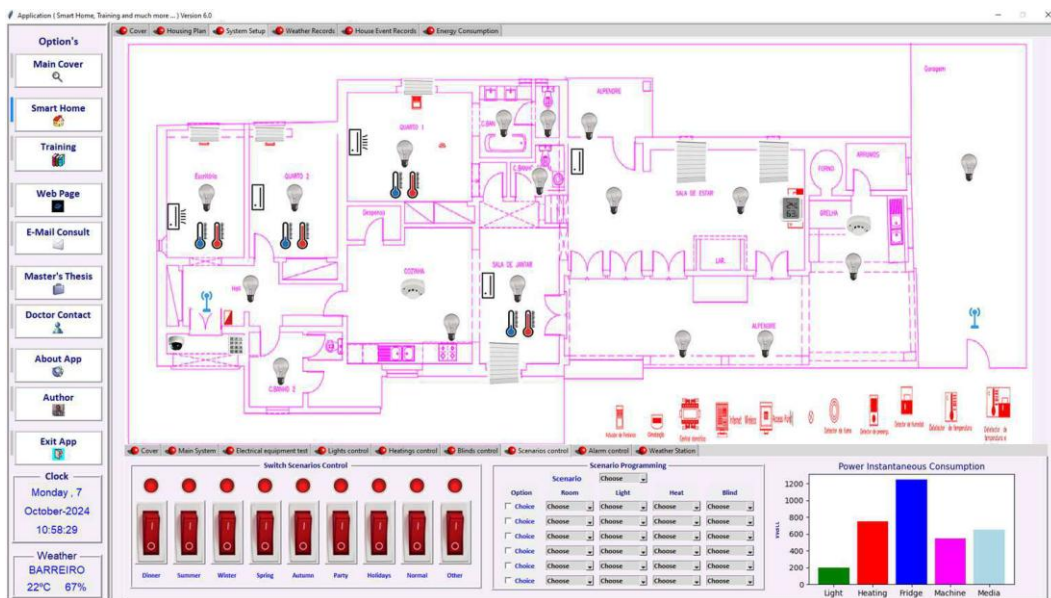


Figura 17– Gráficos de consumo energético diário

A Figura 17 representa os gráficos de consumo energético diário obtidos a partir dos dados registados pelo sistema. Estes gráficos evidenciam os períodos de maior utilização de energia elétrica e permitem identificar padrões de consumo ao longo do dia. A representação gráfica facilita a análise visual dos resultados, servindo como ferramenta de apoio à gestão energética e à otimização do funcionamento do sistema domótico

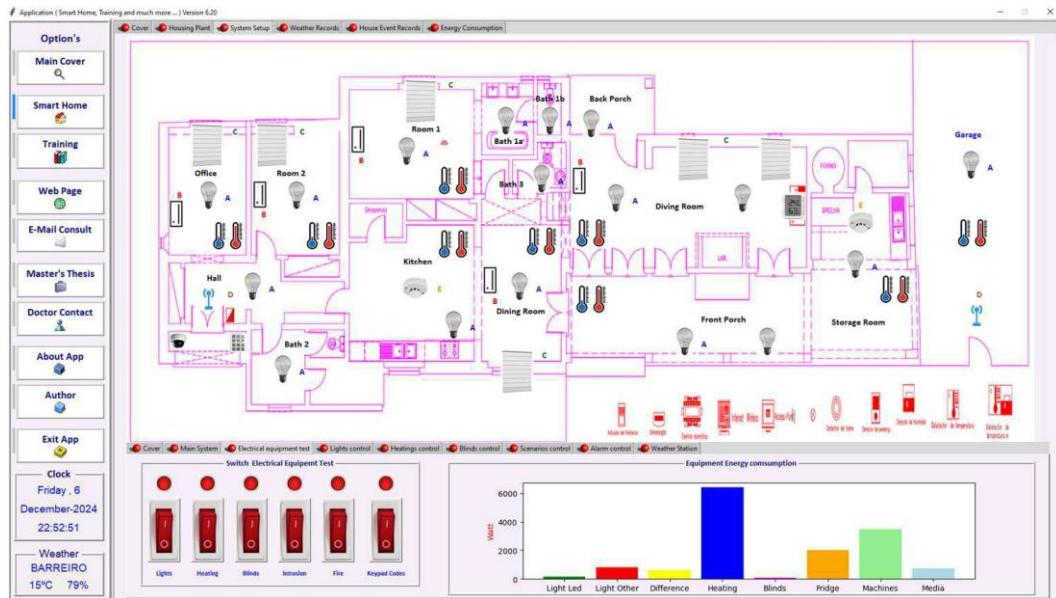


Figura 18– Gráficos de consumo energético mensal

Por fim, a Figura 18 apresenta a consolidação dos dados de consumo energético mensal. Este registo de longo prazo permite avaliar o desempenho energético global da instalação e verificar o impacto das rotinas de automatização implementadas. A visualização dos dados mensais contribui para a tomada de decisões mais informadas relativamente à eficiência e sustentabilidade do sistema, permitindo propor ajustes ou melhorias no controlo das cargas e na programação dos dispositivos automatizados.



## **CAPÍTULO 5 - CONCLUSÃO**

### **5.1 Conclusões Finais**

O desenvolvimento e utilização deste protótipo no contexto educativo revelam um conjunto significativo de oportunidades de aprendizagem transversal, fundamentais para a formação de técnicos e engenheiros na área da automação, eletrónica e sistemas inteligentes.

A primeira oportunidade reside na integração efetiva entre teoria e prática, permitindo aos estudantes compreender conceitos que frequentemente permanecem abstratos — tais como sensores, atuadores, protocolos de comunicação, lógica de controlo, redes IoT, eficiência energética e interfaces de utilizador. A interação direta com o hardware, aliada ao desenvolvimento em Python, reforça a aprendizagem baseada em projetos (Project-Based Learning), amplamente defendida pelas metodologias pedagógicas contemporâneas.

Além disso, o protótipo estimula a aprendizagem multidisciplinar, uma vez que envolve áreas como eletrónica digital, programação, redes de comunicação, sistemas embebidos, web design e análise de dados. Este carácter transversal é particularmente relevante na formação profissional e superior, pois aproxima os estudantes das competências exigidas pela Indústria 4.0.

Outro aspeto significativo é a promoção da autonomia e pensamento crítico, uma vez que os alunos são incentivados a identificar problemas, formular hipóteses, testar soluções, ajustar parâmetros e avaliar os resultados. Tal abordagem fomenta competências essenciais como troubleshooting, resolução de problemas reais, tomada de decisão e interpretação de métricas de desempenho.

O protótipo também permite abordar questões centrais da era digital, como eficiência energética, sustentabilidade, cibersegurança e interoperabilidade de sistemas, preparando os alunos para lidar com desafios que estão presentes no ambiente profissional atual.

Finalmente, destaca-se o potencial do sistema como ferramenta de experimentação contínua, dado que pode ser facilmente expandido com novos sensores, protocolos de comunicação ou módulos de integração. Assim, o protótipo constitui não apenas um recurso didático, mas uma plataforma evolutiva que acompanha a progressão dos conteúdos programáticos, permitindo a realização de projetos de curta ou longa duração, incluindo trabalhos finais, estágios, competições tecnológicas e atividades laboratoriais regulares.

Apesar de o protótipo desenvolvido ser adequado aos objetivos propostos, a sua arquitetura apresenta um conjunto de oportunidades de evolução que podem elevar significativamente a sua maturidade técnica e o seu potencial pedagógico. A implementação destas melhorias permitirá aproximar ainda mais o sistema de ambientes domésticos reais, tornando-o um recurso formativo mais completo e versátil.

Numa perspetiva de continuidade e reforço da investigação, apresentam-se de seguida as principais linhas de desenvolvimento identificadas.

Uma primeira área de evolução reside no reforço da infraestrutura de comunicação e de processamento do sistema. A adoção de protocolos específicos para IoT, como MQTT ou CoAP, permitiria atingir um desempenho superior na transmissão de dados, aumentando a eficiência e a escalabilidade da arquitetura. Estes protocolos, pela sua leveza e robustez, garantem um fluxo de informação mais estável, sobretudo quando o sistema incorpora um número elevado de sensores ou é distribuído por múltiplos ambientes.

Outra melhoria relevante passa pela integração de uma base de dados estruturada, seja local (SQLite) ou remota (InfluxDB, MariaDB ou serviços em nuvem). Esta evolução permitiria não apenas registar de forma sistemática os dados recolhidos pelos sensores, como também possibilitaria análises temporais, criação de históricos e o desenvolvimento de modelos de previsão, enriquecendo significativamente o valor educativo e investigativo do sistema.

No caso dos estores, a introdução de motores dotados de feedback de posição possibilitaria um controlo mais preciso e gradual dos movimentos, permitindo níveis intermédios de abertura e ajustando automaticamente os parâmetros de acordo com as variações ambientais. Este tipo de controlo é particularmente relevante em sistemas de eficiência energética e conforto térmico.

Finalmente, a integração com assistentes de voz (Alexa, Google Assistant ou HomeKit) representaria um avanço natural, tornando o protótipo mais alinhado com as práticas contemporâneas de automação residencial.

Embora a interface atual ofereça uma interação funcional e intuitiva, é possível elevar a sua ergonomia e capacidade de visualização através da adoção de frameworks modernas, como React ou Bootstrap. A criação de dashboards dinâmicos e responsivos tornaria a supervisão do sistema mais clara, permitindo ao utilizador aceder a gráficos, estados de operação e alertas em tempo real.

A médio prazo, a criação de uma aplicação móvel dedicada surgiria como evolução natural, proporcionando um canal adicional de interação com o sistema. A aplicação poderia incluir funcionalidades como notificações push, visualização instantânea de sensores e configuração de modos automáticos, ampliando a acessibilidade e a flexibilidade da solução.

A crescente digitalização das infraestruturas domésticas requer a implementação de mecanismos de segurança informática que garantam a integridade e a privacidade dos utilizadores. Assim, recomenda-se a introdução de encriptação nas comunicações (HTTPS/TLS), mecanismos de autenticação reforçada e segmentação de rede dedicada aos dispositivos IoT.

Do ponto de vista da confiabilidade operacional, a inclusão de sistemas de tolerância a falhas, tais como watchdogs e fontes de alimentação ininterruptas (UPS), permitiria ao protótipo operar de forma estável mesmo em contextos adversos, assegurando recuperação automática em caso de interrupções energéticas ou falhas de software.

Um dos aspetos mais promissores do protótipo é o seu potencial enquanto ferramenta de ensino. Neste sentido, a sua expansão deverá contemplar não apenas melhorias técnicas, mas também o desenvolvimento de materiais pedagógicos complementares, como kits de exercícios, guias de laboratório, propostas de projetos e desafios orientados para diferentes níveis de complexidade.

A possibilidade de simular sensores e atuadores em ambiente virtual, antes da sua implementação física, representa outra dimensão relevante para o ensino, permitindo reduzir custos, minimizar erros iniciais e facilitar a compreensão dos conceitos estruturantes.

Por fim, a integração de ferramentas de análise de dados e de inteligência artificial abre caminho a novos módulos de aprendizagem relacionados com previsão de consumo energético, otimização automática de parâmetros e deteção de padrões anómalos. Estas aplicações, alinhadas com os paradigmas atuais da Indústria 4.0, reforçam a relevância contemporânea do protótipo enquanto recurso formativo.

As melhorias propostas visam transformar o protótipo atual numa plataforma mais robusta, escalável e alinhada com as exigências tecnológicas modernas. A sua evolução permitirá não apenas reforçar o desempenho técnico, mas também ampliar o alcance pedagógico, potenciando o desenvolvimento de competências multidisciplinares essenciais à formação de profissionais na área da automação, eletrónica e IoT.

## Bibliografia

- [1] M. A. Mazidi, R. D. McKinlay, D. Causey, *Fundamentals of Microcontrollers and Applications in Embedded Systems with the PIC18 Microcontroller Family*, Pearson, 2007.
- [2] A. Dunkels, J. Alonso, T. Voigt, "Making TCP/IP Viable for Wireless Sensor Networks," *Proceedings of the First European Workshop on Wireless Sensor Networks*, 2004.
- [3] Raspberry Pi Foundation, "Raspberry Pi Documentation," <https://www.raspberrypi.org/documentation>, Acesso em: julho 2025.
- [4] H. Karl, A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, Wiley, 2005.
- [5] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [6] KNX Association, "KNX Technology," <https://www.knx.org>, Acesso em: julho 2025.
- [7] D. E. Comer, *Internetworking with TCP/IP: Principles, Protocols, and Architecture*, Prentice Hall, 2006.
- [8] Adafruit Industries, "DHT22 Temperature-Humidity Sensor," <https://learn.adafruit.com>, Acesso em: julho 2025.
- [9] E. W. Dijkstra, "Go To Statement Considered Harmful," *Communications of the ACM*, vol. 11, no. 3, pp. 147–148, 1968.
- [10] M. Collares, "Domótica e Edifícios Inteligentes: Conceitos, Tecnologias e Tendências," *Revista Técnico-Científica da Engenharia Elétrica*, vol. 10, n.º 2, 2022.
- [11] Mathe, S. E., Sarker, P., & Das, D. R. (2024). *A comprehensive review on applications of Raspberry Pi*. *Procedia Computer Science*, 232, 140–155. <https://doi.org/10.1016/j.procs.2024.02.019>
- [12] Islam, S., Hasan, M., Akter, M., & Uddin, M. M. (2021). Implementation and analysis of an IoT-based home automation framework. *Journal of Computer and Communications*, 9(10), 45–59. <https://doi.org/10.4236/jcc.2021.910004>
- [13] Viani, F., Robol, F., Polo, A., & Massa, A. (2013). Wireless architectures for heterogeneous sensing in smart home applications: Concepts and real implementation. *Proceedings of the IEEE*, 101(11), 2381–2396. <https://doi.org/10.1109/JPROC.2013.2260298>
- [14] Gole, R., Sangale, K., & Ramesh, R. (2019). Smart air conditioning control system: A literature review. *International Journal of Information and Computer Science*, 6(2), 51–59.
- [15] Al Kuwari, M., Abdulrahman, R., Ismael, Y., Al Sada, K., & Sallabi, F. (2018). Smart-home automation using IoT-based sensing and monitoring platform. *IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG)*, 1–6. <https://doi.org/10.1109/CPE.2018.8372548>

- [16] Majeed, A. H., Shahzad, F., & Javed, Y. (2020). An intelligent, secure, and smart home automation system. *Wireless Communications and Mobile Computing, 2020*, 4579291. <https://doi.org/10.1155/2020/4579291>
- [17] Khan, M., Raza, S., & Tariq, U. (2023). A secure and smart home automation system with speech control. *Sensors, 23*(13), 5784. <https://doi.org/10.3390/s23135784>
- [18] Petrovic, N., & Grgic, S. (2021). Smart home automation—Use cases of a secure and integrated model. *Systems, 9*(4), 77. <https://doi.org/10.3390/systems9040077>
- [19] Kim, D., & Park, Y. S. (2025). Smart home advancements for health care and beyond. *JMIR Medical Informatics, 13*(1), e62793. <https://doi.org/10.2196/62793>
- [20] Jain, V., & Mishra, S. (2023). Smart home system: A comprehensive review. *International Journal of Smart Technology & Engineering, 11*(2), 25–37.
- [21] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>
- [22] John, J. (2024). *Smart home automation system*. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.4731421>
- [23] Rahman, S., & Haque, A. (2020). IoT based home automation using Raspberry Pi. *International Journal of Engineering Research & Technology, 9*(6), 512–518.
- [24], A., & Saha, P. (2023). Smart homes of the future: A comprehensive review of IoT-based home automation. *Journal of Embedded Systems and Internet of Things, 7*(4), 201–218.

# Anexos

## Anexo A – Plano de Aula: Introdução à Domótica com Raspberry Pi

### Unidade Curricular

Domótica / Internet das Coisas / Sistemas Embebidos / Automação Residencial

### Tema da Aula

### Introdução à Domótica com Raspberry Pi

### Duração

3 horas (180 minutos)

#### 1- Enquadramento da Aula

A domótica assume um papel central nos edifícios inteligentes e na Indústria 4.0, integrando sensores, atuadores, sistemas embebidos e comunicação em rede. O **Raspberry Pi**, enquanto plataforma de baixo custo, aberta e versátil, constitui uma ferramenta ideal para o ensino, permitindo a articulação entre **teoria, prática e projeto**.

Esta aula introduz os fundamentos da domótica através da implementação de um sistema domótico simples, com ênfase pedagógico e experimental.

#### 2- Objetivos de Aprendizagem

No final da aula, o estudante deverá ser capaz de:

##### Conhecimento

- Definir domótica e enquadrá-la no contexto da IoT
- Identificar os principais componentes de um sistema domótico

##### Compreensão

- Explicar o papel do Raspberry Pi como unidade central de controlo
- Compreender a interação sensores–processamento–atuadores

##### Aplicação

- Ligar sensores e atuadores aos GPIO do Raspberry Pi
- Executar scripts básicos em Python para leitura e controlo

##### Análise

- Interpretar dados recolhidos por sensores
- Avaliar vantagens e limitações da solução implementada

#### 3- Metodologia Pedagógica

- Aprendizagem Ativa
- Aprendizagem Baseada em Projetos (PBL)
- Demonstração + Laboratório Experimental
- Trabalho em pequenos grupos (2–3 alunos)

## 4- Recursos Didáticos

### Hardware

- Raspberry Pi 3 ou 4
- Cartão microSD com Raspberry Pi OS
- Sensor DHT22 (temperatura e humidade)
- Sensor LDR (luminosidade)
- Sensor PIR (movimento)
- Módulo relé / LED
- Protoboard, resistências, jumpers

### Software

- Python 3
- Bibliotecas: RPi.GPIO, gpiozero, Adafruit\_DHT
- Editor Thonny / VS Code
- Navegador Web (interface Flask – demonstração)

## 5 - Estrutura da Aula (Cronograma)

### 5.1.Introdução Teórica (30 min)

- Conceito de domótica
- Domótica vs Automação Industrial
- IoT aplicada a edifícios inteligentes
- Arquitetura típica de um sistema doméstico
- Exemplos reais (habitação, energia, segurança)

*Discussão orientada:*

“Que funcionalidades domésticas existem na vossa habitação?”

### 5.2. Raspberry Pi como Controlador Domótico (30 min)

- Características do Raspberry Pi
- GPIO: entradas e saídas digitais
- Sistema operativo e ambiente de desenvolvimento
- Segurança elétrica e boas práticas

Demonstração:

- Boot do Raspberry Pi
- Acesso ao terminal
- Execução de um script simples em Python

### 5.3. Arquitetura do Sistema Domótico (30 min)

- Sensores → Processamento → Atuadores
- Fluxo de dados

- Lógica de controlo
- Introdução à automação baseada em eventos

Apresentação de:

- Diagrama de blocos
- Fluxograma de funcionamento

#### **5.4. Atividade Laboratorial (60 min)**

##### **Objetivo**

Criar um sistema domótico básico com Raspberry Pi

##### **Atividades**

1. Ligação de sensores (DHT22, LDR ou PIR)
2. Leitura de valores em Python
3. Controlo de um LED/relé com base numa condição
4. Execução e teste do sistema

Exemplo de lógica:

“Se for detetado movimento → ligar luz”

#### **5.5. Discussão e Reflexão (20 min)**

- Análise dos resultados
- Dificuldades encontradas
- Possibilidades de expansão:
  - Interface Web
  - Base de dados
  - MQTT
  - Integração cloud

#### **5. 6. Síntese e Encerramento (10 min)**

- Revisão dos conceitos-chave
- Enquadramento da aula seguinte
- Lançamento de desafio/projeto

#### **6. Avaliação (Formativa)**

- Participação ativa
- Funcionamento do sistema
- Interpretação dos resultados
- Capacidade de análise crítica

*Opcional:* relatório curto ou questionário reflexivo

## **7. Trabalho Pós-Aula (Proposta)**

Desenvolver um **mini-projeto** com:

- 2 sensores
- 1 atuador
- Lógica automática
- (opcional) Interface Web

## **8. Resultados Esperados**

- Consolidação de conceitos de domótica e IoT
- Desenvolvimento de competências práticas
- Preparação para projetos avançados
- Aproximação à realidade profissional

## **9. Observações Finais**

Este plano é totalmente adaptável a:

- Aulas de 2h ou 4h
- Ensino presencial ou laboratório híbrido

Para exemplo de trabalho prático apresentam-se três guiões laboratoriais possíveis no anexo D

# Anexo B – Planta Técnica (AutoCAD)

PRODUCED BY AN AUTODESK STUDENT VERSION

PRODUCED BY AN AUTODESK STUDENT VERSION



## Anexo C – Códigos-fonte em Python

main.py: script principal para leitura de sensores e controlo de relé

interface.py: backend para interface web

config.py: configuração dos pinos GPIO e thresholds

O código por ser muito extenso apresenta-se em repositório git e numa pen drive. Apresentam-se excertos para permitirem emular funcionamento em PC.

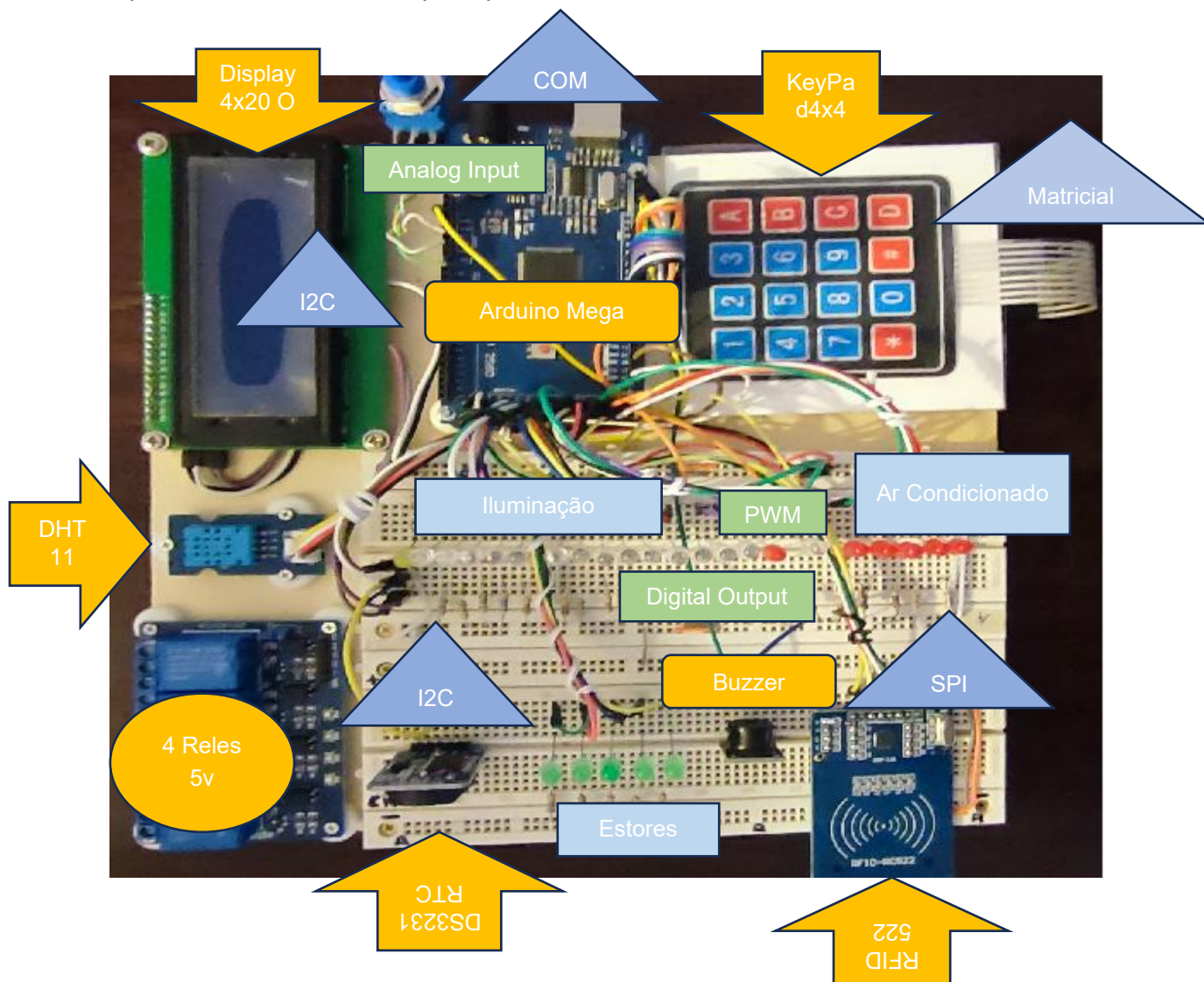


Figura 19 – Representação da Casa Inteligente

O funcionamento da montagem assenta em três possibilidades, a primeira possibilidade o acesso à entrada da habitação, para tal o controlador deteta o acesso à habitação através do módulo RFID-RC522 se este for acionado com a tag correta a porta da habitação vai abrir-se.

A segunda possibilidade faz-se pelo teclado Keypad 4x4, este vai permitir localmente aceder à iluminação das diversas divisões da moradia, ligando e

desligando as luzes das diferentes divisões, subir ou descer os estores e ligar ou desligar o equipamento de ar condiciona das divisões onde existam.

A terceira possibilidade possibilita comunicações serie entre a controlador da casa inteligente e o programa cria em Python e este possibilita a comunicação entre ele e o controlador da casa inteligente.

Com o modulo de relógio é possível a data e a hora dos eventos, quer eles sejam localmente quer sejam pela porta serie.

Códigos usados pelo keypad 4x4, para aceder aos diferentes equipamentos distribuídos pelas divisões da habitação (Iluminação, ar condicionado, Estores).

Índice	Código	Ação
001	000#	Desliga tudo
002	A**#	Liga a iluminação da habitação
003	A01#	Liga Iluminação divisão Office
004	A02#	Desliga Iluminação divisão Office
005	A03#	Liga Iluminação divisão Room 2
006	A04#	Desliga Iluminação divisão Room 2
007	A05#	Liga Iluminação divisão Hall
008	A06#	Desliga Iluminação divisão Halll
009	A07#	Liga Iluminação divisão Bath Room 2
010	A08#	Desliga Iluminação divisão Bath Room 2
011	A09#	Liga Iluminação divisão Room 1
012	A10#	Desliga Iluminação divisão Room 1
013	A11#	Liga Iluminação divisão Kitchen
014	A12#	Desliga Iluminação divisão Kitchen
015	A13#	Liga Iluminação divisão Bath Room 1a
016	A14#	Desliga Iluminação divisão Bath Room 1a
017	A15#	Liga Iluminação divisão Bath Room 1b
018	A16#	Desliga Iluminação divisão Bath Room 1b
019	A17#	Liga Iluminação divisão Bath Room 3
020	A18#	Desliga Iluminação divisão Bath Room 3
021	A19#	Liga Iluminação divisão Dinning Room
022	A20#	Liga Iluminação divisão Dinning Room
023	A21#	Liga Iluminação divisão Porch_Back
024	A22#	Desliga Iluminação divisão Porch_Back

025	A23#	Liga Iluminação divisão Living Room
026	A24#	Desliga Iluminação divisão Living Room
027	A25#	Liga Iluminação divisão Porch_Front
028	A26#	Desliga Iluminação divisão Porch_Front
029	A27#	Liga Iluminação divisão Storage
030	A28#	Desliga Iluminação divisão Storage
031	A29#	Liga Iluminação divisão Garage
032	A30#	Desliga Iluminação divisão Garage
033	B**#	Liga o ar condicionada da habitação
034	B01#	Liga o ar condicionada da divisão Office Room
035	B02#	Desliga Ar condicionado da divisão Office Room
036	B03#	Liga Ar condicionado divisão Room 2
037	B04#	Desliga Ar condicionado divisão Room 2
038	B05#	Liga Ar condicionado divisão Room 1
039	B06#	Desliga Ar condicionado divisão Room 1
040	B07#	Liga Ar condicionado divisão Dinning Room
041	B08#	Dsliga Ar condicionado divisão Dinning Room
042	B09#	Liga Ar condicionado divisão Living Room
043	B10#	Desliga Ar condicionado divisão Living Room
044	C**#	Sobe os estores da habitação
045	C01#	Sobe os estores da divisão Office Room
046	C02#	Desce os estores da divisão Office Room
047	C03#	Sobe os estores divisão Room 2
048	C04#	Desce os estores da divisão Room 2
049	C05#	Sobe os estores da divisão Room 1
050	C06#	Desce os estores da divisão Room 1
051	C07#	Sobe os estores da divisão Dinning Room
052	C08#	Desce os estores da divisão Dinning Room
053	C09#	Sobe os estores da divisão Living Room
054	C10#	Desce os estores da divisão Living Room
055	D**#	

056	D01#	
057	D02#	
058	D03#	
059	D04#	
060	D05#	
061	D06#	

#### Utilização do DHT11

Inicio	SetUp	Loop
<pre>#include &lt;dht.h&gt; dht DHT;</pre>	<pre>int chk = DHT.read11(11);</pre>	<pre>lcd.print(String(int(DHT.temperature)) + "C"); lcd.setCursor(16,1); lcd.print(String(int(DHT.humidity)) + "%");</pre>

#### Utilização do Potenciómetro

Inicio	SetUp	Loop
<pre>// Potenciometro #define potPin A0 //Potenciometro int potValor = 0; float tensao = 0.0; int outputValue; bool ligado = false;</pre>	<pre>//Potenciometro pinMode(potPin, INPUT);</pre>	<pre>//leitura do potenciometro potValor = analogRead(A0); tensao = (potValor * 5.0) / 1024;  outputValue = map(potValor,0,1023,0,255); //actua na luz do Office if (ligado)  analogWrite(led_Office_light, outputValue);</pre>

#### Utilização do RTC\_DS3231

Inicio	SetUp	Loop
<pre>#include "RTClib.h" String TIME1="", TIME2="", DATA1="", DATA2="";</pre>	<pre>rtc.begin();  //rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); /* To manually set date and time, remove the coment // signs and enter new values in the followingline</pre>	<pre>//DS3231 RTC now = rtc.now(); ds = now.dayOfTheWeek();  ANO = String(now.year(), DEC);</pre>

<pre>String DIA="", MES="", ANO="", ANO2="", HORAS="", MINUTOS="", SEGUNDOS="";  int mes; RTC_DS3231 rtc;  String dia_Semana[7] = { "Dom", "Seg", "Ter", "Qua", "Qui", "Sex", "Sab"};  String mes_Ano[12] = { "Jan", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul", "Ago", "Set", "Out", "Nov", "Dec"}; int ds; DateTime now;</pre>	<pre>in this sequence: year, day, month, hour, minute and second.*/ rtc.adjust(DateTime(2025, 4, 3, 19, 50, 0)); if (! rtc.begin()) { Serial.println("Couldn't find RTC"); Serial.flush(); abort(); }  if (rtc.lostPower()) { Serial.println("RTC lost power, let's set the time!"); // When time needs to be set on a new device, or after a power loss, the // following line sets the RTC to the date &amp; time this sketch was compiled  rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // This line sets the RTC with an explicit date &amp; time, for example to set // January 21, 2014 at 3am you would call: // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0)); }</pre>	<pre>ANO2 = String(now.year()- 2000,DEC); mes = now.month()-1; MES = String(now.month(), DEC); DIA = String(now.day(), DEC); HORAS = String(now.hour(), DEC); MINUTOS = String(now.minute(), DEC); SEGUNDOS = String(now.second(), DEC);</pre>
--	--	--

### Utilização do RFID-RC522

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 53
#define RST_PIN 49
MFRC522 mfrc522(SS_PIN, RST_PIN);
byte acessUID[4] = {0xB3, 0xF1, 0xAB, 0x0D };
String rfidCard;
```

```

SPI.begin();
mfr522.PCD_Init();
if (!mfr522.PICC_IsNewCardPresent()) {
// atento ao keypad ou a comm com pc
}
if (mfr522.PICC_ReadCardSerial()) {
  if ( mfr522.uid.uidByte[0] == acessUID[0] &&
      mfr522.uid.uidByte[1] == acessUID[1] &&
      mfr522.uid.uidByte[2] == acessUID[2] &&
      mfr522.uid.uidByte[3] == acessUID[3]) {
    Serial.println("Acessed Garanted Door Open");
    digitalWrite(led_HomeDoor, HIGH);
    DisplayEvent("Hall", "Door", "Open");
    digitalWrite(led_HomeDoor, LOW);  }
  else {
    Serial.println("Acessed Denied");
    DisplayEvent("Hall", "Door", "Error");
    Serial.println();
    mfr522.PICC_HaltA();
  }
} //end mfr522

```

#### Utilização do Display 4x20

Início	Setup	Loop
<pre> #include &lt;Wire.h&gt; #include &lt;LiquidCrystal_I2C.h&gt; </pre>	<pre> LiquidCrystal_I2C lcd(0x27,20,4); </pre>	<pre> lcd.backlight(); lcd.setCursor(0,0); lcd.print(" Teste do Display "); </pre>

#### Utilização do Keypad 4x4

```

#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'} };
//KeyPad
byte rowPins[ROWS] = {2,3,4,5}; //connect to the row pinouts of the keypad

```

```

byte colPins[COLS] = {6,7,8,9}; //connect to the column pinouts of the keypad
Keypad keypad1 = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
key = keypad1.getKey();
  if (key) {
    if (key != '#') {
      Serial.print(key);
      wordKeyPad += key;
      delay(0.1);    }
    else {
      wordKeyPad += '#'; }
  }
//Desligar todos os equipamentos
if (wordKeyPad.equals("000#")) {
  Serial.println("\nKeypad Action : All Equipment Off");
  //Desliga as luzes
  digitalWrite(led_HomeDoor, LOW);
  digitalWrite(led_Office_light, LOW);
  digitalWrite(led_Room2_light, LOW);
  digitalWrite(led_Hall_light, LOW);
  digitalWrite(led_Bathroom2_light,LOW);
  digitalWrite(led_Room1_light, LOW);
  digitalWrite(led_Kitchen_light, LOW);
  digitalWrite(led_Bathroom1a_light,LOW);
  digitalWrite(led_Bathroom1b_light,LOW);
  digitalWrite(led_Bathroom3_light, LOW);
  digitalWrite(led_Dining_light, LOW);
  digitalWrite(led_Porch1_light, LOW);
  digitalWrite(led_Living_light, LOW);
  digitalWrite(led_Porch2_light, LOW);
  digitalWrite(led_Storage_light,LOW);
  digitalWrite(led_Garage_light, LOW);
  delay(0.1);
  //Desliga o aquecimento ar condicionado
  digitalWrite(led_Office_heating, LOW);
  digitalWrite(led_Room2_heating, LOW);
  digitalWrite(led_Room1_heating, LOW);
  digitalWrite(led_Dining_heating, LOW);
  digitalWrite(led_Living_heating, LOW);
  delay(0.1);
  //Desliga os estores
  digitalWrite(led_Office_blind, LOW);

```

```
digitalWrite(led_Room2_blind, LOW);  
digitalWrite(led_Room1_blind, LOW);  
digitalWrite(led_Dining_blind, LOW);  
digitalWrite(led_Living_blind, LOW);  
delay(0.1);  
wordKeyPad="";  
DisplayEvent("All", "Equip", "Off");  
delay(0.1);  
} //end "000#"
```

## ANEXO D- Guiões laboratoriais

### *Guião 1 - Introdução à Domótica com Raspberry Pi*

#### 1. Enquadramento Pedagógico

Este guião laboratorial visa proporcionar uma **aprendizagem prática e experimental** dos conceitos fundamentais da domótica, recorrendo à plataforma **Raspberry Pi** como controlador central. Os estudantes irão implementar um **sistema domótico funcional**, integrando sensores, atuadores e lógica de controlo em Python, aproximando-se de cenários reais de automação residencial.

#### 2. Objetivos da Atividade

No final da atividade, o estudante deverá ser capaz de:

- Identificar os componentes de um sistema domótico
- Efetuar ligações corretas aos pinos GPIO do Raspberry Pi
- Ler sensores ambientais em Python
- Controlar um atuador com base numa condição lógica
- Implementar uma lógica simples de automação
- Analisar o comportamento do sistema

#### 3. Material Necessário

##### 3.1 Hardware

- Raspberry Pi 3 ou 4
- Cartão microSD com Raspberry Pi OS
- Protoboard
- Cabos jumpers
- Sensor DHT22 (temperatura/humidade)
- Sensor PIR (movimento) **ou** LDR (luminosidade)
- LED + resistência 220  $\Omega$  **ou** módulo relé
- Fonte de alimentação 5 V

##### 3.2 Software

- Raspberry Pi OS
- Python 3
- Thonny IDE
- Bibliotecas:
  - RPi.GPIO
  - gpiozero
  - Adafruit\_DHT

## 4. Segurança e Boas Práticas

⚠ Antes de iniciar:

- Desligar o Raspberry Pi durante as ligações
- Confirmar tensões (GPIO = 3,3 V)
- Nunca ligar atuadores diretamente sem proteção
- Verificar polaridades dos sensores

## 5. Descrição do Sistema Domótico

### 5.1 Arquitetura Funcional

**Sensores → Raspberry Pi → Atuador → Utilizador**

- Sensor recolhe dados do ambiente
- Raspberry Pi processa a informação
- Atuador responde automaticamente

## 6. Procedimento Experimental

### Parte A – Preparação do Sistema (20 min)

1. Ligar o Raspberry Pi
2. Aceder ao sistema operativo
3. Abrir o **Thonny IDE**
4. Verificar versão do Python:  
`python3 --version`

### Parte B – Ligação do Sensor DHT22 (30 min)

**Ligações típicas:**

- VCC → 3.3 V
- DATA → GPIO 4
- GND → GND

**Código de leitura:**

```
import Adafruit_DHT
import time
```

```
sensor = Adafruit_DHT.DHT22
pin = 4
```

```
while True:
```

```
    hum, temp = Adafruit_DHT.read_retry(sensor, pin)
    if hum is not None and temp is not None:
        print(f"Temp: {temp:.1f} °C Humidade: {hum:.1f}%")
    else:
```

```
print("Erro na leitura")
time.sleep(2)
```

### **Resultado**

Visualização contínua da temperatura e humidade.

**esperado:**

### **Parte C – Ligação do Sensor PIR (20 min)**

#### **Ligações:**

- VCC → 5 V
- OUT → GPIO 17
- GND → GND
- 

#### **Código:**

```
import RPi.GPIO as GPIO
import time
```

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN)
```

```
while True:
    if GPIO.input(17):
        print("Movimento detetado")
    else:
        print("Sem movimento")
    time.sleep(1)
```

### **Parte D – Controlo de Atuador (LED/Relé) (20 min)**

#### **Ligações LED:**

- GPIO 27 → resistência → LED → GND

#### **Código:**

```
GPIO.setup(27, GPIO.OUT)

GPIO.output(27, GPIO.HIGH)
time.sleep(2)
GPIO.output(27, GPIO.LOW)
```

### **◆ Parte E – Automação (40 min)**

#### **Objetivo:**

Ligar a luz **se houver movimento**

```
import RPi.GPIO as GPIO
```

```
import time

GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN)
GPIO.setup(27, GPIO.OUT)

while True:
    if GPIO.input(17):
        GPIO.output(27, GPIO.HIGH)
        print("Luz ligada")
    else:
        GPIO.output(27, GPIO.LOW)
        print("Luz desligada")
    time.sleep(1)
```

## 8. Testes e Validação

Os estudantes devem verificar:

- Funcionamento correto dos sensores
- Resposta imediata do atuador
- Estabilidade do sistema
- Coerência das leituras

## 9. Questões de Reflexão

1. Que vantagens apresenta o Raspberry Pi face a um PLC?
2. Que limitações tem este sistema?
3. Como poderia ser integrado um controlo remoto?
4. Que riscos de segurança existem num sistema IoT?

## 10. Extensões Propostas (Opcional)

- Interface Web (Flask)
- Base de dados de leituras
- Comunicação MQTT
- Integração cloud
- Controlo por smartphone

## 11. Avaliação da Atividade

- Execução correta das ligações
- Funcionamento do sistema
- Qualidade da lógica implementada
- Capacidade de análise e explicação

## 12. Conclusão

Este guião permitiu a implementação de um **sistema doméstico real**, consolidando competências técnicas em sensores, programação e automação, fundamentais no ensino superior técnico.

## GUIÃO LABORATORIAL 2

### Controlo Automático de Iluminação com Sensor de Luminosidade (LDR)

#### 1. Identificação

- **Tipo:** Aula prática / laboratório
- **Duração:** 3 horas
- **Trabalho:** Grupos de 2–3 estudantes

#### 2. Enquadramento

Neste guião os estudantes implementam um **sistema de iluminação automática**, baseado na medição da luminosidade ambiente, simulando um cenário real de **eficiência energética** em edifícios inteligentes.

#### 3. Objetivos de Aprendizagem

No final da atividade o estudante deverá:

- Compreender o funcionamento de sensores analógicos
- Ler valores de luminosidade com Raspberry Pi
- Definir limiares de atuação
- Implementar lógica automática ON/OFF
- Relacionar domótica com eficiência energética

#### 4. Material Necessário

##### Hardware

- Raspberry Pi 3/4
- Sensor LDR
- Resistência 10 k $\Omega$  (divisor de tensão)
- LED + resistência 220  $\Omega$  **ou** módulo relé
- Protoboard e jumpers

##### Software

- Python 3
- Bibliotecas: gpiozero, RPi.GPIO

#### 5. Arquitetura do Sistema

Luminosidade → Raspberry Pi → Atuador (Iluminação)

## 6. Procedimento Experimental

### Parte A – Ligação do Sensor LDR (30 min)

Criar um **divisor de tensão**:

- LDR + resistência 10 k $\Omega$
- Ligação ao GPIO via entrada digital simulada

Alternativa pedagógica: usar gpiozero.LightSensor  
from gpiozero import LightSensor, LED  
from time import sleep

```
ldr = LightSensor(4)
```

```
led = LED(17)
```

```
while True:
```

```
    if ldr.dark:
```

```
        led.on()
```

```
        print("Ambiente escuro → luz ligada")
```

```
    else:
```

```
        led.off()
```

```
        print("Ambiente iluminado → luz desligada")
```

```
    sleep(1)
```

### Parte B – Ajuste de Limiar (30 min)

```
ldr = LightSensor(4, threshold=0.3)
```

Discussão:

- Como escolher o limiar?
- Influência da luz natural

### Parte C – Testes e Registo (30 min)

Os estudantes devem:

- Simular dia/noite
- Registrar comportamento do sistema
- Avaliar estabilidade

## 7. Questões de Reflexão

1. Que impacto tem este sistema no consumo energético?
2. Em que situações o controlo automático pode falhar?
3. Como introduzir controlo manual?

## 8. Conclusão

Este guião demonstra como a domótica contribui para **eficiência energética**, reforçando a importância da automação baseada em sensores ambientais.

## **GUIÃO LABORATORIAL 3**

### **Sistema Domótico com Controlo Remoto via Interface Web (Flask)**

#### **1. Identificação**

- **Tipo:** Laboratório avançado
- **Duração:** 3 horas
- **Pré-requisitos:** Guião 1 ou 2

#### **2. Enquadramento**

A interface homem–máquina é um elemento central na domótica moderna. Neste guião os estudantes desenvolvem um **painel web simples** para controlo remoto de dispositivos domóticos.

#### **3. Objetivos de Aprendizagem**

O estudante deverá:

- Compreender o conceito de controlo remoto
- Criar uma aplicação web simples em Flask
- Integrar hardware com software web
- Avaliar vantagens e riscos de controlo remoto

#### **4. Material Necessário**

##### **Hardware**

- Raspberry Pi
- LED ou relé
- Protoboard e jumpers

##### **Software**

- Python 3
- Flask
- Browser Web

Instalação:

```
pip3 install flask
```

#### **5. Arquitetura do Sistema**

**Browser → Servidor Flask → GPIO → Atuador**

#### **6. Procedimento Experimental**

**Parte A – Estrutura da Aplicação (30 min)**

```
Criar ficheiro app.py:  
from flask import Flask, render_template  
import RPi.GPIO as GPIO
```

```
GPIO.setmode(GPIO.BCM)  
GPIO.setup(17, GPIO.OUT)
```

```
app = Flask(__name__)
```

```
@app.route("/")  
def index():  
    return """  
    <h1>Controlo Domótico</h1>  
    <a href="/on">Ligar</a><br>  
    <a href="/off">Desligar</a>  
    """
```

```
@app.route("/on")  
def on():  
    GPIO.output(17, GPIO.HIGH)  
    return "Luz ligada"
```

```
@app.route("/off")  
def off():  
    GPIO.output(17, GPIO.LOW)  
    return "Luz desligada"
```

```
app.run(host="0.0.0.0")
```

### **Parte B – Teste do Sistema (30 min)**

- Aceder via browser:  
http://IP\_DO\_RASPBERRY:5000
- Ligar/desligar o atuador remotamente

### **Parte C – Discussão Avançada (30 min)**

- Diferença entre controlo local e remoto
- Segurança em sistemas IoT
- Possíveis melhorias:
  - Autenticação
  - HTTPS

- Interface gráfica

### **7. Questões de Reflexão**

1. Que riscos existem num sistema domótico acessível pela rede?
2. Que mecanismos de segurança deveriam ser implementados?
3. Como integrar sensores nesta interface?

### **8. Conclusão**

Este guião introduz os estudantes à **domótica conectada**, aproximando-os de soluções reais usadas em edifícios inteligentes e smart homes.