



Universidade do Minho
Escola de Engenharia

João Gabriel Ramos Ferreira Bernardo

**A contribution for data processing and
interoperability in Industry 4.0**

Tese de Mestrado

Mestrado em Engenharia de Sistemas

Trabalho efetuado sob a orientação dos

Professora Doutora Leonilde Varela

Professor Doutor Luís Ferreira

Outubro de 2019



DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada. Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do Repositório da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição CC BY

<https://creativecommons.org/licenses/by/4.0/>



ACKNOWLEDGMENT

First, I would like to thank professor Leonilde Varela and Engineer Joana Vinhas who made the bridge between the system engineering course and the company. Without this opportunity, the project would not be possible.

I would like to thank my advisors, professor Leonilde Varela for the continuous support and professor Luis Ferreira for sharing his knowledge with me.

My sincere gratitude and congratulations to the process engineering department of ZF Ponte de Lima for their patience and help along the process. They have the task to integrate all the company processes using not many resources.

Without the IT manufacturing team help, the code would not be completed until today.

A special thanks to my mentor in-company, Diogo Silva, for the motivation and help.

Lastly, a profound thank to my brother, parents and family for encouraging me through the entire process. Without my mom and dad supports I could not be who I am today.

To all my sincere and profound thank you.



STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.



RESUMO

Espera-se que a Indústria 4.0 seja uma mudança significativa no crescimento das empresas. O objetivo é agrupar informações importantes de toda a cadeia de suprimentos da empresa, proporcionando uma tomada de decisão mais acertada, ao mesmo tempo que permite interações entre seres humanos e máquinas em tempo real.

Sistemas autônomos equipados com Tecnologias da Informação possibilitam a Indústria 4.0 como a Internet das Coisas (IoT), sistemas ciber-físicos (CPS) e *Big Data* e *analytics*.

A IoT coleta informações de cada peça do grande quebra-cabeça que é o processo de fabricação. *Cloud Computing* lida com armazenamento de toda essa informação em um só lugar. As pessoas compartilham informações em toda a empresa, na cadeia de abastecimento e níveis hierárquicos por meio da integração de sistemas. Por fim, *Big Data* e *analytics* são de inteligência que melhorarão a Indústria 4.0.

Os métodos e ferramentas da Indústria 4.0 são projetadas para aumentar a interoperabilidade entre os *stakeholders*. Para tornar possível essa interoperabilidade, um padrão em toda a empresa deve ser implementado.

Dois modelos de referência para a Indústria 4.0 foram estudados - RAMI 4.0 e IIRA. RAMI 4.0, a iniciativa alemã, concentra-se na digitalização industrial, enquanto IIRA, a iniciativa americana, foca no mundo da Internet das Coisas, como energia, saúde e transporte.

As duas iniciativas visam obter dados inteligentes dos processos e, ao mesmo tempo, permitir a interoperabilidade entre os sistemas. Representantes dos dois modelos de referência estão a trabalhar juntos para discutir os padrões de interface tecnológica que podem ser usados pelas empresas que entram nessa nova era.

Este estudo visa a interoperabilidade entre sistemas. Embora deva haver um modelo para orientar a empresa na Indústria 4.0, esse modelo deve ser mutável e flexível o suficiente para lidar com diferenças no processo de fabricação, como exemplo a indústria 4.0 automotiva não terá a mesma abordagem que a Indústria 4.0 de aviação.

PALAVRAS-CHAVE

IIoT, Indústria 4.0, Integração de sistemas, Interoperabilidade, Padronização.



ABSTRACT

Industry 4.0 is expected to drive a significant change in companies' growth. The idea is to cluster important information from all the company's supply chain, enabling valuable decision-making while permitting interactions between machines and humans in real time.

Autonomous systems powered with Information Technologies are enablers of Industry 4.0 – like Internet of Things (IoT), Cyber Physical-Systems (CPS) and Big Data and analytics.

IoT gather information from every piece of the big puzzle which is the manufacturing process.

Cloud Computing store all that information in one place. People share information across the company, between its supply chain and hierarchical levels through integration of systems.

Finally, Big Data and analytics are of intelligence that will improve Industry 4.0.

Methods and tools in Industry 4.0 are designed to increase interoperability across industrial stakeholders. In order to make the complete process possible, standardisation must be implemented across the company.

Two reference models for Industry 4.0 were studied - RAMI 4.0 and IIRA. RAMI 4.0, a German initiative, focuses on industrial digitalization while IIRA, an American initiative, focuses on "Internet of Things" world, i.e. energy, healthcare and transportation.

The two initiatives aim to obtain intelligence data from processes while enabling interoperability among systems. Representatives from the two reference models are working together on the technological interface standards that could be used by companies joining this new era.

This study aims at the interoperability between systems. Even though there must be a model to guide the company into Industry 4.0, this model ought to be mutable and flexible enough to handle differences in manufacturing process, as an example automotive industry 4.0 will not have the same approach as aviation Industry 4.0.

KEYWORDS

IIoT, Industry 4.0, Interoperability, Systems Integration, Standardization.



CONTENTS

Acknowledgment	iii
Resumo.....	v
Abstract	vi
List of figures	x
Abbreviations and Acronyms	xi
1. Motivation and background	1
1.1 Objectives	1
1.2 Structure of the Dissertation	2
2. Literature Review	3
2.1 Historic Background.....	3
2.2 Third Industry Revolution	5
2.2.1 Sensors	5
2.2.2 Actuators	5
2.2.3 Programmable logic controllers (PLC).....	5
2.2.4 Communication and OPC	6
2.2.5 Supervisory Control and Data Acquisition (SCADA)	6
2.2.6 Manufacturing Execution System (MES).....	7
2.2.7 Enterprise Resource Planning (ERP).....	7
2.3 Industry 4.0 (I4.0)	7
2.3.1 Enabling Technologies.....	9
2.3.2 Industry 4.0 implementation	10
2.3.3 Systems integration.....	11
2.3.4 Cyber-physical systems (CPS).....	13
2.3.5 Reference Architecture Model of Industry 4.0 (RAMI 4.0)	13
2.3.6 Reference Architecture (IIRA)	17
2.3.7 IIRA and RAMI 4.0 interoperability	21
2.4 Technologies.....	24
2.4.1 Database.....	24



2.4.2	Structured Query Language (SQL).....	24
2.4.3	Conceptual database design	25
2.4.4	Logical database design.....	25
2.4.5	Physical database design.....	26
2.4.6	Views	26
2.4.7	MS Office Access	26
2.4.8	Microsoft .NET and Microsoft .NET Framework	27
2.4.9	Microsoft Visual Studio .NET	27
2.4.10	Object Relational Mapping.....	27
2.4.11	ActiveX Data Objects, ADO.NET and Entity Framework	27
2.4.12	Simulation.....	28
2.4.13	Knime Analytics platform	28
2.4.14	Power BI	28
3.	Company Profile.....	29
3.1	ZF Group	29
3.2	Company Information systems.....	31
4.	Problem Description	32
5.	Proposed Improvements	32
5.1.1	Contribution	32
5.1.2	Development of the conceptual model	33
5.1.3	Development of the logical model.....	34
5.1.4	Development of SQL physical model	34
5.1.5	Integrating SQL and Visual Studio	35
5.1.6	Knime and Simio Analysis.....	36
5.1.7	Tools for decision-making towards Industry 4.0 – under development.....	37
6.	Conclusion.....	39
6.1	Difficulties	39
6.2	Overview.....	39
6.3	Future works.....	40
	Bibliography.....	43



Annex I – Data Dictionary: entities description	47
Annex II – Data Dictionary: relationship description	47
Annex III – Data Dictionary: attributes description.....	47
Annex IV – referential integrity	50
Annex V – SQL structure statement	51
Annex VI – SQL views	51
Annex VII – C# input class.....	53
Annex VIII – C# department class.....	53
Annex IX – C# priority class	53
Annex X – C# localization class	54
Annex XI – C# issue class	54
Annex XII – C# project code	55
Annex XIII – Graphical User Interface	68



LIST OF FIGURES

Figure 1 - The four stages of the Industrial Revolution (source: Ślusarczyk, 2018).....	4
Figure 2 - PLC (source: Alphonsus & Abdullah, 2016)	6
Figure 3 - The McKinsey Digital compass for Industry 4.0 (source: Mckinsey, 2016).....	11
Figure 4 - Industry 4.0 vertical integration (source: i-SCOOP, 2019).....	13
Figure 5 - RAMI 4.0 (source Federal Ministry for Economic Affairs and Energy, 2018)	14
Figure 6 – RAMI4.0 Layers basic questions	16
Figure 7 - IIRA constructs (source: Lin et al., 2019).....	18
Figure 8 -IIRA viewpoints (source: Lin et al., 2019).....	18
Figure 9 - Functional domains (source: Lin et al., 2019)	20
Figure 10 - Functional Domains, Crosscutting Functions and System Characteristics (source: Lin et al., 2019).....	21
Figure 11 – Correspondences between architectures (source: Lin et al., 2018)	22
Figure 12 - IIRA and RAMI4.0 interoperability (source: Lin et al., 2018)	23
Figure 13 Occupant safety systems worldwide (adapted from ZF Friedrichshafen AG, September 2017).....	30
Figure 14 - Million units sold in 2018.....	30
Figure 15 - ER Diagram	34
Figure 16 - view for location and lines	35
Figure 17 - employees table	35
Figure 18 - entity framework	36
Figure 19 - simulation model	37
Figure 20 - simulation experience	37
Figure 21 - knime workflow.....	37



ABBREVIATIONS AND ACRONYMS

CPS Cyber-Physical System

CPPS Cyber-Physical Production System

DBMS Database Management System

ERP Enterprise Resource Planning

IDE Integrated Development Environment

IIRA Industrial Internet Reference Architecture

IIoT Industrial Internet of Things

IoT Internet of Things

MES Manufacturing Execution System

OPC-UA Open Platform Communications Unified Architecture

ORM Object Relation Mapping

PLC Programmable Logic Controllers

RAMI 4.0 Reference Architecture Model Industry 4.0

RFID Radio Frequency Identification

SCADA Supervisory Control and Data Acquisition

SOA Service Oriented Architecture

SQL Structured Query Language

GUI Graphical User Interface



1. MOTIVATION AND BACKGROUND

The present dissertation emerged under the scope of the master's in systems engineering. This project was developed at ZF Passive Safety Division in the process engineering department of Ponte de Lima.

This division is growing at a fast pace entailing huge data processing. The products sold to the final client are assembled from the manufacturing process pass through a rigid process of data traceability. Those data are stored in Microsoft SQL Server databases and tables.

Handling difficulties in information management have been an important task to any company's grow.

The difficulties were perceived and motivated this research. In this project, the main purpose is the integration of the in-company information and with this organized information, improve decision-making.

Global competition is becoming fiercer every day. Countries in Europe, Asia and the U.S.A have been recognizing the trend to deploy new digital technologies in the manufacturing Industry. This emphasizes that the fourth industrial revolution is close.

In Germany, "Industrie 4.0" represents a revolution in Information and Communication Technologies (ICT) where Internet of Things, Cyber-Physical Systems, Industrial Information and Integration are crucial to enable a transformation on how to optimize work and improve decision-making.

Nowadays organizations aim to manage their business, clients and operations using data analysis provided from different sources as sensors, web applications, social media and wearable devices. According to Russom (2011), these analysis and sources are called "Big Data".

The organizations that holds more knowledge and meet their client expectations have more opportunities to grow in the market. However, organizations find it difficult to storage and process data.

1.1 Objectives

The objectives of this dissertation are:



1. Literature review of Industry 4.0. This review will gather the necessary theory concepts that will be the base for the architecture implementation – from the history context of industrial revolutions to today’s disruption, the study of reference models for Industry 4.0 and the importance of standards using relational databases.
2. The actual industrial scenario for the case studied.
3. Understand how pieces of technology fit together to form a business solution:
 - How database tables, from different schemas, connect with each other;
 - How database tables are populated by the user using a Graphical User Interface (GUI);
 - How to create the GUI via programming in C# and connect with databases;
 - Gather information for decision-making.

1.2 Structure of the Dissertation

This document is divided into 6 chapters. The first chapter comprises of an introduction, motivation and objectives of this document.

The following chapter contains a literature review of Industry 4.0: the historic background, how can information created in the shop floor can reach the enterprise managers for decision-making, the enabling technologies, horizontal, vertical and end-to-end integration, Cyber-Physical Systems, reference architectures (RAMI 4.0 and IIRA) and the interoperability between these architectures.

This second chapter also contains a literature review of technologies that motivated this research: structure query language (SQL), database conceptual, logical and physical designs, database views, Microsoft office Access, .NET and .NET Framework, Visual Studio, Object Relational Mapping (ORM), ActiveX Data Objects, Entity Framework, Simulation and Knime Analytics platform.

Chapter 3 there are information about the company that provided the tools and knowledge that enabled this project. This chapter describes the information system the plant has.

In chapter 4, the problem description is introduced and, in the following chapter, the improvements to avoid ambiguity are proposed.

The last chapter refer to the difficulties observed during the literature review of Industry 4.0 and the overview of this project while integrating information.



2. LITERATURE REVIEW

As stated by Mulrow (2014), a systematic literature review has the following benefits:

1. Reduces the quantity of information into small pieces that are more likely to be understood by the reader;
2. Integrates pertinent pieces of information for decision-making;
3. It reduces the cost of creating a new research and can avoid survey of an already explored path;
4. Through multiple reviews it is possible to develop a generalization on the topic;
5. Evaluates the consistency of different variables;
6. Explains data inconsistencies and their correlation;
7. With more statically data, map future directions;
8. Qualitative analysis allows increased precision;
9. Improved reflection of reality.

The purpose of this topic is to show a summarized review of the actual industrial revolution, how we achieved today's state and how to achieve Industry 4.0.

2.1 Historic Background

It is important to understand how production evolved to the current scenario. According to Sabo (2015), the first industrial revolution started in 1760 and lasted up to 1830, where the mechanization changed the manufacturing process from hand production to machine production. These machines were powered by steam. Coal started to replace wood.

Machinery was powered by electricity in the second revolution, which took place between 1840 and 1870. Opportunities aroused to deliver huge amounts of goods from one place to another with the help of railroads and the capability of machinery combined with electric power led to the line production and mass production.

Between 1950 and 1970 the digital revolution took place. This was the third revolution, where digital and communication technologies were the beginning of the Information Age. Automation of the labour and the use of Information Technologies (ITs) characterized this disruption.



Nowadays any enterprise can gather great amount of information. As an example, it can acquire information from all their customers, their needs and how they manage their social life. This digital era is enabling another industrial revolution. In Germany, this revolution is called “Industrie 4.0”.

Figure 1 summarizes the industrial revolutions:

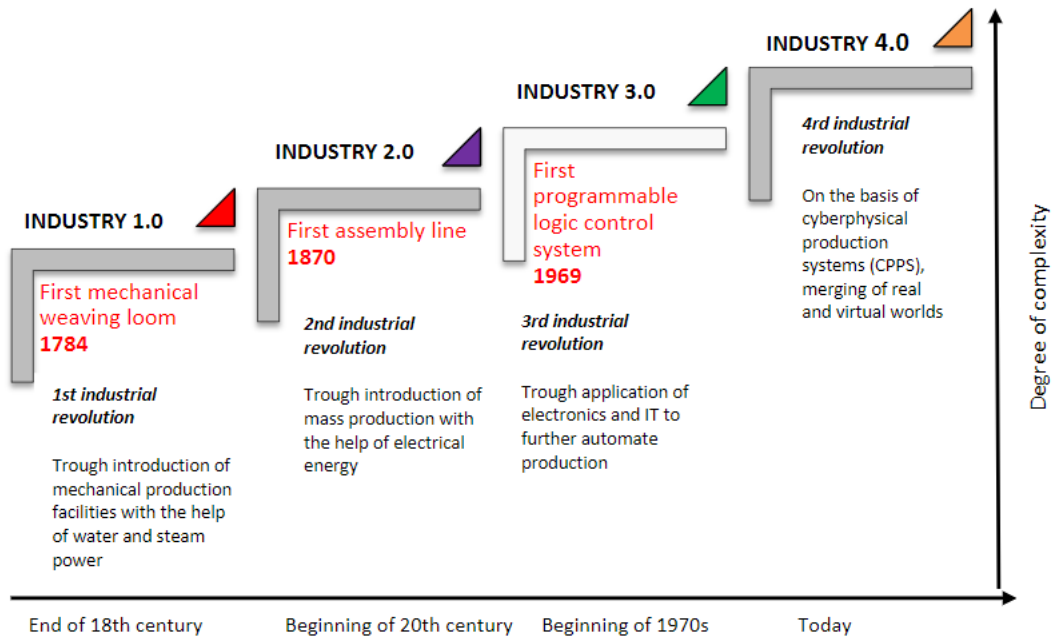


Figure 1 - The four stages of the Industrial Revolution (source: Ślusarczyk, 2018)

It is important to understand how programmable logic controllers, from the third industrial revolution control, works as they are a crucial technology to achieve industry 4.0.



2.2 Third Industry Revolution

2.2.1 Sensors

As stated by Macedo (2017), sensors in general transmits an electric signal from a received impulse. Their goal is to send a signal to the command center (that can be a programmable logic controller or other industrial computer).

2.2.2 Actuators

A piece of hardware dispositive that converts a signal from a controller to a physical magnitude. Actuators are responsible for actuation of mechanical systems, on summary actuators can convert electrical energy into mechanical energy (e.g., compressed air in pneumatics and oil in hydraulics) (Macedo, 2017).

2.2.3 Programmable logic controllers (PLC)

According to Alphonsus & Abdullah (2016) the first PLC came in the late 1960s in automotive industry. The authors state that PLC is the central component and are determinant in any autonomous system. The basic blocks are:

- Rack assembly;
- Power supply;
- Programming device;
- Input/Output sector;
- Central processing unit.

According to Macedo (2017) the PLC receive data from sensors, command buttons and other inputs, execute instructions from a program saved on his memory, and builds a system response by modifying the state of the outputs where all the actuators are connected.

Figure 2 exhibit sensors, PLC and actuators:



Figure 2 - PLC (source: Alphonsus & Abdullah, 2016)

2.2.4 Communication and OPC

Open Platform Communications (OPC) arrive two decades after the first PLC. OPC foundation defines standards

between communication from controllers regardless of the communication protocols used by their manufacturers (Macedo, 2017). According to the author, the primary goal is to transfer real data across PLCs, distributed control systems, Supervisory Control and Data Acquisition (SCADA), Human Machine Interface (HMI) and other production applications.

OPC data access depends on the technology DCOM (distributed component object model) with security limitations and closed to Windows platform. Those problems led the OPC Foundation to create a new protocol in 2006: OPC unified architecture (OPC UA), that is more flexible and adaptable (Macedo, 2017).

2.2.5 Supervisory Control and Data Acquisition (SCADA)

Supervisory Control and Data Acquisition (SCADA) are mainly associated with critical infrastructures that request high level of support in real time such as chemical processes, railways, water treatment and distribution and power plants (Macedo, 2017).



2.2.6 Manufacturing Execution System (MES)

According to Huang, Mohanty, Ashfahani, & Pratama (2018) a manufacturing execution system (MES) “is used to monitor the information flow from manufacturing objects on the manufacturing shop floor”. Manufacturing objects are work in process, workstation, tools and employees.

2.2.7 Enterprise Resource Planning (ERP)

According to Slack, Chamber, & Johnston (2009), ERP is an information system that integrate all enterprise information from different functions (and that information most of the times comes in high volumes), necessary to the planning and controlling of the operations. This integration occurs in a common database where all information can be shared.

As stated by the authors, this control informs when and where the activity must occur, who is responsible and how much capacity will be needed to complete the task and so on.

2.3 Industry 4.0 (I4.0)

Germany has global leadership in the manufacturing sector (Kagermann, Wahlster, & Helbig, 2013). Global competition is becoming more aggressive every year. Not only Germany is taking measures to mitigate this impact but also the U.S.A is avoiding deindustrialization with initiatives to promote advanced manufacturing (Fiesp, 2017).

The term “Industrie 4.0” (Industry 4.0) proposed in Hannover Fair in 2011 was announced in 2013 as a German strategic action to implement technologies mainly Cloud Computing, Internet of Things (IoT) and cyber-physical systems (CPS) in a way to revolutionize the manufacturing industry (Xu et al., 2018). As stated by the authors, Industry 4.0 collects information from sensors, artificial intelligence, data analytics to improve manufacturing work in real-time.

Industry 4.0 planning is to achieve all these three major integrations, with addition to hardware, software, data and information integration (Xu et al., 2018).

The term Industry 4.0 described in 2016 by Smit, Kreuzer, Moeller, & Carlberg is: “the organization of production processes based on technology and devices autonomously communicating with each other along the value chain: a model of the ‘smart’ factory of the future where computer-driven systems monitor physical processes, create a virtual copy of



the physical world and make decentralized decisions based on self-organization mechanisms”. With the increasing of digital manufacturing, physical objects are integrated into the information network.

As stated by Smit et al. (2016) for the success of Industry 4.0 there must be a standardization of systems, platforms, protocols, flexibility to adjust to new business models, security, research and investment, and a common legal framework to support the dissemination of Industry 4.0.

Wortmann, Combemale, & Barais (2017) describes Industry 4.0 as a “vision of manufacturing in which smart, interconnected production systems optimize the complete value-added chain to reduce cost and time-to-market”.

Stephen (2018) affirms that Internet of Things (IoT) is expected to generate “\$1.2 to \$3.7 trillion value globally by 2025, in four primary forms: 1) operational efficiency, 2) predictive and preventive maintenance, 3) supply chain management, and 4) inventories and logistics.

According to a survey from Mckinsey in 2016, most Industry 4.0 players (United States, Germany and Japan) expect cost and revenue increases of 10 to 15 percent.

According to Ślusarczyk (2018), the 4th Industrial Revolution is happening globally and concurrently. In Europe Fiesp (2017) presents the main initiatives of Industry 4.0 across the globe: UK (High Value Manufacturing, Innovative UK), Germany (Smart Service World, Autonomik Fur Industrie 4.0, Industrie 4.0), France (Usine du Future, The Nouvelle France Industrielle), Sweden (Produktion 2030), Italy (Fabbrica Intelligente), Belgium (Made Different), Netherlands (Smart Industry), Spain (Industria Conectada 4.0), Portugal (Produtech), Austria (Produktion der Zukunft), Poland (INNOMOTO, INNOLOT), Finland (Finnish Metals and Engineering Competence Cluster, Industry Internet Business Revolution, IoT Pilot Factory).

The US incentives their automatization with a more sustainable and digital industry (e.g. Advanced Manufacturing Partnership, Industrial Internet Consortium, National Robotics Initiative). In China the term “made in China 2025” is the Chinese plan to make their manufacturing industry a global leader and in 2015 Japan launched the Industry Value Chain forum to discuss Industry 4.0 and robotics.

Rodič (2017) states that today, the main enabling technologies and development tendencies in Industry 4.0 are:



- Horizontal and vertical system integration;
- The Internet of Things;
- The Cloud;
- Big Data and analytics;
- Green Information Technology;
- Augmented reality;
- Additive manufacturing (3D printing);
- Simulation modelling;
- Autonomous robots;
- Cybersecurity.

In summary, Industry 4.0 is the manufacturing process in an interoperability, adaptability, optimized and service-oriented approaches correlated with algorithms, big data, and high technologies.

2.3.1 Enabling Technologies

The main technologies that permits the implementation of Industry 4.0 are Internet of Things (IoT), Cloud Computing, Big data storage and analytics (Sony, 2018).

At the beginning of the Internet of Things, objects were connected using radio-frequency identification (RFID) technology. This enabled tracking in real-time objects identified with unique tags that were connected to the Internet. RFID is the keystone to IoT (Sony, 2018).

RFID made possible today's use of the IoT, aggregating this technology to global positioning systems, actuators, sensors operating through Wi-Fi, Bluetooth, Near Field Communication (NFC) or cellular networks (Xu, Xu, & Li, 2018).

According to Vermesan & Friess (2014), IoT is the new revolution of the Internet. The goal of IoT is to enable connection with anyone, anytime and anywhere using any path/network.

In 2011 the number of Internet-connected devices (ICD) surpassed the population. By 2020 ICD are expected to be a number between 26 billion and 50 billion (Vermesan & Friess, 2014) creating massive volume of data to be analysed.

This large volume of data lead to the term "Cloud Computing". This technology relies mainly on huge volume (Xu et al., 2018).



For instance, the design of a new product can be developed by engineers and fabricators in virtual community. Those individuals may not be in the same physical place (e.g., in-company) but the information can be shared.

The integration of IoT in Cloud platforms helps creating new services that would not be obvious without this high level of connectivity and analytical intelligence. The collection of a huge amount of data, results in the Big Data storage (Frank A G, Dalenogare L S, & Ayala N F, 2019).

According to Grable & Lyons (2018) the term Big Data means a large amount of information that requires a more robust data process than traditional methods. This complex data analysis may provide more precise insights showing patterns that could not be realized.

Together with analytics (like data mining and machine learning) it is considered one of the most important business drivers for Industry 4.0(Grable & Lyons, 2018).

Wortmann, Combemale, & Barais (2017) affirm that according to these disruptions, Industry 4.0 design principles are:

- Interoperability: connect people, production, devices and sensors;
- Information simplicity and interconnection: collect data from sensors, reduce or even eliminate data redundancy;
- Technical support: provide the right abstraction to understand the complexity of Industry 4.0;
- Decentralized decision making: enable self-sufficient systems.

2.3.2 Industry 4.0 implementation

McKinsey survey (2016) sustains that some manufacturers are struggling to implement Industry 4.0, as it is not clear to them what Industry 4.0 really requires. One concern of these manufacturers is the lack of courage to push through radical transformation.

Figure 3 represents tools and methods necessary to implement industry 4.0:



Figure 3 - The McKinsey Digital compass for Industry 4.0 (source: McKinsey, 2016)

One of McKinsey's conclusion for capturing value from Industry 4.0 is that manufacturers are achieving most progress when they focus on a limited number of Industry 4.0 applications and not all of them.

As stated by Fred Schulemann, Head of Digital Production at Audi AG: "We are aiming at an enormous progress in efficiency and speed particularly in engineering, controlling, administration, and decision making. But this value will only be unlocked if we manage to integrate data across the entire product lifecycle".

The World Economic Forum in 2015 (WEF, 2015) reported that 65% of companies in Europe and North America are finding it difficult to implement industrial digitalization due to the lack of interoperability or standards.

KPMG survey from 2017 (Gates & Bremicker, 2017) states that one key driver for manufacturers to implement Industry 4.0, is that the enterprise must plan ahead.

2.3.3 Systems integration



Sony (2018) declare that Industry 4.0 is possible using three kinds of integration:

- 1 Horizontal integration: collaboration between organizations in the value chain. With the use of digitalization, a new synergy for all the stakeholders is created. As stated by Xu et al (2018), horizontal integration handles several Information Technology (IT) systems used in different moments of the manufacturing.
- 2 Vertical integration: integration of several hierarchical systems within the organization are connected to the Enterprise Resource Planning (ERP) system. Xu et al. (2018) affirm that this integration work with various IT systems at different hierarchical levels. The different hierarchical levels are: shop floor, control, production and operations and enterprise planning.

For the shop floor management, there are sensors and actuators. To manufacturing controlling there are programmable logic controllers (PLCs). At the next level, there is a Supervisory Control and Data Acquisition (SCADA) System. A manufacturing execution system (MES) will be for production planning and quality management. On top there is an Enterprise Resource Planning (ERP) to integrate all data from an organization into one system (i-SCOOP, 2019). The vertical integration is presented in Figure 4.

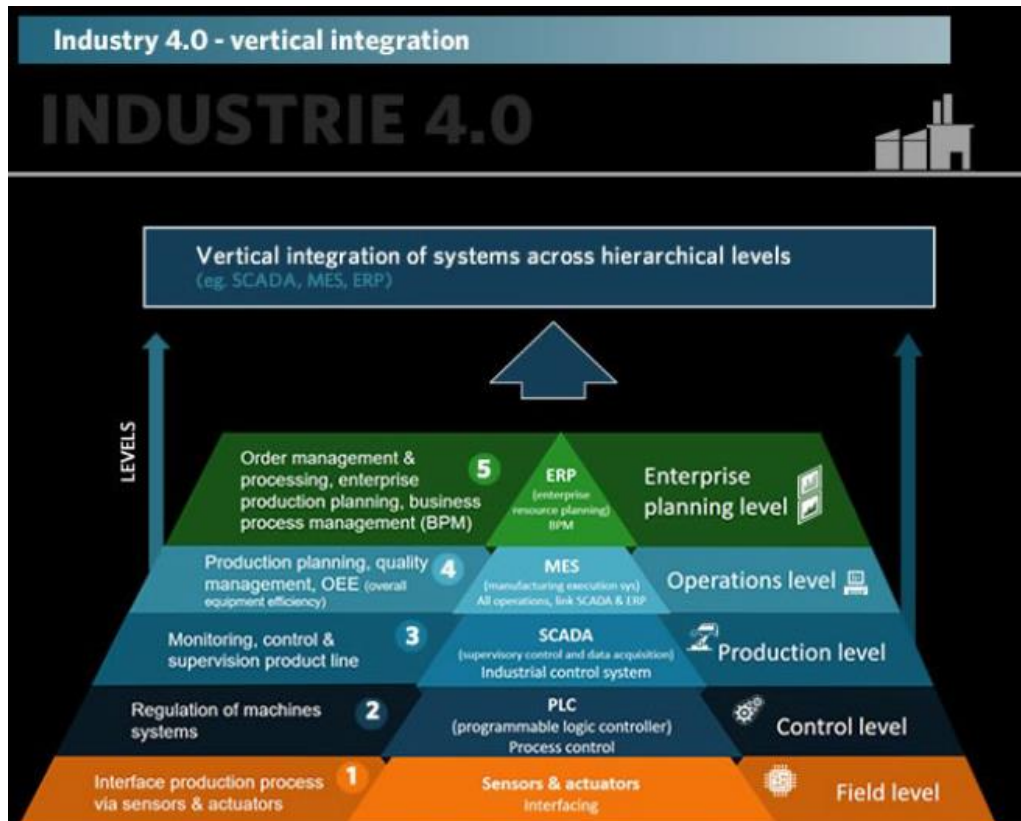


Figure 4 - Industry 4.0 vertical integration (source: i-SCOOP, 2019)

- 3 End-to-end engineering integration: According to Xu et al. (2018) this process aggregate digital and real world, information from the entire value chain, different companies, and customer requirements. According to Adolphs et al. (2015) end-to-end engineering means that technical, commercial and administrative data generated in the means of production are accessible at all times within the entire supply chain.

2.3.4 Cyber-physical systems (CPS)

“CPS are engineered systems that are built from, and depend upon the seamless integration of computational algorithms and physical components” (Xu et al., 2018).

According to Rodič (2017) with the use of CPS it will be feasible to coordinate the intelligence network of all subsystems in a central computer, making it possible to work with rising independence.

2.3.5 Reference Architecture Model of Industry 4.0 (RAMI 4.0)



As reported by the Federal Ministry for Economic Affairs and Energy in 2016, Germany vision is to encourage more businesses to get involved in Industry 4.0. To make it a reality, five work groups were created:

- 1) Reference Architectures, Standards and Norms;
- 2) Research and Innovation;
- 3) Security and Networked Systems;
- 4) Legal Framework;
- 5) Work, Education and Training.

The first group main development is the reference architecture model of industrie 4.0 (RAMI 4.0), certified and published as DIN (*Deutsches Institut für Normung*, German Institute for Standardization) specification 91345.

RAMI “enables the identification of relevant standards” and if no standard is used, RAMI helps to identify gaps in the missing architecture.

A common vocabulary between machines communication is the basic requirement to ensure interoperability in Industry 4.0 (Federal Ministry for Economic Affairs and Energy, 2016).

RAMI 4.0 is a three-dimensional map (Figure 5) showing the most important aspects of industrie 4.0 (Federal Ministry for Economic Affairs and Energy, 2018):

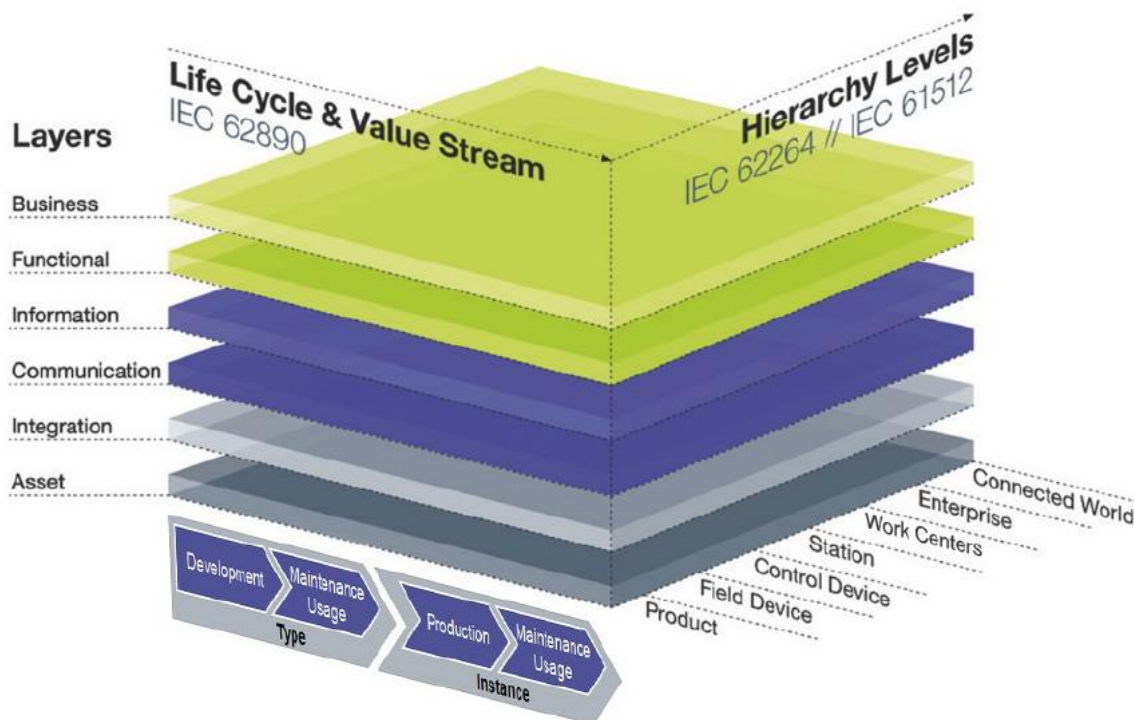


Figure 5 - RAMI 4.0 (source Federal Ministry for Economic Affairs and Energy, 2018)



1. Layers:

As reported by the Federal Ministry for Economic Affairs and Energy (2018) “this correspond to IT thinking where complex projects are split into clusters of manageable parts”:

- 1.1. Business layer: represents the business vision, models, economic market structures, organizational policies (legal frameworks) and provides a link between different business processes. It is the bridge to the functional layer.
- 1.2. Functional layer: permits changes of information between real and virtual worlds. An Enterprise Resource Planning (ERP) system typically will be located under this layer as well as rules and decision-making logic. It also permits access to horizontal integration.
- 1.3. Information layer: Ensures data integrity, how data is utilized and changed between processes, and obtains new and higher quality data. In this layer, we can have databases about the company’s products or data from the equipment used in operations.
- 1.4. Communication layer: Describes protocols standards, permitting communication between Programmable Logic Controllers (PLCs) in the shop floor and Manufacturing Execution Systems (MES) or ERP.
- 1.5. Integration Layer: Allow the integration between virtual world and real world. Contains elements connected with IT, such as RFID, sensors, routers, terminals, QR-Codes with the goal to convey the created content to asset layer.
- 1.6. Asset layer: Represents the real world, like machines and human beings. Humans are connected to the virtual world via the Integration Layer.

In Figure 6 platform Industrie 4.0 (“RAMI4.0 - a reference framework for digitalisation,” 2018) provide basic questions about the business idea:



Figure 6 – RAMI4.0 Layers basic questions

2. Life Cycle and Value Stream:

Relationships and links for the product, machines, and factories life-cycles are designed in this dimension. The life-cycle differs between type and instance. Type is the initial idea, generating a prototype that after testing and validation are produced and become an instance. The instance has a unique serial number. These are sold to final customers. If the product needs an improvement, the product goes back to type stage (Federal Ministry for Economic Affairs and Energy, 2018).

3. Hierarchy levels

It represents the different functionalities within factories or facilities. This dimension describes the functional classification of various situations of Industry 4.0 (Federal Ministry for Economic Affairs and Energy, 2018).

Products are the electronic devices (e.g. machines). Field device are sensors to gather data. Control devices are the main component of the machine, like PLCs. Station analyse data in real time to monitor the system. Work centers may control the production state, requiring new components and restoring the production goals. Enterprise is the business management like planning and control, statistics, marketing, sales. Connected world is the communication between stakeholders, making it possible the information sharing (Carvalho, 2018).



IEC stands for International Electrotechnical Commission and provides norms used in Industry. IEC 62264 are for enterprise IT and control systems. IEC 62890 are for life-cycle management and IEC 61512 are for batch control (Gotze, 2016).

Lydon states that a service-oriented architecture (SOA) is defined in RAMI 4.0 where application components provide services to the other components through a communication protocol over a network. Xu et al. (2018) states that OPC-UA (Ole for Process Control Unified Architecture) protocols establish a communication between Industrie 4.0 layers. OPC-UA supports service-oriented architecture (SOA) that integrate platforms, improve flexibility and reduce the gap between enterprise architecture and information and communication technology (ICT) infrastructure redesign.

2.3.6 Reference Architecture (IIRA)

RAMI 4.0 is not the only standard for Industry 4.0. Industrial Internet Reference Architecture (IIRA) from Industrial Internet Consortium (IIC) was designed by AT&T, Cisco, GE, IBM and Intel in march 2014 with the goal to connect and integrate people, processes and data (Gotze, 2016).

The Industrial Internet Architecture Framework “identify and classify stakeholder concerns into appropriate categories” (Lin et al., 2019).

According to IIC, at IIRA core are viewpoints that specify system concerns. Concern indicate problems related to the system. Model kinds are modelling constructs to resolve those concerns. Architecture views and architecture models can be considered as the representation of architecture.

Figure 7 summarizes the framework, feedback and improvement for IIRA:

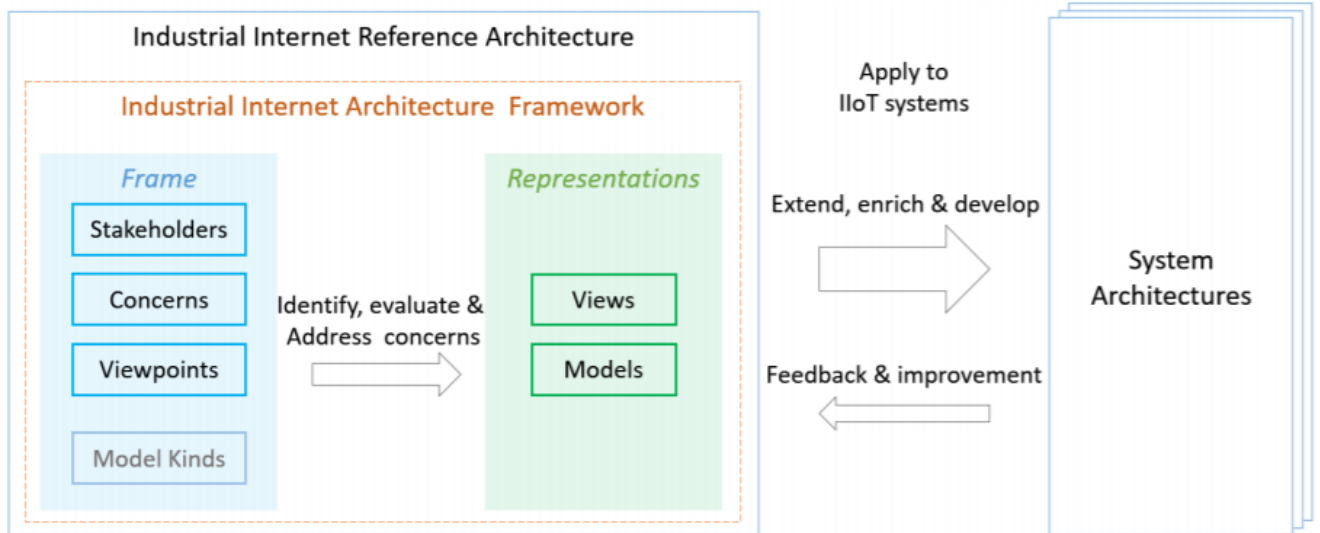


Figure 7 - IIRA constructs (source: Lin et al., 2019)

According to IIC, views are the starting point for concrete architecture and may be replaced by better ones according to the needs of the IloT (Industrial IoT) system at hand.

The four viewpoints defined by IIRA fitting IloT concerns are (Figure 8):

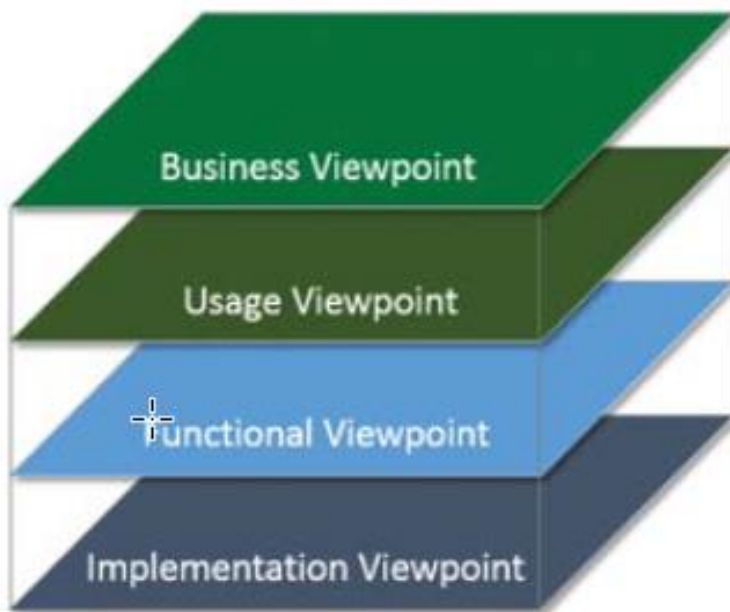


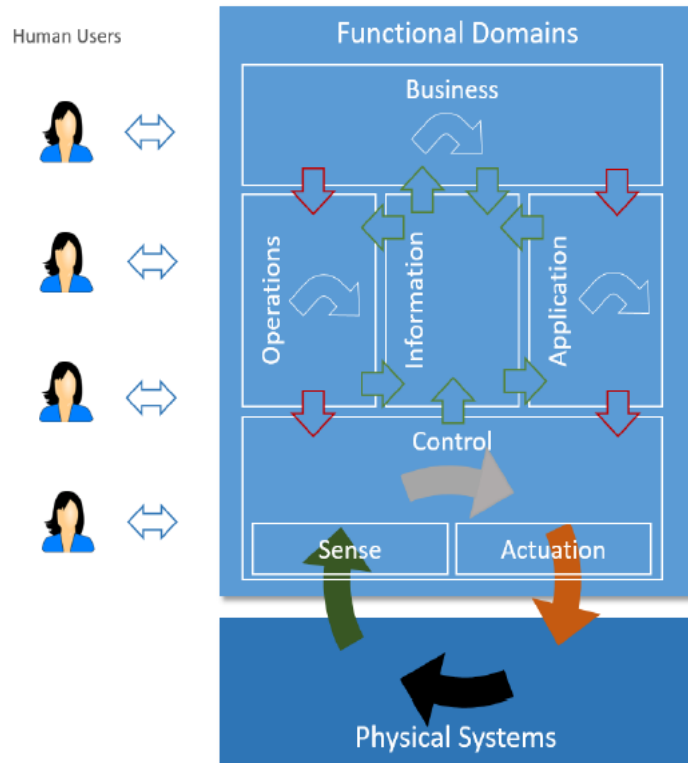
Figure 8 -IIRA viewpoints (source: Lin et al., 2019)

1. Business: Identify stakeholders and their business vision, values and objectives in setting up an IloT system and how IloT could improve the enterprise activities. Values and objectives emerge from this vision. This viewpoint is about enterprise motivation and who benefits from IoT system.



2. Usage: This viewpoint focuses on how to address the concerns described in the business viewpoint. These concepts are developed to coordinate activities of the system:
 - 2.1. Tasks: Basic unit of work connected to operation, data transfer or action. The role (set of capacities assumed by an entity to do some tasks required by an activity), functional map and implementation map are concepts of a task.
 - 2.2. Activity: Coordinate different tasks. An activity can be executed recursively having four elements: trigger (how the activity will be initiated), workflow (sequential, parallel, conditional, iterative organization of tasks), effect (the phase of the IIoT system after successful completion of an activity) and constraints (characteristics that must be maintained like integrity and reliability).
 - 2.3. Party: Agent, human or robot that have interest in the execution of a task.
3. Functional: How the IIoT system will interact and interfaces between external environment and other systems to support the usages and activities of the overall system. A typical IIoT system is decomposed into five functional domains:
 - 3.1.1. Control Domain: Is the domain for implementing industrial control system. It reads data that comes from other systems, determine component rules and control those rules using actuators. An example is the control unit in autonomous vehicle.
 - 3.1.2. Operations Domain: Management and operation of the control domain with the goal to optimize activities and generate value for stakeholders.
 - 3.1.3. Information Domain: Manages and process data to acquire high-level intelligence about the overall system.
 - 3.1.4. Application Domain: Is the domain for implementing logic.
 - 3.1.5. Business Domain: Functions for integrating information across business systems and applications to achieve business objectives. Examples are ERP, Customer Relationship Management (CRM), Manufacturing Execution System (MES), Human Resource Management (HRM) and Product Life Management (PLM).

Functional domains are summarized in Figure 9.



Green Arrows: Data/Information Flows; Grey/White Arrows: Decision Flows; Red Arrows: Command/Request Flows

Figure 9 - Functional domains (source: Lin et al., 2019)

4. Implementation: Integration is the key topic on this viewpoint. It presents the communication schemes, technologies, lifecycles to coordinate activities (usage viewpoint) and supportive of system capabilities (business viewpoint), i.e. technical representation.

IIRA uses ISO/IEC/IEEE 42010:2011 (system and software engineering-architecture description) standard.

Additional functions must exist to enable a non-generic IIoT. These functions must be available across many of system functional, that are called crosscutting functions, like connectivity or data management and new characteristics will emerge when the system is integrated. Figure 10 represents the integration of the three aspects of industrial internet reference architecture:

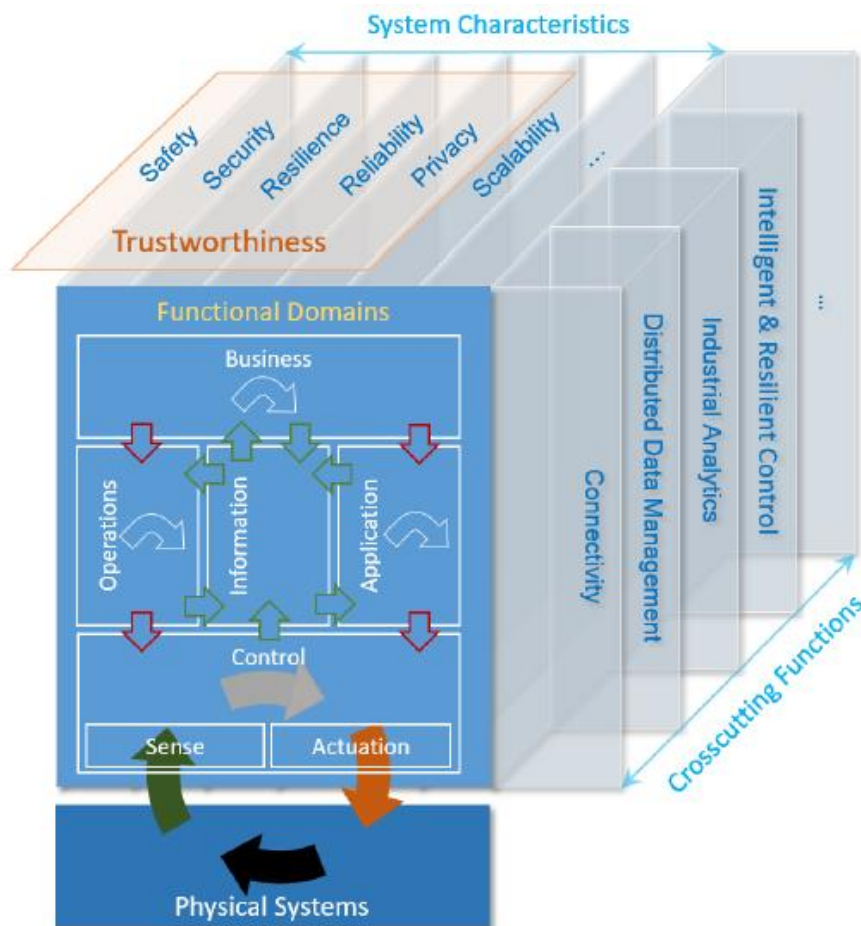


Figure 10 - Functional Domains, Crosscutting Functions and System Characteristics (source: Lin et al., 2019)

2.3.7 IIRA and RAMI 4.0 interoperability

(Fiesp, 2017) states that representatives of Industrie 4.0 in Germany are meeting with representatives of Industrial Internet Consortium from U.S.A to discuss the technological interface standards.

In Germany and the U.S.A there have been an increase on creation and usage of testbeds to simulate productive systems and evaluate the reference architecture (Fiesp, 2017).

As reported by the Federal Ministry for Economic Affairs and Energy at the beginning of march 2016, Industrie 4.0 started to cooperate more closely with IIC. The former initiative focuses on industrial digitalization while the latter focus on Internet of Things world, like energy, healthcare and transportation, stressing cross-industry communication and interoperability. Both initiatives complement each other as they both aim at obtaining intelligence data from processes while enabling interoperability among systems.

IIRA and RAMI4.0 have different emphasis in scope and depth, but these different perspectives can complement each other in addressing challenges into digitalization. “For



example, industrial analytics, an idea that is emphasized in IIRA would be beneficial for building intelligent Industrie 4.0 systems” (Lin et al., 2018).

Comparison between IIRA and RAMI 4.0 are done in Figure 11.

IIRA crosscutting functions and domains	RAMI4.0
<i>Physical Systems</i> (not formally defined)	Physical aspect of the <i>Asset Layer</i> Physical components, documents, software and human actors
<i>Control Domain</i> Functions performed by the industrial assets or control systems executing closed-loop control that may involve sensing, control and actuation.	<i>Integration Layer</i> Digitally represents the physical assets and their digital capability; Provides computer-aided control of the technical process; generates events from the assets <i>Functional Layer</i> If the functions are Industry 4.0 compliant then they refer to the functional layer
<i>Connectivity and Distributed Data Management and Integration Crosscutting Function</i>	<i>Communication Layer</i> Provides standard communication for services and event/data to the information layer Provides control commands to the integration layer
<i>Information Domain</i> Functions for collecting, transforming and analyzing data to acquire high-level intelligence of the entire system	<i>Information Layer</i> Describes (logical) services and events/data of an asset (technical functionality) with regard to its role in the Industrie 4.0 system (as a “semantic self-description”)
<i>Operations and Application Domains</i> Functions for assets and control systems management and maintenance to ensure their continuing operations Functions for applying use-case-specific logic, rules and models based on the information obtained from the information domain to achieve system-wide optimization of operations	<i>Functional Layer</i> The runtime environment for applications and technical functionality enabling tasks in support of the business processes
<i>Business Domain</i> Functions for integrating information across business systems and applications to achieve business objectives, such as work planning, customer relation management (CRM), enterprise resource planning (ERP), manufacturing execution system (MES), etc.	<i>Business Layer</i> Orchestrating functions to form business processing and linking between different business processes in supporting of the business models under the legal and regulatory constraints

Figure 11 – Correspondences between architectures (source: Lin et al., 2018)

Interoperability can be defined as “the ability of two or more systems or components to exchange information and to use this exchanged information” (Blanc-Serrier, Ducq, & Vallespir, 2018).

Figure 12 represents the interoperability between the two references architectures studied:

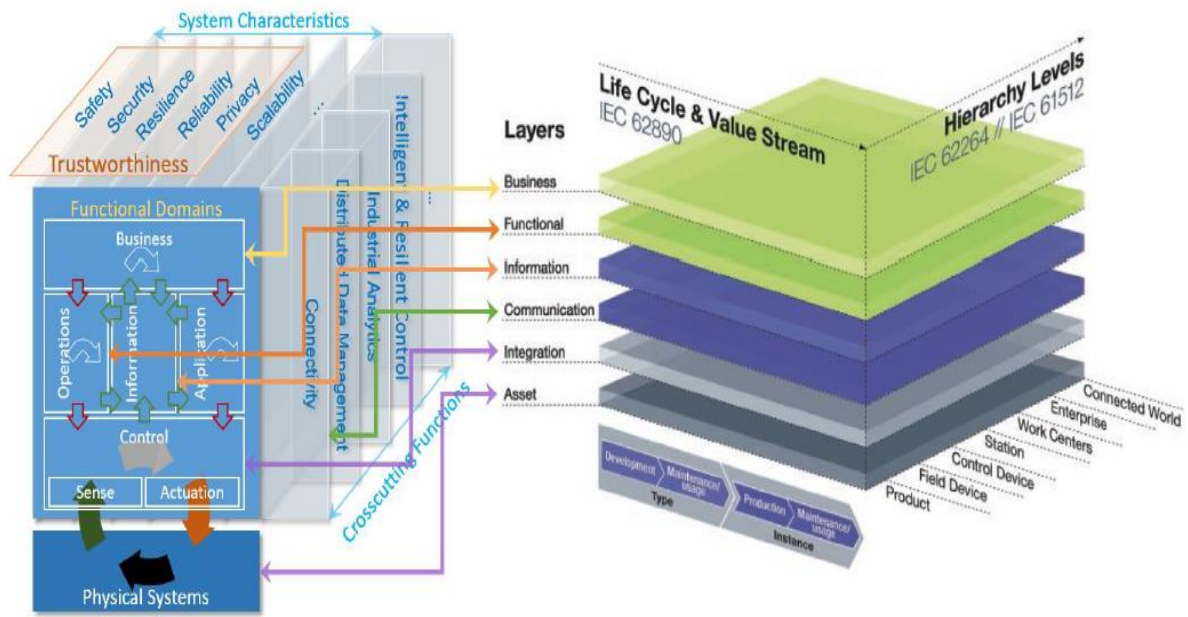


Figure 12 - IIRA and RAMI4.0 interoperability (source: Lin et al., 2018)



2.4 Technologies

This topic presents the importance of the tools that made possible the contribution. All tools were developed in Windows platform.

2.4.1 Database

This technology can aid an optimized approach where data consistency will enable a readily data analysis to support decision-making.

Database architecture methodology assist the planning, management and control, modelling the base requirements in a standard (Connolly & Begg, 2005).

The relational database is being used for more than 30 years all over the world. This kind of database changed significantly the way the productivity of organizations. The development of this technology created a more intuitive use. Therefore, the database is not a department property, but a resource shared throughout the enterprise (Connolly & Begg, 2005).

Cristian (2017) affirms that the database involves share of data and applications and present countless benefits, like:

- Reduce and even eliminate data redundancy using data integrity, thus avoiding copies of the same data. With this minimization, data consistency may be maximized.
- Security measures to avoid data manipulation from people without authorization.
- The database allow access by multiple users at the same time.

The database management system (DBMS) is a software that allow the creation, maintenance, control and access to databases. DBMS interacts with applications created by the user that can be written in procedural language (that describes how to do something) coming from the third generation of programming language (3GL) like C, C++, C#, Java, Visual Basic, COBOL, Fortran, Ada and Pascal or through the integration of programming languages from the fourth generation (4GL) (that describes what shall be done) with 3GL like the structure query language (Connolly & Begg, 2005).

Using this integration there is a segregation between data structure (where the data is stored) and applications (Connolly & Begg, 2005).

2.4.2 Structured Query Language (SQL)



According to Connolly & Begg (2005), SQL is a language developed to use the relationships between tables and transform inputs in outputs using two components:

- Data definition language – DDL to define the database structure and control the information;
- Data manipulation language – DML to retrieve and update data.

SQL is a global standard to define and manipulate relational databases (Connolly & Begg, 2005).

2.4.3 Conceptual database design

The conceptual database design is independent of all implementation details such as: hardware, software, applications and programming languages (Connolly & Begg, 2005).

According to Connolly & Begg (2005) a conceptual model comprises:

- Entity types;
- Relationship types – using entity-relationship modelling;
- Attributes and their domains: a user can define the meaning and sources of the values that attributes can hold;
- Primary and alternate keys: primary key identifies rows uniquely within the relation;
- Integrity constraints: which ensures that the data is accurate, for example: no primary key can be null (cannot have an unknown value).

Entity-relationship model grant data integrity and their adequate usage to solve a problem from the company. It creates a model free of ambiguities and avoiding the differentiated understandability between architects, programmers and final users (Connolly & Begg, 2005). Data dictionary is produced thought out the development of the model. This dictionary identify entities, relationship and attributes description (Connolly & Begg, 2005).

2.4.4 Logical database design

The logical database design phase maps the conceptual model onto a logical model. The logical model is influenced by the data model (for example, relational model) but is independent of a particular DBMS and other physical considerations (Connolly & Begg, 2005).



According to the authors, relational databases have certain amount of data redundancy. This redundancy is in the form of a primary key from table A, acting as foreign key in table B, to enable modelling relationships.

Also stated by the authors, the derived structures from the conceptual model relationships, are identified in the logical model:

- One-to-many (1, N) relationship: parent entity must be identified as well as child entity. To represent the relationship using a foreign key, the primary key from the parent entity will be a foreign key in the child entity.

Connolly & Begg (2005) affirms that integrity constraints are represented in logical model. These actions can be performed in the child entity if a value from its parent entity is modified:

- NO ACTION is to prevent the update or deletion,
- CASCADE is to automatically update or delete any reference child row when parent row is deleted,
- SET NULL command automatically set a foreign key to null.

2.4.5 Physical database design

The physical database design is tailored to a specific DBMS. There is feedback between physical and logical design as the decisions from the physical model can alter the logic behind the database (Connolly & Begg, 2005).

SQL enable multiple users input at the same time (Connolly & Begg, 2005).

2.4.6 Views

The view is a virtual relation that does not exist in the current database, but can produce a new table with useful information (Connolly & Begg, 2005).

They are dynamic, implying that changes made in the base data table will be updated automatically in the view (Connolly & Begg, 2005).

2.4.7 MS Office Access

MS Office Access or MS Access is used for relational DBMS for Microsoft Windows environment. It provides a Graphical User Interface (GUI) to create tables, queries, forms, reports and customized applications using macro in Visual Basic for Applications (VBA)



language, it also provides a “wizard” to simplify the process for building applications (Connolly & Begg, 2005).

2.4.8 Microsoft .NET and Microsoft .NET Framework

Microsoft .NET is the current evolution in Microsoft’s Web solution strategy. Microsoft .NET have various tools, services and technologies such as Windows Server, business processes with xml, application center, mobile server, object-relational database management system and Microsoft Visual Studio .NET (Connolly & Begg, 2005).

According to the authors, Microsoft .NET Framework, in addition to .NET, has two main components that are:

- Common Runtime Language (CRL): loads, executes and manages codes like C#, C++, F#, J#, Java, Visual Basic. In CRL one language can call another language, inherit and modify objects from another language.
- .NET Framework Class Libraries: is a collection of interfaces, types and classes that integrate with CRL providing standards. The three main components are: windows forms, APS.NET (to support Web development) and ADO.NET.

2.4.9 Microsoft Visual Studio .NET

Microsoft Visual Studio .NET is an integrated suite of application development tools such as J#, C#, C++) (Connolly & Begg, 2005). This integrated development environment (IDE) includes many aspects of software development such as compilers, code completion tools, graphical designers(visual-studio-ide @ docs.microsoft.com, 2019).

2.4.10 Object Relational Mapping

Object relational mapping is a technique that abstracts the communication between two different programming paradigms like, object-oriented models and relational databases Various technologies were offered to handle this gap between object-oriented and relational databases, like Entity Framework from Microsoft (Zimányi, E., Jallow, B., Kashef, 2018).

2.4.11 ActiveX Data Objects, ADO.NET and Entity Framework

ActiveX Data Objects (ADO) is a programming tool for database connection (Connolly & Begg, 2005).



In ADO just one request to a database can be made using a single connection. On the other hand, ADO.NET that is designed by .NET framework can request multiple transactions using a single connection (ado-difference @ net-informations.com).

Entity framework is an open source ORM framework for ADO.NET which simplifies mapping between objects to columns of relational database (Tannock, 2016).

As referred by Tannock (2016), entity framework let developers focus on the business domain. The application cannot be limited by what the database is able to do.

2.4.12 Simulation

For simulation, it was used the software Simio. According to Vik, Dias, Pereira, & Oliveira (2012) Simio supports:

- 3D animation;
- Importing data from excel worksheets;
- Writing own logic functions in languages like C++ and Visual Basic;
- Creating own libraries and objects.

2.4.13 Knime Analytics platform

For workflow, it was used the Knime Analytics platform (Konstanz Information Miner). This environment allows easy visual assembly, dynamic data analysis and data cleaning. It is suited to process files in csv, xls, xlsx, xml, url and relational databases (eg., SQL, MySQL, Oracle) (Feltrin, 2015).

2.4.14 Power BI

Power BI converts all data that comes from reports, worksheets, charts and emails into charts and graphs that are more meaningful to the reader (power bi @ docs.microsoft.com, 2019).



3. COMPANY PROFILE

3.1 ZF Group

The following topic presents the company profile where this master thesis was developed.

In the year 2000 was founded Safelife, a bag plant part of Dalphimetal Group. In May of the same year, in Vila Nova de Cerveira started the production, research and development activities. In November occurred the delivery of the first bag to the group PSA (known as PSA Peugeot Citroën from 1991 to 2016).

In January 2001 started the construction of the facility at Industrial area in Gemieira, Ponte de Lima. In October, the production was transferred to new facilities in Ponte de Lima.

In January 2004 was founded Safebag (airbags modules assembly plant) and in the same month was acquired a building in Fornelos, Ponte de Lima. In April 2004 started the production and delivery of the first airbag to Renault.

In October 2005 Dalphimetal group was acquired by TRW Automotive. In February 2008, Safe Life and Safebag lines were reorganized by products and were integrated as Ponte de Lima Plant.

In May 2015 TRW Automotive was acquired by ZF Friedrichshafen AG. Ponte de Lima plant integrated the ZF Active and Passive safety technologies division.

ZF Group have seven divisions:

- 1) Car Powertrain: Manual, automatic and dual clutch transmissions, axle drivers and powertrain modules;
- 2) Car Chassis Technology: Chassis systems, chassis components and suspension technology;
- 3) Commercial Vehicle Technology: Truck and Van driveline technologies, axle and transmission systems for buses and coaches and steering systems;
- 4) Industrial Technology: Off-Highway systems, industrial drives, marine and special driveline technology, test systems, aviation and wind power technologies;
- 5) E-Mobility: Electronic systems, electric traction drive and system house;
- 6) ZF Aftermarket: Independent aftermarket, original equipment service / specific original equipment, manufacturing, services and friction materials group;



7) Active & Passive Safety Technology: braking systems, steering systems, occupant safety systems, electronics and body control systems.

Figure 13 shows the worldwide ZF presence for occupant safety systems:

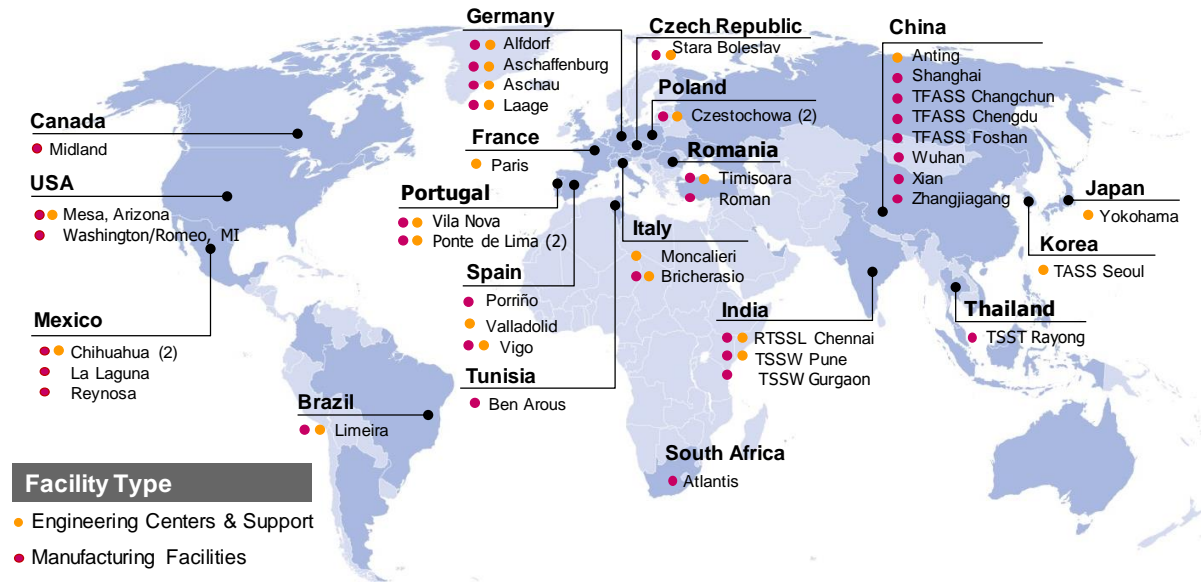


Figure 13 Occupant safety systems worldwide (adapted from ZF Friedrichshafen AG, September 2017)

In 2018, Safebag delivered approximately 14 million airbag modules. The clients were: PSA, Seat-Volkswagen, GM, Renault-Nissan, Volvo, Ford, Hyundai, Fiat and Jaguar. Figure 14 shows millions of units sold in 2018 among five airbags modules:

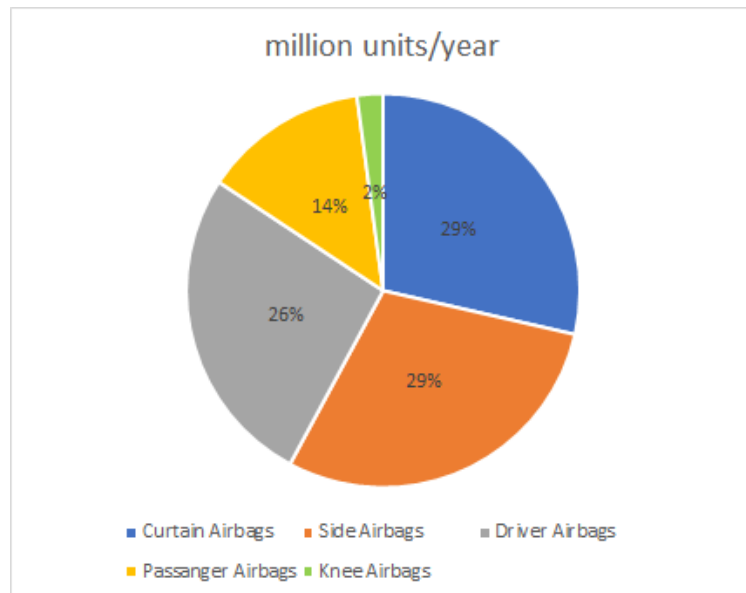


Figure 14 - Million units sold in 2018

According to Fernandes (2018) in 2017 ZF had a revenue of 36.4 billion Euros and Safebag had a revenue, at the same year, of 192 million Euros.



3.2 Company Information systems

Safebag modules are composed of information in different applications that are fragments of the overall picture of their manufacturing process.

The programming language used in assembly lines is Visual Basic and C#.

- One module that is called traceability register all finished products that comes from assembly lines, the hour they were finished and their date.
- Another one – documentation – registering all processes documentation.
- A third module – flexibility matrix – showing the level of formation from every employee, so the employee can operate in one specific line.
- A fourth module – on time requests – the operator can request maintenance or raw materials.
- The fifth module informs if the component necessary to assemble the airbag, requests quality control. Depending on the sensor output to the PLC, the assembly line may reject the component during assembling.

Most of the information are registered in Microsoft SQL Server, although some old lines are working with MS Access database.

Once finished goods are registered in SQL database this information goes to logistics in SAP (ERP), so the transport goes to the line to retrieve those finished products. Using this same logic, the operator can request more components as the products are finished. Even so, the operator must always read the labels of finished goods, so they can be registered in SQL database and the information goes to SAP.



4. PROBLEM DESCRIPTION

Problem solving of complex issues are not an easy task. The main issues detected are:

- Information duplication in different tables and different databases;
- Difficulties on the flow of information (sharing);
- Difficult in understanding the relationships between tables;
- Difficulty in obtaining meaningful data for decision-making.

Crossing the objective of this dissertation with the specific problems detected, it was possible to evaluate and propose some significant improvements.

5. PROPOSED IMPROVEMENTS

- **Standardization:** a standard permit share of information across various enterprise levels – being that vertically or horizontally. Information share is possible between machines once there is a standard. Across employees the information standardization is important because every person cannot deliver the data as they want.
- **Eliminate data redundancy:** one table from one database might gather all information necessary to other applications.
- **Data integration:** one table might not have all information to other application, but two tables might have that information. They must be related via programming so there is no need to create a third table with information from those two.

The topics below presents the companies modules and the contribution.

5.1.1 Contribution

Standardization and interoperability are the main concepts to make it possible to gather and share production lines or locations across responsible to act and fix those problems.

At the beginning, every department (six in total) had their vision and excel sheets to inform the process engineering department, making it hard to cope with those different views.

To avoid the creation of different sheets, the process engineering department implemented an excel spreadsheet to maintain all information in one standard. Even so, the excel sheet permitted any filling on the fields.



MS Office Access was used in the first weeks of this project and Knime, Konstanz Information Miner, was used to see if the data registered in database could provide decision-making dashboards.

The database used in the company was Microsoft SQL Server. Though C# programming and Windows Form, a graphical user interface was developed. Dashboard, that were easily understandable, were created in Power BI.

Some tools important to make this integration and already existed before this contribution:

- Databases with information regarding the production lines and employees;
- Shared network between employees.

5.1.2 Development of the conceptual model

Issues were identified during production lines auditing. Those issues have a responsible for implementing an action to resolve the issue. This issue must have an end date and a priority.

The entities identified were:

- Input: How the issue was identified (auditing, quality inspection, plant tour).
- Department: Which department identified the issue.
- Priority: What is the priority to resolve the issue.
- Localization: Where the issue is occurring (production lines or external facilities e.g. parking lot).
- Issue: The problem description, the action provided, the responsible delegated to solve the issue and the concluded date of the solution.

In 1976 Peter Chen presented the entity-relationship model (ER model) (Connolly & Begg, 2005). Crow's feet notation redraws the ER model as show in Figure 15:



The DBMS used was Microsoft SQL Server. Previous databases were already created using this database management system.

The structure statement is in the annex.

5.1.5 Integrating SQL and Visual Studio

The tool for integrating the structure with procedural language was Entity Framework.

Some information already existed in the form of SQL tables, for example lines table. Creating a view concatenating location and lines (Figure 16) in SQL would enable this automation:

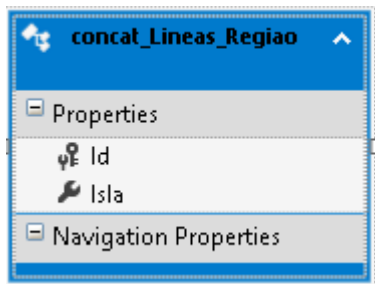


Figure 16 - view for location and lines

This tool also permits programming a new platform using different databases. This was the case of employees table (TPessoal, Figure 17). Any modification in this database would appear to the user:

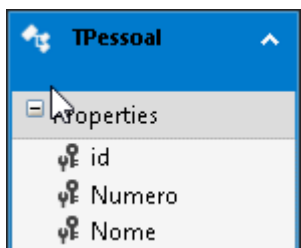


Figure 17 - employees table

Figure 18 is a mirror of the ER diagram created in SQL:

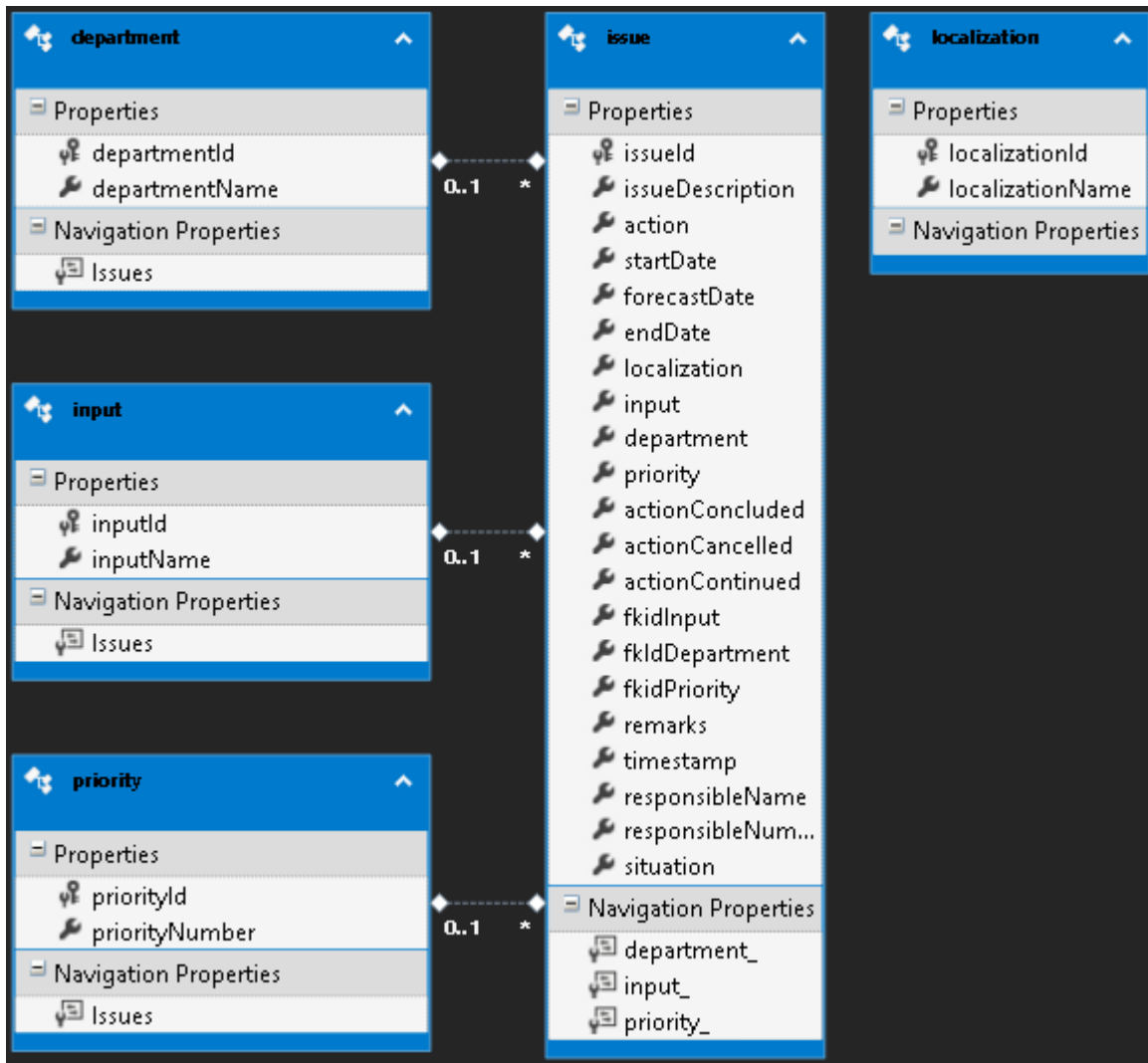


Figure 18 - entity framework

5.1.6 Knime and Simio Analysis

Simulation was used to compare the usage of the integrated system created in Excel and SQL. This experience compared the server capacity, where in Excel only one person could use the spreadsheet to create, read, update or delete issues whereas in SQL every employee can be at the same time using the system.

During this experience, done 200 times, both systems shared the same entities per arrival (1 to 52 employees use this integrating tool), the interarrival time and the processing time, where we see that the usage in SQL without any server considerations (e.g., the server can be inaccessible) shows more efficiency than Excel.

The model and experiment are show in Figure 19 and Figure 20.

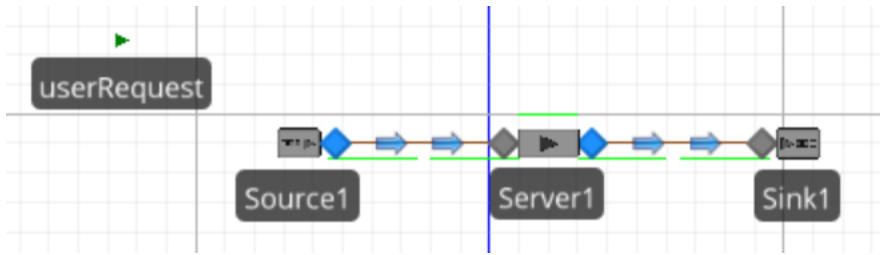


Figure 19 - simulation model

Scenario		Replications		Controls	Responses			
<input checked="" type="checkbox"/>	Name	Status	Required	Completed	initialCapacityServer	requests	exited	output
<input checked="" type="checkbox"/>	Scenario1	Idle	200	200 of 200	1	1278.24	40.815	3.20894
<input checked="" type="checkbox"/>	Scenario2	Idle	200	200 of 200	52	1268.91	1226.77	96.6867

Figure 20 - simulation experience

SQL had an efficiency of almost 97% and Excel almost 4%.

The initial capacity of 52 users in scenario2 from the experiment was acquired from the real usage of the integrated system developed in SQL, where 52 different employees made requests. The workflow created in Knime is as show in Figure 21:

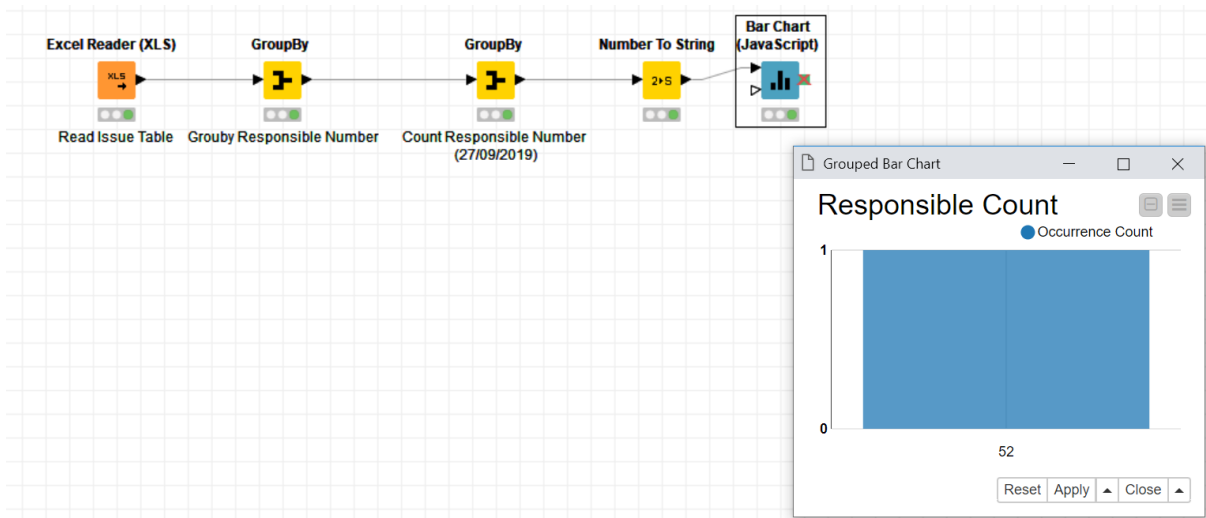


Figure 21 - knime workflow

5.1.7 Tools for decision-making towards Industry 4.0 – under development

The shop floor employees must assemble airbags modules as informed by the logistics module in their integrated system for enterprise management (SAP ERP).

Logistics informs the reference to produce, the quantity and the delivery date. The production presents the lines, the employees times, quantities and the traceability to check the planning and the real data.



A contribution for data processing and interoperability in Industry 4.0.

This integration is to stop using an Excel sheet where the shop floor employees enter the product they do as their will.

The tool was developed in web using PHP.



6. CONCLUSION

6.1 Difficulties

Several literatures focus on Industry 4.0 but according to Wortmann et al. (2017) “only few papers address modelling for smart product which is supposed to control its production processes in many visions on Industry 4.0”. Overall, 61% publications contribute with methods, 17% share concepts but less often publications commit with analyses, metrics or tools.

Handling technical issues while having a business background are challenging. To develop interoperability, tools and methods created in programming languages must be connected. Having just one background is not enough.

6.2 Overview

Structuring a reference architecture entails a high level of abstraction to gather specific stakeholders needs and technologies. Even so, the main requirements are simple to understand: connectivity and communication to collect data and disseminate information across the enterprise in order to improve decision-making.

As industrial revolutions happen, its complexity raises the skills needed by the works. Intellectual labour will become increasingly more important relative to physical labour.

Industry 4.0 will be an inevitable disruption to all global enterprises. Companies that act locally will also be impacted by this revolution. Industry 4.0 aims not only on smart manufacturing but also on increasing efficiency.

Enhance the creation of more favourable conditions for the best performance of industrial processes are needed in Europe not only to avoid the impact of the workforce bottleneck (where more people are leaving the market than people going in the market), but also to prevent the loss of industrial leadership to emerging economies.

All company must be involved in this revolution. Lack of employee engagement, skepticism, and fear can cause the failure of the project.

Industry 4.0 must be implemented at a slow pace but constant pace (this transformation may take years), although technological evolution still happens, and this development is at a fast pace.



How to implement Industry 4.0 constantly dealing with these technological changes are huge concerns to be taken into consideration. Machine learning (ML) and predictions are the second phase of Industry 4.0, but without a good architecture implemented in Industry 4.0 first phase, ML will not deliver feasible responses for decision-making.

The theme Industry 4.0 demands a continued study, not only because there are lots of documents, tools and methods to make Industry 4.0 possible, but also because it is highly connected with the Information Technology world. This world of information keeps its dynamics.

Reference architectures in Industry 4.0 are in development. There will not be a model that every company can follow and create his entire smart manufacturing from it. Industry 4.0 in railway or aviation sectors will not share the same approach as Industry 4.0 in automotive industry, simple because the final product is different. Even so, a standard architecture across companies must be implemented. One solution from one company may be related to the solution from another company.

The information system developed aims at integrating employees, the identified issues, and the responsible to act in resolving the issue. Once in a common database, it is easier to share information across departments.

6.3 Future works

The information created by the company must be measured, using data analysis.

Lord Kelvin affirms:

“When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be.” (Kaganski, Majak, Karjust, & Toompalu, 2017).

Knime analytics platform could be used for this task: organizing data (removing data that is not important to the analysis), statistics, data mining and machine learning with minimal or no programming.

The platform created in C# in Windows Forms must be developed in web for the following reasons:



- The SQL must be restructured. Some attributes must be implemented in their own table and new relationships must be created (for example, one issue can have more than one end date).
- In C# programming, an update might not be shared among users, causing some employees to have the newest version and some to have the older version.
- All users must have their credentials (login, password and email) while with every update there must be a traceability of who did what where.



A contribution for data processing and interoperability in Industry 4.0.



BIBLIOGRAPHY

- ado-difference @ net-informations.com. (n.d.). Retrieved from <http://net-informations.com/faq/ado/ado-difference.htm>
- Adolphs, P., Bedenbender, H., Dirzus, D., Ehlich, M., Epple, U., Hankel, M., ... Wollschlaeger, M. (2015). The Reference Architecture Model Industry 4.0 (RAMI 4.0). *Zvei*, (July). Retrieved from <https://www.zvei.org/presse-medien/publikationen/das-referenzarchitekturmodell-industrie-40-rami-40/>
- Alphonsus, E. R., & Abdullah, M. O. (2016). A review on the applications of programmable logic controllers (PLCs). *Renewable and Sustainable Energy Reviews*, *60*, 1185–1205. <https://doi.org/10.1016/j.rser.2016.01.025>
- Blanc-Serrier, S., Ducq, Y., & Vallespir, B. (2018). Organisational interoperability characterisation and evaluation using enterprise modelling and graph theory. *Computers in Industry*, *101*(April), 67–80. <https://doi.org/10.1016/j.compind.2018.04.012>
- Carvalho, M. S. V. R. (2018). *Modelos de Referência e Interoperabilidade no Âmbito da Indústria 4.0: caso de demonstração PRODUTECH SIF na indústria transformadora*.
- Connolly, T., & Begg, C. (2005). *Database Systems: a practical approach to Design, Implementation and Management* (4th ed.). Pearson Education Limited.
- Cristian, M. (2017). The importance of databases in economy - some general coordinates. *Revista Economica*, *6*, 5–9.
- Federal Ministry for Economic Affairs and Energy. (2016). *Digitization of Industrie - Plattform Industrie 4.0*.
- Federal Ministry for Economic Affairs and Energy. (2018). *Alignment Report for Reference Architectural Model for Industrie 4.0 / Intelligent Manufacturing System Architecture*.
- Feltrin, L. (2015). KNIME an Open Source Solution for Predictive Analytics in the Geosciences [Software and Data Sets]. *IEEE Geoscience and Remote Sensing Magazine*, *3*(4). <https://doi.org/10.1109/MGRS.2015.2496160>
- Fernandes, F. S. (2018). safebag: airbags made in ponte de lima equipam automoveis. Retrieved from <https://www.jornaldenegocios.pt/negocios-iniciativas/premios-exportacao---internacionalizacao/detalhe/safebag-airbags-made-in-ponte-de-lima-equipam-automoveis>



- Fiesp. (2017). A corrida tecnológica rumo à Indústria 4.0 : quem está na pole position?
- Frank A G, Dalenogare L S, & Ayala N F. (2019). Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, 210, 15–20. <https://doi.org/10.1016/j.ijpe.2019.01.004>
- Gates, D., & Bremicker, M. (2017). Beyond the hype: separating ambition from reality in i4.0. Retrieved from <https://assets.kpmg/content/dam/kpmg/xx/pdf/2017/05/beyond-the-hype.pdf>
- Gotze, J. (2016). Reference Architectures for Industry 4.0. Retrieved from <https://coe.qualiware.com/reference-architectures-for-industry-4-0/>
- Grable, J. E., & Lyons, A. C. (2018). An Introduction to Big Data. *Economics and Investment Management*, 72(5), 17–20.
- Huang, S., Mohanty, S., Ashfahani, A., & Pratama, M. (2018). The Study on Indoor Localization for Manufacturing Execution System. *2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV 2018*, 1863–1867. <https://doi.org/10.1109/ICARCV.2018.8581192>
- i-SCOOP. (2019). Retrieved from <https://www.i-scoop.eu/industry-4-0/>
- Kaganski, S., Majak, J., Karjust, K., & Toompalu, S. (2017). Implementation of Key Performance Indicators Selection Model as Part of the Enterprise Analysis Model. *Procedia CIRP*, 63, 283–288. <https://doi.org/10.1016/j.procir.2017.03.143>
- Kagermann, H., Wahlster, W., & Helbig, J. (2013). Recommendations for implementing the strategic initiative industrie 4.0.
- Lin, S.-W., Miller, B., Durand, J., Bleakley, G., Chigani, A., Martin, R., & Crawford, M. (2019). *The Industrial Internet of Things Volume G1 : Reference Architecture*.
- Lin, S.-W., Murphy, B., Clauer, E., Loewen, U., Neubert, R., Bachmann, G., & Hankel, M. (2018). Architecture Alignment and Interoperability An Industrial Internet Consortium and Plattform Industrie 4.0 Joint Whitepaper, 19. Retrieved from http://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf
- Lydon, B. (n.d.). RAMI 4.0 Reference Architectural Model for Industrie 4.0: three-dimensional map showing how to approach Industry 4.0 in a structured manner. Retrieved from <https://www.isa.org/intech/20190405/>
- Macedo, M. A. (2017). Automação e integração de equipamentos na indústria cerâmica.
- Mckinsey. (2016). McKinsey Digital 2016. Retrieved from



- <https://www.mckinsey.com/~media/mckinsey/business-functions/mckinsey-digital/our-insights/getting-the-most-out-of-industry-4-0/mckinsey-industry-40-2016.ashx>
- Mulrow, C. D. (2014). Systematic Reviews Rationale for systematic reviews, (May), 1–4. <https://doi.org/10.1136/bmj.309.6954.597>
- power bi @ docs.microsoft.com. (2019). Retrieved from <https://docs.microsoft.com/en-us/learn/modules/explore-power-bi-service/1-why-power-bi>
- RAMI4.0 - a reference framework for digitalisation. (2018).
- Rodič, B. (2017). Industry 4.0 and the New Simulation Modelling Paradigm. *Organizacija*, 50(3), 193–207. <https://doi.org/10.1515/orga-2017-0017>
- Russom, P. (2011). Big Data Analytics Fourth Quarter 2011.
- Sabo, F. (2015). Industry 4.0 - A Comparison of the Status in Europe and the USA, (January).
- Slack, N., Chamber, S., & Johnston, R. (2009). *Administração da Produção*. (E. A. S.A., Ed.) (3. ed).
- Ślusarczyk, B. (2018). Industry 4.0 - are we ready? <https://doi.org/10.17512/pjms.2018.17.1.19>
- Smit, J., Kreutzer, S., Moeller, C., & Carlberg, M. (2016). Industry 4.0.
- Sony, M. (2018). Industry 4.0 and lean management: a proposed integration model and research propositions. *Production and Manufacturing Research*, 6(1), 416–432. <https://doi.org/10.1080/21693277.2018.1540949>
- Stephen, E. (2018). Why Manufacturing Digitalization Matters and How Countries Are Supporting It, (April), 1–66.
- Tannock, J. D. T. (2016). Entity Framework. *Organizational Behavior*, 123.
- Vermesan, O., & Friess, P. (2014). *Chapter 1 Introduction. Internet of Things – From Research and Innovation to Market Deployment*. River Publishers.
- Vik, P., Dias, L., Pereira, G., & Oliveira, J. (2012). Using Simio for the Specification of an Integrated Automated Weighing, 1534–1546.
- visual-studio-ide @ docs.microsoft.com. (2019). Retrieved from <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- WEF. (2015). *Industrial Internet of Things : Unleashing the Potential of Connected Products and Services*.
- Wortmann, A., Combemale, B., & Barais, O. (2017). A Systematic Mapping Study on Modeling



for Industry 4.0. <https://doi.org/10.1109/MODELS.2017.14>

Xu, L. Da, Xu, E. L., & Li, L. (2018). Industry 4.0: state of the art and future trends. *International Journal of Production Research*, 56(8), 2941–2962.
<https://doi.org/10.1080/00207543.2018.1444806>

Zimányi, E., Jallow, B., Kashef, S. (2018). Object Relational Mapping and Entity Framework, 1–16.

**ANNEX I – DATA DICTIONARY: ENTITIES DESCRIPTION**

Entity Name	Description	Occurrence
Input	How the issue was found (e.g.: audits).	Each Input can determine many issues.
Department	The department that identified the issue.	Each department can find many issues.
Priority	Priority on solving the issue.	One priority (1, 2 or 3) can be in many different issues.
Localization	Where the issue happened. Its scope goes beyond the shop floor.	Each localization can have many issues.
Issue	Problem found.	Lines and localization can have multiply issues.

ANNEX II – DATA DICTIONARY: RELATIONSHIP DESCRIPTION

Entity name	Multiplicity	Relationship	Multiplicity	Entity name
Input	1..1	Identifies	1..*	Issue
Department	1..1	Identifies	1..*	Issue
Issue	1..*	Has	1..1	Priority

ANNEX III – DATA DICTIONARY: ATTRIBUTES DESCRIPTION

Entity name	Attributes	Description	Data Type & Length	Allow nulls	Multi-valued
Input	idInput	Uniquely identifies the input.	2 ³¹ -1 variable characters.	No.	No.
	inputName	Designation of the input.	50 variable characters.	No.	No.
Department	idDepartment	Uniquely identifies the Department.	2 ³¹ -1 variable characters.	No.	No.
	departmentName	Designation of the Department.	50 variable characters.	No.	No.
Priority	idPriority	Uniquely identifies the Priority.	2 ³¹ -1 variable characters.	No.	No.
	priorityNumber	Identifies the priority number.	255 variable characters.	No.	No.



Localization	idLocalization	Uniquely identifies the Localization.	$2^{31}-1$ variable characters.	No.	No.
	localizationName	Designation of the Localization.	255 variable characters.	No.	No.
Issue	idIssue	Identifies the issue number.	$2^{31}-1$ variable characters.	No.	No.
	issueDescription	Designation of the issue.	$2^{31}-1$ variable characters.	No.	No.
	action	Designation of the action.	$2^{31}-1$ variable characters.	No.	No.
	startDate	The date for started issue.	8 variable characters.	No.	No.
	forecastDate	The expected end date for the issue.	8 variable characters.	No.	No.
	endDate	The actual date that the issue ended.	8 variable characters.	No.	No.
	localization	Designation of the Localization.	50 variable characters.	No.	No.
	input	Designation of the input.	50 variable characters.	No.	No.
	department	Designation of the Department.	50 variable characters.	No.	No.
	priority	Identifies the priority number.	$2^{31}-1$ variable characters.	No.	No.
	actionConcluded	Identifies if the action is concluded.	1 variable character.	Yes.	No.
	actionCancelled	Identifies if the action is cancelled.	1 variable character.	Yes.	No.
	actionContinued	Identifies if the action is continued.	1 variable character.	Yes.	No.
	remarks	Identifies user observations.	$2^{31}-1$ variable characters.	Yes.	No.



	timestamp	Identifies the date, hour, minutes and seconds that the issue was created or modified.	14 variable characters.	No.	No.
	responsibleName	Identifies the responsible name.	100 variable characters.	No.	No.
	responsibleNumber	Identifies the responsible number.	$2^{31}-1$ variable characters.	No.	No.
	situation	Identifies the situation the issue has.	255 variable characters.	No.	No.



ANNEX IV – REFERENTIAL INTEGRITY

Input (idInput, inputName)

Primary key idInput

Department (idDepartment, departmentName)

Primary key idDepartment

Priority (idPriority, priorityNumber)

Primary key idPriority

Localization (idLocalization, localizationName)

Primary key idLocalization

Issue (idIssue, issueDescription, action, startDate, forecastDate, endDate, localization, input, department, priority, actionConcluded, actionCancelled, actionContinued, responsibleName, responsibleNumber, situation, fkidInput, fkidDepartment, fkidPriority)

Primary key idIssue

Foreign key fkidInput **references** Input (idInput) on UPDATE NO ACTION on DELETE NO ACTION

Foreign key fkidDepartment **references** Department (idDepartment) on UPDATE CASCADE on DELETE NO ACTION

Foreign key fkidPriority **references** Priority (idPriority) on UPDATE SET NULL on DELETE CASCADE



ANNEX V – SQL STRUCTURE STATEMENT

```
create database A
```

```
use A
```

```
create table Input (  
idInput int not null identity (1,1) primary key,  
inputName varchar(50) not null,  
)
```

```
create table Department (  
idDepartment int not null identity (1,1) primary key,  
departmentName varchar(50) not null,  
)
```

```
create table Priority (  
idPriority int not null identity (1,1) primary key,  
priorityNumber int not null,  
)
```

```
create table Localization (  
idLocalization int not null identity (1,1) primary key,  
localizationName varchar(255) not null,  
)
```

```
create table Issue (  
idIssue int not null identity (1,1) primary key,  
issueDescription text not null,  
action text not null,  
startDate date not null,  
forecastDate date not null,  
endDate date not null,  
localization varchar(50) not null,  
input varchar(50) not null,  
department varchar(50) not null,  
priority int not null,  
actionConcluded bit,  
actionCancelled bit,  
actionContinued bit,  
remarks text,  
timestamp datetime not null,  
responsibleName varchar(100) not null,  
responsibleNumber int not null,  
situation varchar(50) not null,  
fkidInput int not null,  
fkidDepartment int not null,  
fkidPriority int,  
constraint fkidInput foreign key (fkidInput) references Input (idInput) on UPDATE NO  
ACTION on DELETE NO ACTION,  
constraint fkidDepartment foreign key (fkidDepartment) references Department  
(idDepartment) on UPDATE CASCADE on DELETE NO ACTION,  
constraint fkidPriority foreign key (fkidPriority) references Priority (idPriority) on  
UPDATE SET NULL on DELETE CASCADE,  
)
```

ANNEX VI – SQL VIEWS

```
use A
```



```
create view localization_concatenate_lines
as
select * from DBlines.dbo.lines
union all
select * from A.dbo.Localization
```

```
use A
create view TPessoal
as
select * from DBGIP.dbo.TPessoal
```



ANNEX VII – C# INPUT CLASS

```
1. namespace GAM_Projeto
2. {
3.     using System;
4.     using System.Collections.Generic;
5.
6.     public partial class input
7.     {
8.         [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:
DoNotCallOverridableMethodsInConstructors")]
9.         public input()
10.        {
11.            this.Issues = new HashSet<issue>();
12.        }
13.
14.        public int inputId { get; set; }
15.        public string inputName { get; set; }
16.
17.        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:
CollectionPropertiesShouldBeReadOnly")]
18.        public virtual ICollection<issue> Issues { get; set; }
19.    }
20. }
```

ANNEX VIII – C# DEPARTMENT CLASS

```
1. namespace GAM_Projeto
2. {
3.     using System;
4.     using System.Collections.Generic;
5.
6.     public partial class department
7.     {
8.         [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:
DoNotCallOverridableMethodsInConstructors")]
9.         public department()
10.        {
11.            this.Issues = new HashSet<issue>();
12.        }
13.
14.        public int departmentId { get; set; }
15.        public string departmentName { get; set; }
16.
17.        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:
CollectionPropertiesShouldBeReadOnly")]
18.        public virtual ICollection<issue> Issues { get; set; }
19.    }
20. }
```

ANNEX IX – C# PRIORITY CLASS

```
1. namespace GAM_Projeto
2. {
3.     using System;
4.     using System.Collections.Generic;
5.
6.     public partial class priority
7.     {
```




```
8.     [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:
DoNotCallOverridableMethodsInConstructors")]
9.     public priority()
10.    {
11.        this.Issues = new HashSet<issue>();
12.    }
13.
14.     public int priorityId { get; set; }
15.     public Nullable<int> priorityNumber { get; set; }
16.
17.     [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:
CollectionPropertiesShouldBeReadOnly")]
18.     public virtual ICollection<issue> Issues { get; set; }
19. }
20. }
```

ANNEX X – C# LOCALIZATION CLASS

```
1. namespace GAM_Projeto
2. {
3.     using System;
4.     using System.Collections.Generic;
5.
6.     public partial class localization
7.     {
8.         public int localizationId { get; set; }
9.         public string localizationName { get; set; }
10.    }
11. }
```

ANNEX XI – C# ISSUE CLASS

```
1. namespace GAM_Projeto
2. {
3.     using System;
4.     using System.Collections.Generic;
5.
6.     public partial class issue
7.     {
8.         public int issueId { get; set; }
9.         public string issueDescription { get; set; }
10.        public string action { get; set; }
11.        public Nullable<System.DateTime> startDate { get; set; }
12.        public Nullable<System.DateTime> forecastDate { get; set; }
13.        public Nullable<System.DateTime> endDate { get; set; }
14.        public string localization { get; set; }
15.        public string input { get; set; }
16.        public string department { get; set; }
17.        public Nullable<int> priority { get; set; }
18.        public Nullable<bool> actionConcluded { get; set; }
19.        public Nullable<bool> actionCancelled { get; set; }
20.        public Nullable<bool> actionContinued { get; set; }
21.        public Nullable<int> fkidInput { get; set; }
22.        public Nullable<int> fkidDepartment { get; set; }
23.        public Nullable<int> fkidPriority { get; set; }
24.        public string remarks { get; set; }
25.        public Nullable<System.DateTime> timestamp { get; set; }
26.        public string responsibleName { get; set; }
27.        public Nullable<int> responsibleNumber { get; set; }
28.        public string situation { get; set; }

```



```
29.
30.     public virtual department department_ { get; set; }
31.     public virtual input input_ { get; set; }
32.     public virtual priority priority_ { get; set; }
33. }
34. }
```

ANNEX XII – C# PROJECT CODE

```
1. using System;
2. using System.Collections.Generic;
3. using System.ComponentModel;
4. using System.Data;
5. using System.Drawing;
6. using System.Linq;
7. using System.Text;
8. using System.Threading.Tasks;
9. using System.Windows.Forms;
10. using System.Data.Entity;
11. using Excel = Microsoft.Office.Interop.Excel;
12. using Microsoft.Office.Interop;
13.
14.
15. namespace GAM_Projeto
16. {
17.
18.     public partial class Form1 : Form
19.     {
20.         public Form1()
21.         {
22.             InitializeComponent();
23.         }
24.         //Issue e tabelas consulta
25.         public GAMEntities context = new GAMEntities();
26.         GAMEntities db = new GAMEntities();
27.         //view
28.         GAMEntities1 context2 = new GAMEntities1();
29.         //TPessoal
30.         GestaoPessoalEntities context3 = new GestaoPessoalEntities();
31.
32.
33.         private void Form1_Load(object sender, EventArgs e)
34.         {
35.             // TODO: This line of code loads data into the 'gAMDataSet.Issue' table.
36.             You can move, or remove it, as needed.
37.             this.issueTableAdapter.Fill(this.gAMDataSet.Issue);
38.             txtboxcontadororiginal.Visible = false;
39.             txtboxcontadoreeditado.Visible = false;
40.             dateTimePickeratual.Text = string.Format("{0: yyyy-MM-dd}", DateTime.UtcNow);
41.             dateTimePickerInicio.Text = string.Format("{0: yyyy-MM-dd}", DateTime.UtcNow);
42.             dateTimePickerPrevista.Text = string.Format("{0: yyyy-MM-dd}", DateTime.UtcNow);
43.             dateTimePickerPrevista.Enabled = true;
44.             dateTimePickeratual.Enabled = false;
45.             dateTimePickeratual.Visible = false;
46.             dateTimePickerTimestamp.Visible = false;
47.             txtboxnomeresponsavelBD.Visible = false;
48.             txtboxnumeroresponsavelBD.Visible = false;
49.             txtboxnumeroresponsavelFORM.Visible = false;
50.             txtboxnumeroresponsavelFORM.Visible = false;
51.             txtemail.Visible = false;
```



```
51.         txtlocal.Visible = false;
52.         int i = advancedDataGridViewIssue.Rows.GetRowCount(DataGridViewElementSt
ates.Visible);
53.         lbcount.Text = Convert.ToString(i);
54.         //renomear as colunas da base de dados para inglês e não visualizar a co
luna
55.         advancedDataGridViewIssue.Columns["situaçãoDataGridViewTextBoxColumn"].H
eaderText = "Situation";
56.         advancedDataGridViewIssue.Columns["idIssueDataGridViewTextBoxColumn"].He
aderText = "Id";
57.         advancedDataGridViewIssue.Columns["idIssueDataGridViewTextBoxColumn"].Vi
sible = false;
58.         advancedDataGridViewIssue.Columns["descriçãoDataGridViewTextBoxColumn1"]
.HeaderText = "Issue";
59.         advancedDataGridViewIssue.Columns["açãoCorretivaDataGridViewTextBoxColum
n"].HeaderText = "Action";
60.         advancedDataGridViewIssue.Columns["dataInícioDataGridViewTextBoxColumn"]
.HeaderText = "Start";
61.         advancedDataGridViewIssue.Columns["dataPrevistaDataGridViewTextBoxColumn"]
.HeaderText = "Preview";
62.         advancedDataGridViewIssue.Columns["Data_Fecho"].HeaderText = "Forecast /
End";
63.         advancedDataGridViewIssue.Columns["localDataGridViewTextBoxColumn"].Head
erText = "Project or Location";
64.         advancedDataGridViewIssue.Columns["departamentoAfetadoDataGridViewTextBo
xColumn"].HeaderText = "Department";
65.         advancedDataGridViewIssue.Columns["prioridadeDataGridViewTextBoxColumn1"]
.HeaderText = "Priority";
66.         advancedDataGridViewIssue.Columns["nomeResponsavelDataGridViewTextBoxCol
umn1"].HeaderText = "Responsible";
67.         advancedDataGridViewIssue.Columns["observaçãoDataGridViewTextBoxColumn"]
.HeaderText = "Remarks";
68.         advancedDataGridViewIssue.Columns["numeroResponsavelDataGridViewTextBoxC
olumn1"].HeaderText = "Responsible_Number";
69.         advancedDataGridViewIssue.Columns["numeroResponsavelDataGridViewTextBoxC
olumn1"].Visible = false;
70.         advancedDataGridViewIssue.Columns["açãoConcluídaDataGridViewCheckBoxColu
mn"].HeaderText = "Concluded";
71.         advancedDataGridViewIssue.Columns["açãoConcluídaDataGridViewCheckBoxColu
mn"].Visible = false;
72.         advancedDataGridViewIssue.Columns["açãoCanceladaDataGridViewCheckBoxColu
mn"].HeaderText = "Cancelled";
73.         advancedDataGridViewIssue.Columns["açãoCanceladaDataGridViewCheckBoxColu
mn"].Visible = false;
74.         advancedDataGridViewIssue.Columns["açãoContínuaDataGridViewCheckBoxColum
n"].HeaderText = "Ongoing";
75.         advancedDataGridViewIssue.Columns["açãoContínuaDataGridViewCheckBoxColum
n"].Visible = false;
76.         dateTimePickerInicio.Enabled = false;
77.         //resize columns
78.         advancedDataGridViewIssue.Columns["descriçãoDataGridViewTextBoxColumn1"]
.Width = 250;
79.         advancedDataGridViewIssue.Columns["açãoCorretivaDataGridViewTextBoxColum
n"].Width = 180;
80.         advancedDataGridViewIssue.Columns["prioridadeDataGridViewTextBoxColumn1"]
.Width = 65;
81.         Initialize();
82.     }
83.     //add in a textbox the responsible number, to compare with previous register
ed in the function *private void txtboxnumeroresponsavelFORM_TextChanged(object send
er, EventArgs e)*
84.     private void dgvpessoal_CellClick(object sender, DataGridViewCellEventArgs e
)
85.     {
86.         int ind = e.RowIndex;
87.         DataGridViewRow selectedRows = dgvpessoal.Rows[ind];
```



```
88.         txtboxnumeroresponsavelFORM.Text = selectedRows.Cells[2].Value.ToString(
89.     );
90.     }
91.     private void Initialize()
92.     {
93.         //Iniciar combobox
94.         //prioridade
95.         comboBoxPrioridade.DataSource = context.priority.ToList();
96.         comboBoxPrioridade.DisplayMember = "priorityNumber";
97.         comboBoxPrioridade.ValueMember = "priorityId";
98.         comboBoxPrioridade.Text = "0";
99.         //input
100.        comboBoxInput.DataSource = context.input.ToList();
101.        comboBoxInput.DisplayMember = "inputName";
102.        comboBoxInput.ValueMember = "IinputId";
103.        comboBoxInput.Text = null;
104.        //departamento
105.        comboBoxDepartamento.DataSource = context.department.ToList();
106.        comboBoxDepartamento.DisplayMember = "departmentName";
107.        comboBoxDepartamento.ValueMember = "departmentId";
108.        comboBoxDepartamento.Text = null;
109.        //Funcao para atualizar as situacoes dos issues
110.        RefreshStatus();
111.
112.        //todos os registros da view
113.        var view = from concat_Lineas_Regiao in context2.concat_Lineas_Regiao
114.                  select new
115.                  {
116.                      concat_Lineas_Regiao.Isla,
117.                  };
118.        dataGridSQLView.DataSource = view.ToList();
119.        //preencher o datagrid no total de acordo com o tamanho da linha, projet
120.        o ou local
121.        dataGridSQLView.Columns["Isla"].AutoSizeMode = DataGridViewAutoSizeColumn
122.        nMode.Fill;
123.        //alterar nome da coluna
124.        dataGridSQLView.Columns["Isla"].HeaderText = "Project or Location";
125.
126.        //todos os registros de pessoal
127.        var pes = from pessoal in context3.TPessoals.OrderBy(s => s.Numero)
128.                  select new
129.                  {
130.                      pessoal.id,
131.                      pessoal.Numero,
132.                      pessoal.Nome,
133.                      pessoal.Tipo,
134.                  };
135.        dgvpessoal.DataSource = pes.Where(a => a.Tipo.Contains("I")).ToList();
136.        this.dgvpessoal.Columns["id"].Visible = false;
137.        this.dgvpessoal.Columns["Tipo"].Visible = false;
138.        //preencher o datagrid no total de acordo com o tamanho do nome
139.
140.        dgvpessoal.Columns["Nome"].AutoSizeMode = DataGridViewAutoSizeColumnMode
141.        .Fill;
142.        //alterar nome das colunas
143.        dgvpessoal.Columns["Nome"].HeaderText = "Name";
144.        dgvpessoal.Columns["Numero"].HeaderText = "Number";
145.        //Colocar campo como vazio assim que o formulario abre e se for chamada
146.        a função Initialize()
147.        textBoxIssue.Text = "";
148.        textBoxAcaoCorretiva.Text = "";
149.        txtboxnomeresponsavelBD.Text = "";
150.        txtboxnumeroresponsavelBD.Text = "";
151.    }
```



```
148.         //save register
149.         private void button1Save_Click_1(object sender, EventArgs e)
150.         {
151.             dateTimePickerTimestamp.Text = string.Format("{0: yyyy-MM-
152.                 ddTHH:mm:ss}", DateTime.UtcNow);
153.             // o local escolhido vai para o textbox local
154.             foreach (DataGridViewRow dr in dgvpessoal.Rows)
155.             {
156.                 foreach (DataGridViewRow er in dataGridSQLView.Rows)
157.                 {
158.                     if (er.Cells["Escolher"].Value != null && er.Cells["Escolher"].V
159.                         alue.ToString().ToLower() != "false")
160.                     {
161.                         txtlocal.Text = Convert.ToString(er.Cells["Isla"].Value);
162.                     }
163.                     //o nome e o numero do funcionario selecionado vai para textbox func
164.                     ionario
165.                     {
166.                         if (dr.Cells["Selecionar"].Value != null && dr.Cells["Selecionar
167.                             "].Value.ToString().ToLower() != "false")
168.                         {
169.                             txtboxnumerosresponsavelBD.Text = Convert.ToString(dr.Cells["
170.                                 Numero"].Value);
171.                             txtboxnomeresponsavelBD.Text = Convert.ToString(dr.Cells["No
172.                                 me"].Value);
173.                         }
174.                     }
175.                     //os campos do formulario nao podem ser deixados vazios
176.                     int issueid = Convert.ToInt32(labelIssueID.Text);
177.                     if (textBoxIssue.Text == "")
178.                     {
179.                         MessageBox.Show("Empty Issue", "Error", MessageBoxButtons.OK, Messag
180.                             eBoxIcon.Error);
181.                         return;
182.                     }
183.                     else if (textBoxAcaoCorretiva.Text == "")
184.                     {
185.                         MessageBox.Show("Empty Action", "Error", MessageBoxButtons.OK, Messa
186.                             geBoxIcon.Error);
187.                         return;
188.                     }
189.                     else if (comboBoxInput.Text == "")
190.                     {
191.                         MessageBox.Show("Empty Input", "Error", MessageBoxButtons.OK, Messag
192.                             eBoxIcon.Error);
193.                         return;
194.                     }
195.                     else if (comboBoxDepartamento.Text == "")
196.                     {
197.                         MessageBox.Show("Empty Department", "Error", MessageBoxButtons.OK, M
198.                             essageBoxIcon.Error);
199.                         return;
200.                     }
201.                     else if (txtboxnomeresponsavelBD.Text == "")
202.                     {
203.                         MessageBox.Show("Responsible was not selected", "Error", MessageBoxB
204.                             uttoms.OK, MessageBoxIcon.Error);
205.                         return;
206.                     }
207.                     else if (txtlocal.Text == "")
208.                     {
209.                         MessageBox.Show("Project or Location was not selected", "Error", Mes
210.                             sageBoxButtons.OK, MessageBoxIcon.Error);
211.                         return;
212.                     }
213.                 }
214.             }
215.         }
216.     }
217. }
```



```
202.         }
203.
204.         if (issueid == 0)
205.         {
206.             //create an issue object if id = 0
207.             issue problema = new issue
208.             {
209.                 //inserir os texto na tabela issue no SQL
210.                 issueDescription = textBoxIssue.Text,
211.                 action = textBoxAcaoCorretiva.Text,
212.                 actionConcluded = checkBoxAcaoConcluida.Checked,
213.                 actionCancelled = checkBoxAcaoCancelada.Checked,
214.                 actionContinued = ckacaocontinua.Checked,
215.                 startDate = dateTimePickerInicio.Value,
216.                 forecastDate = dateTimePickerPrevista.Value,
217.                 endDate = dateTimePickerEnd.Value,
218.                 input = Convert.ToString(comboBoxInput.Text),
219.                 department = Convert.ToString(comboBoxDepartamento.Text),
220.                 priority= Convert.ToInt32(comboBoxPrioridade.Text),
221.                 localization = Convert.ToString(txtlocal.Text),
222.                 timestamp = dateTimePickerTimestamp.Value,
223.                 remarks = txtobs.Text,
224.                 responsibleName = txtboxnomeresponsavelBD.Text,
225.                 responsibleNumber = Convert.ToInt32(txtboxnumeroresponsavelBD.Te
                xt),
226.                 //inserir as chaves estrangeiras na tabela issue no SQL
227.                 fkidPriority = Convert.ToInt32(comboBoxPrioridade.SelectedValue)
                ,
228.                 fkidInput = Convert.ToInt32(comboBoxInput.SelectedValue),
229.                 fkidDepartment = Convert.ToInt32(comboBoxDepartamento.SelectedVa
                lue),
230.             };
231.             context.issues.Add(problema);
232.             MessageBox.Show(this, "Data saved successfully", "Saved", MessageBoxButtons.OK, MessageBoxIcon.Information);
233.             context.SaveChanges();
234.             labelIssueID.Text = "0";
235.             advancedDataGridViewIssue.SelectionMode = DataGridViewSelectionMode.
                RowHeaderSelect;
236.             //Refresh datagrid
237.             issueTableAdapter.Fill(gAMDataSet.Issue);
238.             advancedDataGridViewIssue.Invalidate();
239.             advancedDataGridViewIssue.Refresh();
240.             RefreshStatus();
241.             textBoxAcaoCorretiva.Text = "";
242.             txtpesquisarfuncionario.Text = "";
243.             txtboxnomeresponsavelBD.Text = "";
244.             txtboxnumeroresponsavelBD.Text = "";
245.             checkBoxAcaoConcluida.Checked = false;
246.             checkBoxAcaoCancelada.Checked = false;
247.             ckacaocontinua.Checked = false;
248.             int i = advancedDataGridViewIssue.Rows.GetRowCount(DataGridViewElemen
                tStates.Visible);
249.             lblcount.Text = Convert.ToString(i);
250.         }
251.         //data modification
252.         else
253.         {
254.             var issueToUpdate = context.issues.SingleOrDefault(issue => issue.is
                sueId == issueid);
255.             if (issueToUpdate != null)
256.             {
257.                 //modificar os atributos da tabela issue
258.                 issueToUpdate.issueDescription = textBoxIssue.Text;
259.                 issueToUpdate.action = textBoxAcaoCorretiva.Text;
260.                 issueToUpdate.actionConcluded = checkBoxAcaoConcluida.Checked;
```



```
261.         issueToUpdate.actionCancelled = checkBoxAcaoCancelada.Checked;
262.         issueToUpdate.actionContinued = ckacaocontinua.Checked;
263.         issueToUpdate.startDate = dateTimePickerInicio.Value;
264.         issueToUpdate.forecastDate = dateTimePickerPrevista.Value;
265.         issueToUpdate.endDate = dateTimePickerEnd.Value;
266.         issueToUpdate.input = comboBoxInput.Text;
267.         issueToUpdate.department = comboBoxDepartamento.Text;
268.         issueToUpdate.localization = txtlocal.Text;
269.         issueToUpdate.priority = Convert.ToInt32(comboBoxPrioridade.Text
    );
270.         issueToUpdate.timestamp = dateTimePickerTimestamp.Value;
271.         issueToUpdate.remarks = txtobs.Text;
272.         issueToUpdate.responsibleName = txtboxnomeresponsavelBD.Text;
273.         issueToUpdate.responsibleNumber = Convert.ToInt32(txtboxnumeroresponsavelBD.Text);
274.         //modificar as chaves estrangeiras
275.         issueToUpdate.fkidPriority = Convert.ToInt32(comboBoxPrioridade.Selected
    Value);
276.         issueToUpdate.fkidInput = Convert.ToInt32(comboBoxInput.Selected
    Value);
277.         issueToUpdate.fkidDepartment = Convert.ToInt32(comboBoxDepartame
    nto.Selected
    Value);
278.         MessageBox.Show("Changes made", "Successfully", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
279.         context.SaveChanges();
280.         labelIssueID.Text = "0";
281.         advancedDataGridViewIssue.SelectionMode = DataGridViewSelectionMode.
    RowHeaderSelect;
282.         //Refresh datagrid
283.         issueTableAdapter.Fill(gAMDataSet.Issue);
284.         advancedDataGridViewIssue.Invalidate();
285.         advancedDataGridViewIssue.Refresh();
286.         checkBoxAcaoConcluida.Checked = false;
287.         checkBoxAcaoCancelada.Checked = false;
288.         ckacaocontinua.Checked = false;
289.         txtpesquisarlinha.Text = "";
290.         Initialize();
291.     }
292. }
293. }
294. //data selection
295. private void advancedDataGridViewIssue_SelectionChanged_1(object sender, Eve
    ntArgs e)
296. {
297.     DataGridViewRow selectedRow = null;
298.     if (advancedDataGridViewIssue.SelectedRows.Count > 0)
299.     {
300.         selectedRow = advancedDataGridViewIssue.SelectedRows[0];
301.     }
302.
303.     if (selectedRow == null)
304.         return;
305.     dateTimePickerPrevista.Enabled = false;
306.     labelIssueID.Text = selectedRow.Cells["idIssueDataGridViewTextBoxColumn"
    ].Value.ToString();
307.     textBoxIssue.Text = selectedRow.Cells["descricaoDataGridViewTextColumn
    n1"].Value.ToString();
308.     textBoxAcaoCorretiva.Text = selectedRow.Cells["açãoCorretivaDataGrid
    ViewTextBoxColumn"].Value.ToString();
309.     checkBoxAcaoConcluida.Checked = Convert.ToBoolean(selectedRow.Cells["açã
    oConcluidaDataGrid
    ViewCheckBoxColumn"].Value.ToString());
310.     checkBoxAcaoCancelada.Checked = Convert.ToBoolean(selectedRow.Cells["açã
    oCanceladaDataGrid
    ViewCheckBoxColumn"].Value.ToString());
311.     ckacaocontinua.Checked = Convert.ToBoolean(selectedRow.Cells["açãoContín
    uaDataGrid
    ViewCheckBoxColumn"].Value.ToString());
```




```
312.         txtobs.Text = selectedRow.Cells["observaçãoDataGridViewTextBoxColumn"].V
         alue.ToString();
313.         dateTimePickerInicio.Text = selectedRow.Cells["dataInícioDataGridViewTex
         tBoxColumn"].Value.ToString();
314.         dateTimePickerPrevista.Text = selectedRow.Cells["dataPrevistaDataGridVie
         wTextBoxColumn"].Value.ToString();
315.         dateTimePickerEnd.Text = selectedRow.Cells["Data_Fecho"].Value.ToString(
         );
316.         txtboxnomeresponsavelBD.Text = selectedRow.Cells["nomeResponsavelDataGri
         dViewTextBoxColumn1"].Value.ToString();
317.         txtboxnumeroresponsavelBD.Text = selectedRow.Cells["numeroResponsavelDat
         aGridViewTextBoxColumn1"].Value.ToString();
318.         comboBoxInput.Text = selectedRow.Cells["inputDataGridViewTextBoxColumn1"
         ].Value.ToString();
319.         txtlocal.Text = selectedRow.Cells["localDataGridViewTextBoxColumn"].Valu
         e.ToString();
320.         comboBoxDepartamento.Text = selectedRow.Cells["departamentoAfetadoDataGr
         idViewTextBoxColumn"].Value.ToString();
321.         comboBoxPrioridade.Text = selectedRow.Cells["prioridadeDataGridViewTextB
         oxColumn1"].Value.ToString();
322.     }
323.
324.
325.     //atualizar as situacoes de cada issue
326.     private void RefreshStatus()
327.     {
328.         for (int x = 0; x < advancedDataGridViewIssue.Rows.Count; x++)
329.
330.             if (advancedDataGridViewIssue.Rows[x].Cells["idIssueDataGridViewText
         BoxColumn"].Value != null && !string.IsNullOrEmpty(Convert.ToString(this.advancedDat
         aGridViewIssue.Rows[x].Cells["açãoCanceladaDataGridViewCheckBoxColumn"].Value)) && !
         string.IsNullOrEmpty(Convert.ToString(this.advancedDataGridViewIssue.Rows[x].Cells["
         açãoConcluídaDataGridViewCheckBoxColumn"].Value)))
331.             {
332.                 if (this.advancedDataGridViewIssue.Rows[x].Cells["idIssueDataGri
         dViewTextBoxColumn"].Value == null)
333.                 {
334.                     this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDataGr
         idViewTextBoxColumn"].Value = null;
335.                 }
336.                 {
337.                     if (Convert.ToBoolean(this.advancedDataGridViewIssue.Rows[x]
         .Cells["açãoCanceladaDataGridViewCheckBoxColumn"].Value) == true)
338.                     {
339.                         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
         taGridViewTextBoxColumn"].Style.BackColor = Color.Gray;
340.                         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
         taGridViewTextBoxColumn"].Value = "Cancelled";
341.                         //insere a situacao do issue na tabela "issue" da base d
         e dados
342.                         int ID = Convert.ToInt32(this.advancedDataGridViewIssue.
         Rows[x].Cells["idIssueDataGridViewTextBoxColumn"].Value);
343.                         var issue = context.issues.Where(s => s.issueId == ID).F
         irstOrDefault();
344.                         issue.situation = "Cancelled";
345.                         context.SaveChanges();
346.                     }
347.                     else if (Convert.ToBoolean(this.advancedDataGridViewIssue.Ro
         ws[x].Cells["açãoConcluídaDataGridViewCheckBoxColumn"].Value) == true)
348.                     {
349.                         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
         taGridViewTextBoxColumn"].Style.BackColor = Color.Green;
350.                         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
         taGridViewTextBoxColumn"].Value = "Concluded";
351.                         //insere a situacao do issue na tabela "issue" da base d
         e dados
```




```
352.         int ID = Convert.ToInt32(this.advancedDataGridViewIssue.
353.     Rows[x].Cells["idIssueDataGridViewTextBoxColumn"].Value);
354.         var issue = context.issues.Where(s => s.issueId == ID).F
355.     rstOrDefault();
356.         issue.situation = "Concluded";
357.         context.SaveChanges();
358.     }
359.     else if (Convert.ToBoolean(this.advancedDataGridViewIssue.Ro
360.     ws[x].Cells["açãoContínuaDataGridViewCheckBoxColumn"].Value) == true)
361.     {
362.         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
363.     taGridViewTextBoxColumn"].Style.BackColor = Color.Orange;
364.         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
365.     taGridViewTextBoxColumn"].Value = "Ongoing";
366.         //insere a situacao do issue na tabela "issue" da base d
367.     e dados
368.         int ID = Convert.ToInt32(this.advancedDataGridViewIssue.
369.     Rows[x].Cells["idIssueDataGridViewTextBoxColumn"].Value);
370.         var issue = context.issues.Where(s => s.issueId == ID).F
371.     rstOrDefault();
372.         issue.situation = "Ongoing";
373.         context.SaveChanges();
374.     }
375.     else if (Convert.ToDateTime(this.advancedDataGridViewIssue.R
376.     ows[x].Cells["Data_Fecho"].Value) >= Convert.ToDateTime(dateTimePickeratual.Text) &&
377.     Convert.ToBoolean(this.advancedDataGridViewIssue.Rows[x].Cells["açãoCanceladaDataGr
378.     idViewCheckBoxColumn"].Value) == false && Convert.ToBoolean(this.advancedDataGridVie
379.     wIssue.Rows[x].Cells["açãoConcluídaDataGridViewCheckBoxColumn"].Value) == false)
380.     {
381.         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
382.     taGridViewTextBoxColumn"].Style.BackColor = Color.Orange;
383.         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
384.     taGridViewTextBoxColumn"].Value = "Processing";
385.         //insere a situacao do issue na tabela "issue" da base d
386.     e dados
387.         int ID = Convert.ToInt32(this.advancedDataGridViewIssue.
388.     Rows[x].Cells["idIssueDataGridViewTextBoxColumn"].Value);
389.         var issue = context.issues.Where(s => s.issueId == ID).F
390.     rstOrDefault();
391.         issue.situation = "Processing";
392.         context.SaveChanges();
393.     }
394.     else if (Convert.ToDateTime(this.advancedDataGridViewIssue.R
395.     ows[x].Cells["Data_Fecho"].Value) < Convert.ToDateTime(dateTimePickeratual.Text) &&
396.     Convert.ToBoolean(this.advancedDataGridViewIssue.Rows[x].Cells["açãoCanceladaDataGri
397.     dViewCheckBoxColumn"].Value) == false && Convert.ToBoolean(this.advancedDataGridVie
398.     wIssue.Rows[x].Cells["açãoConcluídaDataGridViewCheckBoxColumn"].Value) == false)
399.     {
400.         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
401.     taGridViewTextBoxColumn"].Style.BackColor = Color.Red;
402.         this.advancedDataGridViewIssue.Rows[x].Cells["situaçãoDa
403.     taGridViewTextBoxColumn"].Value = "Delay";
404.         //insere a situacao do issue na tabela "issue" da base d
405.     e dados
406.         int ID = Convert.ToInt32(this.advancedDataGridViewIssue.
407.     Rows[x].Cells["idIssueDataGridViewTextBoxColumn"].Value);
408.         var issue = context.issues.Where(s => s.issueId == ID).F
409.     rstOrDefault();
410.         issue.situation = "Delay";
411.         context.SaveChanges();
412.     }
413.     }
414. }
415. //ordenar o datagrid
```



```
391.     private void advancedDataGridViewIssue_SortStringChanged(object sender, Even
    tArgs e)
392.     {
393.         this.issueBindingSource.Sort = this.advancedDataGridViewIssue.SortString
    ;
394.         RefreshStatus();
395.     }
396.     //filtrar o datagrid
397.     private void advancedDataGridViewIssue_FilterStringChanged_1(object sender,
    EventArgs e)
398.     {
399.         this.issueBindingSource.Filter = this.advancedDataGridViewIssue.FilterSt
    ring;
400.         int i = advancedDataGridViewIssue.Rows.GetRowCount(DataGridViewElementSt
    ates.Visible);
401.         lbcount.Text = Convert.ToString(i);
402.         RefreshStatus();
403.     }
404.     //clear all filters
405.     private void btnclearfilter_Click(object sender, EventArgs e)
406.     {
407.         advancedDataGridViewIssue.ClearFilter();
408.         this.issueBindingSource.Filter = this.advancedDataGridViewIssue.FilterSt
    ring;
409.         RefreshStatus();
410.         int i = advancedDataGridViewIssue.Rows.GetRowCount(DataGridViewElementSt
    ates.Visible);
411.         lbcount.Text = Convert.ToString(i);
412.     }
413.     //limpar o formulario
414.     private void buttonClear_Click(object sender, EventArgs e)
415.     {
416.         {
417.             //clear txt, checkbox
418.             labelIssueID.Text = "0";
419.             // labelnewissue.Text =
420.             textBoxIssue.Text = string.Empty;
421.             textBoxAcaoCorretiva.Text = string.Empty;
422.             checkBoxAcaoConcluida.Checked = false;
423.             checkBoxAcaoCancelada.Checked = false;
424.             ckacaocontinua.Checked = false;
425.             dateTimePickerPrevista.Text = string.Format("{0: yyyy-MM-
    dd}", DateTime.UtcNow);
426.             dateTimePickerEnd.Text = string.Format("{0: yyyy-MM-
    dd}", DateTime.UtcNow);
427.             dateTimePickerPrevista.Enabled = true;
428.             comboBoxDepartamento.Text = "";
429.             comboBoxInput.Text = "";
430.             comboBoxPrioridade.Text = "0";
431.             txtpesquisarlinha.Text = "";
432.             txtpesquisarfuncionario.Text = "";
433.             txtobs.Text = "";
434.             txtboxnomeresponsavelBD.Text = "";
435.             txtboxnumeroresponsavelBD.Text = "";
436.             txtemail.Text = "";
437.             txtboxnomeresponsavelFORM.Text = "";
438.             txtboxnumeroresponsavelFORM.Text = "";
439.             txtlocal.Text = "";
440.             txtboxcontadoreredito.Text = "";
441.             txtboxcontadororiginal.Text = "";
442.             dateTimePickerInicio.Text = string.Format("{0: yyyy-MM-
    dd}", DateTime.UtcNow);
443.             foreach (DataGridViewRow dr in dgvpessoal.Rows)
444.             {
445.
446.                 {
```



```
447.         if (dr.Cells["Selecionar"].Value != null && dr.Cells["Selec  
onar"].Value.ToString().ToLower() != "false")  
448.             {  
449.                 dr.Cells["Selecionar"].Value = false;  
450.             }  
451.         }  
452.     }  
453. }  
454. }  
455. //ação concluída = verdadeiro, situação = "concluída"  
456. private void checkBoxAcaoConcluída_CheckedChanged_1(object sender, EventArgs  
e)  
457. {  
458.     if (checkBoxAcaoConcluída.Checked == true)  
459.     {  
460.         dateTimePickerPrevista.Enabled = false;  
461.         dateTimePickerInicio.Enabled = false;  
462.         dateTimePickerEnd.Text = string.Format("{0: yyyy-MM-  
dd}", DateTime.UtcNow);  
463.         dateTimePickerEnd.Enabled = false;  
464.     }  
465.     else  
466.     {  
467.         dateTimePickerPrevista.Enabled = true;  
468.         dateTimePickerEnd.Enabled = true;  
469.     }  
470. }  
471. //ação cancelada = verdadeiro, situação = "cancelado"  
472. private void checkBoxAcaoCancelada_CheckedChanged(object sender, EventArgs e  
)  
473. {  
474.     if (checkBoxAcaoCancelada.Checked == true)  
475.     {  
476.         dateTimePickerInicio.Enabled = false;  
477.         dateTimePickerPrevista.Enabled = false;  
478.         dateTimePickerEnd.Text = string.Format("{0: yyyy-MM-  
dd}", DateTime.UtcNow);  
479.         dateTimePickerEnd.Enabled = false;  
480.     }  
481.     else  
482.     {  
483.         dateTimePickerPrevista.Enabled = true;  
484.         dateTimePickerEnd.Enabled = true;  
485.     }  
486. }  
487. //ação continua = verdadeiro, situação = "contínuo"  
488. private void ckacaocontinua_CheckedChanged(object sender, EventArgs e)  
489. {  
490.     if (ckacaocontinua.Checked == true)  
491.     {  
492.         dateTimePickerEnd.Value = dateTimePickerEnd.MaxDate;  
493.         dateTimePickerEnd.Enabled = false;  
494.     }  
495.     else  
496.     {  
497.         dateTimePickerPrevista.Enabled = true;  
498.         dateTimePickerEnd.Enabled = true;  
499.     }  
500. }  
501. //Pesquisar o funcionário que está na base de dados GestãoPessoal.TPessoal  
502. private void txtpesquisarfuncionario_TextChanged(object sender, EventArgs e)  
  
503. {  
504.     dgvpessoal.DataSource = context3.TPessoals.Where(c => c.Nome.Contains(tx  
tpesquisarfuncionario.Text) && c.Tipo.Contains("I")).ToList();  
505.     this.dgvpessoal.Columns["id"].Visible = false;
```



```
506.         this.dgvpessoal.Columns["Turno"].Visible = false;
507.         this.dgvpessoal.Columns["Seccao"].Visible = false;
508.         this.dgvpessoal.Columns["Funcao"].Visible = false;
509.         this.dgvpessoal.Columns["F_M"].Visible = false;
510.         this.dgvpessoal.Columns["Altabaixa"].Visible = false;
511.         this.dgvpessoal.Columns["Outro"].Visible = false;
512.     }
513.     //pesquisar funcionarios
514.     //pesquisar locais
515.     private void txtpesquisarlinha_TextChanged_1(object sender, EventArgs e)
516.     {
517.         //comboBoxLinha.DataSource = context.TC_Local.Where(c => c.Localizacao.C
ontains(txtpesquisarlinha.Text)).ToList();
518.         dataGridSQLView.DataSource = context2.concat_Lineas_Regiao.Where(c => c.
Isla.Contains(txtpesquisarlinha.Text)).ToList();
519.         this.dataGridSQLView.Columns["Id"].Visible = false;
520.     }
521.
522.     //selecionar a linha toda no datagrid ao escolher somente um campo da linha
523.     private void advancedDataGridViewIssue_CellClick(object sender, DataGridView
CellEventArgs e)
524.     {
525.
526.         advancedDataGridViewIssue.SelectionMode = DataGridViewSelectionMode.Full
RowSelect;
527.         txtboxcontadororiginal.Text = (textBoxIssue.Text).Length.ToString();
528.
529.     }
530.     //exportar para excel
531.     private void btnexportar_Click_1(object sender, EventArgs e)
532.     {
533.         exportarparaexcel(advancedDataGridViewIssue);
534.     }
535.     public void exportarparaexcel(DataGridView tabela)
536.     {
537.         Microsoft.Office.Interop.Excel.Application excel = new Microsoft.Office.
Interop.Excel.Application();
538.         excel.Application.Workbooks.Add(true);
539.         //gerar os nomes das colunas
540.         excel.Cells[1] = "Situation";
541.         excel.Cells[2] = "Input";
542.         excel.Cells[3] = "Departament";
543.         excel.Cells[4] = "Project or Location";
544.         excel.Cells[5] = "Issue";
545.         excel.Cells[6] = "Action";
546.         excel.Cells[7] = "Responsible";
547.         excel.Cells[8] = "Start";
548.         excel.Cells[9] = "Preview";
549.         excel.Cells[10] = "Forecast / End";
550.         excel.Cells[11] = "Priority";
551.         excel.Cells[12] = "Remarks";
552.         excel.Cells[13] = "Id";
553.         excel.Cells[14] = "Responsible_Number";
554.         excel.Cells[15] = "Concluded";
555.         excel.Cells[16] = "Cancelled";
556.         excel.Cells[17] = "Ongoing";
557.         //preencher as linhas
558.         int indicelinha = 0;
559.         foreach (DataGridViewRow row in tabela.Rows)
560.         {
561.             {
562.                 indicelinha++;
563.                 int indiceColuna = 0;
564.                 foreach (DataGridViewColumn col in tabela.Columns)
565.                 {
```



```
566.             indiceColuna++;
567.             excel.Cells[indicelinha + 1, indiceColuna] = row.Cells[col.N
ame].Value;
568.         }
569.     }
570. }
571. excel.Columns.AutoFit();
572. excel.Rows.AutoFit();
573. excel.ActiveCell.HorizontalAlignment = Excel.Constants.xlCenter;
574. excel.Visible = true;
575. }
576. private void exitToolStripMenuItem_Click(object sender, EventArgs e)
577. {
578.     Close();
579. }
580.
581.
582. private void textBoxIssue_TextChanged(object sender, EventArgs e)
583. {
584.     txtboxcontadoreditado.Text = (textBoxIssue.Text).Length.ToString();
585. }
586.
587. private void Form1_FormClosing(object sender, FormClosingEventArgs e)
588. {
589.     DialogResult result = MessageBox.Show("Do you really want to exit?", "Qu
estion", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
590.     if (result == DialogResult.Yes)
591.     {
592.         e.Cancel = false;
593.     }
594.     else
595.     {
596.         e.Cancel = true;
597.     }
598. }
599. private void dateTimePickerPrevista_Leave(object sender, EventArgs e)
600. {
601.     if (ckacaocontinua.Checked == false)
602.     {
603.         {
604.             dateTimePickerPrevista.Enabled = false;
605.             dateTimePickerEnd.Text = dateTimePickerPrevista.Text;
606.         }
607.     }
608. }
609.
610. private void dateTimePickerEnd_Validated(object sender, EventArgs e)
611. {
612.     {
613.         if (dateTimePickerEnd.Text == dateTimePickerPrevista.Text)
614.         {
615.             return;
616.         }
617.         else if (checkBoxAcaoConcluida.Checked == true)
618.         {
619.             return;
620.         }
621.         else if (checkBoxAcaoCancelada.Checked == true)
622.         {
623.             return;
624.         }
625.         else if (ckacaocontinua.Checked == true)
626.         {
627.             return;
628.         }

```



```
629.         else if (Convert.ToString(txtobs.Text).Contains(Convert.ToString(dat
eTimePickerEnd.Text))
630.         {
631.             return;
632.         }
633.         else
634.         {
635.             DialogResult result = MessageBox.Show("Write a remark about the
changed date", "End Date Changed", MessageBoxButtons.OKCancel, MessageBoxIcon.Exclam
ation);
636.             if (result == DialogResult.OK)
637.                 //escrever a data de fim na observação
638.                 {
639.                     txtobs.Select();
640.                     txtobs.SelectionStart = txtobs.TextLength;
641.                     txtobs.SelectedText += "\r\n" + Convert.ToString(dateTimePic
kerEnd.Text) + ": ";
642.                 }
643.             }
644.         }
645.     }
646. }
647. }
```



ANNEX XIII – GRAPHICAL USER INTERFACE

Concluded Cancelled Ongoing

Issue ID: 0 Start: 25 de setembro de 2019 Preview: 25 de setembro de 2019 Forecast / End: 25 de setembro de 2019

Issue:

Select	Project or Location
<input type="checkbox"/>	
<input type="checkbox"/>	
<input type="checkbox"/>	

Select	Number	Name
<input type="checkbox"/>		
<input type="checkbox"/>		

Remarks:

Action: Input: Department: Priority: 0

Row Count: 471

Situation	Input	Department	Project or Location	Issue	Action	Responsible	Start	Preview	Forecast / End	Priority	Remarks
Concluded											
Concluded											
Concluded											
Cancelled											
Concluded											
Concluded											
Processing											
Concluded											
Concluded											
Cancelled											