

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Plataforma IoT para Equipamentos Industriais

João Henrique de Oliveira Peixoto

Mestrado em Engenharia Eletrotécnica e de Computadores

Orientador: Prof. Doutor Paulo Portugal

28 de julho de 2022

Este trabalho foi realizado com o apoio da instituição de acolhimento "Azevedos Indústria" e supervisionado na instituição pelo Eng. Bruno Pinho.

Colaboraram com a equipa de orientação na realização deste trabalho as seguintes pessoas: Prof. Américo Azevedo, Eng. Carlos Batista, Eng. Daniel Fernandes, Eng. Marco Abreu, Eng. Tatiana Santos e Bruno Matos.

Resumo

Este documento apresenta a dissertação desenvolvida no âmbito do Mestrado em Engenharia Eletrotécnica e Computadores, da Faculdade de Engenharia da Universidade do Porto (FEUP). O tema da dissertação é o desenvolvimento de plataforma IoT para equipamentos industriais.

A empresa proponente é a Azevedos Indústria e produz diversos equipamentos industriais para o ramo da cortiça. No entanto, esses mesmos equipamentos não comunicam com sistemas capazes de centralizar informação, nem realizar análises preditivas. Neste contexto, necessita-se um equipamento que, quando interligado com o equipamento industrial, consiga recolher diversos dados e realizar análises preditivas. Deve também apresentar ao utilizador dados relevantes e ter um grande foco na vertente de manutenção. Não obstante, deve conseguir ler códigos QR com a ordem e produção associada. Os dados processados devem ser enviados para um sistema *cloud*. Esse sistema deve recolher, armazenar e expor informação ao utilizador, apresentando um resumo da sua planta fabril.

Para isso, realizaram-se duas abordagens: A procura de soluções comerciais capazes de satisfazer os requisitos e a escolha de componentes necessários para o desenvolvimento de uma solução de raiz. Tendo em conta que as soluções comerciais encontradas não satisfaziam os requisitos, optou-se pelo desenvolvimento de uma solução própria. Esta operação permite à empresa ter um maior controlo sobre a tecnologia utilizada, entre outras vantagens.

Dividiu-se a arquitetura do sistema em duas vertentes: hardware e software. A nível de hardware utilizou-se um Raspberry Pi, um multisensor (que mede a temperatura, humidade e analisa as vibrações), um leitor de RFID, uma consola HMI, uma câmara e uma carta de entradas e saídas. Esta carta permite conectar sensores (analógicos e digitais) e atuadores de gama industrial. Ao nível de software é necessário considerar o processamento realizado a nível local e o processamento a nível da *cloud*. O processamento local, desenvolvido no Node-RED, baseia-se em 3 módulos: recolha de entradas, ativação de saídas e partilha de dados por OPC-UA e para a *cloud*; verificação de aspetos de manutenção e set de regras. Este último pretende permitir ao utilizador desenvolver os seus próprios alarmes, definir o estado das saídas, entre outras operações. A plataforma *Thingsboard* apresenta diversa informação ao utilizador, bem como uma representação da sua planta fabril.

Além dos requisitos definidos nesta dissertação, construiu-se um encapsulamento próprio para o protótipo desenvolvido. Também foi integrado um serviço VPN P2P que permite prestar assistência remota ao dispositivo IoT.

Após os testes ao sistema desenvolvido verificou-se que cumpria 90% dos requisitos elaborados. Uma grande parte dos requisitos não cumpridos deve-se à escassez de componentes elétricos. O Thingsboard continua a ser uma opção com bastante potencial quando comparado com as restantes, mas é necessária a substituição da subscrição para uma versão mais elaborada, com análise tendencial já integrada.

Palavras-chave: Automação industrial, Planta fabril, OPC-UA, VPN, Internet of Things (IoT), Raspberry Pi, *cloud*, Node-RED

Abstract

This document presents the dissertation developed for the Master in Electrical and Computer Engineering at FEUP. The goal of this dissertation is the development of IoT platform for industrial equipment.

The hosting company is Azevedos Indústria that produces several industrial equipment for the cork sector. However, these devices do not communicate with systems capable of centralizing information or performing predictive analyses. In this context, the goal is to develop equipment that, when interconnected with industrial equipment, can collect different data and perform predictive analysis. It should also present relevant data to the user and have a strong focus on maintenance. Also, it must be able to read QR codes with the order and associated production. The processed data must be sent to a cloud system. This cloud system must collect, store and present information to the user, with a summary of the shop floor.

For this, two approaches were analysed: The search for commercial solutions capable of satisfying the requirements and the choice of necessary components for the development of a proprietary solution. The commercial solutions found did not satisfy the requirements, so, the objective was to develop a proprietary architecture. This operation allows the company to have greater control over the technology used, among other advantages.

The system architecture was divided into two aspects: hardware and software. At the hardware level, a Raspberry Pi was used, a multisensor (which measures temperature, humidity and vibrations), an RFID reader, a HMI console, a camera and an input output card. This card allows connecting industrial range sensors (analog and digital) and actuators. At the software level, it is necessary to consider the processing performed locally and the processing at the cloud level. The local processing, developed in Node-RED, is based on 3 modules: input collection, output activation and data sharing via OPC-UA and to the cloud, verification of maintenance aspects and set of rules. The latter one allows the user to develop her own alarms, define the status of outputs, among other operations.

The *Thingsboard* platform presents a variety of information to the user, as well as a representation of your shop floor.

In addition to the requirements defined in this dissertation, a specific encapsulation was built for the prototype developed. A P2P VPN service was also integrated that allows remote assistance to the IoT device.

After testing the developed system, it was found that it met 90% of the requirements. The major part of the unfulfilled requirements is due to the shortage of electrical components. *Thingsboard* remains an option with a lot of potential when compared to the others, but it is necessary to replace the subscription to a more elaborate version featuring "trend analysis".

Keywords: Industrial automation, shop floor, OPC-UA, VPN, Internet of Things (IoT), Raspberry Pi, *cloud*, Node-RED

Agradecimentos

Começo por agradecer ao Prof. Paulo Portugal, Prof. Américo Azevedo e Eng. Bruno Pinho por todo o empenho, dedicação e paciência que tiveram durante todo o decorrer da dissertação. Ao Eng. Carlos Baptista, o meu agradecimento pela ajuda no desenvolvimento do encapsulamento do protótipo. Também o meu apreço a todos os elementos do gabinete de engenharia da empresa pela ajuda, ideias e amizade. Foram fantásticos! Não posso deixar de incluir o Sr. Américo Azevedo, o Sr. Fernando Azevedo, o Tiago Azevedo e Bruno Matos.

O culminar desta aventura só foi possível graças ao apoio de 3 irmãos que a faculdade me deu: André Telha, Francisco Rodrigues e Rodrigo Afonso. Para eles, um grande abraço! Deixo também uma palavra de carinho e agradecimento ao Francisco, Débora, Marcelo e Tiago por estarem sempre presentes com o seu espírito alegre contagiante!

À minha família, que se mostrou sempre presente, mesmo naqueles momentos em que tudo desmoronou. Sem eles, esta jornada não seria possível! Sou-vos eternamente grato por tudo!

À Fatinha, minha namorada, um grande obrigado pela paciência e apoio que deu ao longo dos últimos anos.

Por último, mas não menos importantes, resta agradecer a todos aqueles que contribuíram de forma direta ou indireta para a minha formação profissional e pessoal.

Bem haja,

João Peixoto

"Pensar grande, começar pequeno, mas já"

Prof. Francisco Jardim

Conteúdo

1	Introdução	1
1.1	Contexto e Motivação	2
1.2	Objetivos	2
1.3	Estrutura da Dissertação	3
2	Definição do Problema	5
2.1	Empresa proponente	5
2.2	Equipamentos Industriais	5
2.3	Análise de requisitos	8
2.4	Análise crítica	11
3	Estado de Arte	13
3.1	Manutenção	13
3.2	Estudo de Mercado	15
3.2.1	Soluções similares	16
3.2.2	Soluções comerciais	16
3.3	Estrutura típica e conceitos base	18
3.4	Protocolos de comunicação	19
3.4.1	Backgroud	19
3.4.2	Comunicação com sensores e atuadores	20
3.4.3	Comunicação com a <i>cloud</i>	21
3.5	Análise de componentes	21
3.5.1	Sensores	22
3.5.2	Interfaces Homem-máquina	28
3.5.3	Processamento Local	29
3.5.4	Ambiente de desenvolvimento	29
3.5.5	Processamento na <i>cloud</i>	31
3.5.6	Serviço VPN	32
3.6	Análise Crítica	33
4	Arquitetura proposta	35
4.1	Estrutura do sistema	35
4.2	Escolha de componentes	36
4.2.1	Hardware	36
4.3	Software	38
4.4	Projeto de Software	39

5	Implementação	43
5.1	Hardware	44
5.1.1	Ligações Interiores	45
5.1.2	Ligações Exteriores	47
5.1.3	Encapsulamento	48
5.2	Software	50
5.2.1	Processamento Local	50
5.2.2	Processamento na <i>cloud</i>	63
5.3	Análise Crítica	68
6	Validação da solução proposta	69
6.1	Ambiente de testes	69
6.2	Validação de requisitos	71
7	Conclusões e Trabalho Futuro	75
7.1	Satisfação de objetivos	76
7.2	Trabalho Futuro	76
A	Imagens	77
	Referências	81

Lista de Figuras

2.1	Máquina de Capsular	6
2.2	Balança	7
2.3	Máquina de Tratamento	7
2.4	Esquema base do sistema	8
2.5	Diagrama conceptual do sistema	12
3.1	Estudo de mercado	15
3.2	Exemplo de utilização das soluções da EWON (retirado de [1])	17
3.3	Arquitetura padrão dos sistemas de IoT (retirado de [2])	19
3.4	Modelo OSI	20
3.5	Panóplia de sensores a analisar	22
3.6	Multisensores	24
3.7	Medidores de consumo energético	25
3.8	Câmaras	26
3.9	RFID e Detecção de vibrações	27
3.10	Cartas de entradas e saídas	28
3.11	Sumário do Capítulo 3	34
4.1	Estrutura do sistema	35
4.2	Arquitetura do software	40
4.3	Arquitetura Funcional do Sistema	41
5.1	Organização do capítulo	43
5.2	Sistema implementado	44
5.3	Esquema das ligações Interiores do dispositivo IoT	45
5.4	Diagrama elétrico das ligações Interiores	46
5.5	Pi-EzConnect (retirado de [3])	47
5.6	Caixa produzida em impressora 3D	48
5.7	Pinout do dispositivo IoT	49
5.8	Ligações físicas no interior da caixa	50
5.9	Páginas da consola HMI	53
5.10	Esquema do fluxo de entradas e saídas	54
5.11	Parte de fluxo desenvolvido no Node-RED - IO's AutomationHAT	55
5.12	Comunicação com a consola HMI	56
5.13	Utilização do RFID	57
5.14	Visão global do fluxo de Manutenção	59
5.15	Tarefas do fluxo de Manutenção	60
5.16	Exemplo de um Set de Regras	61
5.17	Dashboard do HusarNET	62

5.18	Acesso ao dispositivo IoT recorrendo à VPN	62
5.19	Dashboard Geral do Thingsboard - Parte 1	64
5.20	Dashboard Geral do Thingsboard - Parte 2	65
5.21	Dashboards desenvolvidos no Thingsboard	66
5.22	Gráficos produzido pelo script em Python	67
6.1	Validação do Sistema	70
6.2	Montagem com o medidor de consumo energético	70
A.1	ST BLE Sensor	77
A.2	Fluxo "Inputs&Outputs"desenvolvido no Node-RED	78
A.3	Variáveis disponíveis por OPC-UA	79
A.4	Fluxo "Maintenance"desenvolvido no Node-RED	80

Lista de Tabelas

2.1	Requisitos do Sistema	10
5.1	Parâmetros de configuração do protocolo FINS	57
6.1	Validação de Requisitos - Sistema como um todo	71
6.2	Validação de Requisitos - Dispositivo IoT	72
6.3	Validação de Requisitos - Sistema cloud	72

Abreviaturas e Símbolos

AI	Inteligência Artificial
API	Application Programming Interface
FFT	Fast Fourier Transform
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HMI	Human Machine Interface
IIoT	Industrial Internet of Things
IoT	Internet Of Things
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KPI	Key Performance Indicators
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
OEE	Overall Equipment Effectiveness
OPC-UA	OPC Unified Architecture
OSI	Open Systems Interconnection
OTA	On-The-Air
P2P	Peer-to-peer
PIN	Personal Identification Number
PLC	Programmable Logic Controller
RFID	Radio Frequency Identification
RTC	Real Time Clock
TTL	transistor-transistor logic
VOC	Volatile Organic Compounds
VPN	Virtual Private Network
Wi-Fi	Wireless Fidelity
XML	Extensible Markup Language

Capítulo 1

Introdução

O mundo tende, cada vez mais, a ter acesso à informação na palma da mão. Os *smartphones*, *smartwatches*, *smart houses* são conceitos já existentes e utilizados no nosso quotidiano. Por outro lado, verifica-se também esta tendência na indústria. A quarta revolução industrial traz consigo conceitos inovadores, desde *Big Data*, análises avançadas, *cloud computing* e o próprio IoT (Internet of Things)[4].

Esta revolução industrial integra a troca de dados entre equipamentos. Estes dados podem estar relacionados com a manufatura, uso de materiais, ciclo de vida, cadeia de abastecimento, entre muitos outros aspetos. O grande objetivo é a existência de *smart factories*, onde é possível analisar todos os históricos e estado atual de todas as peças utilizadas[5].

O IoT consiste em interligar sistemas eletrónicos e estabelecer comunicações entre eles. O termo *things* tanto se refere a pessoas, como máquinas (sensores e atuadores), bem como informação (roupas, comida, medicina, livros, etc)[6]. Historicamente, este conceito aparece em 1999/2000 no MIT quando tentava-se conectar as etiquetas RFID (*Radio Frequency Identification*) à Internet. O objetivo era poder comunicar com estes equipamentos e, a partir de um computador, fazer a sua gestão. Nessa época existiam muitas restrições relacionadas com as comunicações sem fios, nomeadamente ao nível de utilização e custo. Uma evolução necessária para uma melhor implementação deste conceito foi a utilização do protocolo de comunicação IPv6, que permitiu albergar uma quantidade muito maior de dispositivos conectados face ao seu antecessor (IPv4). A partir desse ponto, e com o surgimento de dispositivos *low-cost*, verificou-se um rápido crescimento deste conceito.

O IIoT (Industrial Internet of Things) nasce num ambiente industrial como uma subcategoria de IoT, mais específica, tendo como objetivo trazer inteligência a objetos já existentes no chão de fábrica.

Considerando uma linha de produção industrial, com vários equipamentos e tapetes de transporte, o IIoT poderia permitir ou ajudar a:

- Fazer o seguimento (*tracking*) do bens e materiais;
- Coordenar as operações de forma inteligente;

- Monitorizar, em tempo real, os equipamentos;
- Introduzir melhorias no processo de gestão;

1.1 Contexto e Motivação

Mesmo com a existência destes conceitos, verifica-se que ainda muitos dos equipamentos relacionados com a indústria, nomeadamente a corticeira (foco do trabalho realizado nesta dissertação), não dispõem de funcionalidades que permitam a troca de dados entre si. As soluções apresentadas pela empresa não contemplam a integração com um sistema capaz de centralizar a informação proveniente desses mesmos equipamentos.

Além disso, verificam-se que os equipamentos industriais:

- Sinalizam as avarias, mas não são capazes de as prever;
- Produzem dados estatísticos, mas os mesmos só estão presentes na sua consola;
- Não monitorizam o consumo energético do próprio equipamento;
- Não apresentam painéis próprios para manutenção;

Os fabricantes de autómatos disponibilizam plataformas próprias, assim como PLC's (*Programmable Logic Controller*) preparados para lidar com este cenário. No entanto, este facto obrigaria a trocar os PLC's nos equipamentos industriais já existentes, assim como a adaptação de todo o *software* já desenvolvido para os mesmos.

1.2 Objetivos

Pretende-se, então, desenvolver um pequeno dispositivo IoT que, quando conectado com os equipamentos industriais produzidos pela empresa proponente, consiga recolher dados dos mesmos. Através de sensores extras deve monitorizar outros fatores-chave, tais como temperatura, humidade, consumo energético, gama de vibrações do equipamento e ordem de produção associada. Com estes dados deve calcular indicadores de desempenho e fazer uma análise preditiva da manutenção do equipamento industrial. Este dispositivo deve apresentar informação relevante ao utilizador localmente, bem como controlar as intervenções de manutenção praticadas pelo cliente e fabricante. Realizando um pré-processamento da informação, deve também conseguir comunicar com um sistema cloud.

Esse sistema cloud, por sua vez, deve suportar vários clientes e vários tipos de equipamentos industriais. Visa recolher, armazenar e processar a informação proveniente do dispositivo IoT. O utilizador deve ter acesso a uma visão generalizada do seu chão de fábrica, bem como aspetos gerais e particulares dos seus equipamentos industriais. Por último, este sistema deve permitir a exportação de dados para aplicações desenvolvidas por terceiros.

Salienta-se que a arquitetura deve ter em conta o conceito de "add-on", ou seja, deve ser integrado com o equipamento industrial já existente. Além disso, devem ser considerados aspetos relacionados com a fiabilidade e dependência de serviços de terceiros.

1.3 Estrutura da Dissertação

Para além da introdução, esta dissertação é composta por mais 6 capítulos. No capítulo 2 é definido o problema, apresentada a empresa que propôs o trabalho e alguns equipamentos industriais produzidos pela mesma. Além disso, é também apresentada a análise de requisitos. No capítulo 3 discute-se o estado da arte segundo duas abordagens: a análise de equipamentos e serviços já existentes ou a construção de uma solução de raiz. Na segunda abordagem são também apresentados conceitos teóricos de relevo para a construção da solução. No capítulo 4 é apresentada a arquitetura proposta, bem como a sua ligação com os requisitos previamente analisados. No capítulo 5 são abordados os aspetos de implementação ao nível de software e de hardware. No capítulo 6 é apresentada a validação da solução, bem como a conformidade com os requisitos. Por último, no capítulo 7, encontram-se as conclusões, assim como os trabalhos futuros.

Capítulo 2

Definição do Problema

Neste capítulo é apresentado o problema em questão. Para isso, é inicialmente apresentada a empresa proponente, bem como alguns equipamentos industriais produzidos pela mesma. Serão abordados alguns detalhes desses mesmos equipamentos pois, de seguida, será realizada a análise de requisitos para a solução do problema que se pretende apresentar.

2.1 Empresa proponente

A empresa proponente do projeto é a "Azevedos Indústria" com sede em Lourosa, Santa Maria da Feira, e é responsável pela produção de equipamentos para a indústria da cortiça. “As principais aplicações dos seus produtos são a transformação da cortiça e a otimização da eficiência e eficácia dos processos de produção e acabamento de rolhas”[7]. Disponibiliza soluções que vão desde linhas de produção completas e personalizadas até aos próprios equipamentos industriais prontos a integrar noutras soluções. Os seus equipamentos servem um vasto leque de operações no ramo da cortiça. No que diz respeito à forma das rolhas disponibilizam brocas, topejadeiras e chanfradeiras para lidar com as questões do tamanho e diâmetro das mesmas. Disponibilizam também máquinas de tratamento, balanças e de escolha. Para tratamento final dispõe de opções para marcação, embalagem e contagem.

2.2 Equipamentos Industriais

Na impossibilidade de, numa fase inicial, desenvolver um equipamento aplicável a todas, sem exceção, os equipamentos industriais da Azevedos Indústria, foram escolhidos 3 de maior relevo produzidos pela empresa. Os equipamentos industriais escolhidos foram:

- Máquina de Capsular
- Máquina de Tratamento
- Balança

A máquina de capsular tem como objetivo colar cápsulas de madeira ou plástico nas rolhas de cortiça. A figura 2.1 é uma representação 3D do equipamento em questão.

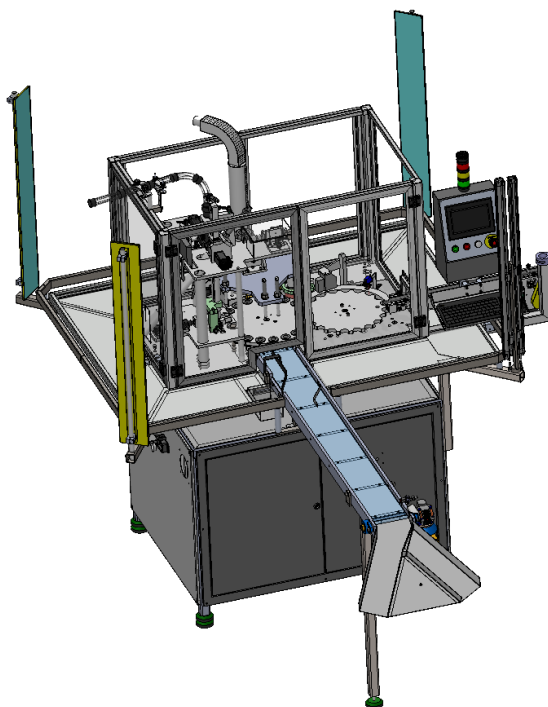


Figura 2.1: Máquina de Capsular

A balança serve para pesar as rolhas e classificá-las como leves, boas ou pesadas e, com base nessa informação, encaminhá-las para a saída correspondente. Esta balança em específico possui duas células de carga. A figura 2.2 é uma representação 3D do equipamento em questão.

A máquina de tratamento tem como objetivo realizar várias operações, nomeadamente aquecimento, injeção de produto, despoeiramento, etc. Este equipamento, em específico, permite injetar até 4 produtos de tratamento. No entanto, só 3 produtos é que podem ter uma injeção controlada por peso, ao invés do controlo por tempo. Na figura 2.3 encontra-se uma representação 3D deste equipamento.

Destaca-se que os PLC's utilizados em todos estes equipamentos são da *OMRON*. Essa informação será pertinente para o desenvolvimento da arquitetura do sistema. Além disso, importa destacar que todos os equipamentos têm já implementados temporizadores, agrupados em 3 categorias: Desde ligada, desde reset e desde sempre. Estes contadores monitorizam os tempos em que o equipamento está ligado, em produção, parada e em alarme. No entanto, cada equipamento também possui variáveis específicas, isto é, que não são comuns a todos os equipamentos.

As variáveis específicas da máquina de capsular dizem respeito, essencialmente, a alarmes e a dados estatísticos do equipamento. A balança inclui contadores, retentivos e não retentivos, do número de rolhas classificadas. Encontram-se ainda valores de cadência, média, variância, valor máximo e mínimo. Estes últimos valores são retentivos. Na máquina de tratamento, existem os valores das quantidades de produto injetado e rolhas tratadas (desde ligada e desde sempre). Além

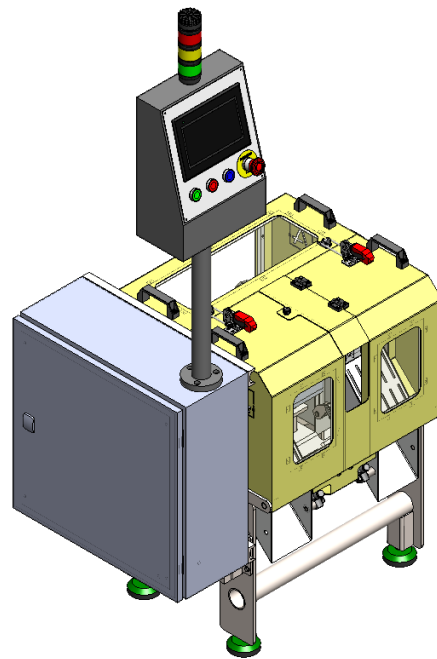


Figura 2.2: Balança

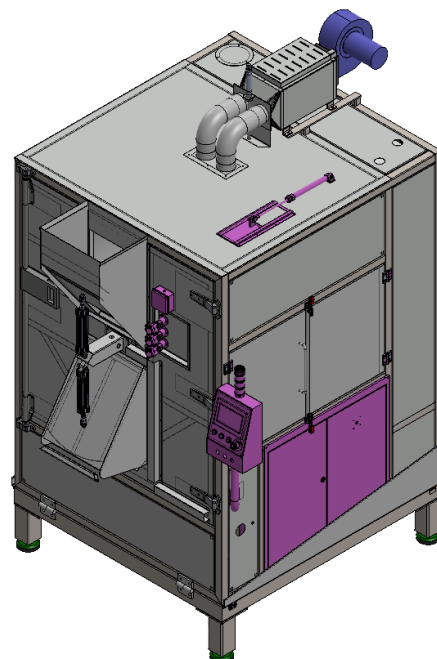


Figura 2.3: Máquina de Tratamento

disso, é possível verificar o estado de todos os atuadores, bem como a quantidade de produto presente no equipamento.

2.3 Análise de requisitos

Para a resolução do problema exposto no capítulo 1, é necessário um dispositivo que permita trazer inteligência às máquinas produzidas pela empresa. Trata-se então de um dispositivo IoT capaz de recolher, processar e publicar dados, provenientes das máquinas, e enviar para um sistemas cloud.

O grande objetivo é ter um equipamento que, quando interligado ao controlador do equipamento industrial (microcontrolador ou PLC), recolha dados. Esses dados podem ser, por exemplo, consumo energético, KPI's (*Key Performance Indicators*), imagens de monitorização de comportamentos de utilizador. Após o seu processamento, podem ser aplicadas ações inteligentes como, por exemplo, verificação de aspetos de manutenção preditiva. Podem ainda ser adicionados sensores que complementem a informação presente no equipamento industrial já existente. O dispositivo deve ser visto como um "add-on" para as máquinas industriais. Posteriormente, este dispositivo deve ser integrado num *sistema cloud* capaz de expor ao utilizador algumas métricas, bem como alertas e *tracking* de ordens de produção.

Com o objetivo de simplificar a exposição inicial das componentes do sistema, a figura 2.4 mostra os 3 elementos base:

- Equipamento industrial;
- Dispositivo IoT;
- Serviço *cloud*.



Figura 2.4: Esquema base do sistema

Importa também mencionar que o conceito de IoT e IIoT traz, por si só, algumas características associadas. É necessário ter em conta que os sistemas industriais estão expostos a condições adversas, nomeadamente poeiras, vibrações e humidades. Os custos associados também são um fator a ter em conta pois os próprios custos operacionais são um fator importante na introdução de novos dispositivos. Por último, o *downtime* também deve ser o mais baixo possível pois os equipamentos podem operar ininterruptamente. Assim, o sistema IIoT deve ter em conta aspetos relacionados com:

- **Robustez:** Os dispositivos escolhidos devem ter em conta o ambiente onde vão ser utilizados;
- **Tempo de resposta:** Alguns dispositivos funcionam com base de *triggers*. Desse modo, a resposta dos sistemas IIoT devem ser a mais rápida quanto possível;

- **Diagnóstico Manutenção Remota:** Na medida do possível, a implementação do sistema IIoT deve suportar a realização de um diagnóstico remoto em caso de falha e, se possível, permitir a sua reparação de forma também remota;
- **Atualizações:** O sistema deve suportar atualizações durante um prazo de vida útil definido, além de garantir que a introdução de novas atualizações não tornam inoperacionais os dispositivos existentes.

Tendo em conta os aspetos acima mencionados, bem como a motivação que origina esta dissertação, é possível fazer um levantamento para a análise de requisitos. Para isso, os requisitos podem ser analisados de acordo com 3 grupos:

- Requisitos do dispositivo IoT a adicionar junto do equipamento industrial;
- Requisitos do serviço *cloud*.
- Requisitos do sistema visto como um todo;

A tabela 2.1 expõe os requisitos de forma agrupada. Posteriormente será realizada uma discussão sobre os mesmos. Os requisitos funcionais dizem respeito àquilo que o sistema deve fazer, enquanto os requisitos não funcionais referem-se a características que o sistema deve ter, como aspetos de performance, design, entre muitos outros.

Tabela 2.1: Requisitos do Sistema

	Requisitos funcionais	Requisitos não funcionais
Sistema como um todo (T1)	T1.1: Analisar diversos aspetos relacionados com a manutenção preventiva e preditiva de cada equipamento industrial	
	T1.2: Calcular diversos indicadores de performance e manutenção	
	T1.3: Monitorizar o consumo energético, temperatura e humidade	
	T1.4: Gerar um alerta sempre que algum aspeto não esteja conforme previsto	
	T1.5: Informar o utilizador sempre que for gerado um novo alerta	
	T1.6: Realizar tracking de ordens, provenientes de código QR ou de barras	
	T1.7: [Máquina de capsular] Gerar um alerta quando a taxa de rejeição for muito elevada	
	T1.8: [Balança] Procurar tendências nas pesagens de cada célula	
	T1.9: [Máquina de tratamento] Monitorizar o caudal de um determinado produto (g/s), sinalizando os declínios pois podem existir alguma obstrução nos tubos.	
Dispositivo eletrónico (T2)	T2.1: Recolher dados de diversas fontes, tais como PLC's, sensores externos, etc	T2.2: Ter hardware e encapsulamento robustos para serem inseridos no chão de fábrica
	T2.3: Comunicar com o sistema cloud	T2.4: Ser configurável
	T2.5: O utilizador deve ser capaz de criar o seu próprio Set de Regras e adicionar funcionalidades extra	T2.6: Ser modular
	T2.7: Permitir o acesso remoto através de uma VPN	T2.8: Possuir entradas e saídas de gama industrial
	T2.9: Disponibilizar um servidor OPC-UA com variáveis de relevo	T2.10: Possuir uma interface responsável por expor dados de relevo ao utilizador
		T2.11: A interface deve apresentar painéis próprios, dedicados aos aspetos de manutenção, que só podem ser acedidos por utilizadores autenticados
Sistema cloud (T3)	T3.1: Permitir adicionar clientes	
	T3.2: Permitir adicionar novos dispositivos	
	T3.3: Os dados apresentados a cada cliente devem poder ser personalizados	
	T3.4: Permitir associação entre dispositivos e clientes	
	T3.5: Calcular métricas de desempenho do equipamento industrial	
	T3.6: Sinalizar alarmes específicos que ocorram com mais regularidade do que a prevista	
	T3.7: Apresentar indicadores de performance e manutenção	
	T3.8: Analisar as tendências dos dados disponíveis	
	T3.9: Permitir exportação de dados para aplicações externas (por exemplo, desenvolvidas pelo cliente)	

O dispositivo deve recolher dados do equipamento industrial já existente. Visto que existem métricas que não estão disponíveis nos controladores já instalados, este sistema deve ser facilmente dotado de sensores que permitam a medição de outras grandezas relevantes. Importa salientar que a qualidade da rolha é influenciada pela temperatura e humidade ambiente, e consequentemente devem ser consideradas como relevantes, bem como a potência instantânea e as vibrações da máquina. Analisando a relação entre a corrente de arranque a corrente nominal de um motor elétrico é possível prever se existe alguma anomalia associada ao mesmo. O espectro de vibrações de um motor elétrico permite verificar se existe alguma anomalia no equipamento industrial.

Com a informação recolhida neste ponto, o sistema deve ser capaz de calcular diversos indicadores e analisar diversos aspetos relacionados com a manutenção preventiva e preditiva dos equipamentos. Sempre que algum aspeto não esteja conforme esperado, deve notificar o utilizador.

O sistema deve estar dotado de uma interface que apresente ao utilizador o estado do equipamento e deve ter opções que só devem ser acedidas pelos técnicos de manutenção. Importa fazer a distinção entre a manutenção dada pelo fabricante e a manutenção realizada pelo utilizador. Também deve analisar códigos de barras e QR que contém informação de ordens de produção e realizar o seu *tracking*. O sistema deve estar dotado de entradas e saídas de gama industrial. Com estes IO's (*Inputs&Outputs*) e com a restante informação já recolhida, o utilizador deve ser capaz de criar o seu próprio conjunto de Regras.

O sistema deve conseguir comunicar com outros equipamentos industriais bem como, após processamento, disponibilizar variáveis de relevo através de um servidor OPC-Ua e enviar informação para o sistema cloud. O sistema cloud deve permitir a adição de clientes e dispositivos. Os clientes são os proprietários de equipamentos industriais com o novo “add-on”. Os dados apresentados a cada utilizador devem ser personalizados, assim como só devem ter acesso aos equipamentos a ele associados, garantindo assim a privacidade. Esta componente, além de expor dados, deve sinalizar alarmes que ocorram com maior regularidade do que a prevista. Deve estar também disponível uma visão global de todo o chão de fábrica associado a cada cliente. Por último, o envio de dados para aplicações externas (desenvolvidas pelo cliente, por exemplo) deve ser possível, desde que autorizado.

Importa salientar que cada tipo de equipamento utilizado para a prova de conceito tem requisitos próprios. Desse modo, na máquina de capsular, deve gerar-se um alerta quando a taxa de rejeição for muito elevada. Na balança deve procurar-se tendências nas pesagens de cada célula. Esta necessidade surge, pois as células de pesagem são muito sensíveis o que pode originar problemas na pesagem e criar incertezas aos utilizadores. Pode verificar-se a presença de *offset's* ou discrepância de valores, ou seja, uma maior presença de valores pico a pico elevados. Na máquina de tratamento, deve-se monitorizar o caudal de um determinado produto (g/s), sinalizando as variações pois pode existir alguma obstrução nos tubos. Também deve ser apresentada a evolução da temperatura da máquina ao longo do tempo.

Além disso, deve ser garantida uma forma de prestar assistência remota ao dispositivo IoT.

2.4 Análise crítica

Neste capítulo foi já apresentada toda a problemática, bem como o mapeamento dos requisitos. Para simplificar a análise, a figura 2.5 mostra o diagrama conceptual inerente aos requisitos. Verifica-se então a presença de 3 atores principais: o *utilizador* do equipamento industrial, o *técnico de manutenção do cliente* e o *fabricante*. O fabricante pode aceder a todos os aspetos referentes ao dispositivo IoT, assim como do equipamento industrial. Os outros atores têm acessos restritos a certas ações. O dispositivo IoT deve ser capaz de comunicar com o equipamento industrial e sensores externos. Esses sensores externos podem monitorizar aspetos do equipamento industrial, bem como aspetos do meio envolvente. Com estes dados, deve processá-los, calcular métricas e enviar mensagens para o sistema *cloud*.

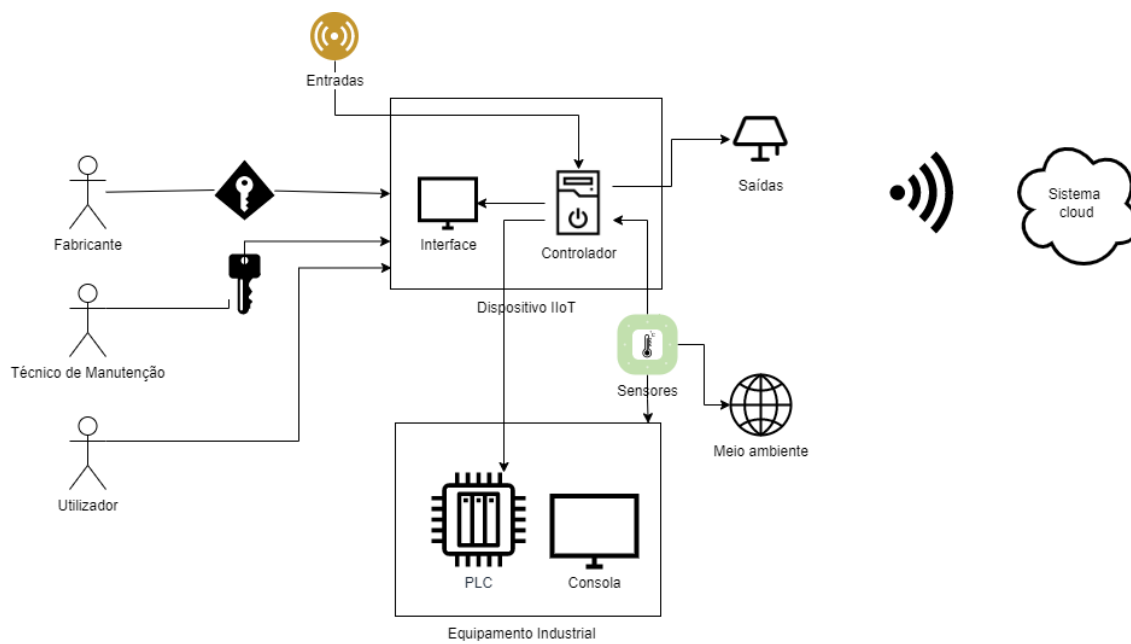


Figura 2.5: Diagrama conceitual do sistema

Neste ponto, é possível proceder à análise de soluções já existentes, bem como de componentes para o projeto de uma solução própria, o que será endereçado no próximo capítulo.

Capítulo 3

Estado de Arte

Neste capítulo apresenta-se o estado da arte. Inicialmente, são abordados os aspetos teóricos relacionados com a manutenção e, posteriormente, duas abordagens para a resolução do problema: a primeira abordagem consiste na análise de soluções personalizadas (chave na mão) que cumpram os requisitos expostos na secção 2.3, bem como soluções similares já implementadas; a segunda abordagem consiste no desenvolvimento uma solução de raiz. Para isso, será analisada a estrutura típica, conceitos base e conhecimentos sobre protocolos de comunicação. Também será analisado um conjunto extenso de componentes relativos a sensores, interfaces, processamento local, redes, processamento na *cloud* e serviços VPN necessários à implementação.

Por último será realizada uma análise crítica sobre as duas abordagens realizadas, sustentando a opção seguida no desenvolvimento do sistema.

3.1 Manutenção

O correto funcionamento dos equipamentos industriais é fundamental para a vitalidade da indústria. Importa aqui definir os conceitos de erro, falha e avaria. O erro consiste na discrepância entre um valor calculado ou medido e o seu valor verdadeiro. Por outro lado, o conceito de falha consiste na incapacidade de aturar conforme planeado, devido a um estado interno do equipamento. A avaria é a perda de capacidade de executar uma tarefa como planeada. Importa salientar que um erro pode originar uma falha e esta, consequentemente, provocar uma avaria[8].

Para garantir o bom funcionamento destes equipamentos é necessário realizar manutenção. Neste contexto, é possível definir 3 tipos de manutenção:

- Manutenção Preventiva
- Manutenção Corretiva
- Manutenção Preditiva

A manutenção preventiva tem como base evitar avarias e garantir o correto funcionamento do equipamento industrial. É uma manutenção planeada, baseada no uso do equipamento, que

evita "a ocorrência de avarias e garante o funcionamento seguro e eficiente do equipamento"[9]. A manutenção corretiva ocorre quando é detetada uma avaria ou falha no equipamento. É uma intervenção não planeada e normalmente dispendiosa, pois além do preço da reparação, obriga à não utilização do equipamento.

Será dado foco à manutenção preditiva, pois permite a redução de falhas nos equipamentos e melhora o seu desempenho. Esta manutenção necessita de informação histórica e atual proveniente de sensores e, recorrendo a algoritmos preditivos, consegue orientar o cronograma de manutenção. "São intervenções que se enquadram na chamada manutenção inteligente, realizada apenas em caso de necessidade"[9]. Possui diversas vantagens, pois permite reduzir a ocorrência de falhas ou degradações, bem como aumentar a eficiência e qualidade de operação do sistema, pois evitam as paragens não programadas e ajudam na melhoria da qualidade das operações e na prevenção de acidentes. Além disso, reduzem os custos de manutenção a longo prazo. Esta manutenção pode ser realizada tendo em conta as seguintes técnicas[10]:

- Monitorização das vibrações;
- Utilização da termografia para medição da temperatura do equipamento;
- Análise do óleo, quando utilizado;
- entre outros.

A análise da corrente consumida por um motor elétrico permite identificar, por exemplo, se alguma das fases do estator está em curto-circuito, a quebra de barras rotóricas, a influência do momento de inércia, entre outros[11]. Além dos sensores e algoritmos preditivos, é possível recorrer a medidas de desempenho e fatores relevantes de manutenção. Tendo em conta os equipamentos selecionados pela empresa proponente, consideram-se os seguintes indicadores:

- MTBF: *Mean Time Between Failures*[12];
- MTTR: *Mean Time To Repair*[12];
- OEE: *Overall Equipment Effectiveness*[13].

O MTBF consiste no tempo médio entre falhas e pode ser calculado considerando o número de falhas ocorridas e o tempo de correto funcionamento do equipamento. Desse modo, este indicador pode ser calculado através da seguinte expressão:

$$MTBF = \frac{\sum \text{Tempo Em Correto Funcionamento}}{\text{Paragens pela necessidade de manutencao corretiva}} \quad (3.1)$$

O MTTR consiste no tempo médio de reparação e pode ser calculado considerando o tempo total de reparação e o número de intervenções. Desse modo, este indicador pode ser calculado através da seguinte expressão:

$$MTTR = \frac{\sum \text{Tempo Em Reparacao}}{\sum \text{Intervencoes realizadas}} \quad (3.2)$$

O OEE (Eficácia Geral do Equipamento) é dado pela multiplicação de 3 fatores: disponibilidade, desempenho e qualidade. Tendo em conta os equipamentos seleccionados pela empresa proponente, é possível obter o seguinte conjunto de equações:

$$OEE = Disponibilidade \times Desempenho \times Qualidade \quad (3.3)$$

$$Disponibilidade = \frac{\sum tempo producao}{\sum (tempo ligada - tempo em manutencao planeada)} \quad (3.4)$$

$$Desempenho = \frac{Rolhas Acumuladas}{\sum Tempo ligado \times constante de producao} \quad (3.5)$$

$$Qualidade = \frac{\sum Rolhas produzidas conforme}{\sum Rolhas produzidas} \quad (3.6)$$

3.2 Estudo de Mercado

Conforme previamente mencionado, a primeira abordagem consiste na busca de soluções comerciais que cumpram os requisitos já analisados na secção 2.3, bem como soluções similares já implementadas. Este ultimo grupo é de relevo para a construção de uma solução própria. A figura 3.1 mostra como estão agrupadas as soluções encontradas.

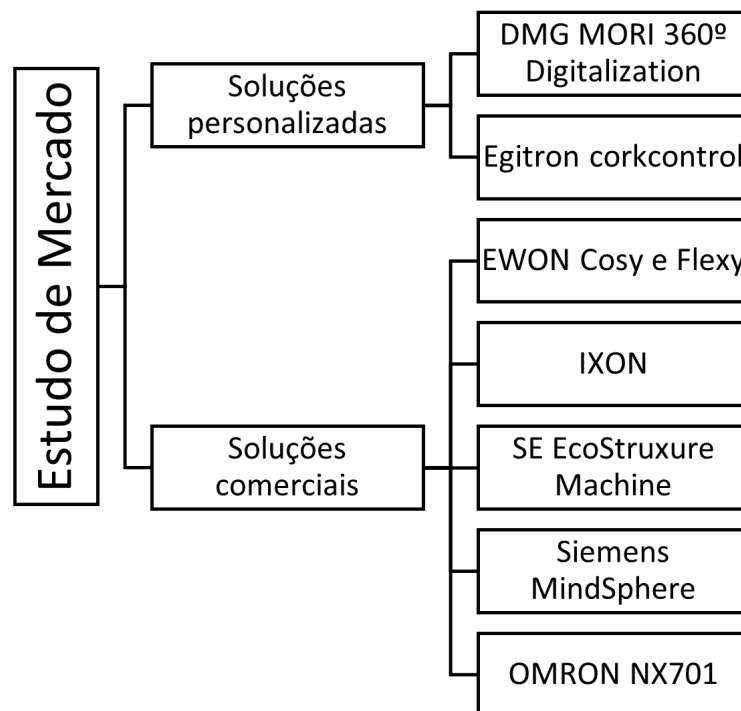


Figura 3.1: Estudo de mercado

3.2.1 Soluções similares

A DMG MORI é fabricante de maquinaria para produção de peças metálicas e apresenta um produto denominado “360° Digitization”[14], uma solução de software que suporta o planeamento, preparação, assim como a monitorização, manutenção e serviço. A sua solução assenta num módulo de comunicações com os equipamentos recorrendo ao *OPC UA*, *umati*, *MTconnect* & *MQTT*. Disponibiliza ainda diagnóstico remoto e ligação segura. O serviço *cloud* monitoriza em tempo real, mostra painéis informativos e oferece notificações instantâneas. O *dashboard* apresentado ao utilizador também tem uma interface completamente personalizável.

Consideraram-se também os produtos desenvolvidos por uma empresa muito próxima geograficamente da Azevedos Indústria, a Egitron, que apresenta uma solução bastante interessante na área de IoT. A “corkcontrol”[15] possui uma interface amigável cujo objetivo é ajudar no controlo de qualidade, otimizar o produto e o tempo dos colaboradores. Utilizam como protocolo de comunicação o *MQTT* e as principais valências apresentadas são a criação de relatórios dinâmicos e a apresentação da interface de dados disponível em várias línguas.

3.2.2 Soluções comerciais

A EWON[1] é uma empresa belga especializada em soluções de comunicações industriais (independentes do fornecedor) e tem como principal objetivo ajudar os fabricantes de equipamentos industriais a tornar os seus processos mais digitais, implementando os conceitos IIoT e Indústria 4.0. Para isso, apresenta diversas soluções tais como: Cosy, Flexy e Talk2M.

O Cosy é um gateway industrial que permite aceder e resolver problemas dos equipamentos industriais de forma remota. Através da plataforma Talk2M, pode configurar-se um serviço de e-mail e SMS para receber notificações do equipamento. Permite também gerar relatórios e *logs* de toda a atividade remota. Possui 4 interfaces Ethernet configuráveis, USB, leitor de cartão de memória, entre outras. Atualmente, a comunicação com o serviço cloud é feita pela porta Ethernet, no entanto está previsto o lançamento do dispositivo como WiFi e 4G.

O Flexy, além de funcionar como gateway industrial, traz consigo aspetos de IIoT. Permite recolher informação de PLC’s, sensores e permite criar “sets” de regras próprios, notificações e monitorização de desempenho através de *dashboards*. Além disso, permite exportar dados para aplicações de terceiros e apresenta diversas cartas de expansão. A figura 3.2 mostra os exemplos de utilização destas soluções.

O Talk2M é a plataforma cloud própria da EWON que centraliza as opções de acesso remoto. Apresenta também várias camadas de segurança, desde *firewall* e encriptação de tráfego.

A IXON[16] apresenta uma solução bastante similar à apresentada pela EWON, com grande foco no controlo remoto. Além disso, realiza o *log* de dados e a monitorização de alarmes. Como produtos de conectividade dispõe do IXRouter e o IXAgent. O seu serviço *cloud*, a IXON Cloud, facilita o acesso ao controlo remoto, disponibiliza API’s integradas, gestão de dispositivos, painéis do equipamento industrial e *white-labelling*, mesmo no *edge device*.

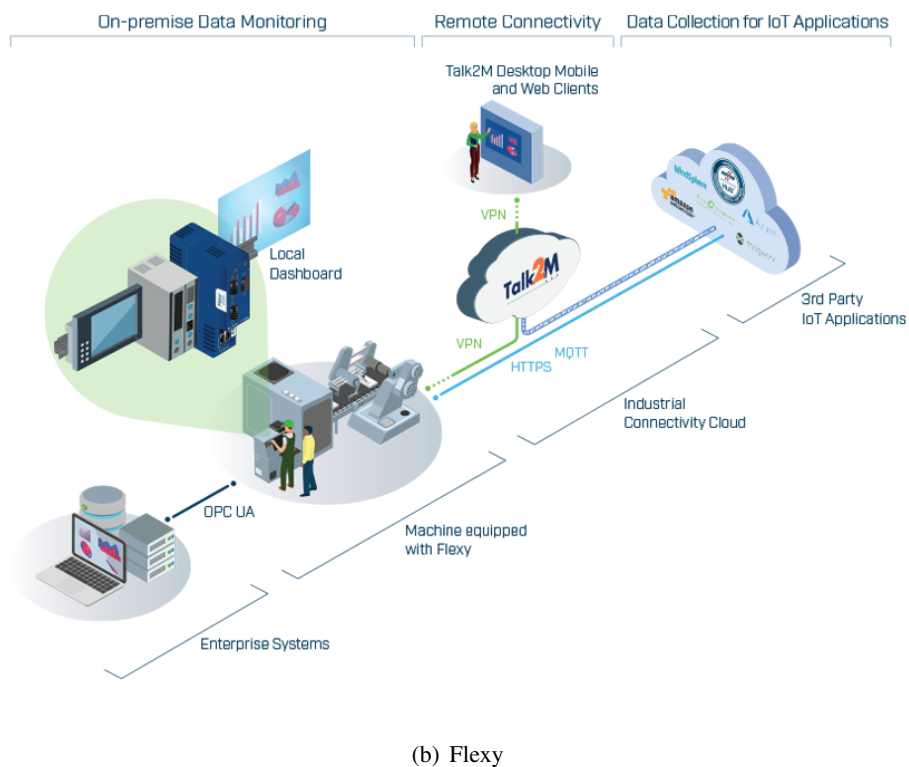
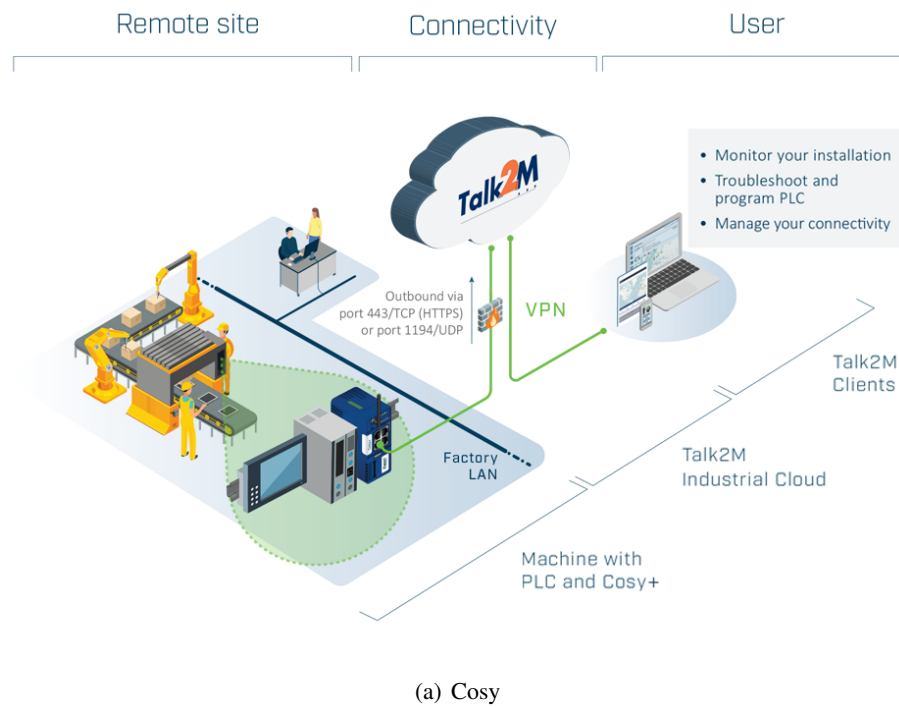


Figura 3.2: Exemplo de utilização das soluções da EWON (retirado de [1])

Não obstante, também se analisaram as soluções apresentadas pelos fabricantes de PLC's. A Schneider Electric apresenta o EcoStruxure Machine[17], que consiste em interligar os dispositi-

tivos recorrendo a *edge devices*. A mais-valia consiste na redução de custos de operações, pois recorre à realidade aumentada, permitindo "que os operadores visualizem dados em tempo real e objetos virtuais sobrepostos sobre um armário elétrico, máquina ou instalação." [17]

A Siemens também apresenta soluções de IIoT. O MindSphere[18] assenta no IIoT como um serviço e promete criar uma solução desde *edge devices* até à *cloud*. Dá ainda acesso a análise de dados em tempo real, e permite ter uma operação proativa, recorrendo a análise avançada e a inteligência artificial. Também é possível recorrer ao Mendix, que é uma plataforma de programação *low-code* que junta todos os recursos desde *edge devices* a sistemas de dados. Em suma, permite criar aplicações em pouco tempo e com poucos recursos.

Importa também mencionar as soluções apresentadas pela OMRON. Esta empresa apresenta os PLC's da gama NX701[19] com controlador de Inteligência Artificial (AI) integrado. Este controlador consegue detetar anomalias rapidamente e com grande exatidão. Tem como principal vantagem ter o controlador de AI no *edge device*, evitando assim sistemas *fog* ou conectividade com *cloud*. Importa salientar que estes conceitos serão abordados com mais detalhe na secção 3.3. Além disso, tem a funcionalidade de *SQL Client*. Assim sendo, consegue fazer operações de *insert*, *update* e *select* em 3 bases de dados distintas. Tem também 2 MB de memória onde é possível armazenar dados caso não seja possível estabelecer ligação com a base de dados.

A OMRON disponibiliza também diversos computadores industriais capazes de correr sistemas SCADA, ERP/SAS e SQL. Permite também a instalação de um *Real Time Operating System*.

3.3 Estrutura típica e conceitos base

A segunda abordagem consiste na criação de um sistema próprio para preencher os requisitos já analisados. Para proceder a esta abordagem é imprescindível conhecer a arquitetura deste tipo de sistemas e conceitos adjacentes.

Relativamente à arquitetura, importa mencionar que ainda não existe um padrão, isto é, uma normalização para este tipo de sistemas. No entanto, a arquitetura mais comum pode ser observada na figura 3.3.

Os sensores e atuadores são os dispositivos que comunicam com o meio ambiente. Surgem neste ponto 2 conceitos fundamentais: *edge computing* e *cloud computing*. O *cloud computing* consiste no envio direto da informação recolhida pelos sensores para o sistema *cloud*, isto é, a informação é enviada sem ter em conta a sua relevância.

O conceito de *edge computing* consiste em processar os dados junto ao meio ambiente, isto é, os dados são enviados para uma camada entre os sensores e atuadores e o sistema *cloud*. No esquema acima mostrado representa-se pelo "Local processing and storage". Só a informação de relevo é enviada para o sistema *cloud* e a restante é eliminada ou mantida no dispositivo junto ao meio ambiente, para uma análise posterior. Este conceito, por sua vez, traduz-se em serviços mais rápidos e confiáveis, pois permite diminuir a latência e aumentar a velocidade de procura de dados.[5]

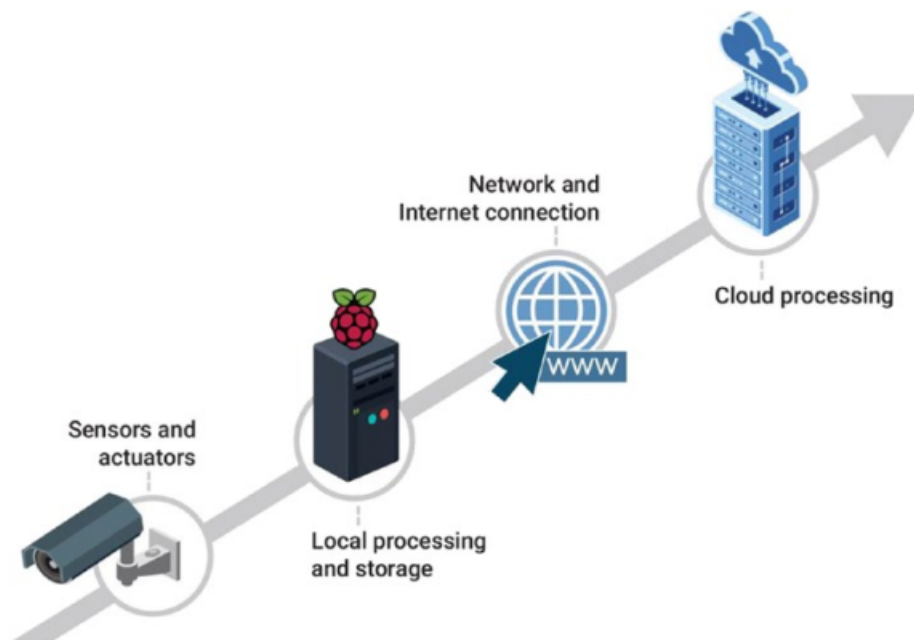


Figura 3.3: Arquitetura padrão dos sistemas de IoT (retirado de [2])

Os *edge devices* podem ser conectados à Internet através de *routers* e *Access Points*. O processamento na *cloud* é onde estão presentes os algoritmos de *machine learning* e inteligência artificial. Para o processamento na *cloud*, pode ser adicionado o conceito de *digital twin*, onde é simulado o ambiente onde o equipamento se encontra com dados reais.[5]

O conceito de IoT e de *Machine Learning* estão muitas vezes interligados. Este último monitoriza online o comportamento de várias variáveis, assim como prevê a performance e a fiabilidade. Os algoritmos de *Machine Learning* são a chave, por exemplo, para a deteção de avarias. Com os algoritmos certos e recorrendo à análise de dados e técnicas estatísticas, é possível criar um ambiente seguro.

3.4 Protocolos de comunicação

Um sistema de IIoT possui diversos dispositivos interligados. Para isso é necessária a utilização de protocolos de comunicação.

Observando novamente a figura 3.3 verifica-se que o “Local processing and storage” estabelece comunicações com os sensores e atuadores, bem como o processamento na *cloud*. Assim sendo, serão abordadas as interfaces e protocolos mais comuns em cada segmento.

3.4.1 Background

Os protocolos de comunicação devem ser vistos como um conjunto de regras que permitem que vários equipamentos troquem dados entre si. Para isso, segundo o modelo OSI (Open Systems Interconnection), existem 7 camadas protocolares (visíveis na figura 3.4), empilháveis entre si.

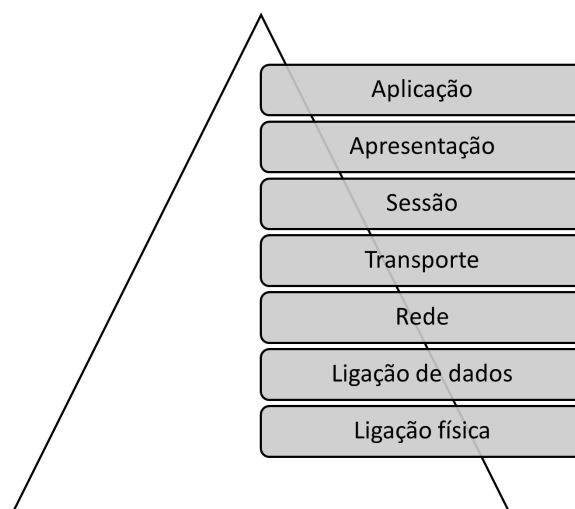


Figura 3.4: Modelo OSI

Na indústria é bastante comum a implementação do modelo OSI colapsado, onde as aplicações acedem diretamente à camada de ligação de dados.

3.4.2 Comunicação com sensores e atuadores

Um dispositivo, inteligente ou não, que pretenda comunicar com outros dispositivos, pode ter ao seu dispor diversos protocolos. Esses protocolos podem estar implementados na camada física e/ou em camadas superiores. Aos protocolos implementados na camada física também se pode dar o nome de interfaces.

As interfaces RS485 são bastante comuns no meio industrial, pois além de serem ligações balanceadas, são multiponto. Possuem também uma elevada imunidade ao ruído e uma troca de transmissão considerável (10 Mbits/s). Podem ser utilizadas em distâncias médias (1 a 2 km) e em modo *simplex* ou *half-duplex*, sendo que o seu uso em *full-duplex* é limitado.

Na camada de ligação de dados, destacam-se os seguintes protocolos:

- I2C é a sigla de *Inter-Integrated Circuit*. É um protocolo que funciona na base de múltiplos *masters/slaves*. Para isso são usadas também duas linhas de comunicação: SDA (Serial Data) e SCL (Serial Clock).
- SPI representa *Serial Peripheral Interface* e funciona na base de 1 *master* e vários *slaves*. Para isso utiliza 3 linhas de comunicação: Master Output/ Slave Input (MOSI), Master Input/ Slave Output (MISO) e clock (SCLK).

Na camada de aplicação destacam-se os seguintes protocolos:

- MODBUS é um protocolo industrial que pode ser implementado em diferentes camadas de transporte, tais como Serial, Ethernet ou Wi-Fi. Tem uma tipologia mestre-escravo. Desse modo, o mestre solicita informação ao cliente, sendo que o mesmo responde ao solicitado. Para este efeito utiliza registros[20].

- OPC-UA (*OPC Unified Architecture*) foi construído sobre o TCP/IP com uma arquitetura *service-oriented*. Desse modo, existem servidores e clientes. O cliente solicita e escreve informação no servidor. O nó é a estrutura base da informação e contém atributos, o valor atual e referência para outros nós[21].
- FINS (*Factory Interface Network Service*) é o protocolo proprietário da OMRON. Serve para trocar mensagens entre controladores numa rede OMRON FA. Pode estar implementado em diferentes camadas físicas, tais como Ethernet, DeviceNET e RS232. As tramas de comando e resposta necessitam os campos DNA, DA1, DA2, SNA, SA1, SA2 que dizem respeito, essencialmente, a elementos que identifiquem a rede e o equipamento de destino[22].

3.4.3 Comunicação com a *cloud*

Um *edge device* deve conseguir comunicar com a *cloud*. Para isso tem de ter ao seu dispor uma ligação à Internet. Essa ligação à Internet pode ser feita via 4G, fibra ótica, 5G, cabo, satélite, entre outros. Nos sistemas IIoT são necessários protocolos que sejam eficientes e tenham como base pequenas mensagens. Assume-se aqui que existe uma ligação à Internet para poder estabelecer a ligação ao serviço de *cloud*. Assim sendo destaca-se o MQTT, HTTP e o CoAP.

- MQTT é a sigla de *Message Queuing Telemetry Transport* e é um protocolo bastante comum no seio do IoT. Está implementado sobre o protocolo TCP/IP e utiliza a tipologia *publisher/subscriber*. Permite filtrar mensagens por assunto, pois cada mensagem tem um tópico associado. Além disso, disponibiliza 3 níveis de "Quality of Service (QoS)". No QoS de nível 0, o cliente publica a mensagem e não aguarda o "acknowledgement" do broker. No QoS de nível 1, o cliente publica a mensagem e aguarda o "acknowledgement"(ACK) do broker, no entanto, caso o ACK seja perdido, a mensagem é novamente enviada. No QoS de nível 2, é garantido que a mensagem chega uma e uma só vez ao broker[23].
- HTTP é a sigla de *Hypertext Transfer Protocol* e utiliza como base uma ligação TCP. Além disso, tem vários métodos disponíveis como GET, POST, PUT, etc.
- CoAP é a sigla de *Constraint Application Protocol* e foi desenhado essencialmente para dispositivos limitados. Tem um funcionamento similar ao HTTP pois utiliza métodos de GET, POST, PUT, etc. No entanto, os pacotes são mais pequenos.

3.5 Análise de componentes

Para desenvolver uma solução de raiz, é necessário analisar os componentes mais comuns para desenvolver uma solução proprietária integrada nos equipamentos industriais. Tendo em conta esses mesmos equipamentos e os requisitos analisados, seria necessário:

- Sensores;

- Interfaces;
- Processamento local;
- Ambientes de desenvolvimento;
- Serviços cloud;
- Serviço VPN.

Na impossibilidade de analisar as milhares de soluções existentes, as componentes expostas nas subsecções seguintes são uma amostra representativa da panóplia existente no mercado. Desse modo, apresentam-se as soluções mais comuns.

3.5.1 Sensores

Esta subsecção serve para expor diversos tipos de sensores. Para facilitar a abordagem, a figura 3.5 mostra as categorias de sensores a analisar, bem como os exemplos em questão.

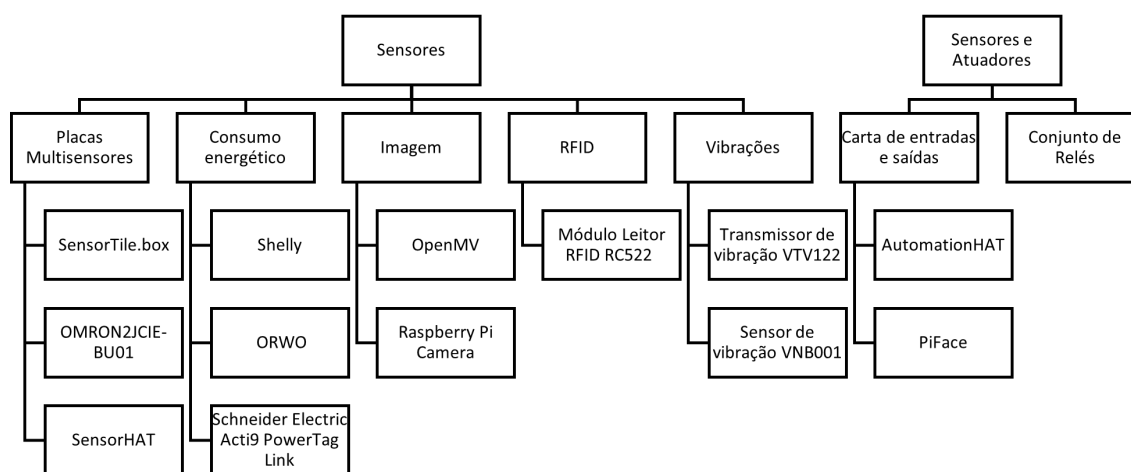


Figura 3.5: Panóplia de sensores a analisar

3.5.1.1 Placas Multisensores

O primeiro grupo de sensores a analisar são vocacionados ao uso em sistemas de IoT. Serão então abordados:

- SensorTile.box
- OMRON 2JCIE-BU01

- SenseHAT

O SensorTile.box [24] (ver figura 3.6(a)) é um multi sensor wireless direcionado para IoT e para aplicações *wearable*. De dimensões pequenas e com bateria própria, este dispositivo disponibiliza sensores baseados em movimento e ambiente. Tem como interfaces USB, Bluetooth e leitor de cartão de memória. Através de uma aplicação para *smartphone*, é possível aceder aos valores de cada componente, bem como criar as próprias aplicações. Disponibiliza diversos IDE's onde é possível aceder a funções de inteligência artificial e redes neuronais. Inclui nativamente uma aplicação para monitorização de vibrações. Inclui os seguintes sensores de precisão:

- Sensor de temperatura (STTS751)
- IMU de 6 eixos (LSM6DSOX)
- Acelerómetro de 3 eixos
- Magnetómetro de 3 eixos
- Sensor de pressão (LPS22HH)
- Microfone (MP23ABS1)
- Sensor de humidade (HTS221)

A OMRON também apresenta, numa gama não industrial, um multisensor compacto: o 2JCIE-BU01[25] (ver figura 3.6(b)). Este dispositivo, de dimensão de uma pequena *pen USB*, disponibiliza dados de temperatura, humidade, luz, pressão barométrica, som e acelerómetro de 3 eixos. Tem como interfaces USB e Bluetooth. Possui já algoritmos integrados de deteção de vibração, bem como de tremor de terra. Consegue também monitorizar a qualidade do ar através de um sensor VOC (*Volatile Organic Compounds*).

A SenseHAT[26] (ver figura 3.6(c)) é uma board de expansão para o Raspberry Pi que contém giroscópio, acelerómetro, magnetómetro, sensor de temperatura, pressão barométrica e humidade. Além disso, dispõe de uma matriz LED 8x8 e de um *joystick* de 5 botões.

3.5.1.2 Potência Elétrica

O consumo energético, conforme já mencionado, tem um papel relevante na manutenção preditiva. Tendo em conta que os equipamentos industriais produzidos pela empresa proponente não possuem nenhum medidor de consumo energético, é necessário escolher uma componente responsável por esta tarefa. Assim sendo, destacam-se 3 opções:

- Shelly
- ORWO
- Schneider Electric Acti9 PowerTag Link

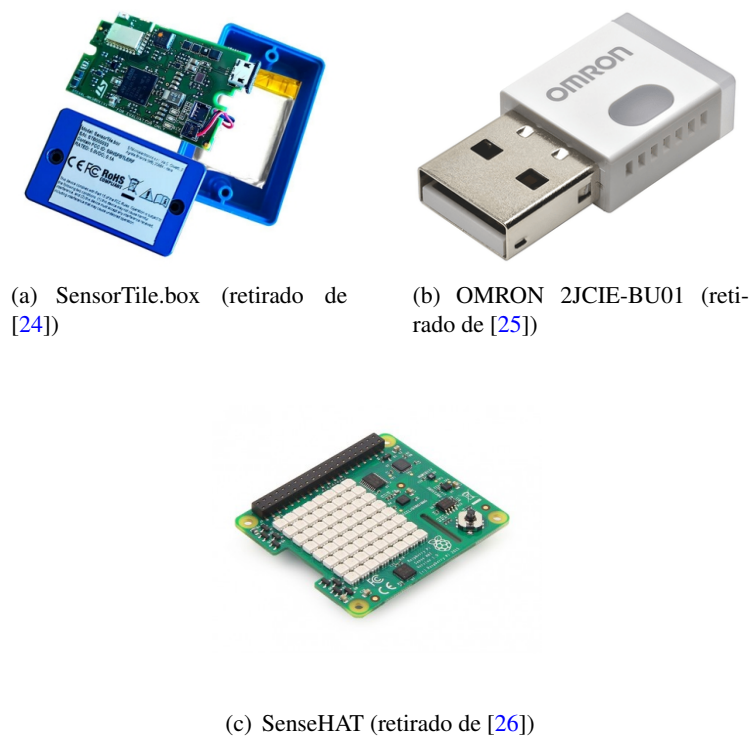


Figura 3.6: Multisensores

O Shelly 3EM[27] (ver figura 3.7(a)) permite monitorizar o consumo de um sistema trifásico com uma corrente máxima, por fase, de 120 A. Possui uma plataforma própria onde é possível visualizar todo o histórico de consumo ou produção, sendo que o dispositivo tem memória para armazenar dados até 1 ano, caso esteja offline. Esta empresa disponibiliza um produto semelhante caso a necessidade seja a monitorização de consumos em sistemas monofásicos, o Shelly EM[28]. Salienta-se que é possível aceder à informação presente neste dispositivo através de MQTT.

Da mesma forma, a ORNO[29] (ver figura 3.7(b)) também apresenta soluções de medição de consumo energético preparadas para sistemas monofásicos ou trifásicos e permitem medição de correntes até 100 A. Apresentam versões que permitem a sua montagem em calhas DIN ou em parede. É possível obter diversos dados destes equipamentos através de MODBUS RTU, saída pulsada ou pelo display incorporado. Está disponível a tensão e corrente aparente por fase, bem como a frequência da rede. É também possível ler a potência ativa, reativa e aparente por fase e total, bem como o fator de potência.

Por último, importa também sublinhar que existem soluções preparadas para o meio empresarial, isto é, desenvolvidas por fabricantes de renome no seio da indústria. Para isso apresenta-se o Acti9 PowerTag Link[30] da Schneider Electric (ver figura 3.7(c)). É um módulo de fácil instalação que permite, além de monitorizar o consumo energético, criar alarmes para o consumo excessivo (*overload*) e para a baixa de tensão. Tem a enorme vantagem dos sensores de medição serem *wireless* e permitir a ligação máxima de 100 dispositivos. Permite realizar as comunicações via Modbus TCP/IP. Há que ressaltar que esta solução só permite medição de correntes até 63 A.



(a) Shelly 3EM (retirado de [28])

(b) ORNO OR-WE-517 (retirado de [29])



(c) Schneider POWERTAG F63A (retirado de [30])

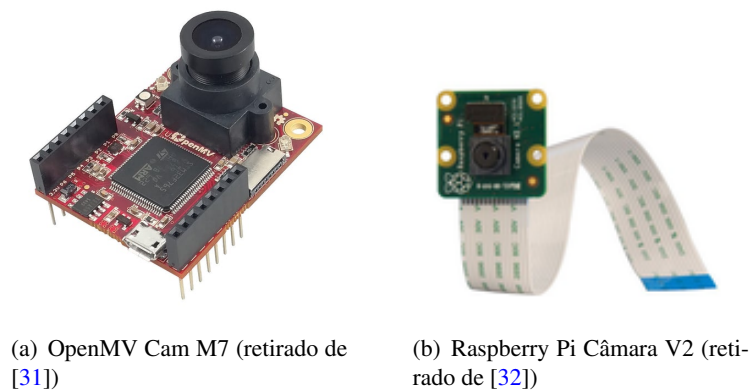
Figura 3.7: Medidores de consumo energético

3.5.1.3 Imagem

O processamento de imagem é uma área muito vasta. Apesar disso, existem dispositivos desenvolvidos com o intuito de facilitar as operações nesta área. O OpenMV (ver figura 3.8(a)) destaca-se devido à sua popularidade e pode ser visto como "o Arduino da Visão Computacional". Este dispositivo torna-se deveras interessante, pois traz consigo grande parte dos algoritmos de reconhecimento de imagem, tornando a sua utilização muito mais simples.[31]

Também existe a Raspberry Pi Camera[32] (ver figura 3.8(b)) que pode ter, ou não, modo de filtragem de IR. A versão “standard” da câmara traz esse filtro, possibilitando assim fotografias em ambiente normal. A câmara pode ser acedida através da linha de comandos do próprio Raspberry Pi ou por uma aplicação em Python. No entanto, esta abordagem requer alguns conhecimentos de processamento de imagem, pois as funcionalidades básicas da câmara são a obtenção de imagens e vídeos, assim como a aplicação de alguns filtros pré-existentes.

Para o processamento de imagem mais avançado, pode utilizar-se redes neuronais implementadas no Google Coral[33] ou no Intel Neural Compute Stick[34].



(a) OpenMV Cam M7 (retirado de [31])

(b) Raspberry Pi Câmera V2 (retirado de [32])

Figura 3.8: Câmaras

3.5.1.4 RFID

O RFID (Radio Frequency Identification) é sinónimo de identificação por radiofrequência. O módulo RC522 (ver figura 3.9(a)) permite ler e escrever em cartões que "seguem o padrão Mifare"[35]. Este módulo é bastante comum em projetos com Arduino. Já traz consigo um cartão e um chaveiro.

3.5.1.5 Vibrações

A monitorização de frequências do motor elétrico é um aspeto extremamente relevante para a manutenção. Através desta prática é possível detetar anomalias, pois cada equipamento produz vibrações numa determinada gama. Os desequilíbrios, provocados pelo desgaste, má utilização, etc, provocam alterações nessa mesma gama. Desse modo, consideraram-se 2 equipamentos desenvolvidos para o efeito:

- Transmissor de vibração VTV122
- Sensor de vibração VNB001

O transmissor de vibração VTV122[36] (ver figura 3.9(b)) deteta a vibração no sistema através da medição e avaliação da velocidade da vibração (mm/s). Tem uma saída em corrente. Para a utilização deste equipamento é necessário garantir que não existem camadas elásticas entre o equipamento e o sensor.

O sensor de vibração VNB001[37] (ver figura 3.9(c)) tem mais vantagens quando comparado com o anterior. Possui duas saídas, sendo que uma delas é configurável. Podem ser utilizadas saídas digitais com alarme prévio e principal. Também pode ser configurada uma saída em corrente entre 4 a 20 mA, com cálculos de média configuráveis. É possível manter o histórico de dados, assim como a tendência.



Figura 3.9: RFID e Detecção de vibrações

3.5.1.6 Cartas de entradas e saídas

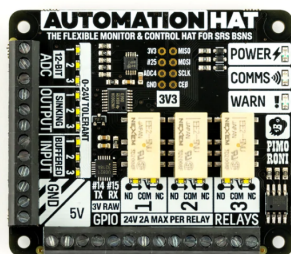
Finalmente é de relevo analisar componentes que permitam dotar o dispositivo IoT com entradas e saídas de âmbito industrial. Só assim é que é possível conectar sensores e atuadores deste mesmo âmbito. Para isso consideraram-se cartas de entradas e saídas, bem como um conjunto de relés.

A Pimoroni apresenta uma solução com bastantes potencialidades, a Automation HAT[38] (ver figura 3.10(a)). Esta carta possui 3 ADC's de 12 bits que permitem ligações de 0 a 24 V, entradas, saídas e relés de 24 V. Disponibiliza as conexões de SPI e UART. Uma grande vantagem desta carta é o facto de já existirem bibliotecas de Python e nós de Node-RED desenvolvidos.

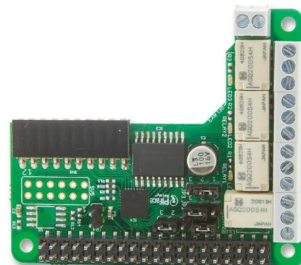
Sob o mesmo ponto de vista, a OpenLX também oferece uma solução semelhante, o projeto PiFace[39] (ver figura 3.10(b)), onde o seu grande objetivo é conseguir ligar o Raspberry Pi ao mundo exterior. Para isso oferece diversas opções para controlo, visualização, RTC (Real Time Clock) e relés. A carta em questão é a PiFace Relay Plus e traz consigo diversas vantagens: é stackable, pelo que podem ser ligadas até 8 boards iguais por Raspberry. Também permite conectar uma board PiFace EXTRA na expansão.

Os relés presentes nas duas soluções acima mencionadas tem a tensão máxima de 24 e 20 V, respetivamente. Eventualmente, é também possível utilizar um conjunto de relés que permitem

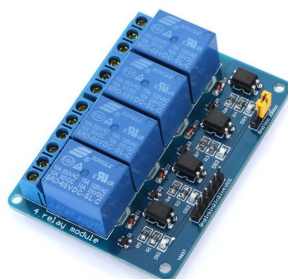
controlar cargas de 250 VAC ou 30 VDC[40] (ver figura 3.10(c)). Para atuar estes relés basta introduzir à entrada um sinal digital. Este componente é bastante utilizado em projetos de Arduino, bem como na domótica.



(a) Automation HAT (retirado de [38])



(b) PiFace Relay Plus (retirado de [39])



(c) Módulo de 4 relés (retirado de [40])

Figura 3.10: Cartas de entradas e saídas

3.5.2 Interfaces Homem-máquina

De acordo com os requisitos expostos na secção 2.3 é necessário uma interface homem-máquina capaz de expor dados ao utilizador. Deve também permitir a existência de secções próprias para os responsáveis de manutenção. Para isso consideraram-se displays. Deste modo, consideraram se:

- HMI's (Human-Machine Interface) Nextion;
- *Display* tátil compatível com Raspberry.

As consolas apresentadas pela Nextion[41] possuem processador e memória integradas, bem como diversos tamanhos. Para a utilização deste componente, é necessário utilizar a GUI (Graphical User Interface) que auxilia no desenho das diversas páginas apresentadas pela consola. Para comunicar com este componente basta ter uma interface série TTL (Transistor-Transistor Logic).

Da mesma forma, existem diversos *touchscreens* compatíveis com o Raspberry Pi, de diferentes tamanhos e resoluções. Para este efeito, apresenta-se um display de 3.5 polegadas e uma

resolução de 320 X 480[42]. A utilização desta componente obriga a desenvolver uma aplicação para apresentar os painéis ao utilizador.

Estas opções de interfaces não são únicas. É possível recorrer a teclados matriciais[43] e a displays LCD[44].

3.5.3 Processamento Local

Os dispositivos de processamento local são responsáveis por recolher a informação proveniente dos sensores e fazer um processamento primário. Depois disso, devem enviar informações para o sistema *cloud*.

O **ESP32** é um microcontrolador muito comum no meio do IoT devido ao baixo custo e ao baixo consumo energético. Permite também a utilização de várias interfaces, tais como I2C e UART. Tem também à sua disposição Wi-Fi e Bluetooth, no entanto, possui uma memória de apenas 520 kB. Também não tem um elevado poder de processamento, sendo que a sua utilização como *edge device* torna-se bastante limitada[45]. Este microcontrolador encontra-se disponível em diversas *shields*, com diversos periféricos já associados[46].

O **Raspberry Pi** é um microcomputador bastante utilizado nas aplicações de IoT devido à sua dimensão, poder de processamento, baixo custo e elevada comunidade. Pode funcionar como um computador pois possui interface HDMI, USB, tomada de áudio e porta Ethernet. Tem também à sua disposição vários conectores GPIO (General Purpose Input/Output). Disponibiliza também interfaces para comunicar com sensores e atuadores, assim como com a *cloud*.

É necessário ter em consideração que estes equipamentos serão colocados em contexto industrial, com elevadas vibrações, poeiras e outras condições que poderão não ser ideais. Consequentemente, apresentam-se as seguintes soluções que integram os microcomputadores e tornam-os mais adequados para a introdução em ambiente fabril:

- Monarco HAT[47]
- Iono Pi [48]
- Unipi [49]
- Revolution Pi [50]
- UNO-220 – Advantech da FFonseca [51]

3.5.4 Ambiente de desenvolvimento

Os ambientes de desenvolvimento apresentam as ferramentas e infraestruturas necessárias para o desenvolvimento do sistema. As soluções abaixo expostas visam simplificar o processo de implementação, bem como de reutilização e adaptação. Apresentam-se 4 soluções bastante promissoras para o projeto:

- Node-RED

- Codesys
- n8n
- Apache nifi

O **NODE-RED** é uma ferramenta de desenvolvimento baseada em fluxo. Originalmente criada pela *IBM's Emerging Technology Services*, em 2013, tinha como objetivo a manipulação de tópicos MQTT. Atualmente suporta uma vasta gama de protocolos, tais como: MODBUS TCP, MC (Protocolo proprietário dos PLC's da Mitsubishi), FINS, entre muitos outros[52]. Traz consigo inúmeras vantagens, tais como:

- Programação realizada em *browser*
- Suporta JSON, que é a notação de objetos do JavaScript. Esta arquitetura simplifica o próprio programa em si, quando comparado com XML.
- É compatível com vários *edge devices*, tais como Raspberry Pi e BeagleBone e Android. Permite também a interação com Arduino.
- É também compatível com várias aplicações de *cloud* de renome, tais como Microsoft Azure, Amazon Web Services e IBM Cloud.

O **CODESYS** é um *softPLC* com base na norma de programação IEC-61131-3 e permite utilizar várias linguagens de programação distintas, assim como diversos protocolos de comunicação industriais, tais como PROFIBUS, EtherCAT, CANopen, PROFINET e Ethernet/IP. É também possível utilizar o Raspberry Pi como um *softPLC*. Esta aplicação também traz consigo várias bibliotecas inerentes ao conceito de IIoT, cujo objetivo é fazer a leitura e escrita nas estruturas de dados. Para o efeito, disponibiliza o acesso a conteúdos por HTTP e HTTPS, MQTT, AWS IoT, Azure IoT, Google Cloud IoT, entre muitas outras[53].

O **n8n** é uma ferramenta também baseada em programação por fluxo e de uso livre que permite manipular dados. Mantém os dados na máquina local e não na *cloud*, sendo que a utilização em cloud não é gratuita. Possui diversos modelos de utilização desenvolvidos por outros utilizadores. O seu objetivo geral é realizar a ligação entre diversos API's, ou seja, entre diferentes interfaces. [54]

Também baseada numa programação de fluxo, o **Apache NiFi** permite criar movimento de dados entre fontes e destinos, garantindo uma transmissão fácil e segura. Para o efeito, assegura a resolução de diversos aspetos, tais como a concorrência. Permite perceber facilmente o fluxo criado. A arquitetura base assenta em JVM (*Java Virtual Machine*). No topo da hierarquia utiliza um servidor Web, que garante o acesso à API da aplicação. A seguir, existe o "*flow control*", que é o gestor de todas as operações. Também pode recorrer a extensões, no entanto, estas extensões correrão na JVM. O *FlowFile Repository* é onde é armazenado o estado de cada FlowFile e se o mesmo está ativo no fluxo. O *Content Repository* é onde são armazenados os FlowFile. O *Provenance Repository* é onde todo o histórico de eventos são armazenados. [55]

3.5.5 Processamento na *cloud*

O sistema *cloud* pretende armazenar e processar a informação recolhida e disponibiliza-la a um operador ou outras aplicações através de uma API normalizada. Deve conseguir sinalizar alarmes e fazer o *tracking* de ordens de produção. A solução deve ser flexível, onde os dados expostos a cada cliente devem ser personalizados. Existem no mercado diversas soluções com as mais diferentes funcionalidades. Desse modo, apresenta-se:

- ThingsBoard
- Amazon Web Services (AWS)
- Google Cloud IoT Core
- Azure IoT Hub

O **ThingsBoard**[56] é uma plataforma IoT *open-source* que permite um desenvolvimento rápido e escalável de projetos IoT. Permite adicionar dispositivos e clientes e estabelecer a relação entre eles. Permite também sinalizar alarmes após uma análise complexa dos dados recebidos (*telemetry*). Os dashboard apresentados aos clientes também são totalmente configuráveis e personalizáveis conforme as suas necessidades.

Existem diversas versões disponíveis deste produto no mercado. A *Community Edition* é uma versão gratuita que suporta um número ilimitado de dispositivos e bens associados. Pode ser instalado em Ubuntu, CentOS, Windows, Maven e Raspberry Pi. A sua informação é armazenada localmente. Se estiverem associados muitos dispositivos, pode ser necessário adicionar um serviço de filas. Esta versão permite escolher serviços *cloud* próprios ou armazenar os dados externamente, ou seja, em serviços *cloud* externos.

Paralelamente, existe a *Professional Edition*, com diversas modalidades. O que varia entre elas é o número de dispositivos, bens e número de pontos introduzidos mensalmente. Também existe a opção de *white-labeling* que consiste em alterar os logótipos utilizados. Esta vertente pode ser de elevado interesse no seio empresarial.

Importa apresentar a existência de outras opções: *Trendz Analytics* e *Edge*. A *Trendz Analytics* permite detetar anomalias, realizar uma análise preditiva, calcular campos e apresentar graficamente. Tem-se como exemplo o cálculo do KPI. O plano *Edge*, por ser um plano de subscrição, permite o uso ilimitado de dashboards, pontos de dados e mensagens.

Comparativamente, a **Amazon** apresenta soluções completas para IIoT baseadas na recolha, armazenamento e análise de dados de dispositivos. Uma grande vantagem deste serviço é a sua vertente modular, isto é, o preço a pagar pela sua utilização depende dos módulos utilizados, bem como das características específicas de cada um. No entanto, também possui serviços gratuitos das mais variadas gamas, desde análise, integração de aplicações, base de dados, *machine learning*, robótica, segurança, armazenamento, entre muitos outros[57]. O **AWS IoT Core** tem várias funções. Utilizando a metodologia de *publisher/subscriber* consegue comunicar com diversos dispositivos tendo como vantagem baixa latência e alta taxa de transferência. Além disso, caso ocorra

uma falha na comunicação com o *edge device*, mantém o seu último valor e reporta a situação. Outra enorme vantagem deste serviço é permitir a utilização da *Alexa*, que consegue processar o áudio captado e pedir informações ao AWS IoT Core através de MQTT.

A **Google** oferece um serviço bastante similar ao da Amazon, bem como a vertente modular já observada. Permite o acesso a serviços de análise de dados, *machine learning* e com uma grande abstração por parte do servidor. Esta abstração é relacionada com aspetos de "capacidade, confiabilidade ou desempenho". O **Google Cloud IoT Core**[58] permite a centralização e visualização de dados. Recorre também a *machine learning* para prever problemas e criar modelos. Os protocolos de comunicação utilizados por este serviço são MQTT e HTTP. Assim como Thingsboard e no AWS, existe uma chave assimétrica para garantir a autenticação.

Também a própria **Microsoft** tem uma solução para a gestão de IoT. Para isso apresenta-se o **Hub IoT do Azure**[59]. Esta solução, assim como as anteriores, permite estabelecer comunicação com vários dispositivos IoT em segurança e suporta atualizações OTA (*on-the-air*), ou seja, permite que o dispositivo IoT sejam atualizados de forma remota. Além disso, possui um sistema de backup que permite recuperar o dispositivo em caso de falha. Utilizando o Azure IoT Edge é também possível criar soluções IoT híbridas. Permite detetar anomalias e utilizar algoritmos de *machine learning*.

3.5.6 Serviço VPN

A VPN (Virtual Private Network) estabelece uma ligação segura entre dois pontos recorrendo a um túnel ponto-a-ponto. Este sistema garante a encriptação dos dados pessoais e a camuflagem do endereço IP[60]. Além de ser uma excelente forma de segurança e permitir o acesso a conteúdos regionais a partir de qualquer lugar, a utilização destes serviços pode trazer outras vantagens, tais como:

- Acesso a redes internas de forma remota;
- Acesso diferenciado a recursos disponíveis.

Desse modo, analisaram-se as seguintes opções:

- HusarNET
- Deeper Network
- NordLayer

O HusarNET[61] é uma VPN P2P que permite estabelecer comunicações entre diversos equipamentos, tais como computadores, servidores e microcontroladores com uma fácil configuração. Oferece suporte para diversas aplicações tais como acesso por SSH, acesso ao ambiente de trabalho remoto, comunicar com robots utilizando ROS2 e Docker, atualizar um ESP32 via OTA, entre outros. A grande vantagem deste serviço é permitir a sua utilização gratuita até 5 dispositivos.

A Deeper Network[62] apresenta soluções de VPN descentralizada bastante focada na segurança dos utilizadores. Garante a privacidade dos utilizadores, tem a capacidade bloquear anúncios publicitários, controlo parental e ainda permite “Blockchaing Mining”. Além disso, as suas soluções baseiam-se em dispositivos *plug and play* e com configuração mínima.

O NordLayer[63] é uma VPN empresarial que, além de garantir a segurança dos utilizadores, permite o acesso a uma rede interna por gateways dedicados. Oferece um endereço IP fixo e permite gerir o acesso aos mais variados serviços. Alerta o administrador caso seja detetado algum dispositivo que possa estar vulnerável e pode utilizar autenticação biométrica.

3.6 Análise Crítica

Para sumarizar as questões abordadas neste capítulo, a figura 3.11 mostra um esquema com os respetivos tópicos.

Neste ponto, importa mencionar as soluções possíveis para a resolução da problemática exposta no capítulo 2.3:

- Aplicar um IXON IXRouter aos equipamentos já existentes e disponibilizar, através da IXON cloud, dashboards para o utilizador;
- Aplicar um EWON Flexy aos equipamentos já existentes e disponibilizar, através do Talk2M, dashboards para o utilizador;
- Utilizar o portefólio de produtos da Schneider Electric EcoStruxure Machine, bem como o EcoStruxure Machine Expert para configurar todo o sistema;
- Utilizar o IIoT Data Package da Siemens juntamente com as capacidades básicas do MindSphere Capability Packages. O Mendix pode ser utilizado para ter uma vertente de *low-code*;
- Desenvolver uma solução de raiz.

As duas primeiras soluções (da EWON e da IXON), apesar de serem compatíveis com os equipamentos já produzidos pela empresa, não apresentam uma interface junto do equipamento nem um acesso diferenciado. Além disso, tem serviços de cloud proprietários, quando um dos objetivos seria evitar esta dependência.

Visto que os PLC's utilizados pela empresa são da OMRON, as soluções apresentadas pela Schneider Electric e pela Siemens obrigariam a reescrever todo o código produzido para os equipamentos e obrigaria à substituição dos PLC's nos equipamentos já existentes. Apesar da OMRON apresentar uma solução de IIoT, a mesma continuaria a forçar a troca dos PLC's pois não são o mesmo modelo. O facto de um equipamento industrial estar conectado à Internet pode constituir um perigo para a segurança do mesmo.

Desse modo, a solução passa por construir uma solução de raiz. Esta opção traz várias vantagens, nomeadamente:

- A possibilidade de construir um sistema à medida da empresa;

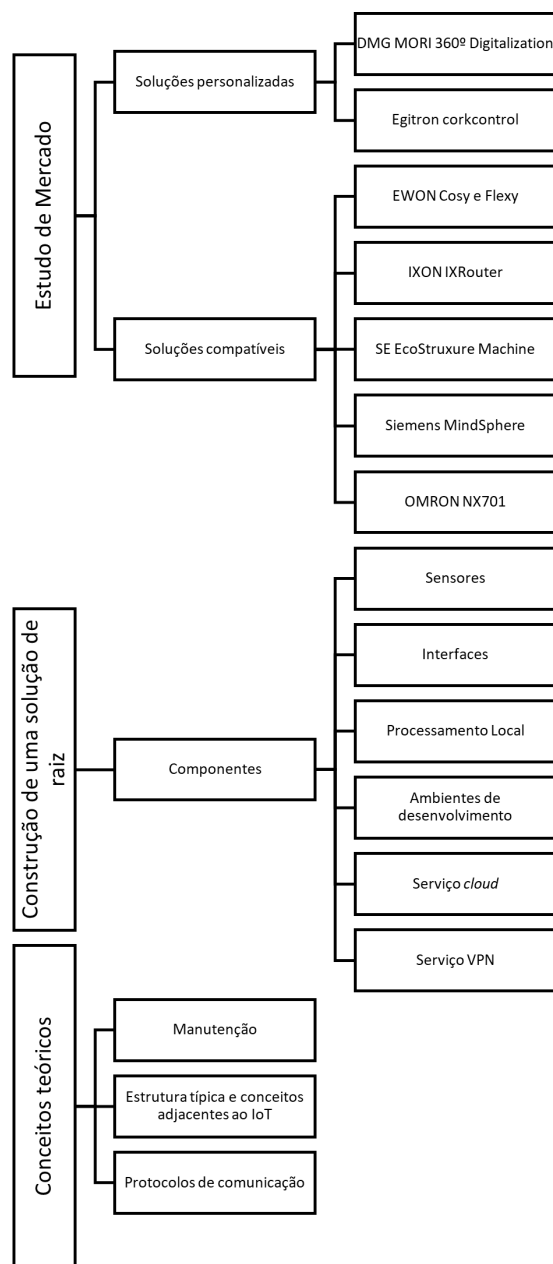


Figura 3.11: Sumário do Capítulo 3

- Poder ser utilizada uma arquitetura modular. Se alguma das componentes falhar ou não for satisfatória, pode ser substituída sem causar a perda de toda a solução;
- É colocada uma proteção indireta, pois o PLC do equipamento industrial não está conectado à Internet. Em caso de ataque, é o sistema IoT que sofre danos;
- A empresa tem um maior controlo de custos;
- A empresa possui um maior domínio da tecnologia utilizada.

Capítulo 4

Arquitetura proposta

Este capítulo apresenta a arquitetura do sistema a desenvolver. A análise parte de um diagrama com a estrutura do sistema e, posteriormente faz uma comparação entre as diversas componentes integrantes já apresentadas no capítulo 3.5. Os aspetos relativamente ao desenvolvimento do software também são abordados, bem como a arquitetura funcional do sistema.

4.1 Estrutura do sistema

Com base nos requisitos expostos na secção 2.3, é possível desenvolver o diagrama apresentado na figura 4.1.

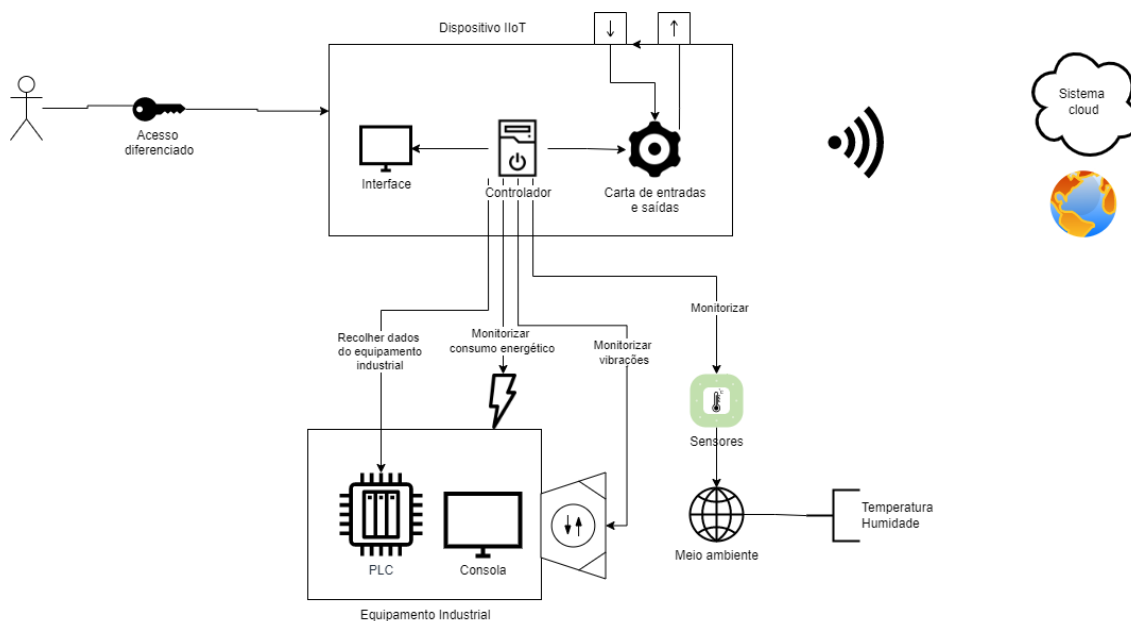


Figura 4.1: Estrutura do sistema

De uma forma geral, o sistema deve:

- Garantir o acesso diferenciado de utilizadores: utilizador, técnico de manutenção e fabricante;
- Ter uma interface interativa com o utilizador;
- Permitir conectar sensores e atuadores de gama industrial;
- Recolher dados do equipamento industrial;
- Monitorizar o consumo energético do equipamento industrial;
- Monitorizar vibrações do equipamento industrial;
- Monitorizar aspetos do meio ambiente, nomeadamente, temperatura e humidade.

4.2 Escolha de componentes

Face à estrutura exposta anteriormente, é necessário escolher os componentes integrantes do sistema. Para isso, serão utilizados os componentes apresentados na secção 3.5. Para facilitar a apresentação, esta escolha será realizada em duas vertentes: *hardware e software*.

4.2.1 Hardware

O primeiro componente a ser escolhido deve ser a unidade de processamento. Só assim é possível garantir a compatibilidade das restantes componentes. Foram analisadas 2 soluções:

- ESP32
- Raspberry Pi

O Raspberry Pi torna-se uma opção mais viável pois tem sistema operativo próprio e um micro-processador, memória RAM e ROM com maior capacidade. Além disso, permite a ligação por *Ethernet*. O ESP32, além de possuir um processamento mais limitado, limita a escolha de cartas de entradas e saídas. Estas características reunidas tornam o Raspberry Pi a melhor opção para este problema.

Para a monitorização do consumo energético foram apresentadas 3 soluções possíveis:

- Shelly
- ORNO
- Acti9 PowerTag Link da Schneider Electric

Destaca-se a solução da ORNO não só devido ao baixo custo relativamente às restantes, mas também pelo facto de disponibilizar a informação via MODBUS. Este aspeto torna-se fundamental para evitar a dependência de serviços prestados por terceiros, como seria no caso da Shelly. A

solução apresentada pela Schneider também seria bastante interessante, no entanto, o seu preço torna-se uma desvantagem para a implementação de um sistema IIoT.

Para monitorizar não só o meio ambiente, mas também as vibrações, encontraram-se as seguintes soluções:

- SensorTile.box
- OMRON 2JCIE-BU01
- SenseHAT
- Transmissor de vibração VTV122
- Sensor de vibração VNB001

A solução apresentada pela STMicroelectronics (SensorTile.box) será a mais indicada para este projeto devido ao seu baixo custo e a integração de redes neuronais para a monitorização de vibrações. A SenseHAT também permite medir as grandezas do meio ambiente, mas como estaria no encapsulamento, os seus dados não fiáveis. Além disso, obrigaria ao *edge device* a realizar uma FFT (Fast Fourier Transform) dos valores do acelerómetro. O transmissor de vibração VTV122 e o sensor de vibração VNB001 são analógicos, o que obriga a recorrer a um ADC para ter acesso aos seus valores. São, de facto, opções bastante dispendiosas para um sistema IIoT. A opção da OMRON (2JCIE-BU01) também pode ser uma boa alternativa para este sistema, no entanto, apresenta um custo elevado.

Relativamente ao processamento de imagem, destacam-se as seguintes opções:

- OpenMV
- Raspberry Pi Camera

Visto que a atividade principal da câmara será a deteção de código QR e de barras, julga-se que a Raspberry Pi Camera é suficiente para o efeito e não será necessário recorrer a serviços de redes neuronais.

Na gama de sensores e atuadores integraram-se soluções de cartas de entradas e saídas ou conjunto de relés. Neste âmbito, destacou-se a AutomationHAT. Comparando com o conjunto de relés, esta opção integra entradas e saídas, e não só somente saídas. O mesmo acontece quando comparado com a PiFace.

Existem duas soluções para garantir o acesso diferenciado a aspetos do sistema: através de um código PIN (Personal Identification Number) ou recorrendo a tags RFID. De forma a evitar o esquecimento ou a passagem indevida de PIN's, considerou-se que o RFID seria a solução mais lógica para esta necessidade. Desse modo, considerou-se o módulo RFID RC522 apresentado anteriormente. Para expor dados ao utilizador foram analisadas as seguintes soluções:

- HMI's (Human Machine Interface) Nextion NX4024T032;
- *Display* tátil compatível com Raspberry A0540BG;

- Display de 2 linhas EF16A0037OK.

Considera-se o HMI da Nextion uma solução robusta, pois permite desenhar a própria interface num GUI e comunicar com o processamento local através de uma ligação série, para além de fazer um isolamento indireto do sistema quando comparado com a utilização do *Display* tátil compatível com Raspberry. Permite também obter uma interface mais limpa e interativa quando comparado com um display de 2 linhas.

4.3 Software

A nível de software, é necessário escolher duas tecnologias: O ambiente de desenvolvimento a utilizar no dispositivo local e a solução de processamento na cloud.

A escolha do ambiente de desenvolvimento tem uma importância elevada pois o objetivo é a utilização real desta arquitetura, bem como a facilidade da sua implementação na indústria. A programação em fluxo torna-se um aspeto chave para esta escolha. Assim, apresentam-se 5 soluções:

- n8n
- Codesys
- obroker.net
- Apache nifi
- Node-RED

O Node-RED é a solução indicada porque tem compatibilidade com os protocolos desejados (FINS, OPC-UA, CoAP, HTTP, HTTPS, etc) e com o Raspberry Pi. Convém referir que baseia na programação em fluxo e disponibiliza diversos nós para as mais variadas funções.

A escolha da plataforma deve ter em conta aspetos relacionados com a segurança, performance e número de dispositivos conectados. Destaca-se então:

- ThingsBoard
- Amazon Web Services (AWS)
- Google Cloud IoT Core
- Azure IoT Hub

Após uma análise cuidada, considera-se que o *Thingsboard* devido à compatibilidade com vários protocolos e à existência de uma versão gratuita (desde que alojada num servidor externo). Além disso, permite a substituição da subscrição a qualquer momento da implementação.

Para a escolha do serviço VPN é necessário um serviço que, além de garantir a segurança, permita a comunicação entre dois dispositivos fora das redes locais. As opções analisadas foram:

- HusarNET
- Deeper Network
- NordLayer

Destaca-se o HusarNET devido ao seu custo (gratuito até 5 dispositivos associados), pela fácil configuração e pela compatibilidade com Raspberry. Os dispositivos da Deeper Network, apesar de serem bastante interessantes, não estabelecem um túnel de ligação entre dois dispositivos localizados em redes distintas. A NordLayer oferece características únicas, no entanto, a sua configuração no Raspberry seria mais complexa.

4.4 Projeto de Software

Neste ponto, já são conhecidas as componentes de hardware, ambiente de desenvolvimento e processamento na cloud. No entanto, ainda não existe arquitetura de software a implementar no dispositivo IoT.

O software a desenvolver deve ter 3 partes fundamentais: Entradas e saídas, gestor de manutenção e set de regras.

O primeiro elemento deve ser responsável por trocar dados com o PLC do equipamento industrial, o sensor, o medidor de consumo energético, o módulo de RFID e a consola HMI e, após um pré-processamento dos mesmos, enviar a informação necessária para o segundo bloco.

O segundo bloco, por sua vez, deve gerir as entradas recebidas, calcular diversos indicadores e, caso algum parâmetro esteja fora do valor esperado, alertar o utilizador. Para isso, deve atualizar a informação presente na consola e enviar um e-mail com o mesmo aviso. Deve ainda analisar a problemática específica de cada equipamento industrial. No caso da máquina de tratamento é a análise do caudal da tubagem dos produtos de injeção.

O terceiro e último bloco é o set de regras. Este bloco deve permitir ao utilizador configurar os seus próprios alarmes e adicionar elementos como, por exemplo, dados provenientes das entradas analógicas e digitais. Além disso, este bloco deve ter acesso a toda a informação.

A figura 4.2 mostra a arquitetura base do software a implementar.

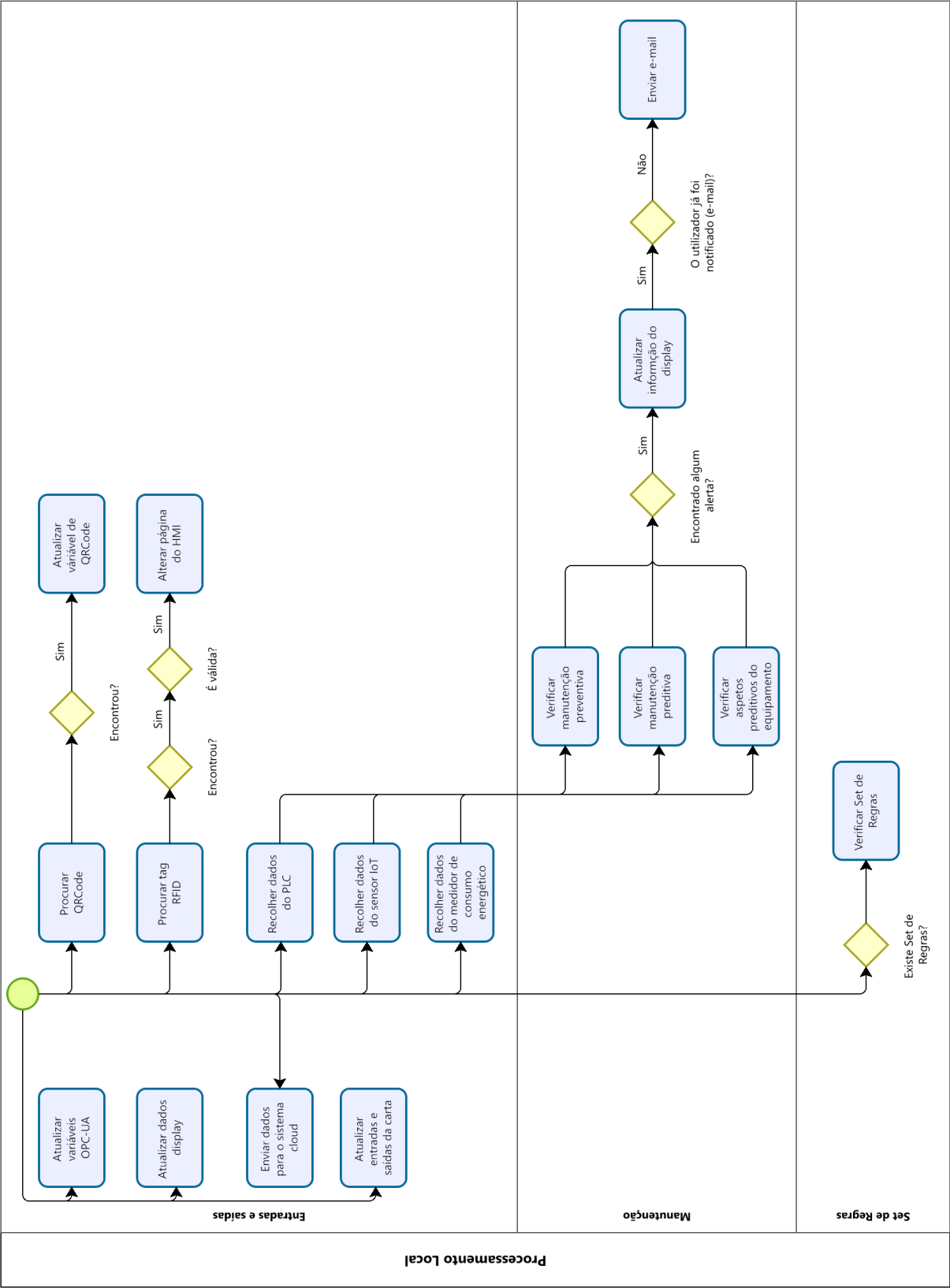


Figura 4.2: Arquitetura do software .

Em suma, a Figura 4.3 mostra a arquitetura funcional do sistema.

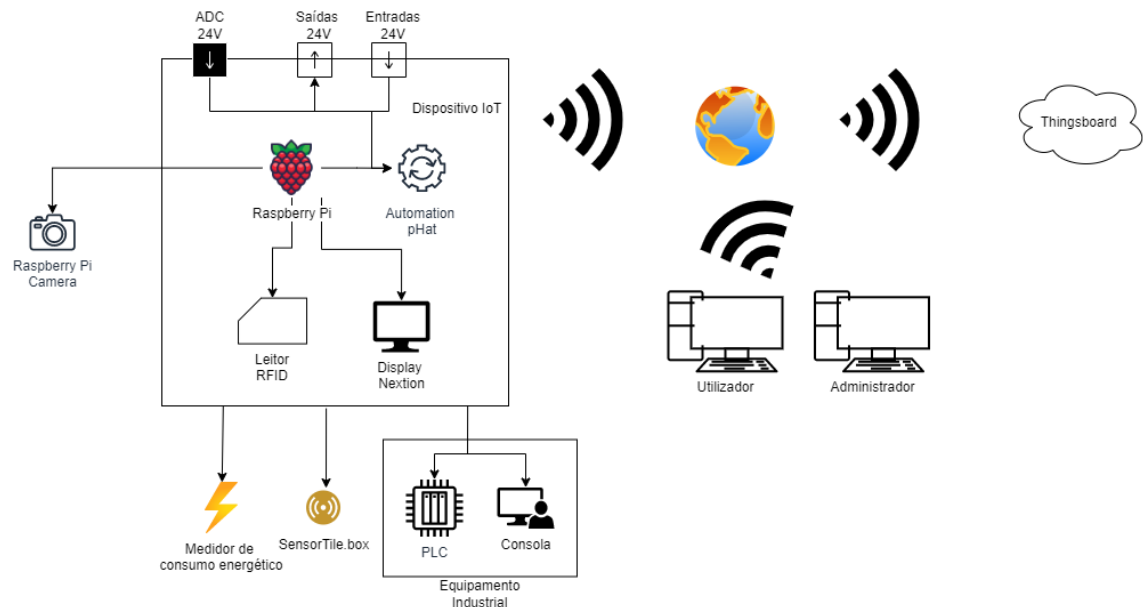


Figura 4.3: Arquitetura Funcional do Sistema

Capítulo 5

Implementação

Este capítulo começa com a exposição do hardware escolhido e as ligações dos diversos componentes. De seguida, aborda as duas camadas de implementação do software: Processamento Local e Processamento na cloud, onde é apresentada a respetiva organização, bem como alguns detalhes de relevo da implementação. A Figura 5.1 mostra um esquema que sintetiza todas as questões abordadas neste capítulo.

Devido à escassez de componentes eletrónicos verificada atualmente, só foi possível implementar o sistema para a máquina de tratamento. As restantes máquinas não puderam ser testadas porque não era possível adquirir os respetivos PLC's. No entanto, manteve-se a arquitetura o mais genérica quanto possível para facilitar a integração de novos equipamentos.

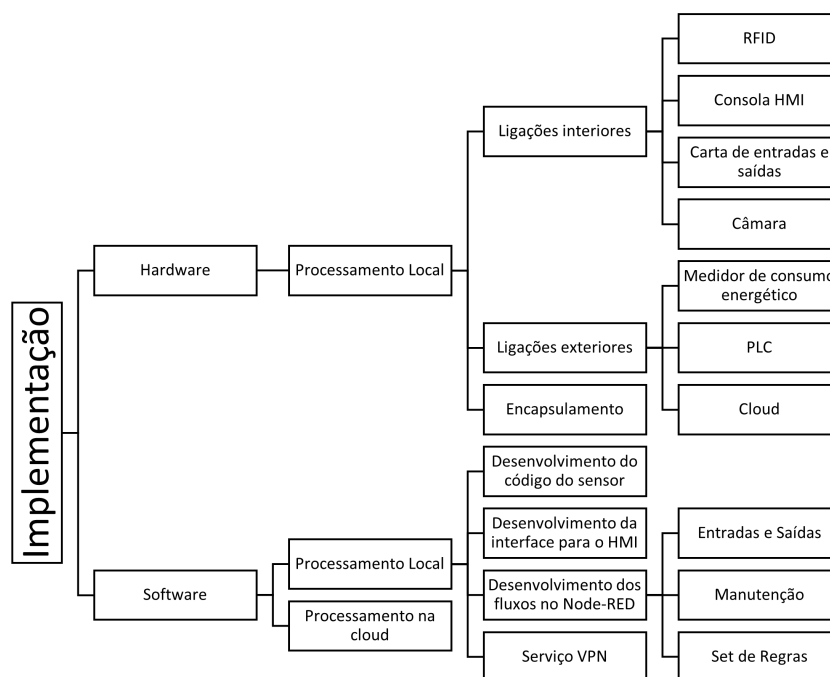


Figura 5.1: Organização do capítulo

5.1 Hardware

A base de todo o sistema de processamento local é o Raspberry Pi.

Importa ressaltar que, devido à localização do dispositivo IoT, não é prática a utilização da fonte de alimentação fornecida com o Raspberry Pi. Para colmatar esta situação, utilizou-se um step-down. Este componente permite colocar tensões de entrada entre 9 a 36 V(DC) e obter uma saída fixa de 5 V[64]. Desse modo, é possível utilizar a fonte de alimentação do PLC (24V (DC)) para alimentar este dispositivo.

Além disso, utilizou-se a *Automation pHAT*[65] em vez da proposta *Automation HAT*, pois a mesma já estava disponível na empresa e as diferenças mais visíveis eram a quantidade de relés disponíveis. Disponibiliza 3 entradas digitais e analógicas e 3 saídas, que estarão disponíveis ao utilizador. Deixa-se também disponível ao utilizador o pino 7 do GPIO do Raspberry Pi por motivos de expansão, pois permite a conexão de sensores 1-Wire.

A descrição geral das ligações realizadas encontra-se na figura 5.2.

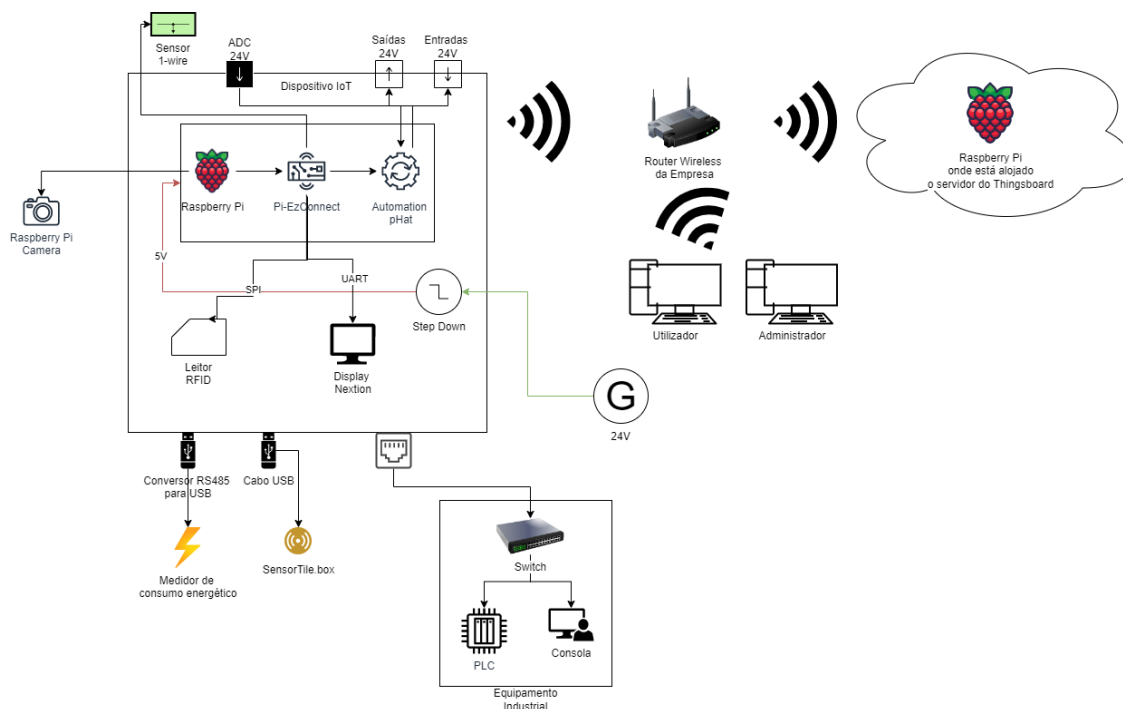


Figura 5.2: Sistema implementado

Após a análise da arquitetura proposta, verifica-se que é necessário garantir nas extremidades do dispositivo IoT:

- Uma entrada de alimentação;
- As saídas e entradas provenientes da Automation pHAT;
- A saída do conector da Câmara;
- O pino 7 do GPIO do Raspberry, que permite conectar sensores 1-Wire;

- As portas USB's e Ethernet;

É possível definir dois grupos de ligações: as ligações interiores e exteriores à caixa do dispositivo IoT. Salienta-se que as portas USB e Ethernet estão disponíveis no exterior do dispositivo.

5.1.1 Ligações Interiores

Foram identificadas as seguintes ligações interiores:

- A ligação SPI para o módulo de RFID, através da interface disponível no Raspberry Pi.
- A ligação UART para o display HMI, através da interface disponível no Raspberry Pi.
- A ligação com a Câmara é estabelecida através da *Camera Serial Interface Type 2* (CSI-2), sendo que o Raspberry tem um conector dedicado para o efeito[66]. Importa relembrar que elemento é responsável por obter a imagem do código QR.
- Uma ligação I2C e vários GPIO's para a *Automation pHAT*.

A Figura 5.3 mostra um esquema das ligações interiores realizadas. A Figura 5.4 mostra o diagrama elétrico. Apesar da *Automation pHAT* utilizar todos os pinos do GPIO do Raspberry, representam-se aqueles que são necessários para o seu funcionamento.

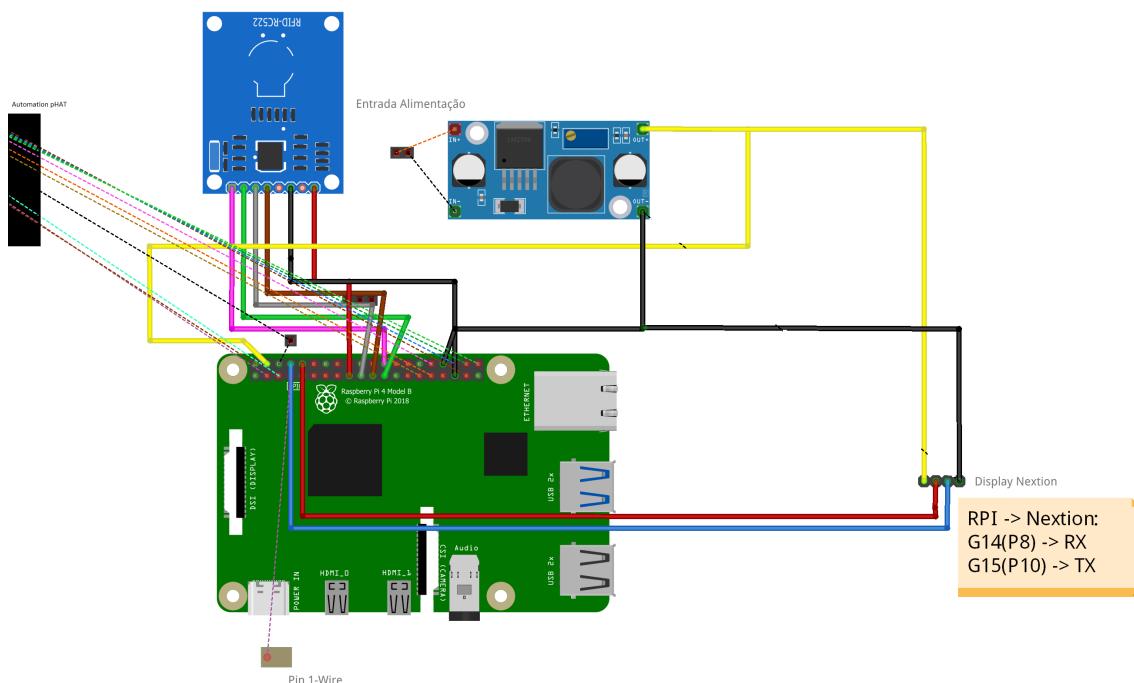


Figura 5.3: Esquema das ligações Interiores do dispositivo IoT

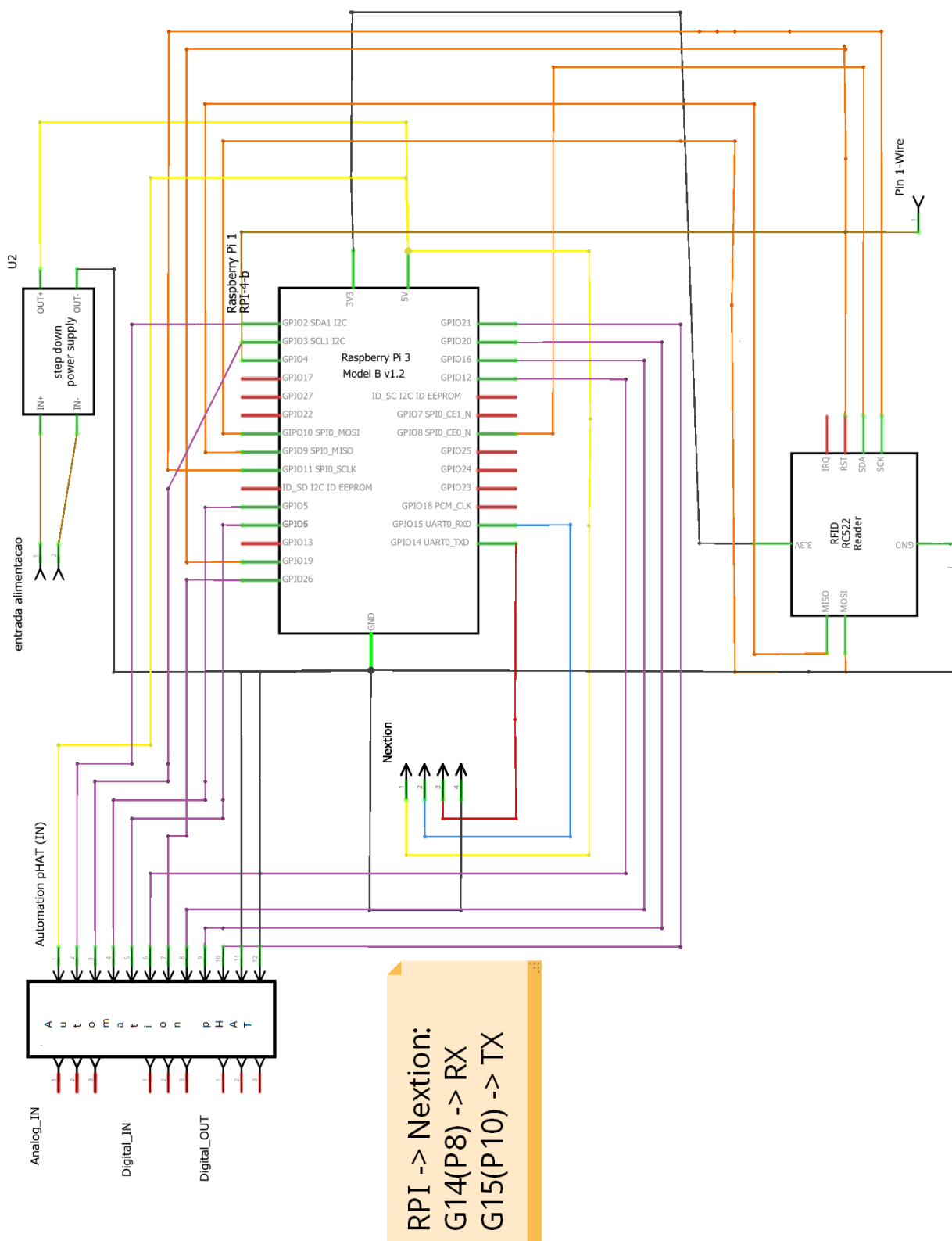


Figura 5.4: Diagrama elétrico das ligações Interiores

Para facilitar as ligações ao Raspberry Pi, utilizou-se a placa de ligação *Pi-EzConnect* da Alchemy Power Inc[3]. Esta placa apenas facilita as ligações a realizar ao GPIO do Raspberry Pi, sem introduzir nenhum componente adicional. Uma grande vantagem desta placa é o facto de ser empilhável, o que permite conectar outros equipamentos na parte superior, como no caso da *Automation pHAT*. A figura 5.5 mostra uma representação da placa utilizada.

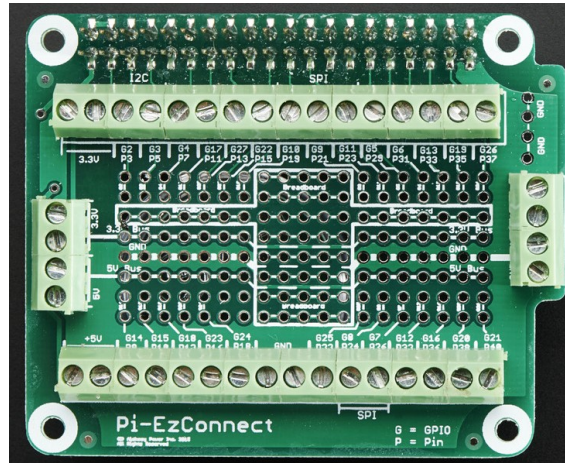


Figura 5.5: Pi-EzConnect (retirado de [3])

5.1.2 Ligações Exteriores

Foram identificadas as seguintes ligações exteriores:

- Uma ligação USB para o multisensor SensorTile.Box. Internamente, este módulo disponibiliza uma interface série virtual, que permite a leitura dos dados do sensor.
- Uma ligação USB para o medidor de consumo energético. O medidor energético possui uma interface RS485. Para isso, utilizou-se o Conversor USB RS485 para ambiente industrial da Waveshare[67].
- A comunicação com o autómato é realizada através da porta Ethernet do próprio Raspberry.
- A comunicação com a cloud é realizada através do WiFi do próprio Raspberry. Ou seja, o Raspberry foi ligado ao router da empresa com acesso à Internet. É possível utilizar um adaptador de rede Ethernet para USB, caso não exista cobertura WiFi no local de instalação, mas esteja disponível uma ligação por cabo.

Foram então criadas duas redes: uma rede local (utilizando a porta Ethernet do Raspberry Pi) e uma rede WiFi com acesso à Internet.

Na rede local está presente o PLC da máquina. Foi também necessário utilizar um switch na rede local, pois o PLC mantém comunicações com outros equipamentos, nomeadamente a consolas HMI e programadores. Fixou-se o IP da porta Ethernet do Raspberry, porque o switch

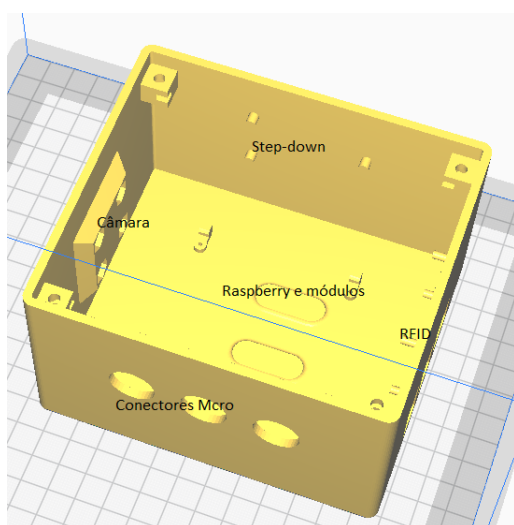
estabelece operações na camada 2 do modelo de OSI (Ligação de dados), isto é, não tem um servidor de DHCP.

Para evitar a aquisição de um serviço web ou a utilização de um IP público, o servidor cloud está alojado num segundo Raspberry Pi também conectado ao router da empresa. Desse modo, o servidor do Thingsboard está ligado, também via WiFi, ao router da empresa.

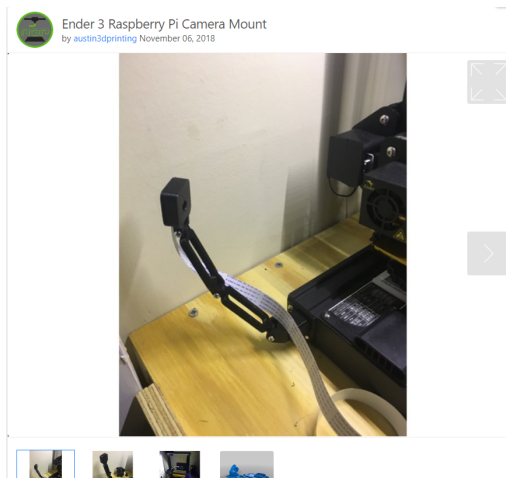
5.1.3 Encapsulamento

Tendo em conta que a junção destes módulos é única, não foi possível recorrer a nenhuma caixa *standard*. Assim sendo, desenvolveu-se uma caixa que garantisse o correto acondicionamento dos componentes utilizados e a robustez. Além disso, tiveram-se em conta aspetos relacionados com a ventilação, pois o Raspberry entrava facilmente em sobreaquecimento. Adicionaram-se dissipadores de calor no CPU, RAM, controlador USB e Ethernet do Raspberry, bem como uma ventoinha de 2x2cm.

Este encapsulamento foi desenvolvido no *SolidWorks* e construído numa impressora 3D. O suporte da câmara é proveniente do Thingsverse[68]. A Figura 5.6 mostra um esquema do encapsulamento, assim como o suporte de câmara utilizado.



(a) Interior desenhada



(b) Suporte da câmara (retirado de [68])

Figura 5.6: Caixa produzida em impressora 3D

Conforme já verificado, existe a necessidade de disponibilizar ao utilizador determinadas entradas e saídas. Desse modo, a caixa disponibiliza:

- Entrada de alimentação do dispositivo IoT;
- Entradas (digitais e analógicas) da Automation pHAT;
- Saídas da Automation pHAT;
- Pino 7 do Raspberry (Para permitir a ligação de sensores 1-Wire);

Utilizando 3 conectores micro GX12 (2,6 e 8 pinos) foi possível desenvolver o pinout apresentado na figura 5.7. Estes conectores permitem ao utilizador realizar a ligação de diversos componentes, tais como sensores analógicos e digitais, e atuadores. Permitem ainda alimentar o sistema.

ID	Forma do conector	Pinout	
1		1	Vin+
		2	GND
2		1	Vin+
		2	Input1
		3	Input2
		4	Input3
		5	Output1-
		6	Output2-
		7	Output3-
		8	GND
3		1	5V
		2	Vin+
		3	Vin-
		4	Pin 1-wire G4 (P7)
		5	Analog1
		6	Analog2

Figura 5.7: Pinout do dispositivo IoT

O conector com ID 1 é responsável por fazer a alimentação do sistema e suporta tensões entre 9 a 35 V DC.

O conector com ID 2 expõe as entradas e saídas do dispositivo IoT e replica a tensão de alimentação, bem como o GND. Importa salientar que as entradas e saídas são provenientes da carta Automation pHAT. Note-se que as saídas são do tipo "sinking", ou seja, NPN.

O conector com ID 3 replica novamente a tensão de alimentação e o GND. Também adiciona um terminal com 5 V, o pino 7 do Raspberry e duas entradas analógicas da Automation pHAT.

A figura 5.8 mostra as ligações realizadas na caixa.

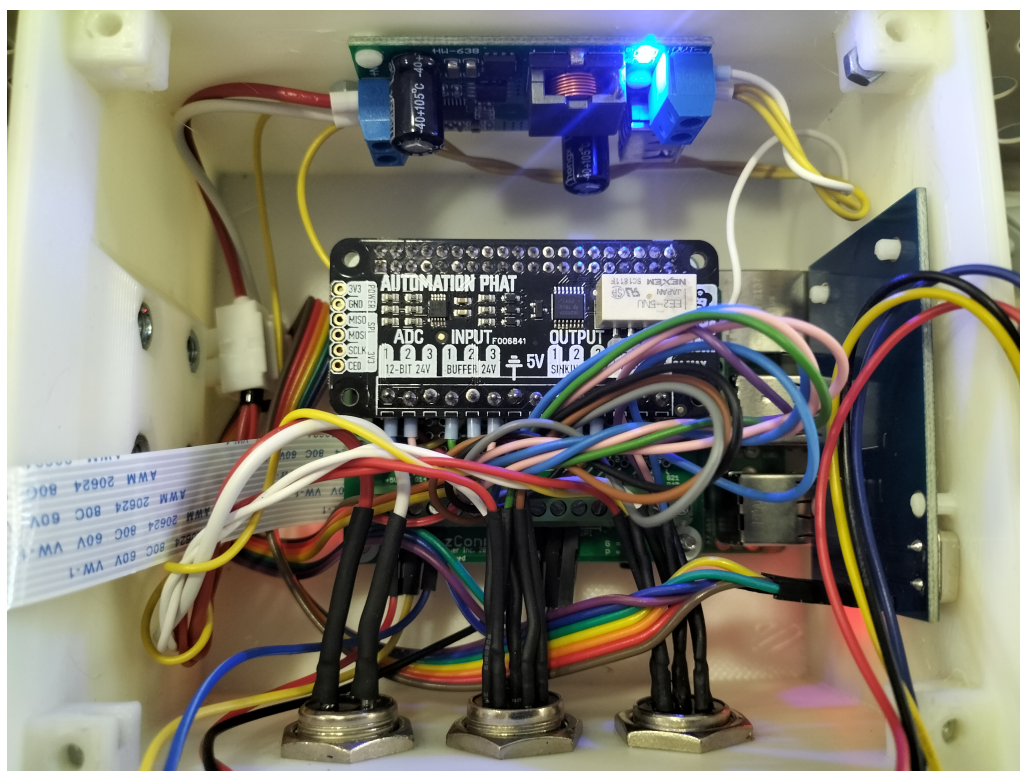


Figura 5.8: Ligações físicas no interior da caixa

5.2 Software

Conforme mencionado anteriormente, esta secção pode ser dividida em duas partes: o software desenvolvido para o Processamento Local e para o Processamento na cloud.

5.2.1 Processamento Local

Ao nível de Processamento Local, também podem ser definidas 3 grandes etapas: desenvolvimento do código para o sensor, desenvolvimento da interface para o HMI e desenvolvimento dos fluxos no Node-RED.

5.2.1.1 Sensor IoT

Relativamente à utilização do sensor, utilizou-se a aplicação Android "ST BLE Sensor"[69] para a configuração. Esta aplicação, disponível para Android e iOS, apresenta os dados obtidos pelo sensor recorrendo ao protocolo BlueST. Além disso, permite fazer um log dos dados recebidos, atualizar o firmware do sensor e escolher várias aplicações. Foram testadas várias, nomeadamente: "Vibration Monitor" e "Barometer". A primeira aplicação consistia em utilizar o acelerómetro, calcular a FFT e comparar com os dados armazenados no cartão de memória. Estes dados viriam de testes realizados anteriormente. A segunda aplicação consistia em devolver os

dados de temperatura, pressão e humidade detetados. A figura A.1 mostra screenshots das opções disponíveis.

No entanto, quando o sensor era conectado a um sistema externo (quer pela ligação USB, quer por Bluetooth), não era possível aceder aos dados. Testaram-se os scripts em Python disponíveis na própria página do fabricante[70], no entanto, era sempre devolvida a mensagem de “Ligação fechada de forma inesperada”. Este evento acontecia no método de estabelecer a ligação com o sensor.

Para resolver esta situação recorreu-se ao STM32CubeIDE[71], um ambiente de desenvolvimento para o STM32, baseado no Eclipse, que permite a escrita e compilação de código. Tomando como base o código de exemplo do "DataLogger", desenvolveu-se um programa capaz de estabelecer ligação através de uma porta série virtual e enviar objetos JSON com a temperatura, humidade e pressão e estado da vibração (detetada ou não detetada).

5.2.1.2 Interface da Consola HMI

Para integrar a consola HMI no projeto foi necessário proceder à conceção da sua interface. Para isso recorreu-se ao GUI do fabricante, o “Nextion Editor”[72]. Utilizaram-se blocos de botões, texto, texto deslizante e imagens. Através de uma vasta gama de comandos[73] foi possível controlar os valores apresentados, a visibilidade desses mesmos elementos e a alteração de páginas. O painel inicia com o logótipo da dissertação e, passados 5 segundos, disponibiliza a interface de utilizador. A interface de utilizador expõe, em vários painéis:

- Rolhas produzidas desde que o equipamento industrial foi ligado;
- Rolhas produzidas desde que foi efetuado *reset* ao equipamento industrial;
- Ordem de produção (proveniente da leitura do código QR);
- Número de rolhas restantes até à próxima revisão periódica;
- Número de horas de produção restantes até à próxima revisão periódica;
- Alertas de manutenção preditiva;
- Estado das entradas analógicas do dispositivo IoT;
- Estado das entradas e saídas digitais do dispositivo IoT;
- Dados do medidor de consumo energético (tensão, corrente, Potência Ativa, Potência Aparente, Fator de potência);
- Parâmetros (valores das constantes a ser atualmente utilizadas).

Além disso, existem dois conjuntos de páginas de manutenção: manutenção do cliente e do fabricante. O cliente pode eliminar os alertas preditivos do dispositivo IoT e sinalizar que ocorreu uma avaria. No caso da máquina de tratamento, o cliente pode fazer a manutenção às tubagens por

onde passa produto. O fabricante, além destas opções, pode confirmar a realização da manutenção periódica e indicar o início e fim de uma manutenção corretiva.

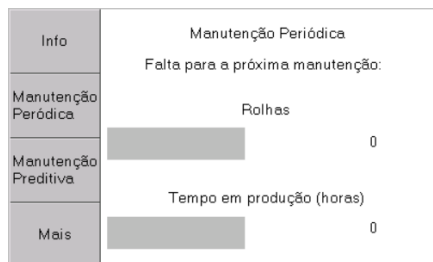
Na figura [5.9](#) mostram-se os painéis desenvolvidos.



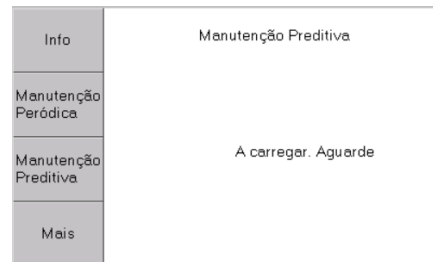
(a) Página de Introdução



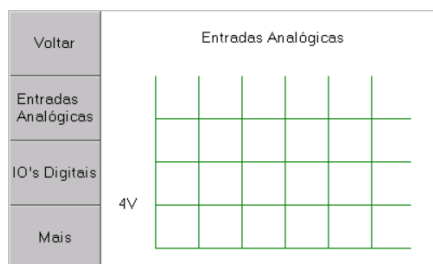
(b) Painel informativo



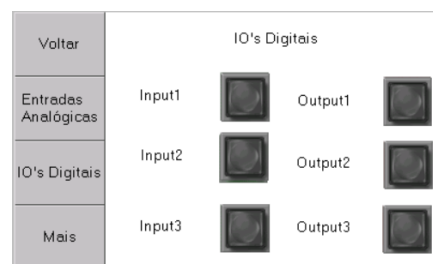
(c) Manutenção preventiva



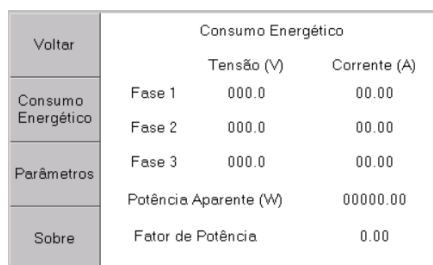
(d) Manutenção preditiva



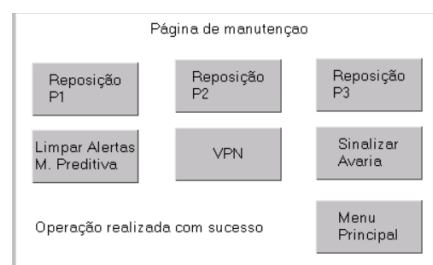
(e) Entradas analógicas



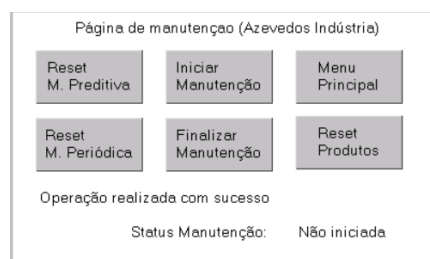
(f) Entradas e saídas digitais



(g) Consumo de energia



(h) Painel de manutenção do cliente



(i) Painel de manutenção do fabricante

Figura 5.9: Páginas da consola HMI

5.2.1.3 Node-RED

Seguindo a arquitetura mencionada na figura 4.2, foram criados 3 fluxos no ambiente de desenvolvimento Node-RED: Entradas e Saídas, Manutenção e Set de Regras. Utilizou-se o conceito de variáveis globais para permitir a partilha de dados entre os diversos módulos, assim como variáveis escritas em memória. Este facto tem uma elevada importância pois permite reter valores, mesmo que o sistema seja reiniciado. Para isso, foi necessário alterar as configurações do Node-RED[74].

O primeiro fluxo (Inputs&Outputs) é responsável por lidar com:

- Entradas e saídas da carta;
- Troca de dados com o autómato;
- Obter dados do sensor SensorTile.Box;
- Obter dados do medidor de consumo energético ORNO;
- Ler o código QR através da câmara;
- Disponibilizar um servidor OPC-UA;
- Enviar dados para o serviço de cloud.

Com o objetivo de facilitar o acesso às diversas componentes, este fluxo encontra-se agrupado em 2 duas colunas. A figura 5.10 mostra um esboço dessa mesma localização, assim como a figura A.2 apresenta o panorama do fluxo desenvolvido (screenshot).

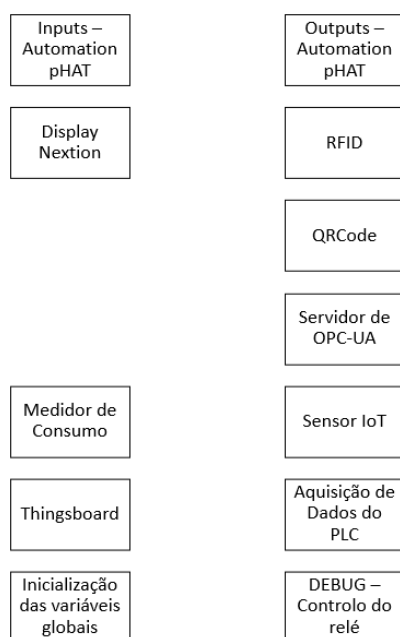


Figura 5.10: Esquema do fluxo de entradas e saídas

A primeira componente deste fluxo lida com as entradas e saídas da Automation pHat. O *node-red-contrib-automation-hat*[75] cria uma abstração sobre todas as comunicações utilizadas por esta carta, devolvendo uma mensagem cujo tópico é o nome da entrada (*input.1, ..., analog.3*) e o *payload* é o valor. Posteriormente a este nó, utilizaram-se blocos de “switch” para reencaminhar mensagens conforme o tópico e definir variáveis globais. A definição das saídas desta carta foi realizada através de uma função em Javascript. Também é disponibilizado um nó para lidar com as saídas. Para isso, envia-se uma mensagem cujo tópico seja o nome da saída (*output.1,...,output.3*) e o *payload* seja o estado que se pretende (*true* ou *false*). Para a implementação, faz-se periodicamente a leitura das variáveis globais relativos a cada atuador e envia-se essa informação para o nó. A figura 5.11 mostra o fluxo acima exposto.

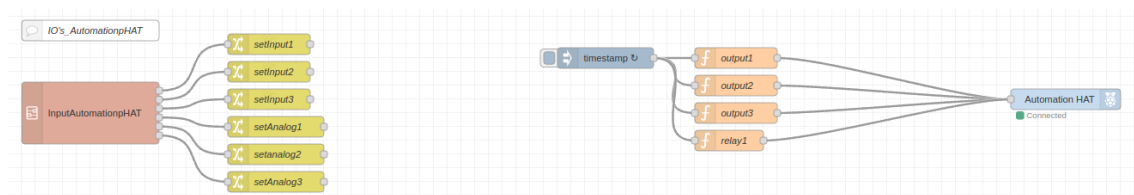


Figura 5.11: Parte de fluxo desenvolvido no Node-RED - IO's AutomationHAT

A consola HMI comunica através de uma interface série e tem ao seu dispor uma vasta gama de comandos. O fluxo desenvolvido tem 3 vertentes: conseguir aceder aos painéis de manutenção, enviar dados para a consola e receber ações provenientes dos botões na consola.

Sempre que é atuado algum cartão RFID válido, a consola passa para a respetiva página de manutenção. Além disso, os comandos provenientes dos botões pressionados pelos utilizadores são encaminhados para os fluxos de manutenção. Por último, sempre que é verificada uma alteração de página ou de forma periódica, é atualizada a informação presente através de um "handshake".

O "handshake" acima mencionado consiste em solicitar em que página é que a consola se encontra (através de o comando *sendme*) e, com base nessa resposta, encaminhar a respetiva informação. Há que salientar que, a qualquer comando enviado para a consola, é necessário adicionar 3 bytes 0xFF 0xFF 0xFF à trama.

A Figura 5.12 representa o fluxo desenvolvido, bem com o *handshake* criado entre a consola e o Processamento Local.

O fluxo responsável pelo RFID tem um funcionamento simples que se encontra ilustrado no diagrama da Figura 5.13. Para isso recorre ao nó presente em *node-red-contrib-mfrc522-rfid*[76]. Este nó faz a leitura de uma tag RFID e devolve o UID (Unique ID) e pode, opcionalmente, devolver blocos de dados.

O fluxo responsável pela leitura do código QR tem um funcionamento mais elaborado e assenta em 2 passos fundamentais. A primeira etapa é recolher a imagem proveniente da câmara. Para isso, invoca o comando *libcamera-still* que, através da câmara do Raspberry Pi, obtém uma imagem e guarda-a num ficheiro. De seguida, esse ficheiro é aberto e é chamado um *script* em Python que, através dos métodos provenientes de *pyzbar*[77] e *PIL*[78], realiza todo o processamento

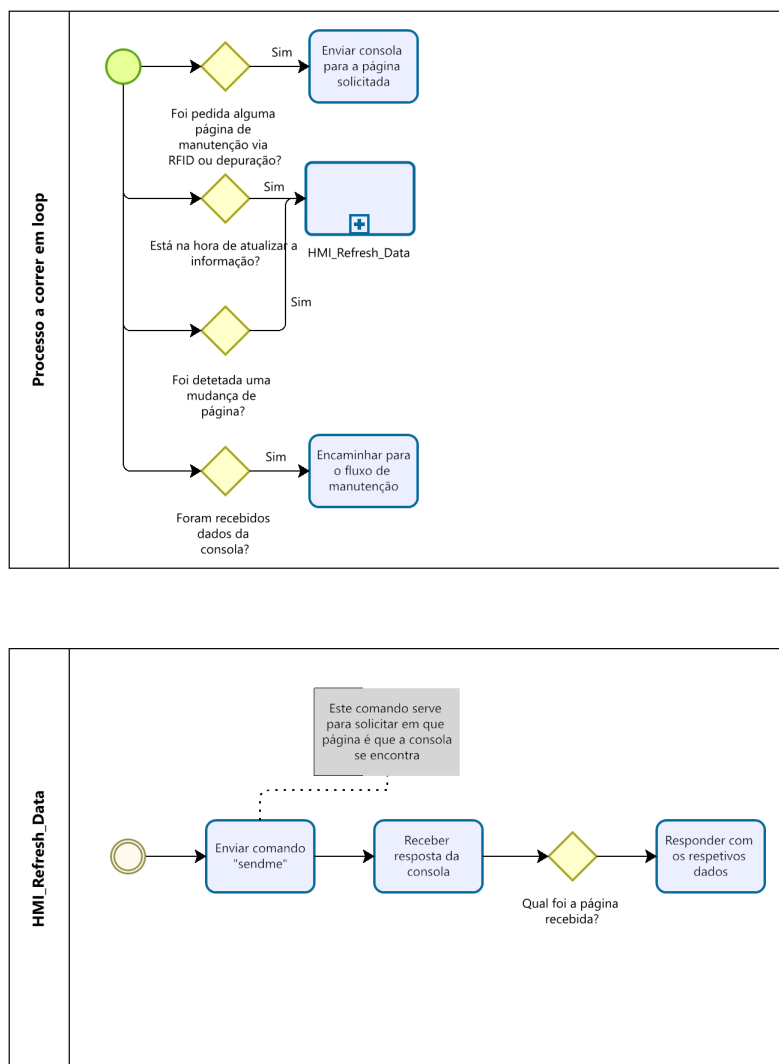


Figura 5.12: Comunicação com a consola HMI

de imagem. Devolve, em string, a informação proveniente do código QR. Em substituição do *script* pode ser invocado o nó *qrdecode* proveniente de *node-red-contrib-qrdecode*[79]. Este nó consegue ler um código QR presente numa imagem. No entanto, apesar de ser mais eficaz, não é recomendado pois compromete bastante a fluidez de todo o sistema.

Para tornar este equipamento ainda mais versátil foi instalado um servidor de OPC-UA recorrendo ao nó *node-red-contrib-opcua-server*[80]. Este nó permite implementar um servidor OPC-UA utilizando menos dependências. Assim, os restantes equipamentos industriais podem ler as variáveis disponibilizadas por este servidor. Na Figura A.3 encontra-se um *screenshot* do UAExpert (Aplicação que atua como cliente de OPC-UA) com as variáveis disponibilizadas pelo sistema IoT.

Para a aquisição de dados do medidor de energia optou-se pela utilização de um script em Python pois foram detetadas dificuldades em utilizar os nós de MODBUS do Node-RED. O script

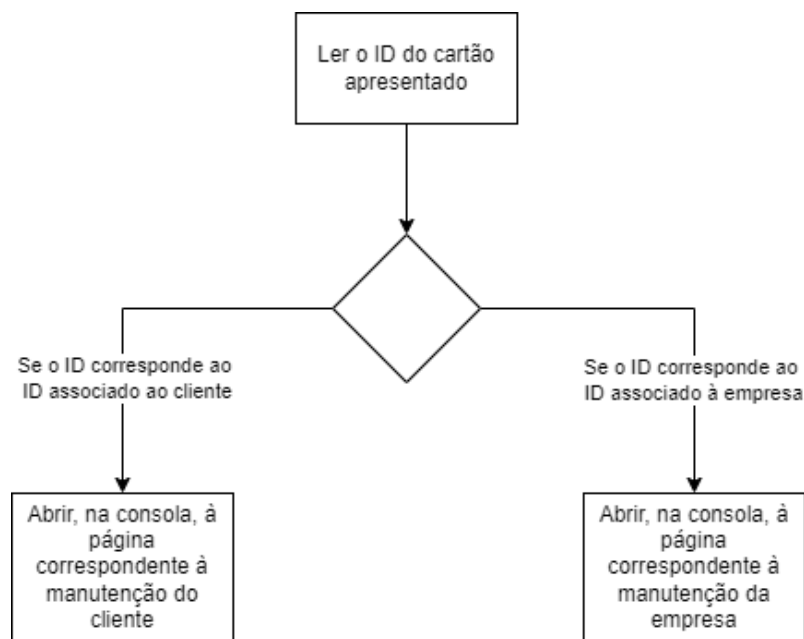


Figura 5.13: Utilização do RFID

utiliza os métodos provenientes do *pymodbus* e devolve um objeto JSON com os vários campos lidos do medidor. Em caso de falha na leitura desse campo, é devolvido o valor -1.

O sensor IoT disponibiliza uma porta série virtual com uma baudrate de 230400 bits/s. Conforme visto na secção 5.2.1.1, o sensor envia todos os dados no formato JSON. Assim sendo, o processamento local só deve guardar os dados provenientes do sensor como variáveis globais e invocar as respetivas funções de manutenção.

Restam ainda duas etapas fundamentais: recolher os dados do PLC e enviar dados para a cloud.

A recolha de dados do autómato é feita através do protocolo FINS. Como tal, recorreu-se aos nós disponibilizados em *node-red-contrib-omron-fins*[81]. Este nó implementa as funções de leitura e escrita do protocolo FINS via Ethernet. No entanto, foi necessário configurar este nó. A Tabela 5.1 mostra os valores utilizados:

Tabela 5.1: Parâmetros de configuração do protocolo FINS

Parâmetro	Valor	Significado
DNA	0	Como os equipamentos estão ligados na mesma rede, o DNA tem valor 0. Caso contrário, teria valor 1
DA1	1	Último octeto do endereço IP do PLC
SA1	10	Último octeto do endereço IP fixo da porta Ethernet do Raspberry Pi
ICF	128	Information Control Field. Tem com valor padrão 128

Foi desenvolvido um subfluxo responsável por aceder aos registos do autómato, processar a informação e guardar os resultados em variáveis globais. São obtidos contadores acumulados dos tempos em que o equipamento industrial esteve:

- Ligado;
- Em produção;
- Em alarme;

É ainda obtido o número de série do equipamento industrial. No caso específico da máquina de tratamento, obtém-se também as quantidades de produtos injetados e a quantidade de produto presente no equipamento.

Estando todos os dados recolhidos, é possível estabelecer a ligação com o serviço de cloud. Para isso recorreu-se ao *Thingsboard* e ao protocolo MQTT. Importa aqui salientar que, caso fosse alterado o serviço cloud, bastaria alterar a configuração do nó de MQTT. Para a utilização deste protocolo é necessário ter:

- O endereço IP e a porta associada ao servidor
- Um tópico a publicar
- Um *username* e *password* (Dados facultativos, no entanto, serve para garantir que só os utilizadores autorizados publicam dados)

No caso do Thingsboard, o tópico associado à telemetria é *v1/devices/me/telemetry* e aos atributos é *v1/devices/me/attributes*. O username é o token de acesso do dispositivo e a palavra-passe é um campo nulo[82].

O segundo fluxo desenvolvido no Node-RED é o responsável por lidar com todos os aspetos relacionados com a manutenção. As Figuras 5.14 e 5.15 mostram os diagramas da estrutura desenvolvida e na figura A.4 encontra-se um screenshot do próprio fluxo.

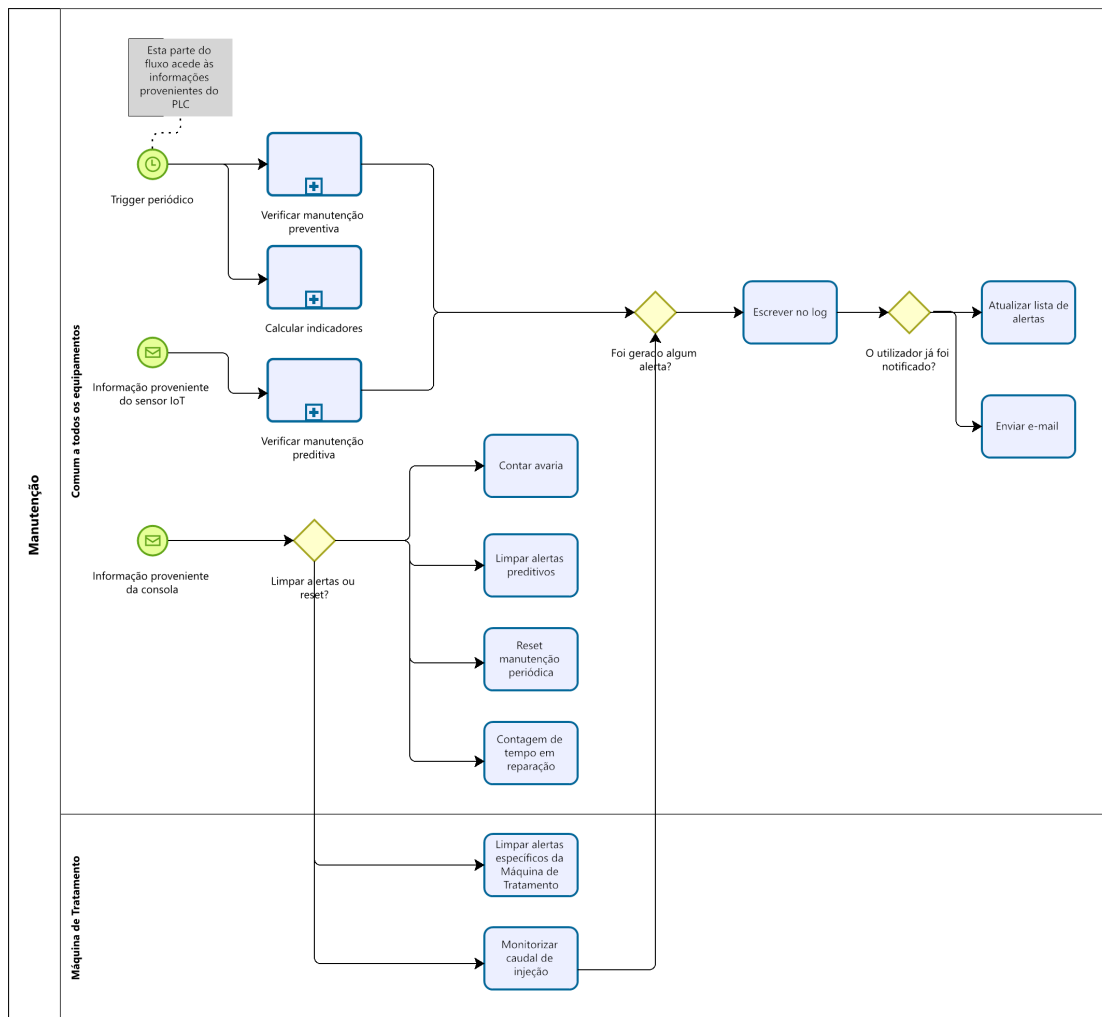


Figura 5.14: Visão global do fluxo de Manutenção

Sempre que é recebida informação proveniente do autómato ou do sensor IoT são verificadas as condições de manutenção preventiva e preditiva. Caso algum alerta seja criado, é enviado automaticamente um e-mail a informar o cliente. Este serviço é realizado através do nó *node-red-node-email*[83] que permite enviar e ler email's simples. É necessário realizar login com a conta por onde o e-mail vai ser enviado. Neste caso, utilizou-se uma conta Gmail, onde foi necessário criar uma "App Password"[84]. Também é necessário preencher o e-mail do destinatário.

No caso da manutenção preventiva, esse e-mail é só enviado uma vez. Na manutenção preditiva, caso o sistema seja reiniciado e ainda existam alertas, o e-mail é enviado novamente. Não obstante, todos os alertas preditivos são escritos num log, em formato CSV, juntamente com um timestamp e os alertas de manutenção preditiva escritos num ficheiro de texto.

Está ainda implementado um temporizador relativo ao tempo passado em manutenção programada. O início e fim da manutenção só pode ser controlado pelo painel de manutenção do fabricante. A componente referente ao *reset* é responsável por atualizar as variáveis globais retentivas e limpar os alertas, caso seja esse o caso. A operação é sempre escrita num log guardado no

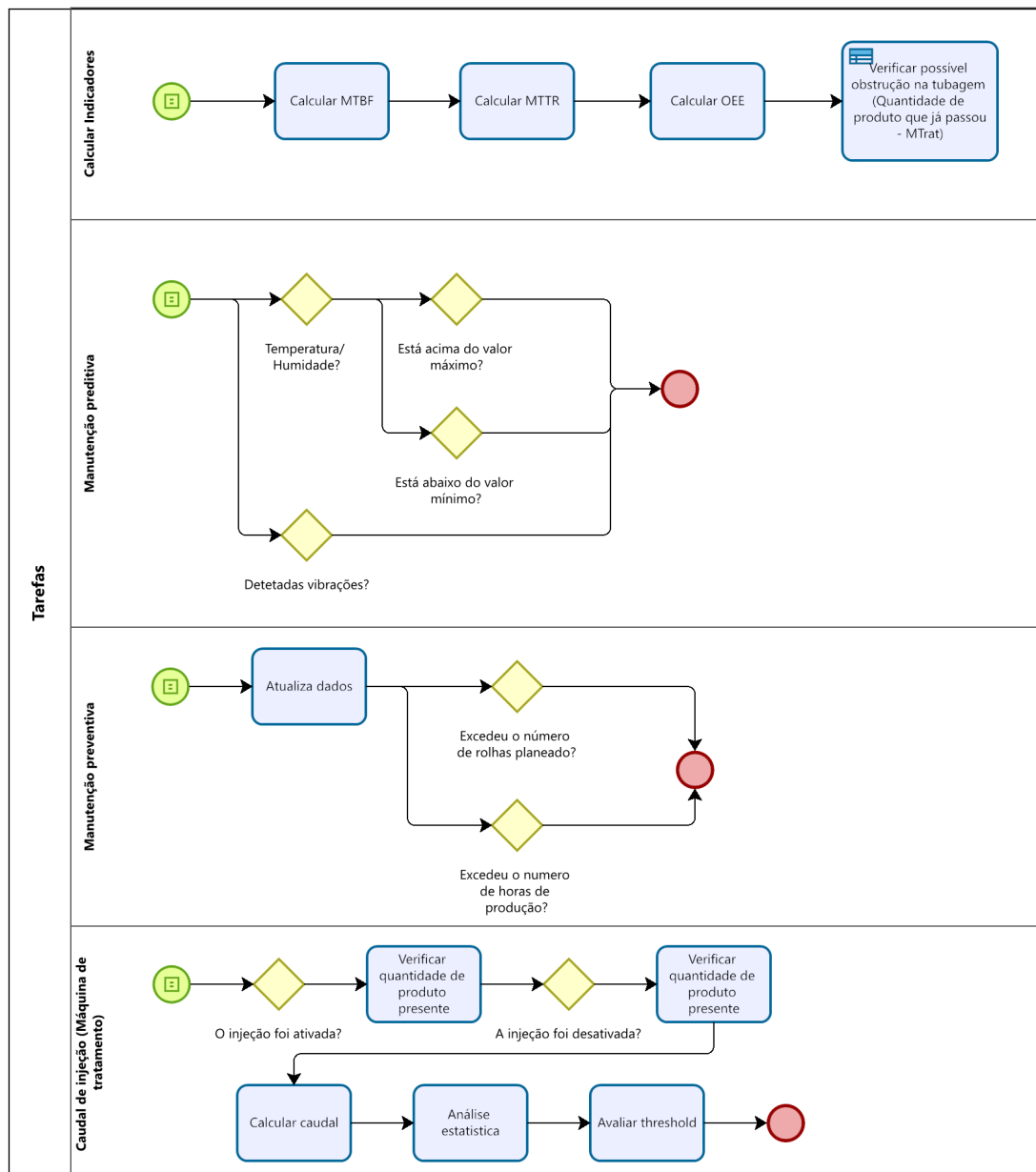


Figura 5.15: Tarefas do fluxo de Manutenção

dispositivo IoT.

Por último, existem os blocos específicos associados à máquina de tratamento. Existe um subfluxo que faz a gestão das variáveis de manutenção específicas. É responsável por lidar com as ações provenientes dos botões da consola e realizar a ação pretendida. Existe ainda um segundo fluxo responsável pela procura do declínio do caudal da tubagem de injeção de produto. Este subfluxo calcula o caudal em causa e, utilizando uma janela de 10 amostras, calcula:

$$0.6 * caudal + 0.3 * caudalmedio + 0.1 * caudalminimo \quad (5.1)$$

Se este valor passar abaixo de uma constante definida, é gerado um alarme.

O último fluxo é responsável pelo conjunto de regras definido por cada utilizador. É idealizado conforme as necessidades de cada cliente final e pode recorrer às entradas e saídas da carta *Automation pHat*. Pode ainda ler e escrever em todas as variáveis globais do programa. Encontra-se desenvolvido um fluxo exemplo que, periodicamente, verifica se os alertas de vibrações, temperatura ambiente e humidade estão ativos. Caso o alerta de vibrações esteja ativo, liga a saída 1. Caso contrário, desativa. A mesma ideia é aplicada para os restantes alertas e saídas. A Figura 5.16 mostra o fluxo desenvolvido.

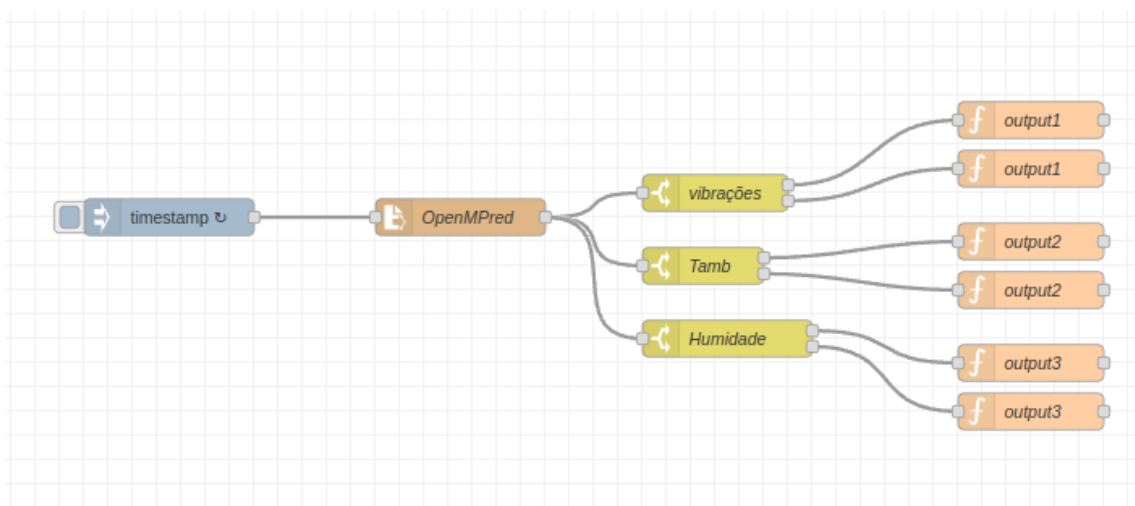


Figura 5.16: Exemplo de um Set de Regras

5.2.1.4 Serviço VPN

Para ser possível prestar assistência remota ao dispositivo IoT, utilizou-se o HusarNET. Contrariamente ao uma VPN tradicional, os dispositivos são associados a uma rede virtual gerida por um dashboard. Encontra-se na figura 5.17 os dois dispositivos associados: O dispositivo IoT da máquina de tratamento e um computador. O cliente pode ainda ativar ou desativar este serviço na sua página de manutenção.

A Figura 5.18 mostra o acesso por SSH (Secure Shell) e por VNC (Virtual Network Computing).

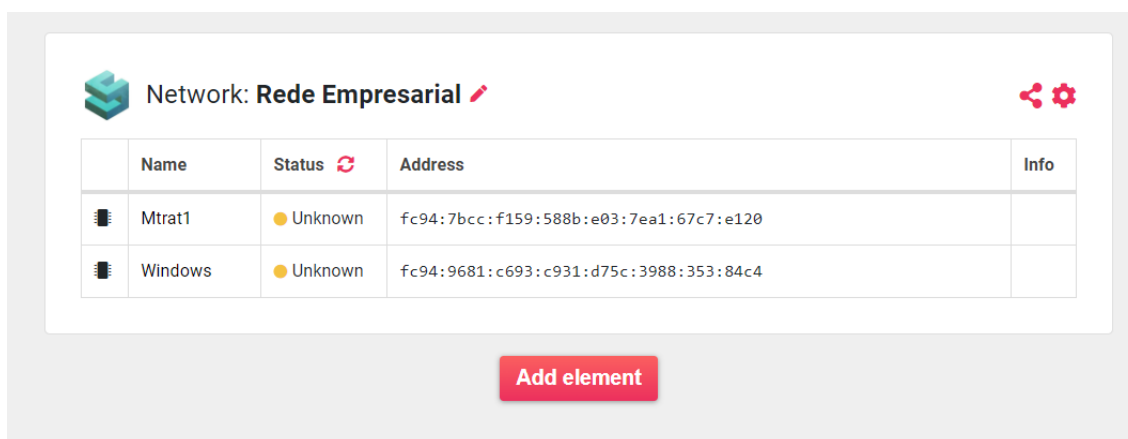


Figura 5.17: Dashboard do HusarNET

```
C:\Users\Utilizador>ping -6 Mtrat1

Pinging mtrat1 [fc94:7bcc:f159:588b:e03:7ea1:67c7:e120] with 32 bytes of data:
Reply from fc94:7bcc:f159:588b:e03:7ea1:67c7:e120: time=117ms
Reply from fc94:7bcc:f159:588b:e03:7ea1:67c7:e120: time=12ms
Reply from fc94:7bcc:f159:588b:e03:7ea1:67c7:e120: time=103ms
Reply from fc94:7bcc:f159:588b:e03:7ea1:67c7:e120: time=8ms

Ping statistics for fc94:7bcc:f159:588b:e03:7ea1:67c7:e120:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 117ms, Average = 60ms

C:\Users\Utilizador>ssh pi@mtrat1
The authenticity of host 'mtrat1 (fc94:7bcc:f159:588b:e03:7ea1:67c7:e120)' can't be established.
ECDSA key fingerprint is SHA256:ku+XKcS/i67NjUecsgwGyFbBH4xD7UxuIW6qmyI/k.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'mtrat1,fc94:7bcc:f159:588b:e03:7ea1:67c7:e120' (ECDSA) to the list of known hosts.
pi@mtrat1's password:
Linux Mtrat1 5.15.32-v8+ #1538 SMP PREEMPT Thu Mar 31 19:40:39 BST 2022 aarch64

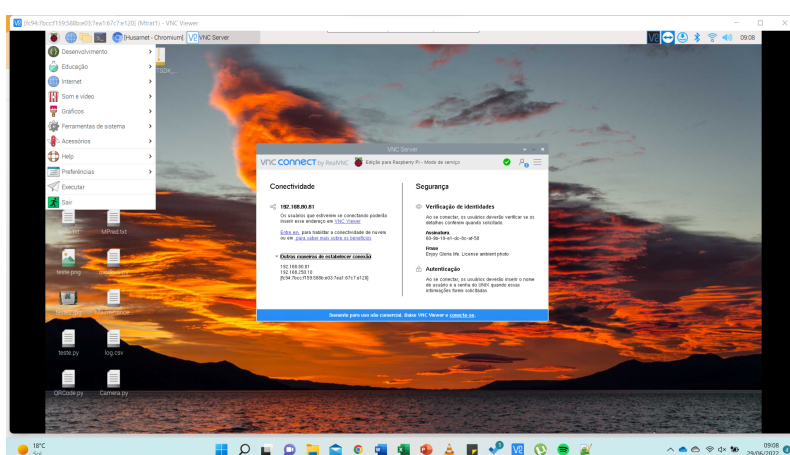
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 29 08:54:47 2022

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@mtrat1:~$
```

(a) Ping e acesso por SSH



(b) Acesso ao ambiente de trabalho por VNC Viewer

Figura 5.18: Acesso ao dispositivo IoT recorrendo à VPN

5.2.2 Processamento na *cloud*

Conforme já mencionado, o servidor *Thingsboard* encontra-se instalado num Raspberry Pi 3 conectado à rede wireless da empresa. Para isso, foi necessário configurar o Raspberry Pi para o efeito[85]. Posto isto, é possível desenvolver uma cadeia de regras que garantisse a verificação dos aspetos definidos para a máquina de tratamento. Nesta parte encontram-se as regras mais complexas, isto é, que não consistem na observação de uma só variável. São também definidas as ações realizadas com os atributos e telemetria recebidos do dispositivo de processamento local [86]. Os alertas aqui definidos foram:

- Inspeção preditiva (quando o tempo desde a última inspeção ultrapassa 70% do MTBF);
- Possível obstrução nas tubagens (quando o subfluxo responsável por esta gestão gera o alarme. Específico da máquina de tratamento)
- Analisar canalização das tubagens (quando já passou uma quantidade de produto superior à definida por uma constante do dispositivo IoT. Específico da máquina de tratamento)

Importa salientar que foi utilizado o conceito de *Server Attributes*, isto é, são valores que, em vez de serem valores fixos guardados na plataforma cloud, são provenientes de cada dispositivo IoT. Este conceito foi utilizado essencialmente na definição de constantes. Isto permite garantir uma arquitetura genérica que se adapta aos parâmetros definidos pelo cliente.

Posteriormente, foi criado o *device profile*, que integra o set de regras criado anteriormente. Aqui encontram-se as regras que assentam na verificação de uma só variável proveniente da telemetria. Estão então criadas as regras:

- Temperatura ambiente baixa
- Temperatura ambiente elevada
- Humidade elevada
- Humidade baixa
- OEE em nível baixo
- Foi atingido o limite para a revisão periódica
- Detetadas vibrações elevadas

Este perfil foi adicionado a um dispositivo e associado a diversos dahsboards:

- Home Dashboard – Aplicado exclusivamente ao cliente. Centraliza todos alertas gerados por todos os equipamentos a ele associados. É afixado na página inicial.
- Dashboard Geral – Responsável por expor ao cliente os dados comuns a todos os equipamentos industriais, tais como: Rolhas e horas restantes, OEE, Tempo médio entre falhas

(MTBF), Potência aparente, Potência consumida, Tensão, Corrente, Fator de Potência, Ordem de produção, Alarmes, humidade e temperatura ambiente, histórico de produção de rolhas (diário) e estado das saídas digitais do equipamento IoT.

- **Máquinas de Tratamento:** Este dashboard expõe ao utilizador dados específicos referentes à máquina de tratamento. Inclui-se aqui: Quantidade de produto (P1, P2 e P3) presente no equipamento e consumida (Valores diários e totais).
- **Azevedos Dashboard** – Este dashboard é para uso exclusivo do tenants do Thingsboard. Sinaliza todos os alertas de manutenção periódica que ocorram nos dispositivos a ele associados. Tem como objetivo monitorizar, além do cliente, o estado da manutenção preventiva.
- **Tracking de Ordens** – Este dashboard deve ser dimensionado para cada cliente de forma individual. Mostra uma representação do chão de fábrica e, passando com o cursor sobre o pin associado a cada equipamento industrial, mostra alguns dados de relevo do equipamento em questão. É também apresentada uma tabela com cada equipamento industrial associado e a ordem de produção correspondente.

As Figuras 5.19, 5.20 e 5.21 mostram alguns dos dashboards acima mencionados.

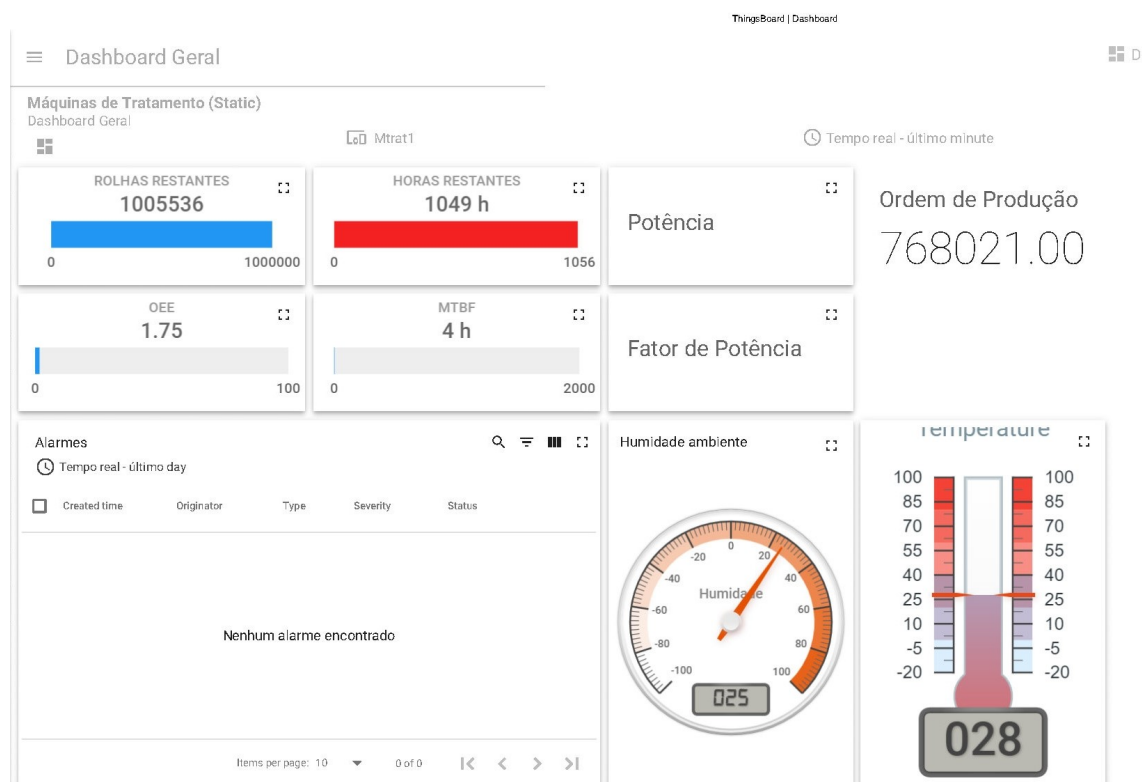
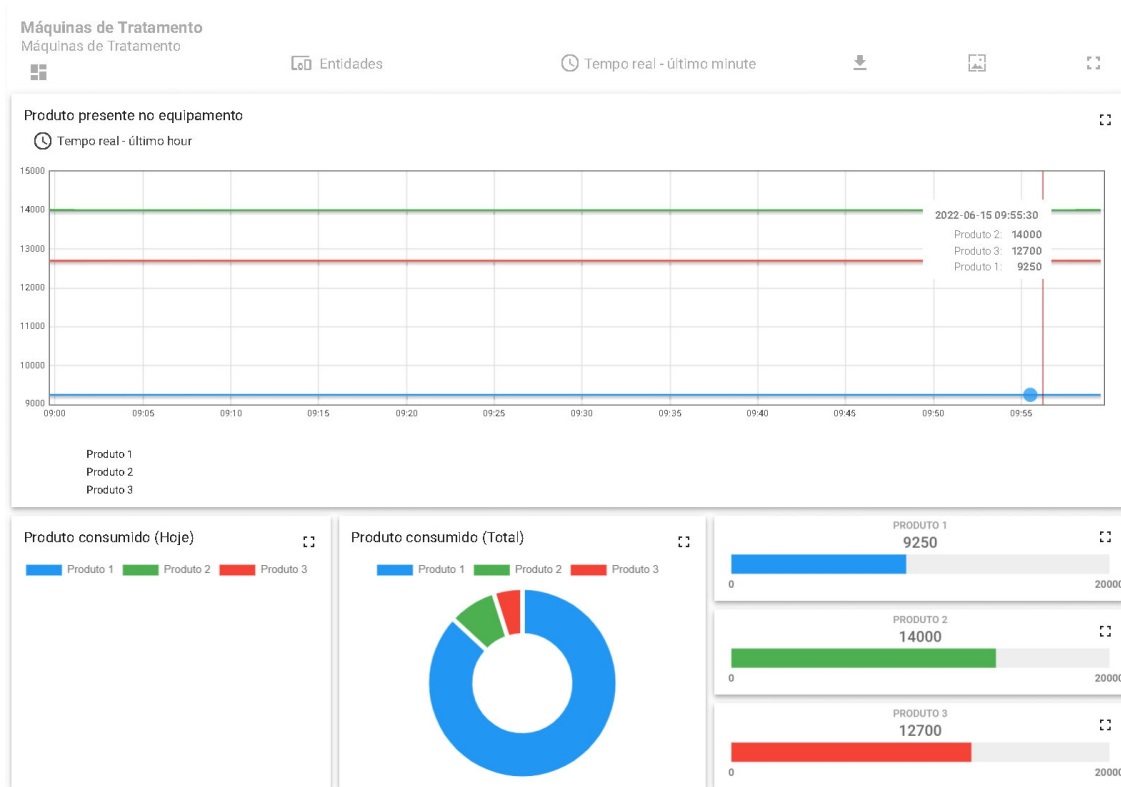


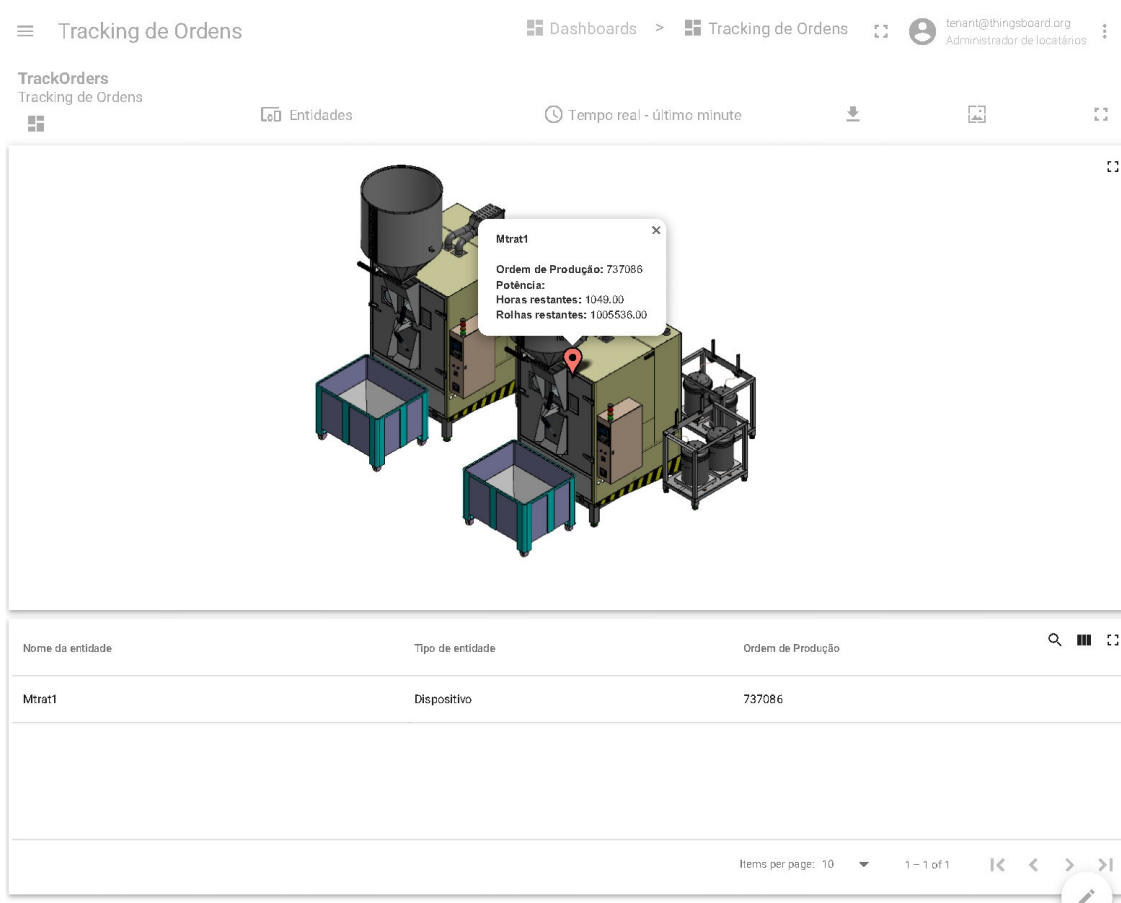
Figura 5.19: Dashboard Geral do Thingsboard - Parte 1



Figura 5.20: Dashboard Geral do Thingsboard - Parte 2



(a) Máquina de Tratamento



(b) Tracking de Ordens

Figura 5.21: Dashboards desenvolvidos no Thingsboard

Este serviço também disponibiliza uma API que permite partilha dados com aplicações de terceiros[87]. Para isso desenvolveu-se uma aplicação que, recorrendo ao protocolo HTTP, solicita um token de acesso e, utilizando esse mesmo token, obtém a última telemetria associada a um dado campo. Para provar este conceito, desenvolveu-se um script em Python capaz de criar um gráfico com a corrente e tensão do equipamento industrial[88]. Para isso, realiza os passos acima mencionados. Na figura 5.22 encontra-se um “screenshot” dos gráficos apresentados. Salienta-se que não estava nenhuma carga associada ao medidor de energia.

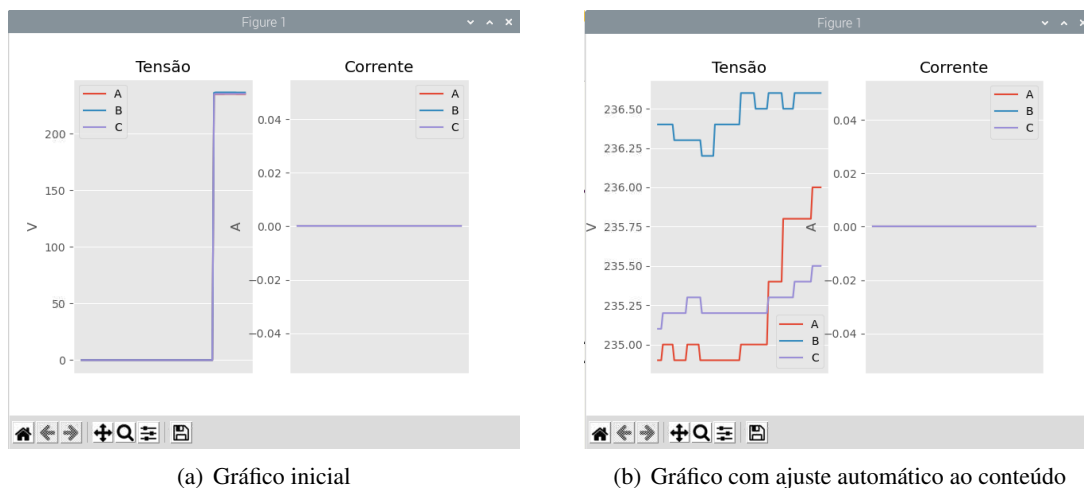


Figura 5.22: Gráficos produzido pelo script em Python

5.3 Análise Crítica

Comparativamente ao inicialmente planeado, não foi possível implementar o sistema IoT a 3 máquinas industriais, sendo apenas possível implementar para a máquina de tratamento. No entanto, manteve-se a filosofia inicialmente delineada de uma arquitetura abrangente. Desse modo, basta alterar o bloco de aquisição de dados do autômato (essencialmente as posições de memória onde estão escritas as variáveis) para que a arquitetura comum do sistema esteja em funcionamento. Além disso, faltaria o desenvolvimento dos blocos associados à problemática de cada equipamento. Os mesmos não foram desenvolvidos, pois, desse modo, não existiria nenhuma forma de validar a implementação realizada.

Destaca-se ainda o desenho e produção de uma caixa própria para o dispositivo desenvolvido, assim como o uso de uma VPN para prestar assistência remota ao dispositivo.

Estando a fase de implementação concluída, é necessário fazer a validação da solução proposta, tal como apresentado no capítulo [6](#).

Capítulo 6

Validação da solução proposta

Estando a implementação da arquitetura proposta já concluída, é necessário verificar se a mesma cumpre os objetivos definidos. Assim sendo, este capítulo discute os testes a realizar ao sistema, apresentando também ainda imagens do sistema implementado.

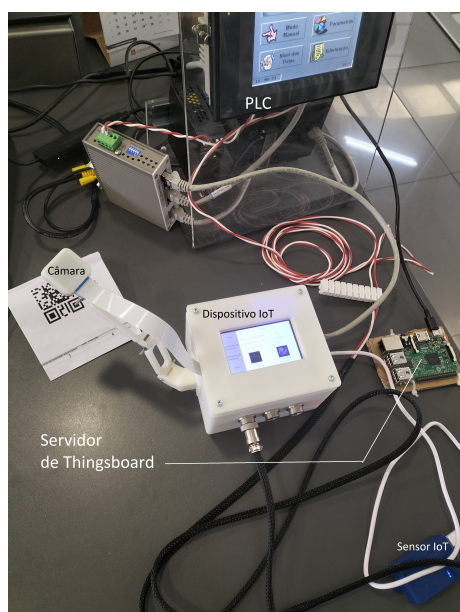
6.1 Ambiente de testes

Grande parte da validação do sistema foi realizada recorrendo a um PLC e ao dispositivo IoT desenvolvido. Adicionou-se ainda uma consola da máquina, um switch e um computador. A Figura 6.1 mostra a montagem realizada.

No computador estava instalado o programa "CX-Programmer", software de programação de autómatos da OMRON. Além disso, estava visível a página com o editor de fluxo do Node-RED, recorrendo ao VNC Viewer. Esta página apresentava algumas mensagens de *debug* para permitir a deteção de falhas.

Os testes basearam-se na manipularam-se diversos contadores e saídas do autómato. De seguida, verificou-se a alteração de variáveis e a criação de alarmes por parte do dispositivo IoT. Também se verificou a receção de *emails* com esses mesmos alertas. Conforme destacado no capítulo 5, só foi possível implementar este sistema para a máquina de tratamento devido à escassez de componentes eletrónicos verificada atualmente.

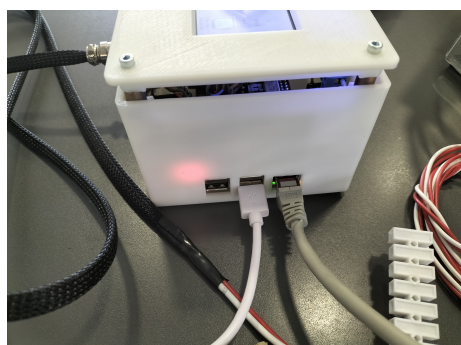
Importa mencionar que, apesar de na figura 6.1 não estar presente o medidor de consumo energético, o mesmo também foi testado com as restantes componentes. No entanto, alguns testes não o utilizaram para simplificar as ligações. Na Figura 6.2 encontra-se a montagem integrando este componente.



(a) Montagem Geral



(b) Interface GX12



(c) Interface USB e Ethernet



(d) Consola HMI

Figura 6.1: Validação do Sistema

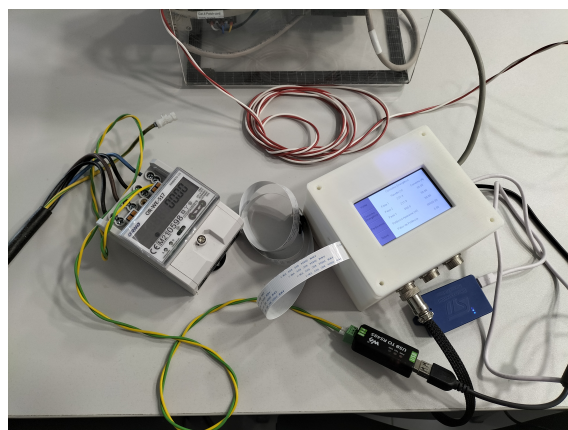


Figura 6.2: Montagem com o medidor de consumo energético

6.2 Validação de requisitos

Tomando como base a Tabela 2.1 é possível delinear testes que comprovem o cumprimento ou não de um determinado requisito. Desse modo, as Tabelas 6.1, 6.2 e 6.3 mostram esses mesmos testes, bem como os seus resultados. Os resultados são expostos como "OK" em caso de sucesso, ou "NOK" em caso de falha.

Tabela 6.1: Validação de Requisitos - Sistema como um todo

	Requisito	Teste	Estado
Sistema como um todo	T1.1: Analisar diversos aspetos relacionados com a manutenção preventiva e preditiva de cada equipamento industrial	Analisar o número de horas de produção (entre revisões) for excedido	OK
		Analisar o número de rolhas (entre revisões) for excedido	OK
		Analisar o espetro de vibrações	OK
		Analisar a temperatura e humidade	OK
	T1.2: Calcular diversos indicadores	Calcular MTBF	OK
		Calcular MTTR	OK
		Calcular OEE	OK
	T1.3: Monitorizar o consumo energético, temperatura e humidade	Monitorizar o consumo energético, temperatura e humidade	OK
	T1.4: Gerar um alerta sempre que algum aspeto não seja verificado	Verificar se o alerta é gerado	OK
	T1.5: Informar o utilizador sempre que for gerado um novo alerta	Verificar se o alerta é gerado na cloud e se o utilizador recebe um e-mail de aviso	OK
	T1.6: Realizar tracking de ordens, provenientes de código QR ou de barras	Aproximar um código QR junto da câmara e verificar se o mesmo aparece na consola HMI e na plataforma cloud	OK
	T1.7: [Máquina de capsular] Gerar um alerta quando a taxa de rejeição for muito elevada		NOK
	T1.8: [Balança] Procurar tendências nas pesagens de cada célula		NOK
	T1.9: [Máquina de tratamento] Monitorizar o caudal de um determinado produto (g/s), sinalizando os declínios pois pode existir alguma obstrução nos tubos.	Verificar se o sistema analisa o estado do atuador de injeção, calcula o fluxo e faz a respetiva análise	OK

Tabela 6.2: Validação de Requisitos - Dispositivo IoT

	Requisito	Teste	Estado
Dispositivo IoT	T2.1: Recolher dados de diversas fontes, tais como PLC's, sensores externos, etc	O sistema recolhe dados do PLC	OK
		O sistema recolhe dados do sensor IoT	OK
		O sistema recolhe dados da carta de entradas e saídas	OK
		O sistema recolhe dados do medidor de consumo energético	OK
	T2.3: Comunicar com o sistema cloud	Comunicar com o sistema cloud	OK
	T2.5: O utilizador deve conseguir criar o seu próprio Set de Regras e adicionar funcionalidades extra	Existe espaço dedicado para o utilizador implementar o set de regras	OK
	T2.7: Permitir o acesso remoto através de uma VPN		OK
	T2.9: Disponibilizar um servidor OPC-UA com variáveis de relevo		OK
	T2.2: Ter hardware e encapsulamento robustos para serem inseridos no chão de fábrica	Utilizar um encapsulamento comercial ou projetar um encapsulamento próprio que garanta a robustez ao sistema	OK
	T2.4: Ser configurável	Utilizar hardware e ambiente de desenvolvimento que permitam a reconfiguração do sistema	OK
	T2.6: Ser modular	Desenvolver um sistema com diversos módulos, onde os mesmos podem ser substituídos com facilidade	OK
	T2.8: Possuir entradas e saídas de gama industrial	Possuir entradas e saídas de 24V	OK
	T2.10: Possuir uma interface responsável por expor dados de relevo ao utilizador	A consola HMI deve expor métricas e alertas	OK
	T2.11: A interface deve apresentar painéis próprios, dedicados aos aspetos de manutenção, que só podem ser acedidos por utilizadores autenticado	Através da autenticação por RFID, devem ser apresentados painéis próprios	OK

Tabela 6.3: Validação de Requisitos - Sistema cloud

	Requisito	Teste	Estado
Sistema cloud	T3.1: Permitir adicionar clientes	Adicionar clientes	OK
	T3.2: Permitir adicionar novos dispositivos	Adicionar novos dispositivos	OK
	T3.3: Os dados apresentados a cada cliente devem poder ser personalizados	Permitir que o mesmo dashboard é partilhável entre utilizadores, mas só permite o acesso aos dados dos equipamentos a ele associados	OK
	T3.4: Permitir associação entre dispositivos e clientes	Associar dispositivos a clientes	OK
	T3.5: Calcular métricas de desempenho do equipamento industrial	Expor ao utilizador a disponibilidade e desempenho do equipamento	OK
	T3.6: Sinalizar alarmes específicos que ocorram com mais regularidade do que a prevista	Ter pelo menos uma regra que só é acionada ao fim de um determinado número de ocorrências	OK
	T3.7: Apresentar indicadores de performance e manutenção	Expor ao utilizador o OEE, MTBF e MTTR	OK
	T3.8: Analisar as tendências dos dados disponíveis		NOK
	T3.9: Permitir exportação de dados para aplicações externas (por exemplo, desenvolvidas pelo cliente)	Desenvolver uma aplicação para obtenção de telemetria enviada para o sistema cloud	OK

É possível concluir que, na componente do dispositivo IoT e do sistema como um todo, todos os requisitos foram cumpridos à exceção dos requisitos associados a balança e à máquina de capsular.

Os testes no sistema cloud basearam-se na observação dos dados expostos nos dashboards associados a uma conta cliente onde a máquina de tratamento está associada. Todos os requisitos associados a esta componente foram cumpridos à exceção da previsão de dados (T3.8). Esta questão deve-se, essencialmente, à versão de produto de cloud utilizado.

Capítulo 7

Conclusões e Trabalho Futuro

O mundo tende cada vez mais a utilizar sistemas inteligentes e a centralizar a informação. Os conceitos de IoT e o IIoT tornam-se, tendencialmente, mais presentes. No entanto, verifica-se que os equipamentos industriais produzidos pela empresa proponente ainda não integravam nenhuma opção de conectividade com outros equipamentos, de centralização de informação, nem de monitorização preventiva e preditiva.

Desse modo, surgiu a necessidade de um dispositivo IoT que, quando conectado com um equipamento industrial, calcule diversas métricas relacionadas, tais como consumo energético e KPI's. Recorrendo a sensores externos, deve analisar aspetos preditivos. Além de recolher informação, deve também ser capaz de publicar dados num sistema *cloud*. Esse sistema cloud deve ser responsável por armazenar, tratar e apresentar dados ao utilizador. Deve garantir a privacidade dos dados e permitir que a informação seja exposta de forma personalizada.

Para a resolução desta questão seguiram-se duas abordagens: a busca de soluções comerciais que pudessem solucionar e o levantamento de componentes necessários para a construção de uma solução de raiz. Foi ainda necessário abordar aspetos teóricos relacionados com a manutenção, estrutura típicos dos sistemas IoT e protocolos de comunicações.

Para a construção de uma solução de raiz, considerou-se o Raspberry Pi, a Automation pHAT, um módulo de RFID RC522, um step down FAL08023, uma consola HMI Nextion NX4024T032 e uma Raspberry Pi Camera Module 2. Para a fixação mecânica do equipamento foi desenvolvido um encapsulamento próprio. A nível de software produzido para o dispositivo de processamento local, utilizou-se o Node-RED e manteve-se a arquitetura o mais genérica possível para facilitar a integração de novos equipamentos industriais. Os principais protocolos de comunicação utilizados foram o MODBUS, OPC-UA, FINS e MQTT.

Utilizou-se ainda uma VPN P2P para permitir o acesso de forma remota ao dispositivo IoT. No entanto, os testes realizados a esta componente foram limitados devido às restrições de rede existentes na empresa.

O sistema cloud apresenta diversos dashboards ao utilizador, assim como uma representação da sua planta fabril.

7.1 Satisfação de objetivos

Atribuindo o mesmo peso a cada requisito, é possível afirmar que o sistema desenvolvido cumpre 90% dos requisitos analisados na secção 2.3. A arquitetura desenvolvida para o dispositivo IoT apresenta-se adequada e modular, visto que permite a substituição de diversas componentes utilizadas sem provocar um elevado prejuízo. Pode então afirmar-se que esta prova de conceito possui bastante potencial e rapidamente poderá ser escalada para uma solução comercial.

7.2 Trabalho Futuro

Apesar de todos os aspetos positivos da prova de conceito apresentada anteriormente, existem ainda aspetos que requerem atenção. O primeiro aspeto diz respeito aos equipamentos industriais onde não foi cumprida a implementação. Tendo em conta que o objetivo deste dispositivo é garantir a transversalidade a grande parte dos equipamentos industriais produzidos pela Azevedos Industria, torna-se pertinente adicionar os restantes equipamentos o mais rapidamente possível.

Além disso, devido às questões verificadas no SensorTile.box (sensor IoT utilizado nesta dissertação), recomenda-se a implementação de um dos seguintes aspetos: adaptar o código utilizado no Sensortile.box para o suporte de redes neuronais e, a partir daí, fazer o seu treino ou a substituição desta componente pelo sensor apresentado pela OMRON.

Apesar de ser possível aceder remotamente ao dispositivo IoT, seria interessante para a empresa proponente poder estabelecer um túnel de ligação ao PLC. Atualmente, a empresa recorre a um serviço VPN externo e, utilizando um computador junto do equipamento industrial, consegue estabelecer ligação com o autómato. No entanto, a solução atual força o cliente a realizar diversas ligações físicas. O dispositivo IoT desenvolvido pode resolver esta questão recorrendo ao projeto OpenWrt[89]. Estabelecendo uma *bridge* entre a porta Ethernet do autómato e a ligação VPN seria possível aceder remotamente. No entanto, existem ainda autómatos que não possuem porta Ethernet. Para isso, considera-se que a solução apresentada pela Eltima, o USB Network Gate[90], conseguiria solucionar esta problemática, pois permite partilhar um dispositivo USB na rede.

Por último apresenta-se um aspeto fulcral: a análise preditiva de dados. O *Thingsboard* apresenta-se bastante robusto face às necessidades da empresa proponente. No entanto, para integrar a análise preditiva de dados, é necessário alterar a versão do produto. Em vez de ser utilizada a "Community Edition", indica-se a "Trendz Analytics". Esta atualização traria enormes vantagens, pois a deteção de anomalias e análise preditiva estão implementadas de forma nativa. Além disso, disponibiliza "white-labeling", que poderá ser uma mais-valia nesta solução.

Anexo A

Imagens

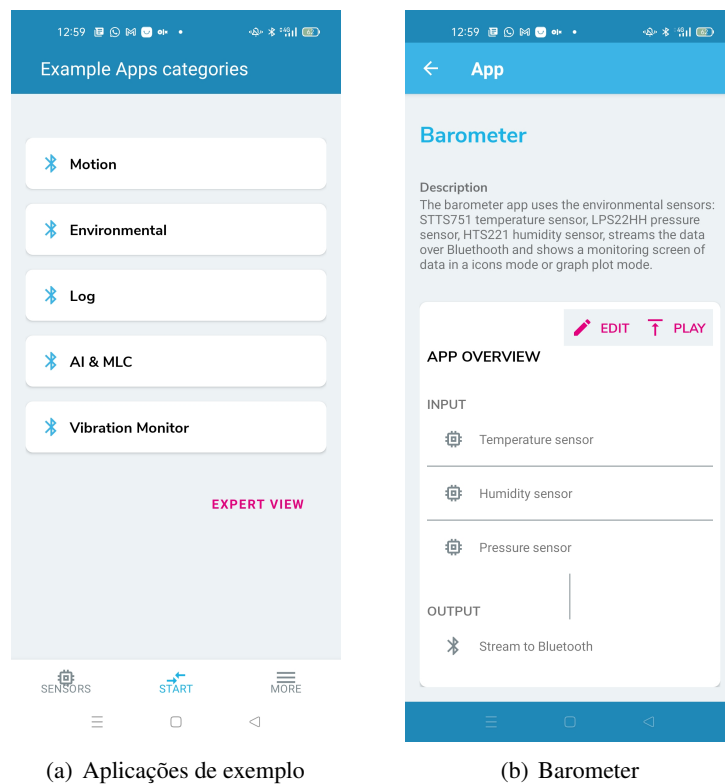


Figura A.1: ST BLE Sensor

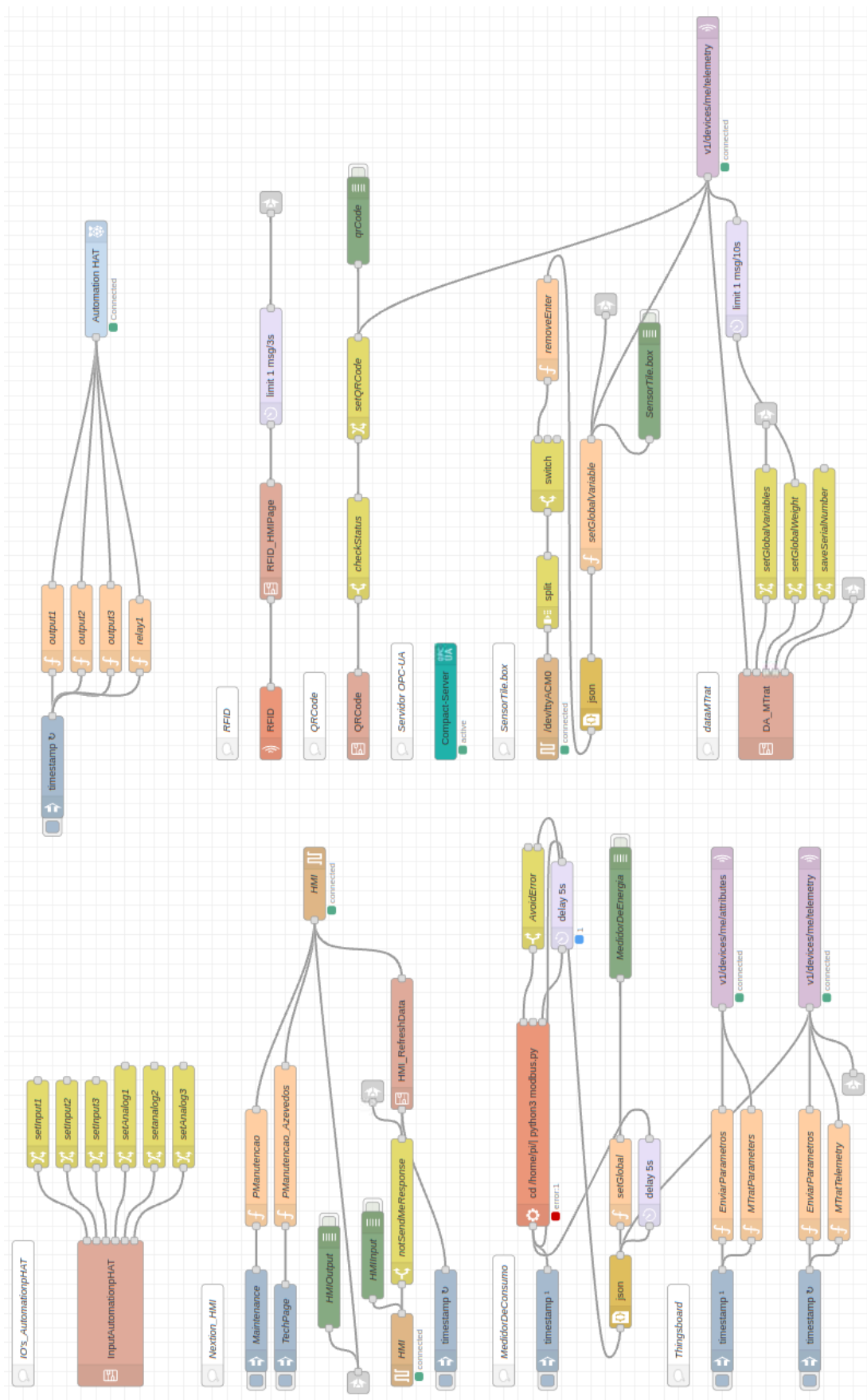


Figura A.2: Fluxo "Inputs&Outputs"desenvolvido no Node-RED

Address Space		Data Access View					
No Highlight		#	Server	Node Id	Display Name	Value	Datatype
Root Objects IoTDevice MTTrat qP1 qP2 qP3 general analog1 analog2 analog3 corksRemain hoursRemain humidity input1 input2 input3 output1 output2 output3 qrCode temperature Server Types Views	1	NodeRED	NS1 String qP1	qP1	13586	Double	
	2	NodeRED	NS1 String qP2	qP2	1280	Double	
	3	NodeRED	NS1 String qP3	qP3	768	Double	
	4	NodeRED	NS1 String analog1	analog1	0.03	Double	
	5	NodeRED	NS1 String analog2	analog2	0.03	Double	
	6	NodeRED	NS1 String analog3	analog3	0.03	Double	
	7	NodeRED	NS1 String corksRemain	corksRemain	1005536	Double	
	8	NodeRED	NS1 String hoursRemain	hoursRemain	1049	Double	
	9	NodeRED	NS1 String humidity	humidity	15	Double	
	10	NodeRED	NS1 String input1	input1	0	Double	
	11	NodeRED	NS1 String input2	input2	0	Double	
	12	NodeRED	NS1 String input3	input3	0	Double	
	13	NodeRED	NS1 String output1	output1	0	Double	
	14	NodeRED	NS1 String output2	output2	0	Double	
	15	NodeRED	NS1 String output3	output3	0	Double	
	16	NodeRED	NS1 String qrCode	qrCode	+	String	
	17	NodeRED	NS1 String temperature	temperature	28.76	Double	

Figura A.3: Variáveis disponíveis por OPC-UA

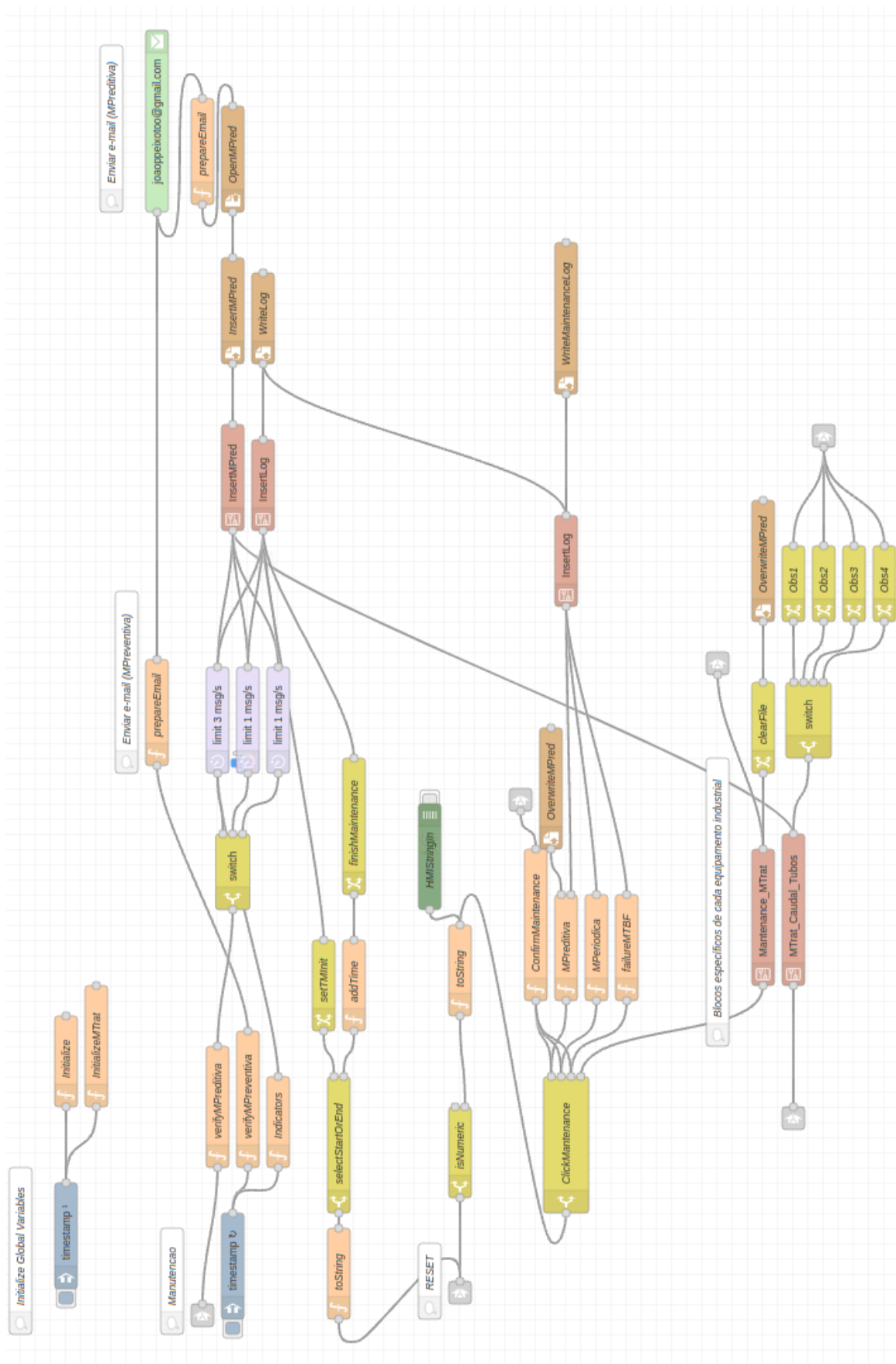


Figura A.4: Fluxo "Maintenance"desenvolvido no Node-RED

Referências

- [1] HMS Networks. Ewon - industrial remote access, remote monitoring data integration, 2022. URL: <https://www.ewon.biz/>.
- [2] Ismail Butun e SpringerLink. *Industrial IoT : Challenges, Design Principles, Applications, and Security*. Springer International Publishing : Imprint: Springer, Cham, 1st 2020. edição, 2020.
- [3] Alchemy Power Inc. Pi-ezconnect - alchemy power inc., 2015. URL: <https://alchemy-power.com/pi-ezconnect/>.
- [4] Vanda Cardoso Pinheiro. Indústria 4.0 a quarta revolução industrial, 2016. URL: https://www.compete2020.gov.pt/destaques/detalhe/Industria_4ponto0.
- [5] RAJIV LAL e SCOTT F. JOHNSON. The digital factory - siemens: Electronic works amberg. 2017.
- [6] Ioana Culic, Alexandru Radovici, Cristian Rusu, e SpringerLink. *Commercial and Industrial Internet of Things Applications with the Raspberry Pi : Prototyping IoT Solutions*. Technology in Action. Apress : Imprint: Apress, Berkeley, CA, 1st 2020. edição, 2020.
- [7] Produtech. Azevedos indústria - máquinas e equipamentos industriais s.a, 2021. URL: <https://portal.produtech.org/company/azevedos-industria-maquinas-e-equipamentos-industriais-s-a>.
- [8] B. Parhami. Defect, fault, error,..., or failure? *IEEE Transactions on Reliability*, 46(4):450–451, 1997. doi:10.1109/TR.1997.693776.
- [9] Pipe Masters. Manutenção industrial: Conheça os 3 principais tipos de manutenção para, 2022. URL: <https://pipemasters.pt/blog/industria/principais-tipos-manutencao-industrial/>.
- [10] Pedro Piovesan. Manutenção preditiva: Guia completo. URL: <https://traction.com/blog/manutencao-preditiva>.
- [11] Pedro Melo. Manutenção e diagnóstico de avarias em motores de indução trifásicos. 2015.
- [12] Gabriel Lameirinhas. 8 indicadores indispensáveis para gestão da manutenção, 2021. URL: <https://traction.com/blog/indicadores-de-manutencao>.
- [13] ACCEPT Quality Control Solutions. Oee: O que É e como calcular?, 2019.
- [14] DMG Mori. 360° digitization by dmg mori, 2021. URL: <https://en.dmgmori.com/solutions/360-degree-digitization>.

- [15] EGITRON. Corkcontrol, 2021. URL: <https://www.egitron.pt/pt/controlo-da-qualidade-corticabebidas/software-para-controlo-da-qualidade/software-para-controlo-da-qualidade-de-producao-corkcontrol-1/>.
- [16] IXON B.V. Plataforma iot sem código para fabricantes de máquinas | ixon, 2022. URL: <https://www.ixon.cloud/pt>.
- [17] Schneider Electric. Ecostruxure machine - alcance a excelência empresarial com soluções para máquinas inteligentes, 2022. URL: <https://www.se.com/pt/pt/work/solutions/machine-control/>.
- [18] MindSphere Siemens. Siemens industrial iot, 2021. URL: <https://siemens.mindsphere.io/en/industrial-iot>.
- [19] Omron Corporation. Nx7 machine automation controller, 2022. URL: <https://automation.omron.com/en/mx/products/family/NX701>.
- [20] Industrial Automation Systems MODICON, Inc. Modicon modbus protocol reference guide. 1996. URL: https://www.modbus.org/docs/PI_MBUS_300.pdf.
- [21] Paessler AG. What is opc ua? definition and details, 2022. URL: <https://www.paessler.com/it-explained/opc-ua>.
- [22] OMRON. *FINS Commands - Reference Manual*. 1993. URL: https://www.myomron.com/downloads/1.Manuals/Networks/W227E12_FINS_Commands_Reference_Manual.pdf.
- [23] The HiveMQ Team. Publish subscribe - mqtt essentials: Part 2, 2015. URL: <https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/>.
- [24] STMicroelectronics. Sensortile.box wireless multi sensor development kit with user friendly app for iot and wearable sensor applications, 2022. URL: <https://www.st.com/en/evaluation-tools/steval-mksbox1v1.html>.
- [25] OMRON Corporation. 2jcie-bu environment sensor(usb type), 2022. URL: <https://components.omron.com/sg-en/products/sensors/2JCIE-BU>.
- [26] The Raspberry Pi Foundation. Sense hat. URL: <https://www.raspberrypi.com/products/sense-hat/>.
- [27] Shelly Cloud. Allterco Robotics LTD. Shelly 3em - shelly cloud, 2022. URL: <https://shelly.cloud/products/shelly-3em-smart-home-automation-energy-meter/>.
- [28] Robert Mauser Lda. Módulo medidor de consumo duplo para automação wifi - shelly em, 2022. URL: https://mauser.pt/catalog/product_info.php?products_id=096-7608&gclid=CjwKCAiA0KmPBhBqEiwAJqKK49k1ajC-j-YmNK6X7eSN6Ivtum6Kn-cwNRSavsTF9VrKs7l2Jg3OyxocIwE.
- [29] ORNO-POLSKA Sp. z o.o. 3-phase multi-tariff energy meter with rs-485, 80a. 3 modules, din th-35mm, 2018. URL: <https://orno.pl/en/product/1087/3-phase-multi-tariff-energy-meter-with-rs-485-80a-3-modules-din-th-35mm>.

- [30] Schneider Electric. Acti9 powertag link hd - wireless to modbus tcp/ip concentrator, 2022. URL: <https://www.se.com/ww/en/product/A9XMWD100/acti9-powertag-link-hd-wireless-to-modbus-tcp-ip-concentrator/?parent-subcategory-id=4160&range=64482-acti9-powertag-link&selected-node-id=44105740929>.
- [31] LLC OpenMV. Openmv | small - affordable - expandable, 2022. URL: <https://openmv.io/>.
- [32] The Raspberry Pi Foundation. Raspberry pi camera module 2. URL: <https://www.raspberrypi.com/products/camera-module-v2/>.
- [33] Vidushi Meel. Google coral for computer vision applications in 2022, 2022. URL: <https://viso.ai/edge-ai/google-coral/>.
- [34] Intel®. Intel® neural compute stick 2. URL: <https://www.intel.com/content/www/us/en/developer/tools/neural-compute-stick/overview.html>.
- [35] Robert Mauser Lda. Módulo leitor rfid rc522 (13.56mhz) compatível com arduino. URL: https://mauser.pt/catalog/product_info.php?products_id=096-8517.
- [36] ifm electronic gmbh. Vtv122 - transmissores de vibração - ifm, 2022. URL: <https://www.ifm.com/pt/pt/product/VTV122?tab=details>.
- [37] ifm electronic gmbh. Vnb001 - sensor de vibração - ifm, 2022. URL: <https://www.ifm.com/pt/pt/product/VNB001?tab=details>.
- [38] PIMORONI LTD. Automation hat - pimoroni. URL: <https://shop.pimoroni.com/products/automation-hat?variant=30712316554>.
- [39] OpenLX SP Ltd. Piface – piface relay plus, 2013. URL: http://www.piface.org.uk/products/piface_relay_plus/.
- [40] ElectroFun. Módulo 4 relés - sinal 5v 4 canais 220v/10a, 2022. URL: <https://www.electrofun.pt/componentes-eletronicos/modulo-rele-5v-4-canais>.
- [41] ITEAD STUDIO. Nx4024t032 - nextion, 2011 - 2022. URL: <https://nextion.tech/datasheets/nx4024t032/>.
- [42] ElectroFun. Display lcd tft touchscreen 3.5"320 x 480 para raspberry pi, 2022. URL: <https://www.electrofun.pt/display/display-lcd-tft-touchscreen-3-5-320x480-raspberry-pi>.
- [43] ElectroFun. Teclado matricial de membrana 4x4 para arduino, 2022. URL: <https://www.electrofun.pt/botoes-e-teclados/teclado-matricial-arduino-16-teclas>.
- [44] ElectroFun. Display lcd 16x2 com fundo azul, 2022. URL: <https://www.electrofun.pt/display/display-lcd-16x2>.
- [45] Robert Mauser Lda. Módulo esp32 nodemcu wifi cp2102, 2022. URL: https://mauser.pt/catalog/product_info.php?cPath=1667_2889_2619&products_id=096-8744.

- [46] Industrial Shields. Industrial controller family esp32 plc. URL: <https://www.industrialshields.com/industrial-esp32-plc-products-family-ideal-for-iot-solutions>.
- [47] REX Controls s.r.o. Monarco hat: Industrial grade i/o board for the raspberry pi. URL: <https://www.monarco.io/>.
- [48] Sfera Labs S.r.l. Iono pi, 2022. URL: <https://www.sferalabs.cc/product/iono-pi/>.
- [49] subsidiary of Faster CZ spol. s r.o. Unipi technology. Unipi: Automation. without limits, 2022. URL: <https://www.unipi.technology/>.
- [50] KUNBUS GmbH. Revolution pi - the only real industrial raspberry pi. URL: <https://revolutionpi.com/revolution-pi-series/>.
- [51] F.Fonseca. Computador industrial compacto uno-220 – advantech, 2020. URL: <https://www.ffonseca.com/pt/prod-compact-industrial-pc-uno-220-advantech>.
- [52] Node-RED. Node-red: Low-code programming for event-driven applications, 2022. URL: <https://nodered.org/>.
- [53] CODESYS GmbH. Codesys group | we software automation, 2022. URL: <https://www.codesys.com/>.
- [54] n8n.io. n8n.io - free and open workflow automation tool, 2022. URL: <https://n8n.io/>.
- [55] Apache NiFi Team. Apache nifi overview, 2022. URL: <https://nifi.apache.org/docs/nifi-docs/html/overview.html>.
- [56] The ThingsBoard Authors. Thingsboard: Open-source iot platform, 2002. URL: <https://thingsboard.io/>.
- [57] Inc. Amazon Web Services. Nível gratuito da aws, 2022. URL: https://aws.amazon.com/pt/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=categories%23iot.
- [58] Google Cloud. Google cloud. URL: <https://tinyurl.com/mu2wkfsk>.
- [59] Microsoft. Hub iot | microsoft azure, 2022. URL: <https://azure.microsoft.com/pt-pt/services/iot-hub/>.
- [60] Microsoft Azure. O que é uma vpn? por que motivo devo utilizar uma vpn? | microsoft azure, 2022. URL: <https://azure.microsoft.com/pt-pt/overview/what-is-vpn/#get-started>.
- [61] Husarnet sp. z o.o. Operate at the edge of latency, 2018-2021. URL: <https://husarnet.com/>.
- [62] Deeper Network. Deeper network, 2022. URL: <https://shop.deeper.network/>.
- [63] NordLayer. Adaptive network access security solutions | nordlayer. URL: <https://nordlayer.com/>.

- [64] Lda Ledistronica Componentes Electrónicos. Redutor tensao dc step-down in 9v-38v out 5v 5a, 2021. URL: <https://ledistronica.pt/produto/redutor-tensao-dc-step-down-in-9v-38v-out-5v-5a/>.
- [65] Adafruit. Pimoroni automation phat for raspberry pi zero. URL: <https://www.adafruit.com/product/3352>.
- [66] Peter Vis. Raspberry pi csi camera interface. URL: https://www.petervis.com/Raspberry_PI/Raspberry_Pi_CSI/Raspberry_Pi_CSI_Camera_Interface.html.
- [67] botnroll.com. Conversor usb rs485 para ambiente industrial, 2022. URL: <https://www.botnroll.com/pt/usb/3548-conversor-usb-rs485-para-ambiente-industrial.html>.
- [68] Austin George. Ender 3 raspberry pi camera mount by austin3dprinting - thingiverse, 2018.
- [69] STMicroelectronics. Stblesensor - stmicroelectronics, 2022. URL: <https://www.st.com/en/embedded-software/stblesensor.html>.
- [70] Nicola Capovilla davidroid, matteoronchi. Bluestsdk python, 2022. URL: https://github.com/STMicroelectronics/BlueSTSDK_Python.
- [71] STMicroelectronics. Stm32cubeide - integrated development environment for stm32, 2022.
- [72] ITEAD STUDIO. Made for hmi gui development, 2011-2022. URL: <https://nextion.tech/nextion-editor/>.
- [73] ITEAD STUDIO. The nextion instruction set. 2011-2022. URL: <https://nextion.tech/instruction-set/>.
- [74] Node-RED. Working with context : Node-red. URL: <https://nodered.org/docs/user-guide/context>.
- [75] Duncan Bellamy Martin Rowan. node-red-contrib-automation-hat, 2021. URL: <https://flows.nodered.org/node/node-red-contrib-automation-hat>.
- [76] frederic bonjour. node-red-contrib-mfrc522-rfid, 2020. URL: <https://flows.nodered.org/node/node-red-contrib-mfrc522-rfid>.
- [77] Python Software Foundation. pyzbar · pypi, 2022. URL: <https://pypi.org/project/pyzbar/>.
- [78] Python Software Foundation. Pillow · pypi, 2022. URL: <https://pypi.org/project/Pillow/>.
- [79] Ryoichi Obara Sandeep Chaturvedi. node-red-contrib-qrcode, 2021. URL: <https://flows.nodered.org/node/node-red-contrib-qrcode>.
- [80] Bianco Royal Space. node-red-contrib-opcua-server (node) - node-red, 2020. URL: <https://flows.nodered.org/node/node-red-contrib-opcua-server>.
- [81] Stephen McLaughlin. node-red-contrib-omron-fins, 2021.
- [82] Steve Cope. Creating an iot or mqtt dashboard using thingsboard, 2021.

- [83] Node-RED. node-red-node-email, 2022. URL: <https://flows.nodered.org/node/node-red-node-email>.
- [84] Google. Sign in with app passwords, 2022. URL: <https://support.google.com/mail/answer/185833?hl=en>.
- [85] The ThingsBoard Authors. Installing thingsboard on raspberry pi 3 model b, 2022. URL: <https://thingsboard.io/docs/user-guide/install/rpi/>.
- [86] The ThingsBoard Authors. Getting started with rule engine | thingsboard community edition, 2022. URL: <https://thingsboard.io/docs/user-guide/rule-engine-2-0/re-getting-started/>.
- [87] The ThingsBoard Authors. Http device api reference | thingsboard community edition, 2022. URL: <https://thingsboard.io/docs/reference/http-api/>.
- [88] Joshua Hrisiko. Real-time graphing in python, 2018. URL: <https://makersportal.com/blog/2018/8/14/real-time-graphing-in-python>.
- [89] Hauke Mehrtens. [openwrt wiki] welcome to the openwrt project, 2022.
- [90] Inc. Electronic Team. Usb over ethernet: software de compartilhamento usb pela rede, 2022. URL: <https://www.eltima.com/pt/products/usb-over-ethernet/>.