Universidade de Aveiro
2023

Daniel Lopes Rodrigues

Real-time monitoring of out-of-flow parts

Monitorização em tempo real das peças fora de fluxo

Universidade de Aveiro
2023

Daniel Lopes Rodrigues

Real-time monitoring of out-of-flow parts

Monitorização em tempo real das peças fora de fluxo

Relatório de Estágio apresentado à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob orientação científica de José Paulo Oliveira Santos, Professor Auxiliar, do Departamento de Engenharia Mecânica da Universidade de Aveiro.

Este relatório de estágio teve o apoio dos projetos UIDB/00481/2020 e UIDP/00481/2020 - Fundação para a Ciência e a Tecnologia; e CENTRO-01-0145 FEDER-022083 - Programa Operacional Regional do Centro (Centro2020), através do Portugal 2020 e do Fundo Europeu de Desenvolvimento Regional.

O júri / The jury

Presidente / President	**Prof. Doutor Miguel Armando Riem de Oliveira** Professor Auxiliar da Universidade de Aveiro
Vogais / Committee	**Prof. Doutora Ana Maria Pinto Moura** Professora Auxiliar da *Universidade de Aveiro*
	Prof. Doutor José Paulo Oliveira Santos Professor Auxiliar da Universidade de Aveiro

Agradecimentos / Acknowledgements

Quero agradecer aos meus pais por sempre terem acreditado e contribuído para eu ser mais e melhor e chegar mais longe na minha vida académica e profissional.
Tenho de agradecer muito a minha namorada porque sem o incentivo dela nunca teria ingressado nesta etapa.
Agradeço ao meu orientador Prof. Doutor José Paulo Santos, não só pela orientação e aconselhamento durante o estágio, mas também pela oferta da oportunidade de ingressar num estagio de verão na Renault que se provou ser uma excelente decisão.
Agradecer ainda ao Eng. Diogo Ramos por toda a orientação e conhecimento que me transmitiu durante o estágio na Renault Cacia.
Por fim quero agradecer aos amigos que ganhei durante esta etapa da minha vida. A todas as vivencias e experiências que passamos juntos ficam as memorias e as amizades que levo para a vida.
Obrigado a todos, pois todos vocês foram importantes nesta etapa que agora concluo.

Keywords Traceability, Industry 4.0, Real-time location, Internet of Things, Automotive Industry

Abstract

The competition between the industries of the automotive components sector for the conquest of the national and international markets is the trigger that sets off the technological innovation in the sector. One of the factors that provide a competitive quality differential is traceability.
The system of traceability of parts leaving the normal flow of the production line, in order to carry out quality control tests, currently implemented at Renault CACIA, is manual and time-consuming. Failures of this method currently represent 50% of customer complaints, since this system doesn't have direct communication between the quality control center and the person in charge of the atelier, there is also difficulty in blocking loads due to the lack of information on the location of all the pieces and also the non-uniformity of the system.
This work presents the creation of a traceability and monitoring system, to be implemented in the automotive industry, at Renault CACIA.
The system implemented uses radio frequency identification hardware technologies and unitary marking, in terms of data processing, and software Flask and Python are used to perform all the backend logic and angular to generate a visualization and interaction interface with the operator.
With this system, workers have access to information about which parts are outside the normal production flow, for what reason, and their location in real-time. The system is integrated with an application that allows quality control requests to be issued.
With the implementation of this project was possible to generate a reduction of nine minutes per every time the process is executed representing 16% in the cycle time, allowing this the reduction of the amount of scrap and a reduction of the time that the machine is stopped. Is expected in the long term to reduce the client complaints to 25%.
The aim of this solution was to provide the workers with a simpler, more up-to-date, and automatic work tool, which reduces effort and increases productivity.

Palavras-chave Rastreabilidade, Indústria 4.0, Localização em tempo real, Internet das coisas, Indústria Automóvel

Resumo

A concorrência entre as indústrias do setor de componentes para automóveis pela conquista dos mercados nacional e internacional, é o gatilho que faz disparar a inovação tecnológica no setor. Um dos fatores que propicia um diferencial competitivo de qualidade é a rastreabilidade.
O sistema de rastreabilidade das peças que saem do fluxo normal da linha de produção, no sentido de efetuar testes de controlo de qualidade, atualmente implementado na Renualt CACIA, é manual e demorado. As falhas deste método representam atualmente 50% das reclamações clientes, uma vez que este sistema não tem uma comunicação direta entre o centro de controlo de qualidade e o responsável do atelier, existe ainda dificuldade no bloqueio de cargas por falta de informação sobre a localização de todas as peças e ainda a não uniformidade do sistema.
Neste projeto é apresentada a criação de um sistema de rastreabilidade e monitorização implementado na indústria automóvel, na Renault CACIA.
O sistema foi implementado recorrendo a tecnologias hardware de identificação por rádio frequência e marcação unitária, em termos de tratamento de dados e software são utilizados o Flask e linguagem Python para efetuar toda a lógica de backend e o angular para gerar uma interface de visualização e interação com o operador.
Com este sistema os trabalhadores têm acesso à informação de quais as peças estão fora do fluxo normal de produção, por que motivo, e qual a sua localização em tempo real. O sistema permite ainda que sejam emitidos pedidos de controlo de qualidade.
Com a implementação deste projecto foi possível gerar uma redução de nove minutos por cada vez que o processo é executado representando 16% no tempo do ciclo, permitindo isto a redução da quantidade de sucata e uma diminuição do tempo que a máquina está parada. Espera-se, a longo prazo, reduzir as reclamações cliente para 25%.
Com esta solução almejou-se proporcionar aos trabalhadores uma ferramenta de trabalho mais simples, atual e automática, que lhes reduza o esforço e aumente a produtividade.

Contents

Intentionally blank page.

List of Tables

Intentionally blank page.

List of Figures

Acronyms and Abbreviations

MM	Mini Metrology
CMD	*Controlo de Medida de Dentado*
LEMM	*Laboratório de Ensaios Mecânicos e Metalúrgicos*
N3	*Nível 3*
CNC	Computer Numerical Control
PLC	Programmable Logic Controller
PS4N	*Plan de Surveillance à 4 Niveaux*
SIL	*Sistema de Informação Laboratorial*
NVA	Non Added Value
RFID	Radio Frequency Identification
Wi-Fi	Wireless Fidelity
IoT	Internet of Things
I2C	Inter-Integrated Circuit
ISO	International Organization for Standardization
IIoT	Industrial Internet of Things
API	Application Programming Interface
UPC	Universal Product Code
EAN	European Article Number
GS1	Global Standards One
QRcode	Quick Response Code
CPU	Central Processing Unit
LF	Low Frequency
HF	High Frequency
UHF	Ultra High Frequency
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DBMS	Database Management System

NoSQL	Not Only SQL
SQL	Structured Query Language
VSM	Value Stream Mapping
USB	Universal Serial Bus
PFF	*Peças Fora de Fluxo*
UET	Unité Élémentaire de Travail
REST API	Representational State Transfer Application Programming Interface
TCP/IP	Transmission Control Protocol/Internet Protocol
SDK	Software Development Kit
RDBMS	Relational Database Management System
I/O	Input/Output
IP	Internet Protocol
ID	Identification
HTTPS	Hypertext Transfer Protocol Secure
URL	Uniform Resource Locator
JSON	JavaScript Object Notation
UI	User Interface
RTLS	Real-Time Location System
UML	Unified Modeling Language

Chapter 1

Introduction

Since the beginning of the first industrial revolution, there has been an increasing need for the industry to adapt to new technology in order to provide the best product possible to customers. As Industry 4.0 took off, businesses realized they needed to adapt in order to compete in the market.

To achieve these goals, the industry has implemented a variety of quality control strategies, including 3D control, visual control, and machine parameters control, among others. Several of these tests require that a component be removed from the normal supply chain flow and analyzed in quality control centers.

Companies use various methods of traceability and inventory management to keep track of parts that have left the normal production flow.

Nowadays, older traceability systems implemented on the factory floor make it difficult to store data and quickly access the information required to solve problems in a reactive manner. As a result, the need for new traceability solutions is becoming more pressing than ever.

1.1 Context

With the necessity of testing the quality of the production, Renault Cacia has built seven quality control centers as can be seen in figure 1.1 below.

They are classified according to the types of tests performed.

MM (Mini Metrology) is where the metrology tests are performed, CMD (*Controlo de medida de dentado*) where the pitch of the teeth is measured, *Proptê* is the chemical laboratory where tests to the water from the washers, as well as the oils and other fluids, are performed, LEMM(*Laboratório de ensaios mecânicos metalurgicos*) where the destructive tests are performed in order to analyze the inner section of the parts, N3(*Nível 3*) where the testes regarding the functionality of the final products are performed.

Because there are so many topics to cover, a case study involving module three of atelier two from the department responsible for gearbox production will be used to facilitate the understanding of the production line. The gearbox crankcase (figure 1.2) used in the fabrication of JT4 gearbox models is manufactured in this module.

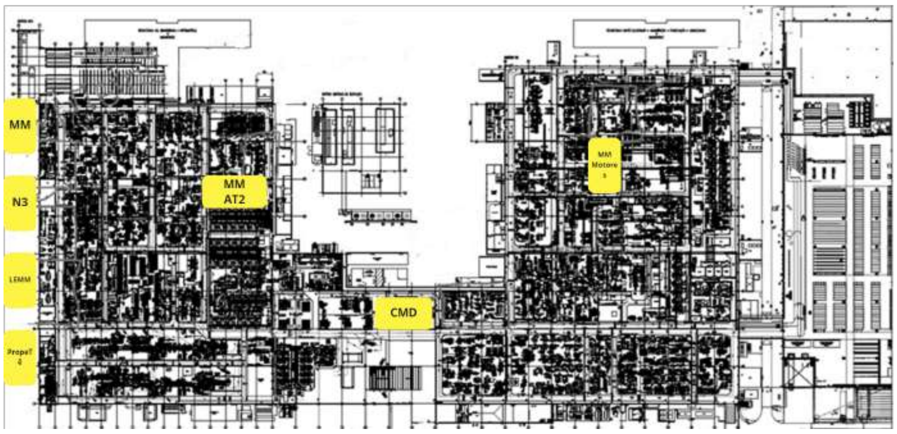


Figure 1.1: Layout of factory quality control centers

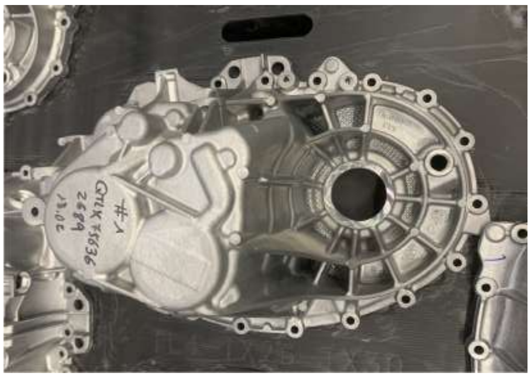


Figure 1.2: Carter TX26 produced in module three

1.1.1 Production flow

It is necessary to first comprehend the standard production flow. The normal production flow is shown in green in figure 1.3, where the raw material (figure 1.4a) cumming from our supplier is introduced in the reintroduction point (marked with an "A" in figure 1.3), at this point the operator does a visual inspection, after which the part enters the machining center area (marked with a "B" in figure 1.3), where another operator inserts the part into one of the various CNC's (Computerized Numerical Control) programmed to produce that type of gearbox crankcase (figure 1.4b).

Following that, the part is placed in a convoy and transported to a washer where all of the grease and oil is washed away. The part then passes through an artificial vision system, figure 1.4c, (marked with a "C" in figure 1.3) where the type of gearbox crankcase is identified with the purpose of registering the production and providing the information to the next step, leak control machine (figure 1.4d), where the robot must place the part in the correct machine where an air tightness test is performed.

If the result is "ok," it is downloaded to the *Balogh* (dynamic tag) attached to the transporter of the part, which is then read by the PLC(Programmable Logic Controller) in the next station and the part is automatically identified with a barcode label.

The operator then reads the barcode label to enter the production data into the system, performs a visual quality control, and finally places the parts in a cart that will

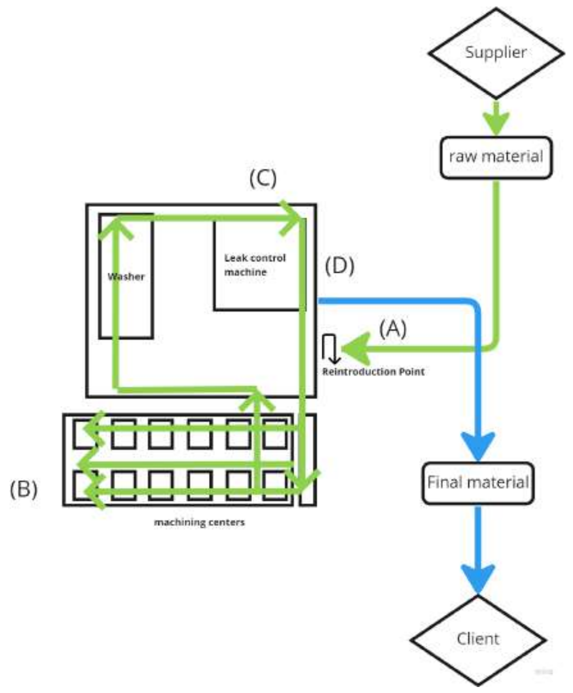


Figure 1.3: Module 3 layout and production line schematics (A)1.4a, (B)1.4b, (C)1.4c, (D)1.4d

be delivered to the client (marked with a "D" in figure 1.3).

1.1.2 Out of flow parts

Quality tests on the machining parameters must be performed to ensure the quality of the parts produced. To accomplish this, Renault Cacia has implemented a protocol known as PS4N (*Plan de Surveillance à 4 Niveaux*), which mandates a schedule of quality control tests for each machine or part in the factory.

In this project use case, this protocol implies that the operator must send a part produced in each CNC to the quality control centers twice a day. In these control centers, the technician performs a metrology test to ensure that the part's dimensions match the tolerances defined by the client.

The protocol also requires that after each intervention in the machine, the next part fabricated in that machine be tested to ensure that the machine is operating properly.

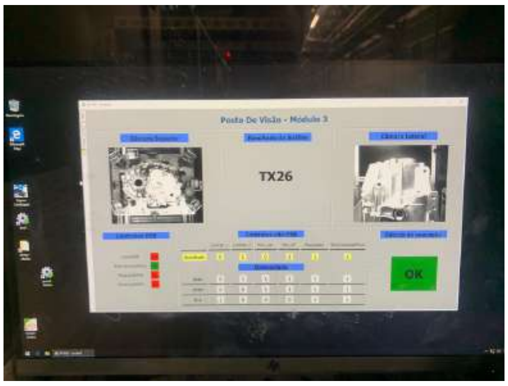
Renault Cacia has today implemented an analog system for keeping track of these parts, which consists of manually filling out two labels as shown in figure 1.5, and a digital request form for the quality control test. This request is made in an application called SIL(*Sistema de Informação Laboratorial*) in which the operator places the order to test the part, as shown in figure 1.6. Here, the operator fills out a form with all of the information about the part, the characteristics he wants to test, and his contact information.

This data is then saved in a SharePoint database and assigned to the request number.

The red label (figure 1.5a) is attached to the part and transported to the quality control centers where the parts wait to be tested (figure 1.7).

(a) Raw part placed in the convoy

(b) Machining area



(c) Artificial Vision result

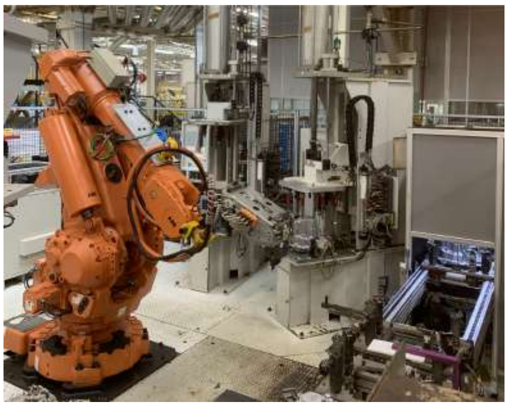

(d) Air tightness control station

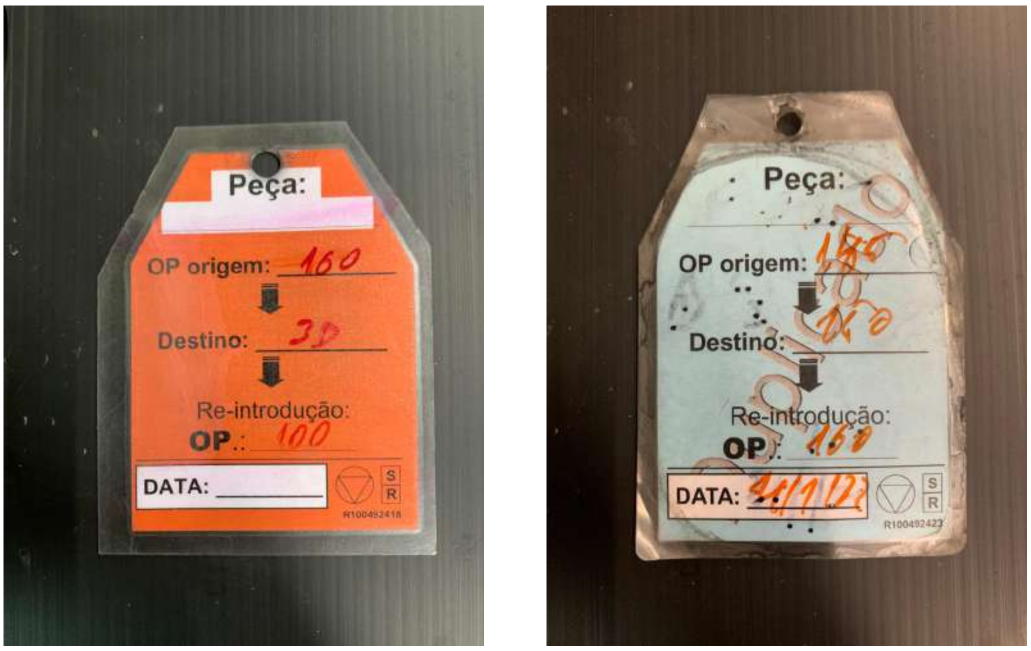
Figure 1.4: Actual components of module 3, in correspondence with the figure 1.3

The blue label (figure 1.5b) is affixed to a board near the workstation. This board, as can be seen in figure 1.8, represents the layout of the machines in that sector. It also contains the workstation designation and the type of parts produced as title, the number of operations, and the order in which they must be completed.

The board also includes a set of icons that help the reader understand where there is a higher, medium, or lower risk of abnormality, where the reintroduction points are, and where they need to perform a visual control.

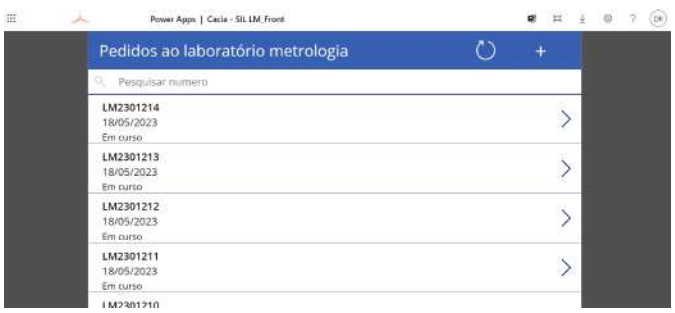
The record of which parts have been sent to be analyzed can be seen in the lower right corner, where each row represents a CNC machine and the columns represent the date and the outcome of the quality control center's quality test.

Once inside the quality control center, the technician consults the application on



(a) Red tag (b) Blue tag

Figure 1.5: Out-of-flow parts identification



Pedidos ao laboratório metrologia

Pesquisar numero

LM2301214
18/05/2023
Em curso

LM2301213
18/05/2023
Em curso

LM2301212
18/05/2023
Em curso

LM2301211
18/05/2023
Em curso

LM2301210

Figure 1.6: Quality control application (SIL) front page

Figure 1.7: Parts waiting to be tested

another computer and begins the test after gathering the necessary information. After the test, the technician goes to the application and uploads the results so that the

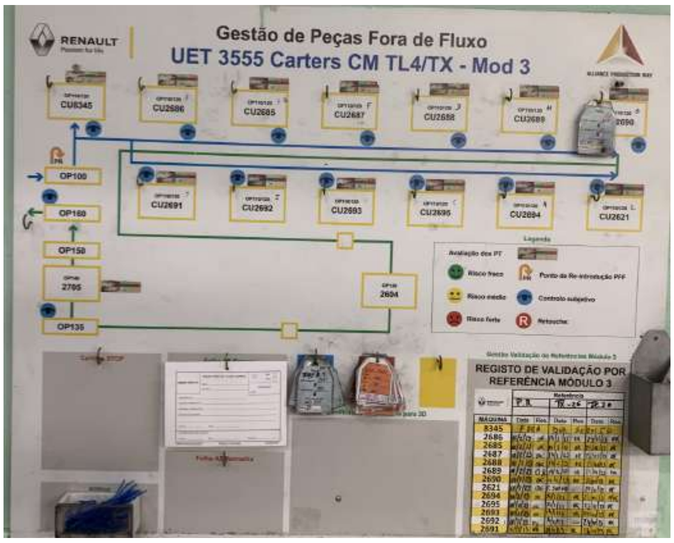


Figure 1.8: Out-of-flow parts management board

operator is notified of the outcome by email, although if the part has some abnormality the technician must call the operator's cellphone to inform the result.

To understand the steps of this procedure, figure 1.9 shows the sequence of tasks the operator must perform in order to send the part to the analytic.

The area marked with number one is where the operator normally stands and in front of him there is a schedule of when he must send the parts to the control, the area with the number two is where the record board is placed and the area with the number three is where the parts wait for testing and are tested.

The process is as follows: first, the operator selects a part from the specific machine, then goes to the board and grabs a blue label and a red label, then goes to area one and fills out the necessary information on labels, then places the red label in the part and places the part into the cart to be sent to the quality control center, and finally, the operator goes to place the blue label in the board.

After filling the trolley with a part from each CNC machine, he transports it to the quality control center, where he fills the request for the control.

Soon after all the parts have been tested, he grabs them all and reintroduces them in the line at the reintroduction point to maintain normal flow.

1.2 Problem

In this context, the studied problem is the lack of traceability of parts that leave the normal production flow, which is present in every process of the Renault Cacia production line.

According to Renault Cacia, this topic now accounts for roughly half of all client complaints. That is related to the fact that parts that have been tested and have abnormalities are reintroduced in the normal flow by human error, or even parts that are supposed to enter at a specific point of reintroduction, are reintroduced at another point, skipping some operations critical to the normal functionality of the part.

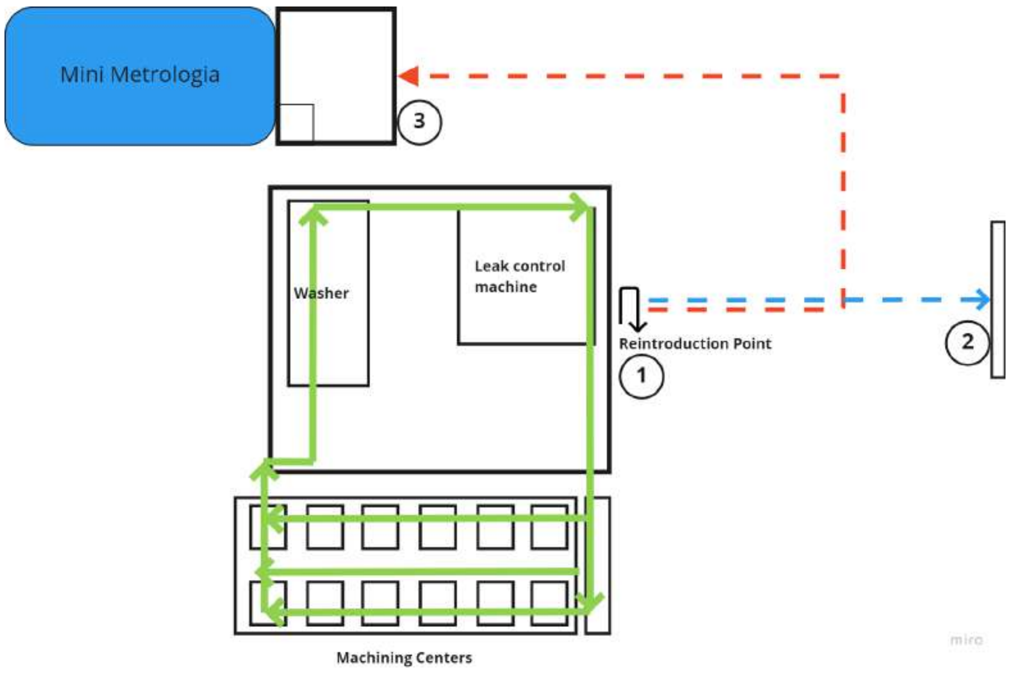


Figure 1.9: Out-of-flow parts flow [1. Reintroduction point/ workstation of the operator; 2. Out-of-flow board; 3. Minimetrology control center]

In addition, there are issues concerning the time spent by the operator manually filling out the forms and labels, as well as the after-work of cleaning them, which contributes to the NVA (No Value Added) of this process, as well as the possibility of losing the identifying labels during transport, and the lack of reactive response, such as locking a shipment, changing tool or fixing the machine parameters, in the event of an abnormality detected during the test.

1.3 Objective

It is intended to develop and implement a solution that allows for the detection and control of out-of-flow parts to improve reactive response in the event of an abnormality.

The ”want to be” is to implement a system that has a database to store all the information regarding the out-of-flow parts, the real-time location of the parts, and an interface to communicate with the user to place the requests and get the results regarding the state of the machines.

This system must also be one hundred percent digital, user-friendly, adaptable to all the modules of the plant, and must be connected with all the applications already in use.

1.4 Methodology

This project's methodology is based on the Agile methodology.

The Agile methodology is a widely accepted project management framework that focuses on delivering a high-quality product in small iterations.

It is based on four core values, including prioritizing individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, and responding to change over following a plan. [10]

Additionally, it is guided by twelve principles that encourage early and continuous delivery of a valuable product, regular communication between business people and developers, and the emergence of best designs from self-organizing teams.

Although the Agile methodology is more commonly used in software development, it is an excellent tool for project management in general because it provides numerous benefits such as faster delivery of working products, improved collaboration and communication, and a focus on customer satisfaction.

1.5 Document Structure

- **Chapter 2 - Literature Review:** Starting with a look at what other writers have done in similar situations led to a search for technologies and tools that could be used in this project. Concepts about these tools and technologies are covered in this chapter.
- **Chapter 3 - Solution proposal:** It covers the proposed solution. In this part, the proposed design for the project's growth is shown, along with how the ideas, tools, and computer languages that will be used will be linked to the project's suggested goals.
- **Chapter 4 - Solution Implementation:** The objective is to explain how the proposed solution was placed on the factory floor of Renault Cacia. A general plan for implementation is given, along with a description of the main functions and ways to use the built-in graphic interface. Each piece of equipment is also listed, along with how it fits into the traceability system.
- **Chapter 5 - Conclusion:** Discussion of the findings and conclusions resulting from the project's analysis.

Chapter 2

Literature Review

This chapter examines the solutions already published for similar problems as well as the concept of Industry 4.0, traceability, and technology used in existing traceable systems. Subjects for data acquisition, processing, and databases, as well as communications used in this type of solution, will also be covered.

2.1 Similar works

2.1.1 "Proposal of an automatic traceability system, in Industry 4.0"(2018), Universidade de Aveiro

This project's [2] solution is based on the deployment of a customizable traceability system that employs four RFID (Radio Frequency Identification) readers to track gearbox crankcase manufacturing on a production line.

As can be seen in the diagram of figure 2.1 the system is composed of three levels. The first layer comprises of RFID readers, the second of an ESP32 IoT(Internet of Things) module (processing unit), and the third of a server that communicates with the ESP32 through Wi-Fi(Wireless Fidelity).

In terms of operation, the readers continually checked for the existence of an RFID label, and if one was detected, it was transferred to the processing unit, which was in charge of transmitting and reading information important to the manufacturing, of the associated gearbox crankcase label. Simultaneously, the ESP32 sends this information to a server via Wi-Fi, which receives and delivers data and messages from the database.

Following that, a graphic interface was created utilizing several web languages in order to more quickly visualize and update information.

The author describes the implementation as successful since he accomplished all the objectives with a working solution.

2.1.2 "Integração de Sistemas de Rastreabilidade em Ambiente Industrial"(2013), Universidade de Aveiro

The integration of two distinct types of RFID scanners is the recommended solution in this project [3]. To combine them, the microcontroller (processing unit) is in charge of managing the messages sent between readers and the local computer, where an application has been developed for reader control and database interaction. This approach

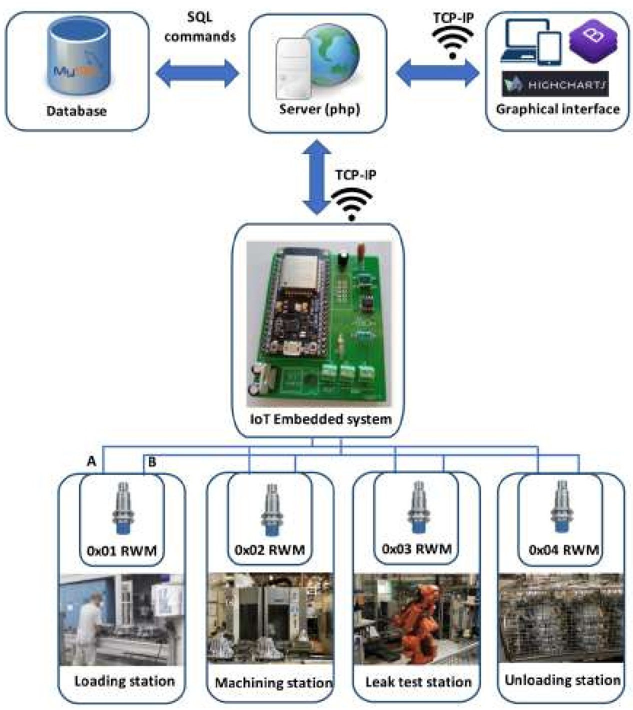


Figure 2.1: D. Rocha's proposed architecture [2]

tries to associate a new label with the product while keeping the existing label and transferring information from one to the other.

As shown in figure 2.2, the system consists of four RFID readers: two Sparkfun SM130s that communicate with the micro-controller via the Inter-Integrated Circuit (I2C) and use the International Organization for Standardization (ISO) standard 14443A, and two Contrinex readers that communicate via RS485 and relay on the ISO standard 15693. The data acquired by the four readers is processed by a microcontroller, which sends the data to a server. Once the information is on the server, it is moved to a database and then delivered to employees via a customized graphic interface for a better ergonomic view of the information.

With his dissertation, the author concluded that there was an incompatibility between traceability systems in various supply chain partners, being necessary to integrate new systems in the businesses. Some of the difficulties encountered were the system integration and the associated costs. The author also mentions the growing use of RFID technology in industrial rastreabilidade systems due to the advantages it offers over other technologies.

2.1.3 "Sistema automático 4.0 de rastreabilidade e atualização de OF na OLI"(2020), Universidade de Aveiro

The goal of this work [4] is to automate the updating of manufacturing orders. To that end, the author proposes the architecture depicted in figure 2.3, where he pretends to obtain data related to the end of a pallet's processing via barcode reading and transmitting it to the company's network, and registering it in the database, allowing for an

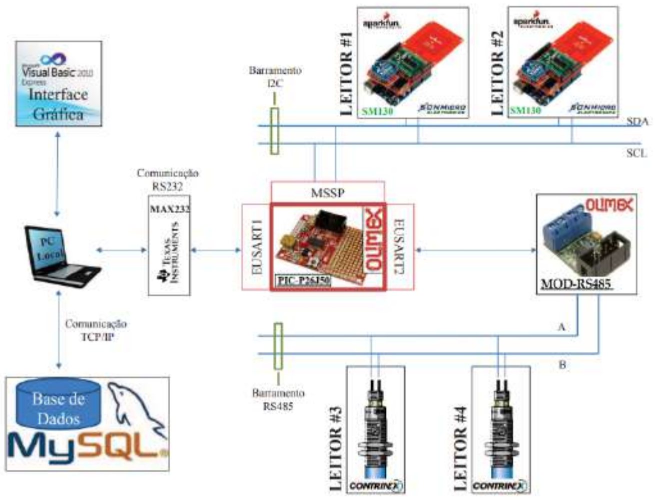


Figure 2.2: Bártolo's proposed architecture [3]

update of manufacturing orders and product traceability.

The author used ESP32 and Raspberry Pi as hardware in addition to devices already implemented in the company such as PLCs and barcode readers. Nodejs was the choice for software to manage the back-end in JavaScript and generate a front-end.

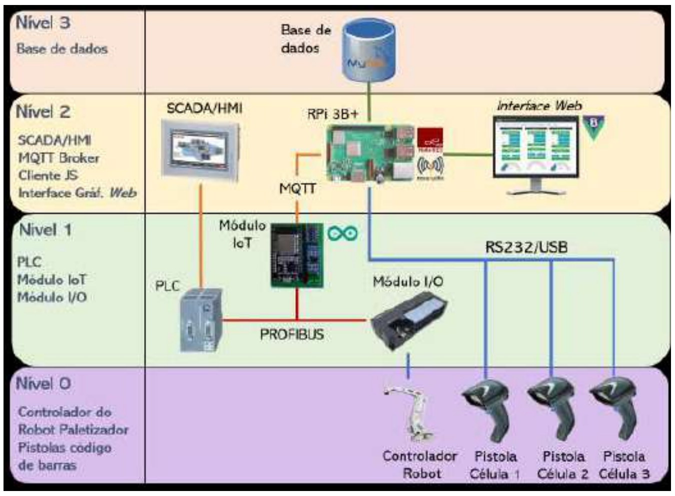


Figure 2.3: F. Rocha's proposed architecture [4]

In conclusion, the author asserts that this type of information record is essential for the continuous updating of finished products and that the elimination of human resources in this type of database is essential for reducing cycle time and avoiding human errors.

When it comes to the performance of the solution's implementation, the author claims that the system has shown to be reliable.

2.2 Industry 4.0

Industry 4.0 refers to technical advancements in industrial production and monitoring systems. These developments result in an industry geared towards service rather than product, in which corporations offer their goods with accompanying services in order to retain clients, resulting in tighter control and monitoring of the process from raw material to finished product [11].

Another significant development associated with the fourth industrial revolution's demand is the formation of new types of industry-related firms, which are required to aid industries in deploying, managing, and maintaining new technologies. [12]

In general, Industrial Internet of Things (IIoT) systems should follow a set of guidelines, including the use of software and open-source communication protocols to enable system connectivity and the use of APIs(Application Programming Interface) in system interactions to facilitate integration with other solutions.

2.3 Traceability

Based on ISO 8432, ISO 9000 and ISO 22005 the definition of traceability can be determined as the capacity to access any or all information regarding the subject matter during its full life cycle via documented identifications. [13]

The advantages of adhering to traceability notions are mostly linked to a more reliable knowledge of current stock, a faster response to discovered faults in the production process, and the assurance of quality and degree of productivity.

2.3.1 Traceability in automotive industry

Customer safety has long been a primary concern in the automotive sector, which has caused leading manufacturers to be extremely demanding in terms of raw materials and manufacturing process quality [14].

In the context of increasingly severe regulations such as ISO/TS 16949, suppliers and manufacturers must apply various controls to ensure the quality of the components manufactured. To satisfy these criteria, manufacturers are implementing complex traceability methods that enable monitoring of the component's production process at all stages. Each manufactured vehicle is granted a unique serial number in accordance with regulations, and it is composed by using a wide range of components, each one of them also marked with a serial number.

As a result, traceability is an essential need for automotive industries to remain competitive on a national and international scale. The tools available today allow for high levels of success in transportation, bringing benefits to the industry through process control, cost savings, and security.

Adoption of current traceability techniques may enable enterprises to differentiate themselves in the market by providing high-quality, trustworthy goods.

2.4 Acquisition

In terms of data capture, the options are expanding and diversifying, but this section will highlight the most common ones currently in the automotive sector in terms of traceability.

This is an extremely important topic in this project since proper data collecting is required in order to get an improved result.

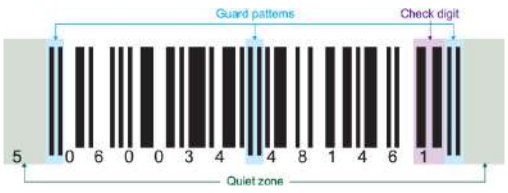
2.4.1 Barcode

Barcodes are one of the leading traceability technologies. They may be found in both the retail and manufacturing industries. Overall, barcodes outperform manually input data in terms of accuracy, traceability, and sorting.

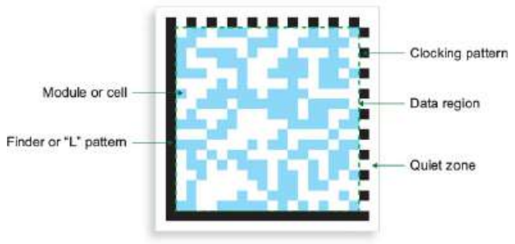
As with many other traceability systems, there are two components: the reader (scanner) and the label, which holds the data to be read. There are basically two types of barcodes:

1. **Unidimensional (1D):** These linear codes solely store alphanumeric information. Each of the characters in the code symbolizes a distinct aspect of the product, and a database explains what each character implies. The widths of the gaps and bars each correspond to a different character in the barcode. The white area to the left and right of the barcode is known as a silent zone or margin (figure 2.4a). The most prevalent codes are UPC (Universal Product Code), EAN (European Article Number), and code 128 according to the GS1(Global System 1), a global organization that regulates good industrial practices [15].

2. **Bidimensional (2D):** This type of barcode can contain data horizontally as well as vertically, enabling it to store far more data. It comprises black and white areas with clearly defined zones such as a quiet zone, a finder zone, and a clocking pattern zone (figure 2.4b), that assure proper reading orientation. QR codes and DataMatrix represent the most common 2D codes used in the automotive industry [6].



(a) Description of 1D barcode areas [15]



(b) Description of 2D barcode areas [6]

Figure 2.4: Overview of Barcode 1D and 2D formation areas

Marking Methods

- **Inkjet Printing:** Mostly used in printing labels for 1D barcodes, consists of printing directly the code in the package, label, or other material.
- **Direct part Marking:** More durable than other printing technologies and are most typically employed in printing 2D codes. Some examples of this type of engraving are:

- **Laser:** A laser beam creates a chemical change in the material, which allows for the creation of a mark.
- **Dot Peen:** This marking consists of the use of a pneumatic marker that operates on a surface, producing a series of closely spaced points.
- **Chemical etching:** This technique uses a sodium-based solution that when in contact with low electrical voltage dissolves the metal, which is subsequently removed using a specific stencil.

According to Lazov, Deneva, and Narica (2015), choosing a marking method should include not only the material but also the cost, processing speed and flexibility, and the ability to automate the process.

Types of Barcode Readers

In this section will be presented the types of Barcode readers and their differences.

A barcode scanner, also known as a barcode reader, is a device that decodes and extracts the information contained in barcodes. [5]

- **Laser:** Equipped with a laser beam as a light source, the beam is projected to the code, the black and white strips reflect the light and the reader decodes the signal, as illustrated in figure 2.5.

  This type of reader is the most commonly used.

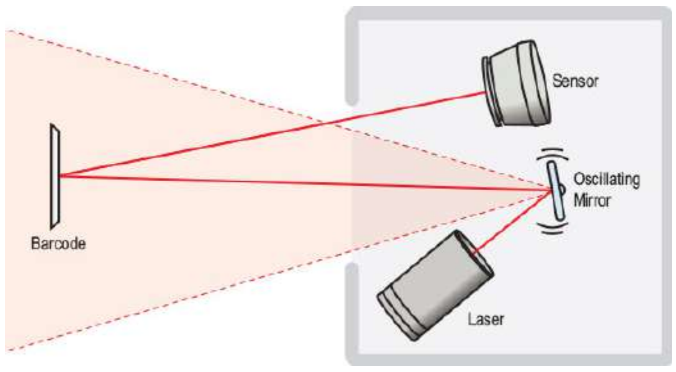


Figure 2.5: Laser reader operation mode [5]

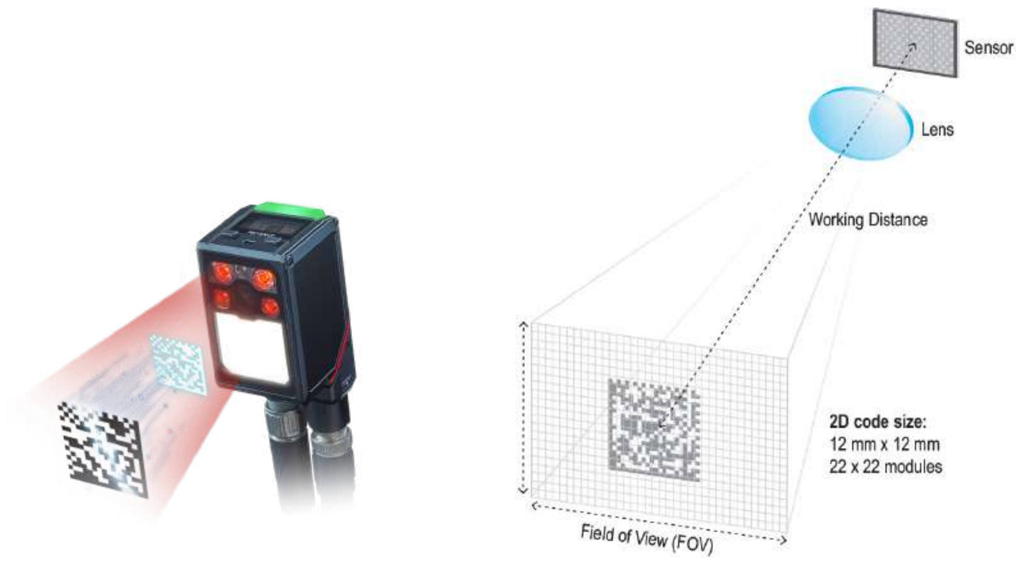
- **Image-based reader:** This type of reader work on the base of an image (figure 2.6a), the code is then located and decoded by a CPU(Central Processing Unit) running specific image-processing software, the built of this devices is depicted in figure 2.6b.

Pros and Cons of Barcodes

This section's purpose is to present the pros and cons associated with the use of barcodes.

Being a widely used technology, the barcode has many advantages associated such as:

- Easy configuration and usage, being able to operate in manual or automatic mode.



(a) Image-based reader (b) Image-based reading process

Figure 2.6: Image-based components and functionality [6]

- Fast and precise reading.
- Low cost and small compared to other technologies.
- Globally spread technology, allowing to maintain product traceability in different countries.

However, it also has some cons associated, for example:

- The readers cannot read and write code, they can only read and process it.
- Limited in terms of range.
- Less resistant and secure.

2.4.2 RFID

RFID (Radio Frequency Identification) has been used for decades. However, it is only recently that the combination of reduced costs and better features has prompted organizations to take an additional look at what RFID may accomplish for them.

This technology similar to many others is composed of two main entities the reader and the tag.

The main idea underlying RFID systems is to label items using tags. These tags may store anything from a single number to a large quantity of data, making it possible to store nearly all the information about a certain product [16].

This allows using tags to trigger events inside the factory, and to register the date and time of these events, as well as store the information directly in the tag if the process is running properly and the location or trajectory made by such identified part.

Both the tag and the reader have an antenna that allows them to communicate. This communication is made through one-way or two-way radio signals, and it is triggered when the tag enters the area covered by the signal of the reader antenna. After that, the reader can read/write information that is stored inside the tag's microprocessor, which manages its memory. [17]

This information can then be passed in real-time to several interfaces and stored in a database, to keep a record of every interaction made in the system, without human intervention.

Tags / Transponders

RFID tags are classified into two main categories, active or passive (figure 2.7), being defined by their electrical power source.

- **Active tags:** Equipped with its own power source, normally an onboard battery. Transmit a stronger signal, allowing the reader to detect their signal from further away. They are more expensive and larger and for that reason, they are more employed in systems that aim to track large items for long distances. These tags can remain asleep until enter the area of a reader and start emitting a signal continuously until exit said area.

- **Passive tags:** Because of the low cost, and diminished size of this type of tags they are the most common in RFID systems implementations. These tags can constantly broadcast their signal or broadcast on demand. Due to their lower complexity, these tags can be incorporated into many materials and products.

  These tags operate based on a radio signal that is sent by the reader, the antenna in the tag receives and converts it into electrical power in order to supply his circuit.

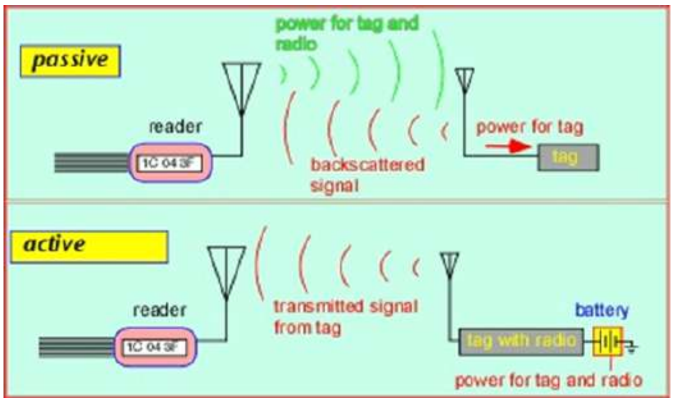


Figure 2.7: Operation mode of RFID passive and active tags [7]

Transmission methods

As for transmission between passive tags and the reader, there are two major methods inductive and backscatter.

- **Inductive:** When the tags enter the reader signal area the reader sends an electromagnetic signal that is received by the tag antenna, this energy is then stored in the capacitor incorporated in the tag.

Afterward building enough energy uses it to power the tag antenna and transmit a modulated signal to the reader, this signal contains the information stored in the tag. [16]

- **Backscatter:** An electromagnetic field is created by the reader. When a tag enters the reader signal area, a portion of the energy generated by the field is used to power up the tag's electrical circuit, while the remainder is reflected to the reader as a modulated signal, which the reader interprets as data. [17]

Frequencies

- **Low Frequency - LF:** With frequency ranges between 30kHz and 300kHz, his reading range is limited.

  Normally utilized in access control, food industries, and animal tagging, it is also a good solution for metallic environments since they suffer less from reflective noise. [18]

- **High Frequency - HF:** With a frequency range between 3MHz and 30MHz, it is the frequency most employed in the business.

  Because it works reasonably well in metallic surroundings, it is commonly utilized in the logistics sector to regulate product flow. [18]

- **Ultra High Frequency - UHF:** With a frequency range of 300MHz to 1GHz, this is the frequency with the greatest read range and data transfer rate.

  Despite these advantages, the signal is more susceptible to metallic noise and is not globally standardized, making it less possible for a tag to be read in different countries. [18]

Pros and cons of RFID

Radio frequency identification, a technology existing for years, has potential uses in a variety of applications, being a great alternative to barcode technology, has many pros such as [19]:

- Versatility in stored data
- Readers can read and write
- Bigger data storage capacity
- Reuse of tags with new data
- Wider reading range
- More resistant to a hostile environment
- Simultaneous reading of numerous tags

Despite all these pros, it has its own cons, such as:

- UHF frequencies not normalized globally

- Lack of normative homogeneity
- Expensive compared to barcode tags
- More affected by a metallic noise

Overall this is a technology that analysts expect to become widespread in the coming years, helping organizations solve problems in supply chain management, security, personal identification, and asset tracking. [16]

2.5 Processing

The term Data processing is directly related to the action of translating data that had been collected and transforming it into a readable format, to be easily interpreted and utilized by the end user in the organization [20], [21].

This section will be presented the most commonly used frameworks to develop both back-end and front-end scripts. These components help process and display data to the end user, making it easier to understand and work with.

To comprehend what will be addressed, it is important to first define a framework. A framework is a collection of libraries and modules that enable programmers to compose software and build applications without being concerned with minor features.

2.5.1 Server-side

The server-side or back-end is the data access layer from a software or device that is not directly accessible to users. It saves and organizes data while also guaranteeing that everything on the client side of the website functions correctly. [22], [23].

Back-end operations include creating APIs, developing libraries, and dealing with system components that lack user interfaces or even scientific programming systems. [24].

Furthermore, it comprises the logic of the program that manages data and communicates to the website's peripherals, such as a database or additional data collection devices [24].

Currently, the development of back-end solutions in significant corporations is divided between Python and JavaScript as the two primary languages. This section will discuss some of the most popular back-end development solutions and tools, depending on the aforementioned languages.

Flask

Flask is a Python-based web framework that facilitates the creation of web applications [25].

Due to the fact that it is written in Python, facilitates the development of web applications for inexperienced developers. It also provides access to programming libraries and tools within the Python open-source ecosystem [26].

Regarding documentation and use cases, Flask is widely referenced on the internet, enabling new developers to learn and execute projects despite their limited capabilities [27].

Some of the features that Flask brings to its users are [28]:

- It has incorporated a development server and a debugger which enables the developers to test and quickly understand their mistakes.
- It offers full assistance for unit testing.
- Easy to understand and code
- Extensive libraries
- Open-source
- Easy to connect with other languages
- Client-side Cookie Support that facilitates the application's operation
- HTTP (Hypertext Transfer Protocol) request processing capabilities
- API foundation formed and consistent.

Despite all the features, there are also some limitations such as:

- Limited built-in features
- Designed for small to medium-sized applications
- Flask does not come with built-in security features

Even though the existence of these limitations, they can be avoided through the use of third-party libraries and extensions.

Flask is relatively young compared to the majority of Python frameworks, it is rapidly gaining recognition among Python web developers. Because Flask is one of the most widely used web frameworks, it is actualized and modern. Capable of extending its functionality and scaling to accommodate complex applications [26], [27].

Nestjs

NestJS is a Node.js framework that focuses on modularity, extensibility, and testability to build robust and scalable server-side applications. It relies on TypeScript to create better organized and maintainable code and has a robust dependency injection framework [29].

NestJS is built on popular libraries and frameworks, with built-in modules for common functionality. Its modular construction enables scalability and simple maintenance.

It is extremely fast and supports both synchronous and asynchronous programming. NestJS is a flexible framework for developing contemporary, high-performance server-side applications [30].

Flask/NestJS

The table 2.1 shows a comparison between the two back-end frameworks. They will be compared regarding the framework type, database integration, testing support, scalability, community support, and learning curve.

Table 2.1: Backend Framework comparison

Backend comparison		
Feature	**Flask**	**NestJS**
Language	Python	TypeScript
Framework Type	Micro-framework	Full-stack framework
Database Integration	Through extensions	Built-in support
Testing Supports	Unit testing, but no built-in	Built-in testing framework
Scalability	small / medium-sized projects	Large / complex projects
Community Support	Large and active community	Growing community
Learning Curve	Easy	Moderate/High

2.5.2 Client-side

The client-side or frontend, often known as the UI (User interface), of a website is the portion of the website with which the user interacts. It includes everything that can be seen and touched [24], [22].

HTML(Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript are frequently used as the primary languages for front-end development.

This section will describe frameworks that enable developers to construct responsive and performant solutions.

Angular

Angular is a Google open-source JavaScript framework that has gained popularity among developers as a result of its flexibility, modularity, and support for current web development standards [31].

Angular allows the creation of single page applications and employs a modular design, making it simpler to create large-scale applications with complicated user interfaces.

Its support for dependency injection and usage of two-way data binding makes it easier to design reactive and flexible user interfaces.

Angular is built on TypeScript, a JavaScript variant that makes Angular applications more reliable and more straightforward to maintain. The wide ecosystem of third-party tools and the active community make Angular a framework that will continue to develop and evolve in the years ahead [32].

However, Angular, has several limitations, such as a long learning curve, possible performance concerns for high-performance applications, and its size, which might be an issue for smaller projects. Compatibility might also be an issue with Angular due to multiple significant upgrades [33].

React

The React.js framework is an open-source JavaScript framework and library. It's used to rapidly and efficiently create reactive user interfaces and applications [34].

React, allows the creation of multiple blocks that when merged, form the user interface for the application.

The ReactJS framework combines JavaScript's speed and efficiency with a more efficient way of rendering web pages quicker and creating highly dynamic and responsive online applications [35].

VUE

Vue is a JavaScript framework designed to create user interfaces. It is built on top of conventional HTML, CSS, and JavaScript and provides a component-based and declarative programming method that allows developers to rapidly construct basic or sophisticated user interfaces [36].

Vue is a framework and environment that covers the majority of the standard front-end development capabilities. However, the web is very varied, and the things built on it can vary greatly in shape and scope. With this in mind, Vue is intended to be adaptable and gradually adopted [36].

Angular/React/VUE

The table 2.2 depicts a comparison between the three front-end frameworks mentioned before. These frameworks are compared taking into consideration, the learning curve, the performance and the ecosystem.

Table 2.2: Frontend Framework comparison [1]

Frontend comparison			
Feature	**VUE**	**React**	**Angular**
Language	Vue.js	JavaScript	TypeScript
Learning Curve	Easy	Easy/Moderate	Moderate/High
Performance	Good	Good	Good
Ecosystem	Smaller than React and Angular	Largest of the three	Large and growing

2.6 DataBases

"Data are facts" [37]. In today's industry, data is one of the most important tools, allowing to know better the process and understand in which sector improvements can be made.

With the increasing flow of data, it is vital to store it in a safe and effective manner while also allowing simple access anytime it is required.

This section will cover the most commonly used data storage tools in the business.

A database is a structured collection of information or data that is often kept electronically in a computer system [38].

Data in a database is structured into tables with rows and columns and is indexed so that it may be quickly updated, extended, and destroyed.

However, in order to do such commands it is necessary to use a database management system (DMBS) a complete database software system that acts as a bridge between the database and its end users or applications. Some examples are MySQL, Oracle, PostgreSQL and SQL Server. [38]

Databases come in a variety of forms, ranging from the most common, the relational database, through a distributed database, a cloud database, and NoSQL(Not Only Structured Query Language) databases [39].

- **Relational Database:** A relational database's items are structured as a series of tables with columns and rows, that interconnect by using primary keys and foreign key. Relational database technology is the most efficient and adaptable method of accessing organized data. [38]
- **Distributed Database:** A distributed database is made up of two or more files that are stored in various locations. The database might be kept on numerous computers, all in a single geographical place, or spread over multiple networks. [38]
- **Cloud Database:** A cloud database is one that is generally hosted on a cloud computing platform. Database service gives database access and hides the underlying software stack from the user. [39]
- **NoSQL Database:** Non-relational database, permits unstructured and semi-structured data to be recorded and managed. As online applications got more frequent and complicated, NoSQL databases gained popularity. [38]

Despite the fact that there are other kinds, the relational database is the most often utilized. This type of database relay in some integrity constrains such as [40]:

- **Entity:** States that the primary key value cannot be null.
- **Referential:** States that if a foreign key in a table refers to the primary key of other table, every foreign key value must be null or be available in the other table where it represents the primary key.
- **Domain:** Specification of a valid set of values for an attribute. For example character, integer, date, and others.
- **Key:** Every table can have multiple keys but only one can be the primary key of such table.

These rules are applied in order to create a well structured database.

SQL

Structured Query Language (SQL), invented at IBM, is the standard computer language for communicating with relational database management systems, allowing database administrators to quickly add, modify, or remove rows of data, only by writing a few lines of code [41], [42].

This coding language was developed so that developers may access and edit data in a consistent and straightforward manner.

It features several fundamental commands, like as SELECT, INSERT, UPDATE, and DELETE, but it also has many others, allowing developers to design a query that presents just the necessary information to the end user. [41]

The fact of it is simple to use, this language has become one of the most popular for database administration.

Donald D. Chamberlin, co-creator of SQL language, said "SQL has been a more successful query language than Ray Boyce and I had any reason to expect in 1974.". [42]

2.7 Power Apps

Power Apps is a Microsoft product that includes a data platform, connectors and services, that create a fast development environment to build applications.

This product is a user-friendly platform that allows even inexperienced developers to design an easy-to-use application in a short period of time. [43]

However, most of the developers of this type of application use SharePoint as their database. [44]

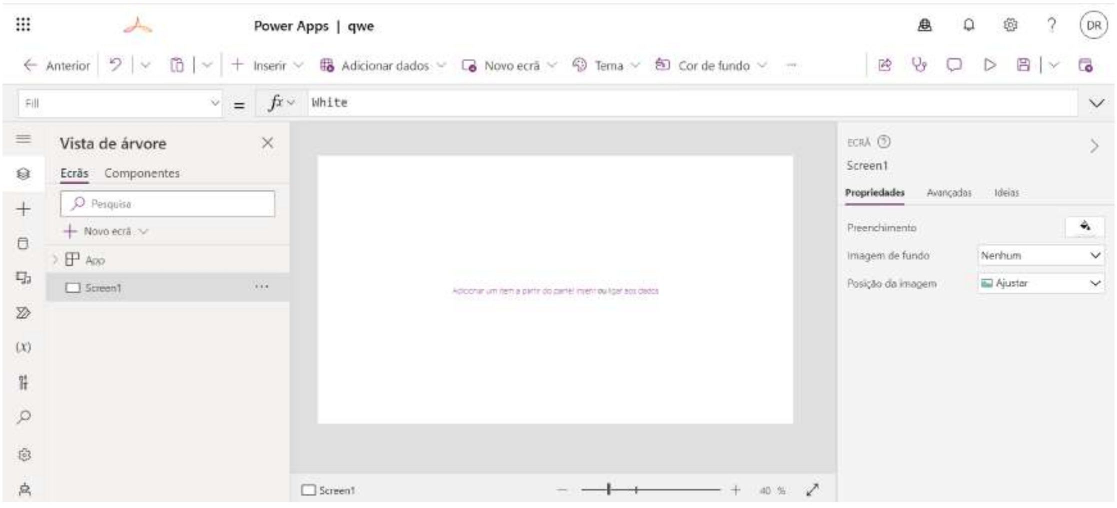
SharePoint is a Microsoft Platform that enables users to manage documentation, store files, organize, share, and edit documents in collaboration with others. [45].

In the words of Ted Pattison and Andrew Connell, "It is safe to say that SharePoint technologies have made into the mainstream of software products used by companies and organizations around the world".

Analyzing the utilities of SharePoint as an all, this product brings many advantages to collaborative work and file management. [46]

However, SharePoint's functionalities as a database are not even close to what a database should offer. [44]

In figure 2.8 is possible to see the main menu for creating an application with Power App.



Fill = White

Adicionar um item a partir do painel Inserir ou ligar aos dados

Preenchimento
Imagem de fundo: Nenhum
Posição da imagem: Ajustar

Figure 2.8: Development page for Power app applications

Intentionally blank page.

Chapter 3

Solution proposal

3.1 Buildup

To better understand the problem and to achieve a solution that corresponds with the client's necessities a schematic was developed (figure 3.1).

This diagram depicts the VSM (Value Stream Map) of the operation under consideration, where can be seen the layout of the area and in the lower portion of the figure the process stages description.

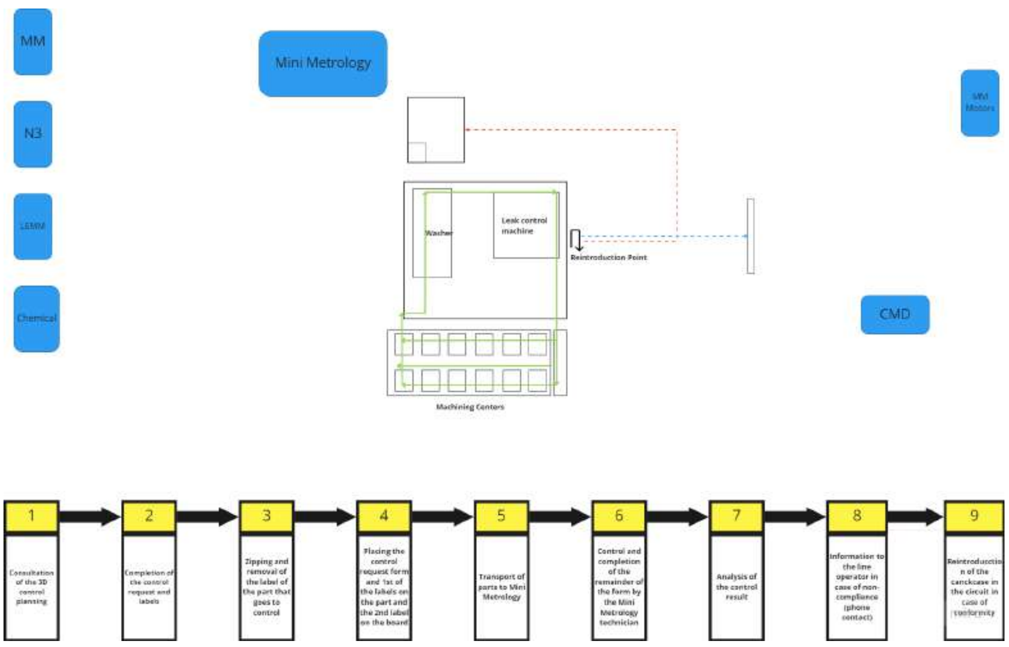

MM
N3
LEMM
Chemical
Mini Metrology
MM Motors
Washer
Leak control machine
Reintroduction Point
CMD
Machining Centers
1
Consultation of the 3D control planning
2
Completion of the control request and labels
3
Zipping and removal of the label of the part that goes to control
4
Placing the control request form and 1st of the labels on the part and the 2nd label on the board
5
Transport of parts to Mini Metrology
6
Control and completion of the remainder of the form by the Mini Metrology technician
7
Analysis of the control result
8
Information to the line operator in case of non-compliance (phone contact)
9
Reintroduction of the crankcase in the circuit in case of conformity

Figure 3.1: Value Stream Mapping of Out-of-flow parts

After the operation mapping was completed, it was presented to the clients(figure 3.2), who were asked to write down what they perceived to be the problems and what features they desired to see in the final product.

With the insight given by the client, it was possible to group the notes and divide them by theme.

Following that, the participants were given four votes to cast on the necessity of implementation.

This exercise culminated in identifying the problems previously described in section 1.2, as well as the objective definition presented in section 1.3. This exercise also allowed the identification of the following requisites:

1. Non-intrusive solution: The solution must not interfere with the already existent process

2. Universal solution: The solution must be applicable to the largest number of similar situations.

3. Integrated solution with Renault Cacia and Group Renault services: The solution shall be developed within the rules of software development imposed by the group Renault, and integrated into the informatics department services.

4. Accessibility: The solution must be accessible from any device within the factory and user friendly.

5. Low-Cost Solution: The solution must maintain the costs for development and implementation at a minimum.

Figure 3.2: First Client Meeting

3.2 Proposal

With the client requisites in mind, a proposal for a solution architecture was developed as shown in figure 3.3.

The proposed solution is intended to integrate the existing software and hardware in use in this factory sector.

The system is divided into three areas the borderline area in the reintroduction point, the second being the mini metrology area and the last being the factory as the gatherer of all the remote components.

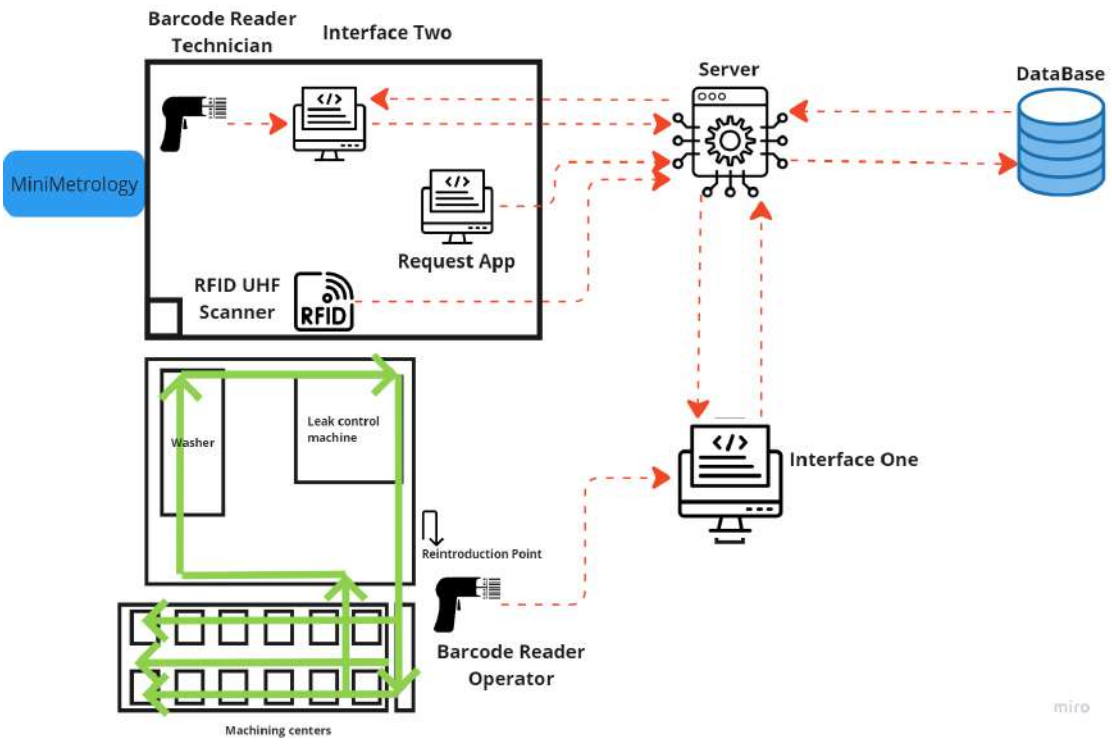


Figure 3.3: Operation principle diagram

3.2.1 Border Line components

There is already a monitor in the area of the reintroduction point, which is the border of the production line, where the operator can monitor other applications in use at Renault, and where the application that this solution will generate will be displayed. This area also includes a barcode scanner, which the operator uses to declare production and register completed parts.

The barcode scanner and monitor communicate via USB(Universal Serial Bus), and the barcode scanner reads the code and transcribes it to the application, much like a keyboard.

In this area will be expected that the operator associates a tag to the part and places the tag in the part.

3.2.2 Mini Metrology components

The mini metrology space is outfitted with several computers, one of which is located near the entrance and is currently used to place testing orders.

The first approach in the solution proposed will be to integrate this already-in-use application with the application to be developed (for easier understanding the application to be developed will be called PFF(*Peças Fora de Fluxo*)). In order to reduce information filling by the operator, the PFF will get some data from the SharePoint database in early life.

It is intended to merge the two applications and remove the SIL application once the PFF has all of the required functionalities.

The mini metrology space will also be outfitted with a barcode scanner, eliminating the need for the technician to go to the computer and search for the part to be tested.

Instead, he will simply read the tag placed in the part and the information about that part will be displayed on the computer.

RFID hardware will be implemented in this space to detect tags attached to parts and thus register the entry and exit of the said part in the mini metrology space. with the responsibility of confirming the presence of the part.

3.2.3 Factory components

The factory components represent all the components that are not accommodated in a physical place but belong to the factory infrastructure. In this project, the scripts created for the PFF will run in a remote server environment, so the application can be accessed from anywhere within the factory.

In this server will also be placed a database to store the data gathered from the peripherals.

3.3 Components description

This section will be presented all the components in the solution proposal as well as their utility.

A database, a server, a user interface, a docker, a radio frequency identification module, and a barcode scanner will comprise the system.

3.3.1 Database

To register and keep a record of all interactions with the solution application, all data will be stored in a database, which will be used not only for storing but also for consulting, indicating a two-way communication.

To create a database that meets the needs of the application, it is necessary to first understand the machine hierarchy.

The hierarchy of the machines as represented in the figure 3.4 states that in every department there are multiple ateliers, in every atelier, there are multiple UETs(*Unité Élémentaire de Travail*), in every UET there are multiple modules, in every module, there are numerous operations and finally, in every operation, there can be many machines.

Representing this a one-to-many relationship for this fraction of the database.

The second fraction of the database will store the data gathered from the peripherals and the interaction with the users.

Figure 3.5 represents the table corresponding to the storage of information obtained from SharePoint.

As previously discussed, this information is obtained from the application already in use (SIL) for placing requests for testing the parts.

This SharePoint will populate the database table with information about the part or parts under test, such as who is responsible (P_cuet, P_equipa), the type of the part, quantity, if it is machined or not, and the supplier of the material (P_peca, P_NumPeca, P_quant, P_orgao, P_Fornecedor, P_molde), the location of the machine of which the part is correlated and if the machine is stopped (departamento, P_uet, P_oper, P_maquina,

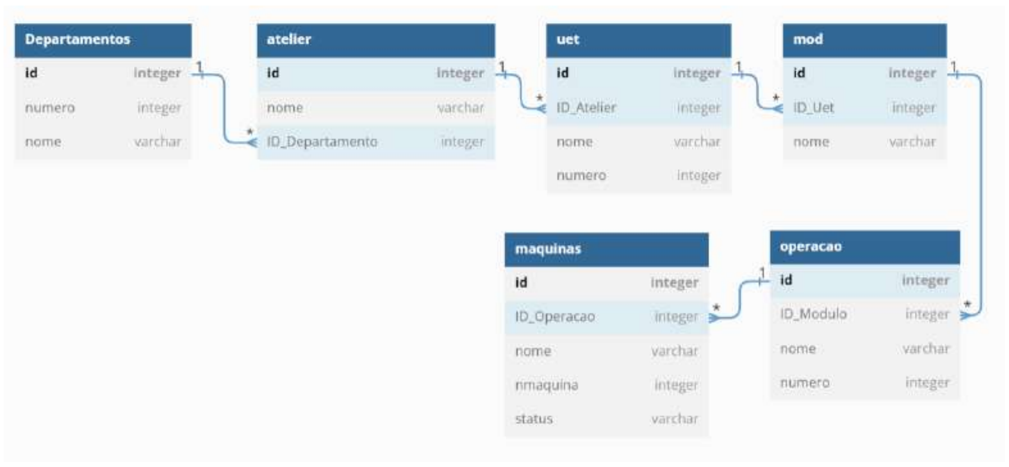


Figure 3.4: Machines Hierarchy Database

P_parada), and finally the information regarding the test (P_tipocontrolo, P_controlo, P_result, Estado, Created, ID_pedido).

The next table, depicted in figure 3.6, will store information about the monitoring of the parts out-of-flow.

It will register the tag number attached to the part(*"etiqueta"*), the machine that made the part(*"ID_Maquina"*), the destination where the part was sent(*"Destino"*), and the actual location of the part(*"Localizacao"*), associating this with the test request identification from the table "sharepoint" presented previously (*"id_Pedido"*).

The relationship of this table with the table "maquinas" will be one-to-many because one machine can have multiple requests, and the relationship to the "sharepoint" table will be one-to-one since for every test must be done a request.

All these relationships and tables must culminate in a database as represented in figure 3.7.

3.3.2 Server

A server is a computer or system that offers resources, services, or capabilities to clients via a network. It mainly works as a REST API (Representational State Transfer Application Programming Interface) and awaits the request from a client.

It can be real hardware or virtualized instances on cloud-based infrastructure it can be dedicated to serving only a client or shared to serve multiple clients, Web servers, file servers, database servers, and application servers are examples of common types.

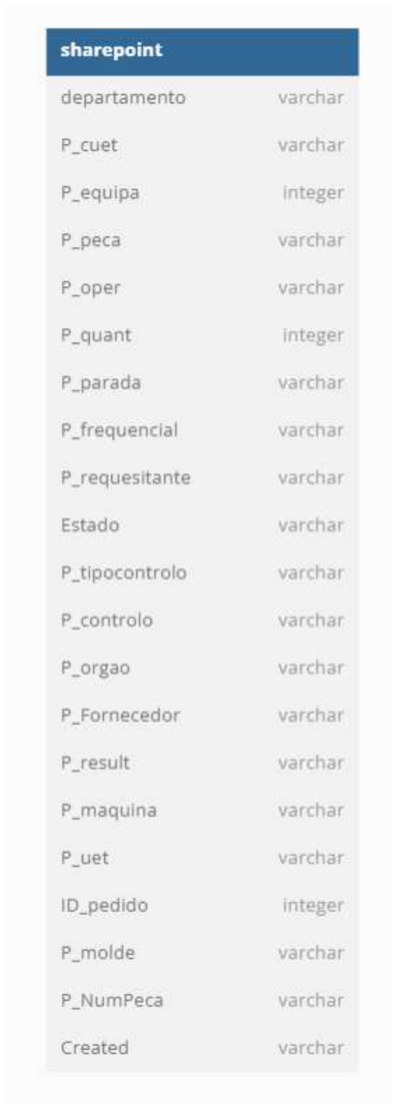
Servers provide network communication, data storage, and resource sharing.

They are key components of modern computing infrastructure since they are utilized in a variety of industries and applications.

In this solution will hold and run the scripts and the database.

3.3.3 User interface

A user interface is how a user interacts with a software application, website, or other digital system, allowing the user to receive information and provide feedback to the system.



sharepoint	
departamento	varchar
P_cuet	varchar
P_equipa	integer
P_peca	varchar
P_oper	varchar
P_quant	integer
P_parada	varchar
P_frequencial	varchar
P_requesitante	varchar
Estado	varchar
P_tipocontrolo	varchar
P_controlo	varchar
P_orgao	varchar
P_Fornecedor	varchar
P_result	varchar
P_maquina	varchar
P_uet	varchar
ID_pedido	integer
P_molde	varchar
P_NumPeca	varchar
Created	varchar

Figure 3.5: Table "sharepoint" from Database

A user interface should be intuitive, user-friendly, and efficient in order to facilitate the user's duties and goals.

This solution must be easy to use from the user's perspective and not add additional labor, otherwise, the users will not use it correctly or not use it at all.

It must contain all the info regarding the time and place of every part that goes out-of-flow and present in real-time the status of the machines, in order to improve the reactive response to irregularities.

The interface will be separated into three sections, the first of which will show all of the machines and their status, as well as the ability to perform activities such as requests and reintroduction.

The second will be where technicians from the control centers can examine and edit data from each request.

Finally, a mobile barcode scanner that can scan any part attached to a tag anywhere

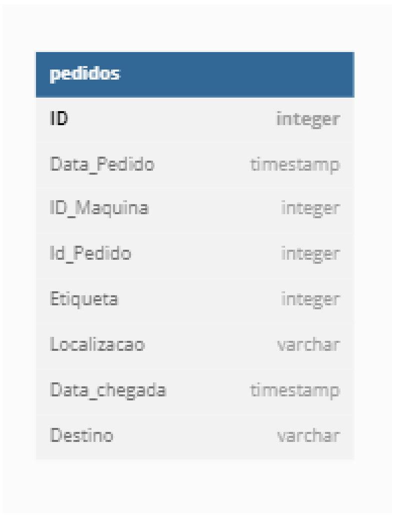


Figure 3.6: Table "pedidos" from Database

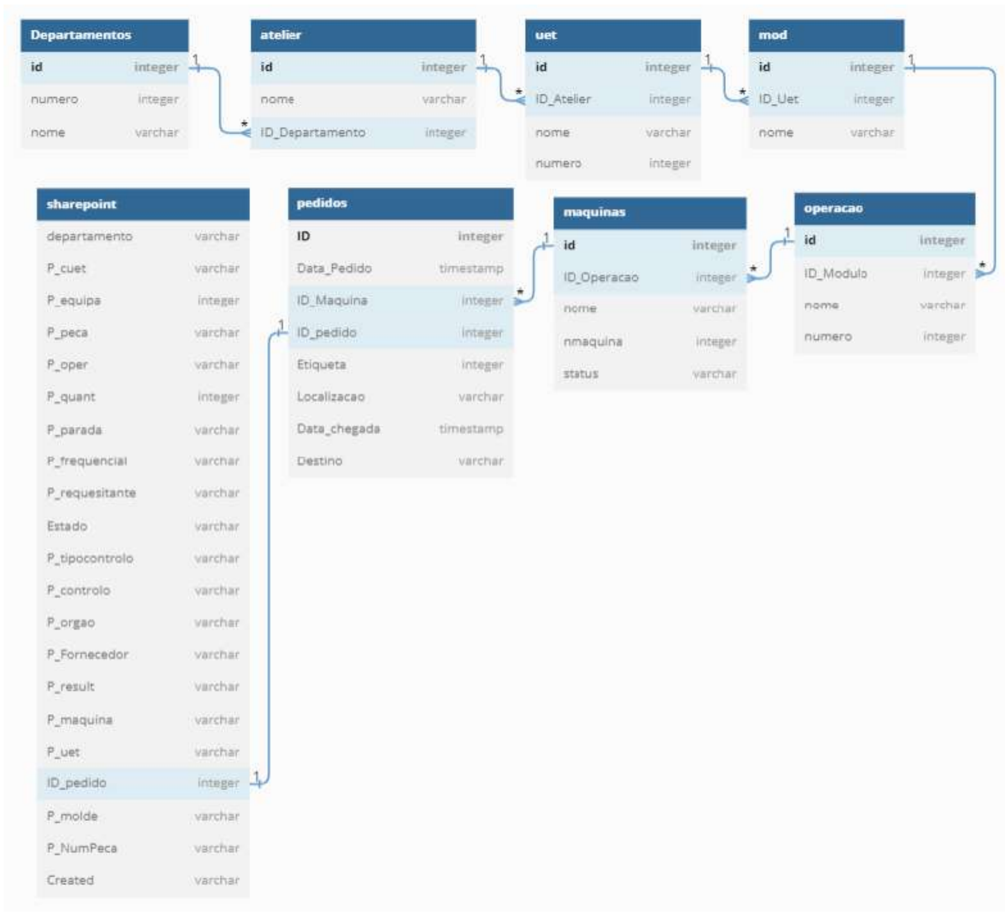


Figure 3.7: Database normalization definition

in the factory.

3.3.4 Docker

To develop, construct, and execute software in virtual environments, we will utilize Docker, which uses containerization technology to create and manage isolated environments that package an application and its dependencies.

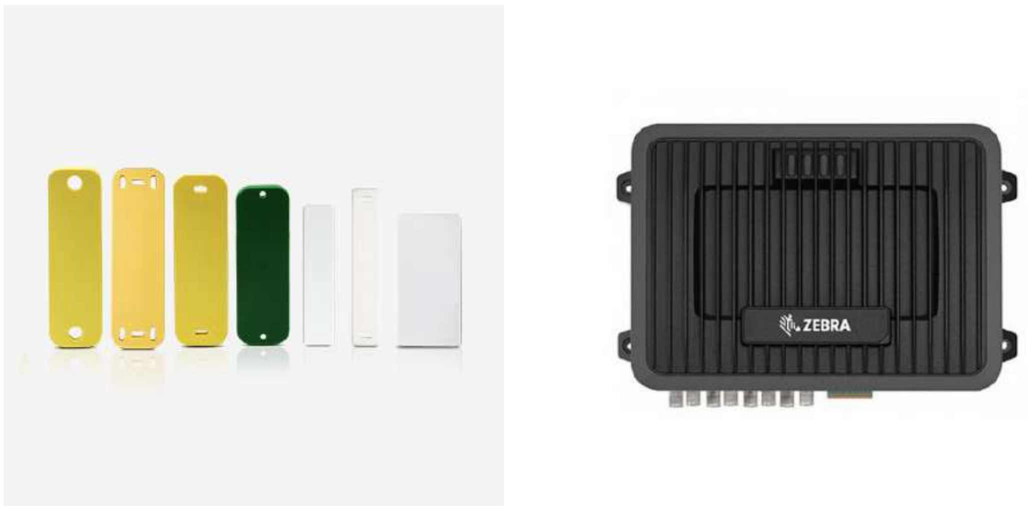
Docker creation requires both a Dockerfile and a Dockercompose file. A Dockerfile contains instructions for constructing a Docker image. Dockercompose is a Docker tool that enables services to be managed and interacted with.

This approach promotes faster code production by shortening the time between developing and executing apps.

Docker will be required in this solution to create an image for the application and allow it to be deployed in the Renault Cacia server to operate with real data.

3.3.5 Radio Frequency Identification

RFID (Radio-Frequency Identification) is a wireless identification and tracking system that uses radio waves. It is made up of two parts: a tag (figure 3.8a), which is a small device with a unique identity, and a reader (figure 3.8b), which sends and receives radio signals to connect with the tags.



(a) Diverse RFID tags [9] (b) RFID reader [47]

Figure 3.8: RFID Technology components

RFID technology is widely utilized in a variety of applications, including inventory management, supply chain tracking, and access control.

RFID provides benefits like non-line-of-sight operation, quick data capture, and durability, making it a popular choice for automatic identification and tracking in a variety of industries.

In this solution it will be used as a tracking solution, to more accurately place the location of a specific part or set of parts.

3.3.6 Barcode technology

Barcode technology has been used for decades and is a well-established technology that has been adopted in a variety of sectors to increase operational efficiency and accuracy.

Taking into consideration the client's requests and recommendations, the barcode reader is included in this solution to avoid adding more work to the user, since it is a technology that is currently in use across the plant and the operator is already familiar with its use.

Intentionally blank page.

Chapter 4

Solution Implementation

With the proposed solution defined, it was time to start developing it so that it could be implemented at the Renault Cacia factory.

This chapter comprises the system architecture, software, hardware, and operating procedure. Not only will be presented how it was developed but also how it was implemented.

Based on the architecture presented in the proposed solution chapter (3), the architecture of the solution implemented is depicted in figure 4.1. The purpose of this figure is to illustrate the location of the components and the communication protocols.

The components of this implementation can be divided into two different areas. The shop floor area that comprises the hardware, namely the RFID reader and the barcode reader (numbers 1 and 2 in the figure 4.1), and the remote server area that comprises the software, this software aggregated inside of a docker that comprises the Flask backend, the Angular frontend and the python scripts (numbers 3, 4 and 5 in figure 4.1).

The information flow starts from the reintroduction point area where the information is introduced in the user interface manually or using the barcode reader by USB keyboard communication, then the data is sent to the Flask server, simultaneously the Flask also receives the data from the SharePoint, and the RFID reader all these via TCP/IP(Transmission Control Protocol/Internet Protocol) communication, after that the Flask sends the info via a query to the database.

More detailed information about the components aforementioned will be presented in the sections ahead.

4.1 Hardware

This section will present the components chosen for solution implementation as well as the detailed reasons for that choice.

4.1.1 RFID Bluebox CX-AVI 5526U-c-opt1-rrr industrial UHF

The selection of the best RFID hardware to implement in this solution was challenging. To accomplish the solution proposed, two different products were kindly borrowed from Accurex an RFID material supplier of Renault Cacia.

The products were a Zebra FX9600 and a Bluebox CX-AVI 5526U-c-opt1-rrr industrial, both ultra-high frequency readers. Aiming to test both pieces of equipment in an

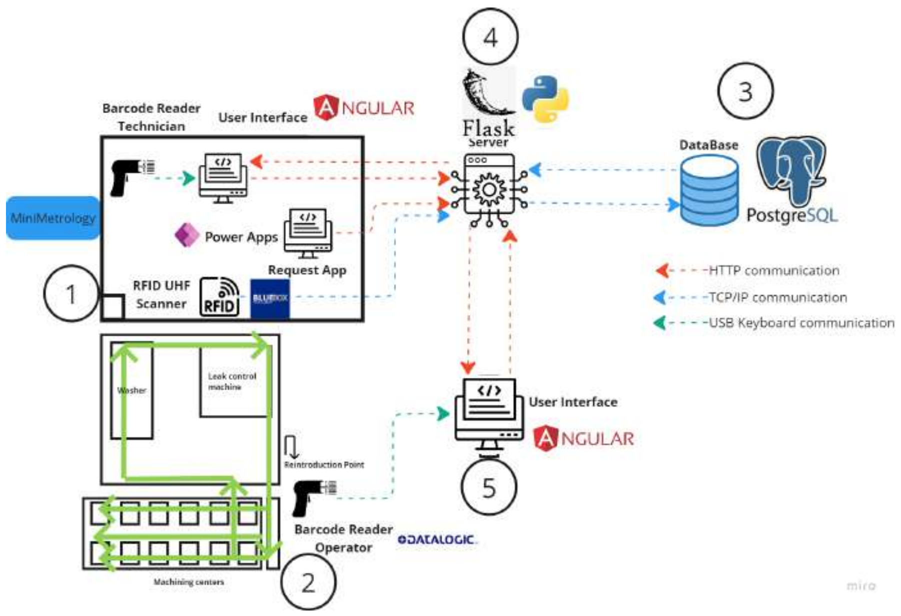


Figure 4.1: Implemented solution architecture: (1) 4.1.1; (2) 4.1.3; (3) 4.2.1; (4) 4.2.2; (5) 4.2.4

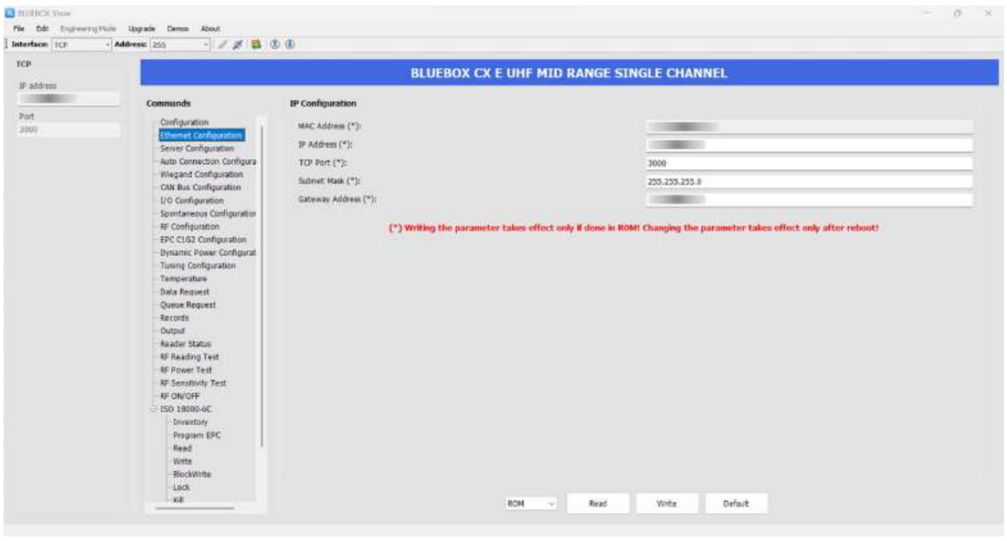
industrial environment, was developed a micro-project that can be read about in the appendix B. The conclusions gathered from that experience were that the most suitable equipment would be the reader from Bluebox since it is more robust and built for an industrial environment.

The Bluebox CX-AVI 5526U-c-opt1-rrr industrial, represented in figure 4.2, is a UHF read/write RFID equipment, that has incorporated in the same device the reader and the antenna, making it ideal for industrial applications. It connects with a 'host' system (usually a Computer or a PLC) through an RS232 / RS485 serial line, an Ethernet connection up to 100 meters, Data bus and Can bus depending on the equipment version.

Figure 4.2: Bluebox CX-AVI 5526U-c-opt1-rrr industrial [8]

The Bluebox serves as a connection point, enabling communication between the host system and one or multiple RFID transponders (or tags) located in close proximity to the antenna. Regardless of the kind of connection (point-to-point, multi-point, or Ethernet), the same 'master/slave' protocol is used for communication between the host system ('master') and the Bluebox ('slave') [8].

It is also possible to modify the functional settings and update the firmware over these communication channels. The 'BLUEBOX Show' is a software that is included with the SDK (Software Development Kit) and is intended to explain how these operations may be carried out [48]. Figure 4.3 shows the Bluebox page where it is possible to define the "IP" the "port" and other parameters that help configure the connection with the reader and the computer or remote server.



Interface: TCP Address: 255

TCP
IP address: [redacted]
Port: 3000

BLUEBOX CX E UHF MID RANGE SINGLE CHANNEL

IP Configuration
MAC Address (*): [redacted]
IP Address (*): [redacted]
TCP Port (*): 3000
Subnet Mask (*): 255.255.255.0
Gateway Address (*): [redacted]

(*) Writing the parameter takes effect only if done in ROM! Changing the parameter takes effect only after reboot!

ROM

Figure 4.3: Bluebox Show Ethernet communication configuration menu

In addition to this, the Bluebox has the capacity to manage two digital outputs (relays) as well as two digital inputs that are opto isolated. Without having to open the device, professionals in the fields of installation and maintenance are able to make all of the necessary connections for the device's power supply, communication, and interface I/O (Input/Output) ports thanks to the design and development of Bluebox. The connections available in this device are detailed in figure 4.4.

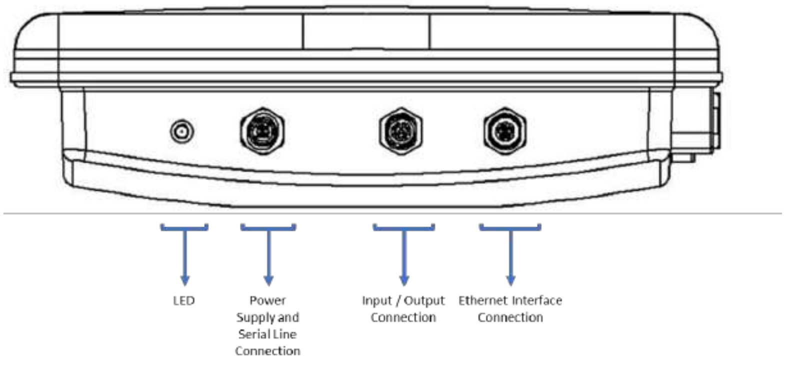

LED
Power Supply and Serial Line Connection
Input / Output Connection
Ethernet Interface Connection

Figure 4.4: Bluebox connections [8]

Despite the fact that this is one of the most complete RFID devices available in the market, just the most important aspects were covered in table 4.1, which is what ultimately led to the selection of this particular piece of equipment.

Table 4.1: Bluebox Reader specifications

Parameter	**Value**
Operating Frequency	865 MHz – 868 MHz
RF Transmit Power Max	0.5W (27dBm) conducted
RF Receive Sensitivity Max	-85dBm
Antenna	Integrated
Reading Distance	10 m
Supported Transponders	ISO 18000-63 (EPC Class-1 Gen-2 V2)
Communication Interface	Serial RS232 / RS485
Ethernet	10-100M
Protection Class	IP65
Dimensions	308 x 308 x 85 mm

The communication utilized and the message format will be addressed in 4.2.2.

4.1.2 Tags RFID SlimFlex 200

The SlimFlex 200 RFID tags (figure 4.5) from HID are innovative, flexible tags designed for a wide range of applications. These tags feature housings made of thermoplastic polyurethane, this housing enables the tags to withstand repeated bending or torsion without compromising their performance and ensures the protection of the embedded electronics, even in harsh conditions felt in an industrial environment [9].

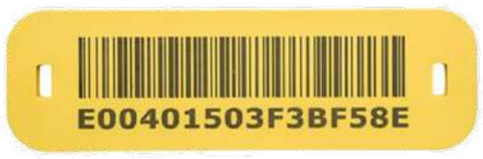


Figure 4.5: SlimFlex 200 RFID Tag with barcode [9]

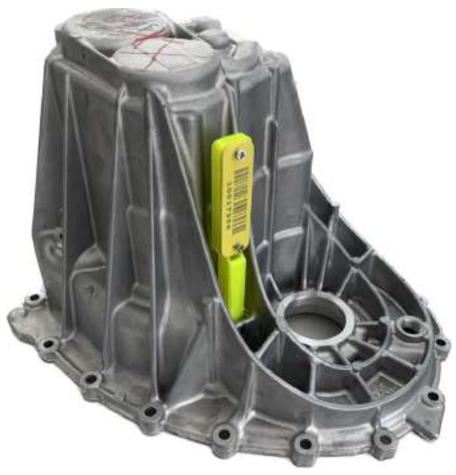
These passive contactless transponders significantly improve the speed and precision of data gathering. These tags provide exact and reliable reading and updating of user memory they possess thanks to anti-collision capabilities, quick data rate communication, and password data security. SlimFlex tags are compatible with ISO-compliant readers and modules, ensuring interoperability. The characteristics of these tags are presented in the table 4.2.

Table 4.2: SlimFlex 200 RFID Tag specifications

Parameter	**Value**
Operating Frequency	860–960 MHz
Memory	128 bit
Reading Distance	6 m
Standards	UHF EPC Class 1 Gen 2, ISO 18000-6C, RAIN
Options	Laser engraving
Protection Class	IP68
Dimensions	83 × 25 × 3 mm

Another benefit is that a one-dimensional barcode may be engraved on them using a laser. This barcode, which is of the 'code 128' type by default, makes it possible for a component that it is connected to be identified in a more straightforward manner.

Following the selection of the Tag, it was necessary to determine the manner in which the tag would be attached to the component. The client was asked for their feedback in order to determine which of the several designs for the tag support would be the most effective. Keeping all of these considerations in mind, the end result was the design shown in figure 4.6.


(a) Placement of the tag in the part

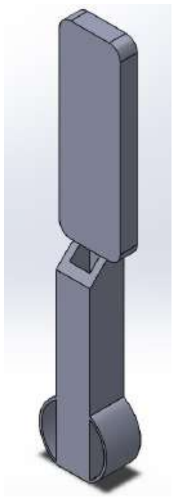

(b) Tag computer-aided design

Figure 4.6: Tag support and assembly to the part

4.1.3 Barcode Scanner - PowerScan PBT9501

Due to the fact that there is already a PowerScan PBT9501 barcode reader (figure 4.7) in the location where this project will be implemented, the decision regarding which barcode reader to use was significantly simplified. This was done with the goal in mind of not weighing down the operator with additional work and with the vision of making use of the resources that have already been implemented in the factory.

Figure 4.7: PowerScan PBT9501 barcode reader

The Powerscan PBT9501 is a versatile handheld barcode reader developed by Datalogic. It is equipped with a scanning technology that enables it to read both 1D and 2D barcodes, regardless of how well they were printed or how badly they were damaged. The scanner is equipped with Bluetooth connection, which makes it possible to communicate via wireless with a variety of devices.

Although it is a wireless reader, it connects first with its charging base, then the charging base is connected to the computer. The communication between the base and the computer can follow protocols such as RS-232, and USB [49].

Some of the features of this device are shown in the table 4.3.

The type of communication selected to implement in this solution was the type of USB Keyboard, in order for the program to be versatile and usable even if the reader is out of order.

4.2 Software

After the selection of the hardware needed to develop the solution to implement, it was time to develop the software to merge all the information and put it at the user's disposal. All of the information related to the development of software, as well as the communications and usage of the program, will be covered in this section.

4.2.1 Database: PostgreSQL

PostgreSQL is a robust relational database management system (RDBMS) for storing and managing structured data, that provides communication between the storage data

Table 4.3: Specifications of the Powerscan PBT9501 Barcode Reader

Specification	**Description**
1D / Linear Codes	Auto-discriminates all standard 1D codes including GS1 DataBar linear codes.
Particulate and Water Sealing	IP65
Reading Ranges	6 to 85 cm
Radio Frequency	2.40 to 2.48 GHz
Bluetooth Wireless Technology	Max. 7 readers per radio receiver
Interfaces	Keyboard Wedge RS-232; USB; USB COM; USB HID Keyboard; Optional Ethernet (Standard, Industrial)

and the user or application. In this solution, the Flask (server-side) in order to access the data storage makes a request to this DBMS.

Using pgAdmin4, an open-source administrative tool for PostgreSQL that allows the creation and management of databases, based on SQL commands the database itself was developed, being included in this all the attributes already presented in section 3.3.1.

During this creation, it was defined which were the primary keys as well as the foreign keys also already depicted in figure 3.7. Alongside that, the definition of the type of variable associated with every attribute was also confirmed and established.

4.2.2 Flask

The framework adopted for the development of the back-end service of this project implemented solution was Flask, as aforementioned this framework is written in Python language and in this project is used as a REST API (Representational State Transfer Application Programming Interface).

The Flask is intended to be the bridge between all the system peripherals, the data collector, and the supplier. Flask receives requests from the user interface and other Python scripts, which can be GET, POST, DELETE, or PUT, and gathers and processes the requests after that retrieves the data, this data can come from the user interface, the database, the RFID script, or SharePoint script.

In order to show the usefulness of Flask in the context of this project, one of the procedures that Flask is responsible for carrying out will be described, after which all the other procedures will be identical to the same foundation.

The procedure is responsible for passing the values for the state of the machines, breaking it down, the Flask receives a GET request from the user interface every 10 seconds, these request passes an array that contains the values of the "Departamento", "Atelier", "UET" and "Modulo", selected and that are being displayed in the screen.

With this information the Flask executes a query to the database, more specifically to the table of the data gathered from the SharePoint, with the objective to track this machines number, and retrieve the state of the test and in case of concluded, retrieve the result.

With the information gathered the Flask executes a function that compares if the test has already ended and what was the result, depending on the result the Flask does a query to the database to another table in order to register the the history of the test results associated to the machines.

Finally, Flask send the response to the user interface with the machine number and the state of the machine.

The process described above is shown in figure 4.8 in a more visual way.

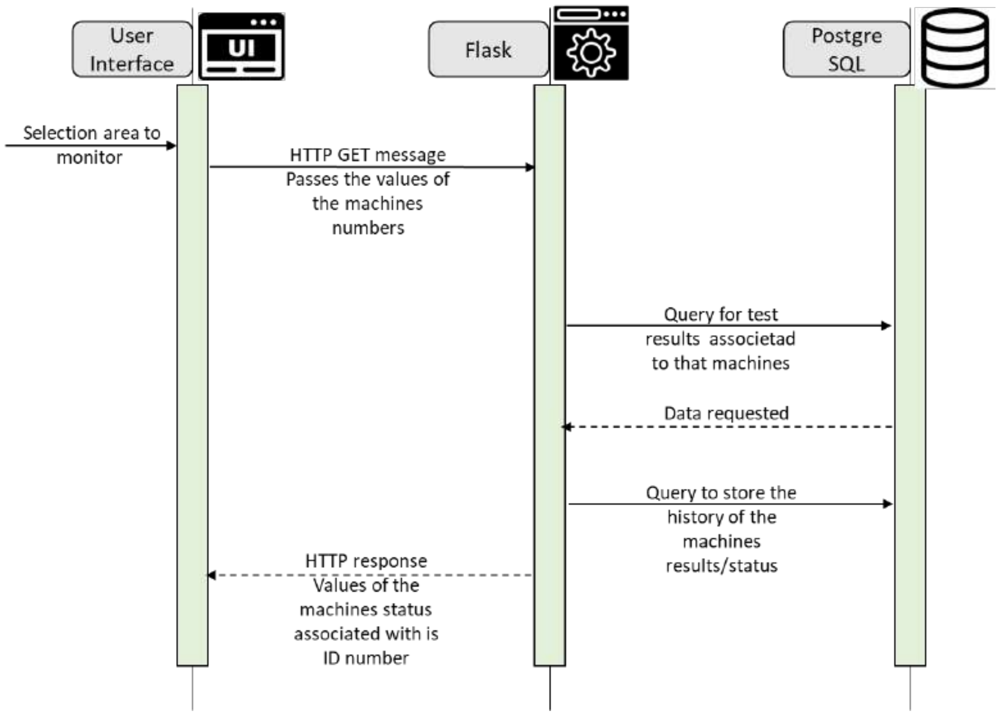

User Interface
UI
Flask
Postgre SQL
Selection area to monitor
HTTP GET message Passes the values of the machines numbers
Query for test results associetad to that machines
Data requested
Query to store the history of the machines results/status
HTTP response Values of the machines status associated with is ID number

Figure 4.8: UML(Unified Modeling Language) representation of communication of Flask component of this project

RFID Script

This topic covers the RFID script, a Python script that displays the process of receiving RFID tag data using a Bluebox RFID reader and then communicating that data to a Flask back-end for additional processing. A sample of this script can be seen in the appendix C.

The operating method described in this script begins with the definition of a variable and an array. The purpose of the first is to record the current status of the connection with the hardware, and the second is to save the current number/numbers of the tag/tags.

After that, the connection with the reader had to be set up. Since TCP/IP is the protocol for the contact between the reader and the script, the IP(Internet Protocol) address of the reader, which is called the 'host', and the 'port' from which it sends messages must be defined. Both the IP and the port were set in the program 'Bluebox show', which was already mentioned, which lets all of the reader's features be set up. This IP was also set up with the fact in mind that this hardware will be linked to the factory intranet network and because of this, the IP had to be one that was not used by any other equipment and was static, so it wouldn't change even if the hardware was removed.

With the parameters of the connection defined, the code enters in a loop. In this loop it tries to connect to the reader and if it is successful it changes the state to connected and breaks the loop.

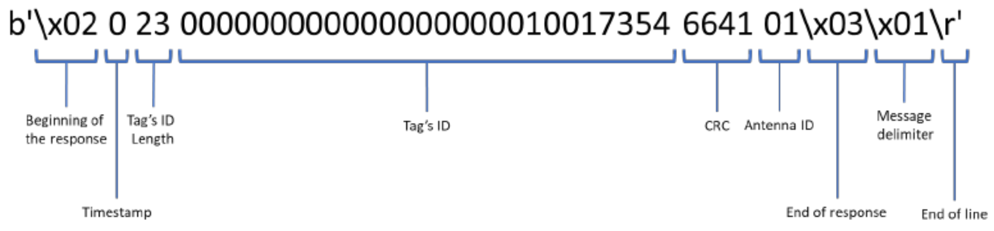
Soon after the established connection the script sends a message to the reader asking for the information regarding the tags that the reader is reading.

The message sent by the script is based in a set of commands, those commands can differ, and based on the command sent, the response from the reader is also different. Some of the most commonly used commands are presented in table 4.4.

Table 4.4: Commonly Used Commands for Bluebox CX-AVI 5526U-c-opt1-rrr

Command	**Description**
Read Tag ID	`b'\x01\x02\x00\x00\x03'`
Read Tag Data	`b'\x01\x03\x00\x04\xA1\xA2\xA3\xA4\x03'`
Set Communication Parameters	`b'\x01\x05\x00\x06\x01\x02\x03\x04\x03'`
Configure Antenna Settings	`b'\x01\x06\x00\x05\x01\x02\x03\x03'`
Start Tag Reading	`b'\x01\x07\x00\x02\x03'`

In this specific case the command sent by the script is the 'Read Tag ID' command, this command retrieves a massage as depicted in figure 4.9. As can be seen, the message is quite extensive, for that reason the next step is to process the message to only get the Tag ID number.



```
b'\x02 0 23 000000000000000000010017354 6641 01\x03\x01\r'
```

Figure 4.9: Message sent by the Bluebox reader

Finally, the script makes a POST request, via HTTP, to the Flask and passes the Tag ID (Identification) identified, then the Flask compares that ID with the IDs in the database, and fills the column of the location with the location name associated with that reader. A more graphic view of this code can e seen in figure 4.10.

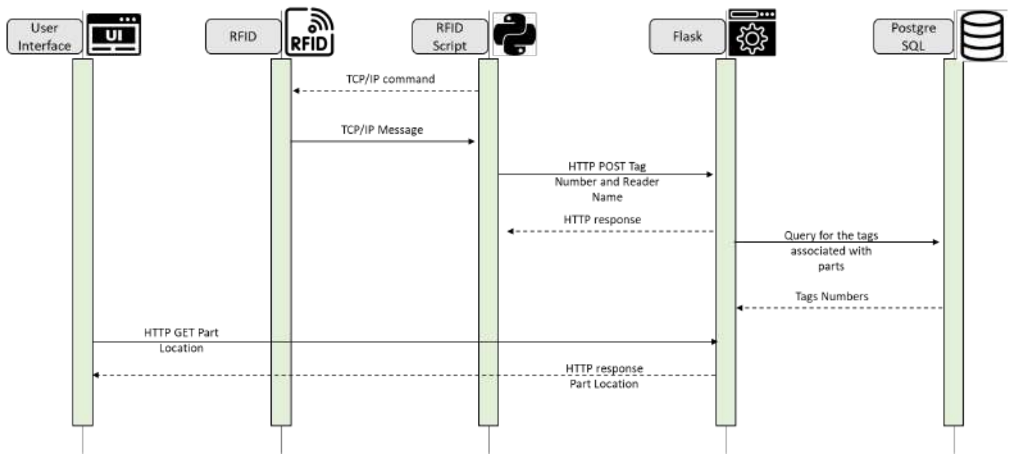


Figure 4.10: UML representation of communication of RFID components of this project

SharePoint Script

This topic covers the SharePoint script, a Python script that is responsible for acquiring data from SharePoint that is connected to the quality test request application (SIL) and then communicating that data to a Flask back-end for additional processing. A sample of this script can be seen in the appendix C.

The process within this script starts with a connection to SharePoint, the script proceeds with the definition of proxies, which are intermediary servers used to route HTTP requests. In this case, the script specifies proxies for both the 'HTTP' and 'HTTPS'(Hypertext Transfer Protocol Secure) protocols.

To get an access token for authentication reasons, the script builds a data package. The package contains the relevant details, such as the grant type, resource, client ID, and client secret. This data packet will be sent in a later request to get the access token.

The script also defines the headers for the requests. These headers specify the content type and will be used to authenticate subsequent requests to SharePoint.

To get an access token, the next step is to send a POST request to a specific URL(Uniform Resource Locators). The data body and headers are both part of the request. When the script gets the answer, it reads the JSON(JavaScript Object Notation) data and pulls out the access token. This token is needed for authentication in future calls.

After the script gets the access token, it changes the headers so that the access token is included. This makes sure that future calls to SharePoint are properly authenticated.

The script then sends a GET call to a SharePoint URL to get information from it. The URL tells SharePoint which list to use and includes query options that let it sort and narrow the results. This request includes the new headers, which have the access token in them.

When the script gets the answer, it reads the JSON data and pulls out the areas of interest. Then, these fields are put into their own variables so that they can be used in other ways.

Finally, the script also sets up a function that makes a POST request to the Flask passing the values retrieved from SharePoint. These values are sent in the request body as a JSON file. Figure 4.11 presents a more graphic perception of the presented script functionality.

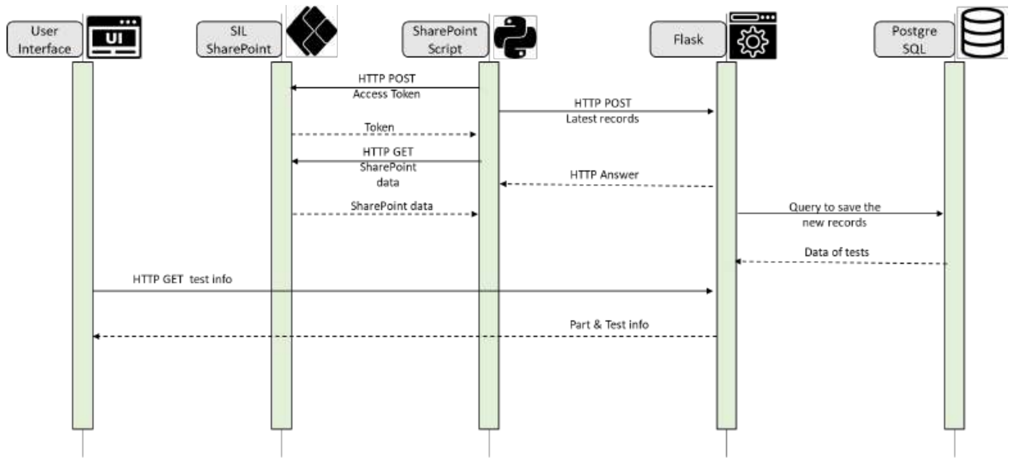


Figure 4.11: UML representation of communication of SharePoint components of this project

4.2.3 Docker

For Docker usage, it was created two files a docker-compose and a docker file which allows the remote server, where it is implemented, to install all the dependencies needed for the scripts to run properly and run the applications without interfering with other applications in the same machine.

In this case, a bash file was also created to allow the simultaneous running of all Python scripts. This improves the whole system's efficiency and efficacy.

4.2.4 User Interface - Angular

As already mentioned it was necessary the creation of an interface so that the end user could interact with the system, for that end the framework selected to develop such UI was Angular, programmable through TypeScript language a variation of JavaScript, HTML and CSS these last two languages are the ones that give the looks to the page.

Although TypeScript had never been thought in this Mechanical Engineering course, JavaScript was addressed in a subject. This framework allows the creation of interactive interfaces, with a large variety of functionalities, most of which belong to the Angular material, which is a UI component library that provides a set of pre-built and ready-to-use UI components.

In this project's user interface development, the main goal was to create a window that allowed the user to check the operational state of the machine and follow the trajectory of the part during the quality test event (figure 4.12a). Later it was asked by the end user to add a second page, this page would serve as a search engine, once the tag number was inserted in the search field the interface would show the data referent to that tag/part (figure 4.12b). Both these pages are depicted in the figure 4.12.

Despite the fact that the appendix A has a thorough explanation of the front-end, which has previously been presented, the most essential menus will be discussed as topics in order to provide a better comprehension of the overall structure.

- **First page of User Interface (figure 4.12a)**

  1. **Area selection menu:** The user has the flexibility to choose whatever part of the factory, in terms of either the sector or the module, he wishes to engage with by using this menu. As was said before, it was projected that this application would be added to each and every module in the factory. This option makes it possible to make that pick.
  2. **Machine status and Tag association:** This section of the page shows the status of the machines regarding the quality test result. As can be seen, there are multiple red and grey areas, these areas represent the operations made in that module (i.e.: OP110/OP120 - represent the machining center's operations) and the little rectangles inside them are the machines, identified by their identity number. The status of the machines is identified by colors:

     Blue: The part has already left the production line and is in transit to the quality control center.

     Orange: The part has arrived to the quality control center and is waiting to be tested.

     Red: The part failed the quality test, the machine must be stopped and fixed.

     Yellow: The part passed the quality test with an 80% score, some parameters of the machine must be corrected, but it can still work.

     Green: The part passed the quality test with 100% score.
  3. **Sidenav menu:** In this menu there are multiple features that can be performed:

     Home: Redirects the user to the home page where he can select the first or the search page.

     "Registos": Opens a dialog with the history of all the parts that have been sent to the quality control center and all the information associated with that part.

     "Pedidos": Opens a dialog and depicts a table that shows the association of the tags with the machines, the destination of that part, and the real location of it.

     "Reintrodução": This function is what brings an out-of-flow part back into circulation. The user inputs the number of the tag connected with the component and the operation in which he intends to place the part. If all of the information matches, the function verifies that the part has been successfully reintroduced, if not, it triggers an alert.

     "Adicionar": This choice gives the user the ability to add a new machine to the module, and it is designed to be used quite sporadic. This is meant to be used in situations in which an older machine is upgraded to a newer one or the module itself gets reorganized.

     Delete: Has the same goal as "Adicionar", except for removing the machine that has to be altered.
  4. **Out-of-flow parts:**This feature was developed in response to a request from the user to make it easier to detect whether a component is still out-of-flow

and to assist the user in gaining an overall understanding of the number of parts that are removed from the production line for the purpose of quality testing.

- **Search page of User Interface (figure 4.12b)**

  Home button: Redirects the user to the home page where he can select the first or the search page.

  "Codigo de Barras": Input box for the user to scan or manually input the tag number. This launches a dialog in which all of the information pertaining to that particular tag/part is shown in its entirety.

4.3 New out-of-flow parts process

In this section will be described the process of sending a part to the quality test center, with this description is intended to deliver the big picture of the new process compared with the initial one.

Firstly the user picks a part that needs to be tested, then in the user interface selects the machine corresponding to that part and associates a tag to that machine, following that the user attaches the tag to the parts using the supports designed for the effect. Meanwhile, the information is passed to the Flask backend and stored in the database, and in the user interface, the machine affected changes its color to blue with the information that the part left the production line and is in transit to the quality control center.

In figure 4.13 is possible to see the area where the operator stands, and initiates the process. As can be seen in the figure the screen has the application open so the operator can place the association of the tag to the part, by using the barcode scanner placed in the table also visible in the figure.

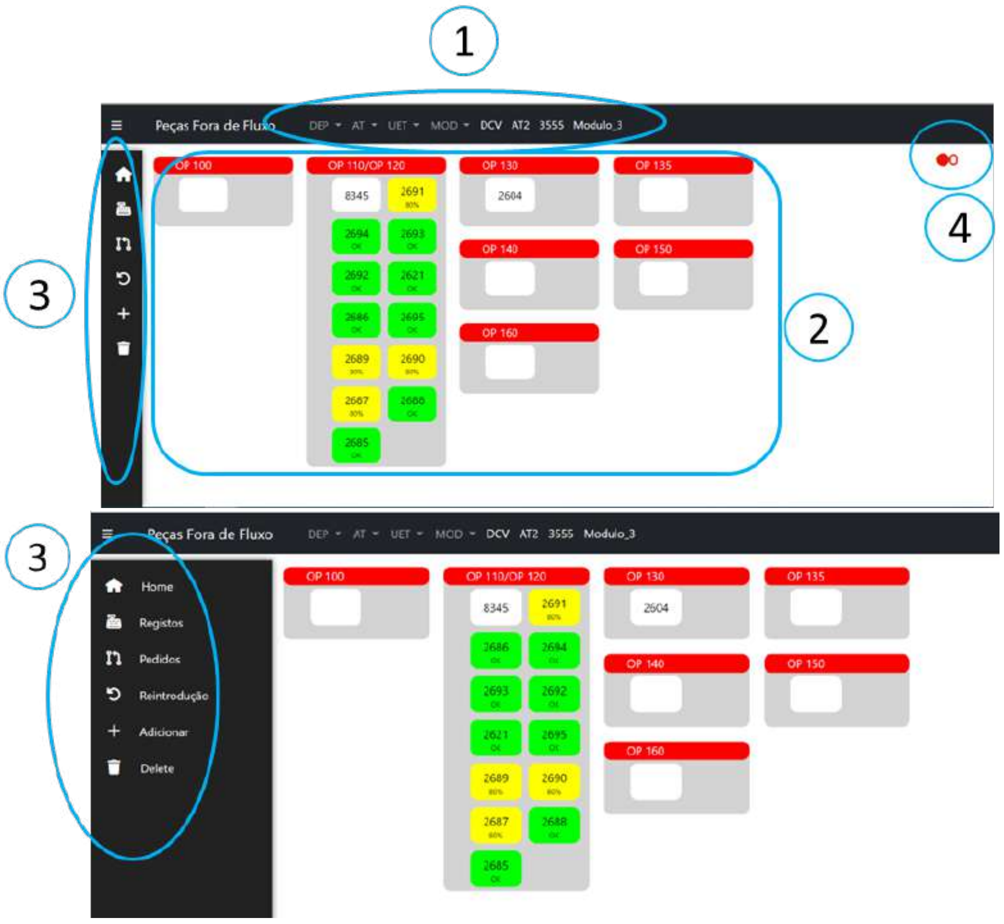
Next, the user takes the part to the quality test center. when it enters that area that is covered by the RFID reader the script sends a HTTP POST to the Flask saying that that specific part is already in the quality control center. That information is aggregated to the table modified before. And the user interface changes the information regarding the location of the part to where it actually is.

The RFID reader is located at the entrance to the mini metrology center, underneath the computer where the user submits the test request and facing the door, as can be seen in figure 4.14. This location was chosen with the intention of reading the components as soon as they entered the control center.

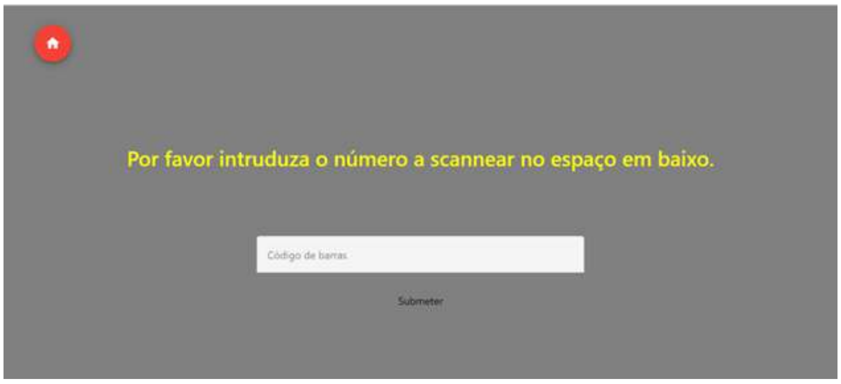
Following that the user places the test request trough the application SIL already implemented, before the beginning of this project. This triggers an HTTP POST to the Flask with the information about the parts that are waiting to be tested. With this input, the user interface changes the color of the machine to orange meaning that the part is being tested and is awaiting results.

In order to initiate the test the technician needs to read the tag number, for that he uses the other page of the user interface and this will do an HTTP GET request to gather the information needed so he can proceed.

The next step occurs when the test is complete, this triggers a refresh of the database with the new information from the SharePoint connected to SIL. Depending on the state



(a) First page of User Interface



(b) Search page of User Interface

Figure 4.12: User Interface interactive pages

of the test the machine changes its color to red, yellow, or green. Giving the user the information if the machine is OK or not to keep working.

Finally, the user grabs the part again and in order to reintroduce the part in the production line it has a menu in the user interface where he must enter the tag number manually or by scan, followed by the number of the operation it is being reintroduced in.

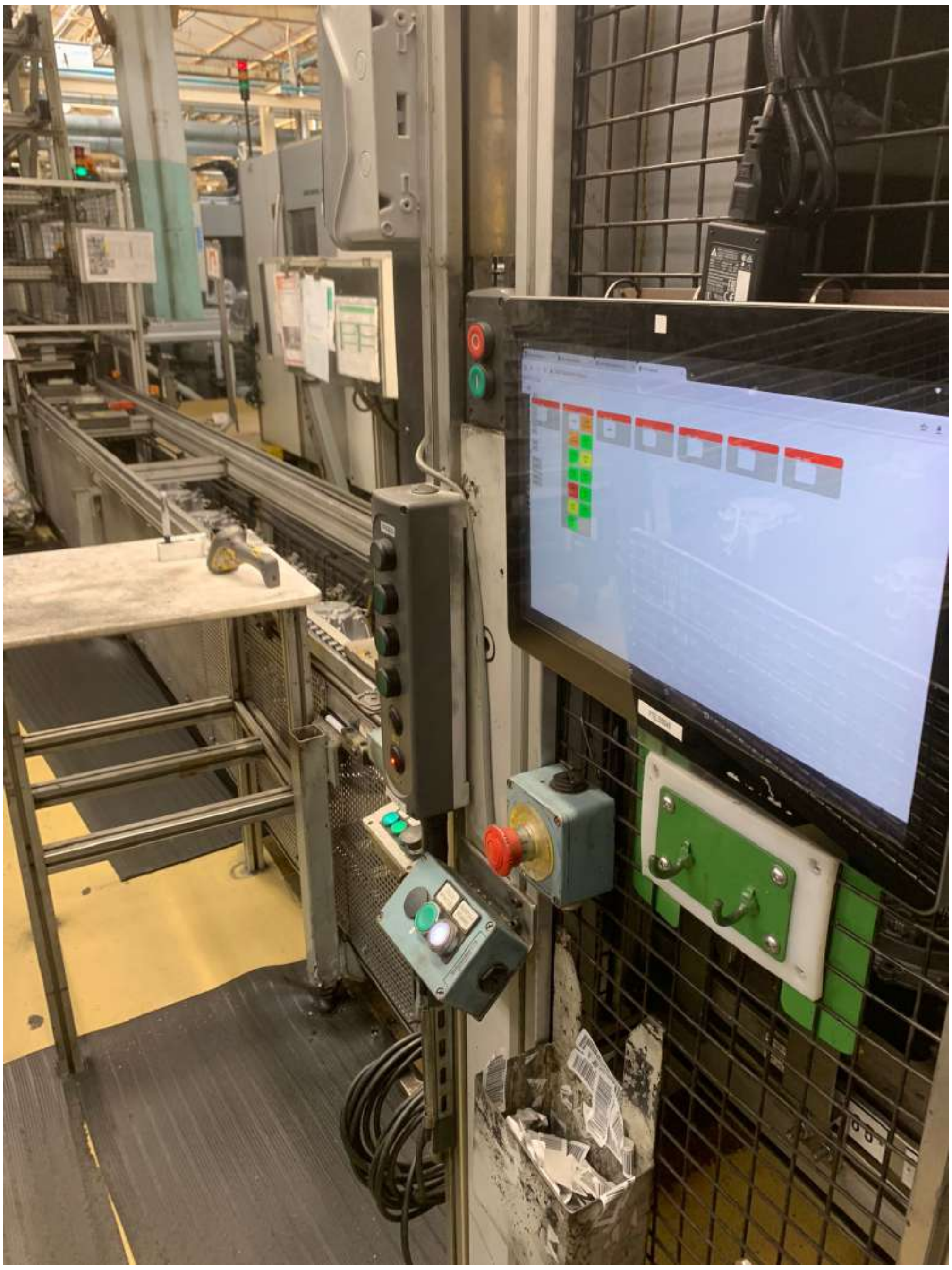


Figure 4.13: Operator work area with the system implemented

This triggers an HTTP POST to the Flask, that then compares the operation numbers. Depending on whether the operation number introduced matches the operation number that the part should be reintroduced the user interface will show a message about part well reintroduced or not.

This last step closes the cycle of a part out-of-flow.

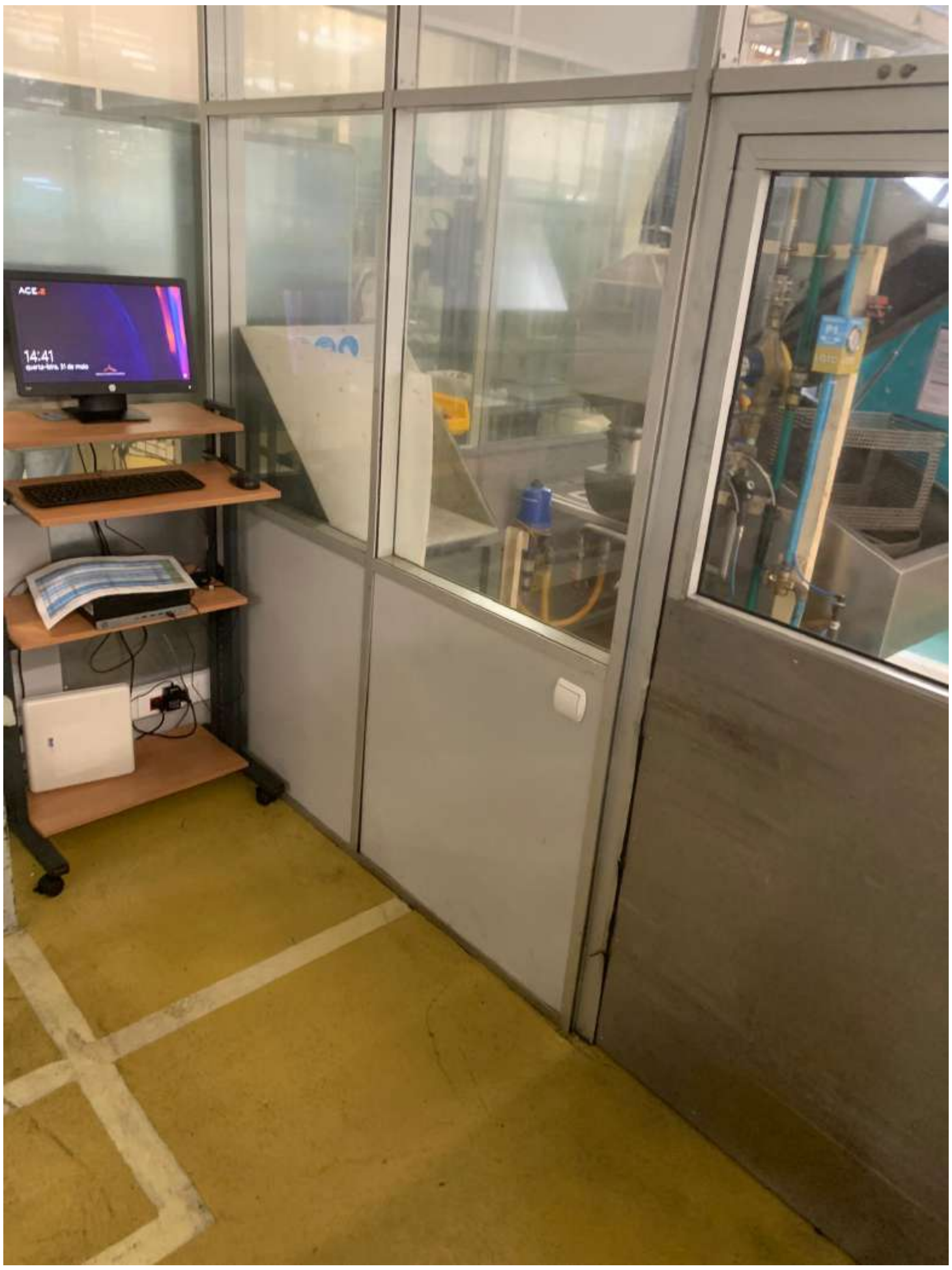


Figure 4.14: Mini metrology area with the RFID system implemented

Chapter 5

Conclusion

5.1 Performance and Results

The fact that this project was developed based on an Agile methodology, allowed achieve the result that the client expected, with all the features needed to facilitate his job and improve the results, without adding to the application functionalities that are not important for the end user.

After the implementation of this project on the factory floor, the feedback gathered from the technicians of the quality control center as well as from the operators that use them on the borderline was that the application was well conceived and that it answers to their needs and reduces the time of writing and cleaning the tags, and the fact that they now have the information always available in any device allows for more reactive response in case of abnormality, a process that was more time consuming before because they had to consult their email to know this information.

Regarding the benefits that this project delivered to the factory, table 5.1 compares the time dispensed for the process before the implementation and after the implementation.

In terms of the time during operation, there was a reduction of approximately nine minutes, this reduction was predominantly associated with the result warning time, the time between the end of the test, and the notification of the operator. This represented a benefit for Renault Cacia because, while one part is being tested, the machine that made that part continues to operate. This means that if the warning is sent sooner, the reaction time to stop that machine in the event of an abnormality is reduced, which prevents the introduction of a new part into the machine and thus generates less scrap. This also enables a quicker restart of the machine, which reduces the amount of time the unit is not generating any income.

Another gain was the decrease in time spent on tag association and tag cleaning since RFID tags do not need extra cleaning after being reintroduced.

One more challenge of this project was that 50% of the client complaints were associated with the out-of-flow parts. This data was impossible to get since for accurate results it was needed to track it for more time, between six to twelve months, although it is predicted that this solution will reduce to 25% the client's complaints.

The fact that this project was implemented on the factory floor allowed to understand the user's view of the application and also gather some future improvements to be made, to upgrade the application to other functionalities that were not in consideration before.

Table 5.1: Comparison of time consumed Before and After Implementation

Step	**Before Implementation**	**After Implementation**
Association Time	50s	25s
Request Placement Time	60s	60s
Test Duration	2700s (45min)	2700s (45min)
Result Warning time	∼450s (∼7.5min)	10s
Reintroduction time	10s	10s
Tag cleaning	30s	0s
Total	3300s (55min)	2805s (∼46min)

5.2 Conclusions

The monitoring of parts is a fundamental resource for companies that aim for traceability and quality improvement of their products. The possibility of knowing the details of the products from the arrival from the supplier until they deliver to the client, combined with all the in-between processes that those parts passed through, and the location throughout the process, is a must-have for every company.

At Renault Cacia one of the biggest challenges is to keep track of the parts that exit the flow for quality control testing and have a reactive response to any abnormality detected in those tests. The supervision of these factors allows Renault to improve their product quality as well as the process. These improvements are associated with the emergence of industry 4.0 technologies, namely the IoT and the RTLS(Real-Time Location System) that allows the creation of smart factories.

Following these ideas was developed a system that meets the tracking of the parts that get out of the normal flow of production, and the improvement of reactive response to abnormalities. This system not only facilitates the storage of data regarding the process of quality control but is also a tool easier to use than the system previously in use in the factory.

The solution developed and implemented on the factory floor, showed to be a useful, low-cost, and user-friendly tool. The User Interface allows the user to access more quickly the information that is fundamental for his work, like the state of quality production of the machines and also reduces the time dispensed on an ordinary task of sending the parts to the quality control center.

Despite the use of RFID technologies already being a reality in the Renault Cacia factory the implemented RFID solution helped to show the factory responsible that this technology can be employed in several sectors of the factory and deliver better results compared with some of the systems already in use.

The connection between the solution developed allied to the process already in execution in the factory, helped to deliver a tool that does not add more work to the operator and reduces the NVA activities associated with this task.

Being this application stored in the factory server allows access to the information to be made in any place inside the factory. This also allows the application developed to be capitalized to all the factory sectors and become a smart factory application.

Finally, analyzing the table 5.2 as the conclusion of this work, the objectives proposed were accomplished, and the expectations of the clients were met.

Table 5.2: Objective Achievements

Objective	**Achieved**
Improve reactive response time	✓
Database to store all interactions	✓
Real-time location of out-of-flow parts	✓
User-friendly interface	✓
Adaptable system for all sectors	✓

5.3 Future improvements

Although the project had met the proposed objectives some improvements can be made, to achieve a better solution and improve the capabilities of the application, and this way provide a more user-friendly and autonomous system.

Some of the improvements that can be made in future works are enumerated below.

1. Add the functionalities from SIL to this application
2. Add an RFID antenna to the other quality control centers
3. Add an RFID reader in every Reintroduction point in order to reduce the operator intervention and the possibility of a tag going with the part for the rest of the line
4. Compile the system with the PLC of every reintroduction point that stops the machine in case of reintroduction failure.
5. Capitalize the program for every sector of the factory.
6. Display the lead time of every quality control operation to optimize the process.
7. Keep track of the last machine verification for every part reference

Although the displayed upgrades will already bring a vast amount of new features, they do not represent all the capabilities that this application can represent to Renault Cacia in years to come.

Intentionally blank page.

References

[1] W. Xu, "Benchmark comparison of javascript frameworks react, vue, angular and svelte," *Hämtad den*, vol. 5, 2021.

[2] D. F. J. d. Rocha, "Proposal of an automatic traceability system, in industry 4.0," Master's thesis, Universidade de Aveiro, Aveiro, Nov. 2018, http://hdl.handle.net/10773/26048.

[3] R. G. L. Bártolo, "Integração de sistemas de rastreabilidade em ambiente industrial," Master's thesis, Universidade de Aveiro, Aveiro, 2013, http://hdl.handle.net/10773/12483.

[4] F. A. J. d. Rocha, "Sistema automático 4.0 de rastreabilidade e atualização de of na oli," Master's thesis, Universidade de Aveiro, Aveiro, Nov. 2020, http://hdl.handle.net/10773/31271.

[5] Cognex, "What are laser scanners," https://www.cognex.com/what-is/industrial-barcode-reading/laser-scanners, accessed: 2023-03-04.

[6] ——, "What is a 2-d barcode," https://www.cognex.com/what-is/industrial-barcode-reading/2-d-matrix-codes, accessed: 2023-03-04.

[7] PolyGAIT, "Types of rfid," https://polygait.calpoly.edu/what-rfid/types-of-rfid, accessed: 2023-04-28.

[8] IDTRONIC, "Bluebox cx-avi uhf user manual," https://download.idtronic.de/BLUEBOX/BLUEBOX%20SDK.zip, accessed: 2023-04-28.

[9] HID, "Slimflex™ tag datasheet," https://www.hidglobal.com/sites/default/files/documentlibrary/idt-rfid-il-slimflex-tag-ds-en.pdf, accessed: 2023-04-28.

[10] B. Yang, X. Ma, C. Wang, and A. Guo, *User story clustering in agile development: a framework and an empirical study*, 6th ed. Frontiers of Computer Science, 2023, vol. 17.

[11] H. Cañas, J. Mula, M. Díaz-Madroñero, and F. Campuzano-Bolarín, "Implementing industry 4.0 principles," *Computers Industrial Engineering*, vol. 158, p. https://doi.org/10.1016/j.cie.2021.107379, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835221002837

[12] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, aug 2014.

[13] P. Olsen and M. Borit, "How to define traceability," *Trends in Food Science Technology*, vol. 29, no. 2, pp. 142–150, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924224412002117

[14] A. Bento, *Uma contribuição para a melhoria de um sistema de rastreabilidade no setor automóvel". Tese de Mestrado em Desenvolvimento de Tecnologia*, Curitiba, 2009.

[15] Cognex, "What is a 1-d barcode," https://www.cognex.com/what-is/industrial-barcode-reading/1d-barcodes, accessed:2023-03-07.

[16] R. Weinstein, "Rfid: a technical overview and its application to the enterprise," *IT Professional*, vol. 7, no. 3, pp. 27–33, 2005, 10.1109/MITP.2005.69.

[17] A. Website, "O que É rfid?" https://www.ncontrol.com.pt/o-que-e-rfid.html, accessed: 2023-03-07.

[18] S. Armstrong, "Which rfid frequency is right for your application?" https://www.atlasrfidstore.com/rfid-insider/which-rfid-frequency-is-right-for-your-application, Feb 2012, accessed: 2023-03-08.

[19] K. Finkenzeller, *RFID HANDBOOK Fundamentals and applications in contacless smart cards, radio frequency identification and near-field communication*, 3rd ed. Munich, Germany: Giesecke Devrient GmbH, 2010.

[20] L. B. Bourque and V. Clark, *Processing data: The survey example.* Sage, 1992, no. 85.

[21] "What is data processing?" https://www.talend.com/resources/what-is-data-processing/, accessed: 2023-03-10.

[22] S. G. Pérez Ibarra, J. R. Quispe, F. F. Mullicundo, and D. A. Lamas, "Herramientas y tecnologías para el desarrollo web desde el frontend al backend," in *XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja)*, 2021.

[23] platzi, "Qué es frontend y backend, diferencias y características," https://platzi.com/blog/que-es-frontend-y-backend/, accessed: 2023-03-14.

[24] palaksinghal9903, "Frontend vs backend," https://www.geeksforgeeks.org/frontend-vs-backend/, Apr 2023, accessed: 2023-03-15.

[25] "What is flask?" https://www.educative.io/courses/flask-develop-web-applications-in-python/qZWAmEGDBkR, accessed: 2023-03-14.

[26] "What is flask python," https://pythonbasics.org/what-is-flask-python/, accessed: 2023-03-14.

[27] "Flask python," https://blog.back4app.com/python-frameworks/#3_Flask, Aug 2021, accessed: 2023-03-14.

[28] M. Grinberg, *Flask web development: developing web applications with python.* " O'Reilly Media, Inc.", 2018.

[29] "Nestjs - a progressive node.js framework," https://docs.nestjs.com, accessed: 2023-03-14.

[30] "What is nest.js? a look at the lightweight javascript framework," https://kinsta.com/knowledgebase/nestjs/, Nov 2022, accessed: 2023-03-14.

[31] "What is angular?" https://angular.io/guide/what-is-angular, accessed: 2023-03-16.

[32] T.-D. Tripon, G. Adela Gabor, and E. Valentina Moisi, "Angular and svelte frameworks: a comparative analysis," in *2021 16th International Conference on Engineering of Modern Electric Systems (EMES)*, 2021, pp. 1–4, 10.1109/EMES52337.2021.9484119.

[33] "Advantages and disadvantages of angular," https://www.edureka.co/blog/advantages-and-disadvantages-of-angular/, Apr 2023, accessed: 2023-04-02.

[34] "React," https://legacy.reactjs.org/, accessed: 2023-03-16.

[35] D. Herbert, "What is react.js?" https://blog.hubspot.com/website/react-js, Jun 2022, accessed: 2023-03-16.

[36] "Vuejs," https://vuejs.org/guide/introduction.html, accessed: 2023-03-16.

[37] P. Beynon-Davies, *Database systems.* Bloomsbury Publishing, 2017.

[38] O. Portugal, "What is a database?" https://www.oracle.com/pt/database/what-is-database/, accessed: 2023-03-09.

[39] tanyasaxena1, "What is a database ?" https://www.geeksforgeeks.org/what-is-database/, Apr 2023, accessed: 2023-04-10.

[40] "Dmbs integrity constraints," https://www.javatpoint.com/dbms-integrity-constraints, accessed: 2023-03-16.

[41] "Create ibm cloud account what is a relational database?" https://www.ibm.com/topics/relational-databases, accessed: 2023-03-10.

[42] D. D. Chamberlin, "Early history of sql," *IEEE Annals of the History of Computing*, vol. 34, no. 4, pp. 78–82, 2012, 10.1109/MAHC.2012.61.

[43] KumarVivek, "What is power apps? - power apps," https://learn.microsoft.com/en-us/power-apps/powerapps-overview, accessed: 2023-03-19.

[44] Nakivo, "Why you shouldn't use sharepoint online as a database," https://www.nakivo.com/blog/why-you-shouldnt-use-sharepoint-online-as-a-database/, accessed: 2023-03-10.

[45] S. Cawood and S. Wallbridge, *Microsoft SharePoint 2013.* McGraw-Hill Education, 2013, accessed: 2023-03-19.

[46] P. Lab, “O que É microsoft sharepoint e quais são seus benefícios?” https://lattinegroup.com/microsoft-sharepoint/o-que-e-microsoft-sharepoint/, Mar 2022, accessed: 2023-03-19.

[47] https://www.zebra.com/br/pt/products/rfid/rfid-readers/fx9600.html, accessed: 2023-04-29.

[48] IDTRONIC, “Bluebox show manual,” https://www.idtronic-rfid.com/d/1278380/d/bbshow.pdf, accessed: 2023-04-28.

[49] Datalogic, “Product reference guideindustrial corded handheld area imager bar code readerpowerscan pd9530/pbt9500/pm9500,” https://cdn.datalogic.com/Download?iddwnfile=29931, accessed: 2023-04-28.

Appendix A

Appendix A

A.1 Angular

To develop the user interface in this project the framework used was angular, as mentioned during the project this framework is based on JavaScript language and provides many features, known as Angular material, that allow quicker development.

It is based on the creation of components these components are composed of an HTML file, a CSS or SCSS file, and a TypeScript file, being responsible for adding the features, giving the style to the page, and all the logical coding of the page, respectively (figure A.1).

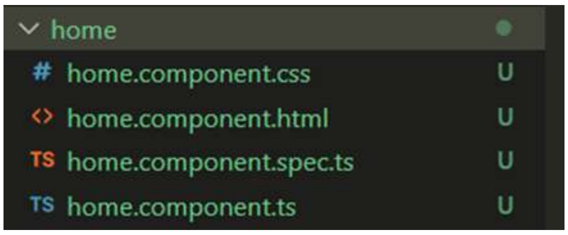


Figure A.1: Angular project base folder

The creation of an Angular project is quite simple, since all the documentation needed for doing so, is displayed on the Angular website.

A.2 User interface functionalities

The figures that are presented next, represent the interactive menus that are part of the application developed.

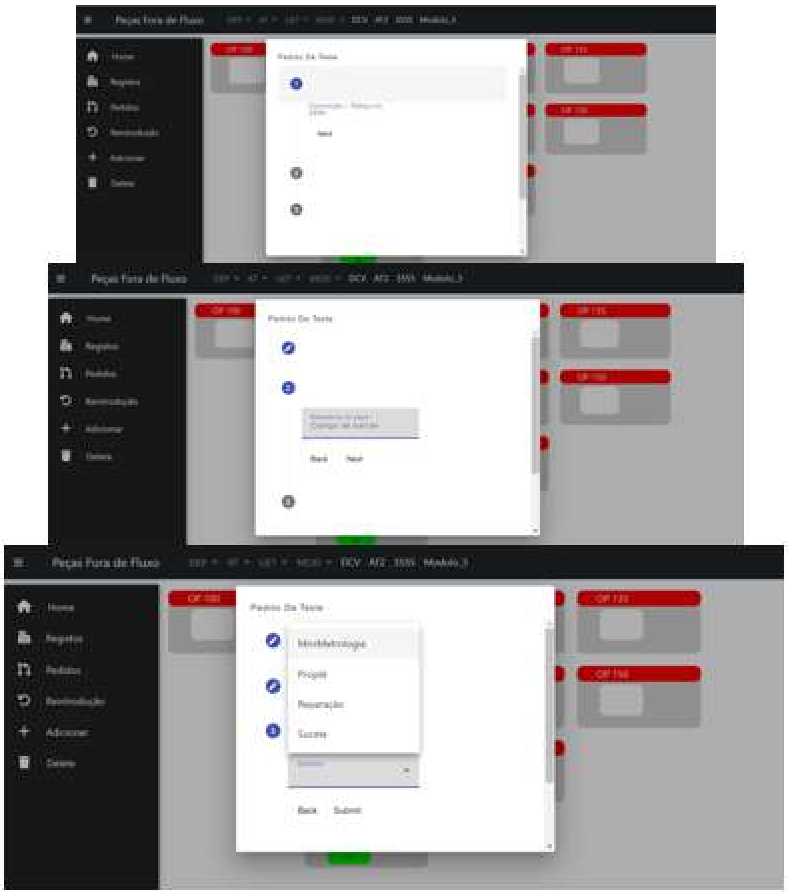
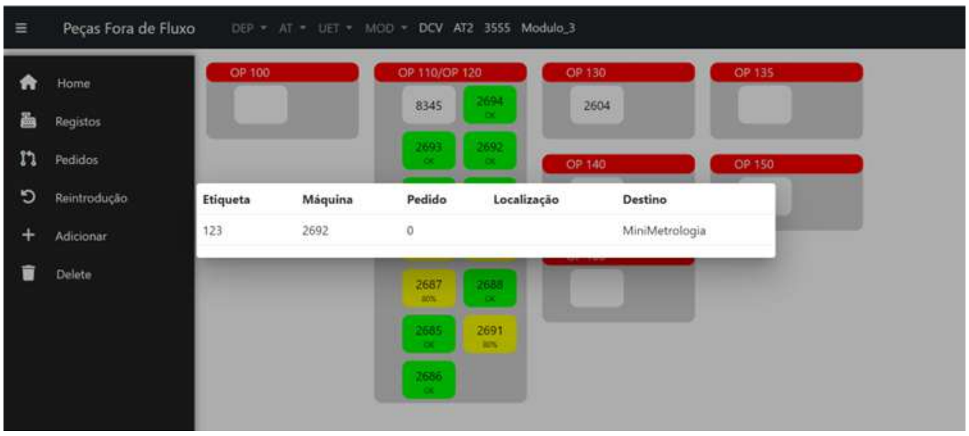


Figure A.2: Quality request placement menu



Etiqueta	Máquina	Pedido	Localização	Destino
123	2692	0		MiniMetrologia

Figure A.3: Data stored from request placement menu

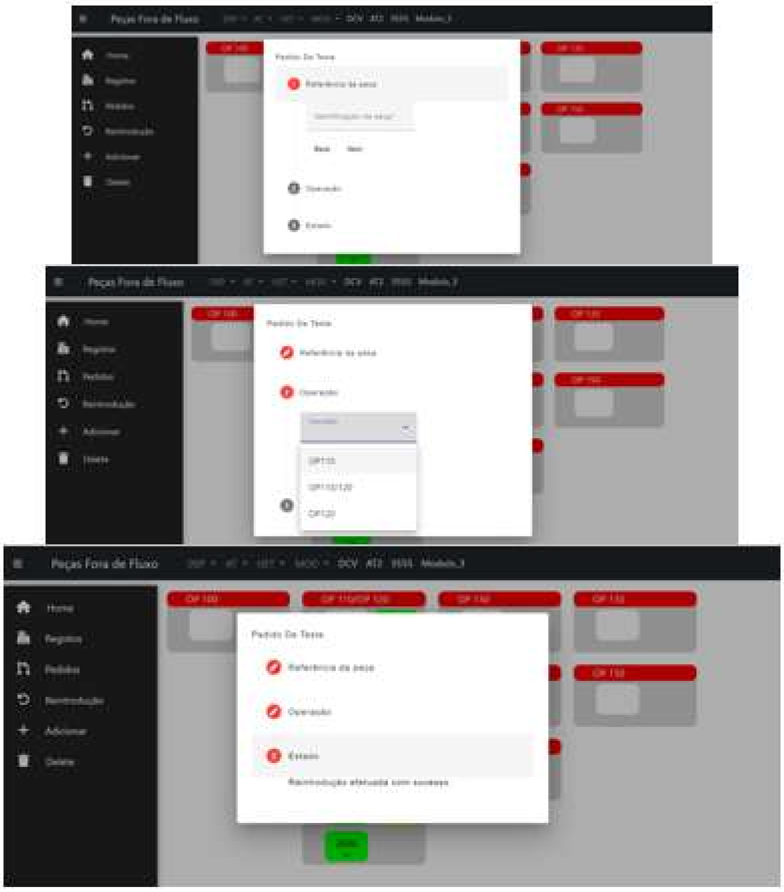
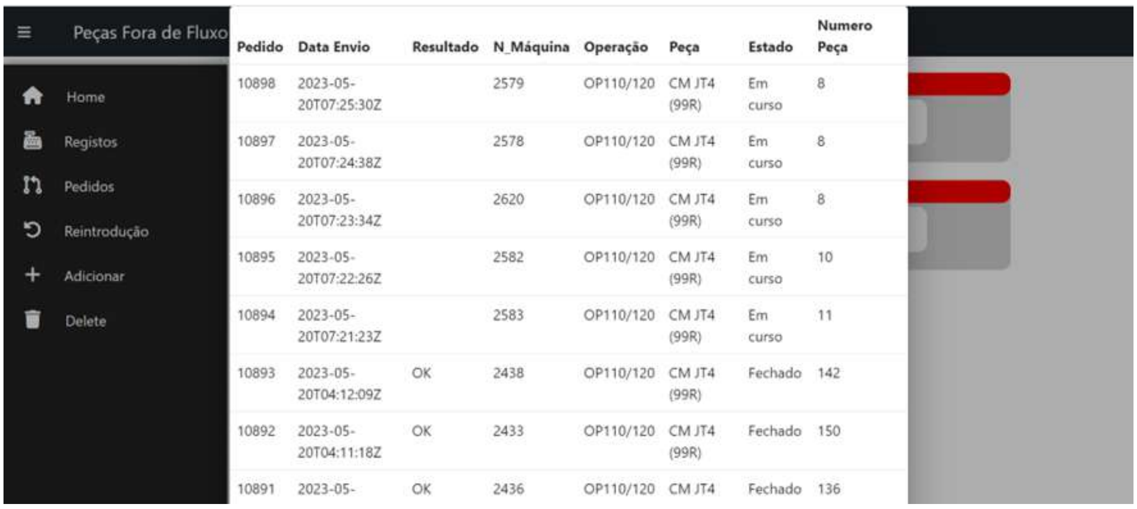


Figure A.4: Parts reintroduction menu



Pedido	Data Envio	Resultado	N_Máquina	Operação	Peça	Estado	Numero Peça
10898	2023-05-20T07:25:30Z		2579	OP110/120	CM JT4 (99R)	Em curso	8
10897	2023-05-20T07:24:38Z		2578	OP110/120	CM JT4 (99R)	Em curso	8
10896	2023-05-20T07:23:34Z		2620	OP110/120	CM JT4 (99R)	Em curso	8
10895	2023-05-20T07:22:26Z		2582	OP110/120	CM JT4 (99R)	Em curso	10
10894	2023-05-20T07:21:23Z		2583	OP110/120	CM JT4 (99R)	Em curso	11
10893	2023-05-20T04:12:09Z	OK	2438	OP110/120	CM JT4 (99R)	Fechado	142
10892	2023-05-20T04:11:18Z	OK	2433	OP110/120	CM JT4 (99R)	Fechado	150
10891	2023-05-	OK	2436	OP110/120	CM JT4	Fechado	136

Figure A.5: Information gathered from SharePoint menu

Intentionally blank page.

Appendix B

Appendix B

B.1 Security equipment project

During the internship at Renault Cacia, it was proposed to solve a problem about the matter of keeping a record of security material that was used and who used it. The material in question was the security equipment used to lift heavy loads, the material was stored in a cabinet as shown in figure B.1.

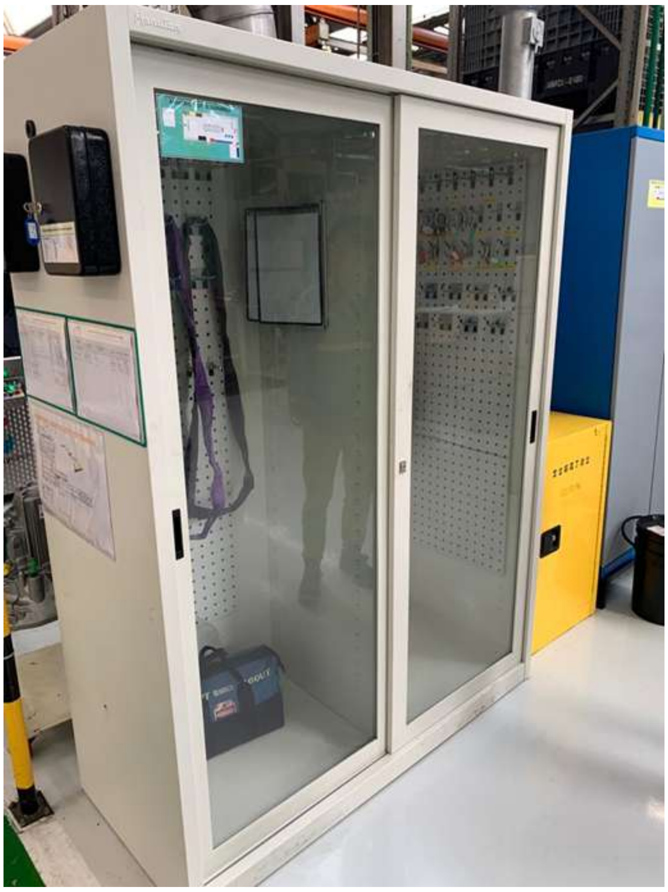
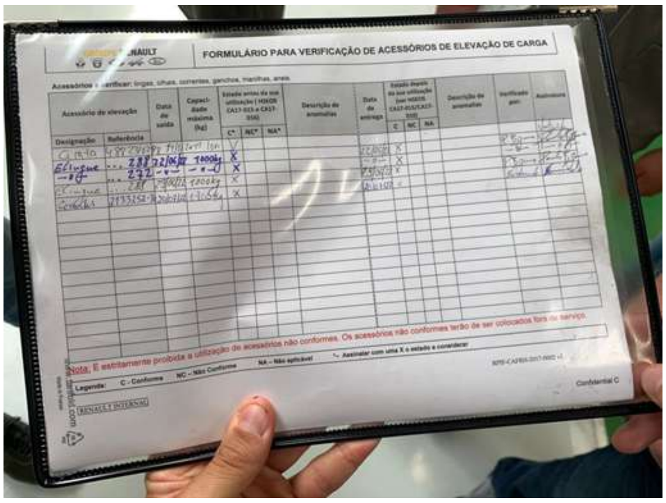


Figure B.1: Security equipment's cabinet

The problem presented was that the user of the told material, for every time of use needed to make a record of what material was taken, the date of when it was requested, and if the material was in the condition of use or if it had any abnormality that could endanger the user. The same record should also be filled at the return time. The sheet to be filled is presented in figure B.2.


RENAULT

FORMULÁRIO PARA VERIFICAÇÃO DE ACESSÓRIOS DE ELEVAÇÃO DE CARGA

Acessórios a verificar: lingas, cintas, correntes, ganchos, manilhas, anéis

Acessório de elevação		Data de saída	Capacidade máxima (kg)	Estado antes da sua utilização			Descrição de anomalias	Data de entrega	Estado depois da sua utilização			Descrição de anomalias	Verificado por	Assinatura
Designação	Referência			C*	NC*	NA*			C	NC	NA			
[illegible]	[illegible]	[illegible]	[illegible]	✓				[illegible]	X				[illegible]	[illegible]
Eslingue	... 288	22/06/22	1000kg	X				[illegible]	X				[illegible]	[illegible]
[illegible]	... 272	[illegible]	[illegible]	X				[illegible]	X				[illegible]	[illegible]
[illegible]	... 288	[illegible]	1000kg	X				[illegible]						
[illegible]	[illegible]	[illegible]	[illegible]	X										

Nota: É estritamente proibida a utilização de acessórios não conformes. Os acessórios não conformes terão de ser colocados fora de serviço.

Legenda: C - Conforme NC - Não Conforme NA - Não aplicável * - Assinalar com uma X o estado a considerar

RENAULT INTERNAL

Confidential C

Figure B.2: Registration sheet for security equipment

Taking advantage of the landed Zebra FX9600 RFID reader B.3, the solution architecture presented in figure B.4 was presented. The solution proposed was well accepted, developed, and implemented.

Figure B.3: RFID reader Zebra FX9600

It consists of the use of an RFID card reader to read the user identifier card, if it has authorization to access the cabinet, the locks unlock and he can open the door.

Then he takes the material he needs and after the closure of the door, the RFID reads all the tags inside the cabinet. These tags are attached to the pieces of equipment

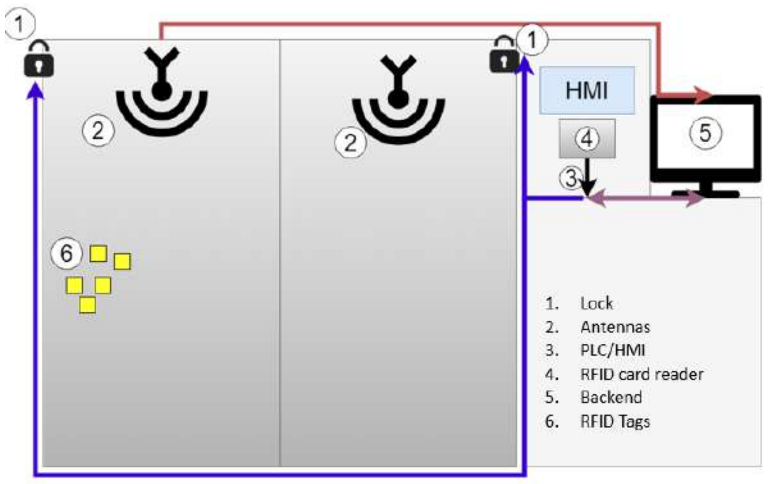


Figure B.4: Solution architecture

and in the database are identified by a number of tag and all the information regarding the equipment.

After reading the tags that remained inside the cabinet the software makes a comparison with what was inside the cabinet before the interaction with the user and displays in the HMI a form for the user to answer regarding the conformity of the equipment. Finally, it registers the information of every interaction in the database, with the finality of being always available for consulting.

In conclusion, this small project was a success, and the acquisition of the material was made. However, it showed that this Zebra FX9600 was designed to be employed in projects of retail, and was less suitable for projects in an industrial environment. This conclusion helped to make the decision of using the BlueBox RFID reader for the project of this thesis.

Intentionally blank page.

Appendix C

Appendix C

C.1 Scripts

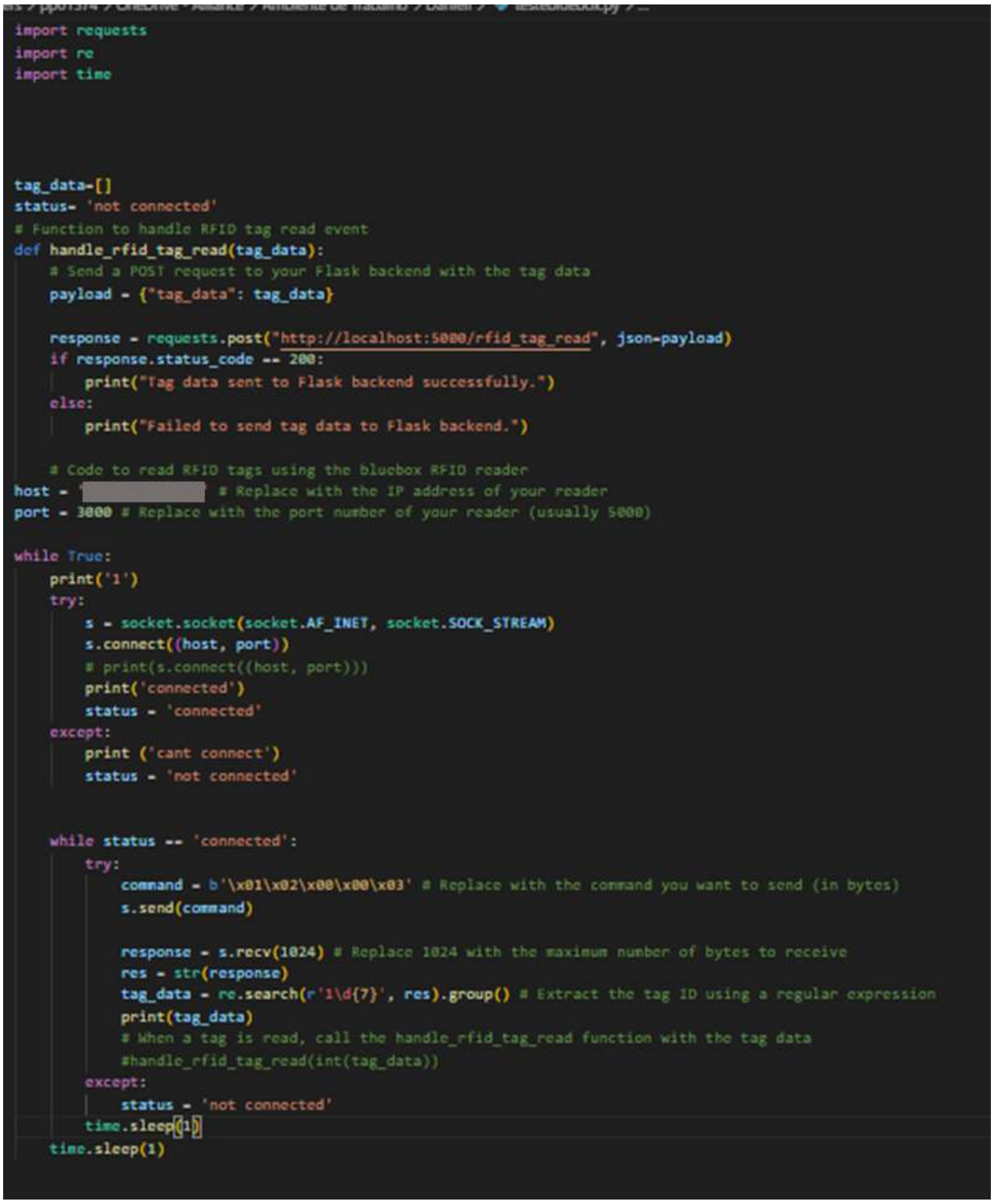
This appendix shows parts of the code that was developed in the project.

C.1.1 RFID script

Figure C.1 bellow depicts the code developed in order to make the connection between the BlueBox RFID reader and the Flask backend.

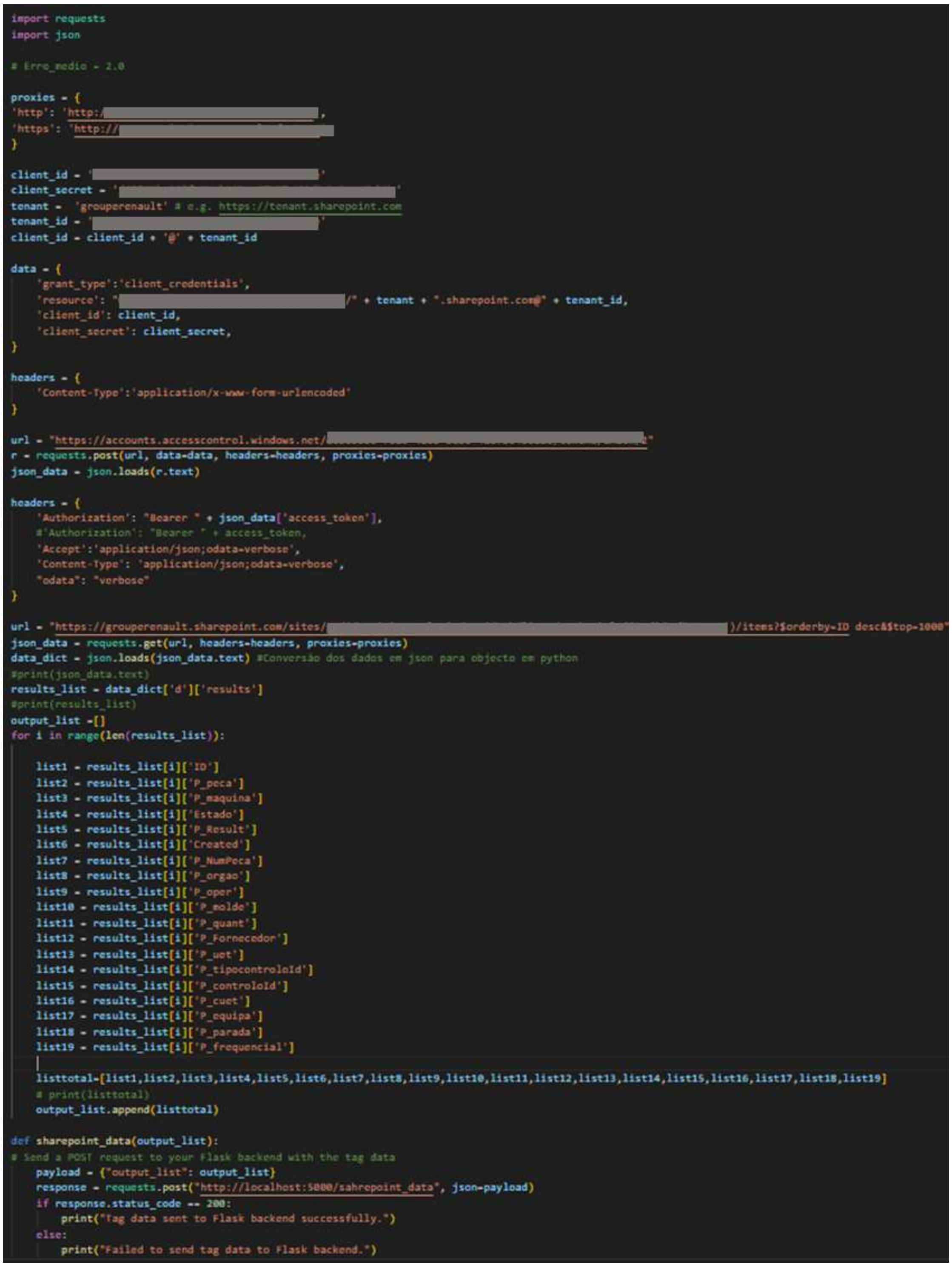
C.1.2 SharePoint Script

Figure C.2 bellow depicts the code developed in order to make the connection between SharePoint and the Flask backend.



```python
import requests
import re
import time




tag_data=[]
status= 'not connected'
# Function to handle RFID tag read event
def handle_rfid_tag_read(tag_data):
    # Send a POST request to your Flask backend with the tag data
    payload = {"tag_data": tag_data}

    response = requests.post("http://localhost:5000/rfid_tag_read", json=payload)
    if response.status_code == 200:
        print("Tag data sent to Flask backend successfully.")
    else:
        print("Failed to send tag data to Flask backend.")

    # Code to read RFID tags using the bluebox RFID reader
host = '[redacted]' # Replace with the IP address of your reader
port = 3000 # Replace with the port number of your reader (usually 5000)

while True:
    print('1')
    try:
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        s.connect((host, port))
        # print(s.connect((host, port)))
        print('connected')
        status = 'connected'
    except:
        print ('cant connect')
        status = 'not connected'


    while status == 'connected':
        try:
            command = b'\x01\x02\x00\x00\x03' # Replace with the command you want to send (in bytes)
            s.send(command)

            response = s.recv(1024) # Replace 1024 with the maximum number of bytes to receive
            res = str(response)
            tag_data = re.search(r'1\d{7}', res).group() # Extract the tag ID using a regular expression
            print(tag_data)
            # When a tag is read, call the handle_rfid_tag_read function with the tag data
            #handle_rfid_tag_read(int(tag_data))
        except:
            status = 'not connected'
        time.sleep(1)
    time.sleep(1)
```

Figure C.1: RFID connection Script



```python
import requests
import json

# Erro_medio = 2.0

proxies = {
'http': 'http:/[redacted],
'https': 'http://[redacted]
}

client_id = '[redacted]'
client_secret = '[redacted]'
tenant =  'grouperenault' # e.g. https://tenant.sharepoint.com
tenant_id = '[redacted]'
client_id = client_id + '@' + tenant_id

data = {
    'grant_type':'client_credentials',
    'resource': "[redacted]/" + tenant + ".sharepoint.com@" + tenant_id,
    'client_id': client_id,
    'client_secret': client_secret,
}

headers = {
    'Content-Type':'application/x-www-form-urlencoded'
}

url = "https://accounts.accesscontrol.windows.net/[redacted]"
r = requests.post(url, data=data, headers=headers, proxies=proxies)
json_data = json.loads(r.text)

headers = {
    'Authorization': "Bearer " + json_data['access_token'],
    #'Authorization': "Bearer " + access_token,
    'Accept':'application/json;odata=verbose',
    'Content-Type': 'application/json;odata=verbose',
    "odata": "verbose"
}

url = "https://grouperenault.sharepoint.com/sites/[redacted])/items?$orderby=ID desc&$top=1000"
json_data = requests.get(url, headers=headers, proxies=proxies)
data_dict = json.loads(json_data.text) #Conversão dos dados em json para objecto em python
#print(json_data.text)
results_list = data_dict['d']['results']
#print(results_list)
output_list =[]
for i in range(len(results_list)):

    list1 = results_list[i]['ID']
    list2 = results_list[i]['P_peca']
    list3 = results_list[i]['P_maquina']
    list4 = results_list[i]['Estado']
    list5 = results_list[i]['P_Result']
    list6 = results_list[i]['Created']
    list7 = results_list[i]['P_NumPeca']
    list8 = results_list[i]['P_orgao']
    list9 = results_list[i]['P_oper']
    list10 = results_list[i]['P_molde']
    list11 = results_list[i]['P_quant']
    list12 = results_list[i]['P_Fornecedor']
    list13 = results_list[i]['P_uet']
    list14 = results_list[i]['P_tipocontroloId']
    list15 = results_list[i]['P_controloId']
    list16 = results_list[i]['P_cuet']
    list17 = results_list[i]['P_equipa']
    list18 = results_list[i]['P_parada']
    list19 = results_list[i]['P_frequencial']

    listtotal=[list1,list2,list3,list4,list5,list6,list7,list8,list9,list10,list11,list12,list13,list14,list15,list16,list17,list18,list19]
    # print(listtotal)
    output_list.append(listtotal)

def sharepoint_data(output_list):
# Send a POST request to your Flask backend with the tag data
    payload = {"output_list": output_list}
    response = requests.post("http://localhost:5000/sahrepoint_data", json=payload)
    if response.status_code == 200:
        print("Tag data sent to Flask backend successfully.")
    else:
        print("Failed to send tag data to Flask backend.")
```

Figure C.2: SharePoint connection Script