FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Real-time database management for the IIOT platform

António Miguel Lopes Maia

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Maria Teresa Delgado

July 25, 2024

Resumo

No contexto do desenvolvimento industrial global e do aumento da automação em todos os sectores, é essencial desenvolver e implementar soluções inovadoras que sejam superiores às ofertas actuais em termos de custo e eficácia. Este projeto responde a esta necessidade, centrando-se na gestão eficiente e no armazenamento em tempo real de dados de máquinas, utilizando uma combinação de ferramentas locais e virtuais. O objetivo principal é estabelecer um sistema ágil e económico para monitorizar máquinas robóticas de 6 eixos em tempo real. O objetivo deste sistema é monitorizar e analisar o estado da máquina, as métricas de produção, o consumo de energia, os tipos de erro e as ordens de produção para melhorar a eficácia global do equipamento (OEE). Está a ser criado um painel de dados dinâmico e em tempo real utilizando linguagens de programação como JavaScript e HTML, juntamente com MQTT para uma transferência de dados robusta. Esta abordagem agiliza o processo de monitorização e contribui para uma tomada de decisões mais inteligente em ambientes de produção. Com um plano de desenvolvimento estruturado que inclui análise, propostas de melhoria, construção de design, implementação de soluções e testes simulados e reais, este projeto visa estabelecer uma referência para o tratamento de dados em tempo real e a eficiência operacional no sector da automação industrial.

Keywords: Base de dados, aplicação web, visualização de dados

Abstract

In the context of global industrial development and the increase in automation across all sectors, it is essential to develop and implement innovative solutions that are superior to current offerings in terms of cost and effectiveness. This project addresses this need by focusing on the efficient management and real-time storage of machine data, utilizing a combination of local and virtual tools. The primary objective is to establish an agile and cost-effective system for monitoring 6-axis robotic machines in real-time. The purpose of this system is to monitor and analyse machine status, production metrics, energy consumption, error types, and production orders to enhance the Overall Equipment Effectiveness (OEE). A dynamic, real-time data dashboard is being created using programming languages such as JavaScript and HTML, along with MQTT for robust data transfer. This approach streamlines the monitoring process and contributes to smarter decision-making in manufacturing environments. Through a structured development plan that includes analysis, enhancement proposals, design construction, solution implementation and both simulated and real-world testing, this project aims to set a benchmark for real-time data handling and operational efficiency in the industrial automation sector.

Keywords: Database, web app, data visualization

Sustainable Development Goals

The 2030 Agenda for Sustainable Development, adopted by all United Nations Member States in 2015, provides a shared blueprint for peace and prosperity for people and the planet, now and into the future. At its heart are the 17 Sustainable Development Goals (SDGs), which are an urgent call for action by all countries - developed and developing - in a global partnership. They recognize that ending poverty and other deprivations must go hand-in-hand with strategies that improve health and education, reduce inequality, and spur economic growth – all while tackling climate change and working to preserve our oceans and forests [1].
My Master's thesis in Electrical and Computer Engineering introduces a real-time monitoring system for 6-axis robotic machines, which significantly enhances industrial efficiency and innovation. This initiative enhances the efficiency of production processes and the resilience of industrial infrastructure, thereby contributing to the goal of Industry, Innovation, and Infrastructure. Furthermore, it supports the principles of responsible consumption and production by optimising the utilisation of resources, and it also contributes to the advancement of affordable and clean energy through the enhancement of energy efficiency. These developments serve to illustrate the vital role that engineering solutions play in the achievement of the United Nations' Sustainable Development Goals.

SDG	Goal	Contribution	Performance Indicators and Metrics
7	1	Promotes efficient energy use in industrial settings	Energy savings, reduction in energy intensity of manufacturing
8	1	Automation enhances productivity and creates high-skilled jobs	Employment rates in tech, wage improvements, economic growth
9	1	Advanced analytics for resilient infrastructure and innovation	Number of innovations, increased operational efficiency
	2	Enhancement of industrial infrastructure with real-time machine monitoring	Increase in OEE, reduction in downtime, accuracy of machine analytics
11	1	Improves environmental footprint of manufacturing in urban areas	Improvement in local air quality, reduction in industrial noise levels
12	1	Optimizes energy usage and reduces waste through precise monitoring	Reduction in energy per production unit, decrease in production waste
13	1	Optimized operations reduce greenhouse gas emissions	Reduction in CO_2 emissions, energy efficiency improvements
17	1	Collaborations to develop sustainable technologies	Number of partnerships, impact on sustainability metrics

Project Contributions to Sustainable Development Goals (SDGs)

Acknowledgements

I would like to express my sincerest gratitude to all those who contributed to the completion of this thesis.
Firstly, I am grateful to my supervisor, Maria Teresa Delgado. Your patience, care, and unwavering belief were invaluable throughout this journey. Your insightful feedback and support have been instrumental in shaping this work. I would like to extend my gratitude to my colleagues in the ESI Robotics. The company provided an exceptional work environment and essential support, along with access to technologies that proved pivotal in the success of this work. I would also like to thank my family for the support and conditions given and my friends, specially Gonçalo, Vicente and Pinho, who accompanied me in this important journey of the last 5 years. Finally, to express my gratitude to the Faculty of Engineering of the University of Porto that prepared me for the world of work and to overcome obstacles that appear in life, which shape me and make me be what I am today.

António Miguel Lopes Maia

"Never say no to an idea -
you never know how that idea will ignite another idea."

Stanley Kubrick

Contents

List of Figures

List of Tables

Abbreviations and Symbols

BI	Business Intelligence
CDN	Content delivery network
DOM	Document Object Model
ERP	Enterprise Resource Planning
IIoT	Industrial Internet of Things
IoT	Internet of Things
JSON	JavaScript Object Notation
KPIs	Key Performance Indicators
LWT	Last Will and Testament
MQTT	Message Queuing Telemetry Transport
OEE	Overall Equipment Effectiveness
OPC UA	Open Platform Communications United Architecture
QoS	Quality of Service
RTDB	Real-Time Database
SPA	Single-Page Application
SSL	Secure Sockets Layer
TLS	Transport Layer Security
TSDB	Time-Series Database
UI	User Interface
ZB	Zettabyte

Chapter 1

Introduction

In the last 20 years, data has increased on a large scale in several areas. According to a report from International Data Corporation (IDC), in 2011, the world created and copied an amount of 1.8 ZB data volume, that increased by almost 9 times in the next five years [3]. The first chapter of this thesis is the background of the company in Section 1.1, where the development of this thesis has taken place. The aims of the thesis are described in Section 1.2, and Section 1.3 provides an overview of the structure of the document, summarising the content and layout of the of the following chapters.

1.1 The Company

ESI Robotics was established 16 years ago as a technology-based company with the primary objective of helping companies improve their competitiveness and profitability. The company promotes progress in the field through extensive expertise and innovative skills. Focusing on production research, ESI Robotics concentrates on overcoming operational difficulties, automating procedures, and refining processes to reduce costs and increase value added.

The organisation is committed to crafting bespoke industrial solutions that provide for the distinct requirements of each customer. Through this customer-focused strategy, ESI Robotics strives to facilitate significant performance improvements, thereby enhancing the prosperity and efficiency of the companies it serves.

ESI Robotics' innovative approach has been demonstrated through the development of custom robotic solutions for industrial process optimisation. One of these was an efficient and sustainable packaging solution for the Amorim Group, which considerably reduced shipping volumes and logistics costs and minimised the environmental footprint. The project highlights ESI Robotics' ability to create technologically advanced and environmentally conscious solutions.

Additionally, ESI Robotics has made significant progress in industrial digitalization, integrating the latest advancements in the Internet of Things and smart manufacturing. The company's focus on digital transformation positions it at the forefront of Industry 4.0, enabling real-time data collection and analysis, which is crucial for modernising industrial processes.

ESI Robotics is a leader in the field of industrial automation and robotics, with a commitment to research and development (R&D) that fosters an environment of continual innovation.

This dedication has been recognized through various awards and patents.As ESI Robotics looks towards the future, their ongoing investment in R&D promises to yield new technologies and solutions that will continue to redefine the standards of efficiency and productivity in the industry.

Figure 1.1: ESI Robotics's logo

1.2 Motivation and Objectives

Robotic systems can already proactively monitor and adapt to changes in a production line. Nowadays, internet of things and robotic systems are key drivers of technological innovation trends [4].

In the rapidly evolving field of electrical and computer engineering, the integration of real-time data monitoring in robotic machines represents a significant step towards more efficient and sustainable industrial processes. The primary objective of this dissertation is to explore and develop a practical solution for this monitoring, with a focus on improving energy efficiency and production quality in automated systems.

- Improved operational efficiency: By analysing real-time data from robotic machines, this project aims to identify inefficiencies and optimise operational parameters. This could lead to significant reductions in energy consumption, resulting in cost savings and a reduced environmental footprint for industrial processes.

- Improved production quality: The ability to monitor and analyse data in real time allows the immediate identification of production anomalies. This can improve the overall quality of output by enabling prompt corrective action, reducing waste and ensuring consistent product standards.

- Advanced data management: The development of a programme to effectively handle and process the data collected by the robotic machines is a cornerstone of this project. By storing this data in an organised database and transferring it to an easily interpretable dashboard, the project facilitates a more user-friendly approach to data analysis. This level of organisation is critical for making data-driven decisions and understanding complex machine behaviour.

- Innovative user interface: The creation of a graphical interface, or dashboard, is critical to translating complex data into actionable insights. By presenting data in an intuitive, visually appealing format, the interface enables users to interpret and utilise the information easily, leading to more informed decisions.

- Broader implications: Beyond immediate operational improvements, this work has the potential to contribute to broader discussions about sustainable industrial practices. Understanding the energy costs and operational efficiencies of machines could lead to industry-wide advances in how energy is consumed and conserved in manufacturing and production.

- Future Scope: The research and development undertaken in this thesis could open doors for further innovation in robotics and automation. It lays the groundwork for future projects that could explore more advanced data analytics, machine learning algorithms and predictive maintenance techniques.

1.3 Document Structure

This document consists of 7 chapters. Chapter 2 provides the relevant background information necessary to understand the context of the research. An introduction sets the stage for a detailed exploration of KUKA Connect, explaining its capabilities and significance in industrial automation. The chapter then discusses communication protocols, focusing on the MQTT Protocol and the OPC UA Protocol, and dissecting their roles and functionalities within the IoT and industrial communication.

Chapter 3 introduces the state-of-the-art technologies and tools relevant to the project. It reviews and compares leading solutions such as MachineMetrics, Google Analytics and Grafana, providing a comprehensive understanding of their features and limitations.

In Chapter 4, the problem statement is outlined. The application, Factory Pulse, is introduced, and the issues it is designed to address defined. The chapter also specifies the requirements, detailing roles, stories, and technical prerequisites, with the intention of ensuring a clear understanding of what is to be achieved and the necessary components to facilitate this.

Chapter 5 details the implementation process. This chapter describes the tools and technologies used, including the backend (*NodeJS*), frontend (*ReactJS*), and the database (Firebase). In addition, the chapter discusses the design and implementation of the user interface for both authenticated and unauthenticated users. Moreover, it covers the setup and functionalities of Firebase services, including the real-time database, firestore, authentication service, and hosting service.

Chapter 6 is dedicated to an analysis of the tests and the results that were obtained. It outlines the methodologies and procedures that were employed in the testing conducted in both simulated and real environments. In addition, the chapter offers a examination of the results, which are discussed with regard to the system's performance and its reliability.

Finally, Chapter 7 brings this document to a conclusion. It provides an overview of the most significant findings and achievements of the project, which are presented in a manner that validates the system that has been developed. The chapter additionally presents potential future developments. It proposes the introduction of additional pages for various types of machines and discusses the difficulties inherent in implementing the system within the context of a factory. The structure of this document ensures a comprehensive understanding of the project's background, methodology, implementation, testing, and conclusions.

Chapter 2

Background

The pursuit of innovation is driven by the development and application of advanced communication protocols and development platforms. This chapter explores the pivotal technologies that are shaping the future of robotics and automation in the dynamic realm of electrical and computer engineering. This text focuses on a selection of modern tools and protocols, including KUKA Connect, MQTT Protocol, OPC UA Protocol, and the Unity development platform. These tools play a crucial role in the development of intelligent industrial systems.

2.1 Industry 4.0

Today, we're at a juncture: industry 4.0 is on the rise due to automation and the ubiquity of computing power and has spurred what many consider yet another industrial revolution. Industry 4.0 refers to the use of automation and data exchange in manufacturing. According to the Boston consulting group, there are nine principal technologies that make up industry 4.0: Autonomous robots, simulation, horizontal and vertical system integration, the industrial internet of things, cyber security, the cloud, additive manufacturing, data and analytics, and augmented reality [5]. As such, the currently ongoing 4th industrial revolution, usually referred to as Industry 4.0 in Europe [[6], [7], [8]] and Industrial Internet in the US [9], aims to introduce and take advantage of the interconnected world along the entire value chain, allowing the sharing and processing of the data available in all of its actors to generate relevant knowledge and optimize the overall process [10].

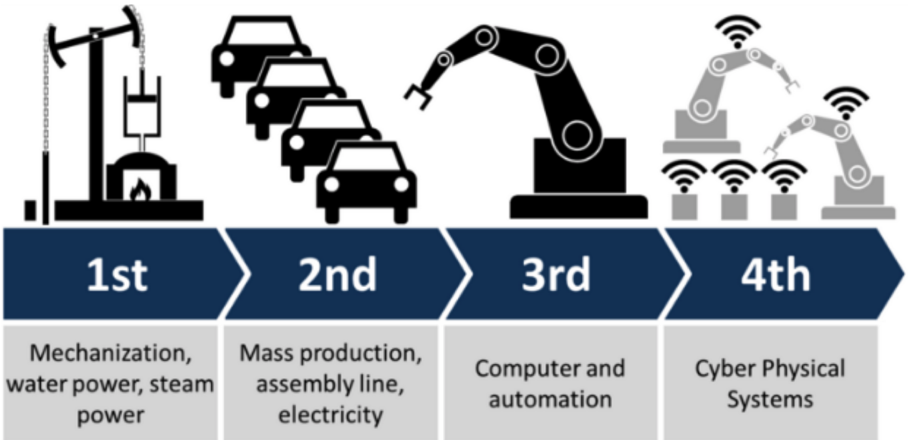

Figure 2.1: Phases of the industrial revolution [2]

2.2 KUKA Connect

Industrial robots are getting smarter all the time. With solutions such as KUKA Connect, you can access your robots from anywhere. This is a cloud software platform that lets you make sense of robot data. Users will be able to react and respond to information received from their automation process [11].

KUKA

Figure 2.2: KUKA's logo

KUKA is a global automation corporation with sales of around 4 billion euros and roughly 15,000 employees. The company is headquartered in Augsburg, Germany. As one of the world's leading suppliers of intelligent automation solutions, KUKA offers customers everything they need from a single source: from robots and cells to fully automated systems and their networking in markets such as automotive, electronics, metal & plastic, consumer goods, e-commerce/retail and healthcare [12].

KUKA Connect is a cloud-based software platform designed by KUKA to augment the capabilities of KUKA robots and automation systems through remote monitoring and maintenance. Its primary aim is to boost the efficiency and productivity of these robotic systems by offering a suite of advanced features, including real-time monitoring, in-depth analysis, and predictive maintenance.

The platform enables users to remotely access and review key performance indicators (KPIs) of their robotic systems, fostering an environment for informed, data-driven decision-making and proactive problem resolution. Among its notable features are dashboard visualization, which provides an intuitive interface for data interpretation, alert notifications for timely responses to system irregularities, and historical data analysis for identifying trends and patterns in robot performance.

KukaConnect is a program that enables communication in OPC UA or MQTT. It will be used for data collection from machines and it's transfer to the workstation.

KUKA Connect demonstrates the evolving nature of industrial automation, where cloud technology and robotics converge to create more intelligent and responsive systems. It plays a crucial role in minimising downtime, optimising operational workflows and ensuring the longevity and effectiveness of robotic assets in a wide range of industrial environments.

2.3 Communication Protocols

2.3.1 MQTT Protocol

MQTT stands for Message Queuing Telemetry Transport and is a widely used messaging protocol in the Internet of Things (IoT) domain. Its lightweight and versatile design makes it ideal for the IoT's dynamic and diverse environment.

MQTT operates on a publisher-subscriber model, allowing devices to publish messages on specific topics or subscribe to those topics to receive relevant messages, enabling efficient and targeted communication.

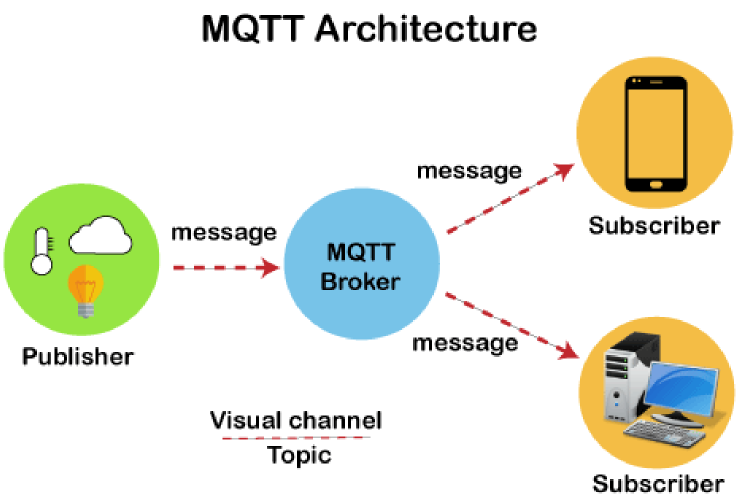

Figure 2.3: MQTT Protocol Operating Process [5]

Key Features and Strengths of *MQTT*:

- Adaptability to Network Conditions: One of the most significant advantages of MQTT is its ability to function effectively in networks with low bandwidth or high latency. This adaptability makes it particularly suited for remote or constrained environments where traditional communication protocols may fail.

- Retained Messages: A vital feature of MQTT is the concept of retained messages. When a message is sent as a 'retained message' on a topic, the MQTT broker stores this message and delivers it to any new subscribers of the topic, ensuring that the most recent information is readily available.
- Quality of Service (QoS) Levels: MQTT offers three levels of QoS for message delivery - "At most once," "At least once," and "Exactly once." These levels provide flexibility in terms of message delivery guarantees based on the requirements of the application.
- Last Will and Testament (LWT): This feature allows a device to publish a final message to a specified topic if it unexpectedly disconnects, which is crucial for maintaining the system's integrity and alerting other devices or systems of potential issues.
- MQTT in IoT Ecosystems: MQTT's lightweight design not only conserves bandwidth but also requires minimal code footprint on devices, making it ideal for the varied devices found in IoT ecosystems, from sensors to full-scale industrial machines. Its ability to support thousands of concurrent devices makes it scalable and robust for large-scale IoT deployments.
- Security Aspects: While MQTT itself is a lightweight protocol focusing on efficient communication, it can be secured using SSL/TLS for encrypted communication channels, thus ensuring data security over the network.
- Use Cases: The versatility of MQTT is evident in its wide range of applications, from home automation systems and vehicle telemetry to industrial IoT solutions. It enables real-time data collection and monitoring, crucial for predictive maintenance, energy management, and other IoT-driven solutions.

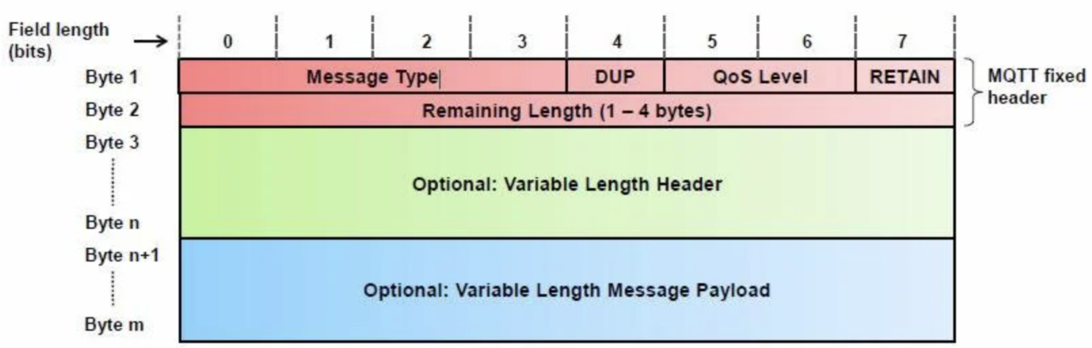

Figure 2.4: MQTT Protocol Message Format [13]

The MQTT protocol message header consists of the first 2 bytes of the message. The first 4 bits indicate the type of message to be sent, which can vary depending on the request. The following section contains a flag (DUP) that is only activated if the quality of service value is greater than 0, indicating that the message needs to be retransmitted. The quality of service value fills bits 5 and

6, indicating the reliability of the message delivery to the receiver. The hold flag used in PUBLISH messages to ensure message delivery is contained in the last bit of this first byte.

The server that receives the message will wait for the message to be received by the current subscribers before it deletes the message. The rest of the message consists of the optional header and payload. The header is filled in depending on the type of message being sent, usually containing a message identifier and similar parameters. The payload contains the data that will be sent.

2.3.2 OPC UA Protocol

The OPC UA specification integrates all the functionality from the existing OPC Classic specifications into a single service-oriented architecture. It adds essential new features, such as platform independence, diagnostics, discovery, rendering of complex information models, security, and reliability. Additionally, OPC UA was released as an IEC Standard, IEC 62541 in October 2011 [12].

OPC (OLE for Process Control) and OPC UA (OPC Unified Architecture) are two generations of standards in industrial automation for data exchange. OPC, developed in the mid-1990s, relied on Microsoft's OLE technology and was primarily used for real-time plant data communication. OPC Classic included specifications such as Data Access (DA), Historical Data Access (HDA), and Alarms & Events (A&E). However, OPC Classic was largely confined to Windows environments and faced challenges with network security and scalability because it was based on Microsoft's COM/DCOM technology.

To overcome these limitations, OPC UA was developed in the mid-2000s. OPC UA is designed to be platform and technology-agnostic, which allows it to overcome the restrictions of OPC Classic.

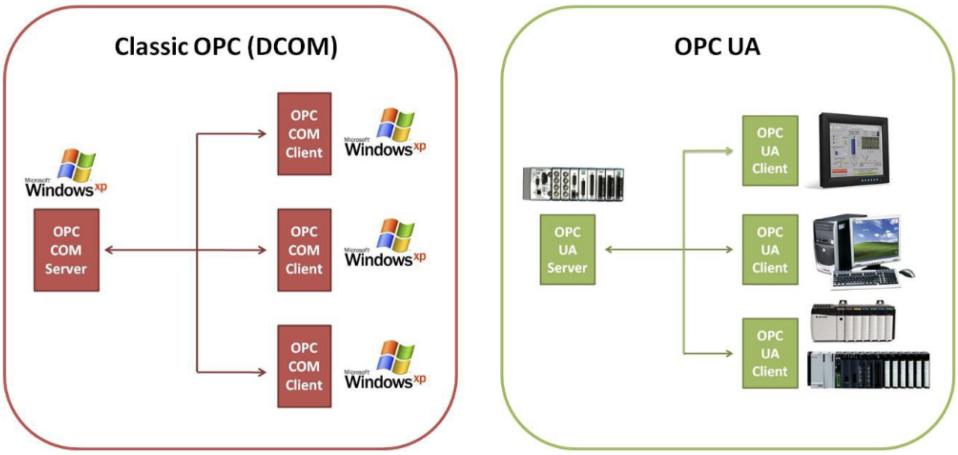

Figure 2.5: OPC UA bypasses the need for a Windows-based component and can communicate directly with embedded OPC UA servers on PLCs [14]

The OPC protocol works based on the client/server architecture. The OPC server is implemented in a PLC (DCS) or embedded device and is part of a distributed control system. The OPC client uses the OPC server to obtain data from the hardware or to send commands to it and provides communication for various applications. The protocol OPC UA runs on TCP / IP and enables secure data transmission in binary or XML format [15].

OPC UA is a comprehensive industrial communications protocol that provides a robust framework for secure and efficient interactions in a variety of industrial environments. Unlike MQTT, OPC UA incorporates a more extensive, object-oriented communication structure, enabling detailed representation of complex data and processes.

This protocol prioritises security by implementing mechanisms such as advanced encryption, user authentication, and authorisation to safeguard data integrity and confidentiality.

The Unified Information Model is central to OPC UA, as it allows the integration and representation of complex industrial data structures. This model promotes interoperability among various systems and devices, creating a cohesive operational environment. Additionally, OPC UA is built on a service-oriented architecture, which enables a wide range of services such as data reading, writing, and monitoring of alarms and events. Flexibility is crucial in adapting to the specific requirements of various industrial applications.

The protocol's ability to function across different transport protocols, including TCP/IP, ensures reliable communication across various platforms and networks. This versatility extends OPC UA's applicability to a broad spectrum of industrial scenarios, from manufacturing to energy management. OPC UA is designed for scalability, enabling it to handle large-scale industrial environments with numerous devices and systems. Its performance remains consistent and reliable even in extensive and complex networks.

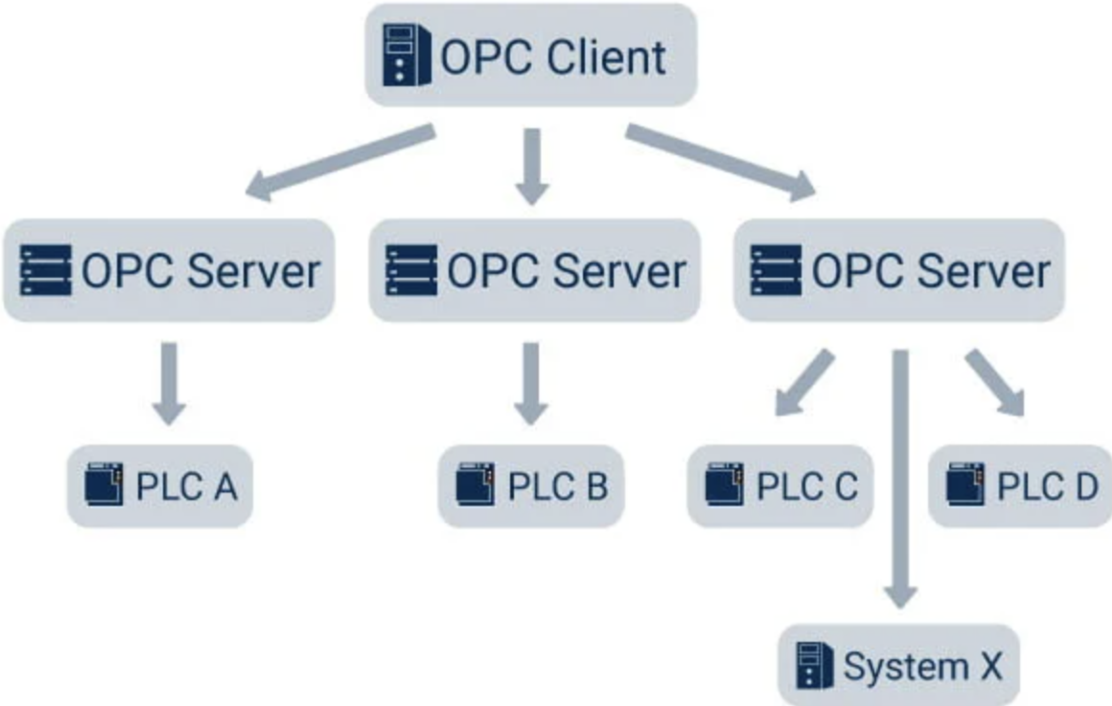

Figure 2.6: OPC Server [12]

In summary, OPC UA's advanced features, such as its comprehensive communication model and robust security measures, make it an indispensable protocol in the realm of industrial automation and IoT, facilitating a secure, efficient, and interconnected industrial landscape.

2.4 Summary

Chapter 2 of this report concludes by establishing foundational knowledge and theoretical frameworks that are instrumental in setting the stage for the upcoming project. The report examines KUKA Connect and analyses the MQTT and OPC UA communication protocols, laying the groundwork necessary for a deep dive into the management and storage of real-time data for industrial machinery.

The technologies discussed in this chapter are not just theoretical concepts but are the fundamental building blocks for the project's infrastructure. We have identified KUKA Connect's cloud-based capabilities, the lightweight yet robust messaging system of MQTT, and the secure, scalable architecture of OPC UA as essential components for our system.

The following stages of the project will involve applying these principles to construct a coherent system that aims to make significant advancements in the real-time monitoring of industrial machinery. The insights obtained from this chapter will serve as a guide, directing the detailed planning and careful execution of the project as it moves from concept to reality.

Chapter 3

State of Art

This chapter introduces the forefront of real-time data analytics tools: MachineMetrics, known for its industrial IoT applications; Google Analytics, celebrated for its integration within Google's ecosystem; and Grafana, acclaimed for its open-source flexibility in data visualization. We briefly explore their key functionalities, aiming to set the stage for a deeper comparative analysis on their unique contributions to data-driven decision-making processes.

3.1 Introduction

Due to continuous advancements in Information and Communication Technologies and the fast-paced nature of the business environment today, organizations generate and deal with increasingly more data. Managers are often overwhelmed with reports and information churned out from a multitude of organizational information systems such as Enterprise Resource Planning (ERP), performance scorecards, and business intelligence (BI) software that compete for managers' attention. This phenomenon is generally known as information overload. The problem is further exacerbated when reports are poorly designed with respect to how information is presented, which often distract than guide decision makers' attention [16].

Dashboards are business intelligence reporting tools that display relevant data according to specific metrics through graphic elements, such as charts, graphs, and other visual elements, allowing a quick visualization [17] [18].

Through its analysis, it is possible to establish relations and draw conclusions, allowing the monitoring of the organization, as it improves the knowledge of its status. Dashboards assist in decision-making as they synthesize extensive data sets, displaying specific information in real or near real-time and highlighting factors that request consideration [19] [20]. The utilization of dashboards facilitates the attainment of organizations' goals by enabling thoughtful and well-informed strategic decision-making. Moreover, it enhances the monitoring of daily organizational activities [18] [21].

3.2 MachineMetrics

Your machines are generating hundreds of data points every milli-second, all of which tell a story of what's happened in the past, what's happening now and, and what's going to happen next. MachineMetrics harnesses the power of your machine data to automatically generate insights for your frontline workers. Now, data can become your competitive advantage. MachineMetrics is manufacturing's first Industrial IoT Platform for Machines. We transform analytics into action through universal machine connectivity, cloud data Infrastructure, and communication workflows that optimize machine operation. Extensible both in the cloud and at the edge, the MachineMetrics platform is powering vertically-focused use cases for manufacturers to digitize legacy manufacturing processes and drive more profitability with their machines [22].

Right now, hundreds of manufacturers have connected thousands of machines to MachineMetrics across global factories. Our platform is enabling these companies to deliver the right information to the right person at the right time to improve their machine performance and productivity, increase their capacity utilization and ultimately win more business to remain globally competitive.

The Current Shift Dashboard views available in MachineMetrics include [23]:

- **Parts Goal**: This shows the machine status, performance, and parts produced each hour compared to the parts goal for the Operation on the machine.
- **OEE**: This shows the machine status, performance, and parts produced each hour compared to the Overall Equipment Effectiveness (OEE) goal for the Operation on the machine.
- **Utilization**: This shows the status and performance of machines compared to the utilization goal for the machine.
- **Downtime**:This shows the current status of the machine in terms of whether the machine is down (idle or stopped) or up (actively processing parts).

The figure 3.1 shows an example of the Parts Goal view of the Current Shift Dashboard.

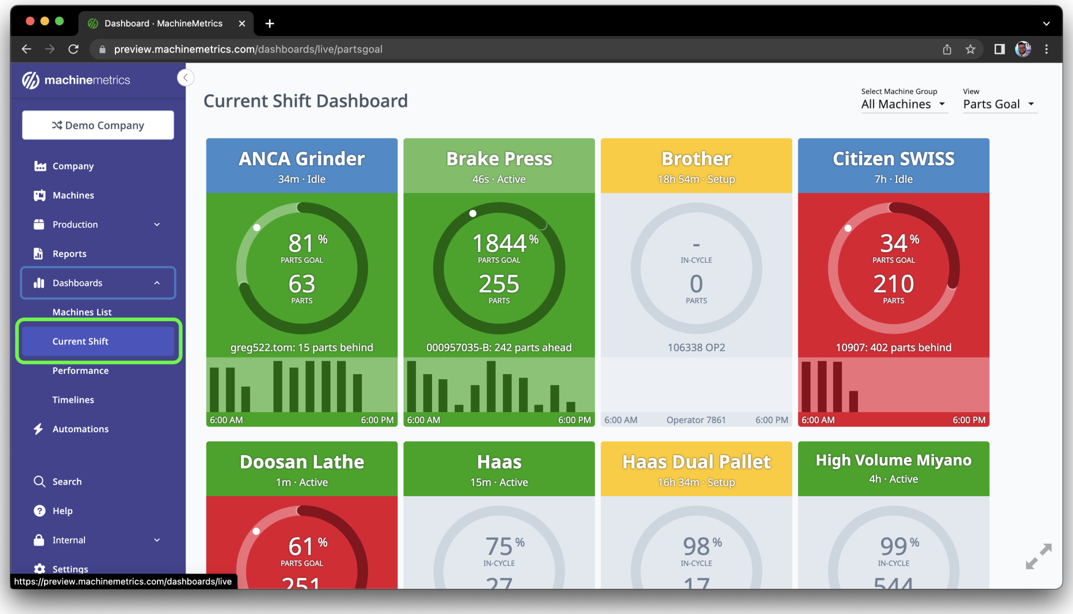

Figure 3.1: MachineMetrics dashboard example [23]

3.3 Google Analytics

Google Analytics allows us to look at our data across platforms — web and app — to understand the full journey of our users. We've been able to cut our reporting time by 50% [24].

Google Analytics is an indispensable tool for webmasters and digital marketers, offering a comprehensive suite of features designed to track and analyse website traffic and user behaviour. At its core, Google Analytics enables website owners to understand how visitors interact with their sites, providing insights into page views, session duration, bounce rates and more. This wealth of data can be used to optimise website content, structure and marketing strategies to better meet users' needs and preferences.

Beyond simple traffic analysis, Google Analytics offers advanced segmentation, allowing users to drill down into the data and uncover patterns between different demographics, devices and traffic sources. Conversion tracking is another key feature, making it possible to measure the effectiveness of different marketing campaigns and achieve specific goals such as sales, sign-ups or other desired actions. Integration with other Google services such as AdWords and Google Search Console further enhances its usefulness, enabling seamless cross-platform analysis and optimisation. With its easy-to-use interface and powerful analytical capabilities, Google Analytics democratises data analysis, making it accessible not only to data scientists, but also to marketers and business owners, empowering them to make data-driven decisions to drive growth and improve their online presence.

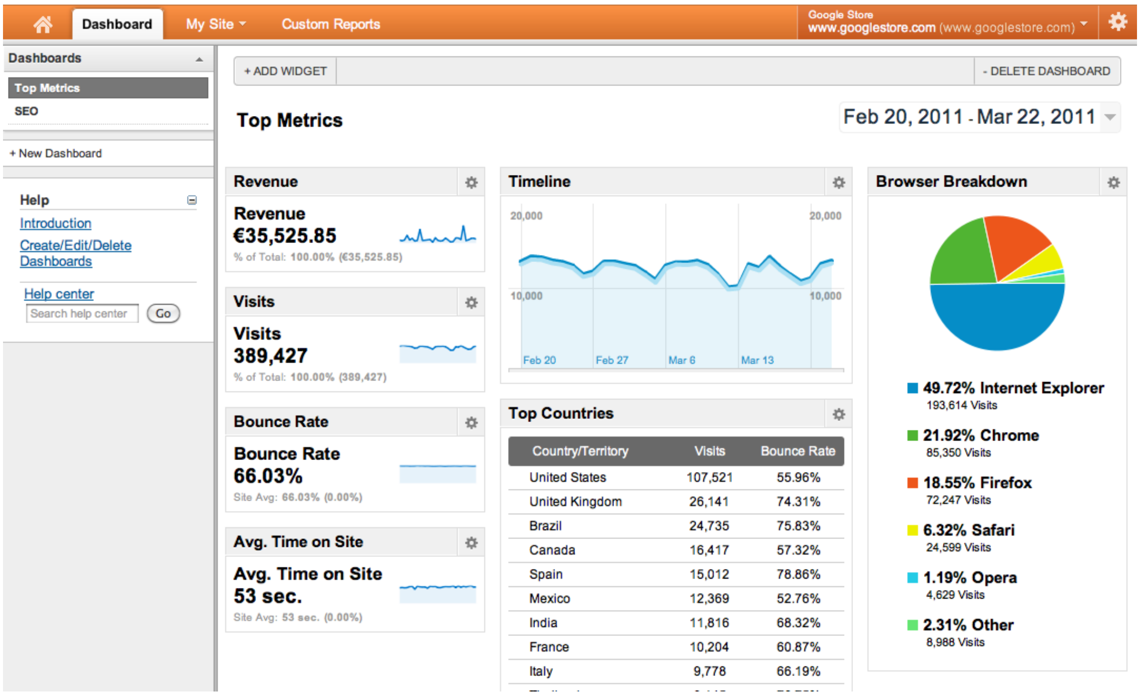

Figure 3.2: Google Analytics dashboard example [25]

3.4 Grafana

Grafana open source is open source visualization and analytics software. It allows you to query, visualize, alert on, and explore your metrics, logs, and traces no matter where they are stored. It provides you with tools to turn your time-series database (TSDB) data into insightful graphs and visualizations. [26]

Grafana offers several features that make it a popular choice for creating dashboards [27]:

- **Unify your data, not your database**: Grafana doesn't require you to ingest data to a back-end store or vendor database. Instead, Grafana takes a unique approach to providing a "single-pane-of-glass" by unifying your existing data, wherever it lives. With Grafana, you can take any of your existing data- be it from your Kubernetes cluster, raspberry pi, different cloud services, or even Google Sheets- and visualize it however you want, all from a single dashboard.

- **Data everyone can see**: Grafana was built on the principle that data should be accessible to everyone in your organization, not just the single Ops person. By democratizing data, Grafana helps to facilitate a culture where data can easily be used and accessed by the people that need it, helping to break down data silos and empower teams.

- **Dashboards that anyone can use**: Not only do Grafana dashboards give insightful meaning to data collected from numerous sources, but you can also share the dashboards you create with other team members, allowing you to explore the data together. With Grafana, anyone can create and share dynamic dashboards to foster collaboration and transparency.

- **Flexibility and versatility**: Translate and transform any of your data into flexible and versatile dashboards. Unlike other tools, Grafana allows you to build dashboards specifically for you and your team. With advanced querying and transformation capabilities, you can customize your panels to create visualizations that are actually helpful for you.

Figure 3.3: Grafana dashboard example [27]

3.5 Tools comparison

In the section we are going to compare all the tools mention before in order to have an ideia of the pros and cons of each other.

Table 3.1: Tool Comparison: Grafana, Google Analytics, and MachineMetrics

Feature/Tool	**Grafana**	**Google Analytics**	**Machine Metrics**
Primary Use Case	Visualization of metrics and monitoring	Web analytics	Industrial IoT and machine monitoring
Data Sources	Supports various data sources	Primarily web and mobile app data	Industrial equipment data

Continued on next page

Table 3.1 – continued from previous page

Feature/Tool	Grafana	Google Analytics	Machine Metrics
Real-Time Data	Yes	Yes, with some limitations	Yes
Customization	High (open-source and plugins available)	Medium (limited to provided features)	Medium (specific to manufacturing data)
Pricing	Free open-source; paid enterprise version	Free version; paid version (GA 360)	Subscription-based, custom pricing
Ease of Use	Moderate (technical skills required)	Easy to moderate (user-friendly UI)	Moderate (specific industry focus)
Analytics Features	Advanced metrics, alerting, dashboards	Traffic, user behavior, conversion rates	Machine efficiency, predictive maintenance

3.6 Summary

This section presents an evaluation of three key analytics and monitoring platforms. The following three platforms were considered: Grafana, Google Analytics, and MachineMetrics. Each tool offers a distinctive range of functionalities that cater to a variety of monitoring and data analysis needs, which are crucial for diverse application requirements.

Grafana is renowned for its comprehensive visualisation capabilities, encompassing a vast array of data sources. It offers real-time data visualisation. Its flexibility in customisation is attributable to its open-source nature, which allows users to modify and integrate with a variety of plugins. Despite providing powerful tools for data monitoring, Grafana requires a moderate level of technical skill to fully leverage its advanced features.

Google Analytics is renowned for its excellence in web and mobile analytics. It offers users a wealth of insights into user interactions and behavior, and is particularly valued for its user-friendly interface and comprehensive data analytics capabilities. These include real-time data, although there are some limitations. While the software is straightforward to use for basic functionalities, more advanced features can pose a learning curve for those new to the software.

MachineMetrics specialises in the field of industrial IoT and machine monitoring, making it an optimal choice for applications requiring the monitoring of heavy industrial equipment. It offers a suite of tailored analytical features, including machine efficiency tracking and predictive maintenance capabilities, and it is a subscription-based service. While it is equipped to provide real-time data analysis, it is especially suited to users who have specific manufacturing data needs.

In conclusion, following the examination of the three tools previously outlined, the decision was taken to proceed with the development of Factory Pulse.

Chapter 4

Problem Statement

This chapter presents the requirements specification of the application Factory Pulse. Furthermore, this chapter delineates the types of users that can potentially exist, presents user stories, and outlines the more technical requirements.

4.1 Factory Pulse

Factory Pulse is an online application that has been developed to address a need in the industry for a solution that provides users with greater visibility and control over their operations. One of the key benefits of Factory Pulse is its accessibility, allowing users to monitor and control their factory operations from anywhere in the world with an internet connection. This accessibility is particularly advantageous for decision-makers and managers who need the flexibility to maintain oversight of their operations without being physically present at the factory site.

As an online application, Factory Pulse ensures that updates and data are transmitted and received in real time across all devices, regardless of the user's location or circumstances. This global accessibility enables a flexible work environment, and it is of paramount importance in emergency situations or when prompt decision-making is required based on the latest operational data.

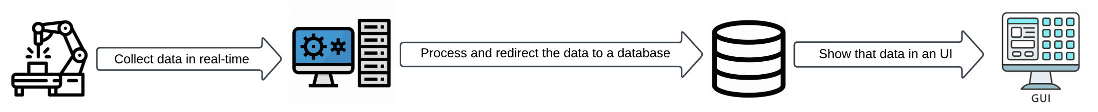

Figure 4.1: Diagram illustrating the data flow

The Figure 4.1 displays a diagram of the data flow in the system. The data is collected from machines, processed in the workstation, and then redirected to a database. Finally, the data is presented on a graphical interface.

4.2 Problem Definition

This study addresses the primary challenge of managing databases in real time, which necessitates the ability to represent this data visually through an online dashboard. Unlike a static tool, the dashboard must be an interactive interface capable of providing real-time monitoring and control capabilities across the entire factory. It allow users to access the factory environment in a granular manner, from broad overviews to specific details.

The real-time data management and visualisation are of critical importance, as they facilitate immediate responsiveness to changes and anomalies in the factory's operations. The ability to access and analyse data in real time via a user-friendly graphical interface supports efficient decision-making and enhances operational control. Consequently, this study seeks to explore and address the challenges of designing and implementing a robust, real-time database management system that integrates with an intuitive, comprehensive dashboard for optimal factory oversight.

4.3 Requirements Specification

This section presents the requirements specification that guided the development of the dashboard to effectively address the problem. It includes the users, user stories, and technical requirements.

4.3.1 Users

A visual representation of existing user categories is presented in Figure 4.2. Upon analysis of Table 4.1, it is possible to gain a general overview of the permissions granted to each user.

Table 4.1: Types of users that exist

Identifier	**Description**
Guest	The unauthenticated user can sign up and sign in
Standard	The authenticated user with the lowest permissions as a client
Supervisor	The authenticated user with medium permissions as a client
Leader	The authenticated user with the highest permissions as a client
Admin	The authenticated user who has control over the application's functionalities

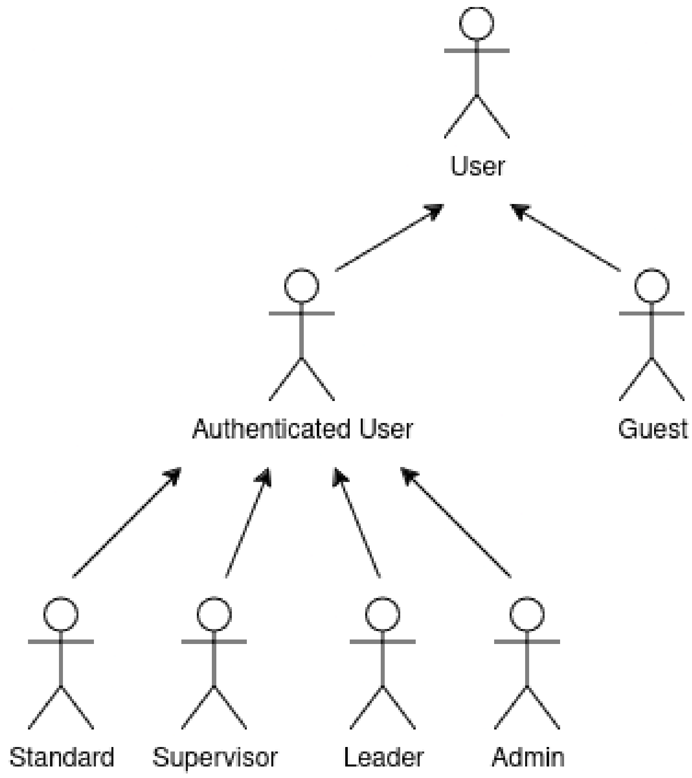

Figure 4.2: Types of users diagram

4.3.2 User Stories

User stories have been widely accepted as artifacts to capture the user requirements in agile software development. They are short pieces of texts in a semi-structured format that express requirements. Each user story follows the form: "As a [who], I want to [requirement], so that [reason]." User stories capture the user's needs, desires, and expected outcomes [28].

The forthcoming user stories are divided into three distinct categories, which correspond to different user roles within the application. The user categories are defined as follows: Guest, Client, and Admin.

4.3.2.1 Guest's user stories

The Guest section delineates the functionalities available to non-registered users who may browse or interact with certain aspects of the platform.

Table 4.2: Guest's user stories

Identifier	Name	Priority	Description
US01	Sign in	High	As a guest, I want to authenticated into the system, so that I can access the application
US02	Sign up	High	As a guest, I want to register an account in the system, so that I can access the application
US03	Change Password	High	As a guest, I want to change the password of my account, so that I can access the application

4.3.2.2 Client's user stories

The Client section is further subdivided to detail varying permission levels: Standard, Supervisor, and Leader. Each of these levels provides different degrees of access to data and features based on the user's role. This ensures that clients receive the appropriate access to data and features commensurate with their responsibilities. The Standard and Supervisor roles both have access to the same set of menus; however, the Supervisor role has greater access to more detailed information due to its elevated permissions.

Table 4.3: Client's user stories

Identifier	Name	Priority	Description
US04	See Dashboard page	High	As a client, I want to access the dashboard page to view essential information about my company
US05	See Account Page	High	As a client, I want to view my account details to edit any related information
US06	See Company information	High	As a client, I want to view the company information that I have permission to see, to view a high-level overview of the factory quickly
US07	See Cell information	High	As a client, I want to be able to have access to the information about the cell that I have permission to see, to ensure it's operating correctly

Continued on next page

Table 4.3 – continued from previous page

Identifier	Name	Priority	Description
US08	See Machine information	High	As a client, I want to view machine details that I have permission to see, to respond promptly to any alarms that are triggered
US09	See Schedule Page	Medium	As a client, I want to access my schedule page to review upcoming meetings and maintenance services
US10	Edit Account Page	Medium	As a client, I want to edit my account details to update my personal information

Table 4.4: Client's user stories with Leader permission

Identifier	Name	Priority	Description
US11	See all my employees	High	As a leader, I want to be able to see all my employees, so that I manage they roles or delete their accounts
US12	Create my employees accounts	High	As a leader, I want to be able to create my employees accounts, so that they can have access to the application with the company information
US13	Edit my employees roles	High	As a leader, I want to be able to change my employees role, so that they I can give more or less permissions to them
US14	Delete my employees accounts	High	As a leader, I want to be able to delete my employees account, so that if they leave the company, they stop having access to the company's information
US15	Edit Account Page	Medium	As a leader, I want to be able to see my account details, so that I edit any information related to my account

4.3.2.3 Admin's user stories

Finally, the Admin section encompasses comprehensive control over the entire application. It is here that administrators can manage users, configure settings, and oversee all operational facets.

This structure ensures a focused and role-specific approach to feature development, aligning with the needs and privileges of each user category.

Table 4.5: Admin's user stories

Identifier	Name	Priority	Description
US16	See Dashboard page	High	As a admin, I want to access the dashboard page to view essential information about all the companies
US17	See Companies information	High	As a admin, I want to view the companies information, so that I can have
US18	See Cells information	High	As a admin, I want to access information about the cells of each company, to ensure the correct operation
US19	See Machine information	High	As a admin, I want to view machine details, to respond promptly to any alarms that are triggered
US20	Create Company	High	As a admin, I want to be able to create a company
US21	Delete Company	High	As a admin, I want to be able to delete any company existing
US22	Edit Company information	High	As an admin, I want to be able to edit a company's information, including its leaders, so that I can keep the information up to date and accurate
US23	Delete Cell	High	As an admin, I want to be able to delete any cell from any company, so that I can manage resources and operational efficiency
US24	Create Cell	High	As an admin, I want to be able to create a cell, so that I can organize and structure our operations more effectively
US25	Delete Machine	High	s an admin, I want to be able to delete any machine from any cell, so that I can manage equipment more effectively and ensure operational safety
US26	Create Machine	High	As an admin, I want to be able to create a machine, so that I can enhance operational capabilities and productivity

Continued on next page

Table 4.5 – continued from previous page

Identifier	Name	Priority	Description
US27	See all Clients	High	As a admin, I want to be able to see all my clients and their public information
US28	Delete a Client	High	As a admin, I want to be able to delete all the clients registered in my application
US29	See Schedule Page	Medium	As a admin, I want to be able to schedule meetings and maintenance services for clients
US30	See Account Page	Medium	As a admin, I want to view my account details to edit any related information

4.3.3 Technical Requirements

The technical specifications for the system, as detailed in Table 4.6, are as follows:.

Table 4.6: Technical Requirements

Identifier	Name	Description
TR01	Usability	The system should be user-friendly to be accessible to all users
TR02	Security	The system should have adequate security measures for sensitive data and secure authentication for access
TR03	Scalability	The system should be scalable in order to accommodate the growing data volume and user base
TR04	Integration	The system should be capable of integrating with other software tools and systems, including APIs and external data sources, in accordance with the user's specifications

4.4 Summary

In this chapter, we present Factory Pulse, our application designed to enhance real-time database management in IoT environments. We commenced by defining the core problem that Factory Pulse is designed to address.

The Problem Definition section elucidated the challenges encountered by contemporary factories in processing voluminous data and the necessity for systems that can process this data expeditiously and accurately. We emphasised the significance of real-time data handling for the purpose of enhancing decision-making and operational efficiency.

The Requirements Specification section then proceeded to delineate the various requirements for Factory Pulse. This section was divided into three main parts:

- Users: The primary users of the system were identified, and they were categorised into different roles, such as managers and operators. Each of these roles was assigned specific needs and interactions with the application.
- User Stories: The desired features and functionalities of Factory Pulse were outlined through the use of user stories. The user stories provided a narrative of the interactions that different users would have with the system, encompassing aspects such as data retrieval, visualization, and project management.
- Technical Requirements: A discussion of the technical specifications necessary for implementing Factory Pulse was held, including system architecture, data security, scalability, and integration with existing systems.

This chapter established the foundation for an understanding of the objectives and specifications of our application, thereby paving the way for its development and implementation.

Chapter 5

Implementation

This chapter presents a review of the methods for implementing and meeting the primary requirements of the project, supported by figures and tables for clarity. It is crucial to acknowledge that there are various methodologies for implementing a project. The selected approach aligns with the recommendations of the project manager and work coordinator.

5.1 Tools and Technologies

In order to develop this project, a number of tools were employed in order to create a sustainable application that met the initial requirements. The table below provides a clear and concise explanation of the rationale behind the selection of each tool, including both software and hardware. The following chapter will provide a detailed account of each tool, elucidating their relevance to the project. Table 5.1 provides a comprehensive list of the tools utilized during the project's development, accompanied by a detailed explanation of their respective purposes.

Table 5.1: Project implementation tools

Tool	**Objective**
NodeJS	Backend of the Web Application
ReactJS	Frontend of the Web Application
Firebase	Database
Overleaf	Document writing

5.1.1 Backend

The backend of the application is developed using *NodeJS*, a powerful and efficient server-side runtime environment. This environment executes JavaScript code in a non-blocking manner, which makes it highly scalable and suitable for real-time applications [29]. Additionally, *NodeJS* is often employed with *ReactJS*. This combination provides a robust platform for building microservices and handling numerous simultaneous connections with high throughput.

NodeJS facilitates integration with a variety of databases and APIs, allowing efficient data manipulation and retrieval. The backend handles all core business logic, including user authentication, authorisation, and data validation. RESTful APIs are created using *ExpressJS*, a minimal and flexible *NodeJS* web application framework [30]. This setup facilitates maintenance and scalability of the application, as it is easily adaptable to changes.

The backend communicates with the Firebase database to store and retrieve user data and other essential information. The Firebase Admin SDK is employed to securely administer and access the database, thereby ensuring that all operations comply with the security rules defined in the Firebase console [31]. This architectural approach ensures that the backend remains performant and reliable under varying loads.

5.1.2 Frontend

The frontend of the application is constructed using *ReacJS*, a prominent JavaScript library for developing user interfaces. *ReacJS* is selected for its component-based architecture, which facilitates the reuse of components and the efficient management of the application's state [32]. This modularity allows for the development of complex and dynamic interfaces with ease.

ReacJS enhances the user experience by providing a fast and responsive UI. The virtual DOM (Document Object Model) employed by *ReacJS* ensures that updates are efficient, as only the parts of the DOM that require modification are re-rendered [33]. This results in a more fluid and engaging user experience.

The frontend application interacts with the backend through RESTful APIs. Axios, a promise-based HTTP client, is used to handle API requests and manage data fetching [34]. React Router is employed for the management of navigation and the creation of a single-page application (SPA) experience [35]. Furthermore, Redux is utilised for state management, ensuring that the application state is predictable and easy to debug [36].

5.1.3 Database

The application employs Firebase as its database solution, leveraging its real-time capabilities and comprehensive feature set to manage and store data. The management of the database is a crucial aspect of this work, as it ensures the integrity, security, and availability of the data, which are essential for the application's performance and reliability.

The Firebase Realtime Database offers a flexible, *NoSQL* cloud database that enables data to be synchronised in real-time with all connected clients [37]. This guarantees that all users have access to the most up-to-date data without delay, which is crucial for maintaining a consistent user experience across the application.

The integration of Firebase with *NodeJS*, facilitated by the Firebase Admin SDK, enables secure and efficient communication between the backend and the database. The SDK provides server-side access to Firebase services, including authentication, database management, and cloud messaging [31]. This integration ensures that data operations are performed in an efficient and secure manner, in accordance with the application's security protocols.

Furthermore, Firebase offers Firestore, a scalable database that is suitable for use in mobile, web, and server development. Firestore's comprehensive querying capabilities and real-time updates render it an optimal choice for applications that necessitate synchronised data across multiple users [38]. The utilisation of Firestore enables the application to process complex queries and real-time data updates, thereby enhancing the overall user experience.

The combination of Firebase Realtime Database and Firestore allows the application to handle various data management scenarios efficiently, supporting both real-time synchronisation and complex queries.Effective database management is of paramount importance in this context, as it ensures that the database layer remains robust, scalable, and easy to maintain. This, in turn, contributes to the success of the application.

5.2 User Interface

This section presents the user interface for all possible user types within the application.

5.2.1 Unauthenticated users

The user interface has been designed for unauthenticated users with the objective of providing a straightforward and focused experience, which includes access only to sign-in, registration and reset password pages. This approach ensures that users are prompted towards authentication before they are able to access the core functions of the application.

The sign-in page, Figure 5.1, serves as the initial point of access for existing users seeking to authenticate their identities within the application. The page contains fields for the user's email address and password, as well as a button that allows the user to submit the login form. In the event that the user inputs incorrect credentials, an error message is displayed to assist them in correcting their input. Additionally, the page contains a button that provides access to the registration page.

Key features of the Sign-In Page:

- Email and Password Fields: Simple and clear fields for user credentials.
- Submit Button: Prominently placed to ensure easy access.
- Register Button: Clicking on the button will direct the user to the registration page.
- Reset Password Button: Clicking on the button will direct the user to the reset password page.
- Error Handling: Provides feedback if login fails due to incorrect credentials.

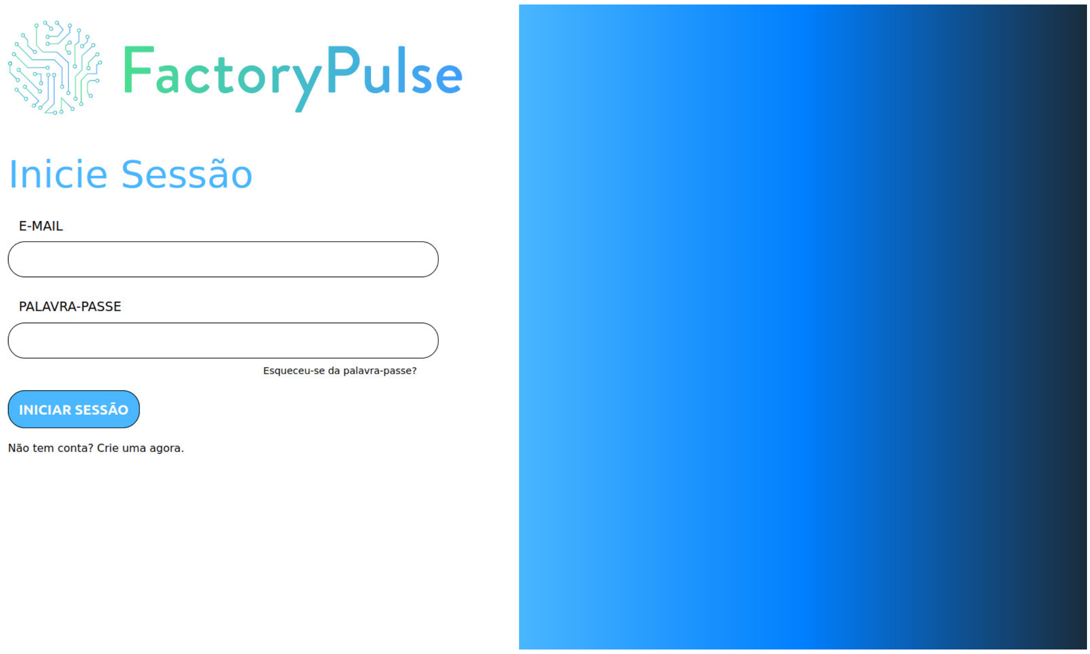

Figure 5.1: Sign in page

Registration is the process by which new users are permitted to establish an account on the website. However, registration is only partially complete until it has been formally approved by an administrator. This page collects essential information from the user, including name, email address, and password, as illustrated in Figure 5.2. Once submitted, the user will receive a verification email, which serves as a means of confirming their account (Figure A.3). After this initial confirmation, the user will be asked to provide additional information, such as their address, phone number, and NIF (Figure A.4). Subsequently, the user is redirected to the account activation page where they must provide the activation code issued by the administrator to complete the account activation process (Figure A.5).

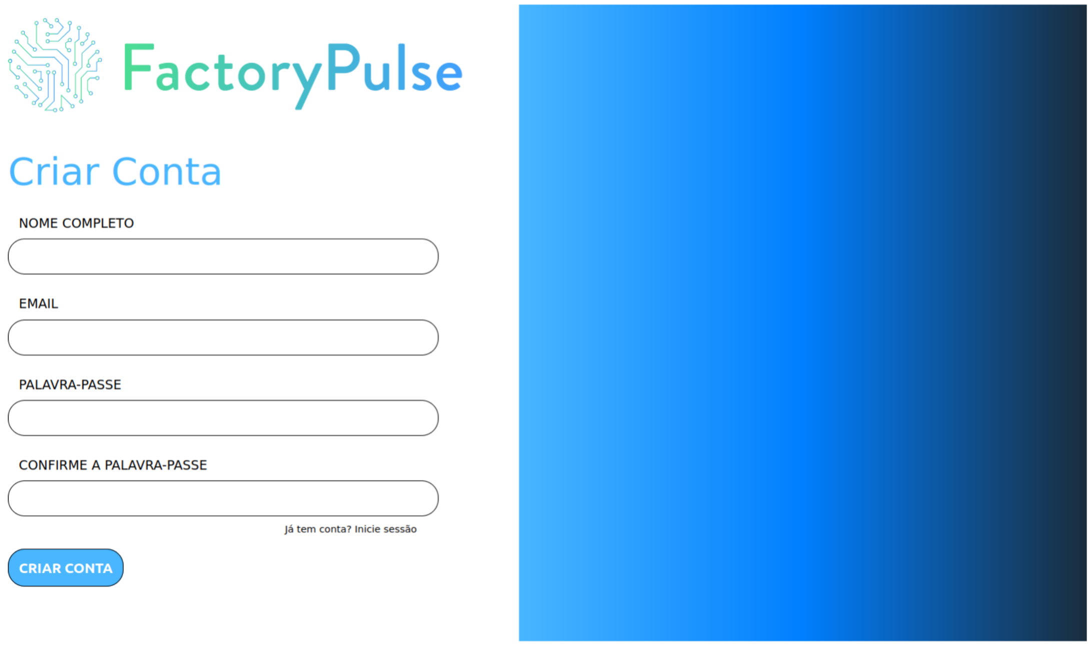

Figure 5.2: Sign up page

In the event that the user in question has an existing account but has forgotten their password, they may recover it by providing their email address (Figure A.1). A subsequent email will then be sent to the user with a link that opens a new page where the user may create a new password (Figure A.2).

The user interface for unauthenticated users is intentionally limited to the sign-in and register pages in order to streamline the authentication process. This approach ensures that users are authenticated prior to accessing the main features of the application, thereby maintaining security and control over user access.

5.2.2 Authenticated users - Client

Upon login, authenticated users are presented with a dashboard that provides an overview of their associated companies and access to various functionalities within the application. The Dashboard page (Figure 5.3) is the central hub for authenticated users, displaying a summary of the companies to which they are linked and providing quick access to detailed information and actions related to those companies.

The dashboard page is comprised of three primary elements: the sidebar, breadcrumbs and the header.

- Sidebar: The sidebar offers users a convenient via to all the primary pages within the application, including the dashboard page, the account page, the calendar page, and, in the case of a leader, the workers page.
- Breadcrumbs: Breadcrumbs are a navigational aid located at the top of the dashboard, just below the header, that displays the user's current location within an application's hierarchical structure. The individual elements of the breadcrumb trail are linked and, as such, permit users to swiftly navigate their way back to previous pages.
- Header: The header is displayed at the top of each page and contains a search bar, a help button, a history of alarms and the current user name. Clicking on the user name redirects the user to the account page.

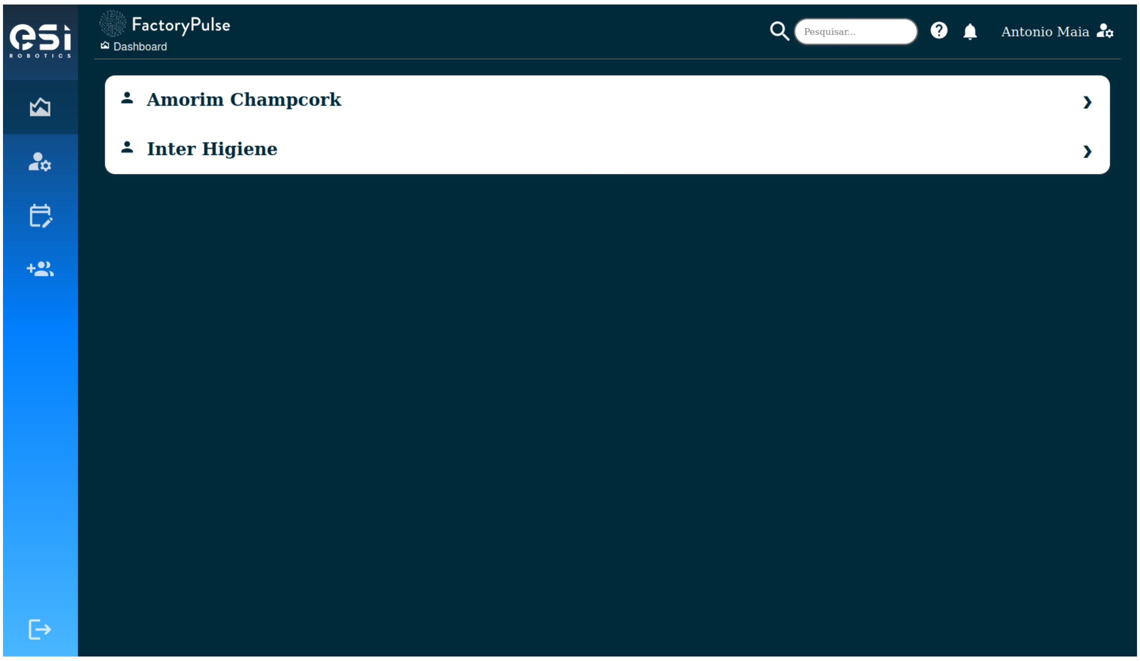

Figure 5.3: Dashboard page

The sidebar contains several functional buttons that provide users with quick access to specific sections of the application. These buttons have been designed to enhance usability by offering direct links to important functionalities.

Clicking the "Conta" button directs the user to their account page (Figure A.6), where they can view their account details and edit certain aspects of their personal information, such as contact details and profile settings. This functionality enables users to keep their information up-to-date and manage their personal settings in a convenient manner.

The "Agenda" button directs the user to their personal calendar page (Figure A.7). This page allows the user to view their own meetings and maintenance schedules. This feature is of great importance for maintaining an organised schedule, as it ensures that the user does not miss any important events or tasks that are scheduled.

For users in leadership roles, an additional button will be displayed, labelled "Funcionários". This provides access to the employee management page (Figure A.8). On this page, leaders can view and create employee profiles, delete employee accounts, and assign roles. This page is crucial for the effective management of teams, enabling leaders to oversee team members and activities in a manner that is conducive to efficient team coordination and communication.

Following these fixed elements, upon clicking to expand a company, users are presented with a detailed view of the various cells within that factory (Figure 5.4). This expanded view provides crucial insights into each cell's status and performance, enabling users to make informed decisions promptly. The detailed view includes essential information such as production metrics, operational status, and other relevant data for each cell. This information is displayed in a clear and organised manner, making it easy for users to assess the current state of their operations at a glance. In order to further enhance the usability of the application, a colour-coding system has been employed to indicate the operational status of each cell. This visual aid allows users to quickly identify any issues that may require attention. The colour codes are as follows:

- Green: Indicates that the cell is operating normally and is currently working. This status reassures users that their operations are running smoothly without any interruptions.
- Red: Indicates that the cell is stopped or turned off.

The colour-coding system, when combined with the detailed information provided for each cell, ensures that users can efficiently monitor their factories and respond promptly to any issues.

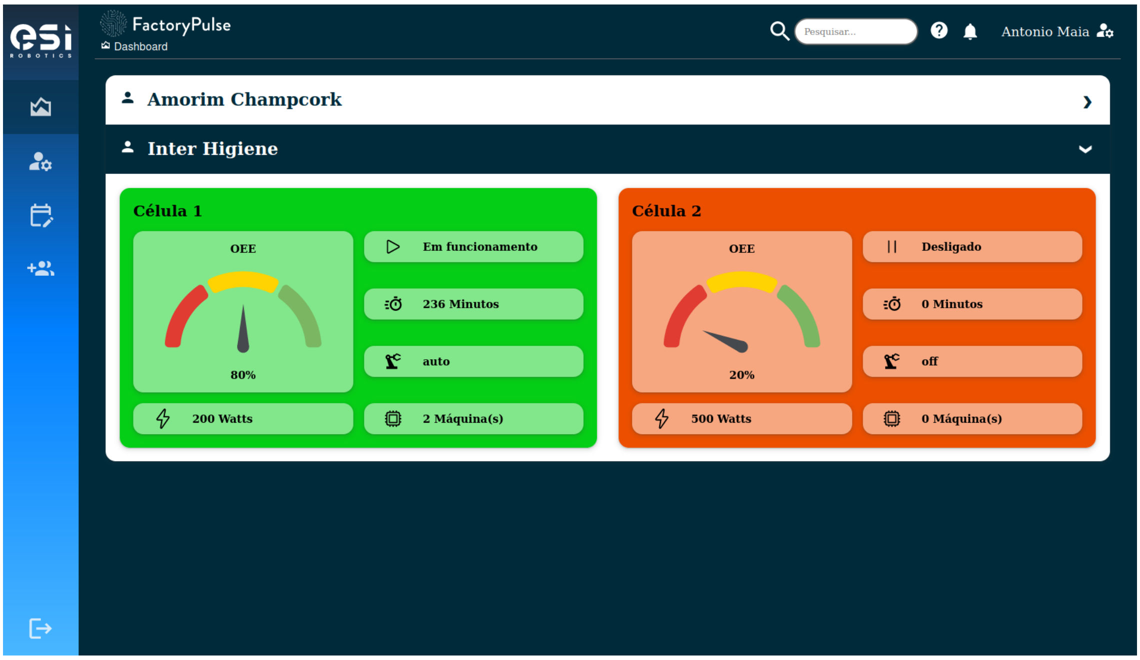

Figure 5.4: Cells page

In order to provide users with comprehensive insights into their operations, the application enables users to delve deeper into the specific details of each cell. Upon clicking on a cell within the expanded company view, users are taken to a detailed page dedicated to that particular cell (Figure 5.5). This page offers an in-depth examination of the cell's performance and the machines which comprise it.

The cell detail page presents more detailed information on the cell's operational characteristics, offering users access to comprehensive metrics. This data is presented in a clear and readily comprehensible format, through graphical and tabular representations, enabling users to observe performance trends and identify potential areas for improvement. Furthermore, the cell detail page provides a comprehensive list of all machines included in the cell. Each machine entry includes essential details such as the machine's status and the active cycle.

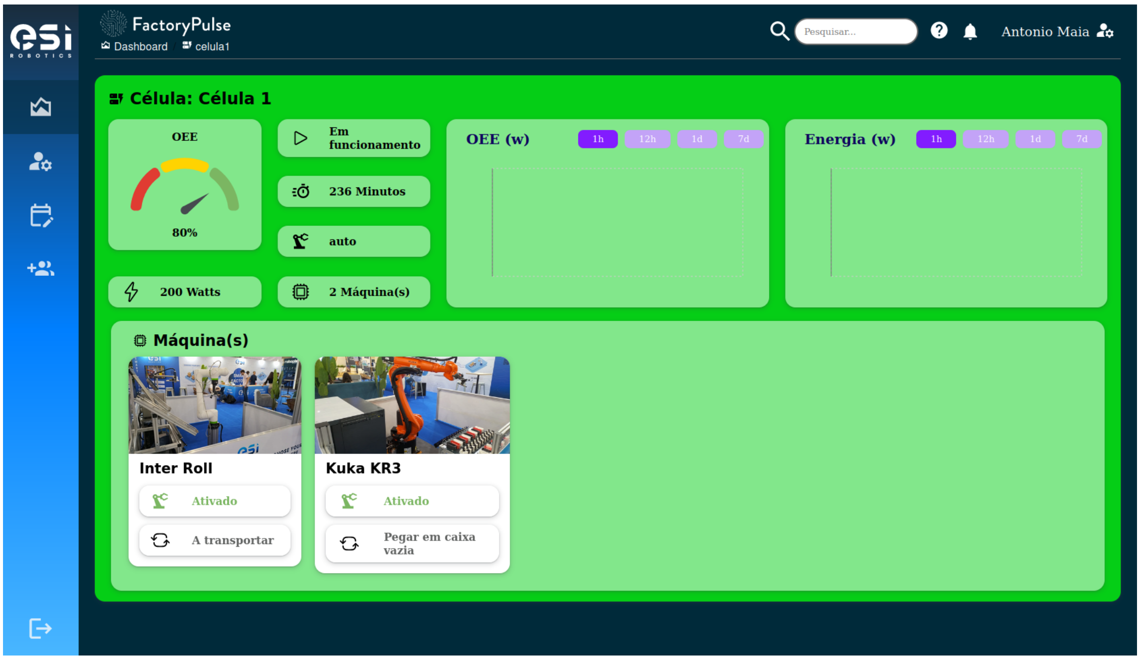

Figure 5.5: Cell details page

The machines may be of two distinct varieties, namely, robots or transporters. Each of these types of machine has its own interface, comprising tables and graphics.

Figure 5.6: Robot page

In the application, a robot page provides comprehensive and detailed information about the status and performance of the robot (Figure 5.6). The panel at the top of the page contains general information about the machine, including its current status, any active alarms, and the cycle in execution. This section provides an overview of the robot's operational state. Adjacent to this panel is a real-time Overall Equipment Effectiveness (OEE) graph, which provides users with an immediate understanding of the robot's efficiency. Subsequently, a detailed table is presented, which lists all alarms with their respective times and statuses. This enables users to track and manage any issues that may arise. One of the most notable features of this page is a 3D model developed in Unity that imitates the movements of the robot in real time (Figure A.9). This offers a visual representation of the robot's actions, enhancing user understanding and monitoring. Furthermore, the page includes several graphs that provide further insights into the robot's operations. The graphs illustrate the number of operations performed and their outcomes, including successful, warning, and failure cases. This allows users to assess the robot's reliability and performance over time. Another graph tracks OEE over a specified period, enabling users to analyse trends and identify areas for improvement. Finally, a graph displaying the robot's energy consumption over time enables users to monitor and optimise energy usage (Figure A.10).

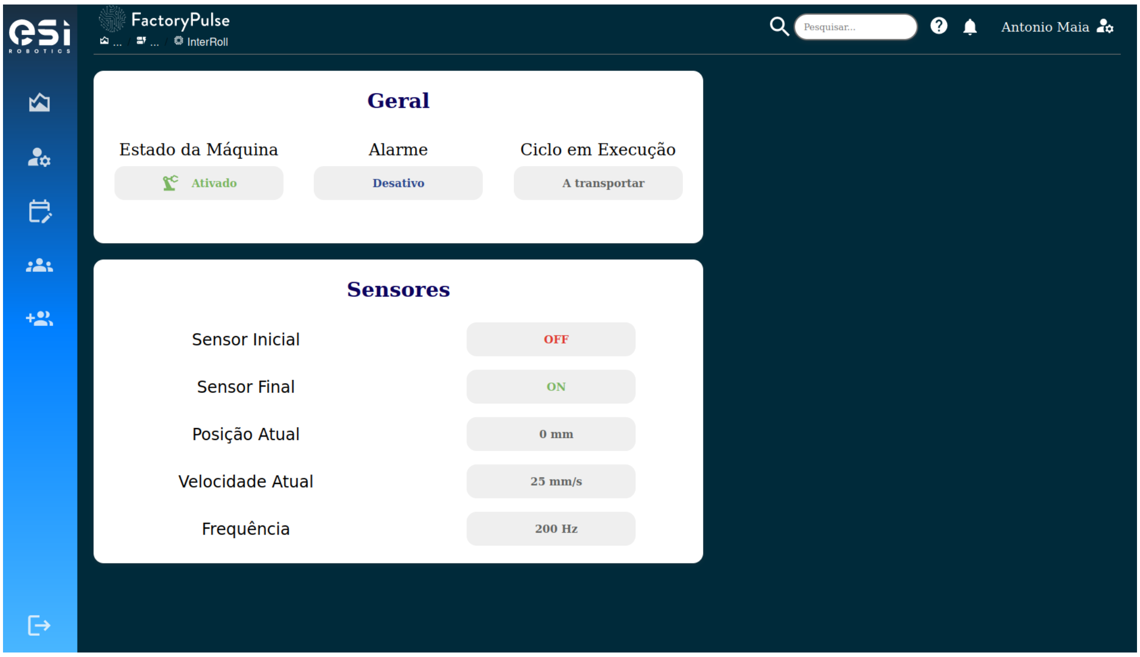

Figure 5.7: Transporter page

The transporter page provides comprehensive data regarding the operational status and performance metrics of the transporter. At the top of the page, a general information panel displays the machine's current status, any active alarms, and the cycle in execution. This section provides a concise overview of the transporter's operational state (Figure 5.7).

The section dedicated to specific sensor data is located beneath the general information panel. These include metrics such as the initial and final sensor positions, the actual position (if available), the velocity, and the frequency. The initial and final sensor values indicate the commencement and conclusion of the transporter's range of motion, while the actual position provides real-time updates on its location. The velocity and frequency metrics provide further insight into the operational speed and cycle frequency of the transporter.

5.2.3 Authenticated users - Admin

The admin interface of the application features a dedicated page for viewing and managing all registered clients and their workers. This page presents a comprehensive overview of each client's information, including their email address, activation code, role, account status, unique identifier (UID), phone number, and NIF (Figure 5.8).

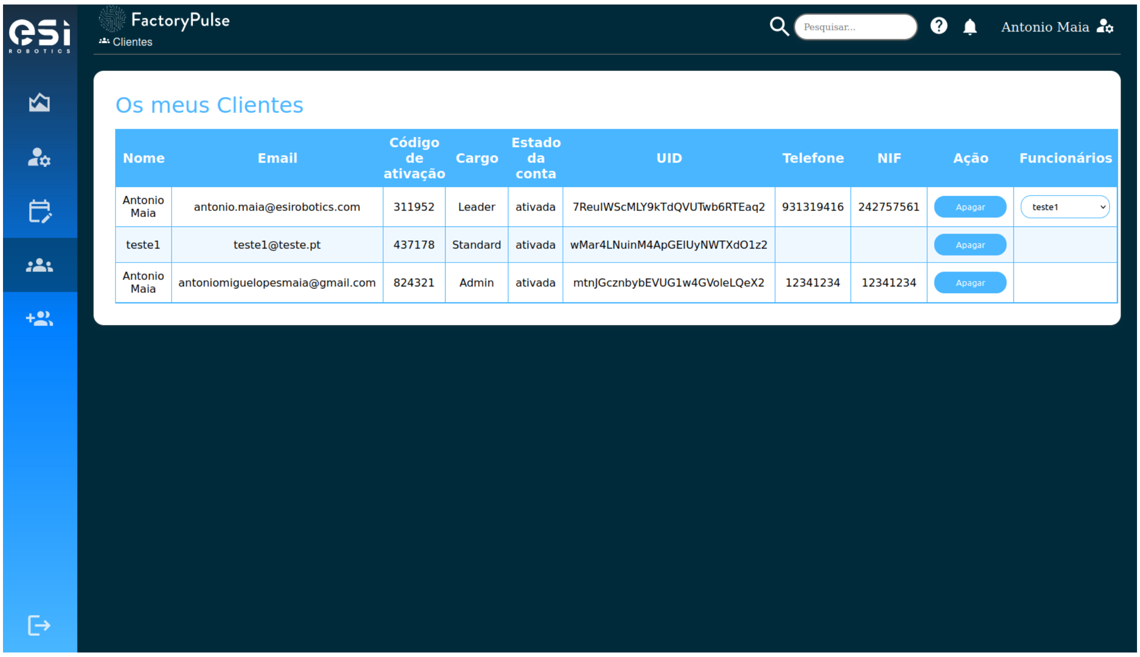

Figure 5.8: My Clients page

The interface has been designed for ease of use, allowing administrators to efficiently navigate through client records. Furthermore, the interface includes action buttons that enable the administrator to swiftly and effectively manage the client records. In addition, a drop-down menu allows for the display of clients' workers, thereby simplifying searches for specific workers, while also enabling the deletion of any desired worker.

The calendar page in the administrative interface enables scheduling and management of maintenance tasks and meetings. This page provides a user-friendly form on which the administrator may specify the details of the appointment, including the subject, start and end dates, client and assigned technician (Figure B.1).

The dashboard page within the admin interface serves as a central hub for the management of companies and their associated operational cells. In the upper right-hand corner of the page is an icon resembling a gear, which enables the admin to create, delete or edit company profiles (Figure B.2). The functionality of this icon ensures that the administrator is able to maintain an up-to-date and accurate representation of the organisational structure as it exists within the application at any given moment.

Upon selecting the gear icon within the dashboard, the interface is transformed into a more comprehensive management view. This view provides the administrator with the ability to create, delete, and edit companies and their respective cells, thereby enhancing their capacity to organise and oversee the operational structure (Figure B.3).

Clicking on the "Add Company" button initiates the completion of a comprehensive form. This form requires the input of crucial company data, including the company name, NIF, address, email, and phone number (Figure B.4). This form ensures that all necessary details are captured for the creation of accurate company records. Once the company has been created, the administrator is then able to edit the aforementioned information through a similar form interface (Figure B.5). In the context of the edit mode, an additional field becomes available, which allows the administrator to associate one or more leaders with the company. This stage is of vital importance in establishing a clear organisational hierarchy and accountability, as it enables the linking of leaders to their respective companies. Additionally, the admin can create and manage operational cells within each company. To do so, simply click on the plus icon within the panel for the company of interest. This will open a modal where the cell name can be entered (Figure B.6). The modal is straightforward in its design to ensure that the process of adding new cells is quick and efficient. If the admin needs to delete a cell or company, they can simply click the delete button, which will trigger a confirmation modal (Figure B.7). This modal serves as a safeguard, asking the admin to confirm the deletion to prevent accidental removals.

The gear icon, located on the cell details page, enables the admin to modify the machines associated with that cell (Figure B.8).

By clicking on this icon, the admin can delete existing machines or create new ones (Figure B.9). When adding a new machine, a modal window appears, prompting the admin to enter the machine's name and select its type. This type can be either a robot or a transporter (Figure B.10). This functionality ensures that all machines are properly categorised and integrated into the cell's operations.

5.3 Firebase

Firebase provides a number of highly useful services which have been integrated into this application. These include the Real-Time Database, the Firebase hosting service, and the Firebase Authentication Service. In this section, we will examine how each of these services has been applied.

5.3.1 Real-Time Database

In this section, we will examine the architecture and internal organisation of our Firebase real-time database (RTDB), with particular emphasis on the tables that represent the data of companies, cells, machines, and users. The database is designed with the intention of managing complex relationships and data hierarchies within an industrial environment. The data table will be discussed first, followed by the users table, and finally, an overview of the relationship between these two tables will be provided.

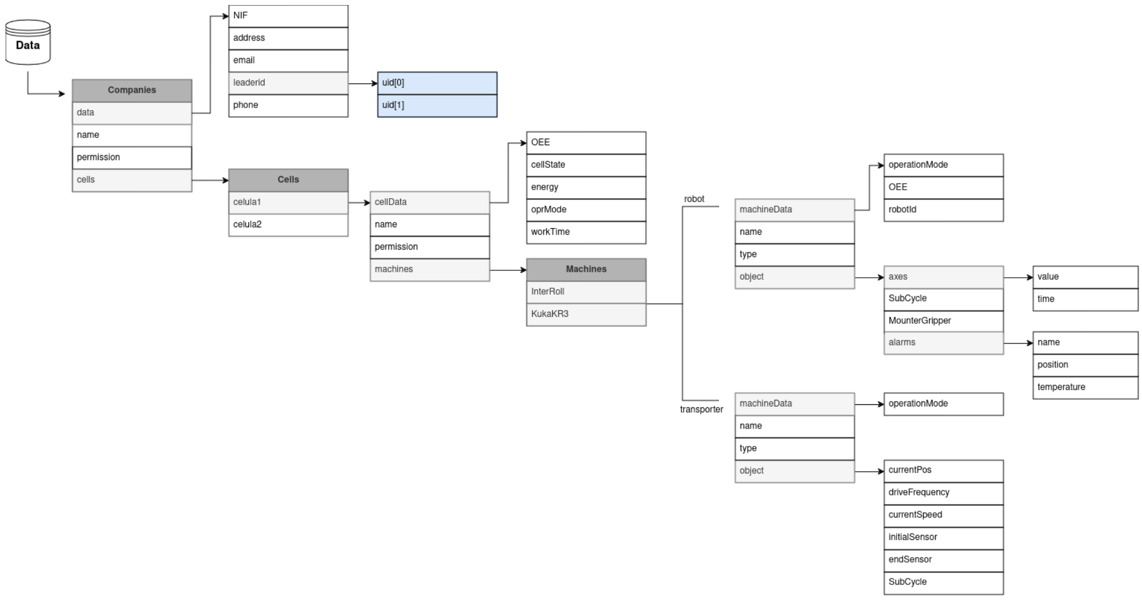

Figure 5.9: Data table diagram

Figure 5.9 illustrates the core structure of the data table in our Firebase RTDB, which serves as the repository for information about companies, their operational cells, and the machines within those cells. At the top-level, the data table contains entries representing individual companies. Each company entry is identified by an index. Within each company entry, there is a cells collection, which organizes the various operational cells associated with that company.

The cells collection comprises a series of entries, each representing a single cell within a given company. Each cell entry encompasses a number of essential fields, including:

- cellData: This field contains specific data about the configuration and operations of the cell;
- name: This field contains the name of the cell;
- permission: The permission field contains the minimum permission required to access the cell information;
- machines: This sub-table references the machines operating within the cell, further subdividing the data structure.

Each machine table contains entries for specific machines, such as InterRoll and KukaKR3. The machine entries include detailed sub-tables and fields, which provide specific attributes and operational parameters of the machines in question. Additionally, the object details components.

The hierarchical structure allows for the efficient organisation and retrieval of data, thereby facilitating comprehensive management of company operations, cell configurations, and machine specifics.

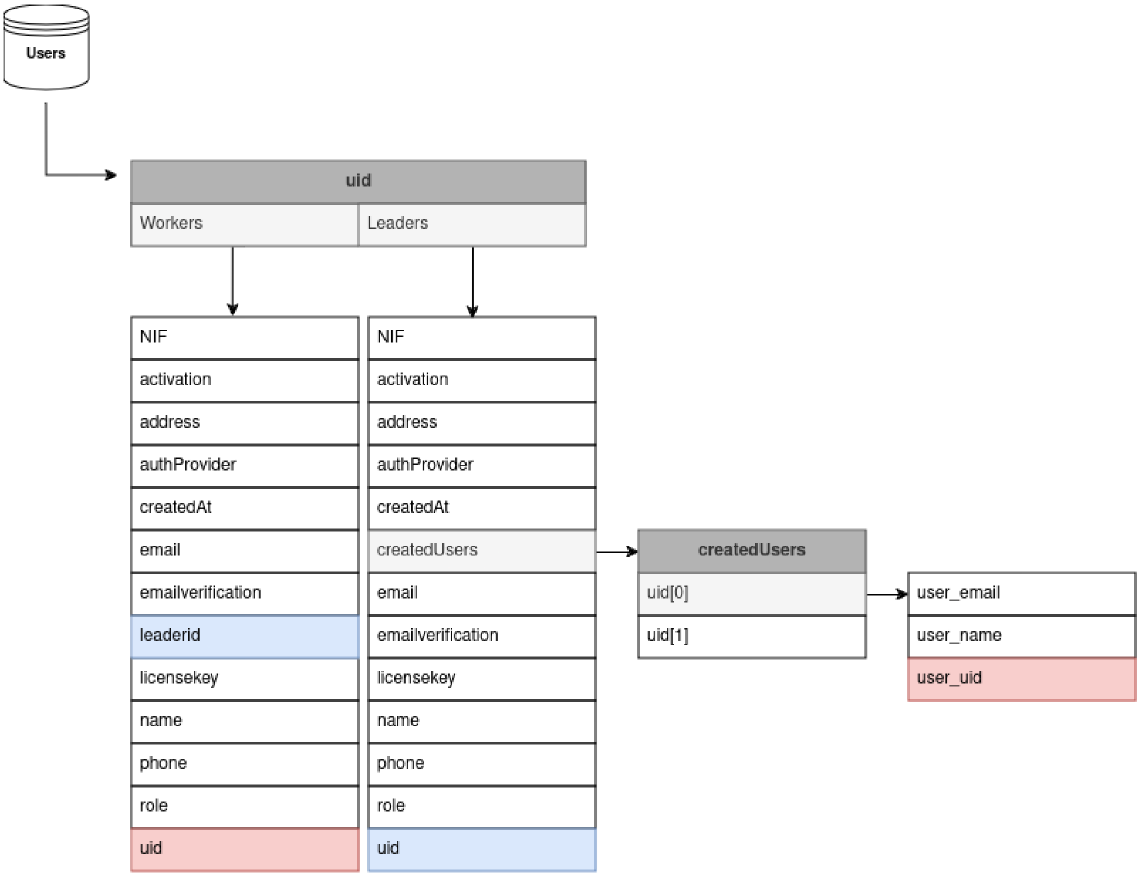

Figure 5.10: Users table diagram

The users table, shown in the Figure 5.10, is structured to manage and categorise users based on their roles within the organisation. At the top level, the uid table distinguishes between Workers and Leaders, storing unique identifiers (uid) for each user category. This distinction is beneficial in managing user permissions and roles within the system. Additionally, the createdUsers table logs the creation of new users, linking them to their respective creators, usually leaders. This table contains fields such as uid[0] and uid[1], which represent individual user IDs and track the user creation process. This structure ensures that user management is streamlined and that user roles and permissions are clearly defined.

The relation between the data and users tables is of critical importance to the overall database structure. Each company in the data table is linked to a leader through the leaderid field. This establishes a direct correlation between the companies and the leaders who manage them. Furthermore, the hierarchical structure within the data table, from companies to cells to machines, ensures that all operational data is organised and readily accessible.

The createdUsers table serves to further enhance this relationship by tracking which leaders create specific users, thereby ensuring accountability and clear user management. The comprehensive relational structure facilitates the management and oversight of data in an industrial environment, as well as the assignment of users to roles.

5.3.2 Firestore

Firestore is a scalable, real-time database from Firebase, which is well-suited to the development of responsive applications. It provides real-time updates, scalability, and offline support. In the application, Firestore is employed to store data that is subsequently utilised to generate dynamic graphics. This ensures that charts and visualisations are updated instantaneously in response to changes in the underlying data.

5.3.3 Authentication Service

Firebase Authentication is a comprehensive service that supports a range of authentication methods, including passwords, phone numbers, and federated identity providers such as Google, Facebook, and Twitter, among others. This flexibility ensures that users can easily and securely log in to our application using their preferred authentication method [39]. The flexibility of Firebase Authentication ensures that users can log in using the method that is most convenient for them. This enhances both the user experience and security.

The decision to utilise Firebase Authentication for the application was made on the grounds of its considerable advantages. The platform offers a comprehensive range of pre-implemented features that streamline the authentication process, including secure credential storage, multi-factor authentication, and real-time authentication state monitoring. These built-in capabilities reduce the time and effort required for development, allowing the focus to be directed towards other critical aspects of the application.

5.3.4 Hosting Service

Firebase Hosting is a web hosting solution from Google that provides rapid and secure hosting for web applications and static content. It incorporates a global content delivery network (CDN), automatic SSL, and integration with other Firebase services, such as Firestore and Firebase Authentication. This makes it an optimal choice for developers seeking to streamline their project's infrastructure by utilising a single platform for hosting, database management, user authentication, and other related functions. The utilisation of Firebase Hosting facilitates the integration of all components of a project, thereby enhancing both the efficiency of development and the user experience.

In the project, Firebase Hosting can be incorporated into a strategic decision to maintain the entirety of the project within the Firebase ecosystem. This encompasses the frontend hosting, database management, user authentication, and server-side logic. The integration of Firebase Hosting into a project facilitates the streamlining of development and deployment processes, while also enhancing the cohesion and integration of service components.

5.4 Summary

The Implementation section of the document provides a detailed account of the selection and integration of the various tools and technologies that are essential for the project. The discussion commences with an outline of the backend and frontend technologies employed, each selected for its capacity to support scalable, efficient, and interactive web applications. The backend is described in terms of its robust capabilities in server management, API interactions, and cloud integrations. Particular emphasis is placed on Firebase services such as Firestore and Firebase Authentication, which play crucial roles in data management and user authentication.

In terms of the frontend, the focus shifts to the use of contemporary JavaScript frameworks that guarantee a dynamic and responsive user experience.

The user interface section examines the various interfaces designed for unauthenticated users, authenticated clients, and administrators. It demonstrates how unauthenticated users can access general features of the application without compromising the functionality available to authenticated users. For authenticated clients, the interface expands to include features that allow for more extensive interaction with the application, such as the submission of data and the customisation of personal profiles. Administrators are furnished with a more comprehensive set of tools for user management, data analytics, and system settings, thereby ensuring that they are able to maintain the application's integrity and performance effectively.

The use of Firebase is a key area of discussion, with particular focus on the integration of its various services, including Firestore for database management, Firebase Authentication for secure user access, and Firebase Hosting for efficient content delivery. Of these, Real-Time Database is highlighted for its ability to support real-time data presentation, which is crucial for analytics and dashboard updates without delay. The final section considers the integration of these different technologies. In particular, attention is given to how the various components of the application—the backend and front end—work together with the robust tools available from Firebase to create a secure, scalable, user-friendly application. The integration of these technologies ensures that the application can handle the demands of a growing user base without sacrificing performance.

Chapter 6

Tests and Results

In this chapter, the process employed to test and validate the application will be described. This process was designed to confirm the functionality and reliability of the application. In order to ensure that the application performs in accordance with its specifications, tests were conducted within both simulated environments and the actual real-world environment.

The initial phase involved the conduct of tests in a simulated environment, which allowed for an exhaustive evaluation of the application's basic functions and its performance under various scenarios. These tests helped to identify and fix issues before the application could be tested in more complex real-world scenarios.

Subsequent testing in real environments was conducted with the objective of verifying the application's performance in real-time operations. This phase aimed to ensure that the application met the needs it was designed to address. Insights were gained into its stability and scalability when deployed in actual use scenarios.

6.1 Simulation Environment Testing

From the earliest stages of the project, it was evident that simulation tests would be an essential element. As each new feature of the application was developed and coded, it was subjected to immediate testing in a simulated environment. This approach facilitated the identification and rectification of errors in a prompt manner, while also ensuring that each component functioned correctly prior to its integration into the larger system. This iterative process of coding and testing provided a robust framework for developing the application with an optimal degree of reliability from its inception.

As the application development process advanced, the focus shifted to a more comprehensive examination of its critical components. Particular attention was paid to features that handle real-time data updates, which are of critical importance in terms of the application's performance but present considerable challenges regarding to data processing accuracy and speed. These components were subjected to meticulous testing in order to ensure that they functioned correctly under all operating conditions that could be reasonably anticipated. Two Python scripts were of particular significance during this phase, each designed to address distinct aspects of the application's capabilities.

The initial script (C.1) was designed to test the robot's movements within the application's integrated 3D model. It generated random values for the six axes of the robot, ranging from 0 to 360 degrees, and published these values to MQTT topics using a local broker with IP 127.0.0.1. This ensured that data flowed from the simulation to the application's visual representation. To establish communication, some configurations were required. Firstly, CorefluxCentral was run to create a local broker on the computer. MQTT Explorer was then used to view all the topics published in the broker. Coreflux provides numerous assets that facilitate the transfer of information in different formats and, in this case, the *mqtt_firebase* asset was employed to read data inserted in a specific topic and then write it to the Firebase Real-time database. The asset has been configured with the objective of enabling the transfer of data from the MQTT topic to the database. The asset is designed to read data published in a specific topic and, having configured access to the database, to write that value to a pre-configured field in the database. Thus, the asset was configured to read six different topics, each with the values of each axis, and to pass them on to the database. This demonstrated the functionality of both the 3D model and the asset in question.

The second script (C.2) was developed with the objective of simulating and testing the functionality of the alarm system, which is of vital importance for operational safety and responsiveness. The script generate and toggle alarm states at 15-second intervals, thereby testing the application's capacity to effectively handle and display alarm conditions. It was essential that the application could manage and relay alerts accurately and promptly under various simulated conditions. As in the preceding experiment, the asset was reconfigured. The results demonstrated that the alarm values were transmitted correctly, confirming the efficacy of the system.

The scripts collectively enabled a comprehensive examination of the application's real time data processing and visualisation capabilities. By simulating realistic operational scenarios, the tests permitted the identification and resolution of potential issues in the application's design and execution.

Consequently, the application not only met the rigorous standards established during the initial testing stages but also demonstrated robust functionality and reliability in handling complex, real-time data interactions, allowing us to advance to the tests conducted in a real-world context.

6.2 Real-World Environment Testing

Following the successful finishing of simulated tests, we proceeded to the stage of real-world testing.

The objective of this test is to obtain the robot's variable values, which will in turn be employed in the application. In order to achieve this, it is first necessary to establish a connection between the robot and the computer.

Initially, a fixed IP address was assigned to the robot (10.2.0.152) in the robot network configuration, and it was verified that both the robot and the computer were connected to the same network. A ping command was employed to test the connectivity between the two entities, and it was confirmed that they were indeed connected to the same network.

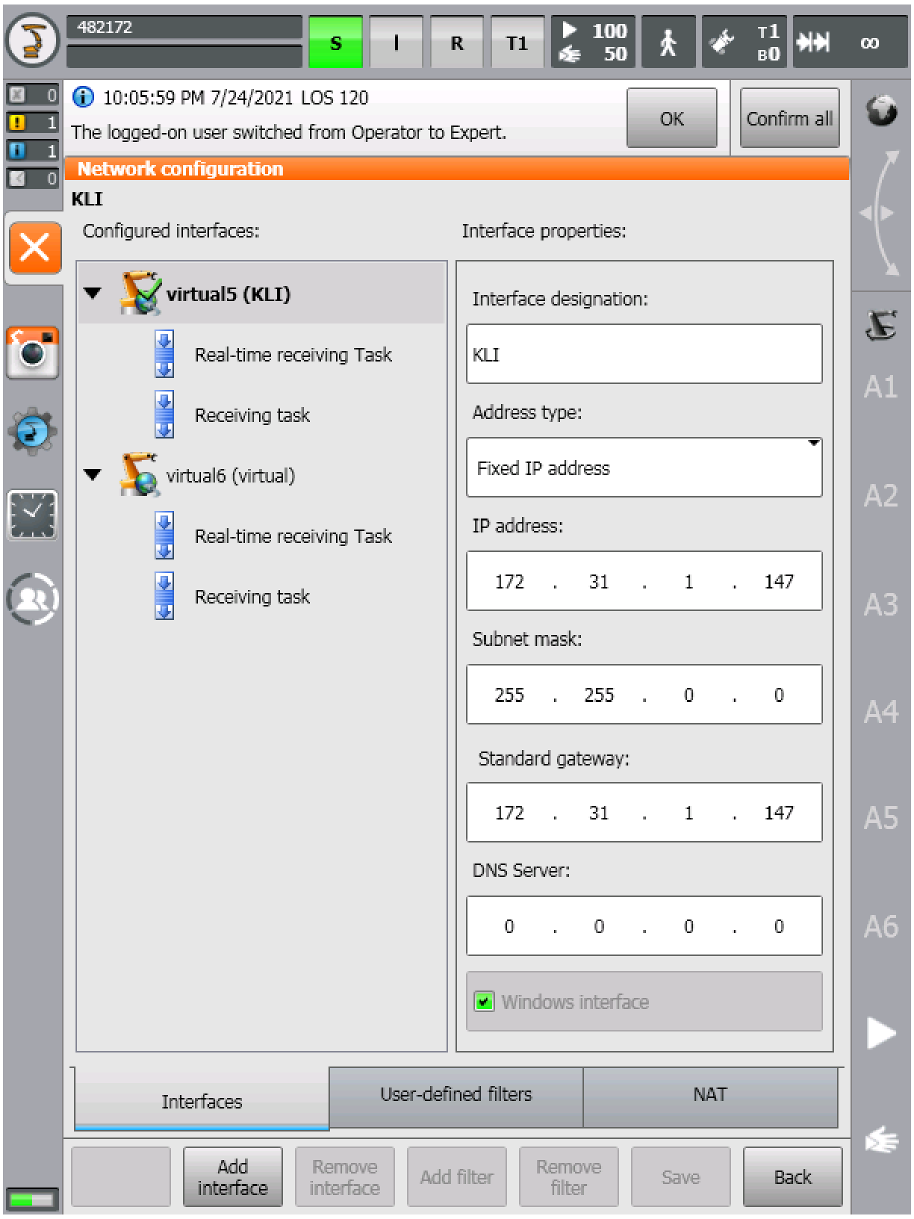

Figure 6.1: Example of network configuration in KUKA KR3

Thereafter, CorefluxCentral was executed on the computer in order to create the local broker, and MQTT Explorer was employed in order to inspect the existing topics in the broker. The robot was configured with the IP address of the broker, which was the IP address of the computer (10.3.0.11).

Given that the KUKA KR3 robot is equipped with the KUKA Device Connect software, it has the capability to directly write various variables with their respective values to MQTT topics.

Subsequently, a program was created on the robot that published a topic every second containing a JSON with various variables, including axis values, motor temperatures, and other data. MQTT Explorer enabled the observation of the topic published by the robot on the broker, thereby facilitating access to the various variables published by the robot. As the robot publishes a JSON with multiple variables instead of a single value, as in the simulated tests, a Python script was developed to deconstruct the JSON and publish individual topics with the values from the JSON. The objective of this test was to verify the movement of the 3D model in the application once again. Consequently, the axis position values were extracted from the JSON and republished to other topics. The *mqtt_firebase* asset then read the simplified values from these topics and wrote them to the database.

Once all the necessary components had been assembled, the robot was operated to verify that it was fully operational. However, upon examination, it was observed that the 3D model's image did not align with the actual reality. It was necessary to create an offset to the initial position of the robot in the model in order to ensure that the positions were identical to those of the real robot.

Following this adjustment, the functionality of the system was verified.

6.3 Summary

This chapter provides a detailed description of the transition from simulated to real-world testing, with a particular focus on the integration of the KUKA KR3 robot with the application. A robust connection was established between the robot and the computer, enabling the accurate transmission and reflection of the robot's variable data within the application. The key steps involved the establishment of network connectivity, the configuration of MQTT communication protocol, and the development of a Python script to handle complex data formats.

The testing revealed discrepancies between the 3D model and the real robot's positions. These discrepancies were resolved by adjusting the model's initial offset. This process guaranteed that the application's real-time data management was reliable. The successful completion of these tests serves to demonstrate the robustness of the application in practical, real-world scenarios, thereby marking a significant milestone in its development and validation process.

Chapter 7

Conclusions

This chapter presents a summary of the work completed and a discussion of potential future work that could be undertaken to enhance the prototype developed.

7.1 Final Conclusions

In the contemporary digital era, the 4th Industrial Revolution has brought about profound alterations, particularly in the domain of the Internet of Things (IoT). The concept of Industry 4.0 places a strong emphasis on connectivity and real-time data exchange, with the objective of enabling factories to enhance their efficiency and performance. In order to remain competitive, factories must continually evolve and adopt new technologies in order to keep up with the constant progress in this digital era.

This research was conducted with the objective of addressing the challenges associated with real-time database management in IoT environments. The system comprises several key components: the working machine, a 6-axis robot (KUKA KR3); the workstation, which is responsible for establishing communication protocols and managing the data pipeline from the robot to the database; the database, which is Firebase and which stores all relevant information; and the application, which has been developed with a backend in Node.js and a frontend in React.js. The application retrieves data from the database, performs additional analytics, and displays the information. The online application can be accessed by anyone via the following link, https://esifactorypulse.web.app.

The KUKA KR3 robot, which is equipped with a range of sophisticated capabilities, serves as the foundation of our system. The workstation serves to mediate communication between the robot and the database, thereby ensuring a correct flow of data. The utilisation of CorefluxCentral to establish a local broker and MQTT Explorer to monitor the topics ensured the effective transmission of data. The Python script, which was developed to handle the JSON data published by the robot, decomposed the data into individual topics, which were then stored in Firebase. This configuration permitted the verification of the movement of the 3D model in the application with precision.

It was essential to perform extensive debugging and testing during the development process. A multitude of tests were conducted in both simulated and real-world environments with the objective of validating the system's performance under various conditions. The simulated tests provided a controlled environment in which potential issues could be identified, while real-world tests ensured the system's reliability and accuracy in practical applications. These tests confirmed that the system could handle real-time data updates efficiently and accurately.

A significant challenge encountered was ensuring the alignment of the 3D model in the application with the positions of the real robot. By creating an offset to the initial position of the robot in the model, the 3D representation could be aligned with the real-world robot. This adjustment was crucial for the system's accuracy and reliability.

The successful completion of the tests validates the proposed system. The real-time database management solution for IoT environments, which integrates the KUKA KR3 robot with an online application, has been demonstrated to be both reliable and effective. This system represents a significant advancement in leveraging Industry 4.0 technologies to enhance manufacturing processes and data management. By enabling real-time monitoring and interaction, a solid foundation has been laid for future improvements and deployments in practical scenarios.

7.2 Future Work

Although the current system provides a robust foundation for real-time database management in IoT environments, there are numerous opportunities for future enhancements. One potential avenue for further development is the incorporation of additional features within the application. For example, the development of new pages tailored to different types of machines would greatly expand the system's versatility and applicability. This would permit the integration of a multitude of industrial equipment, each with its own distinctive data requirements and operational characteristics.

A further significant area of future work concerns the challenge of implementing this system in an actual factory setting. The deployment of the system in a real-world industrial environment would provide invaluable insights and allow for practical adjustments to optimise performance and reliability. The implementation of this system would necessitate the resolution of several challenges, including the assurance of integration with existing factory systems, the management of voluminous data in real-time, and the maintenance of robust security measures to safeguard sensitive information.

Furthermore, further development could be directed towards enhancing the analytical capabilities of the application. The incorporation of advanced analytics and machine learning algorithms could facilitate a more profound comprehension of the data, thereby enabling more informed decision-making and predictive maintenance. Additionally, enhancing the user interface to facilitate intuitive and user-friendly navigation for operators with varying levels of technical expertise would be advantageous.

In conclusion, the future work on this project has the potential to significantly enhance its functionality and impact, thereby making it an even more powerful tool for leveraging Industry 4.0 technologies in manufacturing environments.

References

[1] United Nations. The 2030 agenda for sustainable development. https://sdgs.un.org/goals, 2015. Retrieved from the United Nations website. Accessed on 17th June 2024.

[2] SAICA NPO. The stages of industrial revolution and its impact on jobs. https://www.accountancysa.org.za/the-stages-of-industrial-revolution-and-its-impact-on-jobs, 2008. Accessed on 23th May 2024.

[3] John Gantz and David Reinsel. Extracting value from chaos. In *IDC iview*, page 1142(2011):1–12, 2011.

[4] Sisay Ebabye Tsigie and Alazie Dagnaw Gizealew. The role of robotics technology and internet of things for industry 4.0 realization. *International Journal of Advanced Trends in Computer Science and Engineering*, 10(2), 2021.

[5] JavaTpoint. MQTT Protocol - Image. https://www.javatpoint.com/mqtt-protocol. Accessed on 29th December 2023.

[6] AG Deloitte. Industry 4.0 challenges and solutions for the digital transformation and use of exponential technologies. *McKinsey Global Institute*, 13:1–16, 2015.

[7] Alasdair Gilchrist and Alasdair Gilchrist. Introducing industry 4.0. *Industry 4.0: The industrial internet of things*, pages 195–215, 2016.

[8] Henning Kagermann, Wolfgang Wahlster, Johannes Helbig, et al. Recommendations for implementing the strategic initiative industrie 4.0. *Final report of the Industrie*, 4(0):82, 2013.

[9] Shi-Wan Lin, Bradford Miller, Jacques Durand, Rajive Joshi, Paul Didier, Amine Chigani, Reinier Torenbeek, David Duggal, Robert Martin, Graham Bleakley, et al. Industrial internet reference architecture. *Industrial Internet Consortium (IIC), Tech. Rep*, 2015.

[10] Ricardo Silva Peres, Andre Dionisio Rocha, Paulo Leitao, and Jose Barata. Idarts – towards intelligent data analysis and real-time supervision for industry 4.0. *Computers in Industry*, 101:138–146, 2018.

[11] Robotic Gizmos. Kuka connect: Access your robots from anywhere. https://www.roboticgizmos.com/kuka-connect/, 2016. Accessed on 29th December 2023.

[12] OPC Router. What is opc ua? https://www.opc-router.com/what-is-opc-ua/. Accessed on 7th January 2024.

[13] https://www.rfwireless-world.com/Tutorials/MQTT-tutorial.html. Accessed on 4th January 2024.

[14] Emerson. Why opc ua matters. https://www.ni.com/en/solutions/industrial-machinery/smart-machine-control/why-opc-ua-matters.html. Accessed on 5th January 2024.

[15] Peter Peniak, Emília Bubeníková, and Alžbeta Kanáliková. Extended gateway model for opc ua/iot device integration. In *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2021.

[16] Ogan M. Yigitbasioglu and Oana Velcu. A review of dashboards in performance management: Implications for design and research. *International Journal of Accounting Information Systems*, 13(1):41–59, 2012.

[17] Yance Gusnadi and Aditiya Hermawan. Designing employee performance monitoring dashboard using key performance indicator (kpi). *bit-Tech*, 2:19–26, 01 2020.

[18] J. Furmankiewicz, M. Furmankiewicz, and P. Ziuziański. Implementation of business intelligence performance dashboard for the knowledge management in organization. *Zeszyty Naukowe. Organizacja i Zarządzanie / Politechnika Śląska*, z. 82:43–60, 2015.

[19] Marjan Ghazisaeedi, Reza Safdari, Mashallah Torabi, Mahboobeh Mirzaee, Jebraeil Farzi, and Azadeh Goodini. Development of performance dashboards in healthcare sector: Key practical issues. *Acta Informatica Medica*, 23:317, 10 2015.

[20] Er Mahendrawathi, Danu Pranantha, and Johansyah Dwi Utomo. Development of dashboard for hospital logistics management. In *ICOS 2010 - 2010 IEEE Conference on Open Systems*, ICOS 2010 - 2010 IEEE Conference on Open Systems, pages 86–90. IEEE Computer Society, 2010.

[21] J.A. Mattera S.A. Roumanis M.J. Radford H.M. Krumholz E.H. Bradley, E.S. Holmboe. Data feedback efforts in quality improvement: lessons learned from us hospitals. *Qual Saf Health Care*, 13, Feb. 2004.

[22] Machinemetrics. Explore the digital factory with machinemetrics. https://www.machinemetrics.com/industrial-iot-platform. Accessed on 15th March 2024.

[23] Machinemetrics. Parts goal view of the current shift dashboard. https://support.machinemetrics.com/hc/en-us/articles/360026591133-Current-Shift-Dashboard-Dashboards. Accessed on 15th March 2024.

[24] Google. https://marketingplatform.google.com/about/analytics/. Accessed on 18th March 2024.

[25] Google. https://analytics.googleblog.com/2011/03/new-google-analytics-dashboards.html/. Accessed on 18th March 2024.

[26] Grafana. Grafana open source. https://grafana.com/docs/grafana/latest/fundamentals/. Accessed on 15th March 2024.

[27] Grafana. Dashboard anything. observe everything. https://grafana.com/grafana/. Accessed on 15th March 2024.

[28] Indra Kharisma Raharjana, Daniel Siahaan, and Chastine Fatichah. User stories and natural language processing: A systematic literature review. *IEEE Access*, 9:53811–53826, 2021.

[29] Node.js. About node.js. https://nodejs.org/en/about/, n.d. Accessed on 28th May 2024.

[30] Express.js. Express - node.js web application framework. https://expressjs.com/, n.d. Accessed on 28th May 2024.

[31] Firebase Admin SDK. Introduction to the admin sdk. https://firebase.google.com/docs/admin/setup, n.d. Accessed on 28th May 2024.

[32] React. A javascript library for building user interfaces. https://reactjs.org/, n.d. Accessed on 28th May 2024.

[33] React. Virtual dom and internals. https://reactjs.org/docs/faq-internals.html, n.d. Accessed on 28th May 2024.

[34] Axios. Promise based http client for the browser and node.js. https://axios-http.com/, n.d. Accessed on 28th May 2024.

[35] React Router. Declarative routing for react. https://reactrouter.com/, n.d. Accessed on 28th May 2024.

[36] Redux. A predictable state container for js apps. https://redux.js.org/, n.d. Accessed on 28th May 2024.

[37] Firebase Realtime Database. Realtime database. https://firebase.google.com/products/realtime-database, n.d. Accessed on 28th May 2024.

[38] Firestore. Cloud firestore. https://firebase.google.com/products/firestore, n.d. Accessed on 28th May 2024.

[39] Firebase authentication documentation. https://firebase.google.com/docs/auth. Accessed: 2024-06-06.

Appendix A

User Interface - User View

This appendix presents screenshots of the user interface in the client perspective of the application developed.

Figure A.1: Reset password page

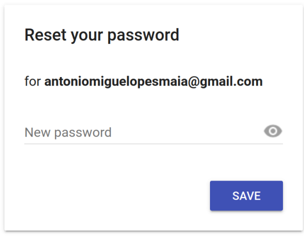

Figure A.2: Reset password page 2

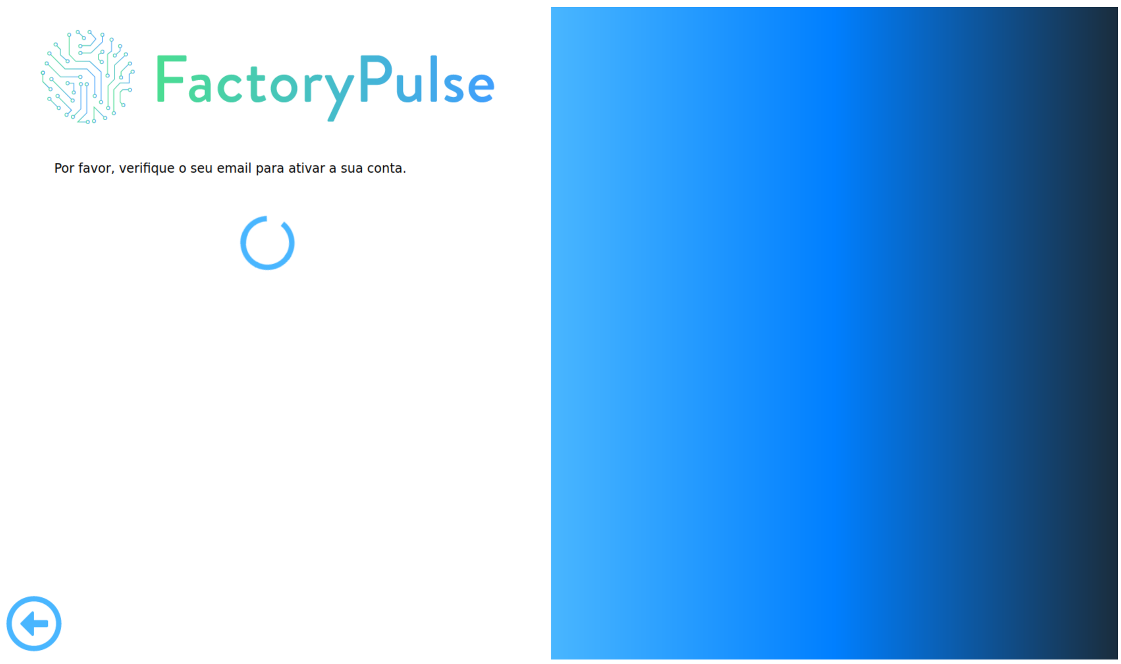

Figure A.3: Confirm email page

Figure A.4: Additional information page

Figure A.5: Activation account page

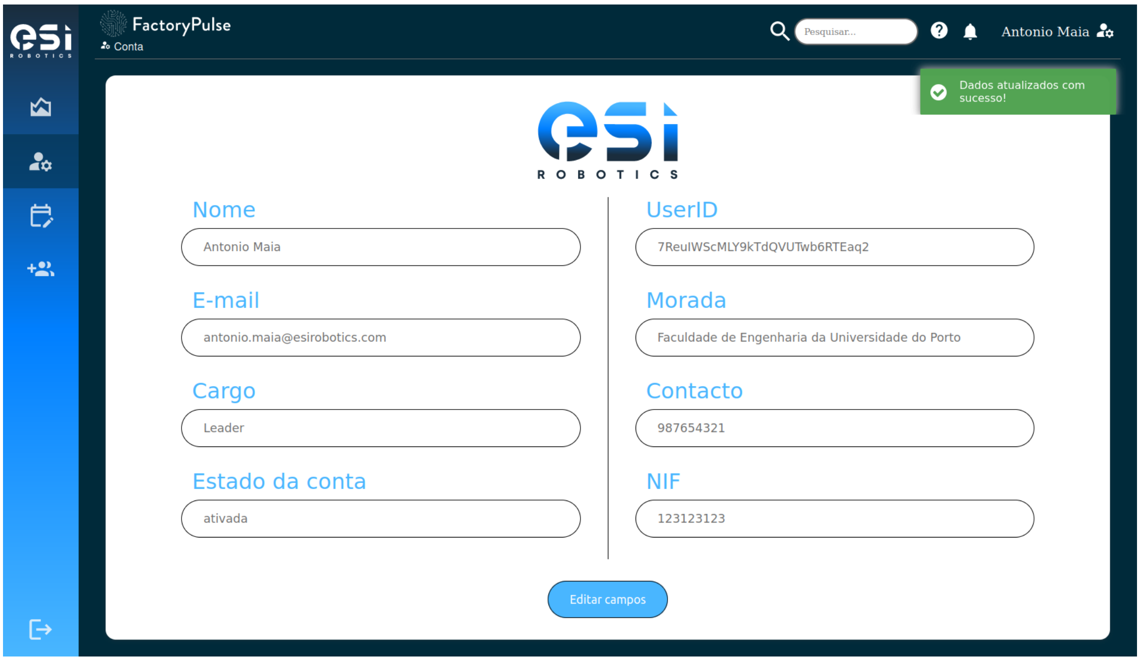

Figure A.6: Account page

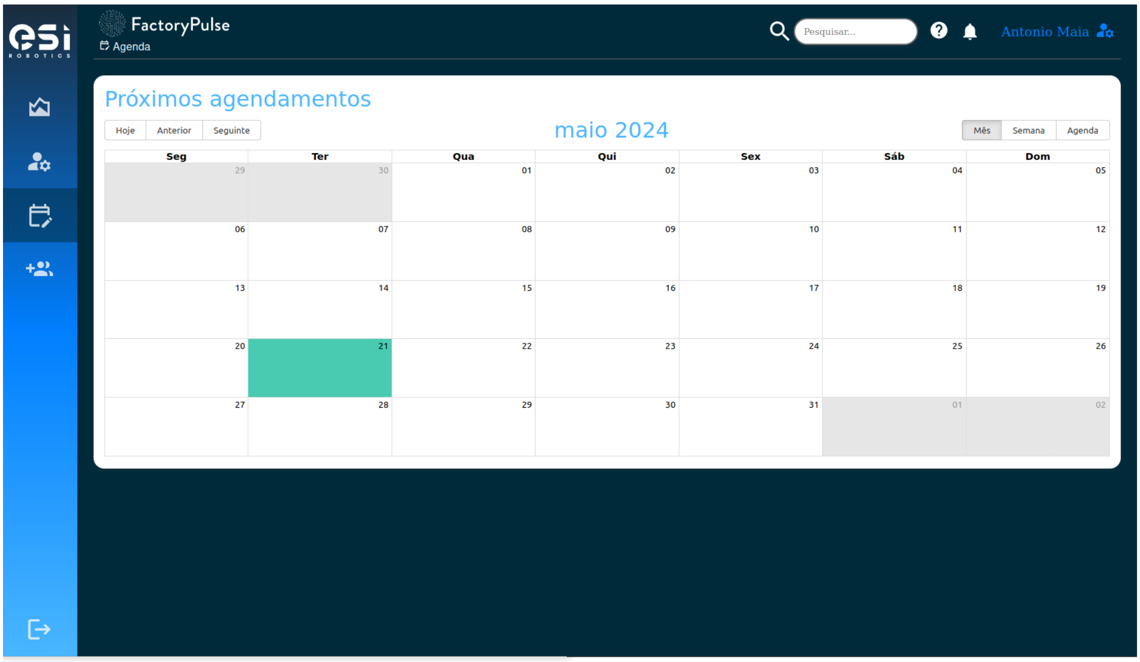

Figure A.7: Calendar page

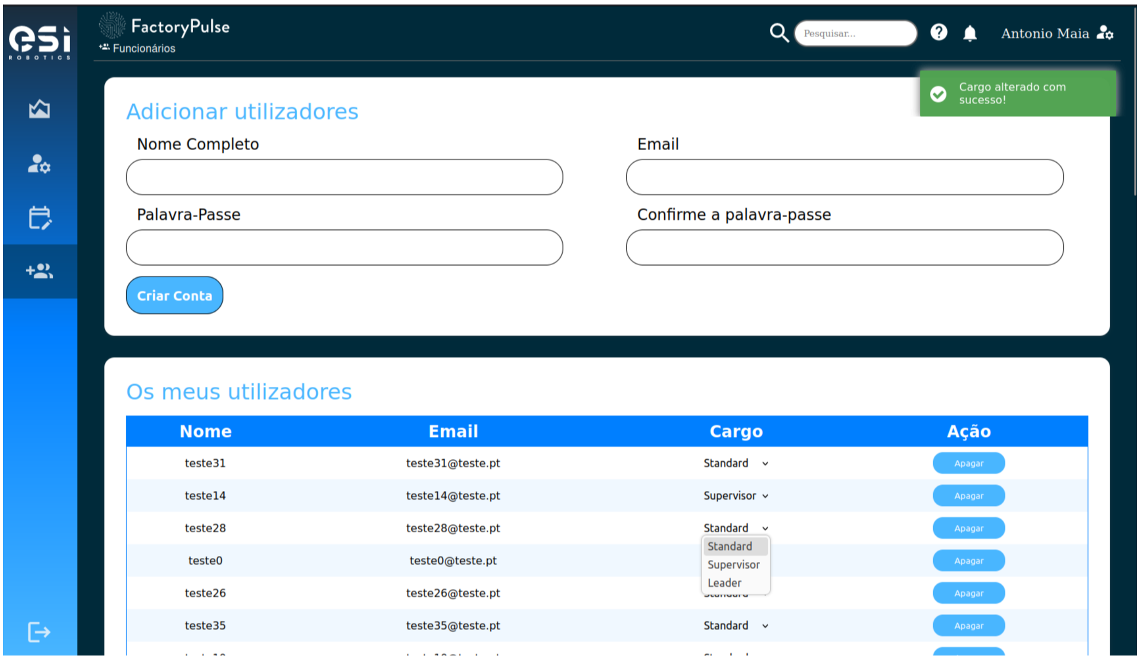

Figure A.8: Users page

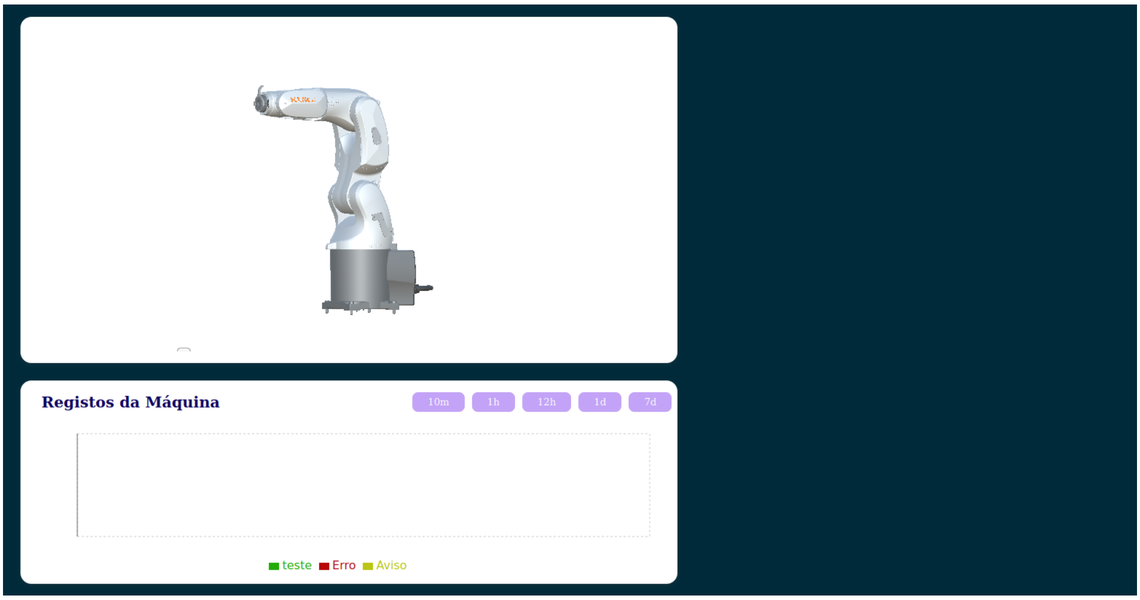

Figure A.9: Robot page 2

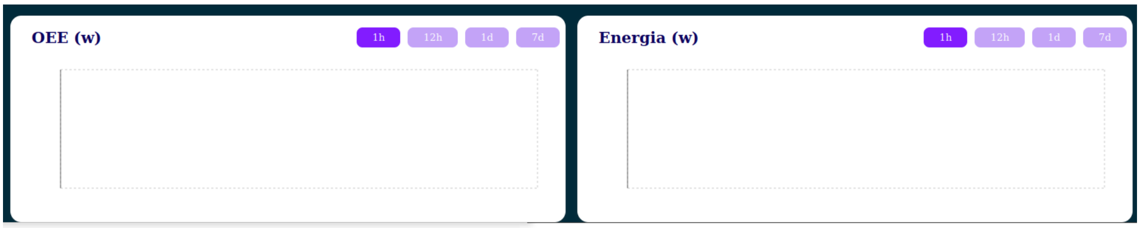

Figure A.10: Robot page 3

Appendix B

User Interface - Admin View

This appendix presents screenshots of the user interface in the admin perspective of the application developed.

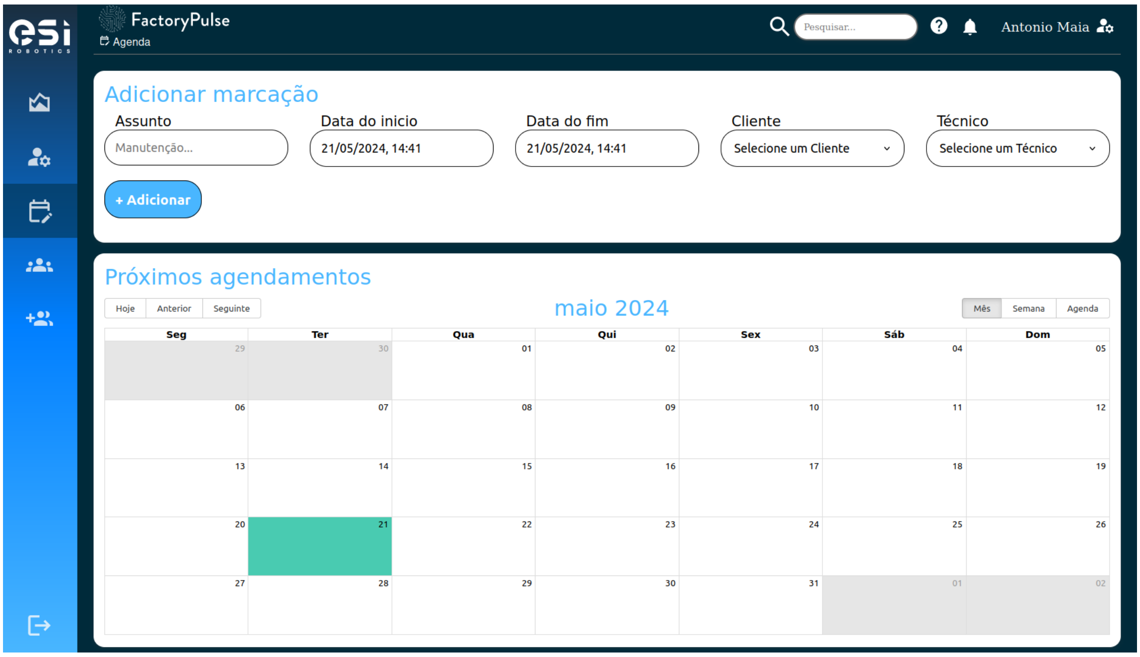

Figure B.1: Calendar page

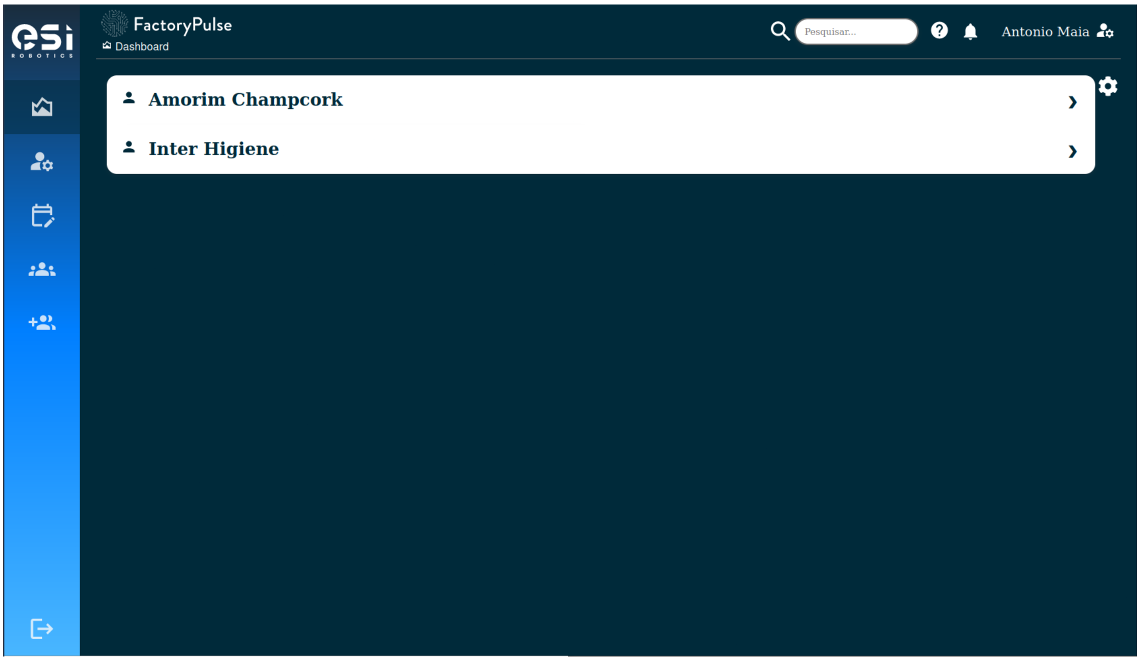

Figure B.2: Dashboard page

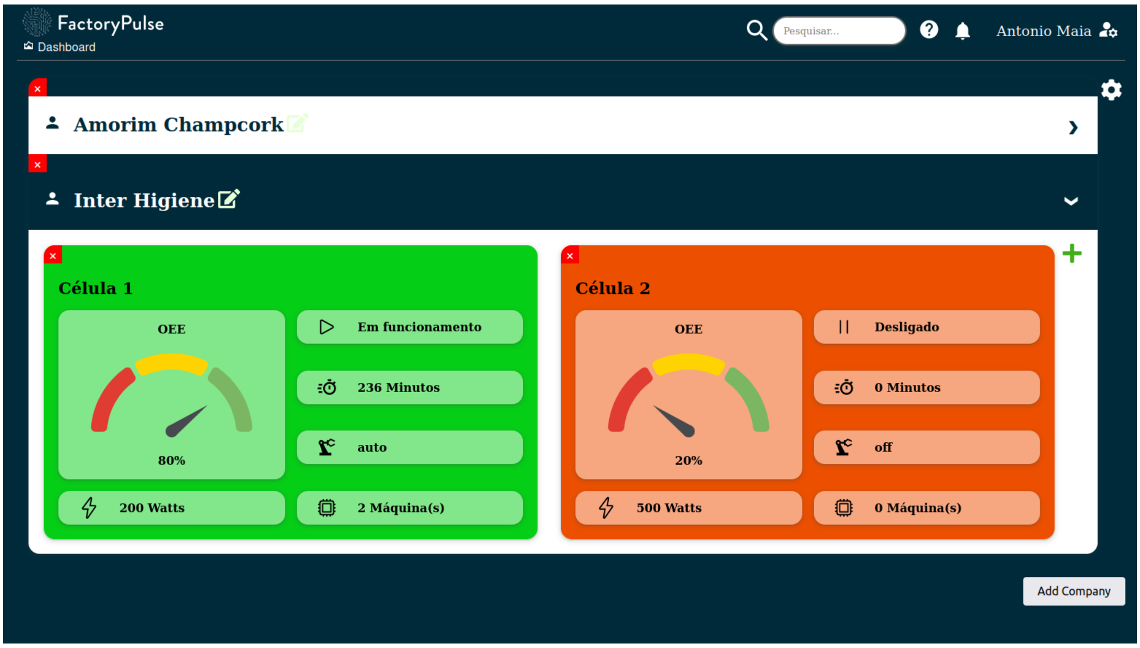

Figure B.3: Dashboard page (edit mode)

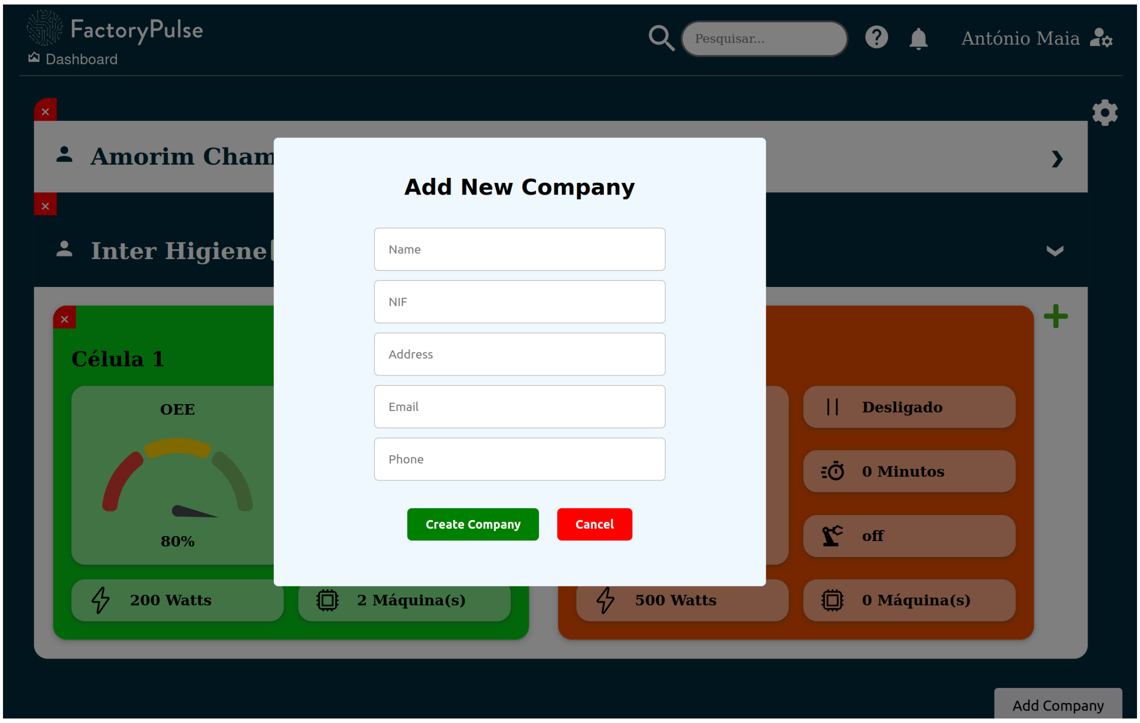

Figure B.4: Create company forms

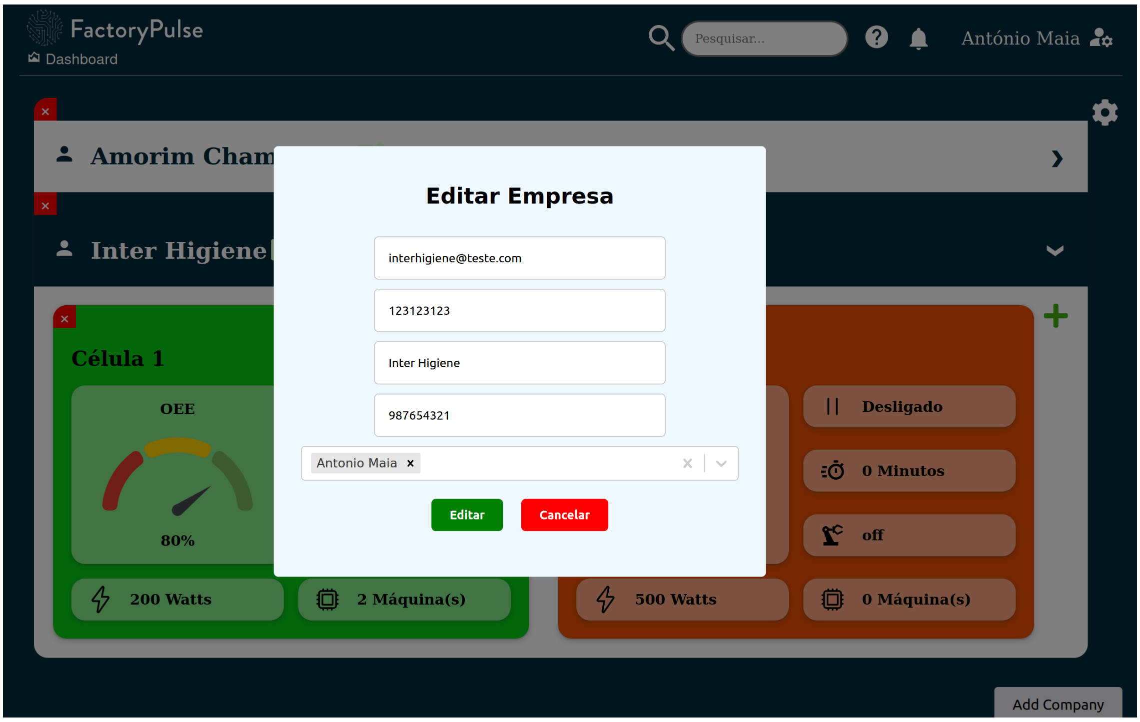

Figure B.5: Edit company forms

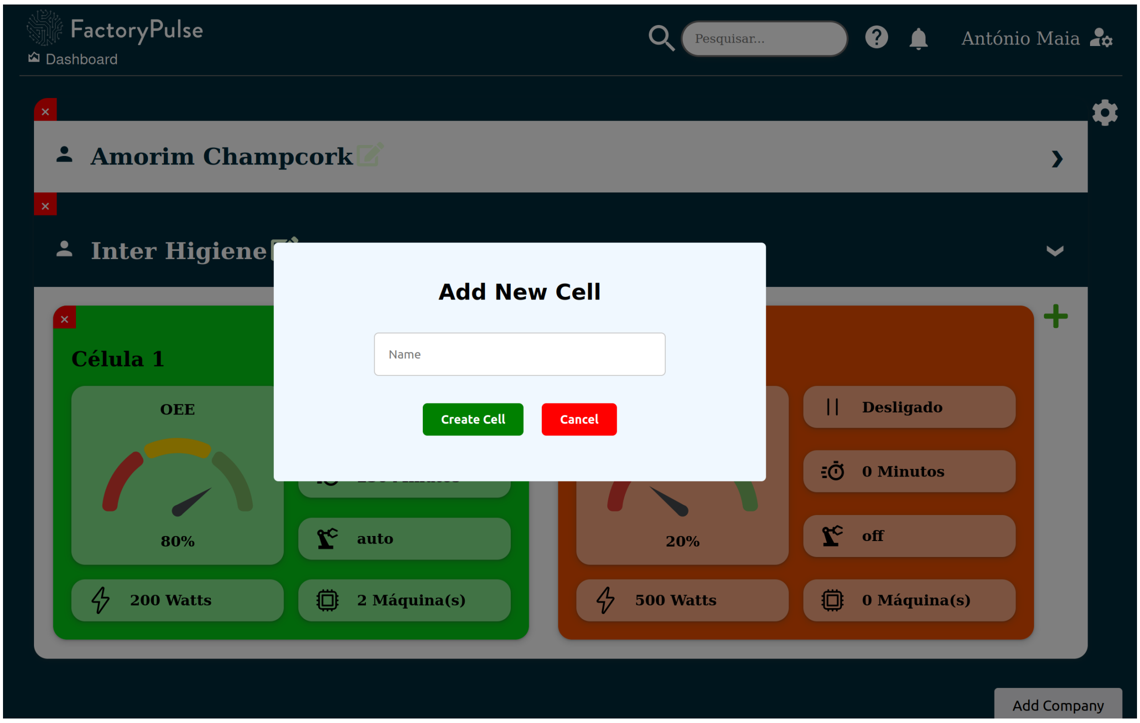

Figure B.6: New cell forms

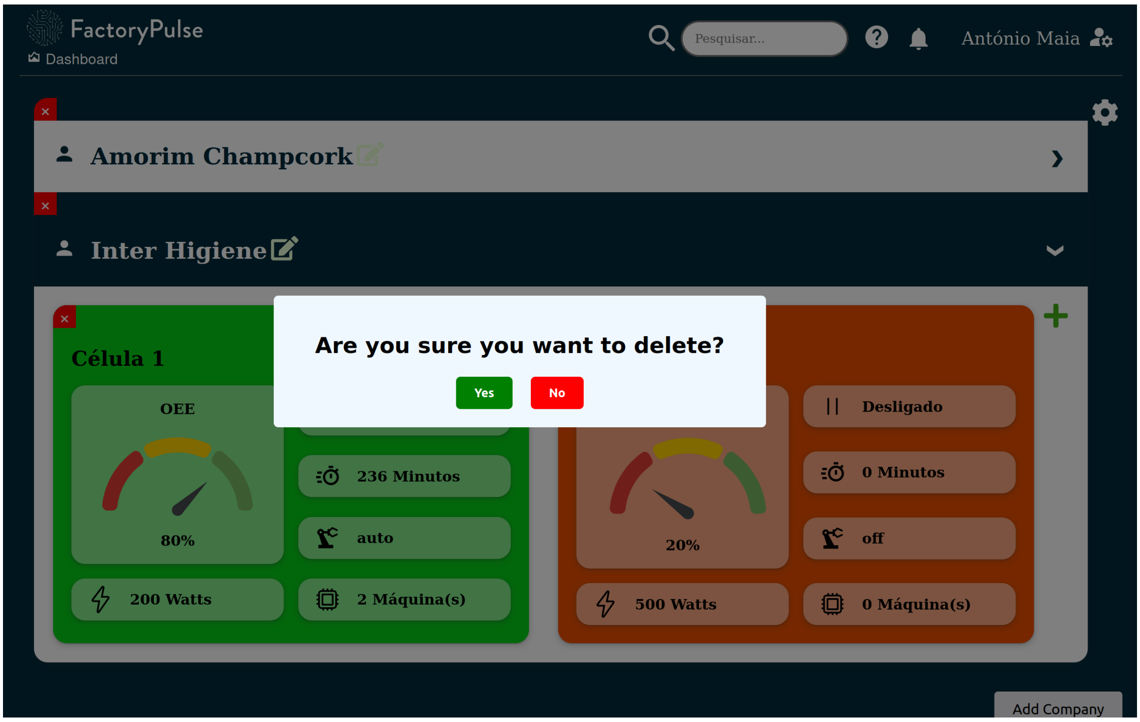

Figure B.7: Delete confirmation modal

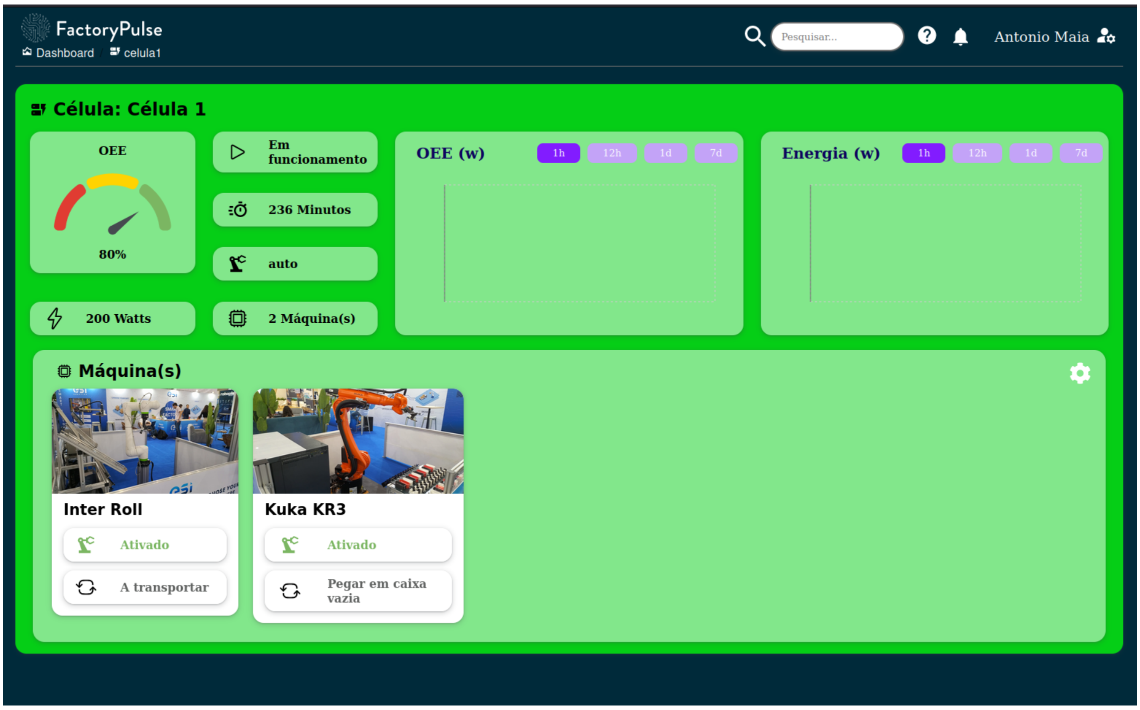

Figure B.8: Cells details page

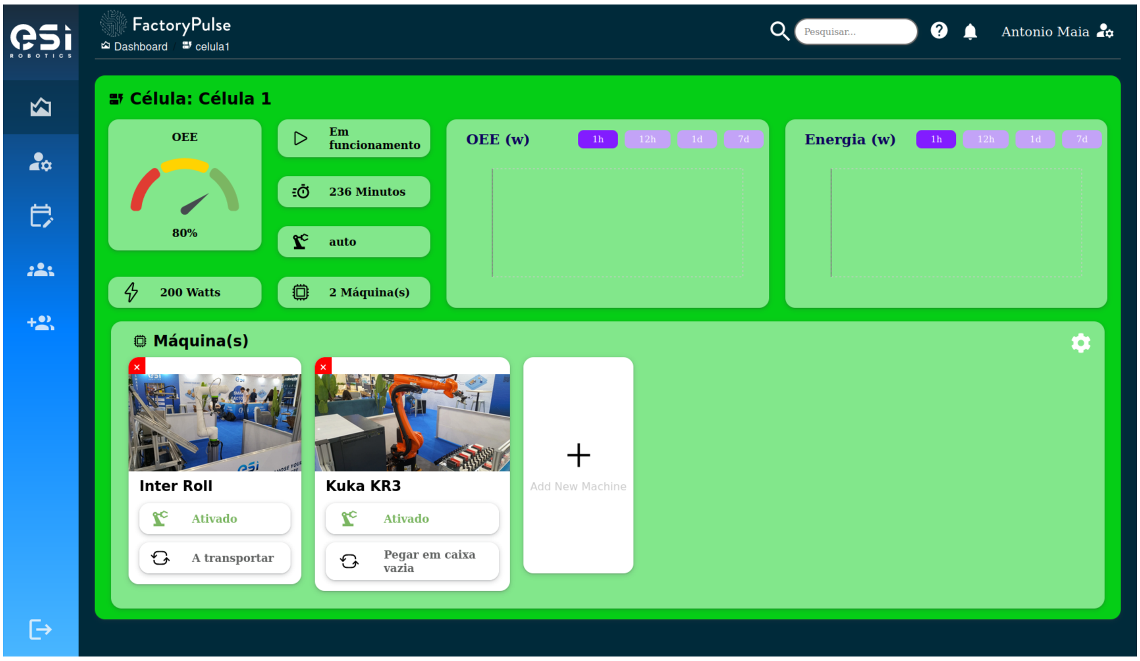

Figure B.9: Cells details page (edit mode)

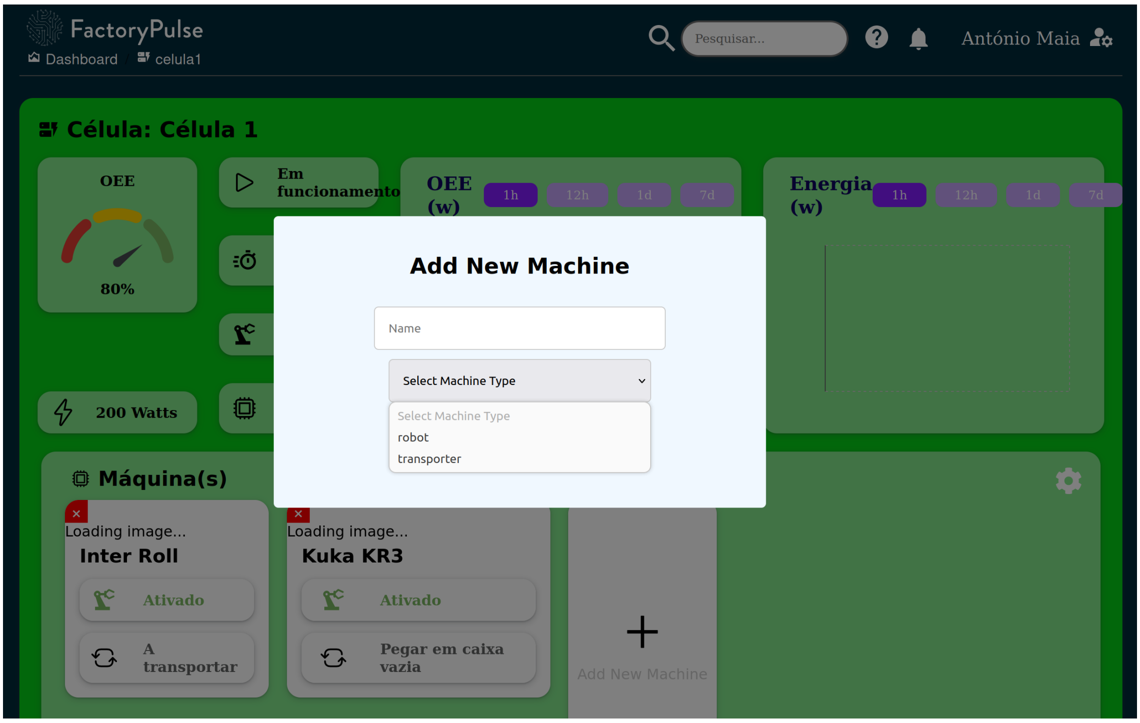

Figure B.10: New machine forms

Appendix C

Code Appendix

This appendix presents the source code for the Python scripts that were used.

```
import time
import paho.mqtt.client as mqtt
import json

# MQTT Broker settings
MQTT_BROKER = '127.0.0.1'  # Adjust as necessary to match your broker address
MQTT_PORT = 1883  # Adjust as necessary to match your broker port

# Topic to subscribe for incoming JSON data
SUBSCRIBE_TOPIC = "Coreflux/Data/Assets"  # This should match the topic from your system
# Topics to publish the axis values
PUBLISH_TOPICS = [f"FirebaseAsset/KukaAxis{i}" for i in range(1, 7)]

def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe(SUBSCRIBE_TOPIC)  # Subscribe to the topic

def on_message(client, userdata, msg):
    data = json.loads(msg.payload.decode())
    # Extract axis values from the data, assuming they are stored under data['axis_values']
    axis_values = data['axis_values']  # You will need to adjust this to match your actual data structure

    for topic, value in zip(PUBLISH_TOPICS, axis_values):
        client.publish(topic, str(value))
        print(f"Published {value} to {topic}")

def main():
    client = mqtt.Client()
```

```
    client.on_connect = on_connect
    client.on_message = on_message  # Set the on_message callback

    try:
        client.connect(MQTT_BROKER, MQTT_PORT, 60)  # Connect to the MQTT broker
        client.loop_forever()  # Start the loop to process callbacks indefinitely
    except Exception as e:
        print(f"An error occurred: {e}")

if __name__ == "__main__":
    main()
```

Listing C.1: axes.py

```
import time
import paho.mqtt.client as mqtt

# MQTT Broker settings
MQTT_BROKER = '127.0.0.1'  # Change this to your MQTT broker address
MQTT_PORT = 1883  # Change this to your MQTT broker port

# List of topics
topics = [f"FirebaseAsset/KukaAlarm{i}" for i in range(1, 11)]
topic_states = {topic: False for topic in topics}  # Initial state set to False

def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    # Subscribing to all topics to receive updates as well
    for topic in topics:
        client.subscribe(topic)

def on_message(client, userdata, msg):
    print(f"Received message '{msg.payload.decode()}' on topic '{msg.topic}'")

def toggle_publish(client):
    while True:
        for topic in topics:
            current_state = topic_states[topic]
            new_state = not current_state  # Toggle the state
            client.publish(topic, str(new_state).lower())
            topic_states[topic] = new_state  # Update the current state
            print(f"Published {new_state} to {topic}")
        time.sleep(15)  # Wait for 15 seconds before next toggle

def main():
    client = mqtt.Client()
    client.on_connect = on_connect
    client.on_message = on_message
```

```

    try:
        client.connect(MQTT_BROKER, MQTT_PORT, 60)  # Connect to the MQTT broker
        client.loop_start()  # Start the loop to process callbacks
        toggle_publish(client)  # Start publishing loop
    except Exception as e:
        print(f"An error occurred: {e}")
    finally:
        client.loop_stop()  # Stop the loop
        client.disconnect()  # Disconnect from the broker

if __name__ == "__main__":
    main()
```

Listing C.2: alarms.py

```
import time
import paho.mqtt.client as mqtt
import json

# MQTT Broker settings
MQTT_BROKER = '127.0.0.1'  # Change to your MQTT broker address
MQTT_PORT = 1883  # Change to your MQTT broker port

# Topic to subscribe for incoming JSON data
SUBSCRIBE_TOPIC = "KUKA/KRC/devices/8100012/vdmainfo"  # Adjusted to the correct topic

# Topics to publish the axis values
PUBLISH_TOPICS = [f"FirebaseAsset/KukaAxis{i}" for i in range(1, 7)]

def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe(SUBSCRIBE_TOPIC)  # Subscribe to the topic

AXIS_TRANSFORMATIONS = [
    lambda x: -x,
    lambda x: x+60,
    lambda x: x,
    lambda x: x,
    lambda x: x,
    lambda x: x
]

def on_message(client, userdata, msg):
    print("Message received on topic:", msg.topic)
    try:
        # Load JSON data from MQTT message
```

```
        data = json.loads(msg.payload.decode())
        # Navigate through JSON to find the 'Messages' list, then 'Payload' dict
        payload = data['Messages'][0]['Payload']  # Assuming the first message in '
            Messages'
        # Extract values from the Payload for axes 1-6
        axis_values = [payload.get(f"ActualPositionA{i}", {}).get('Value', 0) for i
             in range(1, 7)]

        # Apply transformations to axis values
        corrected_values = [transform(value) for value, transform in zip(
            axis_values, AXIS_TRANSFORMATIONS)]

        # Publish corrected values to respective topics
        for topic, value in zip(PUBLISH_TOPICS, corrected_values):
            client.publish(topic, str(value))
            print(f"Published {value} to {topic}")
    except json.JSONDecodeError as e:
        print("Error decoding JSON:", e)
    except Exception as e:
        print("An error occurred:", e)


def main():
    client = mqtt.Client()
    client.on_connect = on_connect
    client.on_message = on_message

    try:
        client.connect(MQTT_BROKER, MQTT_PORT, 60)
        client.loop_forever()
    except Exception as e:
        print("Connection failed:", e)

if __name__ == "__main__":
    main()
```

Listing C.3: robot.py