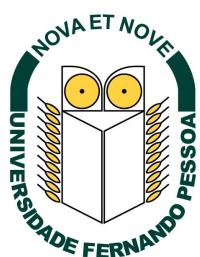


Reconhecimento de atividades em casas inteligentes

Universidade Fernando Pessoa



Luís Mota

Faculdade de Ciência e Tecnologia

Universidade Fernando Pessoa

Proposta final de dissertação

Julho

2023

Resumo

Após mais de duas décadas de existência, a internet das coisas, tem vindo a revolucionar a forma como interagimos com o mundo que nos rodeia. Apesar de, nas suas origens, a adoção de um paradigma de computação em nuvem ter servido de suporte a este modelo de computação ubíqua, a crescente complexidade dos sistemas IoT tem conduzido ao paulatino esvanecer do tradicional modelo hierárquico da computação em nuvem. A procura por soluções para os problemas de latência, escalabilidade e garantia de qualidade de serviço tem, nos últimos anos, impulsionado a deslocação do processamento e armazenamento de dados, da nuvem, para a periferia da rede (computação periférica). Partindo do caso particular de computação periférica que mantém o foco no alargar das fronteiras da inteligência artificial para a periferia da rede - Periferia inteligente - um levantamento do atual estado da arte é levado a cabo, culminando na especificação de uma arquitetura de suporte a cenários de periferia inteligente. Com vista à validação da arquitetura proposta, dois cenários são apresentados.

No âmbito da gestão de resíduos e reciclagem energética, um sistema para classificação de óleo alimentar usado, numa rede nacional de recolha doméstica é apresentado. Com classificação local da veracidade de cada depósito foi possível encurtar significativamente os tempos de resposta, com impacto direto nos níveis de consumo energético.

Direcionado às cidades inteligentes, um segundo cenário de aplicação, propõe uma abordagem baseada em visão computacional e aprendizagem profunda, para deteção local de peões em passadeiras. Neste contexto, um paradigma de periferia inteligente permitiu ultrapassar questões relativas à privacidade na transmissão de dados, assim como reduzir em mais de 80 vezes os tempos de resposta, quando comparado com uma solução de computação em nuvem.

Resumo

After more than two decades of existence, the internet of things has been revolutionizing the way we interact with the world around us. Although, in its origins, the adoption of a cloud computing paradigm supported this ubiquitous computing model, the increasing complexity of IoT systems has led to the gradual fading of the traditional hierarchical model of cloud computing. The search for solutions to the problems of latency, scalability and privacy has, in recent years, driven the movement of data processing and storage, from the cloud, to the edge of the network (edge computing). Starting from the particular case of edge computing that keeps the focus on extending the boundaries of artificial intelligence to the edge of the network - Edge intelligence - a survey of the current state of the art is carried out, culminating into the specification of an architecture to support edge intelligence applications. In order to validate the proposed architecture, two scenarios are presented.

In the scope of waste management and energy recycling, a system for used cooking oil classification in a national domestic collection network is presented. With the local classification of the trustworthiness of each deposit, it was possible to significantly shorten the response times, with a direct impact on energy consumption levels.

Aimed at smart cities, a second application scenario, proposes an approach based on computer vision and deep learning, for local detection of pedestrians on crosswalks. In this context, an edge intelligence paradigm allowed to overcome privacy related issues, as well as reducing response times by more than 80 times, when compared to a cloud computing based solution.

Agradecimentos

Gostaria de expressar a minha sincera gratidão aos meus orientadores de dissertação, o Prof. José Torres, Ph.D., e o Prof. Pedro Sobral, Ph.D. A sua orientação e apoio foram fundamentais para o sucesso deste trabalho.

Ao Prof. José Torres, agradeço pela sua dedicação incansável em fornecer orientações valiosas ao longo do processo de pesquisa e redação desta dissertação. A sua vasta experiência e profundo conhecimento na área foram essenciais para a minha compreensão dos tópicos abordados.

Ao Prof. Pedro Sobral, agradeço pela sua orientação perspicaz e pelo seu constante incentivo ao pensamento crítico. As suas sugestões e comentários construtivos desempenharam um papel crucial no desenvolvimento da minha investigação.

Ambos os orientadores demonstraram um compromisso exemplar com a excelência académica, partilhando generosamente o seu tempo e conhecimento. Agradeço-lhes por estarem sempre disponíveis para responder às minhas dúvidas e por me incentivarem a ultrapassar os obstáculos encontrados ao longo do caminho.

Além disso, gostaria de destacar a sua notável capacidade de inspirar e motivar, encorajando-me a alcançar resultados além das minhas expectativas iniciais. O seu apoio e encorajamento constantes foram fundamentais para manter-me focado e determinado durante todo o processo de pesquisa.

Por fim, gostaria de expressar o meu profundo agradecimento pela confiança depositada em mim ao aceitarem orientar esta dissertação. Sinto-me honrado por ter tido a oportunidade de trabalhar sob a vossa supervisão e sou grato por todas as lições valiosas que aprendi ao longo desta jornada académica.

Os vossos conhecimentos, orientação e apoio têm um impacto significativo na minha formação como investigador e profissional. Estou verdadeiramente grato por esta experiência enriquecedora e por terem contribuído para o meu crescimento académico.

Mais uma vez, o meu sincero agradecimento aos meus orientadores de dissertação, o Prof. José Torres, Ph.D., e o Prof. Pedro Sobral, Ph.D., por todo o seu apoio, orientação e inspiração ao longo deste trabalho.

Tabela de Conteúdos

Tabela de Conteúdos	v
Lista de Figuras	vii
Lista de Tabelas	ix
Definições e Acrónimos	x
1 Introdução	1
1.1 Enquadramento e motivação	1
1.2 Objetivos	3
1.3 Metodologia	3
1.4 Contribuições	4
1.5 Estrutura do trabalho	5
2 Trabalhos relacionados em ambientes inteligentes	7
2.1 Arquitetura de referência	8
2.2 Simulação em ambiente virtual	9
2.3 Sistemas Cliente-Servidor vs Sistemas <i>Peer-to-Peer</i>	12
2.4 Sistemas de monitorização	16
2.4.1 Análise comparativa	27
3 Arquitetura do Sistema <i>DistSense</i>	29
3.1 Introdução	29
3.2 Requisitos	30
3.3 Lógica de funcionamento do <i>DistSense</i>	33
3.4 Módulo de descoberta na rede	36
3.5 Módulo de comunicação	37
3.6 Módulo de aprendizagem computacional	39
3.7 Módulo de processamento e representação do conhecimento	41

4	Implementação do Sistema <i>DistSense</i>	43
4.1	Cenário de aplicação: Sistema de monitorização distribuído para o reconhecimento de atividades domésticas	44
4.2	Simulação	45
4.3	Inicialização do sistema <i>DistSense</i>	48
4.3.1	Serviço de descoberta	49
4.3.2	Processo de eleição	51
4.4	Comunicação dos nós	54
4.4.1	Mensagens, estrutura e procedimento	56
4.5	Aprendizagem computacional	58
4.5.1	Pré-processamento do Conjunto de Dados Audiovisuais	59
4.5.2	Treino dos modelos	60
4.5.3	Conversão do modelo	63
4.6	Processamento e representação do conhecimento	64
5	Testes e Avaliação	70
5.1	Testes em ambiente virtualizado	71
5.2	Testes em ambiente real	71
5.3	Discussão	72
6	Conclusão	73
6.1	Resultados	73
6.2	Trabalho futuro	73
	Bibliografia	74

Listas de Figuras

2.1	Arquitetura de referência - Modelo em camadas	8
2.2	Implementação da infraestrutura de um sistema exemplo no GNS3 (Official Website)	11
2.3	a) Colaboração entre os nós de um sistema distribuído; b) Colaboração entre cliente-servidor de um sistema centralizado	13
2.4	Parâmetros comuns a avaliar em casas inteligentes	16
2.5	Arquitetura <i>edge-fog-cloud</i> apresentada no estudo de (Cao et al., 2019) . .	17
2.6	Arquitetura centralizada de (Rupasinghe and Maduranga, 2022)	22
3.1	Vista geral da arquitetura do sistema <i>DistiSense</i>	30
3.2	Módulos de funcionamento de um nó do sistema <i>DistiSense</i>	33
3.3	Fluxo de dados do sistema <i>DistiSense</i>	34
3.4	Funcionamento do protocolo mDNS (Al-Fuqaha et al., 2015b)	37
3.5	Fluxo de dados no treino dos modelos especializados na deteção de eventos audiovisuais do sistema <i>DistiSense</i>	39
4.1	Processo de construção e registo de um contentor <i>docker</i>	46
4.2	Configuração do contentor <i>docker</i> no GNS3	48
4.3	Registo de um serviço com auxilio da biblioteca <i>zeroconf</i> em <i>python</i> . .	50
4.4	Descoberta de serviços com auxilio da biblioteca <i>zeroconf</i> em <i>python</i> . .	50
4.5	Eleição do nó coordenador	52
4.6	Captura de pacotes <i>keep alive</i> através da aplicação <i>wireshark</i>	56
4.7	Mensagem padrão no formato JSON em <i>python</i>	57
4.8	Fluxo do processo de pré-processamento do conjunto de dados audiovisuais	59
4.9	Janela deslizante utilizada na implementação do modelo <i>YAMNet</i>	61
4.10	Pré-processamento dos dados para inferência do modelo <i>MoViNet</i> em <i>python</i>	64
4.11	Estrutura de cada bloco da <i>blockchain</i> do sistema <i>DistiSense</i>	65
4.12	Fluxo de registo de transação na <i>blockchain</i>	66
4.13	Mensagem enviada pelo nó coordenador via MQTT para o HA	68
4.14	Regra de automação no <i>Home Assistant</i> em <i>python</i>	69

Lista de Tabelas

1.1	Cenários de aplicação no contexto de casas inteligentes	2
2.1	Comparação entre arquiteturas adotadas em sistemas de monitorização . .	14
2.2	Comparação entre sistemas de monitorização em casas inteligentes	27
3.1	Requisitos funcionais do sistema <i>DistSense</i>	31
3.2	Requisitos não funcionais do sistema <i>DistSense</i>	31
3.3	Requisitos de sistema do sistema <i>DistSense</i>	32

Definições e Acrónimos

AAL *Ambient Assisted Living*

ADL *Activities of Daily Living*

AI *Artificial Intelligence*

ANN *Artificial Neural Network*

API *Applicattion Programming Interfaces*

CBR *Case-Based Reasoning*

CNN *Convolutional Neural Network*

CRUD *Create, Read, Update, Delete*

CV *Computer Vision*

DHCP *Dynamic Host Configuration Protocol*

DHT *Distributed Hash Table*

DL *Deep learning*

DNN *Deep neural network*

DNS-SD *Domain Name System - Service Discovery*

DT *Decision Tree*

edge AI *Edge Artificial Intelligence*

e-Health *Electronic Health*

GDPR *General Data Protection Regulation*

GNS3 *Graphical Network Simulator 3*

HA *Home Assistant*

IoT *Internet of Things*

IP *Internet Protocol*

JSON *JavaScript object notation*

LSTM *Long Short Term Memory*

mDNS *Multicast DNS*

ML *Machine Learning*

MQTT *Message Queuing Telemetry Transport*

P2P *Peer-to-Peer*

RGB *Red, Green and Blue*

RNN *Recurrent Neural Network*

R-CNN *Region - Convolutional Neural Network*

SDK *Software Development Kit*

SMAF *Functional Autonomy Measurement System*

SSL *Secure Sockets Layer*

SVM *Support Vector Machine*

TF *TensorFlow*

TFL *TensorFlow Lite*

TCP *Transmission Control Protocol*

TL *Transfer learning*

TLS *Transport Security Layer*

UDP *User Datagram Protocol*

UML *Unified Modeling Language*

UPnP *Universal Plug and Play*

UUID *Universal Unique Identifier*

VM *Virtual Machine*

Capítulo 1

Introdução

1.1 Enquadramento e motivação

De acordo com a Organização Mundial de Saúde(OMS, 2022), entre 2020 e 2030, cerca de 500 milhões de pessoas desenvolverão doenças cardíacas, obesidade, diabetes ou outras doenças não transmissíveis devido à inatividade física.

Este cenário gera a necessidade de criar uma solução que permita prever este tipo de problemas e que promova a saúde e a independência na população em geral, principalmente na população mais idosa.

Considerando que nos dias de hoje uma grande parte da população, acima de 65 anos, precisará de algum tipo de assistência, a longo prazo, para os seus próximos anos de vida, torna-se evidente que o planeamento de um sistema inteligente de cuidados prolongados, em termos financeiros, de gestão e de tecnologia são mais urgentes do que nunca.

Perante estas razões, este estudo tem como foco desenvolver um sistema na vertente *Electronic Health (e-Health)*, com objetivo de fornecer previsões sobre padrões comportamentais num contexto doméstico, de forma a melhorar a sua qualidade de vida.

Atualmente num contexto de *Internet of Things* (IoT), mais especificamente em *Ambient Assisted Living* (AAL), existe um conjunto de técnicas de deteção de atividades com recurso a sensores e dispositivos inteligentes, entre as demais encontra-se a deteção com recurso a câmaras de vídeo e microfones integrados em sistemas inteligentes de deteção.

A utilização e integração destas tecnologias através técnicas de visão computacional em conjunto com técnicas de *Artificial Intelligence* (AI), permitem focar no deteção de padrões de atividades visando obter uma previsão de modo a melhorar a qualidade de vida dos utilizadores.

Todavia, é necessário ter em consideração que cada utilizador é diferente e por essa razão podem existir vários cenários ímpares com diversas atividades domésticas.

Outro problema, é a privacidade e confiabilidade do sistema, uma vez que através da captação de imagens é necessário garantir ao utilizador que está protegido e que não

existem fugas de informação para o exterior.

É com estes problemas em foco, que o presente estudo se realiza de forma a implementar uma solução de baixo custo, capaz de contribuir na resolução dos desafios inerentes à deteção de padrões em atividades domésticas, com recurso a câmaras de vídeo RGB e microfones integrados em dispositivos inteligentes.

Alguns dos desafios que os sistemas de deteção propõem solucionar encontram-se detalhados na tabela 1.1.

Cenário de Aplicação	Descrição
Deteção de quedas	A distribuição estratégica de sensores pela residência inteligente permite uma monitorização precisa de movimentos e acelerometria em diferentes divisões. Através de algoritmos avançados e colaboração entre os agentes, o sistema distingue de forma eficiente entre quedas reais e movimentos normais, reduzindo falsos alarmes. Ao detetar uma queda, aciona um alarme de emergência e envia notificações aos cuidadores ou serviços médicos, promovendo uma resposta rápida em situações críticas. A colaboração entre os agentes garante uma deteção precisa e confiável de quedas, minimizando falsos negativos e contribuindo para a segurança e bem-estar dos residentes.
Assistência pessoal	Mediante a integração de dispositivos e sensores inteligentes em toda a residência, o sistema distribuído almeja prover assistência pessoal num contexto de casas inteligentes. Desde a regulação da iluminação, temperatura e segurança, até a automação de tarefas domésticas, tais como o preparo de alimentos e a gestão de compromissos, o referido sistema viabiliza uma experiência personalizada e conveniente para os moradores, facilitando as suas atividades diárias. Como resultado, a vida dos utilizadores neste contexto é aprimorada, tornando-a mais cómoda e confortável.
Análise de hábitos de sono	Através da captação de dados recolhidos pela integração de dispositivos e sensores inteligentes, o sistema deve ser capaz de identificar padrões de sono e fornecer informação sobre a qualidade do sono de forma a otimizar os hábitos de descanso do utilizador. Este sistema contribui para uma melhor compreensão e promoção de um sono saudável e reparador, proporcionando benefícios significativos para o bem-estar e a qualidade de vida dos moradores.
Análise de padrões comportamentais	Com o intuito de identificar padrões comportamentais, como horários de sono, momentos de maior atividade física, uso de determinados dispositivos, entre outros. Através dos dados recolhidos pelos sensores inteligentes é possível retirar informações valiosas sobre o estilo de vida e bem-estar dos moradores, permitindo a personalização de serviços e a deteção de eventuais anomalias que possam indicar problemas de saúde ou situações de risco. Perante essas razões, o sistema contribui para o monitorização e a promoção de um ambiente residencial mais seguro, saudável e adaptado às necessidades individuais dos moradores.

Table 1.1: Cenários de aplicação no contexto de casas inteligentes

1.2 Objetivos

O objetivo principal desta pesquisa consiste no desenvolvimento de um sistema distribuído para detecção e classificação de padrões de atividades, através da captura de imagens e áudio em tempo real mantendo em consideração a privacidade e segurança como requisitos fundamentais do sistema.

Para alcançar este objetivo, será implementada uma arquitetura *Peer-to-Peer* (P2P) para a transmissão e processamento de dados capturados, com o auxílio de redes neurais criadas com algoritmos de *Machine Learning* (ML).

Cada nó processa a informação obtida através dos seus sensores, tendo apenas como saída eventos de alto nível.

Os eventos obtidos serão analisados por um nó coordenador responsável pelo processamento dos dados, que extrairá eventos relevantes para o utilizador, esse processo assegurará que dados sensíveis, como sequências de imagens, sejam armazenados exclusivamente em ambiente local, preservando a privacidade e segurança do sistema.

Ao longo das últimas décadas o paradigma de computação evoluiu e permitiu o desenvolvimento de várias aplicações, neste estudo mantém-se a discussão do paradigma de computação em relação a IoT. Desta forma, todos os aspectos descritos ao longo dos capítulos seguintes levam em consideração três restrições de projeto pré-estabelecidas:

- Privacidade e Segurança de dados: dados sensíveis devem ser sempre protegidos quando armazenados, transferidos ou processados, de acordo com as regras europeias de proteção de dados (Consulting, 2020).
- Custo: como a solução deve garantir escalabilidade para cenários distintos, o custo unitário deve ser considerado na escolha entre abordagens e tecnologias.
- Conexão de Rede: visto que a solução deve garantir comunicação entre todos os dispositivos, é necessário considerar o tipo de comunicação e protocolos que se devem abordar.

Posto isto, propõe-se a utilização de um conjunto de sensores audiovisuais integrados em dispositivos *Jetson Nano*, para criar um sistema de monitorização inteligente de baixo custo, tendo em conta que os sensores e os dispositivos *Jetson Nano* são recursos comuns e amplamente disponíveis no mercado. Além disso, propõe-se implementar algoritmos de ML e inteligência artificial existentes para processar localmente a informação recolhida e apenas enviar eventos relevantes para o utilizador, sem comprometer a sua privacidade.

1.3 Metodologia

A utilização de sistemas inteligentes tem crescido exponencialmente nas últimas décadas, e com ele a necessidade de avaliar e compreender seu funcionamento e desempenho.

Uma metodologia científica amplamente utilizada para testar estes sistemas é o método experimental, este método permite testar hipóteses sobre como o sistema funciona e se o mesmo é capaz de realizar tarefas específicas, de forma eficiente.

O processo é dividido em várias etapas, incluindo planeamento, execução, análise e interpretação dos resultados.

Desta forma, a metodologia que se propõe utilizar nesta dissertação é sustentada, num primeiro momento, por analisar e testar todo o sistema em ambiente controlado, de forma a colmatar possíveis lacunas do sistema.

A combinação de metodologias experimentais e de estudo de caso permite uma avaliação mais completa e abrangente do sistema de inteligência artificial. Além disso, outras técnicas como a simulação em ambiente controlado, podem ser utilizadas para testar o sistema em diferentes cenários e condições, o que é crucial para avaliar a robustez e escalabilidade do sistema.

É importante destacar que a recolha e análise de dados deve ser realizada de forma ética e segura, sendo importante ter em conta o impacto social e ético do uso do sistema.

O método de estudo de caso, permite-nos compreender como os sistemas inteligentes se comportam em situações reais e permite-nos desenvolver soluções mais eficazes e adaptadas às necessidades dos utilizadores (Clausen et al., 2018).

Perante estas razões mencionadas, é possível compreender as vantagens de utilizar este método, por esta mesma razão é proposto aplicar o estudo de caso no terreno, com utilizadores reais, cujos resultados serão transpostos para uma análise crítica e comparativa, a qual assenta em afirmações referidas no estado de arte.

Por fim, será aplicado um inquérito por questionário aos utilizadores que participaram neste estudo de caso, de forma a compreender a usabilidade e desempenho do sistema.

Para este caso específico, é efetuado, igualmente, um levantamento exaustivo e minucioso por meio de testes em ambientes simulados e reais, com o intuito de avaliar todo o sistema. Esse processo abrangente permite avaliar diversos aspectos do sistema, como desempenho, usabilidade e comportamento em situações reais. Além disso, é importante considerar o impacto social e ético do uso do sistema, garantindo que a recolha e análise de dados sejam realizadas de forma ética e segura

1.4 Contribuições

Nos dias atuais e num contexto *Internet of Things*, o conceito *edge intelligence* ainda está sob forte discussão. Dado o estado de desenvolvimento crescente dessa abordagem de computação, poucas soluções representativas apresentam implementações reais de um paradigma de *edge intelligence*.

Desta forma, com o objetivo de fazer parte desse movimento de tarefas computacionais mais complexas para as camadas inferiores da rede e, com foco em *edge AI*,

apresenta-se uma arquitetura de rede cujo foco é assegurar a privacidade e segurança dos dados dos utilizadores. Cada módulo de rede e canal de comunicação é cuidadosamente definido e descrito, mantendo em consideração os requisitos estabelecidos. Complementando essa especificação, analisamos os cenários de aplicação e avaliamos os méritos da arquitetura proposta.

Nesse sentido, algumas das principais contribuições podem ser destacadas:

- Especificação de uma arquitetura de sistema completa de acordo com os pré-requisitos definidos;
- Especificação de um protocolo de mensagens para coordenação de entidades físicas;
- Deteção de padrões baseado no reconhecimento de atividades em ambiente doméstico;
- Especificação de um sistema capaz de garantir um controlo de versões de dados armazenados;
- Disponibilizar a implantação do sistema em ambiente de simulação para um ambiente de produção através da utilização de contentores *docker*;

Estas contribuições visam impulsionar o desenvolvimento e a adoção de sistemas inteligentes, que ofereçam benefícios significativos para a qualidade de vida e o bem-estar dos utilizadores, ao mesmo tempo em que asseguram a proteção e privacidade de informações pessoais.

1.5 Estrutura do trabalho

Este documento está organizado em seis capítulos distintos: Introdução, Trabalhos relacionados em ambientes inteligentes, Arquitetura do sistema *DistSense*, Implementação do sistema *DistSense*, Testes e Avaliação, e, por fim, a Conclusão.

No primeiro capítulo, é apresentado o contexto do problema e a solução proposta pelo sistema *DistSense*, sendo que o objetivo principal é introduzir a importância da monitorização em ambientes domésticos, destacando os potenciais benefícios de um sistema inteligente distribuído nesse contexto.

De seguida, no capítulo dois, é realizada uma revisão de literatura abrangente, concentrando-se nos sistemas de monitorização em ambientes domésticos, com o intuito de explorar diferentes aspectos relacionados à monitorização em casas inteligentes, incluindo o uso de simulação para facilitar a avaliação do sistema, a arquitetura de referência em sistemas de monitorização e uma revisão da literatura existente sobre sistemas de monitorização em casas inteligentes já implementados. Essa revisão da literatura visa fornecer uma base sólida para a proposta de arquitetura do sistema denominado *DistSense*.

Posteriormente, no capítulo três, descreve-se detalhadamente a arquitetura proposta para o sistema *DistSense*, onde são apresentados os diferentes módulos e componentes que o compõem, incluindo sensores, inicialização na rede, captura e processamento dos dados e a representação do conhecimento ao utilizador. Além disso, são discutidos os requisitos específicos do sistema a serem implementados, garantindo a sua eficácia.

No quarto capítulo, é abordada a implementação dos vários módulos descritos no capítulo da arquitetura do sistema, sendo apresentados os detalhes técnicos e as decisões de projeto adotadas durante o desenvolvimento do sistema. A implementação considera aspetos como inicialização do sistema, comunicação entre os dispositivos, processamento de dados em tempo real e representação do informações para o utilizador.

O capítulo cinco é dedicado à avaliação exaustiva do desempenho do sistema *DistSense* num ambiente de simulação controlado e num ambiente real, a fim de validar a eficácia e a precisão do sistema na deteção de eventos em tempo real. Os resultados obtidos são analisados e processados em relação ao cenário de aplicação implementado.

No último capítulo, são apresentados os resultados obtidos através da implementação e avaliação do sistema *DistSense*. São discutidas as contribuições e as limitações do trabalho realizado, além de serem sugeridas melhorias e possíveis desenvolvimentos futuros. Para além disso, é realizado um balanço sobre o sistema proposto acerca do seu potencial como uma solução inteligente para monitorização de ambientes domésticos, oferecendo benefícios significativos em termos de segurança, conforto e escalabilidade.

Capítulo 2

Trabalhos relacionados em ambientes inteligentes

Este capítulo, apresenta uma revisão de literatura abrangente sobre sistemas de monitorização que utilizam um conjunto de sensores para a deteção de padrões e/ou atividades em ambientes inteligentes. A crescente adoção de tecnologias de monitorização tem impulsionado a pesquisa e o desenvolvimento de soluções inovadoras nessa área, procurando aprimorar a compreensão e a gestão em diversos contextos.

A revisão de literatura é dividida em duas vertentes principais: sistemas distribuídos e sistemas centralizados.

Os sistemas distribuídos são caracterizados pela descentralização das operações e pela colaboração entre múltiplos componentes, conectados por uma rede, permitindo partilha de informações e recursos, sendo que essa arquitetura oferece maior escalabilidade e flexibilidade, minimizando o ponto único de falha.

Por outro lado, os sistemas centralizados adotam uma abordagem onde todas as operações e decisões são concentradas numa entidade central. Esta centralização simplifica a coordenação das operações, tornando a implementação e manutenção do sistema mais diretas, contudo, esta arquitetura é mais suscetível a falhas, pois a avaria da entidade central pode paralisar todo o sistema. Além disso, a escalabilidade pode ser um desafio, uma vez que o aumento do número de componentes pode sobrecarregar o ponto central de processamento.

A revisão de literatura visa fornecer uma compreensão aprofundada dos avanços recentes na área de *Ambient Assisted Living*, destacando as características, os benefícios e as limitações de cada abordagem. Além disso, serão discutidos os desafios enfrentados pelos sistemas de monitorização e as possíveis direções futuras para a pesquisa e o desenvolvimento. Desta forma, será possível compreender o estado atual da pesquisa e identificar possíveis direções para o avanço da tecnologia de monitorização em ambientes inteligentes.

2.1 Arquitetura de referência

A arquitetura implementada nos sistemas de monitorização em casas inteligentes, desempenha um papel crucial na recolha, processamento e análise dos dados provenientes dos diferentes sensores e dispositivos presentes no ambiente inteligente. Esses sistemas são responsáveis por supervisionar e monitorizar o funcionamento dos dispositivos, bem como fornecer informações úteis para a tomada de decisões dos moradores.

Na figura 2.1, é apresentada uma arquitetura comumente implementada nos sistemas de monitorização em casas inteligentes, descrita como arquitetura de rede em camadas. Esta abordagem divide o sistema em diferentes camadas, cada uma com uma função específica, com o intuito de permitir uma melhor organização dos componentes e facilitar a escalabilidade e a manutenção do sistema.

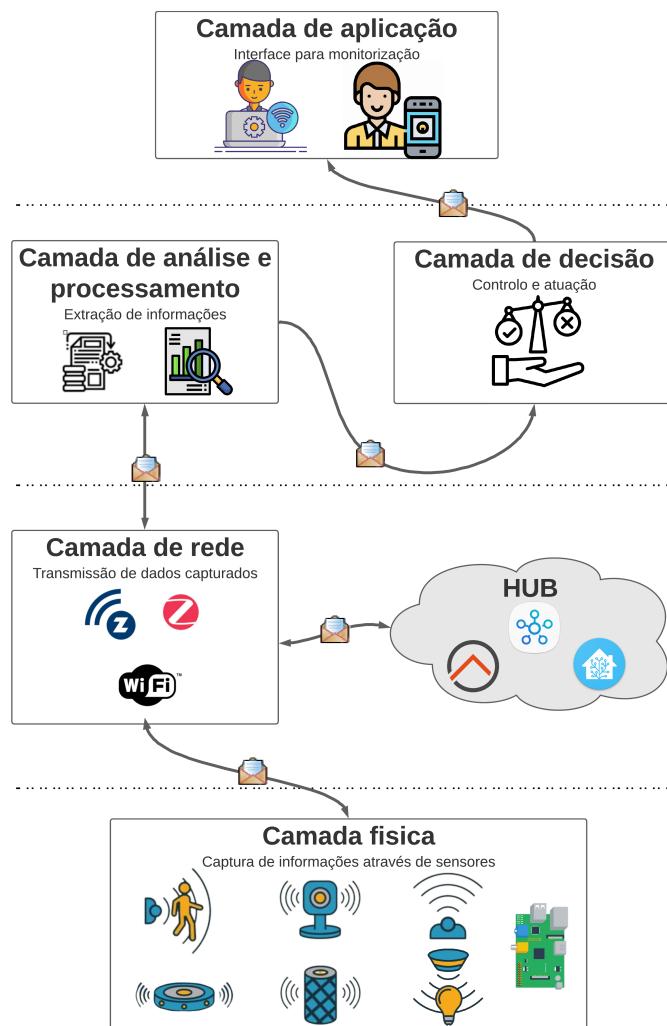


Figura 2.1: Arquitetura de referência - Modelo em camadas

Na camada inicial da arquitetura, denominada de camada física, estão presentes vários sensores com o propósito de realizar a captura de dados. Esses sensores são integrados em dispositivos inteligentes, tais como a *Raspberry Pi* ou *Jetson Nano*, onde a função

primordial é adquirir informações provenientes do ambiente inteligente e encaminhá-las para a camada subsequente da arquitetura.

A segunda camada é a camada de rede, onde os dados recolhidos pelos sensores são enviados para um dispositivo central, como um *HUB* ou um controlador, por meio de tecnologias de comunicação, como *Wi-Fi*, *Bluetooth* ou *Zigbee*, garantindo a conectividade entre os dispositivos e o sistema central de monitorização.

A camada seguinte é referente à camada de análise e processamento, na qual os dados adquiridos pela camada anterior são processados e analisados para extrair informações relevantes. Algoritmos e técnicas de inteligência artificial podem ser aplicados nesta etapa para identificar padrões, realizar previsões e tomar decisões automatizadas com base nos dados recolhidos.

A quarta camada é a camada de decisão, onde as decisões tomadas com base na análise dos dados são convertidas em ações, por exemplo, se a temperatura ambiente estiver acima de um limite definido, o sistema pode acionar automaticamente o sistema de ar condicionado para controlar a temperatura ambiente. Essa camada permite a automação e o controlo remoto dos dispositivos presentes na casa inteligente.

Por fim, a quinta camada é a camada de aplicação para a qual são disponibilizadas interfaces intuitivas e amigáveis, como aplicações para telemóveis inteligentes, que permitem aos moradores interagir com o sistema de monitorização e controlar os dispositivos de forma fácil e conveniente.

Estudos que adotaram o modelo arquitetural descrito demonstram que esta implementação viabiliza a criação de residências inteligentes, que se destacam por serem eficientes, seguras e personalizadas, proporcionando aos seus moradores uma experiência confortável e inteligente.

2.2 Simulação em ambiente virtual

A simulação de sistemas em ambiente virtual tem-se tornado uma abordagem cada vez mais relevante e promissora na área de monitorização. Em contraste com os testes realizados em ambiente real, a simulação oferece uma série de vantagens, permitindo a análise e avaliação de sistemas complexos, de forma mais conveniente, eficiente e económica.

A simulação de sistemas em ambiente virtual é um processo que envolve a criação de um modelo computacional que reproduz o comportamento de um sistema real, sendo que esse modelo pode ser desenvolvido ao utilizar-se uma variedade de técnicas, como a modelagem matemática, a física computacional e a computação gráfica.

Nesse contexto, um sistema de monitorização é um conjunto de dispositivos e sensores que recolhem informações sobre o ambiente e os processos em tempo real, essas informações são analisadas e utilizadas para tomar decisões, identificar anomalias, prever comportamentos futuros e fornecer informações valiosas para a manutenção do sistema.

Uma das principais vantagens da simulação é a possibilidade de ter controlo total sobre as condições de teste, cenários e parâmetros do sistema, isso permite a realização de testes em diferentes configurações, variando as cargas de trabalho e emulando condições extremas, as quais seriam difíceis ou arriscadas de se reproduzir num ambiente real, essa flexibilidade viabiliza uma análise aprofundada do comportamento do sistema e a otimização dos seus componentes antes da implementação em larga escala.

Além disso, a simulação é um processo rápido e eficiente para avaliar o desempenho e a eficácia de um sistema, ao contrário da implementação num contexto real, que necessita de tempo e recursos consideráveis, esta metodologia permite a execução ágil de experiências num ambiente virtual, acelerando o processo de desenvolvimento, possibilitando a identificação precoce de problemas e a realização de ajustes necessários para aprimorar o sistema.

Outra vantagem importante da simulação é a segurança e redução de riscos, uma vez que ao emular o sistema é possível mitigar riscos e garantir a proteção dos dados e recursos envolvidos.

Num ambiente virtual, os impactos de falhas e erros são controlados, permitindo a realização de testes em condições seguras, dessa forma, evita-se possíveis anomalias num ambiente real, enquanto se tem a oportunidade de avaliar a robustez e a resiliência do sistema, sem comprometer a integridade da infraestrutura ou a confidencialidade dos dados.

Adicionalmente, a simulação oferece a possibilidade de recolher dados detalhados sobre o desempenho, a eficiência e a escalabilidade do sistema distribuído inteligente, essas informações podem ser analisadas minuciosamente, identificando obstáculos, pontos de melhoria e oportunidades de otimização.

A utilização do *Graphical Network Simulator 3* (GNS3) e contentores *Docker* para realizar a simulação de sistemas em ambiente virtual é uma abordagem prática e eficiente, tendo em conta que essas ferramentas fornecem recursos poderosos para criar ambientes de rede virtualizados e executar aplicações em contentores, permitindo simular diferentes componentes de um sistema e testar todo o comportamento em condições controladas.

O *Graphical Network Simulator 3* (GNS3) é uma plataforma avançada e versátil de simulação de redes, que permite a criação de topologias complexas de rede por meio da virtualização de *routers*, *switches*, *firewalls* e outros dispositivos de rede.

Através do GNS3, é possível recriar a interconexão e o fluxo de dados entre esses dispositivos, proporcionando uma representação virtual que se assemelha de forma próxima a um ambiente de rede real (Neumann, 2015).

O GNS3 utiliza uma abordagem baseada em emulação e virtualização para reproduzir as funcionalidades dos dispositivos de rede no ambiente virtual, isso é alcançado por meio da utilização de imagens de sistemas operativos específicos, que são executadas dentro do GNS3, simulando o comportamento dos dispositivos de rede reais.

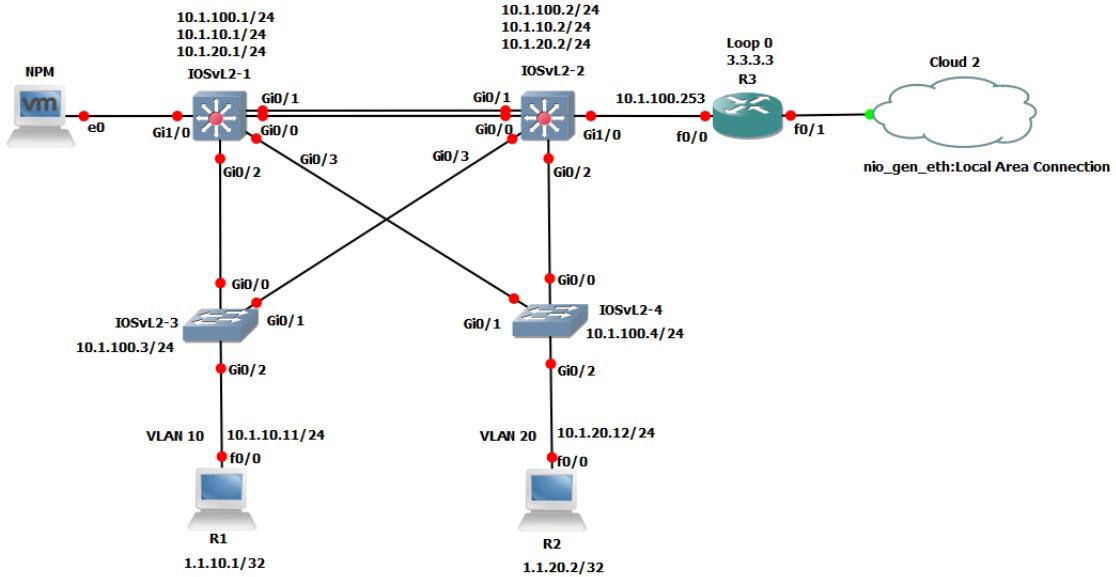


Figura 2.2: Implementação da infraestrutura de um sistema exemplo no GNS3 (Official Website)

Por exemplo, é possível importar e configurar imagens de *Cisco routers*, *switches* ou *firewalls* dentro do GNS3 e interligá-los de acordo com a topologia desejada. Através dessa configuração, ilustrada no exemplo prático 2.2, é possível realizar simulações e testes de conectividade, protocolos de *routing*, políticas de segurança e outras funcionalidades de rede, de forma controlada e precisa.

Além disso, o GNS3 oferece suporte a recursos avançados, como a captura e análise de pacotes em tempo real, permitindo a observação detalhada do tráfego de dados entre os dispositivos virtuais, isso proporciona uma visão aprofundada do comportamento da rede simulada, facilitando a detecção de problemas, a otimização de configurações e a validação de implementações de rede (Neumann, 2015).

Uma tecnologia adicional para a simulação e virtualização é representada pelos contentores *docker*, os quais se caracterizam como uma tecnologia de virtualização de caráter leve, que possibilita o encapsulamento e isolamento de aplicações, juntamente com as dependências em ambientes isolados.

Esses contentores são capazes de simular e executar os diversos componentes do sistema, tais como servidores, serviços e aplicações, de forma independente e escalável, permitindo que ao utilizar os mesmos, se torne viável a criação de ambientes de simulação consistentes e replicáveis, o que facilita o processo de teste e análise de distintas configurações e cenários (Anderson, 2015).

Através da criação de um ambiente simulado controlado, é possível realizar testes automatizados e ágeis utilizando o GNS3 e contentores *docker*, fazendo uso de *scripts* e nós de simulação.

Essas ferramentas fornecem recursos poderosos para simular sistemas complexos em

condições controladas, permitindo uma análise e avaliação mais conveniente, eficiente e económica.

A automação dos testes, por meio da utilização de *scripts*, proporciona a execução de testes coerentes, assegurando, assim, a qualidade do sistema de monitorização, sendo que estes *scripts* podem abranger testes unitários, de integração e até mesmo testes de carga, contribuindo para a validação do desempenho e estabilidade do sistema.

Adicionalmente, a adoção de contentores *docker* simplifica o processo de implantação do sistema num ambiente real e ao encapsular o sistema juntamente com suas dependências num contentor isolado, é possível garantir a consistência entre os ambientes de desenvolvimento, teste e produção.

Neste sentido, os contentores *docker* fornecem uma abordagem padronizada e portável para o encapsulamento do sistema, o que torna o processo de implantação mais rápido e fácil de ser realizado.

Ao combinar a utilização do GNS3 com os contentores *docker*, é possível criar simulações mais abrangentes e realistas de sistemas de monitorização, esta abordagem permite a simulação de uma infraestrutura de rede complexa no GNS3, utilizando dispositivos virtuais como *routers* e *switches*, onde posteriormente, dentro desses dispositivos, permite executar contentores *docker* que representam serviços e aplicações específicas do sistema de monitorização.

Perante estas razões, a integração entre o GNS3 e os contentores *docker* oferece várias vantagens, incluindo a possibilidade de testar o desempenho da rede, avaliar a interoperabilidade entre os componentes do sistema, simular tráfego de dados realista e explorar diferentes configurações e cenários sem impactar o ambiente de produção para a deteção precoce de problemas e para a entrega de um produto final de elevada qualidade aos utilizadores, ao mesmo tempo que se reduz os riscos e a complexidade associada à implantação em ambientes produtivos.

2.3 Sistemas Cliente-Servidor vs Sistemas *Peer-to-Peer*

A monitorização em ambientes inteligentes é uma área em crescimento que visa proporcionar soluções tecnológicas com o intuito de aprimorar a segurança, o conforto e a eficiência energética das habitações.

O termo "Ambientes inteligentes" refere-se a espaços residenciais ou comerciais equipados com dispositivos conectados e sensores, que recolhem dados e permitem o controlo remoto para melhorar a experiência dos utilizadores.

Neste contexto, surgem duas abordagens arquiteturais: sistemas centralizados e sistemas distribuídos. Num sistema centralizado de monitorização de ambientes inteligentes, todas as informações e funcionalidades são concentradas e processadas num único ponto de controlo, como um servidor ou *HUB*, responsável por recolher dados provenientes de

sensores espalhados pela casa e controlar os dispositivos conectados, permitindo que os utilizadores monitorizem e interajam remotamente com o ambiente inteligente.

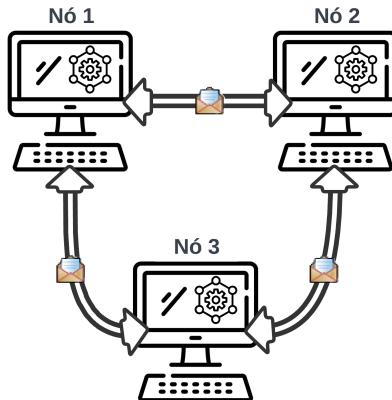


Figura 2.3: a) Colaboração entre os nós de um sistema distribuído; b) Colaboração entre cliente-servidor de um sistema centralizado

Os sistemas centralizados que adotam uma arquitetura cliente-servidor, tal como ilustrado na figura 2.3, têm sido amplamente adotados em espaços inteligentes, proporcionando uma abordagem tradicional para a monitorização e controlo de dispositivos e ambientes domésticos inteligentes.

Nesta abordagem, o cliente representa qualquer dispositivo habilitado a interagir com o ambiente inteligente na rede doméstica. Estes dispositivos podem assumir a forma de sensores, atuadores ou outras unidades inteligentes que recolhem informações sobre o ambiente e enviam os dados resultantes ao servidor central. Adicionalmente, os dispositivos clientes podem enviar solicitações ao servidor com o intuito de obter informações ou aceder a serviços específicos.

O servidor central assume a responsabilidade de receber, armazenar e analisar os dados capturados por esses dispositivos e sensores. Essas informações são então utilizadas para fundamentar a tomada de decisões, que podem abranger uma variedade de ações e respostas, conforme necessidades ou requisitos específicos do utilizador.

Além disso, o servidor central assume a tarefa de enviar comandos de volta aos dispositivos clientes. Esses comandos são projetados para instruir os dispositivos a executarem as ações e tarefas solicitadas pelo utilizador ou em conformidade com as decisões tomadas com base nos dados analisados. Desse modo, o sistema de monitorização avançado torna-se capaz de proporcionar uma experiência mais eficiente e personalizada ao utilizador, além de contribuir para uma gestão inteligente e automatizada dos recursos e equipamentos no ambiente doméstico ou qualquer outro contexto em que a abordagem centralizada seja aplicada.

A comunicação entre o cliente e o servidor geralmente ocorre através da rede local, onde o cliente pode utilizar o *HUB*, fornecido pelo fabricante ou terceiros, para interagir com o ambiente inteligente, essa aplicação estabelece a conexão com o servidor central e

permite que o utilizador visualize o estado do ambiente, receba alertas, defina preferências e envie comandos de acordo com as suas necessidades e preferências.

Em comparação com a abordagem centralizada, os sistemas distribuídos emergem como uma alternativa que proporciona maior flexibilidade e escalabilidade.

A implementação destes sistemas num ambiente residencial inteligente, permite que cada dispositivo possa atuar como um nó autónomo, sendo capaz de recolher dados provenientes de sensores e executar ações com base nesse conhecimento. Estes dispositivos têm a capacidade de comunicar e colaborar entre si, partilhando informações e coordenando atividades para oferecer uma monitorização eficiente e equilibrada da habitação.

Nesse sentido, os sistemas distribuídos que implementam uma arquitetura *Peer-to-Peer* (P2P) caracterizam-se por distribuir as funções de monitorização, processamento e controlo entre múltiplos dispositivos interconectados na rede doméstica, permitindo que todos os dispositivos inteligentes atuem tanto como clientes quanto como servidores.

A abordagem distribuída oferece vantagens significativas, na medida em que a escalabilidade é um dos benefícios proeminentes, permitindo a adição de novos dispositivos à rede sem sobrecarregar um único ponto de controlo.

Adicionalmente, a redundância dos recursos distribuídos aumenta a resiliência do sistema, garantindo uma maior disponibilidade e continuidade da monitorização, mesmo em caso de falhas.

Contudo, esta abordagem apresenta alguns desafios, como a coordenação e gestão dos dispositivos pelo que requerem algoritmos e protocolos adequados. A comunicação entre os dispositivos pode ser afetada por atrasos e falhas na rede, exigindo mecanismos de tolerância a falhas e estratégias de recuperação eficazes.

Critério	Arquitetura Cliente-Servidor	Arquitetura <i>Peer-to-Peer</i>
Escalabilidade	-	++
Interoperabilidade	+-	++
Eficiência	++	++
Segurança	-	++
Integridade	+-	+

Table 2.1: Comparação entre arquiteturas adotadas em sistemas de monitorização

Ambas abordagens visam proporcionar ambientes residenciais mais seguros, eficientes e convenientes. Para interpretar melhor as características de cada uma é realizada uma análise comparativa qualitativa entre ambas as arquiteturas, presente na tabela 2.1, para a qual foram considerados critérios de avaliação como a escalabilidade, interoperabilidade,

eficiência, segurança e integridade do sistema.

A arquitetura cliente-servidor, embora possa oferecer uma visão abrangente do ambiente doméstico, demonstra algumas limitações. Em termos de escalabilidade, esta abordagem enfrenta dificuldades à medida que o número de dispositivos monitorizados aumenta, pois todo o processamento e armazenamento de dados são centralizados numa única entidade. Por outro lado, a arquitetura P2P destaca-se pela sua alta escalabilidade, permitindo a distribuição de tarefas de monitorização em nós descentralizados e facilitando o aumento flexível da capacidade de monitorização à medida que novos dispositivos são adicionados.

No que diz respeito à interoperabilidade, a arquitetura cliente-servidor pode enfrentar desafios na gestão de diferentes protocolos de comunicação ou sistemas heterogéneos, onde a padronização dos dados e protocolos pode ser complexa.

Em contrapartida, a arquitetura P2P pode lidar melhor com a interoperabilidade, pois cada nó distribuído pode ser adaptado para interagir com dispositivos específicos, facilitando a troca de informações através de protocolos padronizados.

Em termos de eficiência, ambas as arquiteturas, cliente-servidor e P2P, apresentam pontos fortes. A arquitetura cliente-servidor pode consolidar todos os dados num único ponto, facilitando a análise e tomada de decisões. No entanto, isso pode levar a perdas de desempenho e atrasos na transferência de dados para o ponto central.

A arquitetura P2P destaca-se em termos de latência e utilização de recursos, visto que cada nó distribuído pode processar localmente os dados capturados, reduzindo a carga do sistema e permitindo uma resposta mais rápida em tempo real.

A segurança e a integridade dos dados são dois aspectos intrinsecamente ligados nos sistemas de monitorização em espaços inteligentes domésticos. Ao adotar uma arquitetura cliente-servidor, embora seja possível consolidar todas as informações num único ponto e estabelecer regras e políticas centralizadas, também existe o risco de criar um ponto único de falha que pode ser alvo de ataques cibernéticos. No caso de ocorrer uma violação de segurança nesse sistema centralizado, todos os dados podem ser comprometidos, afetando tanto a confidencialidade como a integridade das informações.

Em compensação, a arquitetura P2P, ao dispersar os dados e as tarefas de monitorização em diversos nós, apresenta uma maior resiliência em termos de segurança. Se um nó distribuído for comprometido, apenas uma parte do sistema será afetada, preservando a confidencialidade e a integridade dos demais dados de monitorização.

Em última análise, a seleção da arquitetura dependerá dos objetivos de monitorização, das características do ambiente inteligente e da importância atribuída a cada critério de avaliação, visto que seja qual for a escolha, ambas as abordagens têm o potencial de contribuir para ambientes residenciais mais seguros, eficientes e convenientes, contribuindo para a evolução da tecnologia em benefício dos utilizadores.

2.4 Sistemas de monitorização

Atualmente o termo "casas inteligentes" está gradualmente a tornar-se parte integrante do nosso quotidiano. A *Internet of Things* (IoT) acrescenta uma nova dimensão a este conceito, não sendo surpreendente que existam mais dispositivos conectados à IoT do que seres humanos, demonstrando o seu impacto significativo no dia-a-dia do utilizador.

Com o avanço da tecnologia ao longo dos últimos anos, tornou-se viável a recolha de dados sobre as atividades do quotidiano do utilizador. No entanto, interpretar esses dados e reconhecer as ações correspondentes, de maneira eficiente, continua a ser um desafio.

Nesse sentido, diversos estudos abordam sistemas que implementam técnicas de *Machine Learning* (ML), as quais têm-se mostrado promissoras ao lidar com a complexidade e variabilidade dos dados obtidos.

Os sensores desempenham um papel fundamental nos sistemas de casas inteligentes, sendo componentes essenciais para a recolha de dados e monitorização de diversos parâmetros e atividades no ambiente residencial, fornecendo informações úteis ao utilizador.

A figura 2.4 ilustra alguns dos parâmetros a analisar em casas inteligentes, os dados capturados pelos sensores abrangem desde sinais vitais, como frequência cardíaca e pressão arterial, até o controlo do ambiente inteligente, como temperatura e luminosidade.

Adicionalmente, com a utilização de sensores inteligentes é possível capturar atividades realizadas pelo utilizador ao longo do seu dia-a-dia, obtendo uma compreensão mais aprofundada dos padrões comportamentais e da interação com o ambiente residencial inteligente.



Figura 2.4: Parâmetros comuns a avaliar em casas inteligentes

À medida que os sensores recolhem informações sobre as atividades diárias dos utilizadores, como rotinas, preferências e comportamentos, surge a questão de como esses dados são utilizados e protegidos.

Há receios de que terceiros possam ter acesso indevido a essas informações pessoais, o que possibilita a violações de privacidade e até mesmo ao uso indevido dos dados.

A segurança dos dados é uma preocupação crítica no contexto de casas inteligentes,

visto que os dados recolhidos pelos sensores podem ser transferidos e armazenados em servidores remotos, criando potenciais vulnerabilidades para ataques cibernéticos e violações de segurança. Os utilizadores estão preocupados com a possibilidade de que informações sensíveis, como dados biométricos ou padrões de comportamento, possam ser comprometidas e exploradas por agentes maliciosos.

Outra questão importante é a aceitabilidade dos sensores intrusivos por parte dos utilizadores. Embora esses sensores possam fornecer benefícios significativos, como automação conveniente e personalização das configurações domésticas, alguns utilizadores podem sentir-se desconfortáveis com a ideia de ter dispositivos que monitorizam constantemente as suas atividades e capturam dados pessoais sensíveis.

A invasão da privacidade e a sensação de vigilância constante podem gerar resistência e relutância em adotar essas tecnologias.

Assim, surgem duas abordagens principais em relação ao sensoriamento: sensores intrusivos e sensores não intrusivos. Estas abordagens distinguem-se pela forma como são implementadas e pelo grau de invasão ou alteração exigido no ambiente residencial.

Diversos estudos e artigos têm explorado estas duas abordagens em sistemas para casas inteligentes, procurando encontrar soluções eficientes e adequadas às necessidades dos utilizadores.

Alguns destes trabalhos combinam a utilização de sensores intrusivos e não intrusivos, com o intuito de obter uma visão mais abrangente e precisa no contexto do ambiente inteligente, uma vez que estas abordagens híbridas podem, por exemplo, conjugar a instalação de sensores intrusivos, em pontos estratégicos, para monitorizar o ambiente de forma não intrusiva.

No âmbito da monitorização de casas inteligentes, surgem conceitos importantes, como *fog computing*, *edge computing* e *cloud computing*, ilustrados na figura 2.5, que desempenham papéis fundamentais no processamento e armazenamento de dados.

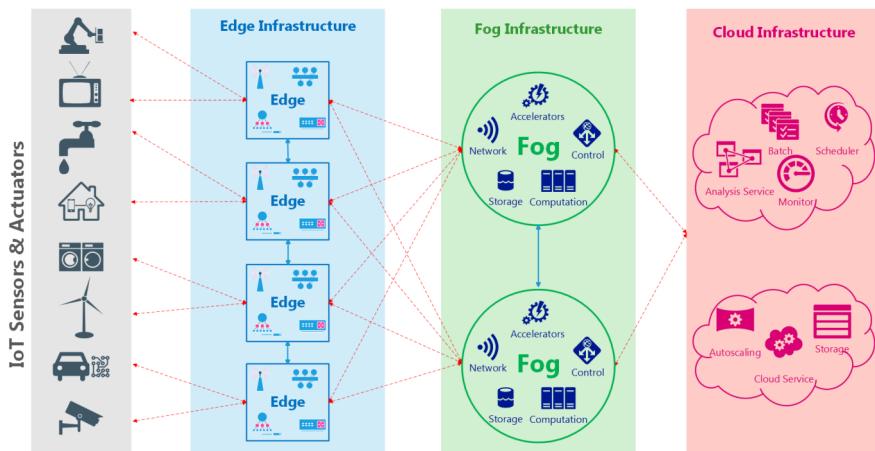


Figura 2.5: Arquitetura *edge-fog-cloud* apresentada no estudo de (Cao et al., 2019)

A *cloud computing* trata-se de um paradigma amplamente utilizado, que permite o armazenamento e processamento de dados em servidores remotos acessíveis pela *internet*, deste modo, os dados capturados pelos dispositivos de monitorização numa casa inteligente podem ser enviados para a nuvem, onde são armazenados e processados, permitindo análises avançadas e o acesso remoto aos dados, facilitando a tomada de decisões informadas e fornecendo uma visão abrangente do estado da casa.

Embora essa abordagem ofereça benefícios em termos de capacidade de armazenamento, análises avançadas e acesso remoto aos dados, também suscita preocupações relacionadas à privacidade, sendo que ao enviar os dados de monitorização para a nuvem, existe a possibilidade de que informações pessoais sensíveis sejam armazenadas em servidores de terceiros, aumentando o risco de violações de privacidade, especialmente se medidas de segurança adequadas não forem implementadas.

No entanto, em alguns casos, é imperativo realizar um processamento mais célere e eficiente dos dados recolhidos, neste contexto, o paradigma de *edge computing* desempenha um papel crucial. Ao contrário da *cloud computing*, o *edge computing* envolve a realização de tarefas de processamento e análise de dados diretamente nos dispositivos de monitorização ou em servidores próximos a estes, permitindo reduzir significativamente a latência e a sobrecarga na rede, possibilitando a respostas rápidas e ações imediatas com base nas informações obtidas.

Por exemplo, caso um sensor detete uma anomalia numa habitação inteligente, o dispositivo de *edge computing* pode agir instantaneamente para garantir a segurança do ambiente.

Adicionalmente, esta abordagem traduz-se numa redução da necessidade de transferir informações sensíveis para a nuvem, o que contribui para minimizar os riscos associados ao armazenamento de dados pessoais em servidores remotos, proporcionando maior controlo sobre a privacidade dos utilizadores.

Para otimizar ainda mais o processamento e a gestão de dados em ambientes residenciais inteligentes, entra em jogo o conceito de *fog computing*, desempenhando um papel essencial. Neste cenário, parte do processamento ocorre nos dispositivos de *edge computing*, enquanto outra parte é realizada em servidores localizados em pontos intermédios, denominados *fog gateways*.

Os *gateways* encontram-se estrategicamente mais próximos da nuvem do que os dispositivos de ponta, permitindo uma combinação eficiente de processamento local e acesso aos recursos da nuvem. Essa abordagem possibilita equilibrar a carga de trabalho entre os dispositivos de ponta e a nuvem, tirando proveito da capacidade de processamento local e da escalabilidade da nuvem, o que garante uma monitorização mais eficiente e contínua das habitações inteligentes.

Este enfoque controlado na distribuição de informações e no processamento, fornece um ambiente mais seguro para os dados recolhidos, aumentando a resiliência do sistema,

para que quando ocorram falhas, estas não comprometam a totalidade dos dados do utilizador.

É fundamental enfatizar que a capacidade de resposta em tempo real dos sistemas de *edge computing* constitui um dos principais pilares desta abordagem, uma vez que se traduz numa redução significativa da latência, permitindo respostas praticamente instantâneas em diversas situações.

Além disso, a redução da comunicação entre dispositivos periféricos e a nuvem contribui para a poupança de recursos de rede e energia consumida pelos nós, sendo esta característica especialmente relevante em ambientes de domótica inteligentes, onde a eficiência energética assume uma preocupação preponderante.

Diversos estudos têm sido conduzidos para explorar as capacidades e benefícios dos sistemas de monitorização em ambientes inteligentes, esses estudos abrangem várias áreas, incluindo saúde, eficiência energética e segurança.

Na área da *e-Health*, há uma ampla gama de estudos que investigam os sistemas de monitorização em ambientes inteligentes, visando aprimorar o acompanhamento e cuidado dos pacientes, com o propósito de melhorar o bem-estar e qualidade de vida do utilizador.

Nesse contexto, os sistemas de monitorização em casas inteligentes são desenvolvidos para monitorizar sinais vitais, como pressão arterial, batimentos cardíacos, níveis de glicose, entre outros, permitindo um acompanhamento remoto contínuo do estado de saúde dos pacientes.

Na investigação de (Tewell et al., 2019), os autores descrevem o desenvolvimento de uma *framework*, designada por *SCAMPI* (*Self-Care Advice, Monitoring, Planning, and Intervention*). Essa *framework* é composta por vários sensores inteligentes, com o objetivo de permitir que pessoas com demência e/ou doença de *parkinson* monitorizem atividades significativas e atividades do quotidiano de forma autónoma.

Para isso, os autores propuseram a utilização de sensores de baixo custo e facilmente disponíveis no mercado, a fim de garantir a aceitação por parte dos utilizadores.

A informação recolhida pelos sensores é armazenada numa base de dados local e, posteriormente, é realizada uma análise crítica dos dados capturados, a qual permite fornecer uma visão da progressão da doença, uma vez que os padrões de atividade variam conforme o estado de saúde do utilizador se deteriora.

O sistema opera com uma variedade de protocolos de comunicação, como *Wi-Fi*, *Bluetooth*, *Zigbee* e *z-wave*, para permitir a integração dos diversos sensores. Um elemento essencial do sistema é a *Raspberry Pi 3*, que atua como um *HUB* central para a monitorização das atividades.

Nesse *HUB* central, os autores optaram por executar o *Home Assistant* (HA) na *Raspberry Pi 3*, possibilitando a integração dos diferentes protocolos de comunicação e a captura dos dados dos sensores, sendo que esta tecnologia desempenha um papel cen-

tral no processamento e armazenamento dos dados capturados, permitindo a realização de análises críticas para identificar padrões e tendências relevantes para a progressão da doença.

Os dados recolhidos pelos sensores são armazenados numa base de dados e posteriormente analisados localmente de forma crítica, a fim de fornecer informações valiosas sobre a progressão da doença e as mudanças nos padrões de atividade relacionados à saúde do utilizador.

Para testar essa abordagem realizaram duas avaliações distintas à *framework* proposta: um estudo em laboratório para testar a instalação do sistema, incluindo a precisão e colocação dos sensores e um estudo em ambiente real, no qual pessoas que não eram utilizadores-alvo, mas que se identificavam como entusiastas de tecnologia, avaliaram a viabilidade da *framework* para monitorizar atividades em e ao redor de residências inteligentes reais.

O público-alvo do estudo, em ambiente real, relatou obstruções mínimas durante a instalação e foram capazes de realizar e desfrutar de atividades do dia-a-dia sem serem prejudicados pelos sensores, revelando que atividades significativas podem ser monitorizadas, remotamente utilizando sensores passivos e acessíveis.

No entanto, os autores afirmam que através do uso de sensores menos invasivos não é possível determinar, com algum grau de certeza, as atividades que o utilizador está a executar, como poderia ser feito através da utilização de sensores mais invasivos, em detrimento da privacidade, como por exemplo sensores audiovisuais.

No estudo conduzido por (Rajan Jeyaraj and Nadar, 2022), os autores propõem um sistema que aborda a questão de obter um serviço de saúde confiável por meio de previsões precisas do provedor de serviços. A arquitetura proposta neste sistema é composta por módulos, descritos da seguinte forma:

1. Módulo de monitorização: Este módulo é composto por um conjunto de sensores inteligentes vestíveis em conjunto com o processador *myRio*, uma vez que o mesmo atua como um dispositivo semelhante à *Raspberry Pi*, mas com capacidade limitada de realizar várias tarefas;
2. Módulo de processamento: Neste módulo, o processador *myRio* transmite os dados adquiridos pelos sensores utilizando o protocolo *Wi-Fi*;
3. Módulo de visualização e armazenamento de dados: Utilizam a plataforma *EVOTH-INGS*, este módulo recebe os dados do módulo de processamento e permite a visualização e análise dos resultados, possibilitando a monitorização contínua dos sinais fisiológicos do paciente.

Para extrair as características dos sinais fisiológicos recebidos pelo conjunto de sensores, os autores propuseram uma abordagem de *Deep neural network* (DNN), composta

por três camadas, seguindo critérios como *linear separability* na primeira camada, conectividade total na segunda camada e construção do modelo de *Deep learning* na terceira camada, em que características como o tempo de aquisição do sinal fisiológico e a magnitude do sinal foram consideradas.

Como forma de validar o sistema, os autores realizaram uma comparação com outros trabalhos relacionados, utilizando diferentes modelos, onde concluíram que o sistema proposto apresenta uma precisão de, aproximadamente, 97,2%, o que indica que o sistema permite garantir a monitorização e a previsão precisa dos sinais fisiológicos monitorizados.

Além de função na extração de informações sobre os sinais vitais dos utilizadores, os sistemas de monitorização também têm desempenhado um papel fundamental no acompanhamento e análise das atividades físicas do quotidiano do utilizador.

Esse sistemas são desenvolvidos com o propósito de adquirir e registar dados pertinentes referentes à atividade física, possibilitando uma avaliação mais precisa e minuciosa do desempenho e comportamento dos utilizadores.

Um exemplo interessante é o estudo de (Rupasinghe and Maduranga, 2022), que descreve o desenvolvimento de um sistema baseado em IoT para monitorização em tempo real das atividades físicas de pessoas idosas, utilizando acelerómetros.

O objetivo subjacente desta pesquisa consiste em superar a limitação de capacidade de monitorização das atividades físicas, particularmente devido ao crescimento do setor socioeconómico, o qual tem resultado num rápido aumento no número de idosos que vivem em áreas remotas, tais como casas de repouso.

Nessas circunstâncias, a saúde de pessoas idosas vulnerabiliza-se devido à diminuição das capacidades motoras e/ou cognitivas. O presente projeto de pesquisa pretende satisfazer a necessidade de um sistema capaz de recolher detalhes vitais dos utilizadores, através de um dispositivo de pulso, de baixo custo capaz de capturar o movimento da mão em três eixos distintos.

Adicionalmente, os autores referem que foi necessário que o sensor escolhido fornecesse leituras precisas de forma consistente, além de ter uma taxa de deteção adequada e baixa latência para operar em tempo real.

No desenvolvimento desta investigação, foi identificada a necessidade de um nó principal, responsável por processar as leituras dos sensores e utilizar técnicas de ML supervisionado para reconhecer as atividades executadas pelo utilizador.

Um dos desafios encontrados durante desenvolvimento desta investigação, foi garantir a conectividade entre o dispositivo vestível IoT e o nó principal, para solucionar essa questão, foi utilizado um *router Wi-Fi* compatível com o padrão *IEEE 802.11b/g/n*, permitindo a transmissão dos sinais detetados para o processamento e reconhecimento das atividades.

A escolha cuidadosa do *hardware* e a atenção à conectividade, são elementos essen-

ciais para o funcionamento adequado deste sistema. Essas considerações garantem que as leituras dos sensores sejam processadas de forma eficiente e precisa, possibilitando a deteção e classificação das atividades físicas dos utilizadores monitorizados.

Na figura 2.6, é ilustrada a vista geral da arquitetura desta investigação. O sistema é composto por um dispositivo vestível IoT, um nó principal, um servidor em nuvem e uma aplicação móvel.

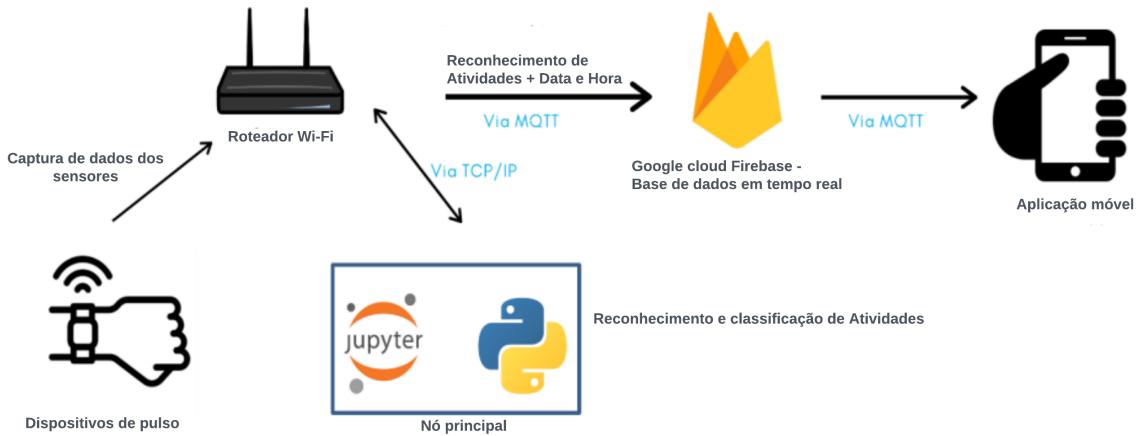


Figura 2.6: Arquitetura centralizada de (Rupasinghe and Maduranga, 2022)
[Figura adaptada para português de Portugal]

O dispositivo vestível IoT desenvolvido, é responsável por medir as flutuações na aceleração do utilizador e transmitir essas informações em tempo real para o nó principal. A conexão entre o dispositivo vestível IoT e o nó principal é estabelecida por meio de um *router Wi-Fi*, onde utilizam o protocolo TCP/IP para a transmissão de dados.

O nó principal desempenha o papel central no sistema, processando os sinais recolhidos e fornecendo previsões ou resultados com base neles, sendo, posteriormente, transmitidas para um servidor em nuvem via *Message Queuing Telemetry Transport* (MQTT).

Outro aspecto significativo deste estudo é a utilização de técnicas de aprendizagem computacional para reconhecer atividades simples como caminhar, sentar, dormir e ficar em pé. O uso destas técnicas é avaliado para superar as barreiras do reconhecimento de atividade em tempo real. O algoritmo de classificação implementado pelos autores foi o algoritmo de árvore de decisão, uma vez que possibilita obter uma precisão do modelo superior a 80%.

Além disso, outra área de interesse na monitorização de ambientes inteligentes é a eficiência energética, onde sistemas centralizados permitem a gestão e controlo inteligente do consumo de energia, otimizando a sua utilização e reduzindo custos.

No artigo de (Franco et al., 2021), os autores propõem uma abordagem intrusiva para a monitorização de consumos energéticos, através de um sistema de reconhecimento de atividades baseado numa arquitetura IoT que combina soluções avançadas nos seus difer-

entes módulos.

A arquitetura referida é composta por cinco camadas: dispositivos, percepção, comunicação, *middleware* e aplicação. Através da implementação desta arquitetura, é garantida a escalabilidade geral do sistema, permitindo suportar diferentes aplicações domésticas, como a classificação das atividades diárias.

Na camada de dispositivo, é estabelecido um fluxo de dados que é posteriormente enviado para a camada superior, designada de camada de percepção, responsável pela aquisição dos dados.

Posteriormente, a camada de *middleware* atua como intermediário entre a interação dos dispositivos IoT e os aplicativos de *software*, funcionam como uma camada de abstração entre a interface do utilizador e todos os dispositivos implantados. No topo da arquitetura encontra-se a camada de aplicação, que se refere aos serviços específicos dedicados aos utilizadores.

Os modelos de ML são aplicados no módulo de reconhecimento do dispositivo, para o qual foram testados três modelos diferentes utilizando o conjunto de dados denominado por *United Kingdom-Domestic Appliance Level Electricity (UK-DALE)*: *Feedforward neural network* (FFNN), *Long Short Term Memory* (LSTM) e *Support Vector Machine* (SVM). Este conjunto de dados contém dados reais sobre dispositivos presentes em cinco casas distintas em Londres. Durante o processo de treino do modelo, os autores consideraram apenas informações da primeira residência, devido à variação no número de dispositivos presentes em cada habitação.

Adicionalmente, realizaram mais alguns testes para analisar o impacto do tamanho do grupo na precisão do classificador de ML. Esses grupos consistiam num número fixo de amostras a partir das quais os aparelhos inteligentes são identificados, se o tamanho do grupo diminuir ou aumentar espera-se o mesmo comportamento em relação à precisão, exceto para o modelo LSTM, que aumenta a precisão quando o tamanho do grupo é maior.

Uma crítica referente a esta abordagem é que o sistema proposto não funciona em tempo real, no entanto é capaz de identificar atividades simultâneas ao analisar sequências de vetores de recursos, independentemente dos intervalos de tempo.

A precisão dos classificadores FFNN e LSTM ficou acima de 90%, enquanto a do SVM ficou em torno de 80%. Com base nos resultados obtidos, os autores concluíram que é preferível realizar um novo treino do classificador com dados atualizados antes de implantar o sistema em pleno funcionamento.

Com a crescente adoção de casas inteligentes e a utilização generalizada de dispositivos conectados surgem preocupações significativas em relação à privacidade e segurança dos dados. À medida que as tecnologias avançam, sensores intrusivos são cada vez mais integrados nos ambientes domésticos, levantando questões sobre a aceitabilidade por parte dos utilizadores.

Para solucionar estas preocupações, o estudo de (Fleck and Straßer, 2010) propõe

uma abordagem baseada em câmaras inteligentes distribuídas para a monitorização sensível à privacidade no cuidado de idosos. Os autores apresentam uma arquitetura de um sistema distribuído que inclui câmaras inteligentes e dispositivos de processamento de imagem para detetar eventos invulgares e rastrear a atividade dos indivíduos. O sistema foi concebido para preservar a privacidade do utilizador, incluindo medidas de segurança e anonimato.

Nesta investigação, são mencionadas algumas vantagens adicionais com a utilização de câmaras inteligentes em sistemas de monitorização para o cuidado de idosos, como a capacidade de reconhecimento facial e de objetos, que podem ser usados para detetar eventos anormais de forma mais precisa e rastrear a atividade dos utilizadores de forma mais detalhada. Desta forma o sistema pode ser expandido para incluir outros sensores, como sensores de movimento e de pressão, para melhorar ainda mais a precisão e a cobertura do sistema.

Em seguida, são discutidas as implicações éticas do uso de câmaras inteligentes no cuidado de idosos, como a questão da vigilância constante e a possibilidade de invasão da privacidade dos utilizadores, os autores argumentam que esses problemas podem ser resolvidos por meio de medidas de segurança e anonimato, bem como por meio de diálogo e consentimento explícito dos pacientes e cuidadores. Além disso, salientam a importância de como o sistema pode ser projetado para evitar o uso indevido ou a exploração dos dados capturados.

Como forma de avaliar e validar o sistema proposto, foram realizados testes em ambientes controlados e ambientes reais com pacientes idosos. Através dos resultados obtidos, o sistema desenvolvido foi capaz de detetar eventos anormais e rastrear a atividade dos pacientes com precisão, e que o sistema foi bem recebido pelos pacientes e cuidadores onde a privacidade foi adequadamente protegida.

No estudo (Facchini et al., 2020), procuram detetar comportamentos maliciosos numa rede IoT através da colaboração entre os dispositivos. Para isso, é utilizado um sistema de deteção de Intrusão Distribuído (IDS) que implementa um classificador binário baseado em ML para analisar recursos extraídos de dados do *kernel*, da rede e da *Distributed Hash Table* (DHT).

Os autores propõem uma arquitetura *Peer-to-Peer* (P2P) composta por dispositivos inteligentes, como televisões inteligentes, colunas inteligentes e frigoríficos inteligentes, que são responsáveis por um conjunto de dispositivos não tão inteligentes, como sensores de temperatura espalhados pela casa, usados pelo sistema de aquecimento inteligente. Estes dispositivos inteligentes são interconectados por meio de uma rede doméstica comum e compartilham dados de aplicativos numa DHT, em que colocam periodicamente dados relacionados com o seu comportamento, num determinado período de tempo. Cada nó inteligente contém um agente do sistema de deteção de intrusão que examina o comportamento dos outros nós em três níveis diferentes: *kernel*, rede e DHT.

Para detetar comportamentos maliciosos, é utilizado um mecanismo de reputação distribuído, para o qual o nó que deteta um comportamento malicioso de outro nó, coloca um recurso na DHT contendo essa informação para convidar outros nós a excluí-lo da rede. De maneira a evitar que nós maliciosos declarem indevidamente nós benignos como maliciosos, os autores sugerem a utilização de um mecanismo de reputação distribuído para atribuir uma pontuação de reputação de forma colaborativa.

Para facilitar a operabilidade do sistema, os autores utilizam um algoritmo popular de DHT denominado de *Kademlia*, que reduz as mensagens de introdução entre os nós, configura automaticamente os nós de rede e direciona as consultas por caminhos de baixa latência. Essa abordagem contribui significativamente para a eficiência e agilidade do sistema como um todo.

Com o aumento da utilização de câmaras inteligentes em ambientes domésticos, o desenvolvimento de algoritmos para redes distribuídas tem despertado bastante interesse.

O reconhecimento de ações humanas em múltiplos pontos de vista enfrenta desafios como invariância de visualização e oclusão. No entanto, devido à enorme quantidade de dados processados e comunicados em aplicações do mundo real, adaptar esses algoritmos para redes de câmaras inteligentes é uma tarefa complexa.

Nesse sentido, a investigação de (Mosabbeb et al., 2013), apresenta uma ferramenta de classificação de atividades distribuída. Neste estudo, várias câmaras observam uma cena e cada uma processa as suas próprias observações, chegando a um acordo sobre a atividade que ocorre por meio da comunicação com outras câmaras.

Esse método baseia-se em completude de matriz por consenso, utilizando otimização convexa para realizar a completude de matriz distribuída.

O foco deste estudo, é o reconhecimento de atividades humanas, e o desempenho é testado nos conjuntos de dados *IXMAS* e *MuHAVi* para demonstrar sua viabilidade.

Os resultados obtidos com o algoritmo demonstram a eficácia nos conjuntos de dados utilizados, alcançando uma precisão de 85,9%. A completude de matriz mostra-se como uma ferramenta valiosa para lidar com dados ruidosos e variações nas atividades humanas.

Como perspectivas futuras, sugerem realizar os procedimentos de treino e teste incrementalmente, para resumir grandes volumes de dados em matrizes menores, isso poderá proporcionar maior eficiência computacional e menor tempo de processamento. Além disso, explorar outras abordagens de otimização e técnicas de aprendizagem computacional pode contribuir para aprimorar o desempenho desta ferramenta em ambientes distribuídos complexos.

Outra área que tem emergido como um tópico de pesquisa em evidência em ambientes inteligentes é a aplicação de tele-assistência e monitorização. Estes serviços têm como objetivo inferir o estado dos pacientes através de arquiteturas centralizadas que recolhem dados de um conjunto de sensores instalados no ambiente residencial. No entanto, quando o cenário aumenta em tamanho e o número de pacientes a monitorizar aumenta, esses

sistemas normalmente enfrentam dificuldades ao processar todos os dados associados e fornecer resultados razoáveis em tempo real.

Nesse sentido, o artigo de (Navarro et al., 2018) apresenta o conceito de uma plataforma distribuída de AAL que visa acionar alarmes com base na deteção acústica de eventos específicos em diversos ambientes residenciais, como áreas residenciais, residências particulares e lares. A arquitetura proposta é inspirada no paradigma de *fog computing* e é projetada para dividir as camadas de deteção, processamento e acionamento de alarmes, atendendo às necessidades crescentes em termos de cobertura de área e tempo de resposta dos sistemas modernos de AAL.

A plataforma é capaz de lidar com cenários de grande escala e foi adaptada para atender aos requisitos da organização sem fins lucrativos *Fundación Ave María*. Além disso, foi implementado um sistema de classificação automática de eventos acústicos sobre essa arquitetura distribuída. O processo de classificação é dividido em duas etapas. A primeira etapa baseia-se numa *Artificial Neural Network* (ANN) e realiza uma deteção de eventos acústicos em tempo real, com uma precisão geral de 85,4% e uma pontuação F1 de 71%. A segunda etapa de classificação considera a evolução temporal dos eventos detectados em um intervalo de 10 segundos, utilizando um algoritmo de *Case-Based Reasoning* (CBR) e um conjunto de heurísticas, o que permite aumentar a precisão geral do sistema para 94,6% e a pontuação F1 para 90,58%, quando duas transmissões de dados acústicos simultâneas são consideradas.

Apesar do conjunto de dados de treino limitado, o protótipo apresenta uma precisão razoável na deteção dos eventos de interesse em cenários amplos de AAL, com uma sobrecarga computacional aceitável e custos arquitetônicos limitados. Ainda há espaço para melhorias no sistema, como a aplicação de técnicas mais sofisticadas de *data mining*, *Transfer learning* ou incrementar o conjunto de dados, o que permitiria melhorar a precisão do sistema. Além disso, ajustes no modelo da ANN e do CBR após a implantação do sistema de acordo com as características do ambiente podem contribuir para otimizar o desempenho.

Como trabalho futuro, os pesquisadores planejam criar um novo conjunto de dados num ambiente real com múltiplos sensores, visando maximizar o desempenho do protótipo proposto. Além disso, procuram aprimorar a deteção de eventos acústicos, considerando todos os sensores da rede como um todo. Com esse aprofundamento, o sistema tem o potencial de se tornar uma base sólida para o desenvolvimento de serviços de AAL cada vez mais eficientes e confiáveis, promovendo uma maior qualidade de vida para idosos e pessoas dependentes no seu quotidiano.

2.4.1 Análise comparativa

A monitorização de ambientes inteligentes tem ganhado reconhecimento substancial como uma abordagem altamente promissora para aprimorar significativamente a eficiência e a segurança em contextos domésticos, ao permitir a transformação de residências convencionais em espaços altamente adaptativos e automatizados, oferecendo uma ampla gama de vantagens notáveis.

Os estudos explorados destacam o potencial dos sistemas centralizados e/ou distribuídos de monitorização em ambientes inteligentes para melhorar a saúde, a eficiência energética e o conforto dos utilizadores, permitindo uma monitorização mais eficaz e uma interação aprimorada com o espaço inteligente.

Sistema	Modelo Arquitetural	Modelos ML	Modelo Computação	Sensores	Privacidade	HUB
(Fleck and Straßer, 2010)	Distribuído	SVM	Edge	Vídeo	Sim	-
(Mosabbeb et al., 2013)	Distribuído	-	Fog	Vídeo	Sim	-
(Navarro et al., 2018)	Distribuído	ANN & CBR	Fog	Vídeo	Sim	-
(Tewell et al., 2019)	Centralizado	-	Cloud	Ambiente	Não	Home Assistant
(Facchini et al., 2020)	Distribuído	Múltiplas Abordagens	Edge	Ambiente	Sim	-
(Franco et al., 2021)	Centralizado	Múltiplas Abordagens	Fog	Ambiente	Não	-
(Rupasinghe and Maduranga, 2022)	Centralizado	Árvore de decisão	Cloud	Vestível	Não	Google Firebase
(Rajan Jeyaraj and Nadar, 2022)	Centralizado	DCNN	Cloud	Vestível	Não	EVO THINGS

Table 2.2: Comparação entre sistemas de monitorização em casas inteligentes

Na tabela 2.2 é realizada uma análise crítica com objetivo de examinar e contrastar os principais sistemas de monitorização investigados na literatura, considerando critérios fundamentais como o modelo arquitetural, modelo de ML, modelo de computação, sensores e *HUBs* utilizados.

Adicionalmente, é dada especial atenção às abordagens adotadas em relação à segurança e privacidade dos dados, considerando o cenário cada vez mais sensível e regulamentado, no contexto da proteção de informações pessoais do utilizador.

Em relação aos modelos arquiteturais explorados, verifica-se que os sistemas que utilizam sensores intrusivos, como áudio e/ou vídeo, optam por uma abordagem distribuída, o que pode ser atribuído ao benefício da descentralização do processamento, privacidade

dos dados e à redução dos pontos de falha. Os sistemas distribuídos permitem uma maior redundância, contribuindo para a melhoria da fiabilidade global do sistema, contudo, os sistemas centralizados investigados, podem oferecer maior facilidade de gestão e controlo, mas podem ser menos escaláveis em cenários de alta procura.

No que diz respeito ao modelo de computação, os sistemas de monitorização distribuídos explorados adotam um modelo *fog computing* ou *edge computing* sendo caracterizados pela descentralização do processamento e armazenamento dos dados, sendo que em vez de enviar todos os dados para um centro de processamento remoto, os dados são processados e armazenados em nós próximos aos dispositivos e sensores que os produzem. Nesta abordagem, os dispositivos inteligentes, possuem capacidade computacional para realizar tarefas de processamento básico e filtragem dos dados antes de enviá-los para um ponto central de agregação ou para a nuvem, permitindo reduzir a latência e aliviar a sobrecarga da rede, tornando a monitorização mais eficiente e em tempo real.

Por sua vez, o modelo de computação *cloud computing* adotado pelos sistemas de monitorização centralizados baseia-se na concentração de recursos e dados numa infraestrutura remota. Neste modelo, os dados são enviados pelos dispositivos locais para servidores remotos, onde são processados e armazenados. A partir destas infraestruturas, as análises mais complexas são executadas, e os resultados são enviados de volta para os dispositivos locais ou para um centro de controlo central, proporcionando grande escalaabilidade e capacidade de armazenamento, além de possibilitar a execução de algoritmos mais avançados.

Além disso, a privacidade é um aspeto crítico quando se trata da monitorização de ambientes inteligentes domésticos. Nesse contexto, é possível constatar que a maioria dos sistemas distribuídos demonstram preocupação com a proteção da privacidade dos utilizadores, utilizando mecanismos de segurança ou pedindo acesso de utilização aos utilizadores.

Através da análise comparativa dos sistemas investigados, é enfatizado dois aspectos fundamentais que são considerados essenciais no desenvolvimento do sistema *DistSense*: a privacidade dos utilizadores e a adoção de uma abordagem de uma colaboração entre os dispositivos inteligentes através da implementação de uma arquitetura distribuída.

A privacidade é um ponto crítico para garantir a confiança e segurança dos utilizadores, onde a importância de proteger os seus dados sensíveis é reconhecida, assegurando que apenas informações de alto nível sejam armazenadas após o processamento local.

Posto isto, é possível afirmar que o modelo de arquitetura distribuído é o mais apropriado para a implementação do sistema *DistSense*, visto que a colaboração entre os dispositivos inteligentes traz vantagens significativas, como representadas na tabela 2.1.

Mediante a utilização de sensores intrusivos uma abordagem distribuída é especialmente importante, visto que com a combinação de informações provenientes de múltiplas fontes, o sistema é capaz de obter uma visão mais abrangente e precisa do ambiente.

Capítulo 3

Arquitetura do Sistema *DistSense*

Este capítulo tem como objetivo fornecer uma descrição detalhada da arquitetura implementada, destacando os diferentes módulos que viabilizam a colaboração entre os vários nós na rede local, a fim de assegurar um desempenho adequado e a segurança abrangente de todo o sistema.

A arquitetura do sistema é composta por diversos componentes interligados, que cooperam para permitir a comunicação e a troca de informações entre os nós. Esses componentes incluem o módulo de inicialização e descoberta na rede, módulo de comunicação, módulo de aprendizagem computacional e o módulo de processamento e representação do conhecimento.

No que diz respeito aos requisitos na implementação do sistema, considerando os cenários de aplicação eleitos, foram estabelecidos requisitos funcionais e não funcionais que desempenham um papel fundamental na garantia do seu sucesso.

Os requisitos funcionais definem as funcionalidades específicas que o sistema deve oferecer, como a partilha de informações, a comunicação em tempo real e a deteção de padrões no quotidiano do utilizador. Já os requisitos não funcionais abrangem aspectos como o desempenho, a segurança, a escalabilidade e a usabilidade do sistema, estabelecendo padrões e objetivos a serem alcançados em cada um desses aspectos.

3.1 Introdução

Os avanços na área de monitorização têm proporcionado melhorias significativas na recolha e análise de dados sobre as atividades dos utilizadores, no entanto e através da revisão de literatura, descrita no capítulo 2, é possível observar que há mais sistemas que adotam uma arquitetura centralizada, o que pode apresentar determinadas desvantagens tais como a falta de privacidade, a segurança dos dados e a tolerância a falhas.

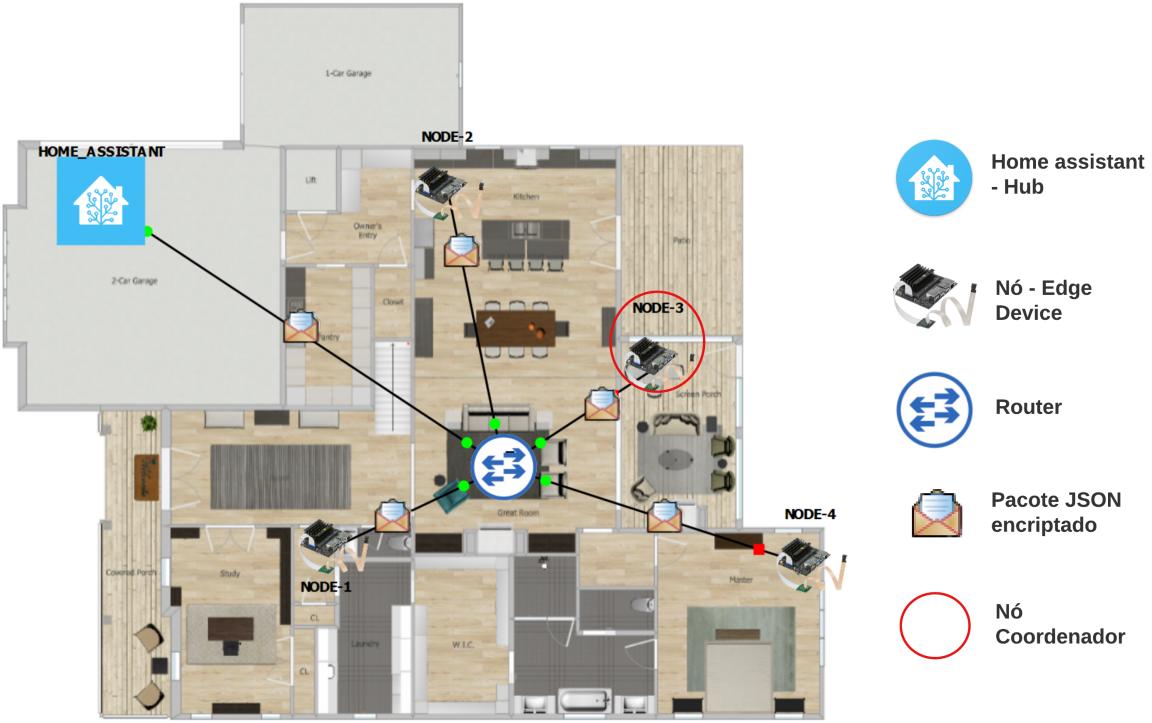


Figura 3.1: Vista geral da arquitetura do sistema *DistiSense*

Neste contexto, propõe-se uma abordagem distribuída que visa fornecer um sistema altamente seguro e confiável para a comunicação e monitorização de dados sensíveis do utilizador. Na figura 3.1, é apresentada uma visão geral da arquitetura proposta nesta investigação, na qual o sistema é composto por nós independentes, designados por *peers*, que estabelecem conexões uns com os outros para formar uma rede do tipo *Peer-to-Peer* (P2P).

Essa arquitetura distribuída permite uma interação descentralizada entre os dispositivos, promovendo maior resiliência e escalabilidade na gestão dos dados sensíveis e favorecendo a segurança e confiança no tratamento dessas informações.

3.2 Requisitos

A monitorização de sistemas desempenha um papel fundamental na garantia da estabilidade e desempenho contínuos de um ambiente computacional. Neste contexto, é essencial identificar e compreender os requisitos necessários para o correto funcionamento de um sistema distribuído para a monitorização de ambientes inteligentes.

Desta forma, os requisitos funcionais do sistema *DistiSense* são especificados na tabela 3.1, onde são descritas as funcionalidades e capacidades específicas que o sistema proposto deve possuir, a fim de responder às necessidades e objetivos do ambiente em que está inserido.

ID	Descrição dos requisitos funcionais
RF001	Descoberta de novos nós na rede doméstica
RF002	Comunicação entre todos os nós disponíveis através de protocolos e tecnologias, dependendo das características da rede e dos dispositivos envolvidos
RF003	Verificar se os nós estão ativos na rede doméstica por meio de monitorização de atividade ou uma verificação periódica de disponibilidade.
RF004	Reconhecer padrões de atividades respetivos aos casos de uso definidos
RF005	Armazenar as imagens de atividades, desde que tenha um grau de certeza que se considere correto, para que no futuro possam ser acedidas pelo sistema e/ou utilizador
RF006	Enviar alertas e/ou informações relevantes para o utilizador através de uma plataforma de automação residencial de código aberto
RF007	Armazenar em histórico, em função de uma linha temporal, informações relevantes que ocorrem durante a execução do sistema;

Table 3.1: Requisitos funcionais do sistema *DistSense*

Os requisitos não funcionais estão intrinsecamente ligados aos critérios de qualidade que validam os requisitos funcionais, estes critérios podem abranger diversos aspectos, tais como desempenho, usabilidade, confiabilidade e robustez (Tockey, 2019).

A tabela 3.2 apresenta uma descrição pormenorizada dos requisitos não funcionais, os quais englobam características como a capacidade de processamento em tempo real de grandes volumes de dados, bem como a necessidade de alta confiabilidade, a fim de evitar a perda de dados ou falhas significativas. Adicionalmente, é fundamental que o sistema seja suficientemente robusto para enfrentar adversidades, como falhas de rede.

ID	Descrição dos requisitos não funcionais
RNF001	Processar sequências de imagens em tempo real
RNF002	Detetar padrões num determinado período de tempo determinado pelo sistema
RNF003	Implementar um sistema que garanta escalabilidade, privacidade e segurança
RNF004	Implementar um sistema de baixo custo através de aplicações e bibliotecas de software gratuitas
RNF005	Implementar um sistema tolerante a falhas de rede

Table 3.2: Requisitos não funcionais do sistema *DistSense*

Na tabela 3.3 estão estabelecidas as necessidades de *software* e *hardware* essenciais para a implementação bem-sucedida do sistema de monitorização proposto.

No desenvolvimento do sistema, é requerido o uso de uma linguagem de programação

de alto nível, essa escolha visa aproveitar as vantagens oferecidas por linguagens modernas e robustas, que facilitam o desenvolvimento, a manutenção e a escalabilidade do sistema. Além disso, linguagens de alto nível proporcionam maior produtividade e flexibilidade no desenvolvimento de algoritmos e na integração com outros componentes do sistema.

Adicionalmente, é adotado algoritmos de inteligência artificial e técnicas de visão computacional. A aplicação destas técnicas permite a análise e interpretação inteligente dos dados audiovisuais capturados pelas câmaras e microfones, possibilitando o reconhecimento de padrões no quotidiano do utilizador através da identificação de eventos relevantes. A sua utilização contribui para uma melhor compreensão do ambiente monitorizado e auxiliam na tomada de decisões.

Em termos de *hardware*, o sistema requer o uso de câmaras de vídeo e microfones integrados num micro computador do tipo *Jetson Nano*, sendo este de placa única (*Single Board Computer (SBC)*) que oferece uma solução compacta, de baixo custo e com baixo consumo de energia. A capacidade de processamento e recursos de conectividade destes dispositivos tornam-nos adequados para aplicações de monitorização.

A utilização de câmaras de vídeo integradas ao *Jetson Nano* permite a captura direta das imagens e o seu processamento no mesmo dispositivo, simplificando a implementação e melhorando a eficiência do sistema.

ID	Descrição dos requisitos de sistema
RS001	No desenvolvimento do sistema utilizar uma linguagem de alto nível, como por exemplo <i>Python</i>
RS002	Utilização de algoritmos de aprendizagem computacional e técnicas de visão computacional
RS003	Utilização de câmaras de vídeo integradas num processador do tipo <i>Jetson Nano</i>

Table 3.3: Requisitos de sistema do sistema *DistSense*

Mediante os requisitos de *software* e *hardware*, é possível desenvolver e implantar um sistema de monitorização que utilize linguagens de programação modernas, algoritmos de aprendizagem computacional, bem como câmaras e microfones integrados num processador *Jetson Nano*. Essa combinação proporciona uma solução tecnologicamente avançada, capaz de processar e analisar dados de vídeo de maneira eficiente, contribuindo para uma monitorização precisa e efetiva.

3.3 Lógica de funcionamento do *DistSense*

O sistema *DistSense* tem como finalidade a aquisição contínua e a análise de dados audiovisuais pertinentes, com o propósito de monitorizar o ambiente inteligente. Através da obtenção destas informações relevantes e representativas, possui a capacidade de detectar tendências, anomalias e padrões, fornecendo dados valiosos para sustentar decisões fundamentadas.

Além disso, o sistema opera de forma distribuída, onde os nós presentes na rede colaboram de forma coordenada, sincronizando esforços para assegurar a certeza na determinação de eventos capturados.

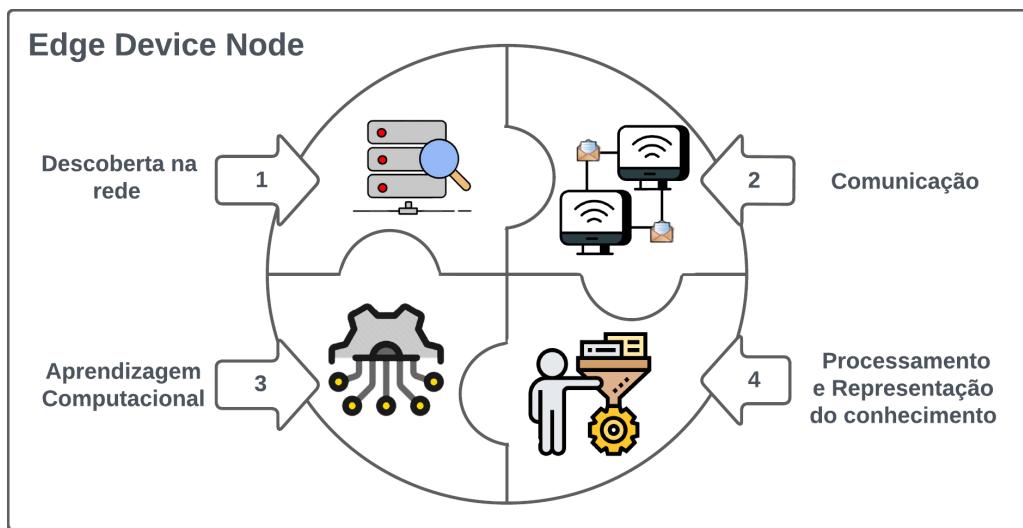


Figura 3.2: Módulos de funcionamento de um nó do sistema *DistSense*

Nesse sentido, cada dispositivo no sistema *DistSense* é composto por quatro módulos principais, representados na figura 3.2, onde desempenham funções específicas, atuando em perfeita sinergia para uma monitorização eficiente do ambiente inteligente.

Num primeiro momento, quando um *peer* é inicializado, é realizado o processo de descoberta de *peers* para encontrar nós vizinhos disponíveis na rede local, esse processo, pode ser realizado por meio de diferentes mecanismos, como protocolos de rede ou mecanismos baseados em consultas, o que permite que os *peers* se encontrem e estabeleçam conexões entre si.

Assim que tenham sido descobertos e conectados, é realizado o processo de eleição para determinar o nó coordenador, que tem como objetivo desempenhar um papel específico no sistema distribuído.

Os algoritmos de eleição são projetados para eleger um nó designado por coordenador entre o conjunto de nós em execução, de forma que em qualquer instante de tempo haja um único líder. A eleição pode ser baseada em diferentes critérios, como capacidade de processamento, disponibilidade ou até mesmo algoritmos de consenso distribuído.

Dessa forma, é possível garantir a presença de um coordenador confiável, facilitando a coordenação e a gestão adequada das tarefas a executar (Mamun et al., 2004).

Após realizado o processo de eleição, ou quando há nós suficientes na rede, os *peers* podem começar a comunicar uns com os outros, sendo que a comunicação entre os nós é realizada por meio de canais seguros e encriptados, para garantir a confidencialidade e integridade dos dados transmitidos. Essa abordagem garante que apenas os *peers* autorizados possam aceder e interpretar as informações enviadas por outros *peers*.

Posteriormente à configuração inicial da rede, cada nó tem como objetivo capturar e processar informações audiovisuais de forma eficiente, permitindo uma análise inteligente em tempo real dos eventos ocorridos no espaço monitorizado

A figura 3.3 ilustra o fluxo de dados de cada dispositivo presente no sistema *DistSense*, sendo que a fase primordial consiste na captação dos dados audiovisuais. Nesta etapa, são utilizados sensores e dispositivos cuidadosamente projetados para recolher informações relevantes do ambiente sob observação.

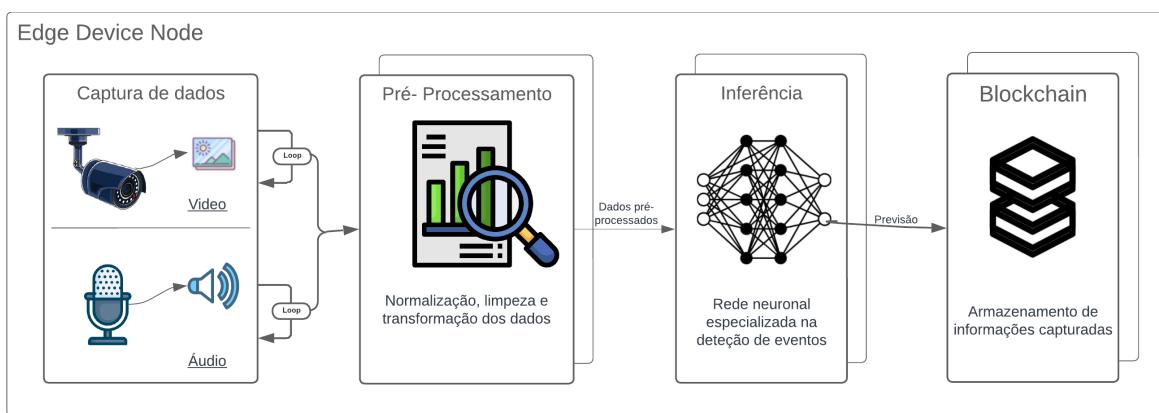


Figura 3.3: Fluxo de dados do sistema *DistSense*

Numa etapa subsequente, os dados capturados são sujeitos a um procedimento de pré-processamento, esta fase assume um papel crucial na garantia da qualidade e fiabilidade dos dados antes da sua submissão à análise. Durante o pré-processamento, os dados são normalizados, limpos e transformados. A normalização visa a padronização dos dados, garantindo que estes assumam um formato consistente e adequado para análises posteriores. Por sua vez, a limpeza tem como objetivo identificar e corrigir possíveis imperfeições e ruídos presentes nos dados, de modo a assegurar a precisão dos resultados obtidos. Em seguida, a transformação dos dados visa otimizar a sua representação, permitindo a extração eficiente de informações relevantes.

Uma vez concluído o pré-processamento local, os dados são enviados ao modelo de análise e inferência. Este modelo é desenvolvido com base em técnicas avançadas de ML, concedendo-lhe a capacidade de efetuar uma análise profunda dos dados audiovisuais, através deste processo, o modelo consegue extrair padrões, identificar eventos e fornecer

previsões acerca dos acontecimentos no espaço monitorizado, assumindo, deste modo, um papel de elevada importância na tomada de decisões e monitorização de situações críticas.

Adicionalmente, os dados capturados sobre as atividades realizadas no quotidiano pelo utilizador são armazenados de forma distribuída e descentralizada através da utilização da tecnologia *blockchain*, como é demonstrado na figura 3.5.

Além disso, com a implementação de uma arquitetura P2P, é possível identificar várias vantagens. Em primeiro lugar, elimina pontos únicos de falha, já que não depende de um servidor central para a operação do sistema. De seguida, a descentralização permite escalabilidade, uma vez que novos *peers* podem ser adicionados à rede sem afetar significativamente o desempenho global do sistema.

Neste contexto, considerando as preocupações relevantes acerca da aceitação e privacidade dos dados relativas à presença de sensores mais intrusivos, como câmaras e microfones, a implementação da tecnologia *blockchain* pode fornecer vantagens significativas, como a segurança e transparência dos dados.

Neste sistema, a *blockchain* funciona como um armazenamento cronológico dos dados recolhidos, sendo que estes ao serem recolhidos pelos sensores, são registados em blocos imutáveis, garantindo a integridade e segurança por meio de técnicas de criptografia avançadas. Esses blocos são distribuídos na rede P2P implementada, o que elimina a necessidade de confiar numa autoridade central única, tornando o sistema mais resiliente a ataques e violações de segurança.

A confidencialidade dos dados é uma preocupação alarmante, especialmente quando sensores audiovisuais são utilizados como principais métodos de recolha de informação. Por meio da tecnologia *blockchain*, é possível estabelecer mecanismos que garantem o acesso restrito aos dados sensíveis apenas por utilizadores autorizados, este comportamento é alcançado por meio de chaves criptográficas e contratos inteligentes, permitindo que os proprietários controlem os privilégios de visualização, modificação e acesso aos seus dados pessoais (Dorri et al., 2017).

O armazenamento descentralizado proporcionado por esta tecnologia permite um maior controlo, privacidade e segurança dos dados, assim, a *blockchain* pode ser utilizada para registar eventos que ocorrem num determinado período temporal, sendo mais tarde processados para detetar padrões comportamentais do utilizador, e como resultado fornecer diagnósticos personalizados, contribuindo para a melhoria da saúde e do bem-estar do utilizador.

Após a deteção de padrões comportamentais do utilizador, através do processamento dos dados, essas informações são enviadas pelo nó eleito como coordenador para o *HUB*. A utilização de um IoT *HUB* é uma abordagem que facilita a comunicação entre o sistema e o utilizador.

Um *HUB* comumente utilizado na área de IoT é o *Home Assistant* (HA), tratando-se

de um *software* de automação residencial de código aberto, desenvolvido para controlar dispositivos inteligentes em ambientes domésticos.

O HA oferece uma plataforma centralizada para integração e controlo de dispositivos de diferentes fabricantes, permitindo a criação de regras personalizadas, automação de tarefas e interação com assistentes virtuais.

De acordo com o estudo de (Akhmetzhanov et al., 2022), a utilização do software HA demonstrou ser uma solução economicamente mais vantajosa, devido à sua natureza de código aberto e à facilidade de implementação de diferentes funcionalidades que se adaptam ao contexto do utilizador.

Essa abordagem permite a criação de um sistema de monitorização eficiente e personalizável, que combina a deteção de padrões comportamentais com recursos de automação e controlo fornecidos pelo HA.

Com a utilização de dispositivos de baixo custo e a integração com o HA, é possível obter um sistema de monitorização acessível, flexível e de fácil implementação para ambientes residenciais inteligentes.

Através dos módulos definidos na figura 3.2, a abordagem colaborativa entre os dispositivos na rede amplia significativamente a precisão e fiabilidade do sistema, conferindo-lhe capacidades poderosas para a monitorização em tempo real de ambientes complexos e em constante mudança.

3.4 Módulo de descoberta na rede

A inicialização do sistema é um processo fundamental no estabelecimento e configuração de um sistema numa rede local, sendo que este processo abrange várias etapas, incluindo a identificação de dispositivos disponíveis na rede e a descoberta de vizinhos próximos para estabelecer conexões e compartilhar recursos e/ou informações. Uma abordagem comum para a descoberta de vizinhos é a utilização do protocolo *Multicast DNS* (mDNS).

O mDNS é um protocolo de descoberta de serviços utilizado em redes locais para permitir que dispositivos descubram e se comuniquem entre si sem a necessidade de um servidor *DNS* centralizado, utilizando o *multicast* para propagar pacotes *DNS* para todos os dispositivos na rede, permitindo a resolução de nomes de domínio localmente.

Essa tecnologia é amplamente utilizada em redes domésticas e em ambientes de IoT, onde a comunicação e a descoberta de serviços são essenciais. Para além disso, é um protocolo de descoberta de serviços semelhante ao *Domain Name System - Service Discovery* (DNS-SD) e *Universal Plug and Play* (UPnP). No entanto, o mDNS é mais simples e prático de configurar comparativamente ao DNS-SD e o UPnP. Além disso, é mais eficiente do que o DNS-SD e o UPnP porque utiliza menos largura de banda da rede (Cheshire and Krochmal, 2013).

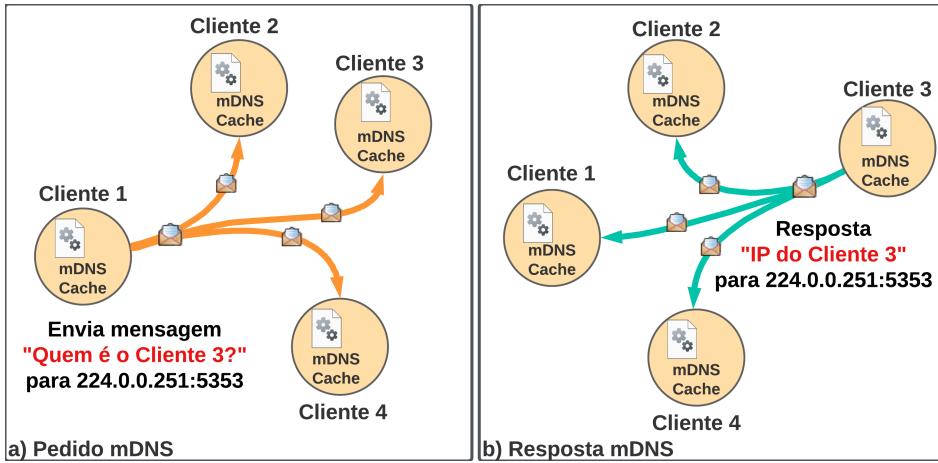


Figura 3.4: Funcionamento do protocolo mDNS (Al-Fuqaha et al., 2015b)

A *Internet of Things* requer uma arquitetura que seja independente de um mecanismo de configuração, permitindo que os dispositivos inteligentes se conectem ou desconectem da plataforma sem impactar o comportamento global do sistema. A utilização dos protocolos mDNS e DNS-SD pode viabilizar esse desenvolvimento.

Essa abordagem de *zero configuration* é particularmente útil em ambientes de rede domésticos, onde a configuração manual de serviços pode ser complicada e demorada, através do DNS-SD/mDNS, os dispositivos podem ser integrados facilmente à rede local e tornar os serviços prontamente acessíveis para outros dispositivos, simplificando a experiência do utilizador e promovendo a interoperabilidade entre os dispositivos numa rede local.

No entanto, uma limitação significativa desses protocolos reside na necessidade de armazenar em cache as entradas *DNS*, especialmente quando se trata de dispositivos com recursos limitados. Contudo, essa questão pode ser solucionada através da gestão de cache para um intervalo específico, seguido da sua limpeza. As implementações conhecidas como *Bonjour* e *Avahi* englobam tanto a utilização do protocolo mDNS quanto o DNS-SD.(Al-Fuqaha et al., 2015a)

3.5 Módulo de comunicação

Num ambiente distribuído, os nós estão espalhados em diferentes locais físicos e podem não ter uma conexão direta entre si. O protocolo de comunicação permite que os nós estabeleçam conexões e se comuniquem uns com os outros, independentemente da sua localização geográfica.

A arquitetura de comunicação adotada neste sistema é baseada num modelo P2P, onde cada nó atua simultaneamente como cliente e servidor. Essa abordagem descentralizada permite que os nós estabeleçam comunicação direta entre si, eliminando a necessidade de um servidor central.

A comunicação entre os nós é estabelecida por meio de uma rede sobreposta, na qual cada nó mantém conexões diretas com outros nós na rede, essas conexões podem ser estabelecidas utilizando uma variedade de protocolos, como UDP(*User Datagram Protocol*) ou TCP/IP, permitindo que os nós troquem mensagens e dados entre si, independentemente da topologia física da rede.

O protocolo TCP/IP é amplamente conhecido pela sua robustez e garantia de entrega confiável dos dados, devido aos mecanismos sofisticados de controlo de fluxo, confiabilidade e retransmissão de pacotes (Xylomenos and Polyzos, 1999). Essas características tornam o TCP/IP mais adequado para aplicações críticas, como sistemas de monitorização, em que a perda de dados é inaceitável e a integridade das informações é primordial. No entanto, é importante ressaltar que o TCP/IP pode impor uma sobrecarga adicional no tráfego de dados devido à necessidade de estabelecer e manter conexões persistentes.

Por outro lado, o protocolo UDP, é uma alternativa mais leve e de menor complexidade em comparação ao TCP/IP, contudo não oferece garantias de entrega confiável dos dados, o que pode resultar em pacotes perdidos ou fora de ordem. A simplicidade torna o UDP mais adequado para aplicações em que a latência é um fator crítico e a perda ocasional de pacotes pode ser tolerada, sem comprometer significativamente a eficácia do sistema de monitorização.

Para garantir a segurança na comunicação P2P, é importante adotar medidas de proteção adequadas. Uma das abordagens possíveis é utilizar o protocolo SSL/TLS para garantir a confidencialidade, integridade e autenticidade dos dados transmitidos. Este protocolo fornece uma camada de segurança adicional, encriptando os dados antes de serem enviados pela rede (Satapathy et al., 2016).

Adicionalmente, pode ser aplicada encriptação ponta-a-ponta nas comunicações entre os nós, essa encriptação pode ser baseada em algoritmos assimétricos, garantindo que somente os nós legítimos possam decifrar e verificar a integridade dos dados transmitidos.

A comunicação entre os nós é realizada por meio do envio e receção de mensagens, sendo que as mesmas são estruturadas num formato adequado, como JSON(*JavaScript object notation*), para facilitar a serialização e desserialização dos dados, isso permite que os nós troquem informações de forma eficiente e comprehensível.

A arquitetura P2P oferece vantagens em termos de escalabilidade e resiliência, uma vez que não depende de um único ponto de falha, como apresentado na tabela 2.1. No entanto, a segurança na comunicação entre os nós deve ser cuidadosamente planeada e implementada, levando em consideração as ameaças e requisitos específicos do sistema proposto.

Nesse sentido, a utilização de um protocolo de comunicação eficiente é essencial para viabilizar a comunicação direta entre os nós, permitir a troca de informações, coordenar as ações dos nós, garantir a confiabilidade e integridade da comunicação, além de prover segurança e escalabilidade ao sistema.

3.6 Módulo de aprendizagem computacional

A identificação e classificação de áudio e vídeo representam desafios significativos no campo da computação. Para abordar essas tarefas complexas, uma solução viável é o uso de redes neurais artificiais, que são sistemas computacionais inspirados nas redes neurais biológicas presentes no cérebro de animais, sendo que são capazes de aprender a executar tarefas por meio do processamento de exemplos.

No contexto da construção de redes neurais, são utilizados algoritmos de ML para criar modelos e ajustar os pesos dessas redes com base nos dados de treino disponíveis. Uma plataforma frequentemente utilizada para esse propósito é o *TensorFlow* (TF), que oferece ferramentas e recursos para a construção e execução desses modelos (Pang et al., 2020).

Dois componentes cruciais na construção de um modelo são o conjunto de dados e o próprio modelo em si. A coleção de dados é utilizada para treinar e validar o modelo. Nesse sentido, esses dados são geralmente divididos em categorias ou classes, permitindo que o modelo aprenda a reconhecer padrões e realizar classificações. Por exemplo, um conjunto de dados para treinar um modelo de identificação de sons de gatos e cães conteria amostras de áudio separadas em categorias correspondentes a esses dois animais.

O modelo, por sua vez, é a estrutura da rede neuronal composta por neurônios artificiais interconectados que processam e transmitem informações entre si. Cada neurônio possui um peso específico, que é ajustado durante o processo de aprendizagem, esses pesos são atualizados iterativamente para que o modelo seja capaz de realizar previsões cada vez mais precisas.

Modelos simples podem envolver funções lineares, enquanto modelos mais complexos, como aqueles utilizados para o reconhecimento de áudio e vídeo, requerem arquiteturas mais sofisticadas, treinadas em conjunto de dados extensos.

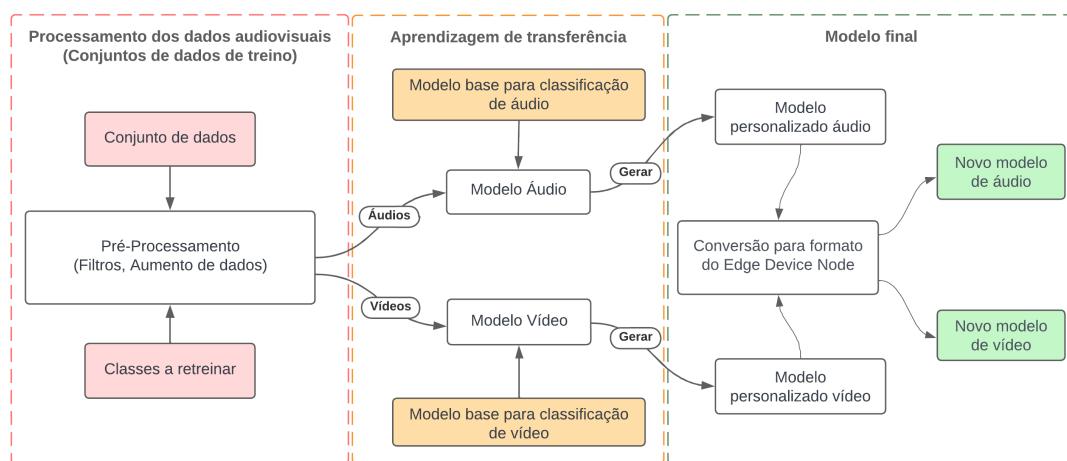


Figura 3.5: Fluxo de dados no treino dos modelos especializados na deteção de eventos audiovisuais do sistema *DistSense*

A figura 3.5 ilustra o fluxo de dados envolvido no processo de criação de modelos especializados para a deteção de eventos em ambientes residenciais. Este sistema visa adquirir, em tempo real, dados audiovisuais utilizando dispositivos equipados com sensores apropriados, tais como câmeras e microfones.

No entanto, antes da fase de inferência dos dados em tempo real, o modelo passa por um processo de treino que utiliza conjuntos de dados disponíveis online.

Esses conjuntos de dados são divididos em dois campos sensoriais: vídeo e áudio. Para o campo de vídeo, podem ser utilizados conjuntos de dados como o *Charades* (Sigurdsson et al., 2016), *UCF50* (Reddy and Shah, 2013), *UCF101* (Soomro et al., 2012) e *Toyota Smart Home* (Das et al., 2019), entre outros. Relativamente ao domínio do áudio, existem diversos conjuntos de dados conjuntos de dados como o *FSD50k* (Fonseca et al., 2022), *UrbanSound8k* (Salamon et al., 2014) e o *ESC-50* (Piczak, 2015), que englobam uma diversidade de sons característicos do ambiente doméstico e urbano.

Tais conjuntos de dados são criteriosamente selecionados como base para o treino do modelo, dada a sua submissão prévia a etapas de filtragem, limpeza e pré-processamento. Essa preparação meticulosa visa assegurar a qualidade e relevância das informações fornecidas ao modelo. Adicionalmente, a utilização de múltiplos conjuntos de dados em simultâneo proporciona exemplos e contextos diversos, conferindo ao modelo a capacidade de generalizar e reconhecer uma vasta gama de eventos e situações, sobretudo no contexto de um ambiente doméstico.

Após o treino do modelo com esses conjuntos de dados, o sistema está pronto para a fase de captura de dados em tempo real, onde durante essa fase os dispositivos capturam dados audiovisuais, que são então submetidos a um processo de pré-processamento. Esse processo pode envolver tarefas como limpeza, normalização e transformação, com o objetivo de preparar os dados para análise e processamento posterior, como demonstra na figura 3.3.

Uma vez pré-processados, os dados são analisados utilizando uma rede neuronal previamente treinada e estruturada, essa arquitetura especializada permite que o modelo capture relações de longo prazo em sequências temporais, possibilitando o reconhecimento de padrões complexos e a previsão de eventos futuros com base nas informações audiovisuais recolhidas.

Por fim, os resultados obtidos a partir da execução do modelo são armazenados localmente na *blockchain*. Essa tecnologia descentralizada garante a imutabilidade e a transparência das informações, fornecendo uma linha temporal de registo confiável dos eventos capturados. Esse armazenamento em *blockchain* é particularmente relevante para aplicações em áreas como segurança, análise de vídeo e automação residencial, onde a integridade e a rastreabilidade dos eventos são essenciais (Ozdayi et al., 2020).

3.7 Módulo de processamento e representação do conhecimento

Através da disseminação de novas tecnologias, surge uma preocupação crescente sobre quais dados sensíveis são recolhidos e como estes são utilizados (Miltgen and Peyrat-Guillard, 2014). A segurança e privacidade dos dados são desafios importantes enfrentados pelos dispositivos IoT em casas inteligentes.

De acordo com a investigação conduzida por (Psychoula et al., 2018), sobre a utilização de dispositivos inteligentes no quotidiano do utilizador, conclui-se que as pessoas idosas mostram uma maior predisposição para partilhar dados e têm menos preocupações acerca de questões de privacidade, quando comparadas com as gerações mais jovens. Os autores explicam este resultado através da tendência dos jovens em estarem mais familiarizados com a tecnologia e serem mais conscientes dos riscos que as tecnologias de IoT representam.

A natureza distribuída e a grande escala das redes IoT tornam difícil garantir a confidencialidade, integridade e autenticação dos dados. Uma abordagem promissora para resolver esses desafios é a implementação de um sistema distribuído, este tipo de sistemas oferecem várias vantagens em termos de escalabilidade, latência e privacidade (Singh et al., 2018).

Além disso, a recolha de dados local permite que os sistemas distribuídos respondam rapidamente a mudanças nas atividades, sem depender da transmissão de dados para um sistema central, dessa forma a privacidade dos utilizadores é protegida, visto que os dados não precisam ser transmitidos para um sistema central e permanecem livres de falhas.

Nesse sentido, a tecnologia de *blockchain* emerge como uma solução relevante para abordar as questões de segurança dos dados, tratando-se de uma abordagem promissora para o processamento de dados em sistemas distribuídos, onde é estabelecido um registo distribuído e imutável de transações, mantido por uma rede de nós (Ozdayi et al., 2020). Cada bloco na *blockchain* contém um registo de múltiplas transações e é encriptado e conectado ao bloco anterior por meio de criptografia, formando assim uma cadeia de blocos. A integridade dos dados é garantida, uma vez que qualquer tentativa de alteração ou falsificação seria detetada e rejeitada pelos nós da rede local.

Ao funcionar como um registo de eventos numa linha cronológica, a *blockchain* permite que todas as transações sejam rastreáveis e verificáveis, onde cada bloco contém um registo encriptado de todas as transações anteriores, formando uma sequência imutável de eventos. Essa característica torna a *blockchain* extremamente útil em diversas áreas, desde transações financeiras até o registo de ações para deteção de padrões no quotidiano do utilizador.

Uma das principais vantagens desta tecnologia é sua imutabilidade, uma vez que após os dados serem registados na *blockchain*, eles não podem ser alterados ou excluídos,

garantindo a integridade dos dados e aumentando a confiança no sistema. Com a natureza descentralizada da *blockchain* significa que não há uma única entidade a controlar os dados, o que aumenta a resiliência do sistema contra ataques e falhas (Tariq et al., 2019).

Adicionalmente oferece várias vantagens em termos de segurança, para a qual a criptografia é usada para garantir a confidencialidade dos dados e para verificar a autenticidade das transações (Tariq et al., 2019). Do mesmo modo que os mecanismos de consenso utilizados na *blockchain* garantem que todos os nós na rede concordem com o estado atual da *blockchain*, o que ajuda a prevenir fraudes e adulteração de dados.

No entanto, a *blockchain* apresenta desafios em termos de implementação como demonstrado no estudo de (Dorri et al., 2017). Alguns desses desafios são referentes à latência e baixa escalabilidade, resultante da necessidade de transmitir transações e blocos para toda a rede.

É importante enfatizar que o sistema proposto prioriza a segurança dos dados sensíveis, como imagens e áudio, capturados pelos sensores. Esses dados não são armazenados localmente ou externamente, sendo apenas processados e transformados em informações de alto nível, como eventos ocorridos num determinado período temporal, que são posteriormente armazenados em formato JSON na *blockchain*. Essa abordagem reforça a segurança, minimizando os riscos de acesso não autorizado ou fuga de informações pessoais sensíveis.

Capítulo 4

Implementação do Sistema *DistSense*

O presente capítulo aborda de forma minuciosa a implementação dos módulos apresentados no capítulo 3 do sistema *DistSense*. O objetivo principal é a deteção e classificação de padrões de atividades diárias do utilizador, por meio da captura em tempo real de imagens e áudio, mantendo a privacidade e segurança como elementos fundamentais do sistema.

Neste capítulo, serão apresentados em detalhe os procedimentos técnicos adotados para a implementação do sistema *DistSense*, onde serão abordadas as estratégias utilizadas para a captura de imagens e áudio, bem como as etapas envolvidas no processamento e análise desses dados.

No âmbito da privacidade, foram adotadas medidas rigorosas para garantir que os dados recolhidos sejam tratados de forma confidencial, respeitando os direitos do utilizador. A implementação foi realizada com base em princípios científicos estabelecidos na área de pesquisa, utilizando estudos e metodologias mencionadas no capítulo 2.

Num primeiro momento, será detalhado o processo de inicialização do sistema e a forma como a comunicação entre os nós é estabelecida de maneira descentralizada, e em seguida, será abordada em profundidade a integração da tecnologia *blockchain* no sistema, explorando o seu funcionamento e destacando os benefícios da sua utilização como um mecanismo de armazenamento de registo em ordem cronológica.

Por fim, os eventos são processados localmente com o intuito de fornecer diagnósticos personalizados, baseados em padrões comportamentais do utilizador ao longo de um período de tempo específico, determinado pelo próprio utilizador. Essa análise é disponibilizada ao utilizador através de um software de automação residencial gratuito e de código aberto, desenvolvido como um sistema de controlo central para dispositivos domésticos inteligentes, com ênfase no controlo local e na privacidade residencial designado por *Home Assistant* (HA).

4.1 Cenário de aplicação: Sistema de monitorização distribuído para o reconhecimento de atividades domésticas

A evolução tecnológica tem impulsionado avanços significativos em várias áreas, incluindo a forma como os dados são monitorizados e geridos. Um dos cenários de aplicação mais promissores é a implementação de sistemas de monitorização em espaços inteligente a longo prazo, que combinam dispositivos conectados, sensores avançados e algoritmos de análise de dados para recolher informações em tempo real.

Essa tecnologia inovadora oferece benefícios notáveis em diversas situações, incluindo a promoção da saúde e bem-estar num contexto residencial inteligente.

Nesse sentido, é exemplificado o caso da Dona Rosa, uma mulher idosa que reside numa pacata vila costeira de Portugal, na qual desfruta dos confortos de uma casa inteligente, onde sensores e dispositivos são utilizados para monitorizar as suas atividades diárias, através da recolha de dados relevantes para sua saúde e qualidade de vida. No entanto, a privacidade dos seus dados pessoais é uma preocupação primordial para Dona Rosa, o que a leva a procurar uma solução segura e confidencial para a gestão dos seus dados.

Como forma de garantir os requisitos da Dona Rosa, o sistema implementado na sua habitação adota uma abordagem distribuída, pelo que, em vez de enviar os dados para um serviço exterior, as informações são processadas localmente, dentro dos limites da sua própria residência, essa configuração permite que a Dona Rosa mantenha o controlo exclusivo sobre os seus dados pessoais, decidindo quando e com quem deseja compartilhá-los.

Esta abordagem distribuída de monitorização oferece à Dona Rosa um nível aprimorado de privacidade, para que os seus dados permaneçam em sua casa, protegidos de olhares indiscretos e acessos indesejados. Desta forma, pode desfrutar dos benefícios da tecnologia inteligente, sabendo que a confidencialidade dos seus dados está preservada e que possui total controlo sobre a forma como os mesmos estão a ser tratados.

Ao longo de uma semana de monitorização contínua, o sistema inteligente compila e analisa os dados coletados, onde é possível fornecer um diagnóstico personalizado para Dona Rosa com base nessas informações. Por exemplo, caso a análise revele uma semana com pouca atividade física, o sistema pode sugerir opções de exercícios adequados à sua condição física, incentivando-a a tornar-se mais ativa.

É importante salientar que, durante todo o processo, a privacidade da Dona Rosa é rigorosamente respeitada, sendo que os dados são utilizados exclusivamente para fins de monitorização e melhoria da sua saúde, para os quais mantém o total controlo sobre o acesso e a utilização dos seus dados pessoais.

A Dona Rosa tem a capacidade de definir as suas preferências de privacidade e pode optar por compartilhar os seus dados com profissionais de saúde, apenas em situações específicas e autorizadas.

Desta forma, a Dona Rosa beneficia da monitorização a curto e longo prazo, recebendo um diagnóstico personalizado e recomendações para melhorar a sua saúde, enquanto mantém a privacidade dos seus dados como uma prioridade absoluta. A aceitabilidade e confiança no sistema são garantidas, permitindo-lhe desfrutar dos benefícios de um ambiente inteligente, sem comprometer a sua privacidade.

4.2 Simulação

A simulação desempenha um papel fundamental no desenvolvimento e implementação dos cenários de aplicação escolhidos, proporcionando uma série de vantagens, mencionadas na secção 2.2, que justificam a sua aplicação preliminar antes da implementação num contexto real.

Ao evitar a necessidade de *hardware* dedicado e outras infraestruturas, a simulação oferece uma opção acessível e económica para o desenvolvimento e teste do sistema *Dist-Sense*, resultando numa redução significativa de custos em comparação com a implementação direta num contexto real.

A escolha do GNS3 em comparação com outras tecnologias, como o *containerlab*, pode ser justificada com base em vários critérios, incluindo a presença de uma interface gráfica amigável.

A interface gráfica disponibilizada pelo GNS3 é caracterizada pela intuitividade e facilidade de utilização, permitindo aos utilizadores criar e configurar topologias de rede de forma visual, visto que a sua interface gráfica permite simplificar a criação, configuração e gestão de dispositivos de rede virtualizados.

Adicionalmente, o GNS3 oferece um amplo suporte a dispositivos de rede virtualizados, como roteadores, *switches* e *firewalls* de diferentes fornecedores. Além disso, suporta recursos avançados, como emulação de protocolos de roteamento, inspeção de pacotes e análise de tráfego, possibilitando aos utilizadores realizar testes e simulações detalhadas em redes virtuais.

O GNS3 conta com uma comunidade ativa e uma vasta gama de recursos disponíveis. Existem fóruns, tutoriais e documentação detalhada que auxiliam os utilizadores a aproveitar ao máximo o *software*, onde a comunidade contribui com modelos e imagens prontas para uso, facilitando a configuração rápida de dispositivos de rede virtualizados.

Outra vantagem do GNS3 é sua capacidade de integração com equipamentos de rede físicos, permitindo a interação entre dispositivos físicos e virtuais, numa única topologia, o que possibilita a criação de ambientes de teste mais realistas.

A integração das tecnologias *docker* e GNS3 permite a criação de cenários de rede

complexos, nos quais é possível implementar e testar o sistema num ambiente virtual controlado. Através da sua utilização, é possível criar contentores que contêm todos os componentes e dependências necessárias para a execução do sistema distribuído, como bibliotecas, ferramentas e serviços específicos. Estes contentores podem ser implantados nos nós virtuais do GNS3, representando os diferentes elementos do sistema distribuído.

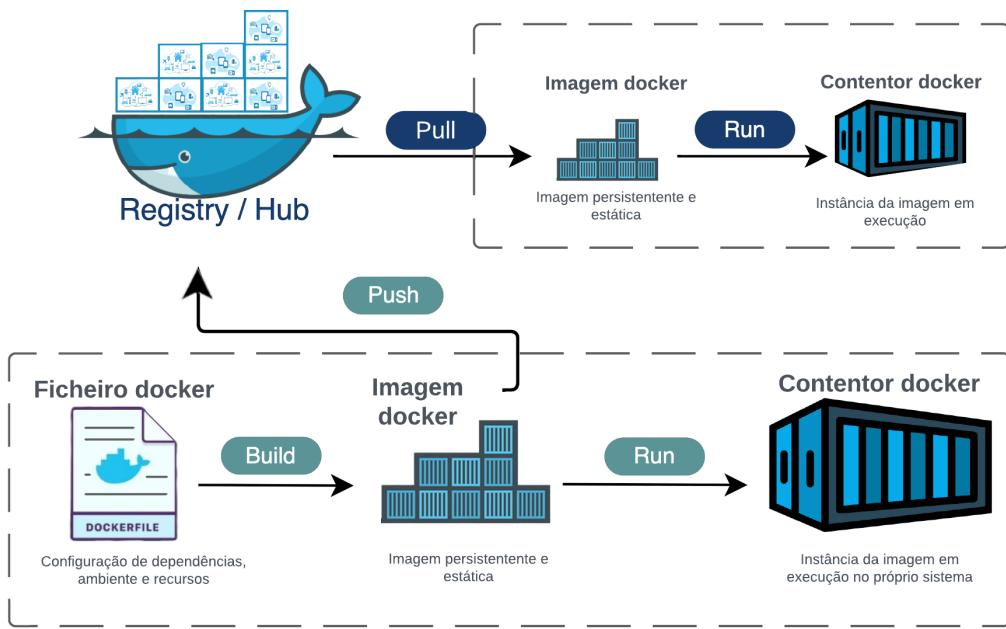


Figura 4.1: Processo de construção e registo de um contentor *docker*

Ao utilizar a tecnologia *Docker*, é possível definir um ficheiro, designado de *Dockerfile*, que especifica de forma precisa os requisitos e configurações necessários para a construção e execução dos contentores, incluindo a instalação de dependências, a configuração do ambiente, a exposição de portas e a transferência de ficheiros relevantes para a imagem *Docker*.

Após a criação do *Dockerfile*, o próximo passo é a construção da imagem *Docker*, ilustrado na figura 4.1. Este processo é realizado utilizando o comando "*docker build*" no terminal, indicando o diretório onde o *Dockerfile* está localizado. O *Docker* analisa o ficheiro construído e executa as instruções nele contidas, gerando assim a imagem *Docker*. Durante este processo, cada instrução é executada em camadas distintas, permitindo uma construção incremental e eficiente.

Com a imagem *Docker* devidamente construída, torna-se apto a executar o contentor, usando o comando "*docker run*". Esta instrução cria uma instância em execução do contentor com base na imagem especificada, além disso, é possível fornecer diversas opções durante a sua execução, tais como mapeamento de portas, montagem de volumes e definição de variáveis de ambiente. Deste modo, o contentor é inicializado e torna-se acessível através das portas que foram expostas.

Posteriormente à construção da imagem, realiza-se o envio da mesma para um registo *docker* utilizando o comando "*docker push*", sendo este um repositório centralizado que armazena as imagens *docker*, permitindo assim o acesso remoto a elas, deste modo, a imagem fica disponível para os dispositivos que necessitam de a executar.

O registo ou *Hub* desempenha um papel crucial no armazenamento e partilha de imagens *docker*. O *Docker Hub*¹ é um exemplo popular de registo, contudo existem outras opções, como o *Amazon Elastic Container Registry*² e o *Google Container Registry*³. É altamente recomendado registar a imagem *docker* no *hub* antes de a partilhar ou implementar, pois isso permite que outras pessoas a accedam e utilizem de forma simples e conveniente.

No GNS3, a integração de imagens *docker* é realizada através da utilização da GNS3 *Virtual Machine* (GNS3 VM). Num primeiro momento, é necessário efetuar o *download* e a instalação da GNS3 VM, tratando-se de uma máquina virtual pré-configurada para executar o GNS3 e suportar a integração com o *docker*.

Após configurar a GNS3 VM e inicializar o ambiente, o próximo passo envolve a realização do comando "*docker pull*", para obter a imagem desejada a partir do *docker Hub*.

Este procedimento é efetuado através do terminal da GNS3 VM, onde se executa o comando "*docker pull [nome da imagem]*" para fazer o *download* da imagem específica do *docker Hub* para a VM. É importante salientar que é fundamental ter uma conexão de rede ativa para aceder ao registo e efetuar o *download* da imagem desejada.

Após realizar o *download* da imagem *docker* do registo, é possível executar o contentor localmente utilizando o comando "*docker run*", conforme mencionado anteriormente. Desta forma, o *docker* verifica se a imagem está presente localmente e, caso não esteja, realiza automaticamente o *download* do registo antes de executar o contentor.

Uma vez que a imagem *docker* esteja disponível na GNS3 VM, é possível prosseguir com a criação de instâncias do contentor *docker* dentro do GNS3.

¹<https://hub.docker.com/>

²<https://aws.amazon.com/pt/ecr/>

³<https://cloud.google.com/artifact-registry>

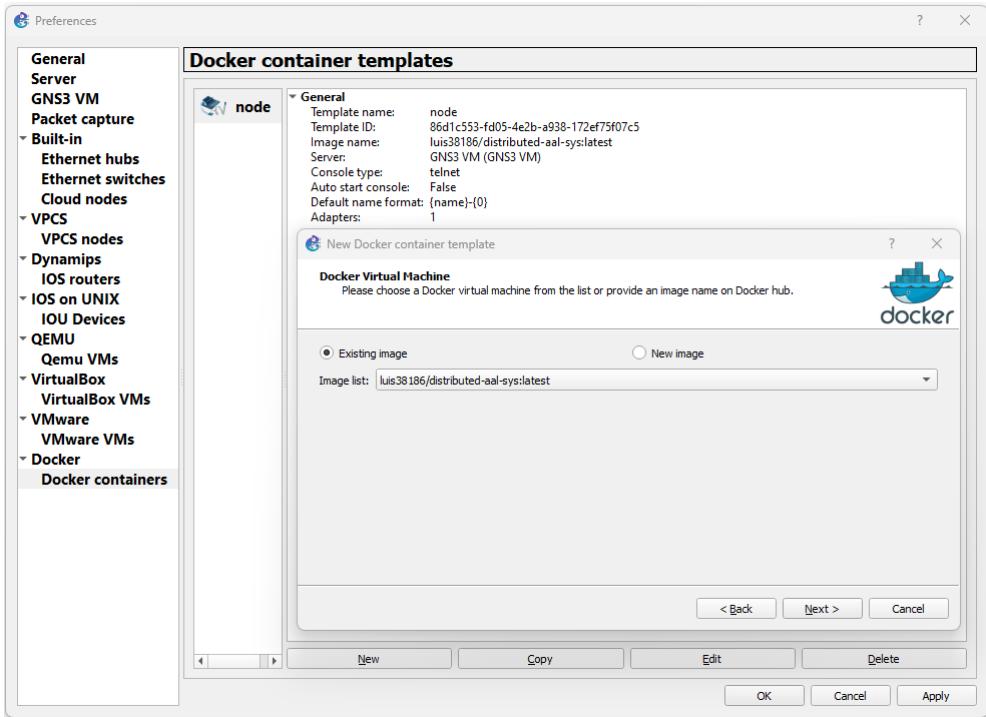


Figura 4.2: Configuração do contentor *docker* no GNS3

Para tal, ao aceder às preferências do GNS3 na secção "*Docker container templates*", como apresentado na figura 4.2, é possível adicionar modelos personalizados de contentores, onde são especificados detalhes como nome, descrição, comando de inicialização, portas expostas e parâmetros adicionais.

Posteriormente, é possível utilizar esses modelos na construção da topologia no GNS3, sendo que ao adicionar um dispositivo *docker* à topologia, é selecionado um modelo de contentor previamente definido. Durante a execução da simulação, o GNS3 criará e iniciará instâncias do contentor *docker* com base nas configurações fornecidas no modelo.

Através da integração do *docker* com o GNS3, é possível criar e interligar dispositivos virtuais baseados em contentores *docker* com outros dispositivos da topologia, como roteadores e *switches* virtuais, esta integração possibilita testar e simular cenários de rede complexos, aproveitando as funcionalidades e a flexibilidade oferecidas pelos contentores *docker*.

4.3 Inicialização do sistema *DistSense*

A inicialização de um sistema em ambientes distribuídos é uma etapa crítica que desempenha um papel fundamental na estabilização da rede, garantindo a conexão e a ativação adequada dos nós. É de extrema importância que esta inicialização ocorra de forma apropriada, uma vez que é essencial para assegurar a escalabilidade e a fiabilidade do sistema, possibilitando a adição e a ligação transparente de novos nós à rede.

Antes de prosseguir com a implementação do módulo de descoberta, torna-se imper-

ativo realizar algumas etapas de configuração inicial, nas quais cada dispositivo obtém o seu endereço IP através do protocolo *Dynamic Host Configuration Protocol* (DHCP). Este procedimento garante uma configuração correta e coerente na rede local, evitando conflitos de endereçamento e permitindo que os dispositivos comuniquem eficientemente.

A inicialização do sistema *DistSense* divide-se em duas etapas fundamentais:

1. Descoberta de dispositivos na rede: Nesta etapa, o sistema realiza uma descoberta ativa dos dispositivos presentes na rede, possibilitando a identificação e o reconhecimento de cada nó pelos demais. Para tal, é implementado um módulo de descoberta específico que rastreia e identifica os dispositivos ativos na rede local. Este processo é crucial para estabelecer um ambiente coeso e apto a integrar sem problemas novos nós à medida que são acrescentados à rede.
2. Implementação do algoritmo de eleição: Uma vez concluída a etapa de descoberta, o sistema dá início à implementação de um algoritmo de eleição entre os nós participantes. Este algoritmo tem como objetivo selecionar um nó específico para assumir uma posição ou função especial dentro da rede. O nó eleito será responsável por coordenar operações, recolher informações agregadas ou desempenhar outras tarefas importantes para o funcionamento do sistema. A escolha criteriosa deste nó líder é crucial para otimizar a eficiência do sistema distribuído e garantir um funcionamento harmonioso.

4.3.1 Serviço de descoberta

Os sistemas distribuídos são compostos por diversos dispositivos interconectados que colaboram para realizar tarefas de forma conjunta, no entanto, para que os dispositivos possam comunicar entre si, é necessário estabelecer um mecanismo eficiente de descoberta de dispositivos disponíveis na rede local.

A implementação de um módulo de descoberta adequado desempenha um papel crucial nesse contexto, possibilitando a conexão e comunicação entre os nós na rede local de forma transparente e segura.

Com o objetivo de atender a um requisito fundamental de não exigir interação direta do utilizador para o funcionamento do sistema, optou-se por utilizar a biblioteca *zeroconf* em *python*, pelo facto de oferecer uma solução que permite a descoberta e a conexão automática entre os nós numa rede local.

A biblioteca *zeroconf* utiliza o protocolo mDNS, que possibilita que os dispositivos presentes na rede local se descubram mutuamente, sem a necessidade de configurações prévias ou de um servidor centralizado, sem exigir tarefas adicionais ou intervenções ativas por parte do utilizador, contribuindo para a simplicidade e praticabilidade do sistema, garantindo que todos os nós sejam automaticamente integrados à rede sem complicações adicionais.

Através da utilização da biblioteca *zeroconf* em *python*, é possível registar e navegar pelos serviços disponíveis na rede local de forma simples e eficiente, sendo que o registo de serviços envolve a criação de um objeto de serviço, no qual são especificados o nome, o tipo e a porta do serviço a ser oferecido, para, de seguida, ser registado na rede local.

```

1 from zeroconf import ServiceInfo, Zeroconf
2
3 service_info = ServiceInfo(
4     type_= "_node._tcp.local.",
5     name=f"{NODE_NAME}.node._tcp.local.",
6     addresses=[socket.inet_aton(NODE_IP)],
7     port=NODE_PORT,
8     weight=0,
9     priority=0,
10    properties={'IP': NODE_IP, 'ID': NODE_ID, 'LOCAL': NODE_LOCAL})
11
12 zeroconf = Zeroconf()
13 zeroconf.register_service(service_info)

```

Figura 4.3: Registo de um serviço com auxílio da biblioteca *zeroconf* em *python*

Na implementação do anúncio de serviços, é utilizada a classe "*ServiceInfo*" para encapsular as informações relevantes do serviço, como o tipo, nome, endereço IP, porta, peso, prioridade e propriedades do serviço.

Ao anunciar um serviço, essas informações são divulgadas na rede, permitindo que outros dispositivos obtenham conhecimento da existência e das características desse serviço.

Este procedimento é de extrema importância em ambientes de rede doméstica, onde a configuração automática é preferida, evitando que os utilizadores tenham que configurar manualmente cada dispositivo para estabelecer comunicação com os restantes nós.

```

1 service_type = "_node._tcp.local."
2 zeroconf = Zeroconf()
3 servicios = ServiceBrowser(zeroconf, service_type, [update_service])
4
5 def add_service(zeroconf, service_type, name):
6     info = zeroconf.get_service_info(service_type, name)
7     if info:
8         ip_list = info.parsed_addresses()
9         for ip in ip_list:
10            if ip != NODE_IP:
11                connect_to_peer(ip, info.port, info.properties.get(b'ID'), info.
12 properties.get(b'LOCAL'))
13
14 def update_service(zeroconf: Zeroconf, service_type: str, name: str, state_change:
15 ServiceStateChange):
16     if state_change is ServiceStateChange.Added:
17         info = zeroconf.get_service_info(service_type, name)
18         if info:
19             add_service(zeroconf, service_type, name)

```

Figura 4.4: Descoberta de serviços com auxílio da biblioteca *zeroconf* em *python*

A descoberta de serviços assume um papel fundamental na implementação do módulo de descoberta, permitindo que outros dispositivos presentes na rede tenham a capacidade de encontrar e conectar-se ao serviço de forma adequada.

Após o registo de um serviço numa rede local, torna-se essencial que outros dispositivos presentes nessa mesma rede possam descobrir tal serviço. Para tal, utiliza-se a classe "*ServiceBrowser*", a qual possibilita a exploração dos serviços disponíveis na rede, aplicando filtros para encontrar um tipo específico de serviço.

Posto isto, ao recorrer à classe "*ServiceBrowser*", os dispositivos têm a capacidade de explorar os serviços que foram anunciados na rede local, facilitando a identificação dos serviços e a obtenção das informações relevantes associadas a cada um deles.

Através da aplicação de filtros, é possível realizar a busca apenas dos serviços que correspondem a critérios específicos, como um tipo de serviço particular ou outras características relevantes, essa capacidade é particularmente relevante em ambientes nos quais a interconexão dinâmica e automática de dispositivos é essencial, sem a necessidade de configurações manuais complexas.

Adicionalmente, o código utiliza a função "*update_service()*", que permanece à escuta de ocorrências de adição, atualização ou remoção de serviços na rede. Cada vez que um serviço é adicionado ou atualizado, o método "*update_service()*" é invocado pelo objeto *zeroconf*. No caso em que o nó atual possui o mesmo nome do tipo de serviço, o método "*add_service()*" é ativado para incluir o serviço na lista de pares conhecidos pelo nó.

Um aspecto de grande importância na implementação de um módulo de descoberta é o tratamento de falhas, tornando-se crucial que o módulo seja capaz de lidar com situações em que um serviço registado não esteja disponível ou quando um nó perde a conectividade com a rede.

4.3.2 Processo de eleição

A eleição de um líder ou coordenador desempenha um papel fundamental na garantia da eficiência, cooperação e ordem na comunicação entre os nós participantes de sistemas distribuídos. Através da implementação de um algoritmo de eleição, torna-se possível selecionar um nó específico entre os demais, conferindo-lhe a responsabilidade de liderança para coordenar as atividades e decisões no sistema.

A implementação de um algoritmo de eleição requer rigor e precisão, uma vez que qualquer imprecisão ou falha no processo pode acarretar consequências graves para a estabilidade e confiabilidade do sistema distribuído. Nesse sentido, é imperativo explorar abordagens algorítmicas eficientes e robustas que garantam uma eleição justa e coerente, mesmo em cenários adversos ou em condições variáveis na rede.

No contexto dos sistemas distribuídos, a aplicação de algoritmos de eleição tem como objetivo designar um nó específico para desempenhar funções especializadas, como a

coordenação de atividades, em determinado momento. No sistema "*DistSense*", onde múltiplos nós independentes colaboram para atingir objetivos compartilhados, a seleção de um nó responsável por tomar decisões e coordenar tarefas torna-se essencial.

O algoritmo "*Bully*" é uma das abordagens clássicas para a eleição do coordenador em sistemas distribuídos, proporcionando uma abordagem hierárquica baseada nos UUIDs dos nós, sendo que este garante que haja sempre um único coordenador para gerir as atividades dos nós vizinhos ativos.

O funcionamento do algoritmo em questão pode ser compreendido através de uma série de passos bem definidos. Quando um nó identifica que o coordenador atual está inacessível, inicia-se o processo de eleição. O nó principiante envia mensagens eleitorais para todos os outros nós com identificadores maiores, comunicando o desejo de tornar-se o novo coordenador. Os nós que recebem a mensagem eleitoral têm duas opções: reconhecer o pedido do nó principiante e retirar-se da eleição, aceitando-o como coordenador, ou iniciar uma nova eleição enviando mensagens eleitorais para os nós com identificadores maiores.

Através deste mecanismo de hierarquia de eleição, o nó com o maior UUID é eleito como o novo coordenador após receber todas as confirmações dos outros nós. O novo coordenador anuncia a sua vitória através do envio de uma mensagem para todos os outros nós, concluindo assim o processo de eleição, tal como ilustrado na figura 4.5.

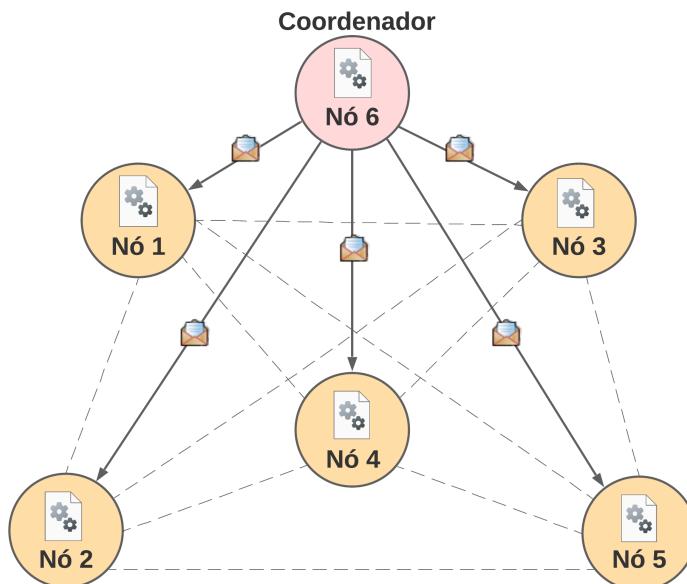


Figura 4.5: Eleição do nó coordenador

Assim, a escolha do algoritmo "*Bully*" apresenta justificativas sólidas quando comparado com outras alternativas disponíveis, sendo que uma das suas principais vantagens reside na eficiência em situações de falhas e recuperação. No caso de ocorrer uma falha no nó coordenador, os nós presentes na rede conseguem prontamente identificar esse evento

e, através do algoritmo mencionado, proceder a uma eleição do novo coordenador de forma ágil e direta, neste sentido esta rápida transição reduz significativamente o tempo em que o sistema poderia operar sem um coordenador funcional, garantindo maior continuidade nas atividades e minimizando possíveis impactos negativos (Coulouris et al., 2005).

Além disso, o algoritmo "*Bully*" destaca-se por uma abordagem robusta na eleição do líder, visto que ao utilizar mensagens diretas e explícitas para comunicar as intenções dos nós candidatos, o algoritmo minimiza ambiguidades e conflitos que poderiam surgir em metodologias mais complexas.

A simplicidade e clareza desta abordagem favorecem uma implementação concisa e de fácil compreensão, tornando-o uma opção preferencial no contexto de ambientes domésticos inteligentes, onde a confiabilidade e transparência são de extrema importância.

Outro aspecto relevante é a capacidade deste algoritmo em lidar eficientemente com sistemas distribuídos de grande escala e à medida que o número de nós aumenta, a complexidade de outros algoritmos pode crescer exponencialmente, tornando a escalabilidade um fator crítico. No entanto, o algoritmo "*Bully*" demonstra um desempenho satisfatório mesmo em ambientes com um grande número de nós, o que o torna uma solução viável e prática para o sistema *DistSense*.

Neste sentido, a implementação cuidadosa deste algoritmo de eleição, proporciona o estabelecimento de uma estrutura de coordenação sólida e eficiente, garantindo a adequada gestão de eventos entre os nós participantes.

Através de uma seleção justa e confiável do coordenador, o sistema assegura a estabilidade e a ordem necessárias para o correto funcionamento do mesmo, face aos desafios e variações nas condições da rede.

O coordenador desempenha um papel crucial na gestão de eventos da *blockchain*, assumindo a responsabilidade de validar e adicionar novos blocos à *blockchain*, essa função garante a integridade e segurança das transações, contribuindo para a manutenção da imutabilidade dos dados e para a consistência do sistema distribuído.

Adicionalmente, o coordenador assume o importante papel de estabelecer uma comunicação bidirecional com o HA, através do protocolo *Message Queuing Telemetry Transport* (MQTT), dado que a interação permite que o sistema *DistSense* envie eventos de alto nível, processados localmente, para o utilizador.

Através desta funcionalidade, o HA atua como uma interface valiosa para o utilizador, proporcionando-lhe informações relevantes e permitindo a interação com o sistema desenvolvido de forma simplificada e eficiente.

A eficiência, abordagem robusta e escalabilidade do algoritmo "*Bully*" tornam-o numa escolha perspicaz e segura para sistemas distribuídos, a sua aplicação no sistema, aliada ao papel ativo do coordenador na gestão de eventos da *blockchain* e na comunicação com o HA via MQTT, contribuindo significativamente para a otimização da comunicação e

cooperação entre os nós, consequentemente, esses esforços contribuem para o sucesso do sistema como um todo, garantindo a confiabilidade e a eficácia de suas operações.

4.4 Comunicação dos nós

No sistema *DistSense*, a comunicação entre os nós desempenha um papel crítico na viabilização de uma rede colaborativa e descentralizada, nesse paradigma cada nó atua como um servidor e um cliente simultaneamente, permitindo a troca direta de informações e recursos entre os *peers*, sem a necessidade de uma entidade central intermediária, visto que a comunicação estabelecida é o responsável para facilitar a cooperação e a partilha de dados, tornando possível a construção de redes escaláveis e resilientes.

A estrutura de comunicação num sistema P2P é constituída por diversos elementos inter-relacionados que possibilitam a interação entre os nós. Posto isto, os protocolos de comunicação, como o TCP/IP e UDP, são amplamente utilizados para facilitar a envio e receção de dados entre os *peers*, através dos mesmos, a comunicação efetua-se recorrendo à transferência de pacotes de dados entre os nós, garantindo a integridade e confiabilidade da informação.

Para estabelecer as conexões e permitir a troca de dados, as bibliotecas de comunicação, como o *socket*, desempenham um papel fundamental, para possibilitar a troca de informações entre os *peers*, garantindo uma comunicação eficiente e flexível.

No contexto da comunicação P2P, a segurança dos dados transmitidos é uma preocupação vital, especialmente em ambientes onde os dados são capturados por sensores intrusivos, como áudio e vídeo. A privacidade e integridade das informações trocadas entre os nós devem ser salvaguardadas para evitar potenciais vulnerabilidades e ataques maliciosos.

Neste sentido, é crucial aplicar protocolos seguros, como o TLS, cujo objetivo é proporcionar uma camada adicional de segurança em comunicações através de redes, esta tecnologia opera através da criptografia dos dados transmitidos entre os *peers*, protegendo-os contra a ação de terceiros mal-intencionados que possam tentar interceptar ou modificar as informações durante a comunicação.

Ao implementar o protocolo TLS no sistema *DistSense*, pode-se assegurar a confidencialidade e integridade dos dados transmitidos, através do processo de criptografia, que torna os dados ilegíveis para qualquer pessoa não autorizada a aceder à informação, contribuindo para a proteção dos dados sensíveis do utilizador, contra eventuais ataques cibernéticos.

Da mesma forma, a integridade dos dados é salvaguardada através do uso de mecanismos de verificação de integridade de mensagens durante a transmissão, significando que qualquer tentativa de modificar os dados durante a sua transmissão seria detetada, assegurando que os dados permanecem autênticos e não foram adulterados.

Desta forma, ao utilizar o protocolo TLS, o sistema *DistSense* promove uma comunicação segura e protegida, o que é fundamental em ambientes domésticos, onde a privacidade e a segurança da informação são essenciais.

Adicionalmente, ao destacar a implementação deste protocolo específico, reforça-se a preocupação na concretização do sistema *DistSense* em adotar práticas de segurança de última geração para proteger os dados dos utilizadores, de forma eficaz e confiável.

Uma abordagem comum para garantir a continuidade das conexões e a resiliência da comunicação entre os nós, é a utilização do mecanismo de reconexão baseado em mensagens de *Keep Alive*. Este mecanismo consiste em enviar periodicamente pacotes de sondagem para verificar se a conexão TCP ainda está ativa.

No caso de não existir resposta aos pacotes de sondagem dentro de um intervalo definido, a conexão é considerada inválida, permitindo que a aplicação tome ações apropriadas.

Para implementar o TCP *Keepalive* é possível utilizar a biblioteca *socket*, que fornece as funcionalidades necessárias para habilitar o mecanismo no *socket* de comunicação, através do ajuste de opções específicas do *socket*.

Ao criar o *socket* TCP cliente, é essencial habilitar o TCP *Keepalive*, definindo os parâmetros de intervalo entre os pacotes de sondagem, número de tentativas de envio sem resposta e tempo máximo de espera para resposta.

Uma vez habilitado o TCP *Keepalive* no *socket*, a aplicação pode prosseguir com a comunicação normal entre os nós. Durante períodos de inatividade, serão automaticamente enviados pacotes de sondagem, como ilustrado na figura 4.6, verificando a validade da conexão, permitindo que a aplicação detete e reaja a desconexões inesperadas, garantindo a continuidade da comunicação, mesmo em ambientes com instabilidades de rede.

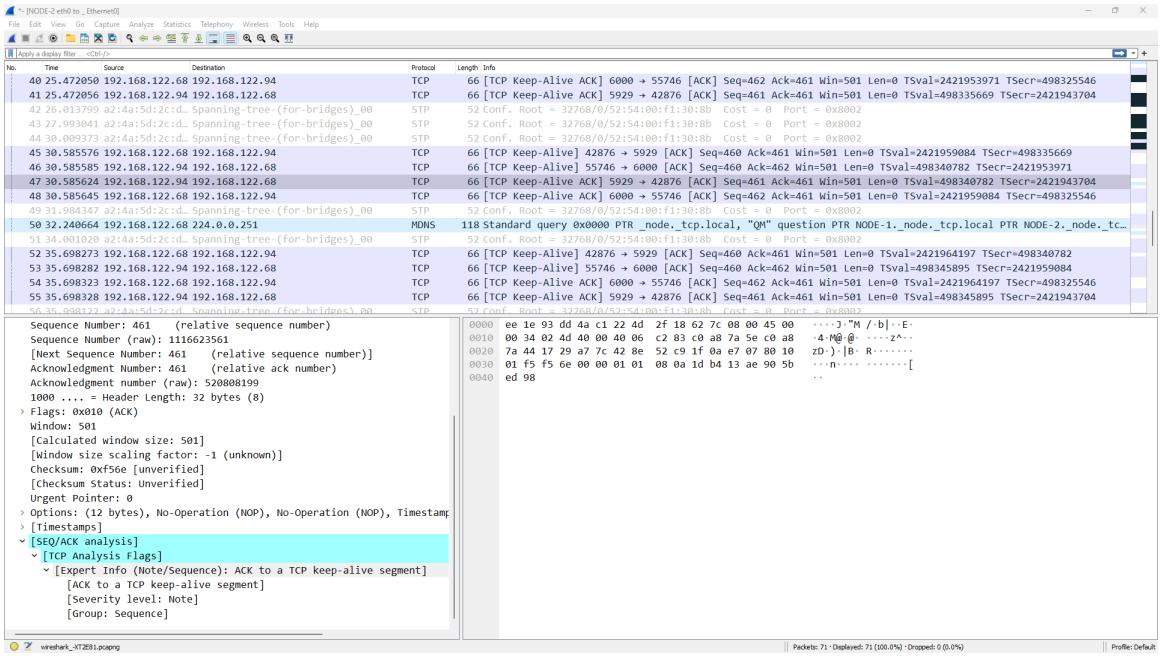


Figura 4.6: Captura de pacotes *keep alive* através da aplicação wireshark

Num ambiente inteligente distribuído, onde falhas na rede podem ocorrer, a implementação de tais mecanismos é essencial para garantir a continuidade das conexões e a resiliência da comunicação entre os nós.

Assim, ao verificar a atividade dos nós e reorganizar dinamicamente a rede, os mecanismos de reconexão asseguram uma comunicação contínua e eficiente entre os *peers*, reforçando a colaboração e a confiabilidade da arquitetura distribuída. Desse modo, a utilização desses mecanismos torna a rede P2P mais robusta e preparada para lidar com possíveis desafios e falhas na rede, garantindo o sucesso do sistema *DistSense* como uma equipa colaborativa.

4.4.1 Mensagens, estrutura e procedimento

Um sistema distribuído consiste num conjunto de processos independentes que executam em diferentes máquinas e comunicam entre si por meio de troca de mensagens, permitindo a criação de redes descentralizadas, onde os nós podem coordenar e compartilhar informações de maneira eficiente.

Assim, é essencial que haja mecanismos para lidar com as mensagens entre os dispositivos. O sistema *DistSense* é projetado para ser executado por nós que atuam como *peers*, ou seja, entidades autónomas que se comunicam entre si para realizar operações como transações e atualizações na *Blockchain*.

As mensagens trocadas entre os nós são representadas em formato JSON, que é uma notação de dados leve e amplamente suportada. Esse formato torna a serialização e desserialização das mensagens mais simples, facilitando a interoperabilidade do sistema.

Cada mensagem é estruturada como um dicionário JSON contendo campos relevantes,

como o tipo de mensagem, o conteúdo da mensagem, e outras informações necessárias para o processamento adequado, como representado na figura 4.7.

```
1  {
2      "META": {
3          "CLIENT": "0.0.1",
4          "FROM_ADDRESS": {
5              "UUID": "07b4ab99-504a-40d9-ad2a-69e4c47e21c8",
6              "IP": "192.168.122.94",
7              "PORT": 5929
8          },
9          "TO_ADDRESS": {
10             "UUID": "d8544564-ec92-421a-94c4-27ff2bd003d4",
11             "IP": "192.168.122.68",
12             "PORT": 6000
13         }
14     },
15     "TYPE": "PING",
16     "PAYLOAD": {
17         "LAST_TIME_ALIVE": 1689886629.4676633,
18         "COORDINATOR": "07b4ab99-504a-40d9-ad2a-69e4c47e21c8"
19     }
20 }
```

Figura 4.7: Mensagem padrão no formato JSON em *python*

O mecanismo para lidar com as mensagens num sistema distribuído, como o *DistSense*, é implementado por meio de um conjunto de métodos e estruturas que possibilitam a receção, processamento e envio adequado das mensagens entre os nós da rede.

Nesse sentido, o sistema mantém uma conexão de rede ativa entre os nós, permitindo a troca de mensagens, sendo que cada nó possui um socket que fica a aguardar a chegada de mensagens, após a chegada de uma nova mensagem, esta é armazenada num *buffer* de entrada, aguardando o processamento com base no tipo de mensagem para tomar as ações apropriadas. Em seguida, é enviada a mensagem de resposta ao nó remetente através do socket de conexão.

Através destes mecanismos implementados, o sistema *DistSense* é capaz de lidar eficientemente com as mensagens entre os nós da rede. A estruturação dos dados em formato JSON, a definição clara dos tipos de mensagem e a atribuição de métodos específicos para o processamento de cada tipo permitem que os nós coordenem as suas atividades, realizem transações, atualizem informações na *blockchain* e garantam a integridade e segurança das operações num ambiente distribuído. A robustez do mecanismo de tratamento de exceções contribui para a confiabilidade do sistema, garantindo que possíveis erros ou falhas sejam tratados de forma adequada e que a rede continue a funcionar de maneira resiliente.

4.5 Aprendizagem computacional

A monitorização e análise de dados em tempo real são elementos cruciais em diversos domínios, como segurança, saúde, transporte e meio ambiente. Com os avanços nas tecnologias de sensores e captura de dados, surgiram sistemas de monitorização sofisticados, capazes de capturar informações detalhadas a partir de múltiplas fontes.

Neste contexto, verifica-se que tais sistemas têm vindo progressivamente a incorporar métodos provenientes da aprendizagem computacional, destacando-se, em particular, o recurso recorrente a redes neurais artificiais, tendo sido amplamente adotadas na interpretação e extração de informações de elevado valor a partir dos dados capturados.

As redes neurais, por sua vez, apresentam-se como uma abordagem de elevado potencial no que concerne à identificação de padrões de elevada complexidade presentes em conjuntos de dados extensos, o que permite efetuar uma análise automatizada de desempenho substancial.

A aprendizagem computacional corresponde ao sub-campo da inteligência artificial que se concentra no desenvolvimento de algoritmos capazes de melhorar o seu desempenho numa tarefa específica por meio da experiência adquirida com a prática. Diferentes técnicas de aprendizagem computacional são aplicadas dependendo da natureza dos dados e da complexidade da tarefa em questão.

No sistema *DistSense*, são utilizadas técnicas de aprendizagem computacional aplicadas à monitorização de áudio e vídeo, que visam detetar atividades do quotidiano do utilizador em tempo real. Nesse sentido, são adotados dois modelos amplamente reconhecidos na comunidade científica: o modelo *YAMNet* para processamento de áudio e o modelo *Movinet* para análise de vídeo.

O *YAMNet* é um modelo de DL pré-treinado para classificação de áudio em várias categorias sonoras. Ele foi desenvolvido com base numa arquitetura de rede neuronal convolucional e já foi treinado em um extenso conjunto de dados para capturar informações sonoras relevantes.

Por sua vez, o *Movinet* é um modelo de DL especialmente concebido para tarefas de processamento de vídeo, como deteção de objetos e reconhecimento de ações. Sua arquitetura baseada em *Transformers* permite capturar contextos temporais e espaciais complexos, tornando-o adequado para análise de sequências de vídeo.

A escolha desses modelos pré-treinados deve-se à sua eficácia em tarefas específicas de processamento de áudio e vídeo, bem como à capacidade de transferência de aprendizagem. A transferência de aprendizagem é uma estratégia poderosa que nos permite aproveitar o conhecimento prévio dos modelos treinados em conjuntos de dados extensos para acelerar o treino e melhorar o desempenho em tarefas relacionadas.

A implementação dos modelos *YAMNet* e *Movinet* foi realizada utilizando a plataforma *TensorFlow* (TF), uma biblioteca de código aberto para aprendizagem computacional e

desenvolvimento de redes neurais, visto que oferece uma variedade de ferramentas e recursos para construir, treinar e avaliar modelos de ML, tornando o processo de implementação mais eficiente e acessível.

A utilização de aprendizagem computacional na monitorização de áudio e vídeo tem o potencial de fornecer informações valiosas para diversas aplicações práticas. Os modelos implementados neste sistema contribuem para a deteção e classificação de eventos sonoros e padrões visuais em tempo real, aprimorando a capacidade de tomada de decisão e fornecendo uma visão mais abrangente dos dados monitorizados.

4.5.1 Pré-processamento do Conjunto de Dados Audiovisuais

Antes de iniciar o treino dos modelos, é necessário realizar o pré-processamento dos conjuntos de dados de entrada, tanto de áudio como de vídeo. No que se refere ao conjunto de dados de áudio, foram escolhidos dois conjuntos principais: o *FSD50k* (Fonseca et al., 2022) e o *ESC-50* (Piczak, 2015). No que concerne ao conjunto de dados de vídeo, optou-se por utilizar o *Toyota Smart Home* (Das et al., 2019) e pelo *Charades* (Sigurdsson et al., 2016). Estes conjuntos foram selecionados para a tarefa específica de identificação e classificação de sons e vídeos relevantes no contexto doméstico.

Um passo crucial nesse procedimento é a seleção das classes mais pertinentes para cada conjunto de dados, sendo necessária uma análise minuciosa de cada áudio e vídeo individualmente, para avaliar a sua qualidade e relevância em relação ao objetivo em questão. Durante esse processo, é importante eliminar os sons e/ou vídeos que contenham várias classes misturadas, uma vez que a presença de várias classes num único ficheiro pode dificultar a correta classificação.

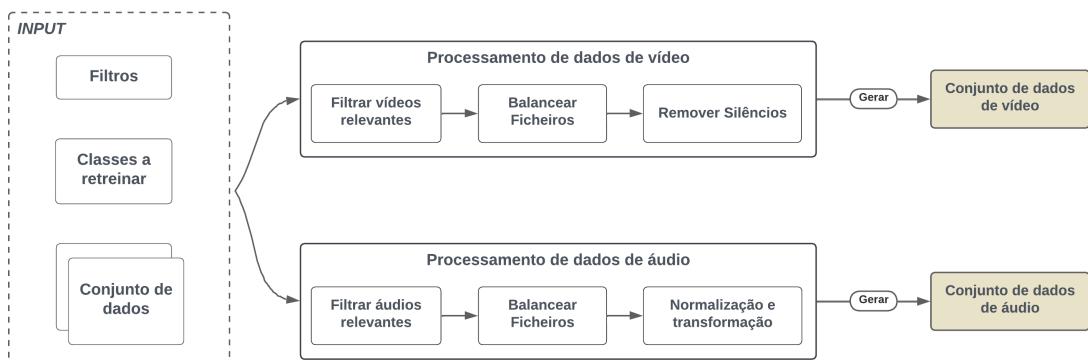


Figura 4.8: Fluxo do processo de pré-processamento do conjunto de dados audiovisuais

Apesar de ser um trabalho minucioso, especialmente num conjunto de dados com milhares de sons e vídeos, esta etapa é essencial para garantir a qualidade dos modelos. A seleção cuidadosa das classes contribui para a eficiência de cada modelo e a precisão das classificações realizadas.

Outro procedimento realizado durante o processamento do conjunto de dados é o bal-

anceamento das classes, sendo que é uma etapa importante para evitar que o modelo se torne tendencioso em relação a classes com mais exemplos e, assim, garantir que a aprendizagem seja equilibrada em todas as classes.

Além disso, no domínio do áudio, é realizada a remoção dos segmentos de silêncio presentes nos sons, o que consiste em eliminar partes dos ficheiros que possuam níveis baixos de decibéis, geralmente localizados no início, meio ou fim de cada som. A remoção destes segmentos de silêncio é relevante para eliminar ruídos indesejados e garantir que apenas informações relevantes sejam consideradas durante o treino do modelo.

Estas etapas de pré-processamento do conjunto de dados são fundamentais para aprimorar a qualidade dos dados utilizados no treino de cada modelo. A eliminação de segmentos indesejados e a construção de um conjunto de dados consistente e limpo garantem que o modelo seja alimentado com informações precisas e relevantes, permitindo que ele aprenda padrões significativos para a tarefa em questão.

A seguir a esta fase de pré-processamento, o conjunto de dados tratado está pronto para ser utilizado no treino do modelo. Através da utilização do modelo *YAMNet* para processamento de áudio, e do modelo *Movinet* para análise de vídeo, é possível obter resultados de alta qualidade na identificação e classificação de sons e imagens, contribuindo para aplicações práticas em sistemas de monitorização de casas inteligentes e diversos outros domínios.

4.5.2 Treino dos modelos

A estratégia de transferência de aprendizagem tem revelado ser altamente eficaz na otimização do desempenho de modelos de redes neurais, como é o caso dos modelos adotados, *MoViNet* e *YAMNet*, para tarefas específicas.

No âmbito da monitorização de áudio, depara-se com desafios consideráveis na extração de informações relevantes a partir de sinais acústicos, neste sentido, a criação de modelos especializados voltados para a análise de áudio assume uma importância fundamental.

Assim, preparar adequadamente os dados de áudio, previamente pré-processados, para o treino destes modelos é uma etapa crucial, englobando a redefinição de variáveis críticas, como o número de canais e a taxa de amostragem, sendo que estes fatores desempenham um papel determinante na asseguração da qualidade do modelo resultante.

A extração de *embeddings* provenientes do modelo original desempenha um papel central na construção de uma nova abordagem simplificada. Os *embeddings*, enquanto representações vetoriais contínuas de variáveis discretas, facultam a aprendizagem de características pertinentes ao contexto do sistema de monitorização. A conversão das variáveis categóricas em espaços vetoriais de dimensão reduzida permite não só a redução da complexidade dos dados, como também a representação significativa das diversas classes

selecionadas.

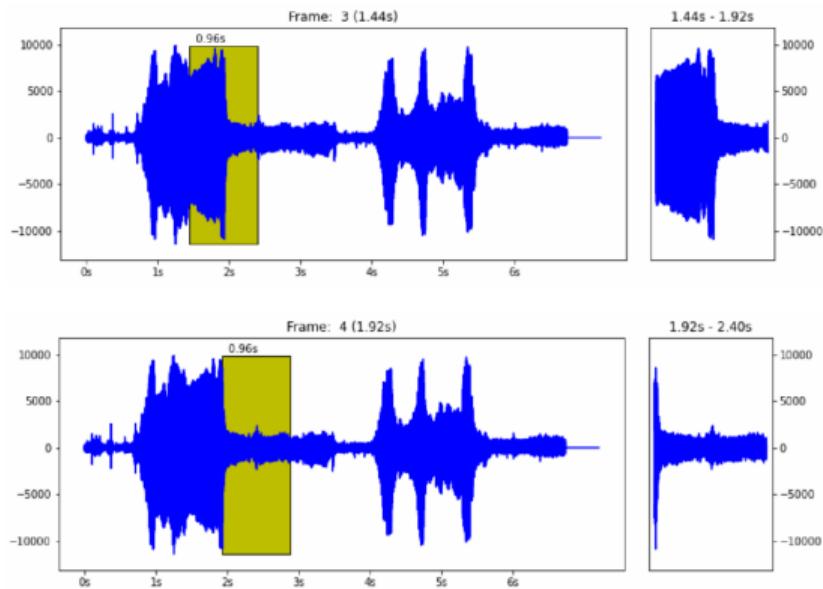


Figura 4.9: Janela deslizante utilizada na implementação do modelo *YAMNet*

No caso particular do modelo *YAMNet*, o mesmo faz uso de *features* com uma dimensão de 1024, as quais caracterizam cada *frame* de áudio, correspondente a um intervalo de tempo de 0.96 segundos. No processo de treino, é imperativo garantir que a sequência de áudio de entrada possua uma frequência de amostragem de 16 kHz e esteja configurada para um único canal. Para explorar aspectos temporais, o áudio é particionado em janelas de 0.96 segundos, com um passo de 0.48 segundos entre cada uma, originando, deste modo, uma abordagem de janela deslizante demonstrada na figura 4.9.

No âmbito da análise visual, com o intuito de solidificar o procedimento de treino do modelo *MoViNet*, são incorporadas abordagens fundamentadas em transferência de aprendizagem, após submeter conjuntos de dados audiovisuais a um processo de pré-processamento meticuloso, onde se procede a uma minuciosa curadoria e equilíbrio dos dados audiovisuais. Este processo visa adequadamente preparar os conjuntos de dados para o propósito de classificação de atividades em tempo real.

O modelo *MoViNet* evidencia-se como um robusto classificador de vídeo, encontrando aplicações em cenários de *streaming* de vídeo e inferência em tempo real, no âmbito do reconhecimento de ações. Contudo, os modelos assentes em *frames* 2D, apesar da sua eficiência na análise de vídeos completos ou *frames* individuais em regime de *streaming*, revelam limitações na consideração do contexto temporal, culminando numa precisão limitada e resultados inconsistentes entre *frames* sucessivos.

Uma abordagem mais sofisticada engloba a utilização de redes convolucionais 3D, as quais incorporam contexto temporal bidirecional, contribuindo assim para um incremento da precisão e consistência temporal. No entanto, estas redes podem exigir mais recursos

computacionais e não são ideais para o processamento de fluxos contínuos de dados, dado o requisito de considerar informações futuras.

Posto isto, o traço arquitetónico distintivo do modelo *MoViNet* reside na adoção de convoluções 3D causais ao longo do eixo temporal, assemelhando-se à operação "*layers.Conv1D*" com o parâmetro "*padding='causal'*". Este design conjuga as vantagens das abordagens anteriores, possibilitando uma análise eficaz em *streaming*.

Adicionalmente, é crucial compreender que o modelo *MoViNet* requere um tensor de vídeo 5D RGB como entrada, apresentando uma estrutura específica: [BATCH_SIZE, NUM_FRAMES, HEIGHT_PIXELS, WIDTH_PIXELS, 3]. Esta configuração permite que o modelo analise cada *frame* dentro de um contexto mais alargado, garantindo, desse modo, a apreensão mais precisa das relações temporais e espaciais presentes na captura de vídeo.

A convolução causal garante que a saída no tempo "*x*" seja calculada apenas com base em entradas até o tempo "*y*". Esta eficiência em regime de *streaming* pode ser ilustrada por meio de uma analogia com as RNN, nas quais o estado é transmitido ao longo do tempo. No contexto do *MoViNet*, esse estado é designado por "*Stream Buffer*" e a sua transmissão evita a necessidade de recálculo do campo receptivo completo.

Neste sentido, o treino do modelo *MoViNet* através da transferência de aprendizagem baseia-se num conjunto de dados previamente tratado e equilibrado, onde implica a utilização de pesos pré-treinados do *MoViNet* num conjunto de dados mais abrangente, seguida de uma afinação com o conjunto de dados preparado especificamente para a tarefa de reconhecimento de ações em vídeos.

O processo de treino exige a adaptação dos parâmetros do modelo de forma a refletir as nuances e particularidades do novo conjunto de dados. As camadas superiores do modelo são ajustadas para se adequarem à tarefa em questão, ao passo que as camadas mais profundas, responsáveis por capturar características genéricas, permanecem inalteradas.

A eficácia do treino é ampliada pela qualidade do conjunto de dados adequadamente tratado e balanceado. A seleção de exemplos representativos de cada classe de ação, em quantidade suficiente para evitar desequilíbrios, é fundamental para o sucesso da transferência de aprendizagem. Este enfoque permite que o modelo generalize com precisão para dados não previamente observados.

Posto isto, o processo de treino do modelo *MoViNet* mediante a transferência de aprendizagem, apoiado por um conjunto de dados devidamente tratado e equilibrado, constitui uma abordagem sólida para aprimorar a capacidade de classificação de ações em vídeos. A combinação da arquitetura *MoViNet*, com as convoluções 3D causais, sendo que a estratégia de transferência de aprendizagem confere ao modelo a capacidade de realizar inferências eficientes e precisas em tempo real, maximizando, simultaneamente, a exploração do contexto temporal.

4.5.3 Conversão do modelo

À medida que os sistemas de monitorização inteligente ganham terreno em ambientes domésticos, surge uma necessidade premente: otimizar o processamento em dispositivos de recursos limitados, como o *Jetson Nano*. Neste contexto, a otimização de modelos de ML complexos, como o *Yamnet* para áudio e o *Movinet* para vídeo, assume um papel crucial, exigindo abordagens inovadoras.

Assim, após o treino dos modelos num conjunto de dados audiovisuais cuidadosamente pré-processados, a etapa seguinte envolve a sua conversão para um formato mais leve, como o formato *.tflite*.

A transformação de modelos complexos para o formato *.tflite* oferece uma série de vantagens significativas. Em primeiro lugar, a compactação resultante otimiza consideravelmente o espaço que esses modelos ocupam, desempenhando um papel vital em ambientes onde o espaço de armazenamento é limitado, permitindo a inclusão de vários modelos no mesmo dispositivo, sem comprometer a qualidade ou a eficiência.

Adicionalmente, apresenta melhorias notáveis em termos de agilidade operacional, sendo que a latência da inferência é reduzida através da eliminação de etapas de análise e descompressão, resultando em interações mais ágeis e na deteção de eventos em tempo real. Este aspeto é crucial na implementação do sistema *DistSense*, onde a resposta imediata é fundamental para o funcionamento eficaz do sistema.

A integração de meta dados durante o processo de conversão é outra vantagem substancial. Esses meta dados não apenas fornecem descrições comprehensíveis das funcionalidades dos modelos, mas também apresentam informações estruturadas que simplificam a configuração do fluxo de trabalho. Num ambiente distribuído, essa característica fortalece ainda mais a eficiência operacional, uma vez que os modelos otimizados podem ser ajustados com precisão às necessidades específicas de processamento.

No entanto, a conversão de modelos complexos para o formato *.tflite* não está isenta de desafios. Um dos desafios mais comuns é a incompatibilidade de operações, onde determinadas operações existentes nos modelos originais não são suportadas pelo formato *.tflite*, como é o caso do modelo de áudio nas operações: *ComplexAbs* e *RFFT2D*. A solução implica a criação de modelos equivalentes que fazem uso de operações compatíveis. Esta tarefa requer uma análise meticolosa e a recriação cuidadosa das partes do modelo afetadas.

Além disso, esta etapa pode levar a uma ligeira perda na qualidade da inferência, sendo atenuada pela aplicação de técnicas de otimização, como a quantização, que permite representar números de ponto flutuante com menos *bits*, mantendo um nível aceitável de precisão, como descrito no estudo de (Verma et al., 2021). Este aspeto é particularmente relevante quando se lida com os recursos limitados dos dispositivos *Jetson Nano*.

Neste contexto, a implementação da conversão de modelos complexos para o for-

mato *.tflite* revela-se altamente vantajosa e estratégica para o sistema *DistSense*, visto que alinha-se perfeitamente com a necessidade premente de otimizar o processamento em dispositivos com recursos limitados, como os dispositivos *Jetson Nano* utilizados.

4.6 Processamento e representação do conhecimento

Com a evolução tecnológica e a crescente necessidade de sistemas de monitorização inteligentes, o processamento e a representação do conhecimento desempenham um papel vital na criação de sistemas eficazes e seguros. Após o treino cuidadoso dos modelos de análise audiovisual, surgem etapas essenciais de captação e processamento de dados, como ilustrado na figura 3.3.

Numa fase inicial, a captação dos dados através de câmaras e microfones estrategicamente posicionados, frequentemente apresentados sob a forma de imagens, sequências de áudio e dados temporais, são submetidos a um processo minucioso de processamento, onde são normalizados e convertidos em formatos que se alinham com os requisitos específicos dos modelos previamente treinados.

```
1 def process_image_inference(image_np):
2
3     image_np = cv2.resize(image_np, dsize=(172, 172))
4     image_np = image_np.astype(np.float32) / 255.0
5
6     mean = np.array([0.485, 0.456, 0.406])
7     std = np.array([0.229, 0.224, 0.225])
8
9     image_np = (image_np - mean) / std
10    image_np_expanded = np.expand_dims(image_np, axis=0)
11
12    return image_np_expanded
```

Figura 4.10: Pré-processamento dos dados para inferência do modelo *MoViNet* em *python*

Neste contexto, são implementados mecanismos que desempenham funções essenciais para o melhor desempenho do sistema, como o redimensionamento da imagem capturada para uma resolução padronizada, a normalização das intensidades de pixels para um intervalo entre 0 e 1 e a aplicação da padronização *z-score* para alinhar as características das imagens ao modelo pré-treinado, otimizando as entradas para a inferência de dados visuais em tempo real, tal como é demonstrado na figura 4.10.

Com a utilização de sensores audiovisuais integrados no *Jetson Nano*, para inferir atividades do quotidiano do utilizador em tempo real, surge a necessidade de adotar tecnologias que desempenham um papel crucial na viabilização, processamento e representação do conhecimento do sistema. Duas das tecnologias em destaque e pertinentes para o sistema *DistSense* são a *blockchain* e o *Home Assistant*.

A utilização da tecnologia *blockchain* oferece uma abordagem segura e transparente para registar e validar as informações recolhidas pelo sistema, como descrito na secção 3.7. Através da sua imutabilidade e descentralização permitem o armazenamento confiável de todas as atividades e dados inferidos, garantindo a integridade e autenticidade das informações. Adicionalmente, permite criar regtos à prova de adulteração, contribuindo para a confiança das atividades inferidas pelo sistema proposto, proporcionando um nível adicional de segurança.

Neste sentido, a arquitetura da *blockchain* é especialmente pertinente para o sistema *DistSense*, uma vez que oferece uma maneira confiável e segura de armazenar e manter um registo cronológico das atividades capturadas.

Esta tecnologia, segura e descentralizada, é alicerçada na estruturação dos blocos, que são os alicerces da *chain*.

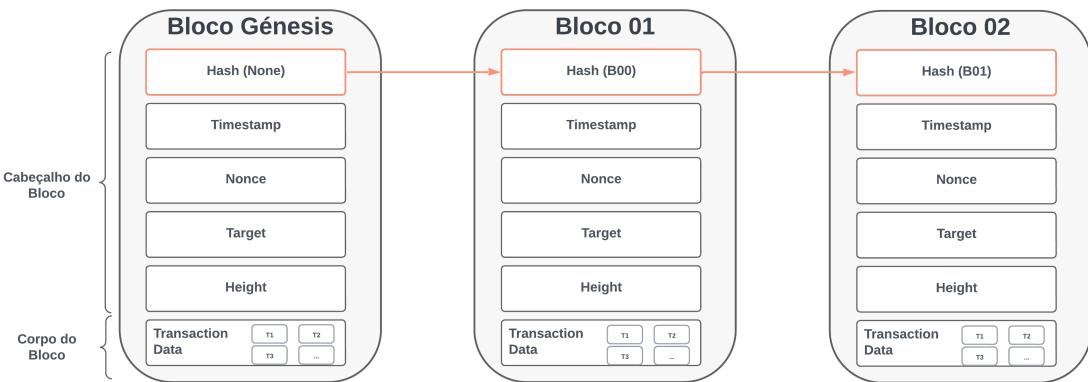


Figura 4.11: Estrutura de cada bloco da *blockchain* do sistema *DistSense*

Cada bloco na *blockchain* segue uma estrutura cuidadosamente definida, sendo composto por elementos essenciais como *timestamp*, *hash*, *target*, *nonce*, *height* e transações, conforme representado na figura 4.11.

O *timestamp*, regista a data e hora exatas da criação do bloco, é de primordial importância para ordenar cronologicamente os eventos na *blockchain*, conferindo à solução a capacidade de distinguir sequências de ocorrências, revelando-se crucial em cenários de monitorização nos quais a ordem das ações é imperativa para a análise dos dados.

O *hash*, por sua vez, consiste numa assinatura digital única derivada do conteúdo do bloco, utilizando um algoritmo criptográfico de função designada por *hash*. Esta representação compacta e segura do bloco serve como um mecanismo de autenticação e identificação infalível, uma vez que ao ligar cada bloco ao seu antecessor através do *hash*, a *blockchain* constrói uma *chain* inquebrável e imutável de transações e eventos.

Em seguida, o parâmetro *target* assume um papel essencial ao determinar a complexidade do processo de *mining*. Ao estabelecer o grau de dificuldade necessário para validar os blocos, o *target* regula a frequência da incorporação de novos blocos na *blockchain*. No âmbito da monitorização, esta parametrização revela-se crucial para garantir uma vig-

ilância eficiente e em tempo real, evitando excessos ou lacunas temporais indesejadas.

Posteriormente, o *nonce*, o componente "*number only used once*", desempenha um papel crucial no algoritmo *Proof of Work (PoW)*. Este valor, subjacente a complexos cálculos criptográficos, é ajustado iterativamente pelos *miners* até que um *hash* válido seja alcançado. A energia computacional exigida por este processo constitui uma camada de segurança adicional, essencial para a confirmação das transações na rede.

Adicionalmente, a altura do bloco indica a sua posição na *chain*, definindo a sequência ordenada dos eventos registados. O conceito de altura fornece uma perspetiva hierárquica na visualização dos dados, promovendo a compreensão da evolução temporal dos eventos monitorizados.

As transações são módulos essenciais da *blockchain*, onde registam a transferência de informação, bens ou contratos. Na perspetiva do sistema *DistSense*, estas transações assumem o papel de dados monitorizados e registados de forma imutável, conferindo uma sequência cronológica de acontecimentos.

Neste sentido, a operação de inferência do sistema *DistSense* consiste na inserção dos dados previamente processados nos modelos, conferindo-lhes a habilidade de identificar padrões, eventos e anomalias. No momento em que a precisão da inferência ultrapassa um limiar crítico de 90%, o sistema entra em ação para registar o evento identificado e todos os pormenores a ele associados, garantindo que apenas sejam armazenadas informações de alto nível.

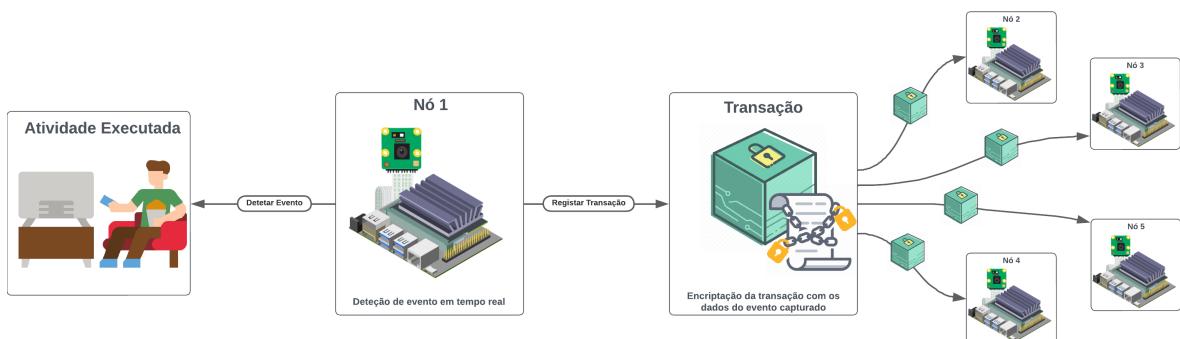


Figura 4.12: Fluxo de registo de transação na *blockchain*

O evento identificado, em conjunto com detalhes como data, hora, localização e o UUID do dispositivo específico que detetou o evento, é integrado numa transação, sendo posteriormente, transmitida para todos os nós presentes na rede doméstica, ilustrado na figura 4.12. No entanto, antes do envio da transação, é de extrema importância garantir a integridade e a confiabilidade dos dados para que os outros dispositivos presentes na rede validem a transação.

Desta forma, a transação passa por um processo de encriptação, onde é submetida a algoritmos criptográficos robustos que assinam a mensagem após esta ser submetida a um processo de *hash*. Posteriormente, a transação encriptada é enviada para os outros

dispositivos na rede local.

Este procedimento não apenas acrescenta uma camada adicional de segurança, mas também assegura que os dados se mantenham íntegros e confidenciais durante todo o processo de transmissão. A assinatura criptográfica resultante atua como garantia da preservação da integridade da transação à medida que é partilhada entre os diversos nós da rede.

Quando a transação chega a cada nó do sistema *DistSense* na rede doméstica, esta segue um processo rigoroso de validação, onde certifica-se de que a transação é legítima e cumpre com as regras pré-estabelecidas pelo sistema, caso a validação seja bem-sucedida, a transação é adicionada a uma lista de transações pendentes.

Quando a transação chega a cada nó da rede, esta passa por um processo rigoroso de validação para garantir a sua legitimidade e conformidade com as regras do sistema *DistSense*, se a validação for bem-sucedida, a transação é adicionada a uma lista de transações pendentes.

À medida que o número de transações pendentes aumenta e atinge o limite estabelecido, o nó coordenador é notificado, desencadeando o início do processo de *mining*. Este processo envolve a resolução de desafios computacionais complexos, culminando na criação de um novo bloco que agrupa as transações previamente validadas, sendo, posteriormente, incorporado à *blockchain*, assegurando de forma segura e imutável a integridade das transações.

Num contexto onde a automação residencial é cada vez mais um paradigma essencial, a ligação entre dispositivos e sistemas inteligentes assume um papel primordial na construção de ambientes residenciais inteligentes e reativos. Perante isto, é necessário a implementação de uma comunicação bidirecional fluida entre a plataforma de automação residencial, o HA, e o sistema *DistSense*, apoiando-se no protocolo MQTT, concebido especialmente para ambientes IoT e de domótica.

O sistema *DistSense* demonstra uma notável sofisticação ao permitir a aquisição, análise e armazenamento de informações originárias de fontes de áudio e vídeo. Através da aplicação de algoritmos de ML, este sistema é capaz de identificar eventos relevantes, como sons anómalos ou atividades do quotidiano do utilizador. Após o processamento dos dados, esses eventos são registados de forma imutável na *blockchain*, garantindo a autenticidade e inviolabilidade dos dados.

Um dos aspectos importantes desta implementação reside na comunicação através do protocolo MQTT. Neste sistema o nó coordenador, envia os eventos inferidos encapsulados em mensagens JSON para tópicos MQTT específicos. O HA, por sua vez, desempenha o papel de subscriptor, aguardando ansiosamente mensagens nos tópicos pertinentes, facilitando a transferência das informações do sistema para o utilizador, estabelecendo um ambiente de partilha de dados em tempo real.

Um dos elementos essenciais desta implementação reside na utilização do protocolo MQTT para a comunicação entre o sistema *DistSense* e o exterior. Neste sistema, o

nó coordenador assume a responsabilidade de enviar os eventos inferidos, os quais são encapsulados em mensagens JSON, para tópicos MQTT específicos.

Em contrapartida, a função do HA é desempenhar o papel de subscritor, permanecendo à escuta pela chegada de mensagens nos tópicos utilizados, simplificando a transferência das informações do sistema para o utilizador, estabelecendo assim um ambiente de partilha de dados em tempo real de forma eficiente e contínua.

A fase inicial requer a configuração de um *broker* MQTT, que atua como intermediário para as trocas de mensagens. Este *broker*, acessível tanto ao sistema *DistSense* como ao HA, desempenha um papel central na facilitação da comunicação bidirecional. A atribuição criteriosa dos tópicos, organizados de maneira hierárquica, destaca-se como um aspecto fundamental para assegurar a fluidez de todo o processo.

Com os tópicos estabelecidos, o sistema distribuído implementado cumpre o seu papel ao identificar e processar localmente eventos relevantes ao contexto do utilizador, que posteriormente são encapsulados em mensagens JSON, contendo detalhes críticos de alto nível para a representação do conhecimento, como o tipo de evento, o registo temporal e uma breve descrição. Uma vez publicadas nos tópicos MQTT correspondentes, as mensagens iniciam a sua transmissão em direção ao HA.

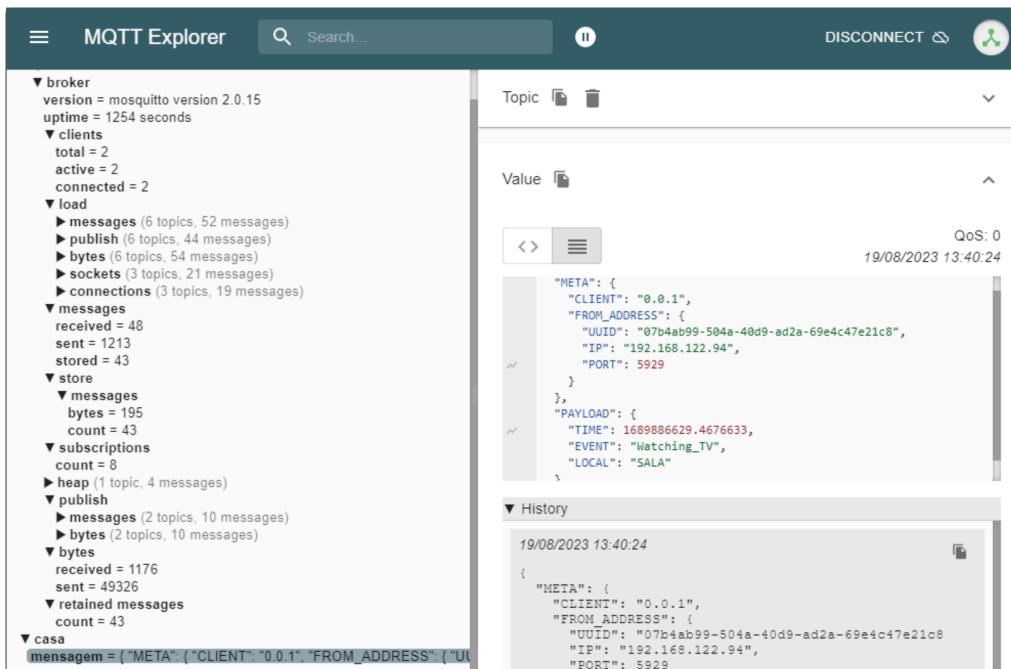


Figura 4.13: Mensagem enviada pelo nó coordenador via MQTT para o HA

Após configurado para subscrever os tópicos específicos, o HA permanece à espera de novas mensagens provenientes do sistema *DistSense* via MQTT, quando recebe uma mensagem, é acionado um gatilho que desencadeia uma automação predefinida, que neste caso se limita a demonstrar visualmente o evento detetado, embora possa ser personalizado conforme as necessidades do utilizador.

A elegância desta integração manifesta-se na automação e visualização dos eventos. O HA, ao receber uma mensagem MQTT, interpreta-a e ativa uma automação correspondente. Por exemplo, os detalhes do evento são extraídos da mensagem JSON e apresentados de forma visualmente apelativa na interface do utilizador, onde um elemento simples, como uma *label*, pode conter informações essenciais, como o tipo de evento, a descrição e o *timestamp* da inferência dos dados.

Para aprimorar a experiência, regras personalizadas podem ser implementadas em *python*. Estas regras facultam uma manipulação mais refinada das mensagens MQTT, permitindo a realização de ações específicas ou interações com outros componentes do HA.

Para enriquecer a experiência, são implementadas regras personalizadas em *Python*, como demonstrado na figura 4.14, permitindo uma manipulação mais refinada das mensagens MQTT, possibilitando a realização de ações específicas ou interações com outros componentes do HA.

```
1 automation:
2   - id: exibir_evento_mqtt
3     alias: Exibir Evento MQTT
4     trigger:
5       platform: mqtt
6       topic: topico/do/seu/json
7     action:
8       service: persistent_notification.create
9       data_template:
10         title: Novo Evento MQTT Recebido
11         message: "Evento: {{ trigger.payload_json.PAYLOAD.EVENT }}
```

Figura 4.14: Regra de automação no *Home Assistant* em *python*

A interação dinâmica entre o sistema desenvolvido e o HA, mediada pelo protocolo MQTT, redefine a automação residencial, uma vez que ao identificar e capturar eventos significativos, potenciados por algoritmos de ML, esta implementação cria um ambiente verdadeiramente seguro e personalizado. Através desta fusão de tecnologias, a automação residencial eleva-se para um novo patamar de inteligência e capacidade de resposta, proporcionando uma experiência envolvente e individualizada aos utilizadores que utilizam o sistema *DistSense*.

Capítulo 5

Testes e Avaliação

A análise exaustiva e sistemática de qualquer sistema distribuído é de extrema importância para garantir não apenas sua eficiência, mas também sua adaptabilidade e confiabilidade num ambiente operacional real. Nesta secção, é dado enfoque na abordagem meticulosa de testes e avaliações aplicados ao inovador sistema *DistSense*. Este sistema, que se propõe a capturar dados audiovisuais através de uma rede distribuída de câmaras e dispositivos de vídeo, apresenta um paradigma único de colaboração entre dispositivos distribuídos, culminando numa abordagem altamente precisa para deteção de eventos e ações específicas num ambiente doméstico.

O sistema *DistSense* se distingue por sua natureza distribuída, com múltiplos dispositivos inteligentes interconectados, cada um contribuindo com informações para uma visão holística das atividades num espaço residencial. No coração desse sistema encontra-se a sinergia entre dispositivos, permitindo uma avaliação mais precisa e eficaz das ocorrências. Esta secção se concentra na exploração detalhada dos testes e avaliações realizados para garantir que cada módulo implementado funcione harmoniosamente em conjunto, assegurando a escalabilidade e a eficiência global do sistema.

A avaliação do *DistSense* foi dividida em dois cenários distintos, com o intuito de abranger tanto o ambiente controlado quanto as complexidades do mundo real. Inicialmente, foram empregados testes em ambiente virtual, que permitiram uma análise minuciosa dos componentes individuais do sistema. A latência, um fator crítico em sistemas distribuídos, foi rigorosamente medida, assegurando que a colaboração entre os dispositivos ocorresse de maneira ágil e sem perda significativa de tempo. Além disso, a resiliência do sistema foi testada em diferentes cenários de falha, demonstrando sua capacidade de se recuperar de situações adversas.

Entretanto, para simular condições do mundo real e verificar a aplicabilidade prática do sistema, testes em ambiente real foram conduzidos. Esses testes expuseram o sistema a uma gama diversificada de desafios, incluindo variações de iluminação, ruídos ambientais e situações imprevistas. A precisão da deteção de eventos foi avaliada sob essas circunstâncias mais dinâmicas, e a escalabilidade do sistema foi examinada à medida que

mais dispositivos foram integrados à rede.

Os testes em ambos os cenários não apenas validaram a robustez e confiabilidade do sistema *DistSense*, mas também forneceram informações valiosas sobre sua capacidade de adaptação a situações do mundo real. As métricas coletadas durante os testes, incluindo latência, resiliência, escalabilidade e precisão, fornecem uma base sólida para a análise quantitativa e qualitativa do desempenho do sistema.

Ao longo deste capítulo, exploramos em detalhes os procedimentos de teste, metodologias adotadas e os resultados obtidos em ambas as abordagens. Através desses testes criteriosos, o sistema *DistSense* demonstrou sua viabilidade e excelência no contexto de uma rede distribuída destinada a capturar dados audiovisuais. Os dados derivados desses testes não apenas validam a eficácia do sistema, mas também fornecem um alicerce confiável para a implementação futura e aprimoramentos contínuos do mesmo.

5.1 Testes em ambiente virtualizado

Abordar o gns3 e/ou container lab, os testes que foram feitos (Performance, etc), deteção em video/audio de eventos e demonstração em blockchain com prints por exemplo e abordagem.

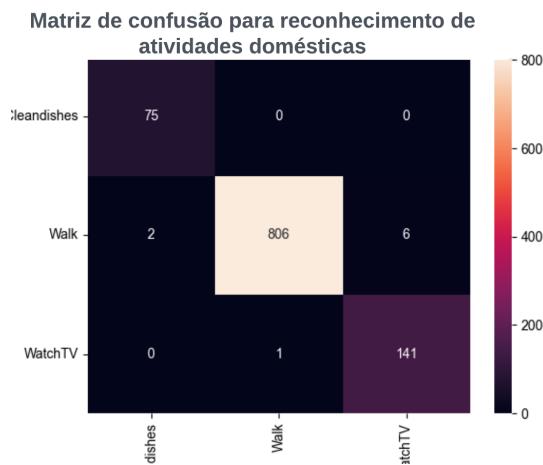


Figura 5.1: Matriz de confusão dos resultados obtidos para as classes treinadas

5.2 Testes em ambiente real

Abordar o equipamento, configuração, os testes que foram feitos (Performance, etc), deteção em video/audio de eventos

5.3 Discussão

Comparação dos resultados com trabalhos anteriores da mesma área de estudo e os resultados obtidos.

Capítulo 6

Conclusão

6.1 Resultados

6.2 Trabalho futuro

Concluir todo o trabalho que foi feito e quais as conclusões que se obteve, o que pode ser abordado em trabalho futuro .

Bibliografia

- Batyrzhan K Akhmetzhanov, Omar Aslan Gazizuly, Zhanserik Nurlan, and Nurkhat Zhakiyev. Integration of a video surveillance system into a smart home using the home assistant platform. In *2022 International Conference on Smart Information Systems and Technologies (SIST)*, pages 1–5. IEEE, 2022. [36](#)
- Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015a. doi: 10.1109/COMST.2015.2444095. [37](#)
- Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 17(4):2347–2376, 2015b. [vii, 37](#)
- Charles Anderson. Docker [software engineering]. *IEEE Software*, 32(3):102–c3, 2015. doi: 10.1109/MS.2015.62. [11](#)
- Hung Cao, Monica Wachowicz, Chiara Renso, and Emanuele Carlini. Analytics everywhere: generating insights from the internet of things. *Ieee Access*, 7:71749–71769, 2019. [vii, 17](#)
- Stuart Cheshire and Marc Krochmal. DNS-Based Service Discovery. RFC 6763, February 2013. URL <https://www.rfc-editor.org/info/rfc6763>. [36](#)
- Marie Clausen, Rolf Apel, Marc Dorchain, Matthias Postina, and Mathias Uslar. Use case methodology: a progress report. *Energy Informatics*, 1:273–283, 10 2018. ISSN 25208942. doi: 10.1186/s42162-018-0036-0. [4](#)
- Intersoft Consulting. General data protection regulation - gdpr. <https://gdpr-info.eu/>, 2020. Accessed: 2022-12-05. [3](#)
- George F Coulouris, Jean Dollimore, and Tim Kindberg. *Distributed systems: concepts and design*. Pearson education, 2005. [53](#)

Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. [40](#), [59](#)

Ali Dorri, Salil S. Kanhere, Raja Jurdak, and Praveen Gauravaram. Blockchain for iot security and privacy: The case study of a smart home. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 618–623, 2017. doi: 10.1109/PERCOMW.2017.7917634. [35](#), [42](#)

Simone Facchini, Giacomo Giorgi, Andrea Saracino, and Gianluca Dini. Multi-level distributed intrusion detection system for an iot based smart home environment. In *ICISSP*, pages 705–712, 2020. [24](#), [27](#)

Sven Fleck and Wolfgang Straßer. Privacy sensitive surveillance for assisted living—a smart camera approach. In *Handbook of Ambient Intelligence and Smart Environments*, pages 985–1014. Springer, 2010. [23](#), [27](#)

Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra. Fsd50k: An open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2022. doi: 10.1109/TASLP.2021.3133208. [40](#), [59](#)

Patricia Franco, Jose Manuel Martinez, Young Chon Kim, and Mohamed A. Ahmed. Iot based approach for load monitoring and activity recognition in smart homes. *IEEE Access*, 9, 2021. ISSN 21693536. doi: 10.1109/ACCESS.2021.3067029. [22](#), [27](#)

Quazi Ehsanul Kabir Mamun, Salahuddin Mohammad Masum, and Mohammad Abdur Rahim Mustafa. Modified bully algorithm for electing coordinator in distributed systems. *WSEAS Transactions on Computers*, 3(4):948–953, 2004. [34](#)

Caroline Lancelot Miltgen and Dominique Peyrat-Guillard. Cultural and generational influences on privacy concerns: a qualitative study in seven european countries. *European journal of information systems*, 23(2):103–125, 2014. [41](#)

Ehsan Adeli Mosabbeb, Kaamran Raahemifar, and Mahmood Fathy. Multi-view human activity recognition in distributed camera sensor networks. *Sensors*, 13(7):8750–8770, 2013. [25](#), [27](#)

Joan Navarro, Ester Vidaña-Vila, Rosa Ma Alsina-Pagès, and Marcos Hervás. Real-time distributed architecture for remote acoustic elderly monitoring in residential-scale ambient assisted living scenarios. *Sensors*, 18(8):2492, 2018. [26](#), [27](#)

Jason C Neumann. *The book of GNS3: build virtual network labs using Cisco, Juniper, and more*. No Starch Press, 2015. [10](#), [11](#)

Official Website. Getting started with gns3. <https://docs.gns3.com/docs/>. URL <https://docs.gns3.com/img/getting-started/what-is-gns3/1.jpg>. [vii](#), [11](#)

OMS. Oms: Falta de atividade física pode causar doenças em 500 milhões de pessoas até 2030. In *OMS: Falta de atividade física pode causar doenças em 500 milhões de pessoas até 2030*, 2022. Accessed: 2022-12-03. [1](#)

Mustafa Safa Ozdayi, Murat Kantarcioglu, and Bradley Malin. Leveraging blockchain for immutable logging and querying across multiple sites. *BMC Medical Genomics*, 13(7):1–7, 2020. [40](#), [41](#)

Bo Pang, Erik Nijkamp, and Ying Nian Wu. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2):227–248, 2020. [39](#)

Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015. [40](#), [59](#)

Ismini Psychoula, Deepika Singh, Liming Chen, Feng Chen, Andreas Holzinger, and Huansheng Ning. Users' privacy concerns in iot based applications. In *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBD-Com/IOP/SCI)*, pages 1887–1894, 2018. doi: 10.1109/SmartWorld.2018.00317. [41](#)

Pandia Rajan Jeyaraj and Edward Rajan Samuel Nadar. Smart-monitor: patient monitoring system for iot-based healthcare system using deep learning. *IETE Journal of Research*, 68(2):1435–1442, 2022. [20](#), [27](#)

Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine vision and applications*, 24(5):971–981, 2013. [40](#)

IDMS Rupasinghe and MWP Maduranga. Towards ambient assisted living (aal): Design of an iotbased elderly activity monitoring system. *International Journal of Engineering and Manufacturing (IJEM)*, 12(2):1–10, 2022. [vii](#), [21](#), [22](#), [27](#)

Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044, 2014. [40](#)

Ashutosh Satapathy, Jenila Livingston, et al. A comprehensive survey on ssl/tls and their vulnerabilities. *International Journal of Computer Applications*, 153(5):31–38, 2016.

[38](#)

Gunnar A. Sigurdsson, Gü̈l Varol, Xiaolong Wang, Ivan Laptev, Ali Farhadi, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ArXiv e-prints*, 2016. URL <http://arxiv.org/abs/1604.01753>. [40](#), [59](#)

Deepika Singh, Ismini Psychoula, Johannes Kropf, Sten Hanke, and Andreas Holzinger. Users’ perceptions and attitudes towards smart home technologies. In *Smart Homes and Health Telematics, Designing a Better Future: Urban Assisted Living: 16th International Conference, ICOST 2018, Singapore, Singapore, July 10-12, 2018, Proceedings 16*, pages 203–214. Springer, 2018. [41](#)

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. [40](#)

Usman Tariq, Atef Ibrahim, Tariq Ahmad, Yassine Bouteraa, and Ahmed Elmogy. Blockchain in internet-of-things: a necessity framework for security, reliability, transparency, immutability and liability. *IET Communications*, 13(19):3187–3192, 2019. [42](#)

Jordan Tewell, Dympna O’Sullivan, Neil Maiden, James Lockerbie, and Simone Stumpf. Monitoring meaningful activities using small low-cost devices in a smart home. *Personal and Ubiquitous Computing*, 23:339–357, 4 2019. ISSN 16174909. doi: [10.1007/s00779-019-01223-2](https://doi.org/10.1007/s00779-019-01223-2). [19](#), [27](#)

Steve Tockey. *How to Engineer Software: A Model-Based Approach*. John Wiley & Sons, 2019. [31](#)

Gaurav Verma, Yashi Gupta, Abid M. Malik, and Barbara Chapman. Performance evaluation of deep learning compilers for edge inference. In *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 858–865, 2021. doi: [10.1109/IPDPSW52791.2021.00128](https://doi.org/10.1109/IPDPSW52791.2021.00128). [63](#)

George Xylomenos and George C Polyzos. Tcp and udp performance over a wireless lan. In *IEEE INFOCOM’99. Conference on Computer Communications. Proceedings. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. The Future is Now (Cat. No. 99CH36320)*, volume 2, pages 439–446. IEEE, 1999. [38](#)