FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Smart Diagnostic System for Machine Problems

Nuno Araújo Campos

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Armando Araújo

July 18, 2017

Resumo

O serviço pós-venda prestado por uma empresa é uma das áreas fulcrais para o seu desenvolvimento, estabelecimento de nível de qualidade no mercado e um dos mais importantes métodos para atrair clientes e aumentar vendas.

Com um crescimento contínuo vem também a necessidade de implementação de novas ferramentas de suporte técnico para tornar este processo mais rápido e eficaz. Seguindo esta premissa, a Adira SA, empresa parceira no desenvolvimento deste projeto de dissertação, quer garantir a qualidade máxima dos seus serviços e a satisfação dos clientes. Atualmente, o serviço pós-venda da Adira é bastante complexo, requerendo constantemente o recurso a engenheiros qualificados e com total conhecimento dos produtos desenvolvidos pela empresa. O processo, de momento implementado, implica assim, demoras a melhorar no serviço pós-venda.

Sendo que a Adira está em constante progresso para optimizar as suas metodologias decidiu apostar numa aplicação inteligente e autónoma para identificação e resolução de problemas associados aos seus produtos, surgindo desta forma, o "Smart Diagnostic System for Machine Problems".

Por forma a alcançar o sucesso no projeto acima apresentado, foi necessário, numa primeira instância, realizar um estudo aprofundado do atual processo implementado do serviço pós-venda da Adira. Para tal, foram ainda identificados os principais problemas na adaptação deste sistema, para o seu funcionamento autónomo e independente, na forma de uma aplicação WEB. Posteriormente, foi realizado um estudo dos dados existentes de forma a desenhar as características necessárias deste sistema assim como, a idealização da sua base de dados e todo o conceito geral do funcionamento do processo. Após delineada toda a metodologia de estudo, foi desenvolvida a aplicação, com recurso a diversas tecnologias, de forma a iniciar uma fase de testes e recolha de dados associados aos produtos da Adira.

Em última instância, após um período de estudos do funcionamento da aplicação, foram recolhidos dados estatísticos para determinar o sucesso da implementação, de forma a ser possível identificar o caminho a seguir na evolução da mesma.

Abstract

The after-sales service provided by any company is one of the key areas for its development, establishment of quality level in the market and one of the most important methods to attract customers and increase sales.

With a continuous growth it also comes the need to implement new technical support tools to make this process faster and more effective. Following this methodology, Adira SA, partner company on the development of this project, wants to assure a maximum quality of its services as well as customer satisfaction. Adira's after-sales service is currently quite complex, requiring the involvement of qualified engineers with knowledge on all of Adira's developed products and, thus, is a process that can take several days.

Since Adira is in constant progress to optimize its methodologies, it decided to invest in an intelligent and autonomous application to identify and solve the problems associated with its products, resulting in the "Smart Diagnostic System for Machine Problems".

In order to successfully develop this project it was necessary, firstly, an accurate study of Adira's after-sales service as well as the identification of the main problems for the adaption to an autonomous and independent operation system in the form of a WEB application. Secondly, it was made a study of the existing data in order to sketch the main requirements of this system moreover an idealization of its database and the whole general concept of this operation procedures. After a careful study involving the whole concept of the project it was finally developed an application with the use of many different, required and available technologies, in order to start a test phase and collection of useful data.

At last, after an analysis period of the Smart Diagnostic System operation mechanism it were made some statistical collections to determine the success of the implementation and to identify the path to follow for a continuous and successful improvement of the developed system.

Acknowledgment

I want to thank Marcos Ribeiro for spotting one missing space on my document.

My sister for the whole thesis review.

All my faculty colleagues for the commitment on overcoming this hard task together.

Michaela Stenclová for giving me motivation and support to finish my Master's degree.

All of the Adira SA personnel for the help and involvement to carry on with this project and a special thanks to Asif Mohammed for all the effort that he made to make it possible.

All the students that participated with me in ESA Product Assurance and Safety course for helping me to clear my mind of worries before my final thesis review.

At last, I want to thank professor Armando Araújo for the commitment on the project and for all the advices.

Nuno Araújo Campos

"Isto não é para todos, é só para alguns.
Não é para quem quer, é para quem pode.
E para poder é preciso estar."
Unknown author

Contents

1 Introduction **1**
1.1 The concept . . . 1
1.2 Context . . . 2
1.3 Problem characterization . . . 3
1.4 Proposed solution and goals . . . 3
1.5 Structure . . . 4

2 State of the Art and Important Concepts **5**
2.1 Industry 4.0 . . . 5
2.2 Smart factories . . . 7
2.2.1 Cloud Computing . . . 8
2.3 IoT . . . 9
2.4 Related Projects . . . 10
2.4.1 SelSus . . . 10
2.5 Java . . . 12
2.6 Databases . . . 13

3 Diagnostic system process **15**
3.1 Introduction . . . 15
3.2 Process study . . . 15
3.2.1 Process flow . . . 16
3.3 Main Challenges . . . 16
3.3.1 Problem 1 - Problem description in open text format . . . 16
3.3.2 Problem 2 - Lack of organized information for analyses . . . 17
3.4 Data mining . . . 17
3.4.1 Used technologies . . . 18
3.5 System Structure . . . 18
3.6 Database . . . 19
3.6.1 Smart diagnostic requirements . . . 19
3.6.2 Database structure . . . 19
3.7 Code structure . . . 23
3.7.1 User Package . . . 23
3.7.2 Admin Package . . . 24
3.7.3 Model package . . . 25
3.7.4 Util package . . . 25

4 Smart Diagnostic System Web Application **29**
4.1 Used Technologies . . . 29
4.1.1 Eclipse . . . 29
4.1.2 Tomcat Server . . . 30
4.1.3 Vaadin . . . 30
4.1.4 JDBC . . . 30
4.2 Client Interface . . . 31
4.2.1 Main page . . . 31
4.2.2 Step by step solutions Page . . . 31
4.2.3 Adira Support Page . . . 32
4.3 Admin Interface . . . 32
4.3.1 Login page . . . 33
4.3.2 Admin - Fault page . . . 33
4.3.3 Admin - Symptom page . . . 34
4.3.4 Admin - Solution page . . . 34
4.4 Edit Database . . . 37
4.4.1 Admin - Edit Locations page . . . 37
4.4.2 Admin - Edit State page . . . 37
4.4.3 Admin - Edit Alarm page . . . 38
4.4.4 Admin - Edit Fault description page . . . 39
4.5 Conclusion . . . 39
4.5.1 Existent Data . . . 39
4.5.2 Data analysis . . . 39
4.6 Smart Diagnostic System Database data collection . . . 41
4.7 Detailed analysis on Press Brakes . . . 41
4.8 Smart Diagnostic System Impact . . . 42

5 Future work and conclusions **43**
5.1 Smart Diagnostic System Conclusions . . . 43
5.2 Fulfilled goals . . . 44
5.3 Future work . . . 44

A Process Development **45**
A.1 Process questionnaire . . . 45
A.2 Diagnostic process flow diagram . . . 46
A.3 Database Structure . . . 47
A.4 Modular Machine Dissection . . . 48
A.5 Data Analysis . . . 50

References **55**

List of Figures

1.1 Industrial Revolutions 2

2.1 Industry 4.0 6
2.2 Smart Factory 7
2.3 Cloud Computing 8
2.4 Internet of Things 9
2.5 SelSus Core Architectural Concept 11
2.6 SelSus Architecture 12

3.1 Data statistics 18
3.2 Machines frequency graph 18
3.3 Adira's diagnostic system structure 19
3.4 Database symptom structure 20
3.5 Database diagnostic structure 21
3.6 Database fault structure 22
3.7 Database solution structure 22
3.8 Eclipse package hierarchy 23

4.1 Apache Tomcat Server Scheme 30
4.2 Main Page 31
4.3 Step by step solutions Page 32
4.4 Adira Support Page 33
4.5 Login Page 33
4.6 Admin - Fault page 34
4.7 Admin - Symptom page 35
4.8 Admin - Solution page 36
4.9 Admin - Edit Locations page 37
4.10 Admin - Edit State page 38
4.11 Admin - Edit Alarm page 38
4.12 Admin - Edit Fault description page 39
4.13 Collected Data distribution 40
4.14 Data analysis 40
4.15 Database - data collection 41
4.16 Press Brakes - chamados information 41

A.1 Process Diagram 46
A.2 Database Structure 47
A.3 Database Inputs 50
A.4 Data analysis excel 1 51

A.5 Data analysis excel 2 52
A.6 Data analysis excel 3 53

List of Tables

A.1 Press Brake Structure 48
A.2 Shear Structure 49

Acronyms

IoT	Internet of Things
SDS	Smart Diagnosis System
DB	Database
CPS	Cyber-physical-systems
ERP	Enterprise Resource Planning
MES	Manufacturing Execution Systems
IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
SQL	Structured Query Language

Chapter 1

Introduction

This document reports all the developed work in order to accomplish the buildup of a Smart Diagnostic System for Machine Problems from the identification of all the related technological parameters until the implementation of the final product.

This chapter consists on the identification of the developed concept, the explanation of the context in which this topic is inserted and its relation with industry advancements. In addition to this, it is also presented the structure of this study and its intended goals.

1.1 The concept

During the course of time the importance of maintenance in the industry world is gradually increasing and currently, it is playing a major role on the manufacturing process, making the difference on productivity and quality [1].

Smart Diagnostic System for Machine Problems is a concept enrolled on the idea of maintenance embed in smart factories and it's based on self-diagnosis of faults and automated troubleshoot [1]. The main goal on the implementation of a Smart Diagnosis System (SDS) is to build a detailed analysis about the tested hardware in order to provide to the user the necessary information about its working condition, possible failures and the appropriated instructions for their solutions as well as usage hints to prevent future damages and increase the equipment life expectancy [2].

The ideal implementation of this concept relies on the usage of simplicity. One example of a revolutionary application of SDS is "Smart diagnosis and method for home appliance" which was developed by the Korean company LG Electronics[1]. This invention provides a home appliance diagnostic system that retrieves parameters from an appliance responsible for outputting product information as a frequency signal and a database (DB) server responsible for the interpretation of the diagnosis and the delivery of a countermeasure against possible encountered faults [3].

[1] http://www.lg.com/au/support/smart-diagnosis.

1.2 Context

The competitiveness in industrial market is in constant evolution and it sets new goals on a daily basis. In order to keep up with the stream it is needed to have steadfast improvements and go along on working to always achieve new boundaries [4].

The technological advancements and the development of new working methods are the main factors that define business success. Hence, in the course of time, the industrial field was mainly pronounced by three high-significant revolutions that brought revolutionary concepts to life [5]. This historical progress can be seen on Figure 1.1.

- Mechanical production, water and steam power
- Mass production, assembly lines and electricity
- Computer assisted and automated production

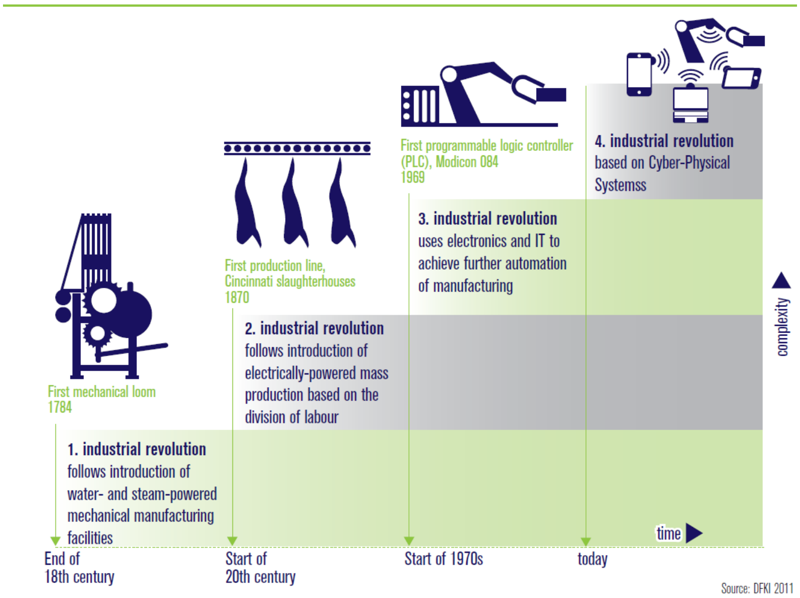


Figure 1.1: Main industrial revolutions during the course of time

Recently, a new concept emerged in industry world and it is already considered by some as the beginning of a fourth industrial revolution. This approach, called Industry 4.0, brings new principles applicable on the modern industry which combines physical-cybernetic systems with Internet of Things (IoT) and cloud computing [4]. SDS is highly connected with this emerging technologies as a result of being an automation process as well as being framed in what we can call Smart Factories.

1.3 Problem characterization

Despite providing a good product, one of the most valuable qualities of a company is based on the after sales support. Adira, as a competitive and innovative company, is stepping up on this field by improving and creating new methods for corrective measures.

The existing problem of the current process relies on the miss-usage of high value resources such as the need of high-skilled engineers to provide direct assistance to clients as they are the only ones with the necessary knowledge to understand and solve possible machine anomalies. In order to solve this problem a new after-sales support method was created so that non-specialized people can easily identify and solve common faults.

1.4 Proposed solution and goals

This thesis consists in a work cooperation with Adira S.A[2] in the development of a Smart Diagnostic System for Machine Problems. Adira is a manufacturer and a global supplier of sheet metal working machinery and one of the world leading companies in its field. In order to accomplish our goal the work was split into four main steps:

1. Text mining
   (a) Understand the process
   (b) Analyze the existing information and requirements of the service department
   (c) Mining the emails for information extraction
2. Development of a relational Database
   (a) Understanding the relations between different objects/components
   (b) Identification of common occurrences
   (c) Sorting and assembling all the information
3. Application development for smart diagnosis
   (a) Sketch of the concept and important functionalities
   (b) Connection with the database
   (c) Development of user interface (front-end)
4. Web application implementation on the overall architecture of industry 4.0
   (a) Simplification and automation of processes
   (b) Application of IoT
   (c) Development of constant monitoring and diagnosis

[2] http://www.adira.pt/

1.5 Structure

Besides the Introduction, this thesis contains 4 more chapters. On chapter 2 it is characterized the state of the art as well as it is presented all the relevant matters in the development of this study.

At chapter 3 it is explored all the study related to the Smart Diagnostic System process flow, the necessary parameters to implement on the application and also, the development of the database structure.

The interface for the SDS is presented on chapter 4, as well as it is made a detailed explanation on all its functionalities by a user point of view.

Ultimately, all the conclusions and further studies are presented on the last chapter, the chapter 5.

Chapter 2

State of the Art and Important Concepts

All the linked concepts needed for the understanding of the used and explained technologies as well as the state of the art are presented on this specific chapter. The first addressed approaches, Industry 4.0, Smart factories, IoT and Cloud computing are allied to automation of processes and the development of the industry. The final explored concepts, Java, Databases and related projects are pertinent for the understanding of the developed technology itself.

2.1 Industry 4.0

Among all the features that are evolving on a daily basis, productivity is the key aspect that describes a industrial revolution [6]. Industry 4.0, being the fourth industrial revolution is no exception to this matter [7, 8]. This concept relies on the communication between humans and machines in Cyber-physical-systems (CPS) making use of large networks and bringing together the IoT concept. [7, 9].

With Industry 4.0 its is possible to handle with the need of rapid, flexible and complex production [9]. IoT and CPS are the main indicators of the guidelines followed by industry 4.0. It converges on the association and interconnection of all the services of a industry process. By converting all the data as digital it is possible to make it available and usable without almost any constrains and fully functional even in real time systems which will result in the optimization of the productivity chain due to the ease provided by this improvements. [10]

The changing capacity of industry 4.0 is built by four main characteristics [7, 8]:

1. Vertical networking
2. Horizontal integration
3. Through-engineering
4. Exponential technologies

As shown on Figure 2.1 presented bellow.

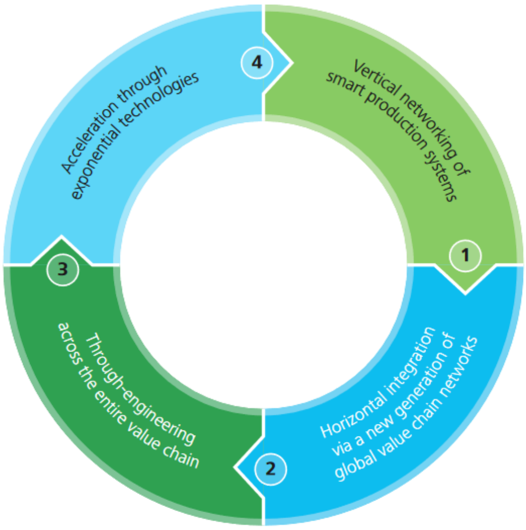


Figure 2.1: Four main characteristics of industry 4.0 [7]

1. **Vertical networking** is focused on resource efficiency and individualized production. By making use of CPS, autonomous organization and maintenance management are highly improved, enabling a fast response to customer-specific and individualized production, to quality fluctuations and even to machine breakdowns [7]. Following this approach, for a further increase of production flexibility and optimization, in the future, all the decision process will be self-optimized by reconfigurable manufacturing systems replacing the need of human experience and expanding the potentiality of vertical network [9].

2. **Horizontal integration** is based on constant traceability of any access on the process chain creating transparency and flexibility. This integration enables new business models due to a faster response for problems and faults. Therefore, horizontal integration facilitates global optimization since it provides the availability of product-data throughout the entire network. It allows any specific, customer or provider, adaption to be made along every stage of the production or supply process chain in a real time domain and dynamically manageable [7].

3. **Through-engineering across the entire value chain** is a characteristic established on the adaption and creation of product systems in order to maintain synergies on new products. The increased availability of data through all process stages is the key aspect for this engineering to impose and thereafter empower the industry 4.0 resilience [7, 8].

4. **Acceleration through exponential technologies** is the combination of all the used innovations to achieve what we denominate as industry 4.0. Technology is the catalyst for every improvement and industry is already requiring high standards for hardware, software and working methods [11]. Industry 4.0 relies on continuous innovation therefore it's always

opened to new and better opportunities provided by emerging concepts that result in more reliability, cost savings, flexibility and self-sustainability like, for example, artificial intelligence or nano-technology [7].

Summing up, industry 4.0 is the transformation of data in knowledge. It is characterized by the strong ability of product customization without the loss of chain flexibility and mass production focused on self configuration, optimization, maintenance and diagnosis.

2.2 Smart factories

The market proliferation and globalization led to the need of higher level enterprise networks [12]. The vision around the smart factory concept is of a physical world interconnected with embed sensors and actuators that interact with each other and with the environment through the use of a network [12]. This enables the production chain to be context-aware and context-sensitive with decentralized information and high dynamical data to assist tasks in real time [8, 12]. Nowadays, the industry management control is mostly preformed with the aid of Enterprise Resource Planning (ERP) and Manufacturing Execution Systems (MES). These applications are framed in the smart factory concept due to the provided aggregation and synchronization of data that they offer [12].

Therefore, as shown on Figure 2.2, a smart factory can be layered in four main categories [8]:

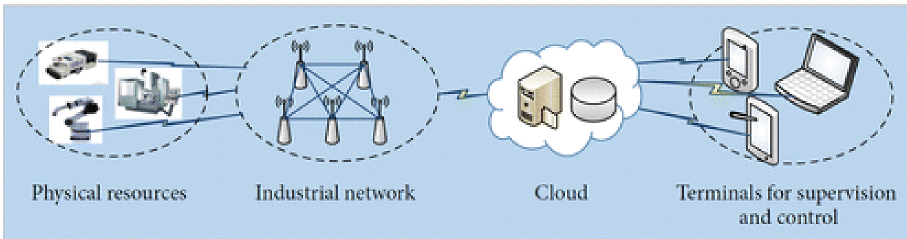


Figure 2.2: Smart Factory layers [8]

- **Physical Resource Layer** - This layer represents the lowest level structure and it consists in self-organized systems of various physical artifacts that interact and understand each other in order to work accordingly to the industrial network demands [8].
- **Industrial Network Layer** - It represents one of the most important structures of a smart factory and it is the layer that enables not only the communication between physical systems, as described previously, but it also provides a communication link between the resource layer and the cloud layer. Industry network is also flexible to variations granting a convenient way of wireless communication [8].
- **Cloud Layer** - This layer consists in a resource pool full of virtualized information. For a proper operation of a smart factory, usually, it's produced a huge amount of data. The cloud provides a flexible solution to deal with this big data and enables the ability of analytics to support management and optimization of the system [8].

- **Supervision and Control Terminal Layer** - It is responsible to link humans and machines in a complementary work development. It enables maintenance, diagnosis, reconfiguration and information access for the user, through a remote and easy way, focused on the interoperability of the whole system [8].

The factory of the future — the smart factory — is a paradise of efficiency where defect and downtime, waste and waiting are long forgotten issues of a long forgotten age [13].

2.2.1 Cloud Computing

As briefly referred before, cloud computing is an information system that provides access to a shared pool of digital resources that can be managed by external parties on remote servers and on-demand [14, 15]. This concept is a fundamental element to pursue the idea of industry 4.0 and for the build up of a smart factory [15].

Cloud computing reinvents the way how industry deals with data and manage not only software but also hardware [15, 16]. For a better understanding, cloud computing, can be layered as shown below on Figure 2.3.

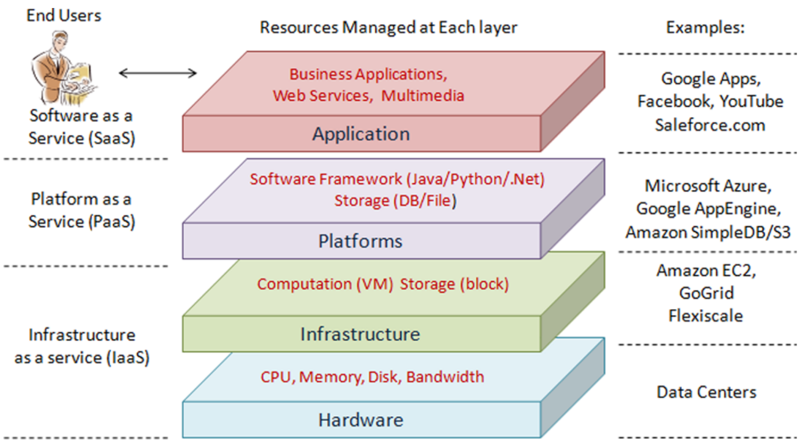


Figure 2.3: Cloud Computing structure [15]

This model is grouped into three main categories:

- Infrastructure as a Service (IaaS) refers to on-demand accessibility of applications over a wide range of devices through a simple and light client interface provisioning infrastructural resources [14, 15].

- Platform as a Service (PaaS) reports to accessibility to set up consumer-created or acquired applications such us software development frameworks, services or libraries and the control and configuration over those applications [14, 15].

- Software as a Service (SaaS) refers to the accessibility of running applications on a cloud infrastructure with the leverage of automatic-scaling adapted to the user needs [14, 15].

Cloud Computing brought up new concepts to the industry world that enhanced the existing work mechanisms and even made possible new and improved ones. These characteristics are summarized bellow [16]:

Dynamic resources provisioning - The system is automatically optimized to ensure resource availability on demand with enough elasticity to scale quickly and appropriated to requirements [14, 15, 16].

Network access - Clouds provide a huge leverage on service utility by being accessible through standard devices connected to the used network facilitating the data control and exchange [14, 15].

Resource pooling - All the data is assigned and reassigned in a dynamic way which is settled accordingly to demands. The user has only an abstract view of what resources he is using, without having control on allocate it, enabling the focus on managing operation costs. It is a method that maximizes the resources utilization [14, 15].

Pay for use - Since the resource sharing is allocated and released as needed, it provides the feature to only pay for the service that you are consuming. Instead of statically determine resources, it will only use exactly what it is needed, avoiding extra costs and providing full exploitation even on abnormal peaks [15, 16].

2.3 IoT

The Internet of things, Gustafsson says, “basically gives a digital voice or a virtual voice to all of these assets, physi- cal things" [13]. On the latest developed systems there is a huge amount of embedded intelligence mechanisms and features, the real problem is the accessibility of all that data and the ease to use it in real time [13]. IoT consists in the integration of objects in an interaction system. This leads to the possibility of devices to communicate with each other and with humans [17].

Figure 2.4 shows a graphical representation of network connections in an IoT view.

Figure 2.4: Internet of Things [18]

A highly distributed architecture with the possibility of controlling the physical world from a distance is a major advancement for industry that helps to optimize processes, reduce costs and maintain a close relation with productivity [19]. Internet of Things is the root of the concepts described before (Industry 4.0, Smart Factories, Cloud Computing and so on), without this communication network it is not possible to step forward into Industry 4.0 and fully automation.

Having all the data in the Web and being collected on an automated process also makes a huge difference for the current industry working mechanisms. It allows not only performance tracking but also and in-depth analysis of big data that is used for many different applications such as prediction, productivity statistics, improvement plan studies and so on [13].

Despite all the advantages of implementing such an architecture the biggest challenge is, without a doubt, the assurance of security [17]. For the data to be remotely accessible it will break some security protocols that might put data accesses in danger.

IoT is also the main characteristic that enables smart objects to be [19]:

- Activity awareness - capability to record information about activities
- Policy awareness - ability to interpret activities respecting predefined standards
- Process awareness - capacity to relate events

These are concepts on a gadget that requires strong links of communication network with all of the factory architecture to receive real time state-updates [20].

2.4 Related Projects

2.4.1 SelSus

SelSus project is a self-healing mechanism for production, resources and systems that counts with the involvement of 15 partners from 6 different countries [21].

This project is a great advancement towards Industry 4.0 and brings innovative concepts of dealing with data to the everyday work policies. The main concept of this project is the "Development of self-healing devices with enhanced capabilities on condition, monitoring, sensing and self-adaption" [21].

The main focus points of SelSus are [21]:

- The development of a new paradigm for self-healing of products, systems and resources that has a high success rate. Hence, it will decrease significantly the repair times.
- Capability of synergy and interaction with the use of Internet of Things in order to do correct diagnostics and prognostics about all the existing equipments on a factory as well as creating a self-aware environment between gadgets. The combination of this characteristics will bring long term sustainability and decrease maintenance efforts and costs on Industrial facilities.

- Cooperative work with human resources. One of the most important factors is enabling an easy channel of communication between humans and machines which will bring productivity benefits.

On figure 2.5 it is possible to analyze the architectural concept of this project and all the involved layers, since data collection and analysis to actuators, for problem solving.

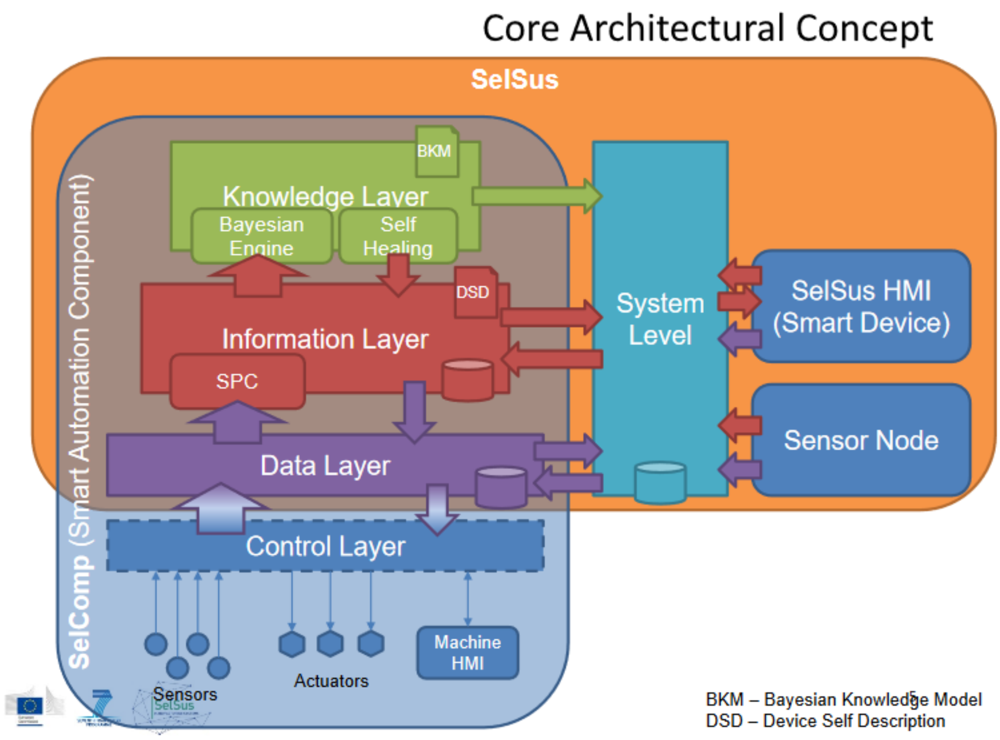


Figure 2.5: SelSus Core Architectural Concept [21]

On figure 2.6 it is possible to analyze the system interfaces involved on the self-healing component an their communication network.

The main benefits of this self-healing devices are stated below [21]:

- Capability of data analysis and interpretation for future improvements and control of productivity
- Real time monitoring system capable of controlling a specific unit based on its current needs
- Pro-active system to repair and maintain equipments without the need of human intervention
- Adaptive structures that can be easily implemented and substituted

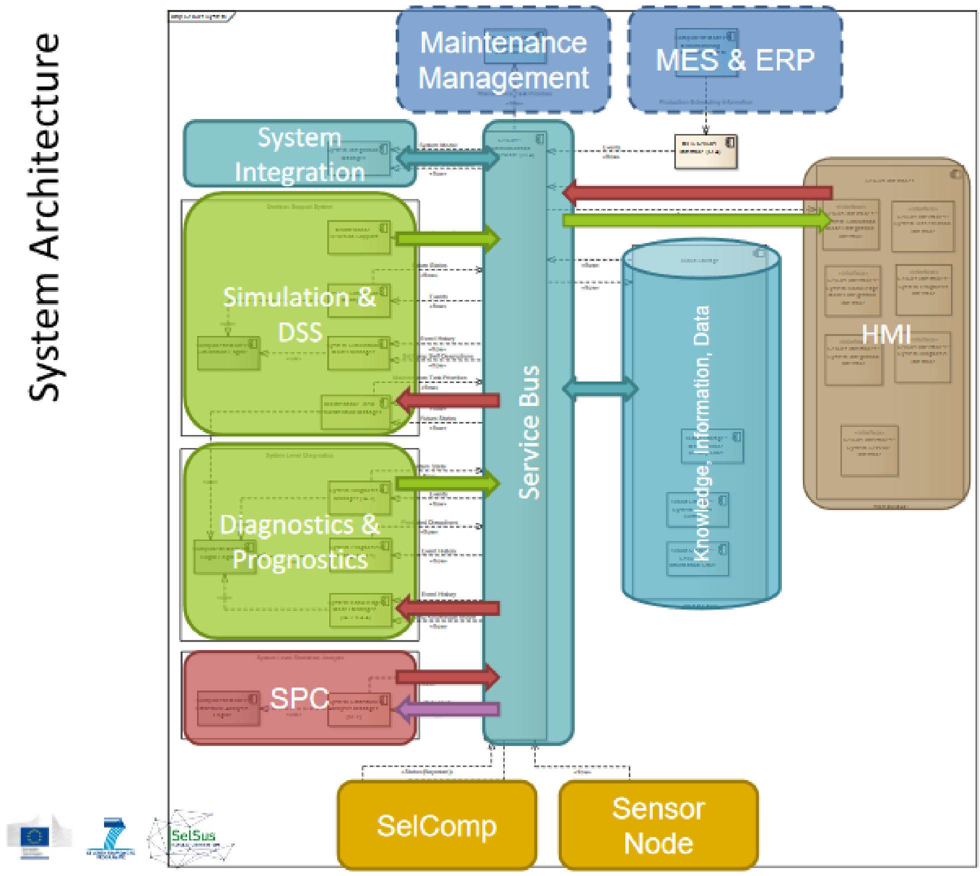


Figure 2.6: SelSus Architecture [21]

2.5 Java

Java is a programming language that is object oriented, using a friendly syntax and it is used for core development to run compiled code on any Java Virtual Machine. Java biggest strength is WORA (write it once run it anywhere) [22].

Java mechanism can be divided in four main points [22]:

- Source - The creation of a document which will hold classes represented in Java language that will be the source of the developed application.
- Compiler - The compiler is first line of defense to test the Java code. It will detect anything that can't be done successfully at runtime and prevents datatype and access violations. It runs the source code and prevents its compilation until everything is correctly set-up to be run.
- Output - After all requirements are satisfied the compiler creates a document in Java bytecode. Java bytecode is an intermediate representation intended to be executed by Java Virtual Machines.

- Java Virtual Machine (JVM) - Implemented software on electronics that can run Java bytecode. It will translate and run the ".class" file that results as Output from the compiled code.

The fact that Java language source files are not compiled into machine code makes it portable and runnable on adequate runtime support [22].

All applications developed in Java need to have one class and one main method. The main method is the first thing that JVM looks for when it's running [22].

In Java, packages are used to control accesses, to prevent name conflicts, to organize the coding structure and so many other organizational advantages. A package is a bundle of related classes.

The latest version developed is Java 8.

2.6 Databases

Data is more important than ever and the demand for big data storage and analysis never stops growing [23]. With big data comes the necessity of big storage methodologies, in this case, Databases.

A Database consists in organized data collections implemented in systems that allow content management, access management and relational functionalities [23]. For this control to be possible there is Database Management System which is a software that serves as the interface for the user to interact with the available data [23, 24].

The usage of a database provides many advantages on dealing with data, such as [25]:

- Efficient access - Customization on users access accordingly to the data management needs
- Data Integrity - Inserting mechanisms with data check to assure that no incorrect information is inputted on the database
- Data protection - Data control and easily backed up to avoid and recover from functional abnormalities.

A Database allows the use of a relational model which is based on affiliation of objects and tables with each other in order to establish interactions and dependencies [25].

The development of a database starts with an analysis of requirements that is followed by conceptual sketches, logic definitions and constant optimization of the system [25].

Chapter 3

Diagnostic system process

This chapter consists on the analyses and understanding of the used process as well as the procedures to create an application able to smartly identify problems on Adira machines and consequently present solutions step by step so that an unqualified person can easily solve any found issues. Firstly it is studied the current Adira service process and defined how it can be implemented and improved by an auto-sufficient application. Secondly, all the available data is mined so it is possible to identify patterns of symptoms/problems and problems/solutions. Finally it is developed the Database structure accordingly to all the needed requirements.

3.1 Introduction

After sales service is one key aspect to maintain a good client share and to stay competitive. Bad services represents not only a client loss but also the loss of credibility and market power.

"Sales go up and down, service stays forever." Jason Goldberg

Following this statement, Adira SA, have continuously invested in a high quality after sales service which can be seen as a "free" way of marketing. High efficiency maintenance provides confidence to the client who usually doesn't mind to pay more for a purchase in order to assure less future inconveniences and more durability. A Smart Diagnostic System for Machine Problems fits in this continuous improvement mentality as well as follows an automated and more efficient work method. To be able to step further there is the need to understand how this process works.

3.2 Process study

Adira produces three types of machines:

- Press Brakes
- Shears
- Laser cut machines

All of the above are developed to work with metal, bending and cutting. We don't need to get in much detail about the machines, only the division in smaller components that can be seen in the correspondent tables on Appendix A and how to interpret symptoms to identify malfunctions which is also shown on Appendix A.

Press Brakes represents the biggest impact on Adira sales as well as in services. A majority of maintenance requests are related to this machines and so it requires a bit more attention on the development of this Smart Diagnostic System. But before diving into specifications it is necessary to understand the global and generalized process for a trouble ticket with all the information related to the customer problem and machine in cause. In Adira it is called "Chamado".

A questionnaire was developed in order to understand all the steps needed since the problem is reported until the final resolution. This questionnaire can be found in Appendix A.

3.2.1 Process flow

Through the answers obtained from Adira workers it is possible to understand that three Adira teams can be involved on this diagnostic/resolution process, such as, Back-office (services), Engineer and Ground team. Each one is responsible for a distinct part of the chain of events. The trouble ticket starts always with the back-office which is responsible to create a chamado for that specific case and follow the course of events to assure the clients satisfaction until the closure of the problem.

In a more detailed view, the work-flow for a general problem including all the possible different paths accordingly to specifications is illustrated on Figure A.1 on Appendix A with the aid of Bonitasoft[1]

3.3 Main Challenges

In order to transform Adira's diagnostic system into a smart application it was made a deeper analyses on the current process and defined the better approach for this implementation. Studying event by event it were identified two main challenges to overcome in order to achieve a good performance and to increase processes efficiency. For each one of these problem a methodological thinking was applied presenting a possible solution and how to achieve it. That methodology is presented bellow.

3.3.1 Problem 1 - Problem description in open text format

Solution: Development of an easy closed questionnaire to obtain problem information based on pre-defined parameters.

How to achieve it:

- What is needed to know about the machine?

[1]Bonitasoft is an open source software to do graphic representations of processes - www.bonitasoft.com

    - Serial number.
    - Problem description.

- What to ask to identify problems?
    - Questions are made by Adira's engineers based on experience.
- Build a questionnaire.
    - Retrieve information from engineers to create pre-defined questions.
    - Organize those questions to filter into possible problems.
- Connect the questionnaire to a database to automatically give solutions/instructions for the encountered problem.
    - If the problem is not found/solution was noneffective -> Allow open text input to obtain more specific information about the problem.

3.3.2 Problem 2 - Lack of organized information for analyses

Solution: Development of a database and application for problem/solution mechanism.
How to do it:

- Where is the Data?
    - Data is engineers knowledge.
    - Data is in emails from previous cases.
- Mine the available data and retrieve information from the available sources.
    - Organize the data so it can be used.
- Build a database with easy "input" and "retrieve" of information.
    - Fill the database with the existing data.
- Develop a Web application for "input" and "retrieve".

3.4 Data mining

Adira's emails, sent by the engineers, in response to customers trouble tickets, contained most of the existent data. There was a huge dump of data needed to be analyzed in order to find useful information. By hand it would be impossible to search through it and separate "good" from "bad" data so, for that purpose, it was needed the help of mining technologies.

3.4.1 Used technologies

QDA Miner [2] is a computer assisted mixed-methods qualitative analysis software. This software was chosen due to the capability of filtering, organizing and extracting information from large collections of data. The available dump used was around 13000 emails in .PST format in which it was run a query with the correspondent parameters to extract useful information from Press Brakes and Shears. Since Laser machines correspond to a smaller percentage of Adira's machinery they were ignored for this pilot project development since that data would be irrelevant for improvements.

The following results showed that Press Brakes represented 82,3% of Adira's after sales services over the last years, since this machine also represents the biggest market sales share.

After confirming the importance of Press Brakes on this system the developed software was sketch taking in consideration that this machine should be the priority of the studies once it has the biggest impact.

Also, after this analysis, at the first stage of development and testing, Shears were partially ignored due to a much smaller effect on Adira's after sales service. Even though they are considered on the studies and statistically analyzed to enhance this Smart Diagnostic System in future developments, the focus of the application will be directed to Press Brakes. This focus brings more efficiency on the whole after sales system and it is a better initial approach to explore the impact and success of the application.

Category	Code	Description	Count	% Codes	Cases	% Cases
Máquinas	Guilhotina		728	17,7%	728	5,8%
Máquinas	Quinadora		3382	82,3%	3382	26,8%

Figure 3.1: Data statistics by machine

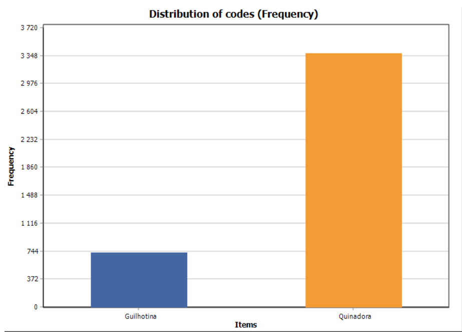


Figure 3.2: Machines frequency graph

3.5 System Structure

The main goal of this project is to create a smart system connected to Adira's server in order to collect and treat data related to machinery problems. Additionally it stores all the occurrences enabling analysis of common issues helping to follow a path of continuous improvement on all the equipments. A more graphic view of this system can be seen on Figure 3.3 below.

[2] https://provalisresearch.com/products/qualitative-data-analysis-software/

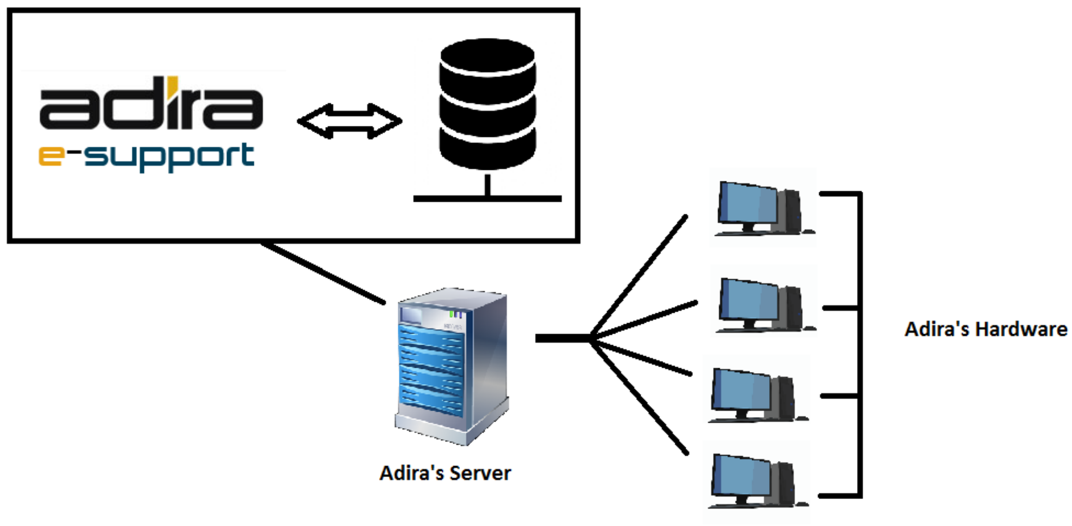


Figure 3.3: Smart Diagnostic System Structure

3.6 Database

3.6.1 Smart diagnostic requirements

In order to store the information used on this system it is needed to structure an adequate database. More details about a database were explained on chapter 2. The first concern to take in consideration is to identify all the data fields needed for this system. Through the analysis of Adira's current process the data is divided in three main stages:

- **Symptom** - Input provided by the client based on pre-defined questions to give information not only about the identification of the machine but also the description of the problem itself specified for the client machine characteristics. Each unique set of symptoms make a **diagnostic** which can be associated to various faults.
- **Fault** - Based on the inputed **diagnostic** the structure has stored a stack of possible faults associated to the symptoms that indicates what machinery part is experience malfunctions.
- **Solution** - Associated to each **fault** the structure has stored a stack of possible solutions that helps the client fixing errors or broken elements.

3.6.2 Database structure

3.6.2.1 Used Technologies

To build the required database it was used a relational database management system in order to ease the developed work and organization of the Structured Query Language (SQL) queries. The chosen software was MySQL. The main points that defined the usage of this software were that there are already more Adira databases using MySQL storage engine (databases which will be used on the application) it is an open-source software and supports JDBC which will be crucial for DB connection in a further point of development.

3.6.2.2 Symptom

The input provided by the client about the machine and the respective malfunction can be divided into different fields. Adira already has a database with the product tree, called Uone. Uone Database is accessed, as seen on Figure 3.4, so that the client can choose his machine from this existent list, provided by Adira's Enterprise Resource Planning system to avoid data duplication inside the company. Besides this, the user needs to input information about the existence of an alarm, the location of the problem (divided into system, module and component as seen on Table A.1 for Press Brakes and Table A.2 for Shears) the operation mode the state of work of the machine and a brief description of the existing problem.

For each one of these fields there is a correspondent table on Adira Smart Diagnostic System database which contains pre-define values for selection. Since this values will need to be motorized and updated in order to keep track of every change and avoid "bad" inputs, those tables also record data related to the creation and edition time of this fields and the creator or editor username.

A unique combination of all of this symptoms results in a diagnostic of the machine. This diagnostic is saved as a "temporary" problem in "tempdiagnostic". Figure 3.4 shows the graphic representation of this database section.

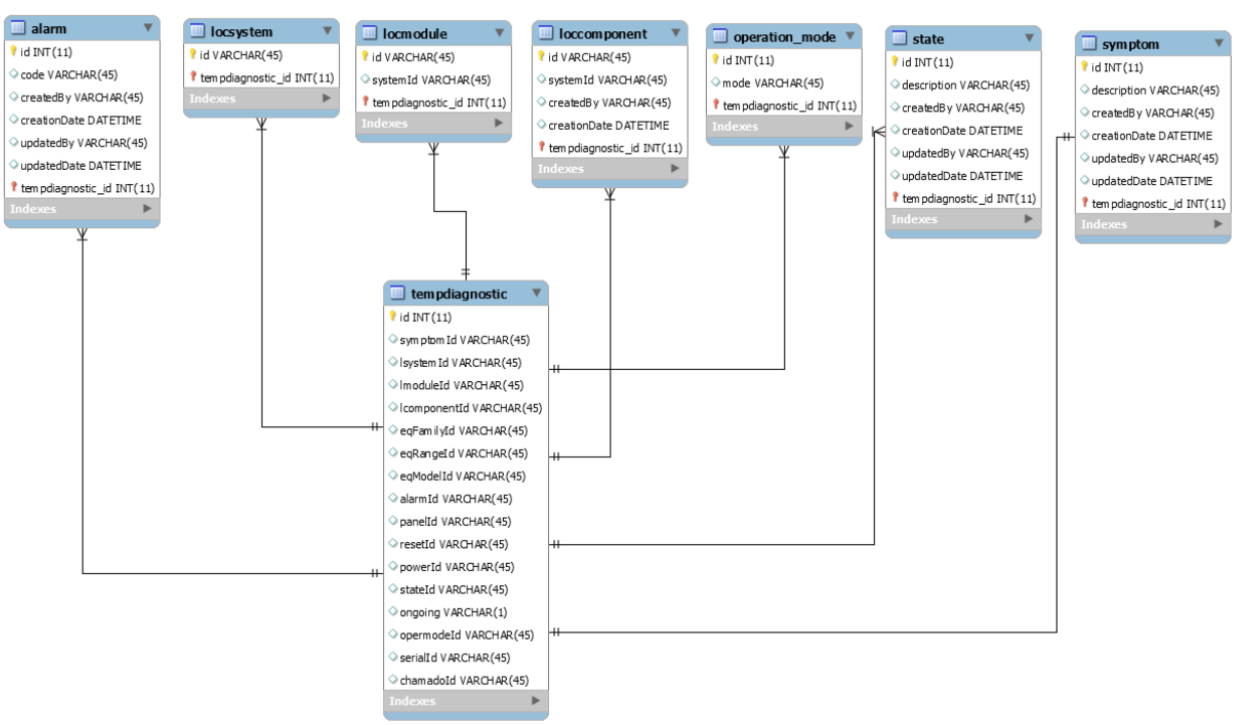


Figure 3.4: Database structure for symptom parameter

3.6.2.3 Diagnostic

The "temporary" diagnostic created based on the input of symptoms is compared to the existent table filled with possible diagnostics that have already a fault associated to them. The "diagnostic" table is inputed on the system by Adira's engineers while solving new problems that come up. Once this input is made the smart diagnostic system will be able to provide these solutions automatically for future occurrences of the same symptoms. Figure 3.5 shows the graphic representation of this database section.

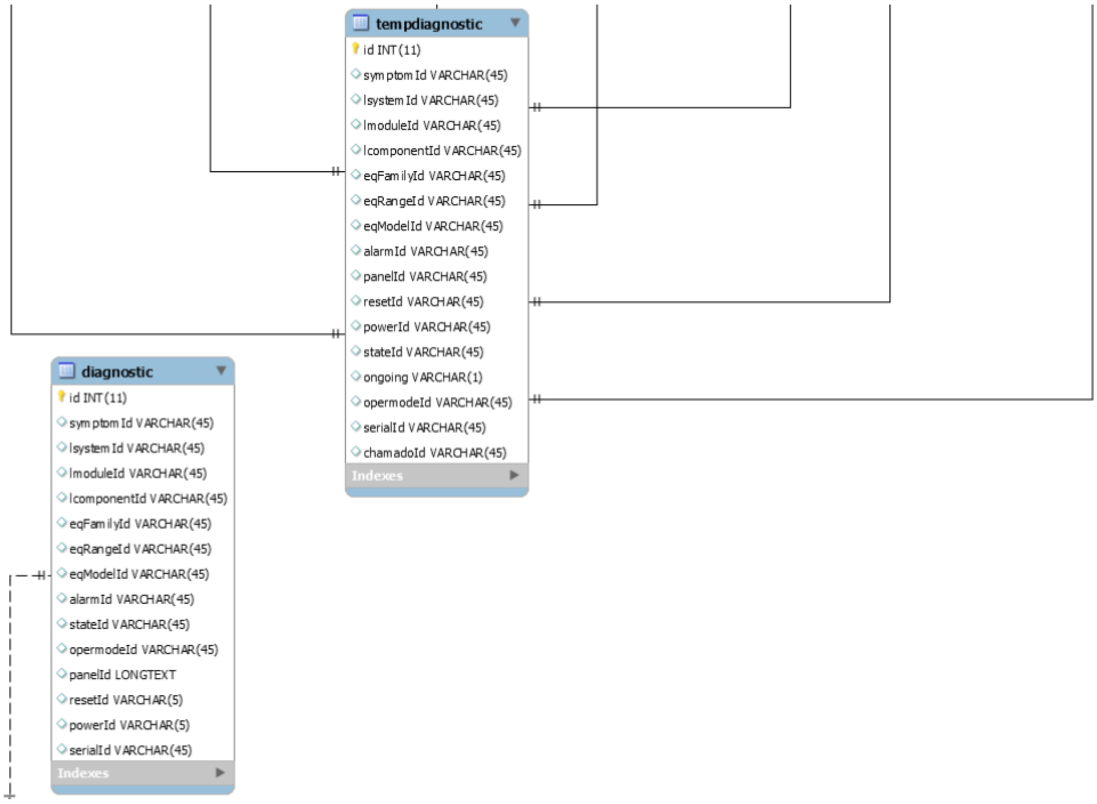


Figure 3.5: Database structure for diagnostic parameter

3.6.2.4 Fault

The unique set of symptoms (diagnostic table) has a many-to-many relationship with a faults table, which means that one diagnostic can be associated to many possible faults and one fault can have more than one different diagnostic.

For this table, besides information about changes and the user who did them (as seen on previous tables) it was also created a mechanism to avoid duplication of data. Since each fault can be described in various different ways each one of them has two keywords associated in order to filter and prevent this encountered trouble. Another table is attached to this system to contain keywords information and association to faults. Figure 3.6 shows the graphic representation of this database section.

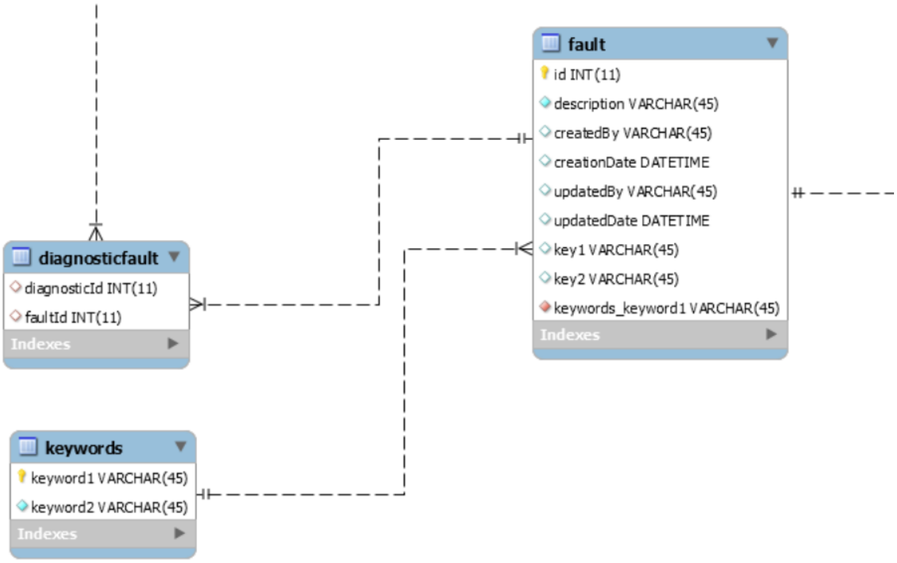


Figure 3.6: Database structure for fault parameter

3.6.2.5 Solution

The previously presented fault table has a many-to-many relationship with another table called "solution". This table contains data associated to solution procedures for problem solving. Since each fault can have many possible solutions that need to be tested by the client, it was implemented a functionality to present the solutions in a structured and ordered way. The first presented solutions are the most common and the easiest ones to implement by the user. This kind of organizational method allows the problem to be solved as easiest and fastest as possible. Figure 3.7 shows the graphic representation of this database section.

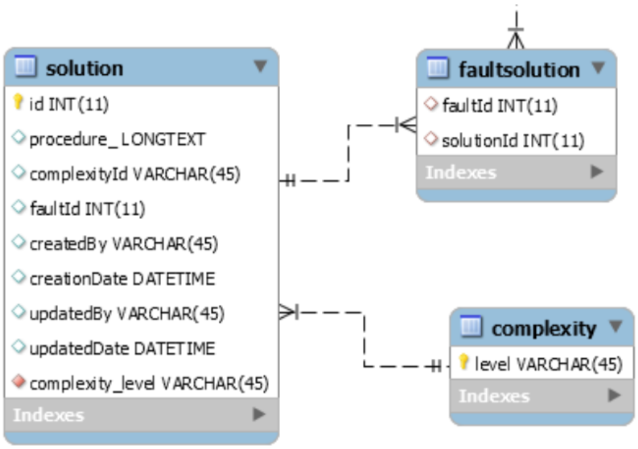


Figure 3.7: Database structure for solution parameter

3.7 Code structure

The developed code was structured following both programming standards as well as company standards. Five different packages were created, that has information related to classes specific to the name of the package. Four most important packages are described in the next sections. Figure 3.8 shows a graphic representation of the packages hierarchy model.

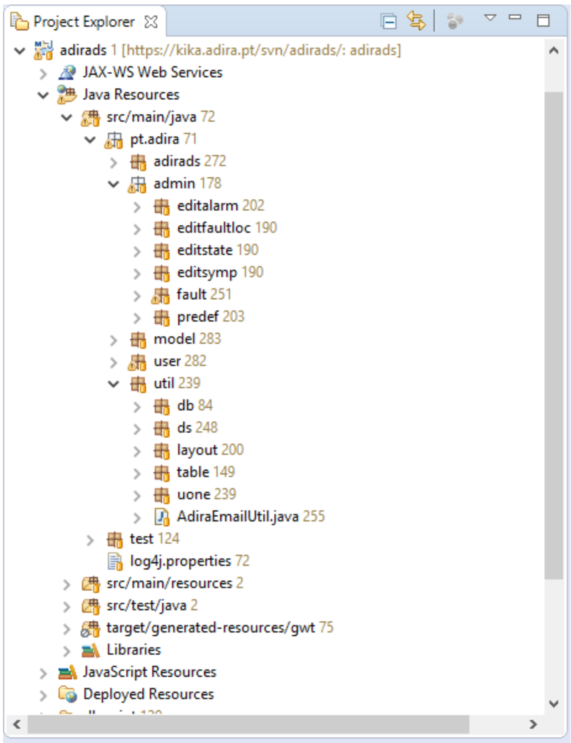


Figure 3.8: Eclipse package hierarchy

3.7.1 User Package

In addition to other classes, the user package consists of three classes which call each other and enable the user to navigate through the different layouts. MainView class, which is responsible for the input of information, SolutionView class that is responsible for output of information, and also, SupportView class which redirects the user for the possibility of more inputs in order to help with more complex problems.

The MainView class layout is filled with combo boxes for the easy input of information which is pre-defined on the system. Having the input of information in this format solves one of the biggest challenges for a successful implementation of a Smart Diagnostic System. Data analysis is the main characteristic and focus of the application, until the development of this software all the trouble tickets were reported directly in open text format. To make this analysis possible all the information input was converted in standard and well defined characteristics that provide a complete set of details that are needed for a successful analysis. This page layout code consists of horizontal layouts of combo boxes that have direct access to the database to give options of choice to the user. Once the input of information is done it will call the class methods which will be explained in section "utility package".

The SolutionView class consists only the output information as no algorithms are run directly in this class. Besides the displayed output, this class is responsible for the organization of the information. Since every solution has associated to it a complexity level it was implemented a method to read the strings of information contained on the system database, get the string complexity ID and organize it on the view panel that is presented to the user. There are also two buttons to get the information from the user if the presented solutions were able to solve the problem or not. If the problem was solved, there is a navigator to redirect to the main page. On the other hand, if the problem wasn't solved at all, there is a navigator redirecting to the SupportView.

The SupportView is, once again, input of information but this time it consists of an open text area. Once the user navigates to this page, since it will provide a direct support system, it will allow the user to input some more information on text format to help on the diagnosis. On this class it was also implemented code to send an automatic email to Adira service with all the detailed information chosen on the combo boxes of the main page, as well as information about the client and the additional information about the issue, so that the case can be easily analyzed and reported.

A Lights Panel Layout page that represents a replica of the lazersafe LEDs panel. This page is currently disabled due to complexity of data input but it is being studied for future improvements on the application. It consists of a dynamic matrix of check boxes used to determinate which LEDs are on and which LEDs are off. One of the main ideas for this page is the automatic fill by image recognition. It would require an upload of a picture and then the algorithm would read from it and save it on the already created matrix string. To enable this characteristics it is needed the development of an independent project, due to the high complexity of this analysis.

3.7.2 Admin Package

In the admin package, different sub packages were created, for each one of the existent tabs of the software. This packages has a data container class that was used to create the tables that show the selected information to the user and also a data view class that was used to create all the interaction mechanisms of the table (add, delete and edit) as well as information pop-ups, display of forms and how to store the information.

The tab responsible for the input of new faults and the correspondent association to solutions and diagnostics had a much more complex implementation than the other remaining tabs. To create this structure it was also created a data container and a data view, but this time associated to a keyword mechanism and with two implemented tabs inside the data view table.

The keyword method forces the user to check for duplicated data which prevents the input of the same information described in different words. This is a simple implementation that has a major impact on the software and its performance.

After filtering by keywords, this fault table, has a click listener implemented so that it can be stored the ID of the selected fault in order to associate information related to it by parsing it through "Add Symptom" and "Add Solution".

By clicking on "Add Symptom" button it will navigate to a new tab which has the same information and displayed in combo boxes replicating the user main view. This serves as permanent input of information. On the user main page the data about the machine diagnostic and characteristics is saved on the "tempdiagnostic" table which will be a temporary storage to enable the algorithm to do comparisons between the actual diagnostics table. On this "Add Symptom" page, the input of information is stored in a permanent table and it represents the actual diagnostics associated to a specific fault that was parsed by its ID to this navigation view.

On this page we also have another difference. There is a new combo box that displays the chamados references that are stored on Adira's ERP. By selecting and applying a chamado reference the code will retrieve information associated to that chamado and run a method to automatically fill all the combo boxes with the information already available and related to that trouble ticket.

By clicking on "Add Solution" button it will navigate to a new tab which consist in a data container class and a data view class. Following the same structure it will display to the user a table with all the existent solutions with an associated click listener so that is possible to associate a solution ID to the selected fault ID. Also the table has the implemented code of adding new elements which opens a form that the user inputs the complexity level and the steps to solve the issue. On the written text with the step by step solution each sentence starts with the special character ">" in order to be used on the code run on Solutions View.

Through the parsed ID on this navigation class it will associate the created or selected solution to the previous selected fault. Those ID's are stored in an independent table to establish a many to many relationship.

3.7.3 Model package

The model package consists of different classes to assign the getters and setters for every object of the Smart Diagnostic System. This characteristic is determinant for program development once it brings many advantages. Getters and setters allow the software to have encapsulation of behaviors to control access levels without affecting the public interface. It also helps for memory management, provides a better debug system, and many other advantages.

3.7.4 Util package

There is also a package focused on utility matters of the software to assure programming standards. This package contains different inner levels for different applications.

Two classes were created to connect to the databases. One class has the code to connect to the Database created specifically for this software system and another class to connect to the "Uone" Database. Uone Database is an existent Adira Database that have the data of the machine models.

Instead of replicating this data on another table the Smart Diagnostic System is connected to this database to always be updated with the existent models of machines.

A sub package was created for SQL queries that have all the classes that contain the code for CRUD operations (Create, Read,Update and Delete). This is the most important work developed and contains all the heuristics of this Smart Diagnostic System for Machine Problems. This package needs a more detailed explanation that will be explored class by class.

- AlarmDS class - Contains SQL queries to insert, update, delete and get alarms details.
- ComplexityDS class - Contains SQL queries to insert, update, delete and get complexity levels.
- FaultDS class - Contains SQL queries to insert, update, delete and get fault descriptions as well as a function to get faults by keywords and to get faults by associated unique ID.
- DiagnosticDS class - This class has the code for the static algorithm to run a search mechanism on the Database. It contains a SQL query to insert each one of the inputs provided by the client on the diagnostic table. It contains a query to retrieve a list of diagnostics ID's and a query to retrieve a diagnostic ID by comparison with "tempdiagnostic" table.

  The developed code responsible for searching firstly identifies the ID of the tempdiagnotic that was inserted by the client. The temporary diagnostic is nothing more than a row of this table that contain a value for each field that can be NULL or a string. The algorithm checks, on the diagnostic table, if there is any matching row with the selected tempdiagnostic row. If there is no matching rows, the algorithm will start checking for rows with NULL values on the diagnostic table that might be not NULL on the tempdiagnostic table. This will provide a searching mechanism that will find the faults associated to diagnostics based on a percentage of approximation. For the test phase this percentage is set to 100% which means that the diagnostic inserted by the user must match completely to the software database. This will result in less derivations on the actual machine malfunction.
- DiagnosticFaultDS class - Contains SQL queries to insert both diagnostic ID and fault ID in order to enable a many to many relationship between them. Also a SQL query to delete both diagnostics and faults from this table as well as a query to get a list of fault ID's based on diagnostic ID's.
- FaultSolutionDS class - Contains SQL queries to insert both fault ID and solution ID in order to enable a many to many relationship between them. Also a SQL query to delete both solutions and faults from this table as well as a query to get a list of solutions ID's based on fault ID's.
- KeywordsDS class - Contains SQL queries to insert, update, delete and get the different keywords associated to machine faults. Since this system uses two keywords it was developed a query for each function to each keyword in order to make them independent. It is possible for a fault to be attached to only one keyword or, if needed, two keywords.

- LocComponentDS class - Contains SQL queries to insert, update, delete and get the different components of an Adira machine. The get function for components is dependent on the machine modules since it is a sub-level of this concept. The query will automatically filter the component based on the selected module.
- LocModuleDS class - Contains SQL queries to insert, update, delete and get the different modules of an Adira machine. The get function for modules is dependent on the machine systems since it is a sub-level of this concept. The query will automatically filter the module based on the selected system.
- LocSystemDS class - Contains SQL queries to insert, update, delete and get the different systems of an Adira machine.
- OperationModeDS class - Contains SQL queries to insert, update, delete and get the information about operation modes of Adira's machines.
- SolutionDS class - Contains SQL queries to insert, update, delete and get solutions for machine problems. This class also contains a query to get a list of strings with step by step solutions ordered by complexity levels.
- StateDS class - Contains SQL queries to insert, update, delete and get states of work related to the machines.
- SymptomDS class - Contains SQL queries to insert, update, delete and get symptoms descriptions of machine malfunctions.
- TempDiagnosticDS class - Contains SQL queries to insert every field of this table that is stated on the Database structure as well as the insert of chamado references and ongoing state (which was defined as a binary variable to check if the trouble ticket is being solved or is already solved). This class also contains queries to individually get the information on every column of the table by using the temporary diagnostic ID, a query to get a string of a full diagnostic, a query to update rows of diagnostic information if the user made any mistake while giving the data to the Smart Diagnostic System and a query to delete rows.

Finally there is another sub-package with all the classes associated to the tables that display information to the user. The tables have implemented code for filtering and searching elements, buttons to add edit and delete information, forms for input, confirmation and warning dialogs, appearance (such as icons) and indexes to define sizes.

Chapter 4

Smart Diagnostic System Web Application

This chapter describes all the work developed to create a web application capable of input and output of data and explains all of the implemented functionalities on it. Firstly it is presented with all the used technologies and the correspondent characteristics that determined those choices. Secondly all the interface is explored through a user point of view with detailed explanation of the work mechanisms of this system. Finally all the acquired data on this test phase is analyzed in order to study frequent problem occurrences, the impact of the software on Adira's current work methods and to identify how to continue the improvements and developments for this service to achieve the most success and investment return.

4.1 Used Technologies

The work developed on the web application required some software technologies that needed to be chosen accordingly to Adira specifications. The used IDE (Integrated Development Environment) was Eclipse Mars. It was chosen because of previous knowledge on this tool and also because of the versatility that it can provide. The programming language for the software development was Java. Since other company projects are developed on Java 7 and based on Apache Tomcat servers, the same technologies were used to facilitate compatibilities and to follow the company standards. It was also used Vaadin 7 as framework for this Java project and JDBC classes to implement database connections.

4.1.1 Eclipse

Eclipse is an open-source community tool that provides an Integrated Development Environment for many programming languages and architectures. For the development of this project Eclipse Mars for Java EE Developers was used. It is a package that contains Java associated tools to create Web applications as well as data tools and eclipse SVN to facilitate team work inside the company and to provide a status of the work progress in real time.

4.1.2 Tomcat Server

Apache Tomcat is an open-source software that is used to execute Java servlets - object that receives and responses to generated requests - to have the capabilities of a server.

When a Web application is developed it should be accessible through Internet or Intranet. To provide this access it is needed a server which, in this case, is Apache Tomcat. Tomcat is built by three main components: **Catalina -** Tomcat's servlet container, **Coyote -** Tomcat's connector to support HTTP protocols and **Jasper -** Tomcat's JavaServer Pages engine for compiling and recompiling files.

Tomcat Apache Server was chosen because it was already used inside the company.

The server working scheme is illustrated on Figure 4.1

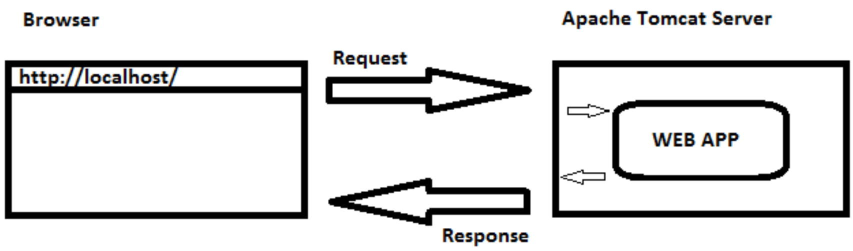


Figure 4.1: Apache Tomcat Server Scheme

4.1.3 Vaadin

Vaadin is an open-source Java framework. This tool enables the development of Web content using Java as the programming language. It is a server-side architecture that uses Ajax on the client side to enhance behavior and data interaction.

Vaadin was used to limit the amount of technologies involved, such as JavaScript and HTML and to decrease the need to program elements from scratch that were not the goal of this project.

4.1.4 JDBC

Java Database Connectivity is an application programming interface that defines the access to a database. This interface provides pre-implemented methods to create and execute statements and it was used to connect all the needed data to the created software in order to exploit the "WORA - Write once, run anywhere" benefits of Java.

4.2 Client Interface

The developed application is divided into two main menus, User Menu and Administrator Menu. The interface is detailed explained in the following pages and consists of a simple and intuitive graphical user interface to input data and to manage database access.

4.2.1 Main page

At the Main page of the Smart Diagnostic System the client is requested to provide information related to the machine characteristics, status of work, problems and output information from the machine sensors.

This page consists of all the data input so that the software can store problems associated data and run the algorithm to identify solutions for the inputted issues. This interface can be seen on Figure 4.2.

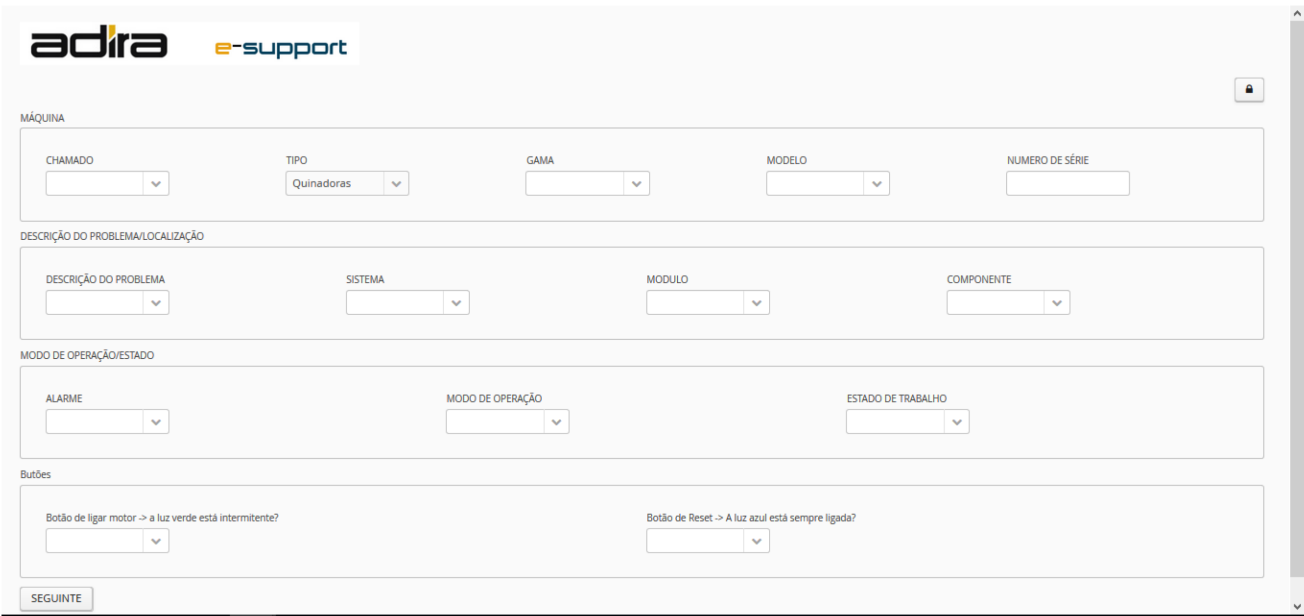


Figure 4.2: Main Page

4.2.2 Step by step solutions Page

After the input the software will save the temporary diagnostic in "tempdiagnostic" table on the existent database and then it will run a static algorithm search through the "diagnostic" to filter possible faults based on the provided symptom. Once it finds any matches it will scan the database to find solutions for the presented faults.

After finishing this step, it will redirect the user to a page that will present two fields of information:

- Faults - Presenting all the possible faults associated to the provided symptoms.

- Solutions - Presenting step-by-step solutions to solve any existent problems.

It was also developed a way for this software to present the information in a user friendly way. The solutions are ordered by complexity and probability of success so that the problem can be solved as easy as possible and in the less time consuming methods.This interface can be seen on Figure 4.3.

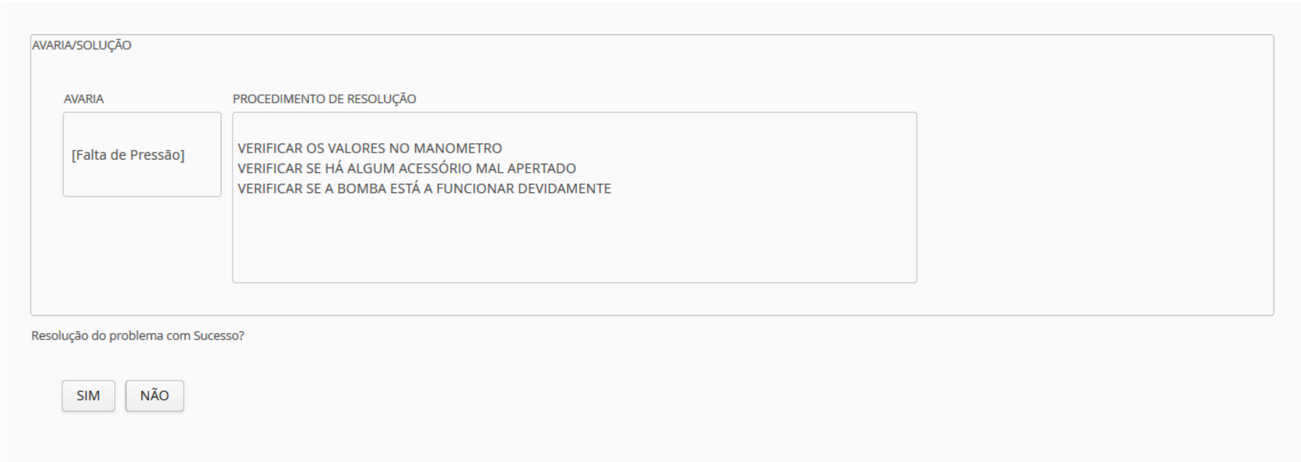


Figure 4.3: Step by step solutions Page

4.2.3 Adira Support Page

If the Smart Diagnostic System can't find any faults associated to the provided symptoms it will redirect the user to the creation of a new ticket to Adira's direct support. This feature also comes with the possibility of adding additional information on text format (once it will be analyzed directly by an Adira engineer) to ease the understanding of the problem. Once this fault is diagnosed and solved by Adira's engineers, if applicable, it can be inserted in the system database and be used to identify future situations. This procedure will be explored at the database input interface on section "Admin - Symptom page". The interface for direct support can be seen on Figure 4.4.

4.3 Admin Interface

Adira's production is in constant innovation which requires the need of a service that can keep up the pace. When the Smart Diagnostic System was sketched it was idealized an administration interface to update and create new features in the software without the need of an IT expert to change the developed code. This section was developed for Adira internal use to fill the requirements of products innovation and development.

This section of the software is used to input solutions, symptoms and faults to the algorithm database and also to update machine information and characteristics.

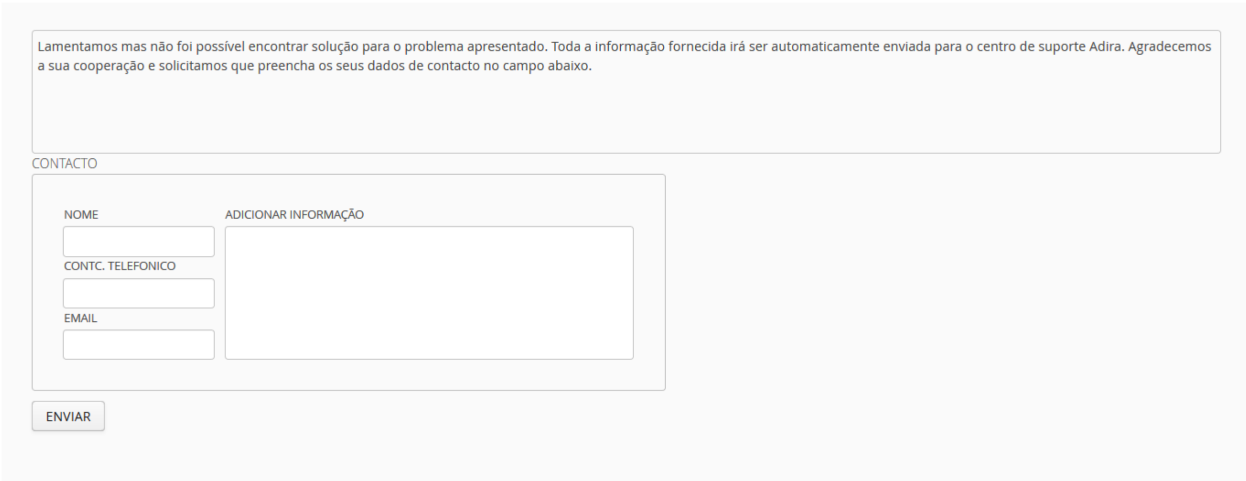


Figure 4.4: Adira Support Page

4.3.1 Login page

To access the administration section of the developed software it was created a log in screen in order to prevent unwanted users to change back-end details that are not implemented on a daily bases and are only changed by skilled engineers. This page interface can be seen on Figure 4.5.

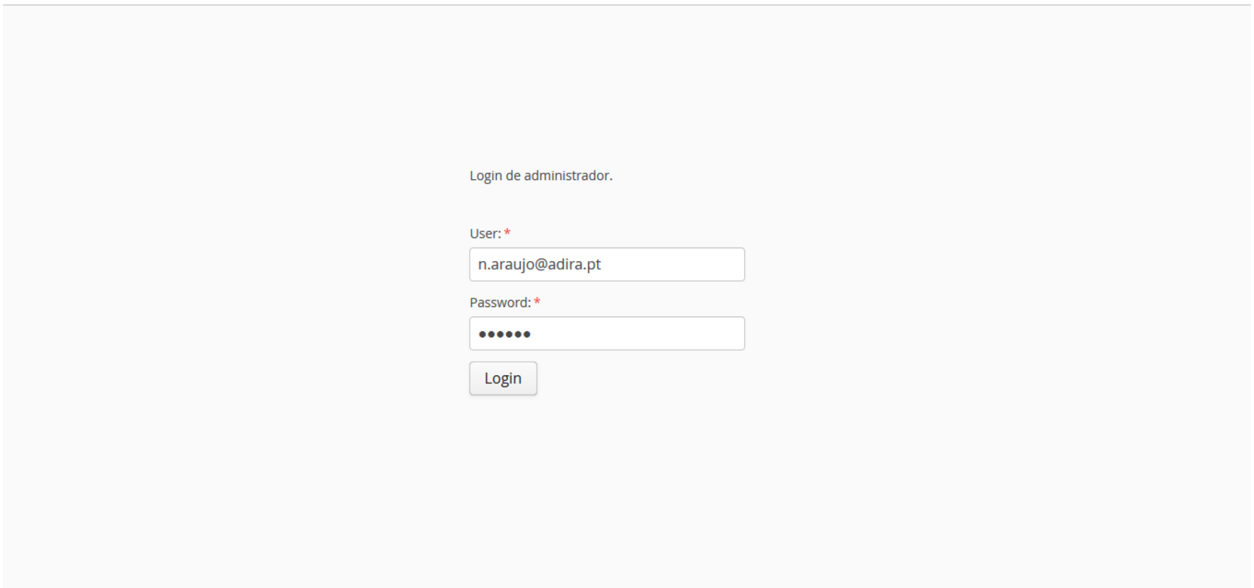


Figure 4.5: Login Page

4.3.2 Admin - Fault page

The first screen accessible after login is the Fault Page. On this page it is possible to view all the faults on Diagnostic System database organized by keywords. The keyword method was created to prevent duplication of data, it filters problems by different fields and displays them in an organized table. This feature is one of the most important factors on this software since the duplication of data is bottleneck of the system.

This is the main page related to back-end because it is the interface for insertion of new faults on the database, new solutions (either for new or existent faults of the database) and new sets of symptoms associated to existent faults. This page main function is to input new faults in the existent database organized be keywords and this interface can be seen on Figure 4.6.

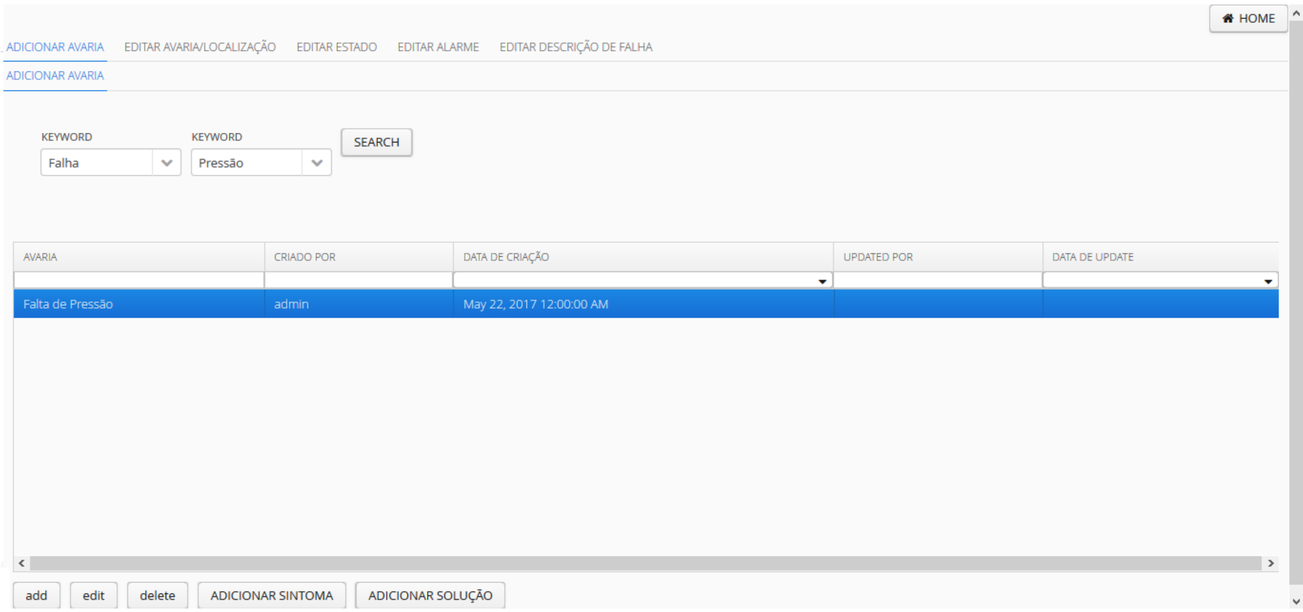


Figure 4.6: Admin - Fault page

4.3.3 Admin - Symptom page

Every fault on the system needs at least a set of symptoms associated to it in order to be useful data. This page is a replica of the user inputs and all the fields can be manually inserted by an engineer that needs to add a new diagnostic for some specific fault or, if there was a ticket that had ended with no automatic resolution and went for Adira's direct support, after solving this issue with the client it can be automatically inserted on the database calling the chamado number. After solving the chamado the responsible engineer can come to the application and input all the information that he needed to solve the ticket, using the automatic feature to get, with the chamado number, the diagnostic provided by the client in the first stage, saving time and preventing human failures on data insertion. This interface can be seen on Figure 4.7.

4.3.4 Admin - Solution page

Following the same scheme explained before, each fault needs to have an associated solution, otherwise it doesn't make sense to find a fault on a machine that has no information about it. This page allows the association of a solution to an existent fault or the input of a new solution if needed. The solution page allows an engineer to organize the inputted solutions by levels of complexity to

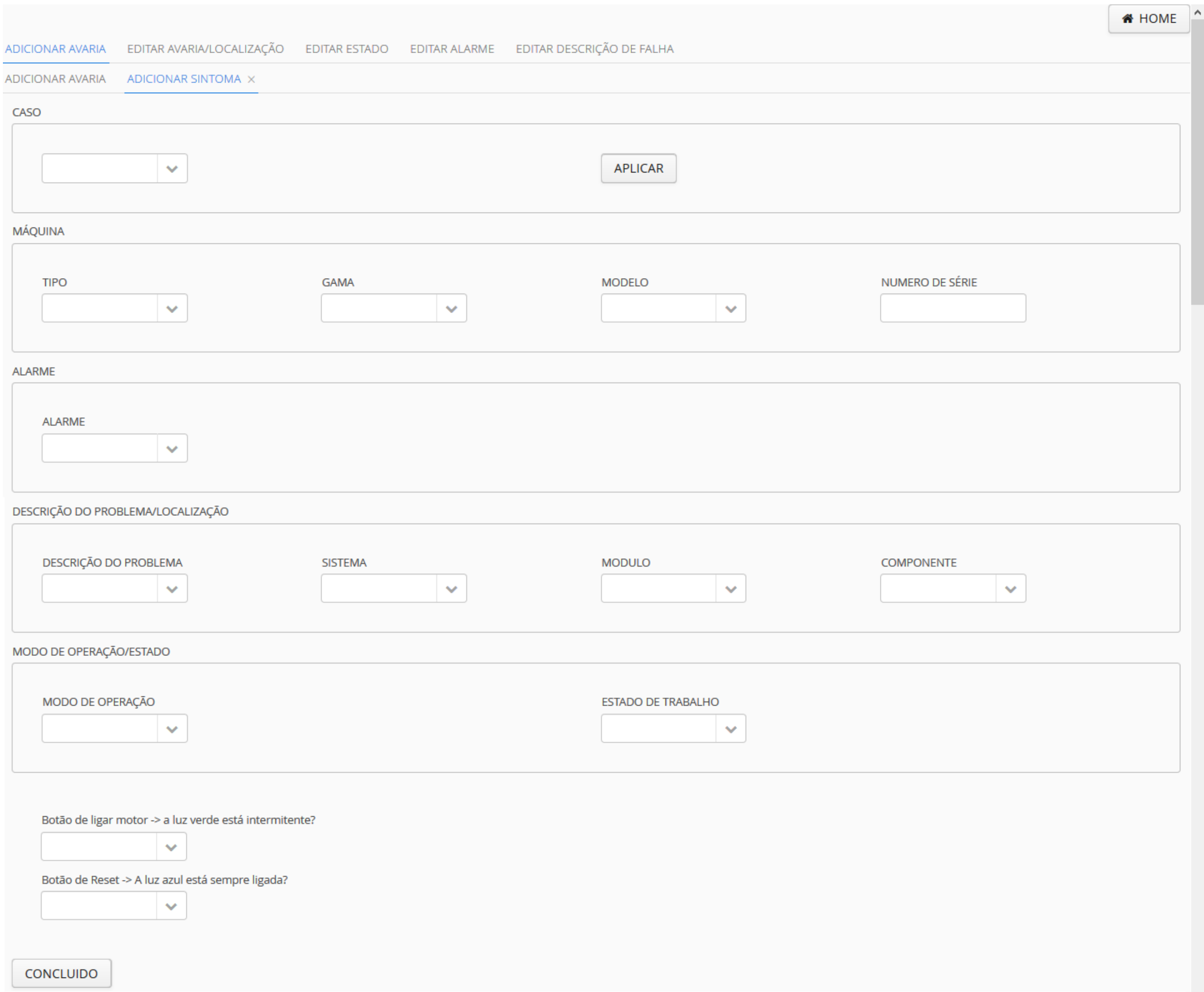


Figure 4.7: Admin - Symptom page

facilitate the resolution of problems and to enhance the functionalities of the developed algorithm. This interface can be seen on Figure 4.8.

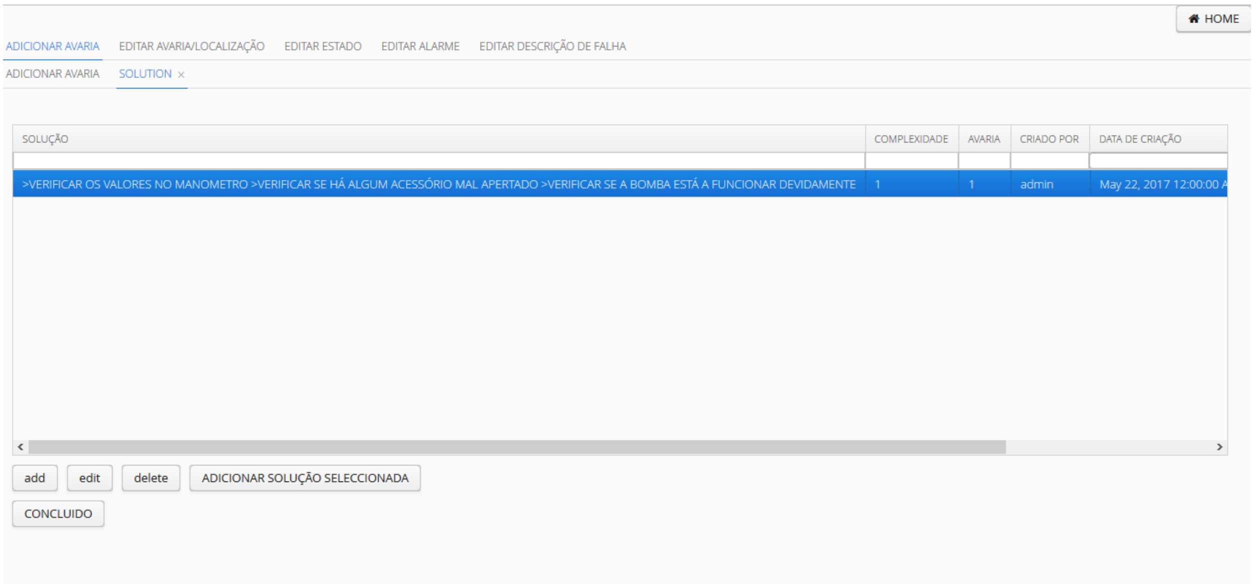


Figure 4.8: Admin - Solution page

4.4 Edit Database

Not only new faults and correspondent solutions need to be inserted on the system but also with bigger innovations comes the need to update database fields such as: new alarms, new sensors, new machine elements and so on. It was developed a structure to face this issue and to enable an easy access to all the tables of the created database.

Since this section of the software requires log in to access all the changes on the database related to the edition or insertion on this fields will have also stored the information about the user who did those changes as well as the time associated to those modifications in order to keep track of all the major database changes.

4.4.1 Admin - Edit Locations page

This page is used to create and update machine sections in case of a major development in one of Adira's machines. It is possible to filter machine components be system and module and then it shows up a table to enable the modifications and insertions of new components. This interface can be seen on Figure 4.9.

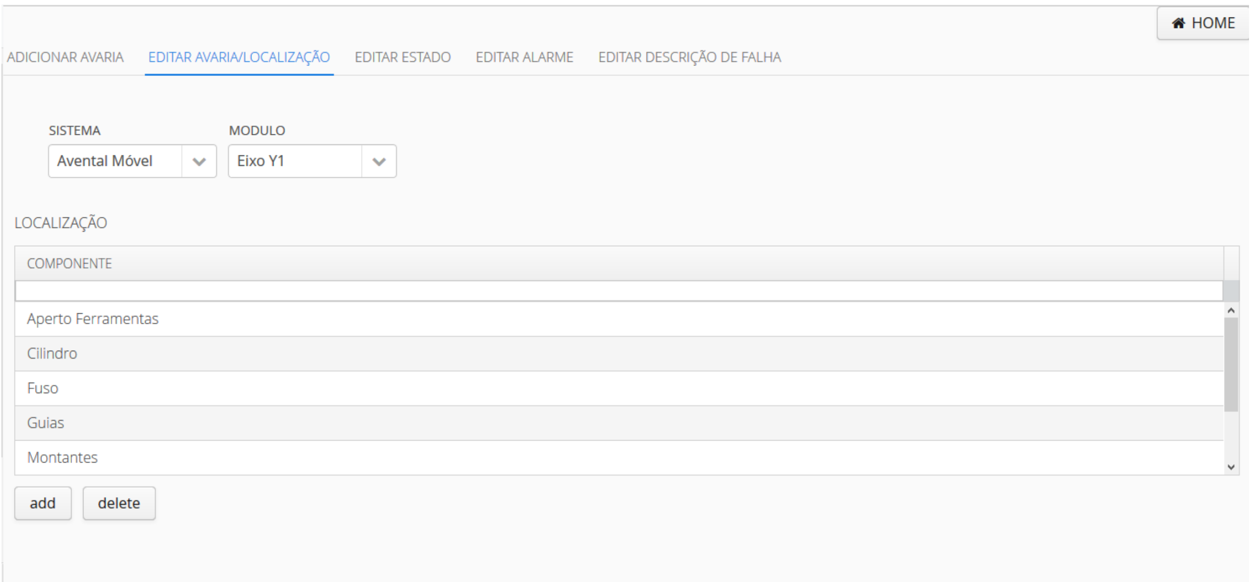


Figure 4.9: Admin - Edit Locations page

4.4.2 Admin - Edit State page

This page is used to create and update machine states of work in case of an implementation of new performances methods in one of Adira's machines. It will present to the user a table with all of the current states of work that the machine can provide giving modification accesses. This interface can be seen on Figure 4.10.

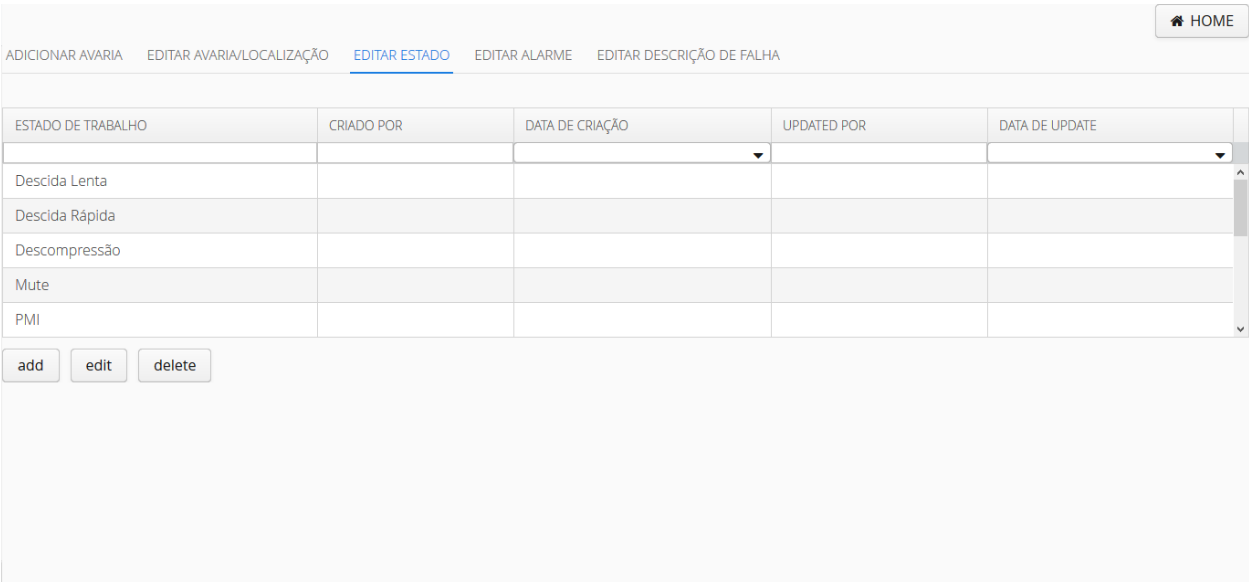


Figure 4.10: Admin - Edit State page

4.4.3 Admin - Edit Alarm page

This page is used to create and update alarms in case of new failures detected by machine sensory analysis. It will present to the user a table with all of the current alarms that the machine can display giving modification accesses. This interface can be seen on Figure 4.11.

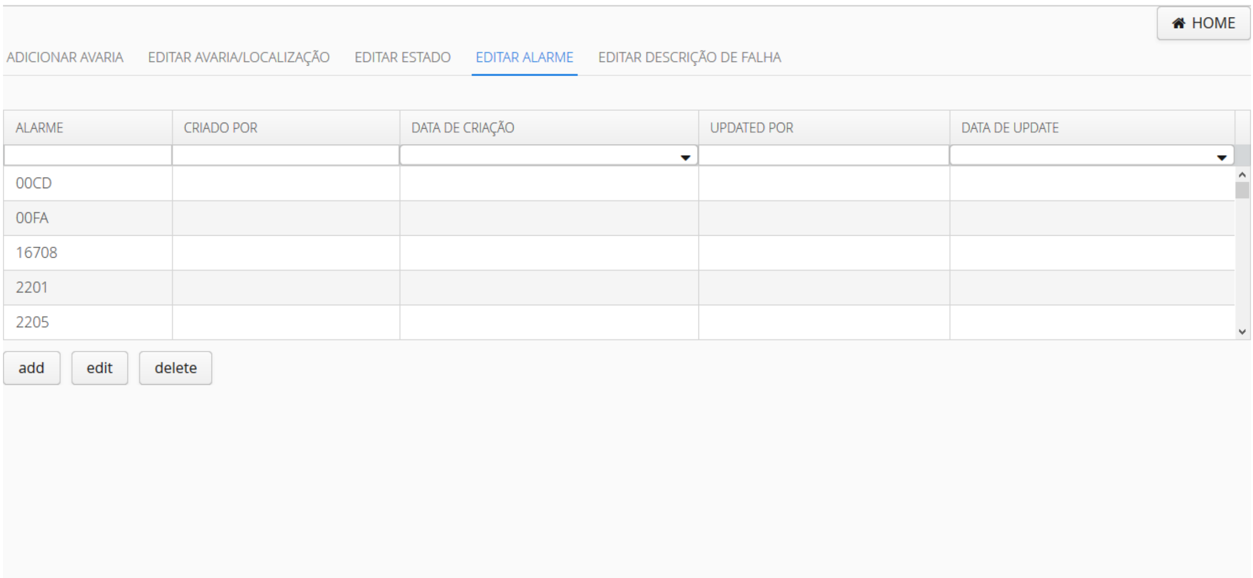


Figure 4.11: Admin - Edit Alarm page

4.4.4 Admin - Edit Fault description page

This page is used to create and update fault descriptions in case of new machine problems are detected. The problem description is a simple sentence of generic information about the malfunction of the machine. This page presents to the user a table with all of the current descriptions on which the client can choose to describe the malfunction and provides modification accesses. This interface can be seen on Figure 4.12.

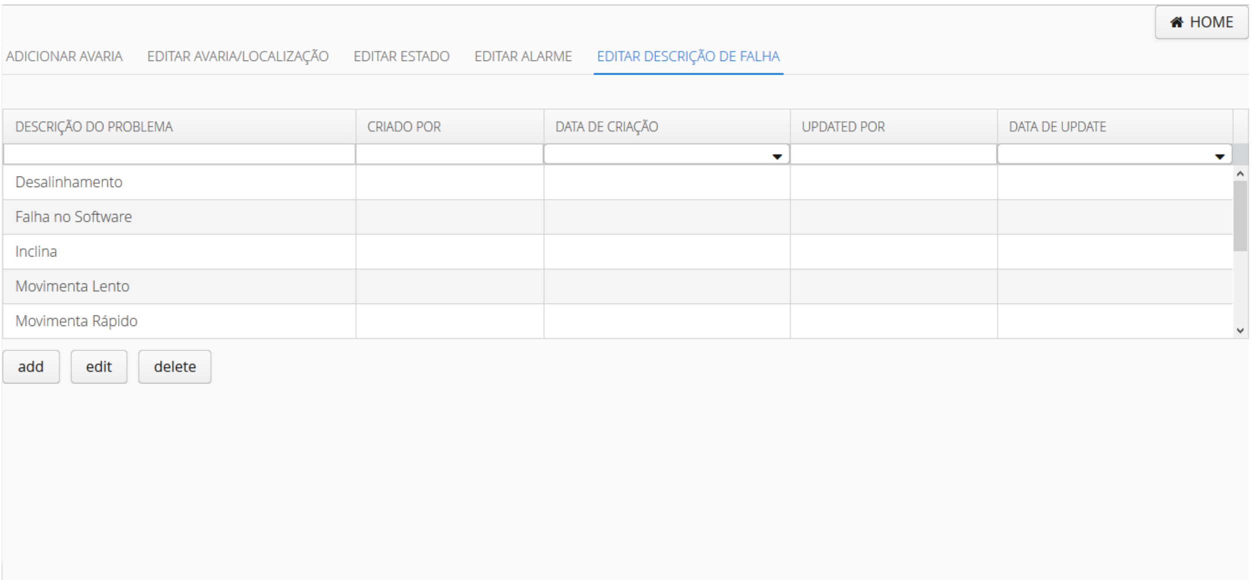


Figure 4.12: Admin - Edit Fault description page

4.5 Conclusion

4.5.1 Existent Data

Through the development of this project it was collected Data by using different methods, platforms and scenarios. On Figure 4.13 it is possible to analyze how the studied and used data were collected to fill the database and to make the software algorithm functional.

On this Figure it is possible to identify the still great potential of database growth due to having a solid percentage of issues that are currently being solved to input on Smart Diagnostic System Database. Through the test phase this factor have been decreasing over time which represents the success of the application. Less tickets to be solved means that the software is taking care of them automatically.

4.5.2 Data analysis

Every ticket sent by clients to Adira requires the creation of a "CHAMADO" which contains all the information related to that case in Adira's platform. During the test phase of the Smart Diagnostic System for Machine Problems, all the chamados related to hardware or software issues

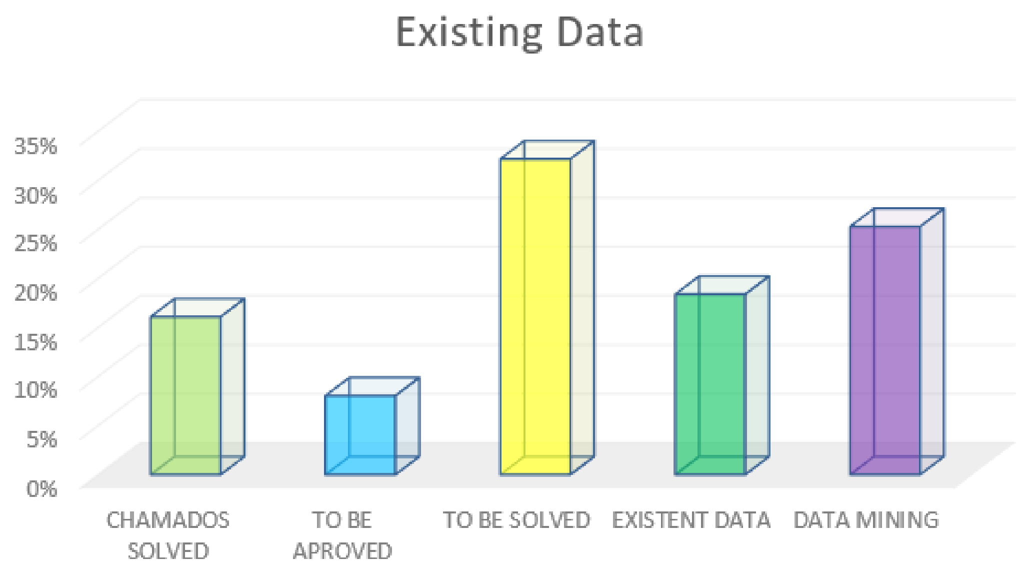


Figure 4.13: Collected Data distribution

on the machines were inputted on the application. Through this work method it was possible to analyze these tickets information concluding that 52% of chamados are related to Press Brakes problems, only 10% to Shears and 38% for other cases, such as hardware budget requests, instruction manuals requests, advices for new work methods implementations, etc. Figure 4.14 represents the distribution of the chamados by subject matter.

On Appendix A in Data Analysis section there is a sample of chamados used for this analysis. This sample is representative of the software studies and follows the same structure as the stored values on our database.

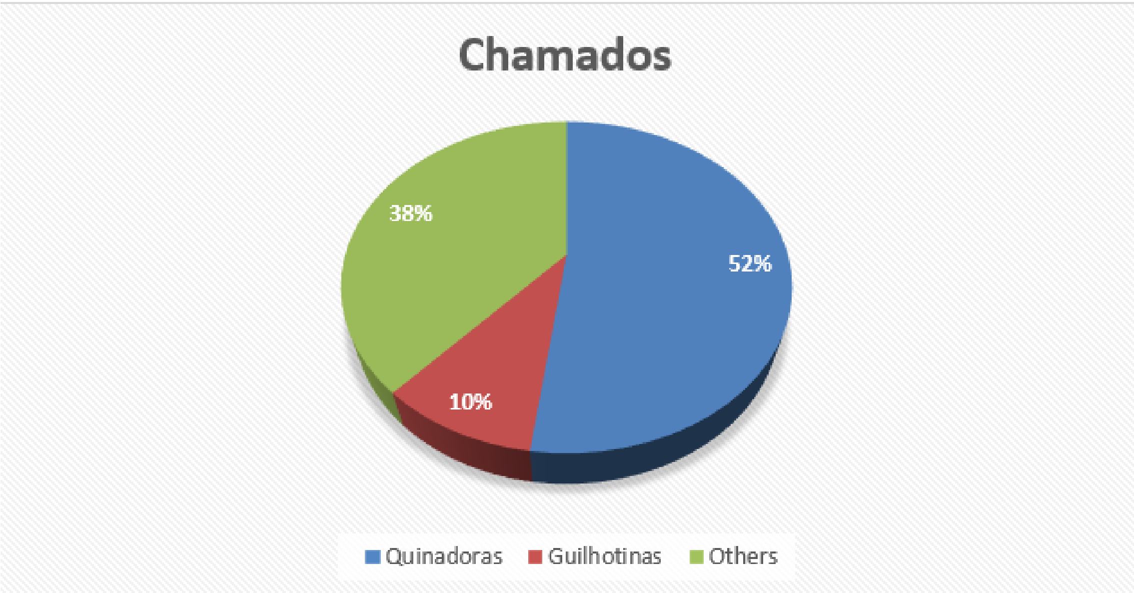
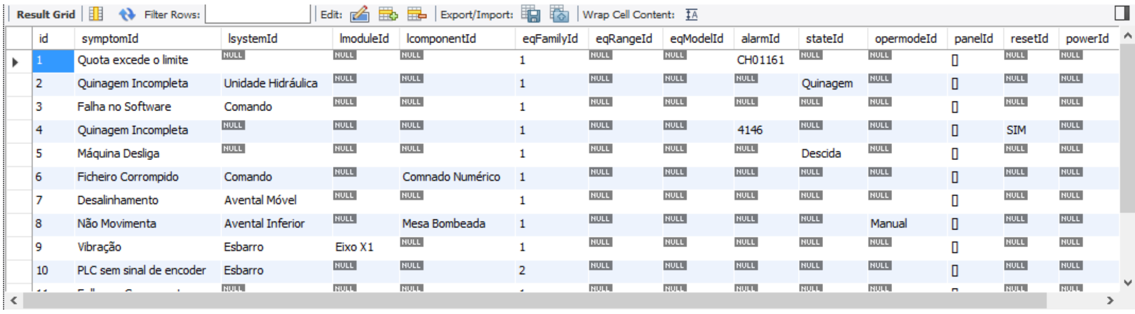


Figure 4.14: Data analysis

4.6 Smart Diagnostic System Database data collection

MySQL Workbench allows us to have a visual interface to analyze the existent data collection. On Figure 4.15 it is possible to see the stored data sample on the database structure.



id	symptomId	lsystemId	lmoduleId	lcomponentId	eqFamilyId	eqRangeId	eqModelId	alarmId	stateId	opermodeId	panelId	resetId	powerId
1	Quota excede o limite	NULL	NULL	NULL	1	NULL	NULL	CH01161	NULL	NULL	▯	NULL	NULL
2	Quinagem Incompleta	Unidade Hidráulica	NULL	NULL	1	NULL	NULL	NULL	Quinagem	NULL	▯	NULL	NULL
3	Falha no Software	Comando	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
4	Quinagem Incompleta	NULL	NULL	NULL	1	NULL	NULL	4146	NULL	NULL	▯	SIM	NULL
5	Máquina Desliga	NULL	NULL	NULL	1	NULL	NULL	NULL	Descida	NULL	▯	NULL	NULL
6	Ficheiro Corrompido	Comando	NULL	Comnado Numérico	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
7	Desalinhamento	Avental Móvel	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
8	Não Movimenta	Avental Inferior	NULL	Mesa Bombeada	1	NULL	NULL	NULL	NULL	Manual	▯	NULL	NULL
9	Vibração	Esbarro	Eixo X1	NULL	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
10	PLC sem sinal de encoder	Esbarro	NULL	NULL	2	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL

Figure 4.15: Database - data collection

4.7 Detailed analysis on Press Brakes

Adira's strongest market is based on Press Brakes sales and provided services. In this scenario it was already predictable that most tickets would be related to those machines.

A more detailed study was made in order to focus on this market once it represents more than half of Adira's total services. As seen on Figure 4.16 it were identified the machine sections that represent the biggest share of malfunctions. These data cannot be shown on this chart due to terms of confidentiality. This studies are very important for Adira's continuous improvement by helping to identify the bottleneck of the sold products. Innovations focused on the less efficient structures and products of a company are the most beneficial in long term resulting in more profitability.

Figure 4.16: Press Brakes - chamados information

The statistics of the test phase were important to identify that the software should be focused on Press Brakes and specifically for these two main components. All the adjustments of the software after the test phase were made in this direction. Also the feedback from Adira's Smart Diagnostic System users were taken in consideration to future improvements.

4.8 Smart Diagnostic System Impact

The influence of the developed software (Smart Diagnostic System for Machine Problems) on Adira's current service process was studied to measure the success of the implementation. During the test phase, with only a small amount of data inserted on the Database it was possible to decrease the workload of Adira's engineers related to chamados by 9,3%. Previously all the trouble tickets were solved individually. With the implementation of this software this percentage was reduced to 19,7%. These results are translated in better time of response provided to customers and better utilization of resources on this department.

Chapter 5

Future work and conclusions

This chapter sums up all the developed studies from the beginning of the proposed concept until the implementation of this project. Firstly it was made an analysis of the Smart Diagnostic System success based on the proposed goals. After this aspect it was explored the possibility of future development of the studied theme either to improve the current system or to use it as a base to implement the concept in more advanced technologies.

5.1 Smart Diagnostic System Conclusions

This project had three main stages of work: the sketch of the database, the development of the Web interface and the software test phase.

In order to idealize this system the first step was to study the current Adira's after sales service and treat all the available data. Firstly the studies were based on doing small surveys and data mining on Adira emails. After getting knowledge on this process it was possible to build the methods to store data and construct the database. It identified the main stages, since the beginning of a technical support until its complete resolution, and based on that it were implemented the main concepts for the Smart Diagnostic System on Java programming language. Finally the Web application was subject of an implementation test phase to assure that the developed software had what it needs to keep up with Adira's expectation and quality excellence.

Overall, the experience on this project was very positive and with important improvements on after-sales support, even though it cannot completely replace the human intervention on this sector.

The main complications of the application are stated bellow:

- Complex problems can't be automatically solved by the application.
- The application requires maintenance to be up to date.

Also, the cooperation with Adira on this thesis allowed to acquire competences related to after-sales on a wide range due to the involvement in other smaller projects. Adira Velox was a project developed in parallel to the Smart Diagnostic System that enabled a better understanding on the data saving on Enterprise Resource Planning as well as getting "know how" on the ground support team.

5.2 Fulfilled goals

The main goals of the application are stated below:

- The time spent by engineers dedicated to this subject was decreased
- The client estimated waiting time for problem solutions was highly decreased
- The availability and organization of data to study and maintain continuous improvement on Adira's products was significantly increased
- Implementation towards Industry 4.0, replacing services for automated mechanisms

With the work developed it was possible to achieve all the points presented above. All the proposed tasks were successfully implemented and, furthermore, deeply analyzed. The Smart Diagnostic System is currently implemented on Adira Server and being used by the after sales service department. The initial expectation was accomplished.

5.3 Future work

The future work of the Smart Diagnostic system can have a great range of applicabilities. As it is, the developed Web application can be expanded to a bigger structure including data in more detail of all Adira products. As stated before, the current software is more focused on Press Brakes, since it represents the biggest impact on after sales services. If the database has more information it will be more efficient and useful for future analysis which makes the Smart Diagnostic System a tool in constant evolution.

One big step towards the future development of this project is the implementation of the acquired data into a completely independent system that works on the machine itself.

With the analysis of the symptoms/solutions provided by the Smart Diagnostic System it can be developed a mechanism that, instead of providing solutions, give those instructions to the machine that can automatically heal itself. A self-healing system of this type will drastically reduce the after sales service costs for the company and improve the product quality by a significant margin. This is a strong applicability of data towards Industry 4.0 which is one of the goals of this project.

Projects that make use of this concept are explained on chapter 2.

Appendix A

Process Development

A.1 Process questionnaire

- **Client Call.**
  - Who receive the call?
    - Is there any standard questionnaire?
      - What information is asked?
  - What is done with that information?
  - That data is forward to whom?
- **Forward the information.**
  - How is the data analyzed?
    - Based in what instructions?
  - What is done with that data?
    - Is there any additional information forwarding?
- **Problem detected.**
  - Who solves the problem?
  - How is that problem solved?
  - Is there any additional information forwarding?
- **Problem resolution.**
  - Is there any problems/solutions documentation done?
    - Who does that documentation?
    - How is that documentation made?
  - The solved issue is analyzed/studied?
    - Who develop those analyzes?
    - What is done with that study?

A.2 Diagnostic process flow diagram

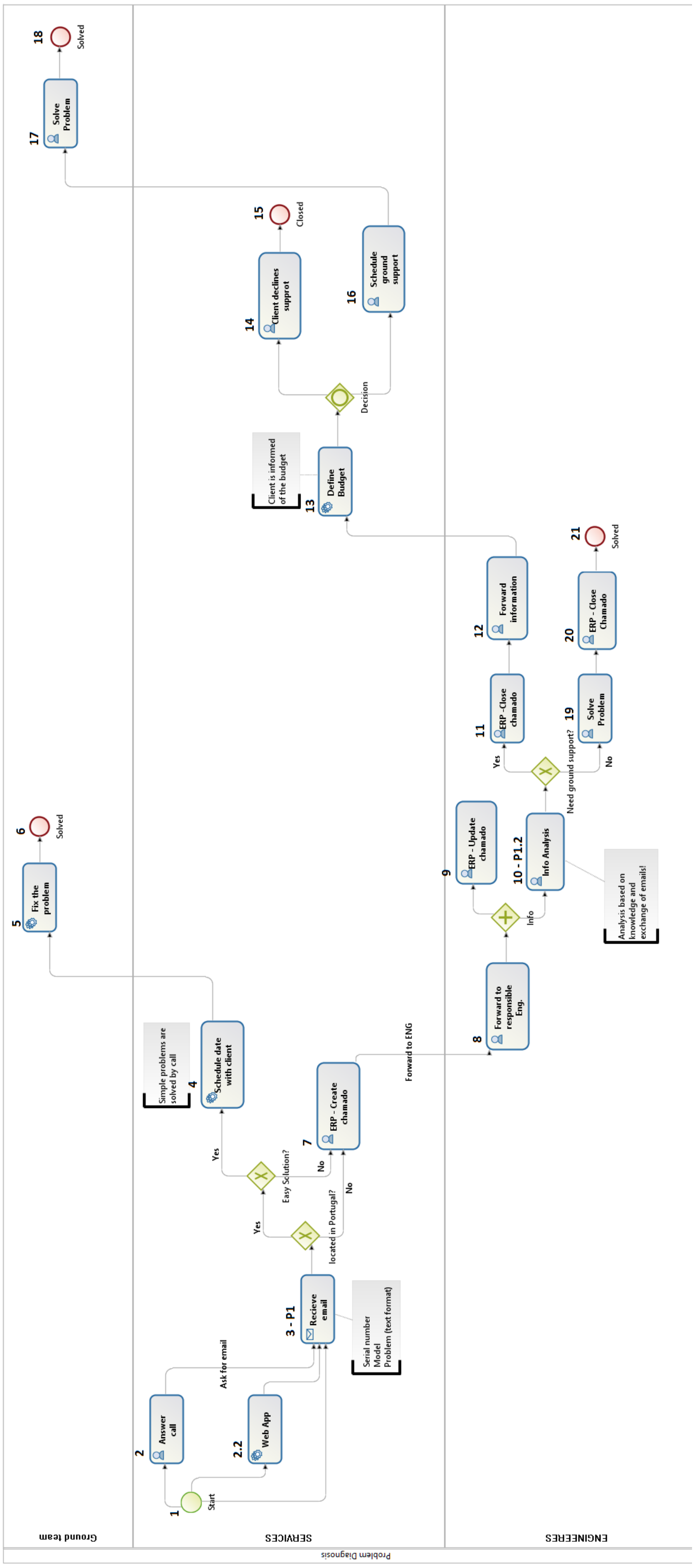


Figure A.1: Machine Diagnosis Diagram

A.3 Database Structure

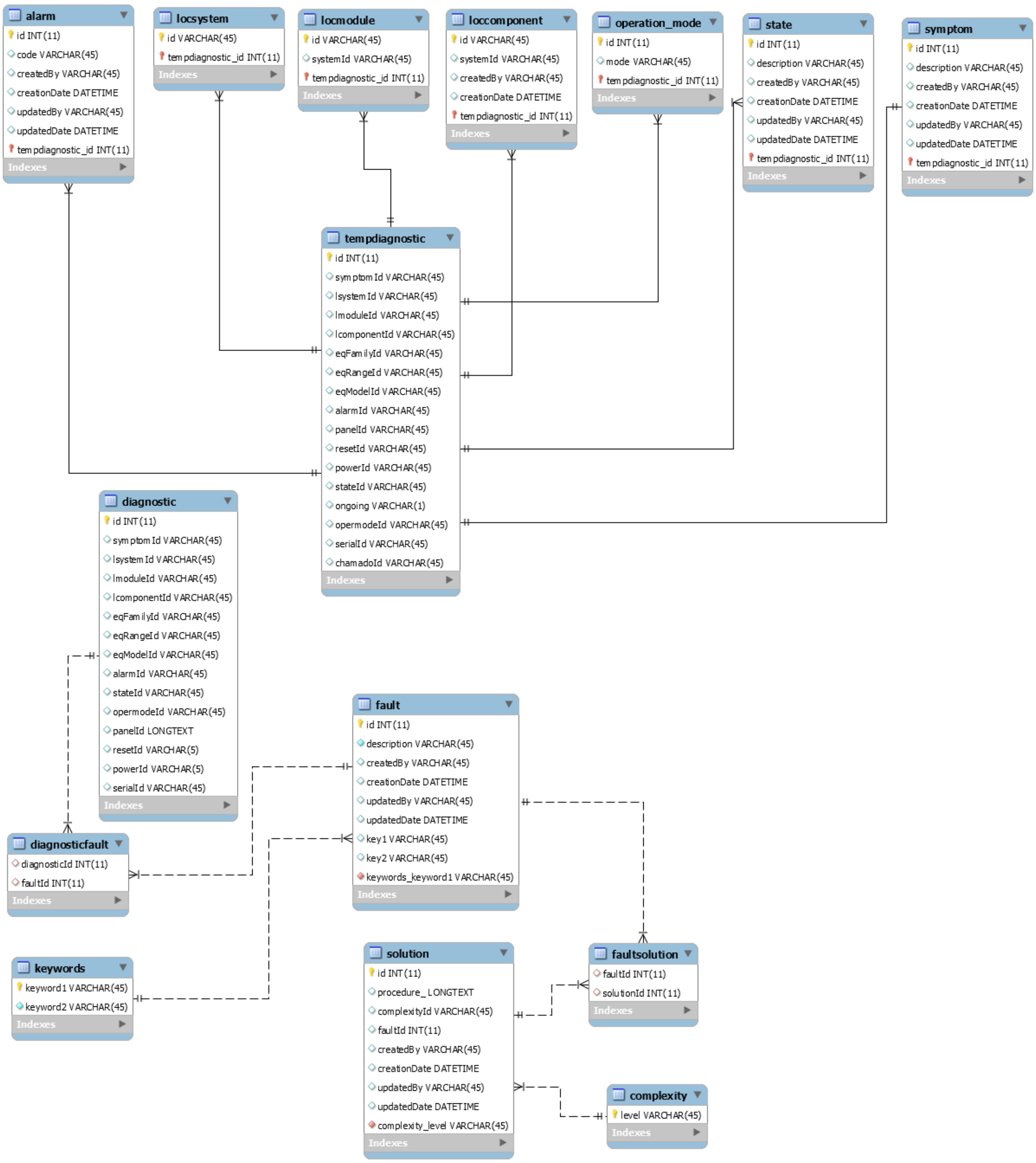


Figure A.2: Database entity–relationship complete model

A.4 Modular Machine Dissection

Table A.1: Press Brake Structure

System	Module	Component
Avental Móvel	Eixo Y1 Eixo Y2	Cilindro Régua Linear Aperto Ferramentas Guias Fuso Motor Montantes
Avental Inferior		Mesa Bombeada Blindagem Matriz Acompanhador de Quinagem Montante
Esbarro	Eixo X1 Eixo X2 Eixo R1 Eixo R2 Eixo Z1 Eixo Z2	Braço Dedos do Esbarro Régua Guias Servo Motor
Unidade Hidráulica		Válvulas Filtros Bomba Hidráulica Motor Depósito do Óleo Mangueiras
Quadro Elétrico		PCSS PVR6 Servo Drives Variador de Velocidade
Comando		Comando Numérico Painel de Controlo Pedal de Acionamento

Table A.2: Shear Structure

System	**Component**
Parte Móvel	Lámina
	Moentes
	Porta Láminas
	Cilindros Hidráulicos
Parte Fixa	Folga Motorizada
	Fins de Curso
	Lámina
	Calcadores Hidráulicos
	Corpo da Estrutura
Mesa Apoio de Chapas Finas	Cilindro Pneumático
	Régua Especial
	Pressostato
	Válvulas
	Apoios Auxiliares
Esbarro	Braço
	Régua
	Guias
	Servo Motor
Unidade Hidráulica	Válvulas
	Filtros
	Bomba Hidráulica
	Motor
	Depósito do Óleo
	Mangueiras
Quadro Elétrico	Relé de Comando
	Contactores
	Autómato (Logo/Zelio)
	Variador de Velocidade
Comando	Comando Numérico
	Painel de Controlo
	Pedal de Acionamento
Elementos de Segurança	Grelha Frontal
	Células Posteriores
Acessórios	Braços de Apoio
	Guias de Esquadria
	Esbarros Auxiliares

A.5 Data Analysis

The data analysis represents the collections of information that the software acquire during the test phase. On Figure A.3 it is possible to have a visual description on how the data is saved on Adira's Database.

Figures A.4, A.5, A.6 represents the first stage of the data collection on an excel file.

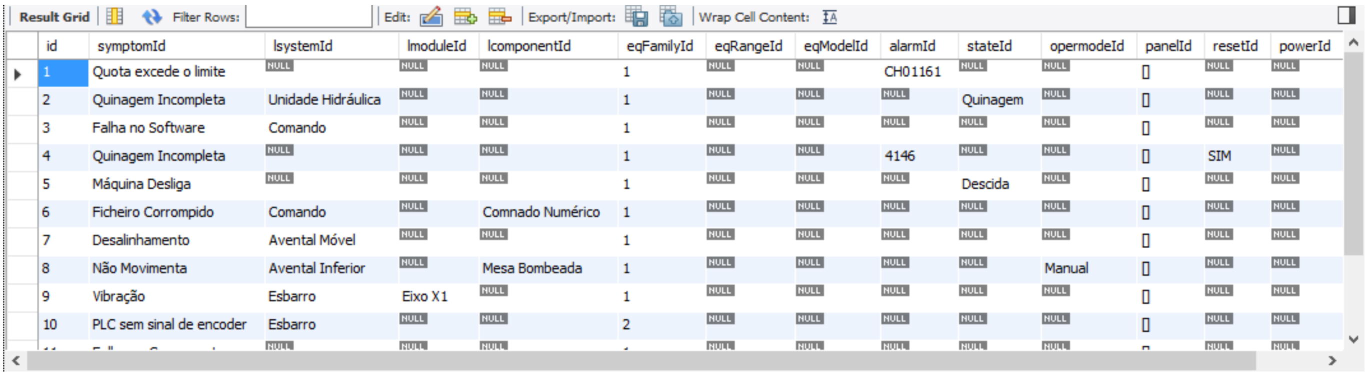
In light green it is represented all the good data obtained by trouble tickets.

In yellow it is represented all the future good data that is still on the process of trouble solving and debugging.

In blue it is represented all the unconfirmed information that is related to already solved issues.

In red it is represented all the information that was retrieved from chamados but it is not useful for the Smart diagnostic system database so it is ignored.

In dark green it is represented all the previously available information that was inputted on the system before the start of the test phase.



id	symptomId	lsystemId	lmoduleId	lcomponentId	eqFamilyId	eqRangeId	eqModelId	alarmId	stateId	opermodeId	panelId	resetId	powerId
1	Quota excede o limite	NULL	NULL	NULL	1	NULL	NULL	CH01161	NULL	NULL	▯	NULL	NULL
2	Quinagem Incompleta	Unidade Hidráulica	NULL	NULL	1	NULL	NULL	NULL	Quinagem	NULL	▯	NULL	NULL
3	Falha no Software	Comando	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
4	Quinagem Incompleta	NULL	NULL	NULL	1	NULL	NULL	4146	NULL	NULL	▯	SIM	NULL
5	Máquina Desliga	NULL	NULL	NULL	1	NULL	NULL	NULL	Descida	NULL	▯	NULL	NULL
6	Ficheiro Corrompido	Comando	NULL	Comnado Numérico	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
7	Desalinhamento	Avental Móvel	NULL	NULL	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
8	Não Movimenta	Avental Inferior	NULL	Mesa Bombeada	1	NULL	NULL	NULL	NULL	Manual	▯	NULL	NULL
9	Vibração	Esbarro	Eixo X1	NULL	1	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL
10	PLC sem sinal de encoder	Esbarro	NULL	NULL	2	NULL	NULL	NULL	NULL	NULL	▯	NULL	NULL

Figure A.3: Database Inputs

MAQUINA	ALARME	PROBLEMA	LOC. SIST	LOC. MO	LOC.COM	MODO OP	EST. TRAB	FALHA	SOL.	BOTAO LIG
										(INTERMITENT
M00000612, PRESS BRAKE	CH01161	QUOTA EXCEDE O LIMITE						ERRO NA POSIÇÃO DO EIXO	>Verificar se há correcções a fazer no eixo X. >Verificar se no Monitor IO o sinal "Eixo em posição" está on ou off.	
A1001523-7179/13312, PRESS BRAKE HIDRAULICAS		FALHA NA QUINAGEM	UNIDADE HIDRAULICA				QUINAGEM	FALTA DE PRESSÃO	>Verificar os valores no manómetro. >Verificar se há algum acessório mal apertado. >Verificar se a bomba está a funcionar devidamente.	
M00000539, PRESS BRAKE		MÁQUINA DESLIGA					DESCIDA	MÁ LIOGAÇÃO NOS DIJUNTORES	>Verificar a continuidade entre o terminal 13 e os terminais de alimentação (92, 192, 392, 492, 592, 692) >Verificar os dijuntores	
A1002421-5751/10687 (ZZ10103), PRESS BRAKE		SOFTWARE NOT WORKING	COMANDO NUMERICO					AVARIA NA FONTE DE ALIMENTAÇÃO	>Necessita de intervenção (compra de nova fonte de alimnentação)	
M00000612, PRESS BRAKE	4146							LAZER MAL ALINHADO	>Verificar se o feixe laser está bem alinhado (paralelo) e fazer o respectivo alinhamento	
PM11030 7242/13424 - ZZ12850	6461									
M00000167/00394, Quinadora com ESA controll		CORRUPTED FILE - APP FAIL.	COMANDO NUMERICO					FICHEIRO CORROMPIDO	>Verificar mensagens no ecra para identifição do ficheiro corrompido	
M00000402		DESALINHADO	AVENTAL MÓVEL					FALHA NO POTÊNCIOMETRO	>Verificar se o potênciometro da mesa bombeada funciona correctamente (substituir potenciômetro)	
QU0-0065-24-0266/PM022040, quinadoras com mesa bombeada		NÃO SE MOVE	MESA DE BOMBAGEM			MANUAL		PROBLEMA NA TRANSMISSÃO MECÂNICA	>Verificar se a transmissão mecânica entre o motor WILA e o sistema de bombeado não está partido.	
M00000433/01283, Quinadora		VIBRAÇÃO	ESBARRO	EIXO X				CALIBRADOR TRASEIRO MECÂNICO	>Se a vibração é durante movimentos de alta velocidade, reduzir a velocidade do parâmetro 130 para 400,00 mm e verificar se a vibração é solucionada. >Verifique se o backgauge mecânico está em funcionamento correcto	
SM630 - GL0-0020-20-0219, guilhotinas	PLC SEM SINAL DE ENCONDER	PLC SEM SINAL DE ENCONDER	ESBARRO(ENCONDER)						>Verificar a conexão do encoder está correcta >Desconectar e voltar a conectar ligação >Verificar se os fios estão bem apertados no bloco de terminais >Verificar se o cabo do encoder não está danificado	
M00000611			Unidade Hidráulica - Depósito do Óleo							
M00000443/01259										
M00000432										
QU3-0067-32-0453		O COROAMENTO NÃO FUNCIONA NO O CONTROLER						O COROAMENTO NÃO FUNCIONA NO O CONTROLER	>Check if there are not any modifications on the Crowning DNC parameters. >Re-introduzir novamente os parâmetros, usar a função de restauro. >Se o problema continuar a ocorrer é necessário modificar os outputs para coroamento. Modificar os terminais Mout1 (terminais 10 e 11) e usar outros terminais livres. Modificar estes novos terminais nas definições dos parâmetros DNC.	
QIH600-400 7172/13466		PROBLEMA NO LASER SAFE						LASER SAFETY NÃO FUNCIONA	>Verificar se não está a trabalhar em "Field Mute" na página de segurança. >Verificar se o botão Run está na	
PA16030 (7197/13334) ZZ12759	4066									IN
M00000402										
M00000205	403F	NÃO DESCE	AVENTAL MÓVEL	EIXO Y						
M00000628		NÃO SE MOVE, SE NÃO PRESSIONAR BOTÃO VERDE	AVENTAL MÓVEL						>Verificar a conecção no contactor 2KM1. >Verificar a connecção número 14 on o cabo número 35 está ligado.	
M00000568		GOES A LITTLE DOWN	AVENTAL MÓVEL				INDEXAÇÃO			
M00000203	AX0583									
M00000480/01524										
QU0-9999-999/A1001355		TANDEM MODE PROBLEM								
M00000311		FALHA NO MOTOR	Avental móvel	Y2						
M00000570/01863	ANDRE									
QHD-5020 5244/9722 - ZZ9127		Não sobe	avental móvel							
QU0-9999-999/A1001564										
M00000203										
M00000655/02349										
M00000611/02111			esbarro	z2						
QU0-9999-999/A1002175										
M00000388									>Check the Auxiliary Axis Contact Monitor input to see if it is low when the Emergency Stop OK Output is low . >Check the wiring is correct , more information on the wiring of the selected Emergency Stop Option can be found in the Technical Manual >Check if kontaktors 10K3/12K3/14K3/25K3/26K3 are fully functional. >Verify all the elctrical connections.	
		DOESN'T GO DOWN	AVENTAL MÓVEL			SMI/AUTOMATIC				
QU0-9999-999/A1002203		DESALINHADO	AVENTAL MÓVEL							

Figure A.4: Data analysis excel 1

								NÃO ATINGE PONTO DE REFERENCIA			
QIHD16040 - 7003/12936 - ZZ12359											
M00000580											
M00000466											
M00000089		DESALINHAMENTO	AVENTAL MOVEL				GOING UP				
M00000460/01456											
M00000165/00444											
QHD-07025 7008/12963D - ZZ12386											
QHD MACHINES		DNC DESLIGA	PEDAL DE ACCIONAMENTO					DNC PERDE OS DADOS	>Desligar a máquina, no seccionador, e voltar a ligar para reiniciar o DNC. No modo de inicialização, no ecran, escrever 1, e no código de acesso para apagar variaveis escrever 817 e ENTER, e continuar normalmente a arrancar com a máquina		
QHD MACHINES		NÃO ARRANCA	MOTOR					PROBLEMA NOS MICROS	>Desactuar micros e voltar a afinar.		
QHD MACHINES		PARA/DESLIGA							>Verificar se a barra atrás do avental móvel está a accionar um micro de inclinação. Afinar micros de inclinação >Verificar se o veio de transmissão da profundidade de quinagem não está solto ou se roda livre. Montar a		
QIH MACHINES		CHOQUE HIDRAULICO	AVENTAL MÓVEL				PMI	CHOQUE HIDRAULICO NA DESCOMPRESSÃO DO PMI	>Verificar o parâmetro 36, Pressão/Descompressão e ajustar para 5%.		
QIH MACHINES		DESALINHADO	AVENTAL MÓVEL			AUTOMATIC O/SEMI	DESCIDA		>Verificar se a montagem da régua linear não está danificada >Verificar em modo Manual se o avental móvel desce, o eixo (Y1 ou Y2) que não desce em Modo Semi-Automático passa a descer. >Verificar e alterar (se necessário) as cotas de Index e de referência para os valores correctos.		
QIH MACHINES		DESALINHADO	AVENTAL MÓVEL			INDEXAÇÃO	SUBIDA		>Verificar o parâmetro 33, pressão de subida, aumentar esse valor para		
QIH MACHINES	DNC80 - NTLDR		COMANDO NUMERICO						>Actualizar dados com o backup		
QIH MACHINES	JEEETER		COMANDO NUMERICO						>Verificar o estado e ligações do cabo da régua linear >Verificar a montagem da régua		
QIH MACHINES		START NÃO APAGA	COMANDO NUMERICO						>Verificar o parâmetro 123 - tolerância de posicionamento, e aumentar o valor.		
QIH MACHINES		POSICIONAMENTO ERRADO	ESBARRO	EIXO R					>Introduzir os seguintes valores nos parâmetros: >Ganho proporcional (142 = 100) >Tempo de aceleração (128= apagar ou 0,5seg) >Tempo de desacelaração (129 = apagar ou 0,5seg) >Velocidade de posiciopnamento (130 = 60mm/s) >Ganho Integral (143 =20) >Distância de retrocesso (125 = (+) entre 0,2 a 0,3mm) >Afinar o freio		
QIH MACHINES		STOP, VAR KO, VARIADOR KO	COMANDO NUMERICO						>Se no servo-amplificador estiver ligado o led vermelho, este deve ser substituido. >Se estiver ligado o led amarelo (intermitente), verificar cabo e ficha do cachimbo mais pequeno da servo-válvula, se não há nenhum fio traçado ou a fazer massa. Se o cabo estiver bem, a causa é a própria servo-válvula. >Se no variador estiver ligado o led vermelho este deve ser substituido		
PHDM MACHINES		NÃO FAZ MUDANÇA DE VELOCIDADE	AVENTAL MÓVEL					DESCIDA	>Verificar se o micro não está encravado mecanicamente (se necessário substituir) >Verificar ligação electrica >Verificar se a válvula H3, pela placa de Leds, desliga no ponto de mudança de velocidade. >Verificar se o cachimbo está igado na válvula H3 >Desmontar válvula H3 e elemento lógico, limpar e desencravar êmbolo >Regular no regulador de caudal até acertar na velocidade lenta. >Substituir Ponte Rectificadora dos 24V		
PHDM MACHINES		SOLAVANCOS	AVENTAL MÓVEL				AUTOMÁTIC O	DESCIDA	>Executar ligação eléctrica à válvula H3 (verificar cachimbo) >Substituir relés ou contactores >Substituir Ponte Rectificadora dos 24V		
PHDM MACHINES		NÃO ARRANCA	MOTOR						>Verificar/Substituir fusiveis >Desligar botão de emergência		
PHDM MACHINES		VELOCIDADE DESCIDA						DESCIDA			
PHDM MACHINES		FALHA VELOCIDADE DESCIDA RÁPIDA	AVENTAL MÓVEL					DESCIDA RÁPIDA	>Aliviar parafusos da tampa inferior (liertá-la um pouco) e ensaiar - quando ocorrer o problema.		
	402A	OPPERAÇÕES NÃO PERMITIDAS	PEDAL DE ACCIONAMENTO					PROBLEMA DE CONEXÃO	>Verificar se não há problemas com os cabos do pedal. Se não houver problemas é sinal que as entradas CN4-3 e CN4-4 estão com o mesmo valor (24v ou 0v) e NÃO pode ser assim. >Verificar se os fios estão bem conectados.		
	AX0008	NÃO MOVE	AVENTAL MÓVEL					PROBLEMA DE PARÂMETROS	>Restaurar os parâmetros de fábrica, verificar o encaixe dos relés. >Verificar se não existem saidas do autómato lasersafe queimadas.		
	00FA	ERRO						PROBLEMA AUTÓMATO	>Verificar ligação dos conectores do autómato (CN16/CN26).		NÃO
		FALHA NA MUDANÇA DE VELOCIDADES	AVENTAL MÓVEL					PROBLEMA DE VELOCIDADES	>Verificar nível do óleo; verificar fugas on "suction hoses". >verificar o funcionamento da valvula H2. >Verificar parametro 45 (commutation time) valores normais - 0,2/0,3. >verificar estado de micros,		
	00CD	PARA A MEIO	AVENTAL MÓVEL					PROBLEMA NO ACCIONAMENTO DO PEDAL	>Para ultrapassar o erro 00CD executar o seguinte procedimento: >Pressionando o botão verde do painel de comando para ligar o motor. >Manter o botão verde pressionado. >Pressionar o botão azul para apagar o erro 00CD (mantendo sempre o botão verde pressionado). >De seguida referenciar a máquina.Neste paço é necessário ter especial cuidado pois o sistema de protecção contra inclinação está		

Figure A.5: Data analysis excel 2

	4212	ERRO						Problema com o laser	>Verificar o alinhamento paralelo do laser >Verificar a sincronização de Y1 e Y2 no CNC quando o laser detecta material. >Se a maquina está a ser operada sem material, a grossura do material deve ser programada para 1 ou 1,5mm. Caso contrário o lâmpada mute pode estar intermitente e pode disparar o alarme 4212.		
		CALIBRADOR TRAZEIRO NÃO MOVE				INDEXAÇÃO		INDEXAÇÃO DEFEITUOSA	máquina são os de origem. >Um dos eixos pode não estar correctamente indexado. Sair do processo de indexação para todos os eixos. >Verificar se o parâmetro 145 corresponde ao valor indicado nos ficheiros da máquina e de seguida limpar este parâmetro, fazer este procedimento pra todos os eixos. >Desconectar a máquina e tentar fazer indexação novamente. A barra deve estar em baixo. Neste caso os eixos que vão indexar são o Y1 e Y2. >Se o led DNC Start estiver on significa que as escalas lineares não estão a funcionar correctamente. Caso contrário o eixo elétrico não está a fazer o procedimento de indexação correctamentende. >Na situação em que o led está off é necessário activar o parâmetro 145 um a um e fazer o procedimento de indexação. Quando encontrar o eixo com problemas na indexação é necessário verificar se o switch de deteção de indexação está activado		S
	00CD	ERRO						PROBLEMA NOS TERMINAIS	>Aumentar os parâmetros de [illegible] >Fazer ponte entre os terminais 968 e 968 e verificar se o problema é solucionado.		
	2205	ERRO						REINICIO DA MÁQUINA DEMASIADO RÁPIDO	>O erro deve-se a desligar e ligar a máquina em menos de 10 segundos. >Volte a reiniciar a máquina para solucionar o problema.		
	4004	O MONITOR DO SERVO-VALVE Y1 NÃO RESPONDE						PROBLEMA NOS CONECTORES	>Verificar e reapertar em CN7 o conector X26. >Verificar se o output de X26 é igual a X27.		
	4006	ERRO						PROBLEMA NA SERVO-VALVE	>Desligar as servo-valves do circuito, limpar e voltar a colocar. >Verificar se o filtro do óleo está limpo, caso contrário substituir o filtro.		
		BARULHO NA MUDANÇA DE VELOCIDADE						PARÂMETROS DE VELOCIDADE ERRADOS	>Alterar o parâmetro de mudança de velocidade para um valor entre 0,4 e 0,5 segundos.		
		FUGA DE ÓLEO	AVENTAL MÓVEL	CILINDRO				ESTRAGO NOS VEDANTES	>Trocar os vedantes		
	7010	ERRO						VALORES DE VELOCIDADE MÁXIMOS ULTRAPASSAI	>Ir ao parâmetro de velocidade ascendente do eixo e reduzir o valor. >Verificar se o problema é solucionado. Caso contrário, ir ao parâmetro de velocidade descendente do eixo e reduzir o valor.		
		VIBRAÇÃO NO MOTOR - EXECUTAR AUTO-TUNNING						AUTO-TUNNING	>Iniciar o motor. Fazer reset à máquina. Mover o eixo para o meio e pressionar stop no comando numérico. >Verificar se o servo-drive está no modo READY. (Caso não esteja, carregar no botão stop no CNC ou remover o cabo do input D10 e D12) >Pressionar o botão de navegação, seleccionar OP e pressionar o botão de navegação. Seleccionar TUN e pressionar o botão de navegação. Seleccionar TUST e pressionar o botão de navegação. (O eixo deve estar agora em movimento lento e em auto tunning) >Quando o auto tunning terminar aparecera a mensagem "done". Pressionar o botão de navegação para guardar os parâmetros. >Pressionar o botão ESC duas vezes. Desligar a máquina. Caso tenha removido os cabos do input D10 e D12 voltar a conectar. Voltar a ligar a máquina.		
		FUSO ENCRAVADO	AVENTAL MÓVEL	FUSO				SUBIDA ANORMAL DO EIXO Y			
									>Programar os seguintes valores: >Y1 = -3,00 ; Y2 = -3,00 ; Pressão = 10,00 >Pressionar ENTER para entrar no novo ecran. Escolher a opção de velocidade lenta descendente: 1 2 3 4 5 | 1 1 0 1 0 >Pressionar STOP. Pressionar START. Pressionar o pedal descendente por um pequeno período de tempo.		
		VIBRAÇÃO NO MOTOR - EXECUTAR AUTO-TUNNING						AUTO-TUNNING	>Iniciar o motor. Fazer reset à máquina. Mover o eixo para o meio e pressionar stop no comando numérico. >Verificar se o servo-drive está no modo READY. (Caso não esteja, carregar no botão stop no CNC ou remover o cabo do input D10 e D12) >Pressionar o botão de navegação, seleccionar OP e pressionar o botão de navegação. Seleccionar TUN e pressionar o botão de navegação. Seleccionar TUST e pressionar o botão de navegação. (O eixo deve estar agora em movimento lento e em auto tunning) >Quando o auto tunning terminar aparecera a mensagem "done". Pressionar o botão de navegação para guardar os parâmetros. >Pressionar o botão ESC duas vezes. Desligar a máquina. Caso tenha removido os cabos do input D10 e D12 voltar a conectar. Voltar a ligar a máquina.		
		FUSO ENCRAVADO	AVENTAL MÓVEL	FUSO				SUBIDA ANORMAL DO EIXO Y			
									>Programar os seguintes valores: >Y1 = -3,00 ; Y2 = -3,00 ; Pressão = 10,00 >Pressionar ENTER para entrar no novo ecran. Escolher a opção de velocidade lenta descendente: 1 2 3 4 5 | 1 1 0 1 0 >Pressionar STOP. Pressionar START. Pressionar o pedal descendente por um pequeno período de tempo.		

Figure A.6: Data analysis excel 3

References

[1] H.P. Wang, H.H. Huang, G.M. Knapp, C.C. Lin, S.S. Lin, and J.K. Spoerre. Machine fault diagnostics system and method, October 15 1996. US Patent 5,566,092. URL: https://www.google.com/patents/US5566092.

[2] Józef Korbicz, Jan M Kościelny, Zdzislaw Kowalczuk, and Wojciech Cholewa. *Fault diagnosis: models, artificial intelligence, applications*. Springer Science & Business Media, 2012.

[3] J. Han, I. Cho, P. Lee, and H. Jeong. Diagnostic system and method for home appliance, March 17 2015. US Patent 8,983,798. URL: https://www.google.com/patents/US8983798.

[4] Fabian Hecklau, Mila Galeitzke, Sebastian Flachs, and Holger Kohl. Holistic approach for human resource management in industry 4.0. *Procedia CIRP*, 54:1–6, 2016.

[5] L Gehrke, AT Kühn, D Rule, P Moore, C Bellmann, S Siemes, D Dawood, S Lakshmi, J Kulik, and M Standley. A discussion of qualifications and skills in the factory of the future: A german and american perspective. *VDI/ASME Industry 4.0 White Paper*, pages 1–28, 2015.

[6] Günther Schuh, Till Potente, Cathrin Wesch-Potente, Anja Ruth Weber, and Jan-Philipp Prote. Collaboration mechanisms to increase productivity in the context of industrie 4.0. *Procedia CIRP*, 19:51–56, 2014.

[7] Audit Tax Consulting Corporate Finance. Industry 4.0 challenges and solutions for the digital transformation and use of exponential technologies.

[8] Shiyong Wang, Jiafu Wan, Di Li, and Chunhua Zhang. Implementing smart factory of industrie 4.0: an outlook. *International Journal of Distributed Sensor Networks*, 2016.

[9] Malte Brettel, Niklas Friederichsen, Michael Keller, and Marius Rosenberg. How virtualization, decentralization and network building change the manufacturing landscape: An industry 4.0 perspective. *International Journal of Mechanical, Industrial Science and Engineering*, 8(1):37–44, 2014.

[10] WOLFGANG SCHROEDER. Germany's industry 4.0 strategy. 2016.

[11] Agnieszka Radziwon, Arne Bilberg, Marcel Bogers, and Erik Skov Madsen. The smart factory: exploring adaptive and flexible manufacturing solutions. *Procedia Engineering*, 69:1184–1190, 2014.

[12] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. Smart factory-a step towards the next generation of manufacturing. In *Manufacturing systems and technologies for the new frontier*, pages 115–118. Springer, 2008.

[13] Travis Hessman. The dawn of the smart factory. *Industry Week*, 14:14–19, 2013.

[14] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.

[15] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18, 2010.

[16] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[17] Feng Xia, Laurence T Yang, Lizhe Wang, and Alexey Vinel. Internet of things. *International Journal of Communication Systems*, 25(9):1101, 2012.

[18] Internet of things picture.

[19] Hermann Kopetz. Internet of things. In *Real-time systems*, pages 307–323. Springer, 2011.

[20] Gerd Kortuem, Fahim Kawsar, Vasughi Sundramoorthy, and Daniel Fitton. Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, 14(1):44–51, 2010.

[21] Mohamed Sayed. Health monitoring and life-long capability management for self-sustaining manufaturing systems (selsus). 2017.

[22] Bert Bates and Kathy Sierra. *Head First Java*. 2011.

[23] Edgar F Codd. Relational database: a practical foundation for productivity. *Communications of the ACM*, 25(2):109–117, 1982.

[24] Lynn Beighley. *Head First SQL: Your Brain on SQL–A Learner's Guide*. " O'Reilly Media, Inc.", 2007.

[25] Raghu Ramakrishnan. Database management systems . pdf. 2000.