UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Business Rules Integration on a Public Illumination Management System

André Leitão Ginja

Mestrado em Engenharia Informática
Especialização em Sistemas de Informação

Dissertação orientada por:
Prof. Doutor Paulo Jorge Cunha Vaz Dias Urbano

2021

Aknowledgements

Firstly, I would like to thank my parents José and Maria, for making this possible.

A special thank you to my brother-like cousin Daniel, whose lessons have guided me through life.

A huge thank you to my girlfriend Daniela for having the patience to deal with my problems while also providing me with priceless motivation and strength, which was very important for completing this work.

Thank you, Paulo Rugeiro and Miguel Chaves, for the countless hours of support and open discussion and the essential friendship in the last couple of years for me to grow as a person and a professional.

A very special thank you to Professor Paulo Urbano for the priceless help as a counselor of this work and for taking the needed attention for every step of the process, helping me create better overall work with his passion for helping others succeed.

I would also like to thank all the personnel I contacted in GMV Portugal. Thank you, Cláudio Antunes, for guiding me through making sure I had a fantastic experience in the company regarding this work. Thank you, João Sequeira, for always having confidence in my work from the time I have done a summer internship at the company, which allowed me to return to produce this work, and Alejandro for always conducting the recruiting logistical process for me to be within the company.

Thank you, João Caseirito and Diogo Avelãs, for also being part of my university experience and becoming such good friends.

On a general note, I want to thank everyone that, in one way or another, contributed to my road leading to this moment, at any moment of my education process, especially all the staff and teachers from Faculdade de Ciências da Universidade de Lisboa who certainly had a major role in my development.

Lastly, I would like to leave a short reference to the current pandemic context, as I hope it becomes something I can look to when I read this document to remembering the past. Producing this document was a mix of emotions, with health always the number one concern and going through all the questioning process to judge our work. Also, with the pandemic, I learned a lot more about this computer science field, which allowed me to develop my capabilities and feel confident about my knowledge.

Resumo

Num mundo em constante mudança, existem cada vez mais avanços tecnológicos e organizações a executarem transformações digitais, com bruscas mudanças de contexto, de modo a conseguirem acompanhar estes avanços. Estas transformações servem o propósito de tentar aproximar pessoas e processos, tentando adaptar os produtos de negócio às necessidades e desejos dos clientes.

No entanto, nem todas as empresas conseguem implementar os processos necessários de modo a conseguir concretizar esta transformação digital. Esta realidade, transversal à maioria dos setores empresariais, faz-nos deparar com organizações que se perdem no tempo em termos de políticas tecnológicas.

Num contexto geral, estas dificuldades derivam de ser necessária uma coesão organizacional muito proeminente de modo a conseguir reagir ao ritmo elevado da **modernização tecnológica** a um nível global. Não acompanhar este ritmo a um nível aceitável implica um atraso no desenvolvimento, que se vai prolongando, gerando cada vez mais um maior défice de avanço.

As **Regras de Negócio** (também denominadas de **Business Rules**) procuram promover um ambiente facilitador às organizações menos adaptadas ao contexto atual, fornecendo-lhes mecanismos de negócio computacional que permitam uma integração mais acessível a certas funcionalidades tecnológicas.

Estas regras funcionam como um intermediário entre a programação dita tradicional e uma abordagem mais simplista, que permite que exista uma abstração mais compreensível de certas áreas do negócio que seriam outrora inacessíveis por parte de qualquer cliente.

Estas Regras de Negócio enquadram-se como tema deste trabalho, que tem como objetivo realizar uma proposta de implementação dessas mesmas regras num sistema de software. As Regras são, essencialmente, nada mais que condições lógicas impostas ao sistema, de modo que este reaja caso o que esteja programado aconteça.

Neste caso, as Regras de Negócio pretendem aumentar o fluxo de trabalho e o nível de eficiência do sistema promovendo a criação simplista de condições que possam posteriormente cooperar com código mais complexo, criando um sistema igualmente capaz, mas com componentes mais acessíveis ao conhecimento tecnológico das organizações com mais dificuldades nesse aspeto.

Este trabalho foi realizado sob o regime de estágio em parceria com a GMV Portugal, que é uma organização tecnológica responsável pela conceção de produtos de **software**. No caso do projeto referenciado para este trabalho, o mesmo destina-se à Divisão de Iluminação Pública da Câmara Municipal de Lisboa, que é a entidade pública responsável pela gestão e manutenção das unidades de iluminação pública da cidade de Lisboa. O estágio promoveu a integração a uma equipa da GMV Portugal que também exercia esforços para o desenvolvimento da aplicação em causa.

O SIIP, como é conhecido o produto em foco, é uma aplicação de software que se destina a realizar e gerir todos os procedimentos que derivam da atividade da entidade em questão. Estas atividades cobrem várias áreas de ação, sendo as mais importantes o tratamento e gestão de ocorrências derivados de problemas de iluminação públicas e gestão de material armazenado para fins de tratamento das ocorrências. Para contexto da aplicação, a ocorrência é o componente mais central dessa mesma aplicação, visto que engloba todas as fases que derivam dessa ocorrência, como análise e execução de tarefas.

No que toca à consideração deste cliente em termos de avanços tecnológicos, deparamo-nos com o caso clássico da falta de avanço nesta frente, tanto em termos de práticas, cultura, e até mesmo infraestruturas atuais. O produto de software SIIP está em processo de criação de modo a reverter toda esta situação, mas os problemas são mais profundos do que aparentam, pois colocam em causa valores e hábitos transversais a muitas sociedades, algo que é difícil reverter com um efeito quase imediato.

Durante este trabalho são explorados os fatores culturais, estruturais e técnicos que levam este cliente a ter uma abordagem mais clássica ao uso das tecnologias. Esta análise acentua a necessidade de uma ação profunda e disruptiva dos processos atuais. As Regras de Negócio representam esta disrupção, almejando também a proximidade dos utilizadores da plataforma para com os fatores tecnológicos da mesma.

A decisão pela abordagem às Regras de Negócio levou a que se fizesse uma pesquisa intensiva para casos de uso dessas mesmas regras, sendo que foram encontradas abordagens semelhantes em setores como a banca, farmacêutica e também em alguns casos de decisão aliados a situações de direitos de autor.

Numa das outras referências pesquisadas, foi também alvo de análise um artigo que refere as diferenças culturais para o uso de tecnologia entre organizações públicas e privadas. Tratando-se o cliente de uma entidade pública, esta referência tornou-se extremamente importante para despoletar mais uma vertente deste trabalho.

Esta divergência entre **setor público** e privado levou a que se fizesse um estudo a ser respondido por cidadãos portugueses, que colocava perguntas específicas a trabalhadores do setor privado, público e por fim questões relativas à perspetiva do cidadão no que toca à comparação entre serviços privados e públicos e a respetiva perspetiva dos serviços públicos existentes em Portugal, incluindo também o atual contexto pandémico e como esse mesmo contexto afetou a integração tecnológica por parte das organizações.

Após toda a análise de contexto, procedeu-se à contribuição de desenvolvimento do software SIIP de modo a serem criados os componentes para que se possa realizar uma posterior integração com as Regras de Negócio. Considerando o contexto da aplicação SIIP mencionado acima, as funcionalidades a serem desenvolvidas são a associação e a desassociação de ocorrências, a serem posteriormente usadas como base para a integração das Regras de Negócio no sistema. Estas integrações foram executadas num contexto de um projeto que recorre a programação orientada a objetos e à linguagem C# como fonte dos ficheiros desenvolvidos.

Por fim, após todos os componentes de software estarem desenvolvidos, procedeu-se à implementação da proposta de implementação das Regras de Negócio. Considerando algumas adversidades e limitações temporais e de contexto mencionadas no documento, o plano ideal de implementar essas mesmas regras no sistema acabou por não ser algo passível de se realizar.

Com isso, apesar de não existir essa componente tão prática de implementação, acabaram por emergir outras possibilidades que podiam tirar partido do facto de a proposta de implementação ser feita de modo externo à aplicação. Esta possibilidade rege-se pelo facto de se permitir pensar o produto da implementação das Regras de Negócio de raiz, isto é, como não existem bases técnicas alicerçadas na aplicação SIIP, era possível utilizar princípios de usabilidade, design, tecnologia e funcionalidade para desenhar um produto de raiz perfeitamente adaptado ao contexto do que se procurava para o produto SIIP.

Assim, foi redesenhada a estrutura de um sistema convencional de Regras de Negócio e adaptado às interfaces e fontes de dados da aplicação, havendo uma integração completa entre ambos os componentes. Os processos de prototipagem, desenho da arquitetura de sistemas, desenho e simulação de implementação serviram para promover o uso das Regras de Negócio no contexto específico do cliente.

Esta implementação é benéfica para alguns casos de uso específicos da aplicação, como por exemplo na formulação de notificações, que podem permitir um melhor controlo do sistema e do produto no geral, havendo um menor compromisso de complexidade tecnológica para a interação direta com a plataforma de Regras de Negócio, sendo a mesma mais acessível a utilizadores menos técnicos e experientes, potenciando o uso de tecnologias por parte dessas mesmas pessoas e, aos poucos, promovendo uma cultura mais moderna e tecnológica, que pode educar os utilizadores e promover cada vez mais a modernização tecnológica de uma organização.

Palavras-chave: Regras de Negócio, Business Rules, modernização tecnológica, setor público, software.

Abstract

This work is based on the proposal of integrating **Business Rules** in a **software** product called SIIP, intended to serve the Public Illumination Division of Lisbon's city council, which is a **public organization**. This application helps manage all types of requests related to the city's public illumination, such as maintenance or damage, storage management of public illumination components, and alert system integration. The current methods the Public Illumination Division uses are considered inefficient and outdated, and this work tries to contribute to that.

The business rules integration will try to improve the features mentioned above by providing an automated alternative to some processes that are very costly with current outdated mechanisms. This implementation is a particular challenge in this project because of the current deprecated status of most public applications, as technological usage is not yet very well integrated within public divisions. This fact is also part of this work, creating a case study to compare the technological knowledge between private and public organizations. This study served as a knowledge base for getting a better context of the SIIP application.

This work explains all the phases that a project of this kind should consider, including research about related applications and why Business Rules are essential, context gathering of the environment, client and application, and the development process of developing the application and creating an integration proposal of the Business Rules within the application in order to promote better technological optimization within organizations.

To further sustain the thesis of why Business Rules is an essential subject to this work, a case **study** was created and conducted, inquiring voluntary people about how public and private organizations differ in technological advances, difficulties, and access provision.

At a final stage, the Business Rules **integration proposal** is then applied, which includes prototyping and development of the whole structure needed to integrate the Business Rules within the SIIP system, trying to promote its use to improve efficiency and better workflows.

Keywords: Business Rules, software, public organization, study, integration proposal

Contents

List of Figures

List of Tables

Listings

List of Acronyms

API - Application Programming Interface
MVC - Model-View-Controller
RBS - Rule-based systems
SIIP - Sistema de Informação da Iluminação Pública
SMTP - Simple Mail Transfer Protocol
SQL - Structured Query Language

Chapter 1

Introduction

In an ever-changing world, digital transformations are happening by the second, leading to significant business competition, with corporations trying to provide the best service to fulfil the ever-increasing client demands. The current context demands a lot of product flexibility, as business requirements and needs can change all the time. With that, systems have to be as flexible and dynamic as possible.

Some companies have more resistance to change than others, which may affect an organization's development. This factor is even more noticeable in technological transformation, with most companies stopping in terms of growth in these areas.

Business Rules is a concept that appears to answer this problem, as it promises flexible and adaptable development by separating business validation conditions from the application logic. Those business rules are compact and straightforward statements that represent portions of different business cases. Those statements are often represented by conditional arguments that define validation and are often more readable than regular code, improving transparency.

All business sectors of society can benefit from this change, as they are more prepared to create new services and products without needing to develop logistics from scratch. On financial examples, some validation tasks can be automated by emailing clients who will have to pay the service's monthly fee. Having an entity that manages these validations is essential to achieve a more agile solution that can promote generalization and reusability.

This work will feature an integration proposal of these rules in an applicational system called SIIP, a product designed for the Public Illumination Department of Lisbon's city council.

1.1 Application context

SIIP is a software product commissioned by Lisbon's state illumination department to improve the overall service and effectiveness. As stated by the article [7], the mentioned division is responsible for some assignments, including these that relate to the project:

- Plan, study, project, and execute new or renewed public illumination installations and ornamental installations in monuments, fountains, and lakes;
- Assure the above-mentioned public illumination installations' maintenance;
- Cooperate with the public energy supplier to assure the total integration of their responsibilities within the whole system;
- Keep public illumination components' georeferenced information.

Besides that, a community product is already related to the division, called "A Minha Rua." that allows citizens to create notifications related to damaged or malfunctioning public illumination, which eases maintenance recognition and workflow. This mentioned application is supposed to converge with the SIIP application, creating requests that the team can handle.

The SIIP project is mainly based on these needs. The main goal is to cover most of the aspects listed above, granting a technological approach to improve the division's effectiveness and overall workflow.

The main planned features are a request interface to process illumination-related problems and an notifications panel to improve the control of all the transactions across the product. With that, a stock manager implementation of illumination products will also be part of the project.

1.1.1 Application technologies and architecture

Identifying the application's elements and architectural structure is vital for integrating the use cases better with the system. The technologies used will be aborded in a different section, as they are a vital talking point to some business decisions. As stated in the context introduction, the SIIP product intends to cover most public illumination division operations.

With that in mind, the application's core business is based on event requests and notifications, which can then be integrated with Business Rules. That implementation can also have an essential role in covering some parameters of the stock management feature.

Storage Management

This is a business component of the application. All materials regarding public illumination are kept in storage units across the city, and the SIIP application is responsible for assuring the stock management of those components.

Requests

Another significant component of this application is a system that allows technicians and supervisors to control and create requests when, for instance, a street light is broken. As stated above, this application intends to merge this functionality with the A Minha Rua application, so community users can also actively report the physical system's malfunctions. This feature intends to cover all the processes, from the report's start to reparation, if needed, and conclusion.

Notifications

The features mentioned above (and other secondary features) are planned to be integrated within an alert system that prompts users about those features.

1.2 Motivation

This work is heavily inspired by how outdated most governmental systems run nowadays, where most departments usually fail to have an efficient service workflow. With this in mind, it is also part of this work to analyze how the department interacts with the current systems and how technological improvements can improve those practices.

1.2.1 Public administration's current context

With that in mind, some articles motivated this work, exposing some situations that fit the current context of the project's client. Digital transformation is much needed in public service organizations due to the critical facts explored in this subsection.

The publication [10] states that the digital transformation in Public Administration is suitable for a country and its citizens and goes as far as saying that implementing digital technology within government institutions is an opportunity that must be addressed as soon as possible.

The example demonstrated in this publication relates to an instance regarding Italian society. Being a developed country in Europe, it is a country that is naturally becoming more tech-friendly, with the community being more and more connected to the internet. Although this transformation is happening quickly, Public Administrations are falling short of aligning to their citizens' digital knowledge and its integration into their lives. According to the publication, the public administration organizations' scenario appears to be stationary compared to the biggest technology companies, lacking a fundamental transformation.

These facts reflect the delay in the Public Administration's transformation, connected to a low level of digital competence, while also affecting the quality of the services pro-

vided, causing a sense of distancing between the citizen and its Public services, due to an evident lack of alignment in terms of use cases.

This context leads to a sense of a Public Administration structure that is inefficient and obsolete procedures. To solve that, according to this publication, five challenges need to be faced, which will be listed below.

- **Raise the level of professionalism** - Much like the private sector, the public sector needs to integrate its workforce with professionals who can provide the skills needed to deal with digital transformation, needing technological, organizational, and managerial skills to deal with this transformation.

- **Change the idea of communication** - Another challenge to achieving this digital transformation would be to update the communication, making it more digital and more in line with the expectation of its increasingly connected citizens.

- **Safety first** - With this digital transformation, we also have to be aware of how a major security issue opens up if things are not properly planned and done, risking sensitive data and storage. Several recent hacker attacks in public bodies suggest that these organizations must enrol in digital transformation but may also need extra security concerns due to the quality of data they have in their hands.
  Security is granted by good software design and workers' practices, which are significant reasons in encouraging organizations to invest in digital education, preventing security attacks caused by human failure, which can directly affect the computer systems.

- **Networking** - A complete digital transformation also involves development in Networking. This Networking is based on creating a match between the demand for innovation that the citizens have and the available digital solutions.

- **Open up to new technologies** - One of the most critical challenges is to correctly integrate the latest technologies made available by digital innovation into administrative procedures. With this, organizations should be more willing to embrace the new technologies faster. They can improve bureaucratic processes, medical and health care, justice, and stimulating active citizenship.

The article [3] lists the systemic challenges for a digital transformation in the public sector. The author has worked for 20 years in technological and data-related subjects in the public sectors and has the needed knowledge to discuss this topic. The first presented challenge is the response to continuous change. This topic states that digital transformations are running rapidly, changing how we work, live, and learn. This transformation eventually has to go through public services to provide modern and adequate services to the community. This transformation also implies a change in speed and global competition due to the worldwide range offered by these developments.

Considering this context and combining it with the vertical accountability place in these organizations, the author believes that the slower evolution of public structures is a natural consequence of the accountability structure. This results in slower systems, efforts, and policies unless there is a special effort to counter that trend, and most organizations do not seem to have that extra effort and end up getting outdated.

In the author's opinion, this effect is also related to the public structure's financial strictness, forcing a system to take a narrower approach to its priorities, maximizing the existing budget, and minimizing risk.

To fight this, the author needs to commit to transforming the culture, practices, infrastructure, funding approaches, and policy landscape to continually adapt to the 21st century. Such efforts would include embracing and planning for change, taking a more agile or segmented approach to all work, and putting persistent and holistic measurement systems in place that can both measure impact and change. For the author, this would translate to building a proactive strategy that can adapt to change. For this, the author states that we need teams whose purpose is to look for patterns and bridge gaps across public sectors to create continuously evolving systemic responses to the current always-changing systems.

In a news article published by [22], Portugal's current Economy and Digital Transition Minister, Pedro Siza Vieira, mentions how significant the digital transformation in the Public Administration is. The Minister, in a quotation, says that the "Public Administration can serve better citizens and companies while increasing transparency and effectiveness of the available digital technologies.". Besides that, the Minister also states that "There is no time to lose.". He also states that "we have to look at the factors that intervene in the formation of our workers, organizations capabilities and the Public Administration transformation to accelerate this transformation.".

To sum everything that has been said above, there is a need to improve the Public Administration, especially in digital knowledge that requires a profound digital transformation. Public administration organizations will need to adapt to our current times, which are primarily technological driven to realign their objectives to comply with the current state of society. The public administration sector is stagnant in old and obsolete technologies, and urgent change is needed. This change needs to take origin from the technological standpoint and the human side, better digital education to improve the current practices. Besides that need, public sector organizations deal with sensitive data related to the citizens, asking for a particular security concern that needs to be put into practice in promoting a secure application design and preventing human error may compromise those systems.

These articles explored above set the baseline for what we should expect for our client, which belongs to this organization type. A case study was developed to explore and further confirm these allegations, and that study will be presented later in the document.

1.2.2 Client context

This subsection intends to briefly describe the client's first impression to confirm the context given in the subsection mentioned above representing public administration-related organizations. After some meetings with the client, it was possible to carve their operational profile, that is based on the following:

- The client is not tech-friendly on a generalized basis, which limits some technological advances that could be made in the application;
- The client is usually not very flexible regarding changes, which does not leave much room for macro technological diversification.
- The client's technological stack is not optimal because most data is usually stored in Microsoft Excel files, limiting theme choices in areas such as database optimizations or machine-learning-related solutions.

1.2.3 Context analysis

As a result of this analysis, it was then agreed that the right approach to this problem would be to create a solution that would positively impact the application while also not being truly invasive to the already agreed business decisions. Applying business rules models to the system seemed perfect, considering all the restrictions mentioned above.

These rules intend to create a better and clearer operational experience by simplifying the creation of business-logic conditions while improving strategy, agility, and transparency. The motivation behind this approach relies on seeing the department's technological practices and realizing that Business Rules could have an immense impact on SIIP's overall workflow.

1.3 Goals

This work's goals are relatively parallel to the project's execution. With that, both the client and the project must benefit from this work's development.

As for the system itself, implementing Business Rules intends to encourage these types of companies to simplify the development process. The goal is to ease the overall configuration operations and streamline the rule-writing process, giving that explicit instructions usually represent those business rules. This approach will also allow the developers to re-utilize business logic across the system, considering the rules' generalization property.

On the client's side, the Business Rules proposal will demonstrate the power to automate some of the application's controls and facilitate changing the needed rules, considering its transparency and readability.

On a personal level, the goal is to use all the topics learned during the four previous years to affect the client, the company, and the product in the best way possible.

The chosen theme described in the motivation topic is a new personal challenge. However, it will hopefully be treated with optimal needs, with the permanent acquisition of new methodologies will improve the theme's execution quality.

Ultimately, the main goal is to provide an implementation proposal of business rules that efficiently fit into the application, and this work will describe the developments in those areas.

Integration of the themes with the application

Considering the theme introduced at the beginning of this introduction, it is now clearer to grasp how they will be integrated within the SIIP application.

Using business rules, some processes of these features can be automated to simplify processes. In this case, most types of notifications can be generated through business rules. Some examples of this can be:

- In a storage unit, alert users when a certain component goes below a certain threshold.
- In a request, inform users if a certain number of days have passed after that request has been opened.

The mentioned examples can help us understand how business rules might impact our project, and we think significant developments and improvements can come from this implementation.

1.4 Change of plans - From Implementation to Proposal

An impeding factor was crucial for deciding the path to take with this work. In this case, this work was thought to take profits from the already existing implementation of the Notifications feature. With that, the Business Rules would be implemented on top of it.

Adding to this, there were also some problems regarding the project's organization. Two of the project's core developers left, with no direct replacement being made to the SIIP application. Adding to this, the client seemed to have abandoned the project. This factor is significant for the project's development because the client stopped communicating with the company. By the time this work started, there were almost weekly communications and monthly meetings with the client. Unfortunately, that stopped near January 2021, as no more communications were made with the client, which affected SIIP's development speed.

Combining these two factors, the company was forced to suspend its efforts to develop the SIIP application to prioritize other projects. There were, temporarily, insufficient human resources to push for a project that saw its clients disappear for unknown reasons. During two to three months, the only SIIP developments were exclusively made to complete this work's objectives, severely affecting this work's final goals and forcing a context change.

With this in mind, a solution was needed. That solution had to reflect the work's status. It was then decided not to apply the integration directly within the SIIP application but as a proposal. Making a Business Rules integration proposal seemed the best alternative as it would not affect the project's workflow and would still complement this work. This decision also allows for better and more stable Business Rules utilization because they can provide a better guideline for the whole scope, not just for specific use cases.

1.5 Contributions

The contributions of this work, and its mapping to the work plan seen in 1.6 are the following:

- Initial contributions on the project's workflow (Initial efforts);
- Finding an adequate work theme considering the client's context (Initial efforts and Context analysis);
- Developing a case study to promote discussion about technological improvements (Case study development);
- Development of the SIIP application features (Request Association development and Request Disassociation development);
- Development of the Business Rules implementation proposal (Business Rules implementation proposal).

1.6 Work plan

The planned work plan involves supporting the project's development in order to be able to apply this work's objectives.

Activity description	Sep/20	Oct/20	Nov/20	Dec/20	Jan/21	Feb/21	Mar/21	Apr/21	May/21	Jun/21	Jul/21	Aug/21	Sep/21
Initial efforts	X	X											
Preliminary report writing		X	X										
Context analysis			X	X									
Request Association Development				X	X	X		X	X	X	X		
Request Disassociation Development							X	X	X	X	X		
Case study development						X	X						
Conflict management										X	X		
Business Rules implementation proposal										X	X	X	X
Dissertation writing					X	X	X	X	X	X	X	X	X

Table 1.1: Work plan

1.7 Document structure

This document is organised as follows:

- Chapter 1 - Introduction
- Chapter 2 - Related Work
- Chapter 3 - System pre-development analysis
- Chapter 4 - SIIP Development
- Chapter 5 - Business Rules Integration Proposal
- Chapter 6 - Conclusion and Future Work

Chapter 2

Related work

This chapter intends to further detail the knowledge base behind Business Rules and the SIIP application. A proper and detailed overview of the application is crucial for a compelling study of the described Business Rules implementations highlighted in this work.

2.1 Application Knowledge Base

As a disclaimer, it is valuable to say that information related to this section will all be provided by the GMV company, as they are the stakeholder of all the business and technological actives associated with this work. This section will explore the processes and architecture needed to understand how the application is structured and how it works on a workflow and component basis.

2.1.1 SIIP Architecture

As mentioned above, this work directly depends on the SIIP application that is being built. This application intends to merge all Public Illumination management, whether it manages documents or materials related to the Public Illumination department's operations.

The SIIP application plans to create relations with some already existing systems that are as follows:

- Integration of external City Council's systems to centralize information.
- Data manipulation of City Council's georeferenced systems

The application follows a less standardized architecture, as some structural decisions are heavily affected by the client's limitations, habits, and requirements. Those requirements vary between automated registry gatherings and processes, developing a dynamic and adaptive application, editing automatic configurations, and making it future-proof and evolutive. More detailed information on requirements will be given in further analysis and development stages.

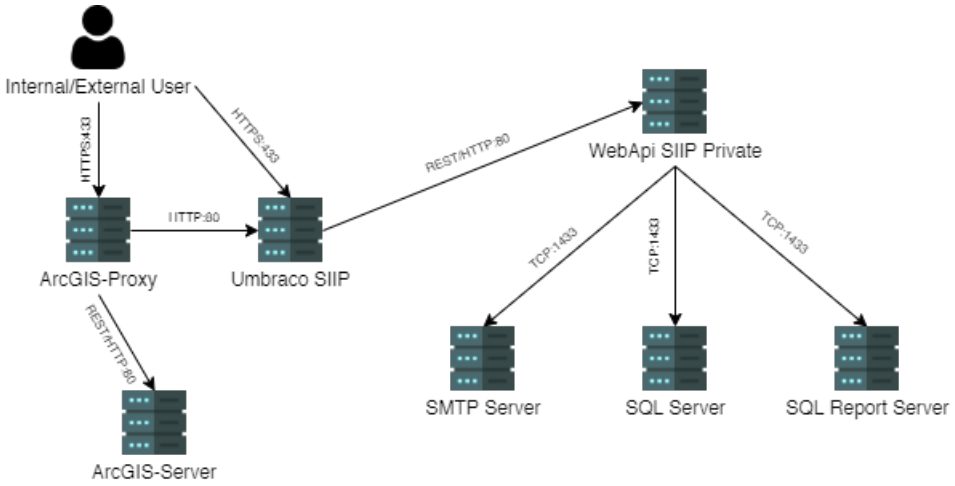

Figure 2.1: SIIP's architecture overview, as displayed in a project's internal document

Components like Umbraco and ArcGIS will probably be used in some way as a support tool for the Business Rules implementation because of how much data is passed through those structures, and so they will now be further detailed.

2.1.2 Umbraco

Umbraco is an open-source content manager system (also known as CMS) designed to serve Windows-based technologies related to .NET projects. This tool intends to ease technological solutions by separating some programming architecture structures from the programming code itself. For example, with the MVC pattern's adoption, Umbraco can manage all types of MVC components, such as models, views, or controllers, and link them together in a fully functional ecosystem.

In addition to this, it allows users to create small collections of values to emulate a database. Those collections are stored in a Microsoft SQL server with a much more optimized structure, given how data is managed in Umbraco. As seen in the image represented above, the SIIP core structure is built around its Umbraco implementation. Every main component created in the code must be designed in Umbraco to produce a mapped version to create a structure that allows getting information across the user and the system. Some more obvious examples of using this tool are represented, for example, in static dropdowns that do not need dynamic options, such as cities, default levels of urgency, and others. These dropdowns can be created in Umbraco and only be referenced in the coding part of the project, which eases making these types of structures.

Besides this, the Umbraco approach complies with one of the client's requests: to have the flexibility to change some values in the future. The client is then given access to Umbraco, allowing them to have permissions that allow modification to these more superficial value structures, such as changing the values of cities to be displayed in a dropdown.

This tool is undoubtedly essential for this work because as all data will pass through Umbraco, some of the Business Rules will control values coming from Umbraco.

2.1.3 Application components

As briefly mentioned in the introduction, the Business Rules theme implementation intends to focus on the application's main business features. These features rely on the Requests, Stock manager, and Alerts, and for better context, it is better to detail each component to introduce how Business Rules can affect them.

Requests

A request is the central origin of most workflows. These requests exist to report a problem that is the responsibility of the Public Illumination Department to solve. Internal and external entities can generate requests. Internal entities are Department-related members, and external entities relate with a current City Council community-oriented application called "A Minha Rua" that allows citizens to report issues related to the Department, mostly illumination occurrences.

Each request must be treated as a process. All operations that relate to that request will be visible within the application. Processing a request demands the need to deploy the required workflows to solve the issue. The mentioned workflows detailed in a later section of this work should also be attached to the request.

The requests will be classified based on their importance and urgency and will keep detailed information about dates and the current operation state. That request also has to be georeferenced by an individual related to the Department, so the request can be accurate enough to avoid process errors.

This component is the core business of this application because it directly impacts the other two features mentioned next, as they are deeply involved in the processes' phases of a request.

Stock management

To support the requests feature and modernize the Department's processes, the integration of a stock manager intends to control the ability to manage products related to the Department's needs, primarily illumination-related products. This feature intends to digitally represent the existing physical structure, including storage units, components, and illumination units. The details are digitally described and can be associated with requests to connect them to a process within a request. The items are then mapped into the system like a digital inventory to track all physical actives and make them available for usage across the platform.

Alerts

The alerts feature will be in the same ecosystem as the features mentioned above. This component will be responsible for gathering events triggered in the requests and stock management workflows. Those events are stock rupture, process approval, component reservations, request redirecting and status changes, responsibility delegations, and many others. As alerts cover a wide range of business logic, they are the optimal approach to business rules integration. The detailed level of integration between all parties will be explored in the analysis section of this document.

Component Integrations

To sum it up, all the described processes interact in some way with all others. SIIP is a partially closed ecosystem in which all its components were created to replicate the existing legacy systems at the Department. Requests are the main component, but the procedure is dependent on all the other elements to complete the entire workflow, whether it is getting stock materials or integrating those processes with alerts. The image below shows a simplistic representation of this integration that will be explored in further chapters of this work.

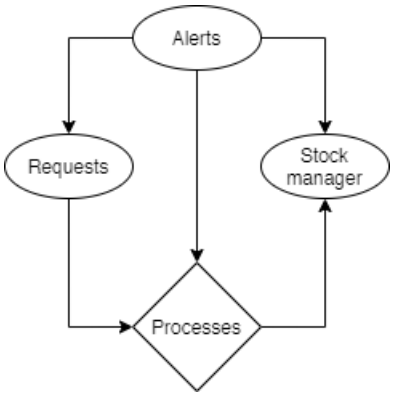

Figure 2.2: SIIP's main components ecosystem

2.1.4 Object-oriented programming

This subsection introduces the functional basis of the SIIP project, which will also serve as a complement for the following subsection.

The book [20], which has a specific section about Object-oriented programming, states that its state characterizes an object and what sort of capabilities it has. The author then proceeds to exemplify an object like a computer, with the computer being the object. As for the capabilities, those would be the memory, screen resolution, and others.

The book [29] defines an Object as an entity that is delimited from its environment while also introducing the concept "composite objects.", which are objects that are made

from other objects. Deeply analyzing objects, their genesis is based on attributes. The author considers it abstractions of a single characteristic possessed by all the entities treated as an object. In other words, object-oriented languages are worth what their name is - a type of programming language where the most critical measuring unit is the object, and all the logic should be object-centred and designed to take the objects as the first consideration.

After that, still considering objects, the author considers some essential concepts necessary to grasp regarding the SIIP project. One of those concepts is the existence of methods within an object. Those methods, also called functions, are used to specify the object's behaviours, with it being based on a set of rules that are defined inside that method, controlling the behaviour of that object. Methods are often used to change a specified object's properties or integrate them within other objects.

Another concept is represented by the word Identity, in which each object is unique, even with equal attributes, which eases out the process of computing each object.

With the existence of an object-centred approach, there was a need to connect objects to another object. With that, the Relations concept was born. The article's author sets an example of how relations should work between objects, considering an information system, such as replicating accounts in a bank. An object could represent an account, and interactions with those accounts can be considered relations between objects.

Lastly, speaking on objects, one other concept that sets the baseline for an object-oriented programmed object is encapsulation. The author states that its purpose is to hide certain types of information from everyone. With that, most attributes from each object are blocked from access and manipulation. Those attributes are only accessible through publicly accessible methods, which is very critical to protect the attributes.

Outside of the objects, a few other concepts are fundamental in defining object-oriented paradigms used in the SIIP project.

One of these is called inheritance, which, as the author said, is considered one of the essential object-oriented concepts. Inheritance is defined as the relation between classes where the definition and implementation of one class are based on another, meaning that objects can be constructed based on other objects. When an object is inherited from another, the superclass gives all its attributes, especially when objects share similarities and have differentiating characteristics.

The final concept regarding this topic is polymorphism, which relates to the meaning of its name, which means taking more than a single form. This use case can be applied, for example, when there is a need to create an object with certain attribute variations to improve the logical specifications of each object.

2.1.5 MVC Pattern

This subsection intends to contextualize better how the SIIP project is organized in regards to Software Development. The section aims to ease understanding the development sections of this work that concern the SIIP platform development.

The MVC pattern, also known as Model-View-Controller, is said in the document [9] to be a pattern or idiom used to create separation between a program's logic and design development.

The author introduces the MVC pattern by talking about how software needs interaction to fulfil its usability. The article mentions that most applications are conceived with a less modular approach, advertising that such a structure will ease the process.

The MVC approach is based on modules to create layers of interaction between the system and the user. The main goal is to abstract all the components that do not need to be known at a particular layer. For example, the client does not need to have direct knowledge of how a function is built. With that in mind, the MVC pattern intends to distance a particular layer from all the layers that do not directly integrate.

MVC Components

As said earlier, MVC stands for Model-View-Controller, which would represent a 3-component connection. The components will now be detailed.

Model

The article classifies the Model component as the pure essence of the logical program. When we consider object-oriented programming languages like the one used in the SIIP project, the model component directly correlates to the sets of classes that define the project's entities.

Combining this role with the information presented above, the Model does not need to receive information from other layers because he produces information.

View

The article describes the View as the Model's interface, which can also be Views (plural). These views will be the ones representing information generated by the Model's classes. They can also generate various types of views, depending on the use case. The views can be graphical user interfaces, command-line interfaces, or even application program interfaces, serving diverse roles.

Controller

The author of this article remarks on the controllers with a simple statement, saying that a Controller is an object that lets a user manipulate a view. Controllers are the layers that

have the most knowledge of the application context and outside information. A perfect example of this can be obtained by noticing that a view has no access to its Controller, but a controller can access all its views.

The ”Model Confusion”

The descriptions mentioned earlier can sometimes be inaccurate regarding the Model component, which applies to the SIIP project. The author questions the current MVC name, stating that it should be called M_dM_aVC. This means that our Model should be separated into two components: the Domain Model and the Application Model.

Domain Model

This Model is intended to remain the ”core” status of a Model before splitting it into two. This means that the current Domain Model should build the objects, classes, supporting the essence of the problem, and not directly contact outer layers (regarding gathering information) because this Model is the information producer.

Application Model

The application model, on its behalf, belongs to a much ”upper” layer positioning, being built to support the View and provide information to add a second layer of data abstraction in a particular project. If the Domain Model is used to profile the logic, the application model would provide the ”know-how” and ease the layer separation process.

MVC on web application development

The book [34] gives an insight into why the MVC pattern is so useful for web applications.

The article states that the Model-View-Controller helps create dynamic and interactive web applications, becoming one of the most well-accepted programming paradigms for developing scalable and dynamic web applications. The main reason for it relies on allowing developers to have a trustworthy guideline to follow and solve recurring problems, creating flexible, reusable, and modular applications. Although more efficient, applying the MVC pattern is also a more complex approach because it goes against the usual practices that encourage developers to partition the application as early as the design phase.

According to the article, the MVC pattern architecture has been widely used as an efficient approach to developing Web-based systems that allow for a back-end (or server-side) programming component, especially those requiring access to a database system. The pattern is supposed to isolate the business logic from the user interface, which allows for better separation of components, making the system easier to change and manage. With this, the purpose of the article is to compare the MVC model to others in terms of development time, maintainability, and ability to support and enhance integration amongst

designers and programmers. It was confirmed that the MVC pattern checks all boxes. A graphic representation of this studies' architecture will be listed below for better familiarity with the MVC concept.

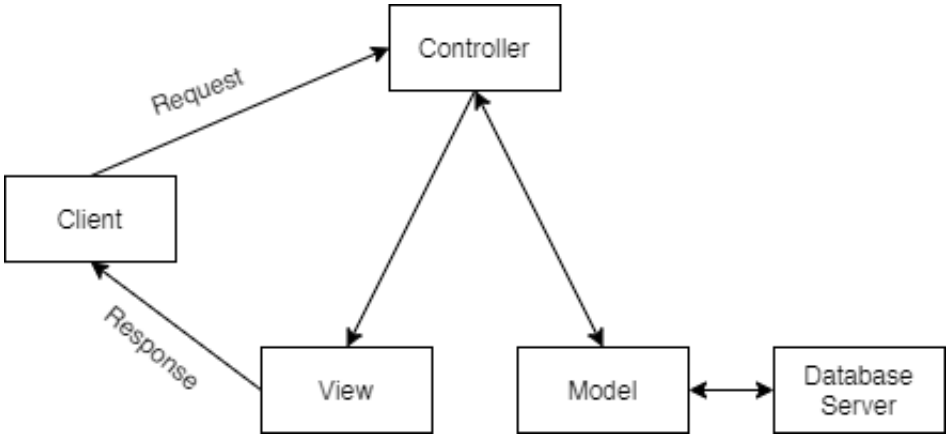

Figure 2.3: MVC Architecture from the study

2.1.6 API technology

API, as described in the article [24], stands for Application Programming Interface, and it is used to establish communication between client programs and web services. A Web API is the face of a web service, directly listening and responding to client requests. In terms of its story, the article [28] refers that the advances of the software development field dictated the emergence of new solutions on how software is developed and tested. One of those solutions is using Application Programming Interfaces (referred to as APIs), being considered a primary way of constructing large software solutions on top of a standard technology platform.

APIs have been present in these systems since the advent of personal computers, exchanging information between two or more programs. Besides this, the API technology only started being used on the web was witnessed around 2020. Since then, the article states that we appear to have reached a point where we live in the API economy due to its popularity, becoming an essential factor of the digital ecosystem.

According to sources in this article, the APIs support software reuse by providing pre-implemented functionalities to be applied when needed, without creating other code instances, improving the project's general workflow. The API is an intermediary gateway between the client and the user, allowing for better communication management from a software development perspective.

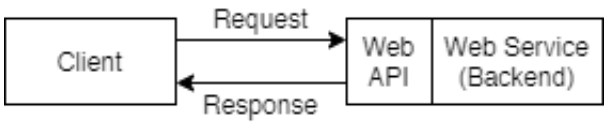

Figure 2.4: Example of an API architecture

2.2 Rule-based systems

Rule-based systems are the core abstract topic behind this work, as the Business Rules concept probably would not exist if the idea of Rule-based systems were not created in the first place. The book [17], in a very early approach, described the Ruled-based systems as the way to automate problem-solving know-how and provide a means for capturing and refining human expertise and provide commercial viability.

In the special section of this document, the author states that Rule-based systems constitute the best currently available means for codifying the problem-solving know-how of human experts. With that in mind, the author then lists properties that most rule-based systems share, those being:

- Incorporation of practical human knowledge in conditional if-then rules;
- Ability scales with knowledge base;
- Solve a wide range of complex problems by selecting relevant rules and combining the results in appropriate ways;
- Adaptively determine the best sequence of rules to execute;
- Explain conclusions by retracing lines of reasoning and translating the logic of each rule employed into natural language.

Rule-based systems are intended to address the need of capturing, representing, storing, distributing, reason, and digitally applying human knowledge. Typically, know-how on these systems is represented as conditional rules, and that know-how defines the systems themselves. This know-how is used for practical problem-solving knowledge. That knowledge consists of various kinds of information that the author lists as the following:

- Specific inferences that follow the specific observations;
- Abstractions, generalizations, and categorizations of given data;
- Necessary and sufficient conditions for achieving some goal;
- Likeliest places to look for relevant information;
- Preferred strategies for eliminating uncertainty or minimizing other risks;

- Likely consequences of a hypothetical situation;
- Probable causes of symptoms.

The author even describes some examples of rule-based systems, which are listed in the above-shown image.

Problem	System functions
Equipment maintenance	Diagnosing faults and recommending repairs
Component selection	Eliciting requirements and matching parts from na electronics catalog
Computer operation	Analyzing requirements, and selecting and operating software
Product configuration	Eliciting preferences and identifying parts that satisfy constraints
Troubleshooting	Analyzing situations, suggesting treatments, and prescribing preventative measures
Process control	Spotting problematic data and remedying irregularities
Quality assurance	Assessing tasks, proposing practices, and enforcing requirements

Figure 2.5: Applications of RBSs

By incorporating know-how acquired incrementally and transparently, Rule-based systems open up relevant computing applications usually not addressable by alternative techniques. Some of these technologies are usually best applied in areas where the supply of quality human workers is insufficient for reasons that may vary with the skill needed to do a task or when repetitive tasks are the primary use case.

Although partially accurate, this article, from 1985, can be somewhat outdated in some principles, considering the digital world we live in nowadays. For this, we will present some alternative and more recent approaches.

The article from [23] remounts to 2019. It defines a Rule-based system as a previously used product for complex use cases, combined with the insurgence of artificial intelligence. These types of products started to be relevant in areas such as finance and medicine. The article enumerated some of the advantages of representing knowledge in Rule-based systems, which are the following:

- Acquisition and maintenance, because using these types of systems can abstract lots of the work from the programming side, easing the overall process;
- Explanation, representing knowledge explicitly allowed systems to reason about how they came to a conclusion and use this information to explain results to users.
- Reasoning, where decoupling the knowledge from that knowledge process enabled general-purpose inference engines to be developed.

In terms of system architecture, the authors describe these systems as a composite of five components. Those components will be explained below:

- Knowledge Base, which is the area that contains all the knowledge necessary for understanding, formulating solutions, and solving problems. The knowledge in this

types of systems contains both factual and heuristic knowledge, one being fact-based and the other being based on experimental experience, respectively;

- Inference Engine, which is compared to the brain of the knowledge-based system, uses the control structure (the rule interpreter) and provides a methodology for reasoning. It is the component responsible for analyzing and processing the rules, performing matching antecedents from the responses given by the users, and triggering rules. The major challenge in this process is to run the set of existing rules and reach a conclusion;
- Knowledge acquisition that is the process responsible for accumulation, transfer, and transformation of problem-solving expertise from experts to the computer program, enabling the program to expand its knowledge base;
- Explanation facility, which is the subsystem that explains the system's actions. That explanation can decide whether the data amount is good enough or if it needs additional information.
- User interface that relates to the system's closest layer to the user. It provides components such as menus, and the graphical interface, built to optimize the relation between the user and the system. Creating an effective interface creates the best way to abstract the other components and create an easy-to-use product.

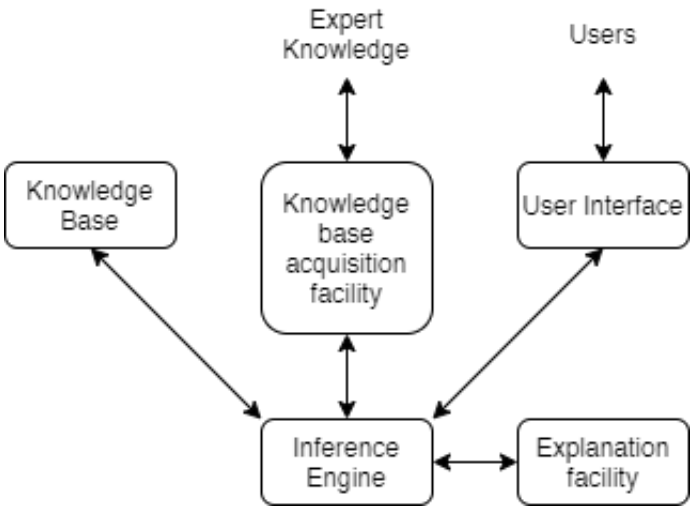

Figure 2.6: Structure typical rule-based system

2.3 Business Rules

As described in the book [15], Business Rules as a concept is not a new subject. Numerous different early approaches classified Business Rules as database constraints, rules, or

policies that govern the behaviour of the enterprise, and many others.

With the current use of Business Rules, its use is more complicated than primitive approaches. We have to consider that these rules are everywhere and do not work just inside an enterprise. Nowadays, its use involves implementing rules between enterprises (for example, with taxation) and with customers, which shifts the paradigm of the early definitions. With the growth of digital solutions, these rules had to have a digital representation, so software products started to be built around them.

A business rule is nothing more than a condition executable on a machine and clearly defines a pair of condition-action chains, creating a response to a given condition. As stated in the article [19], this notion allows us to have two condition-action cases that are demonstrated below:

Figure 2.7: Basic business rules example

Considering this example, some instances that correlate to Business Rules are the following:

- When a customer places his tenth order on a website, offer him a free shipping discount on his next purchase.
- If a system user does not access the website for fifteen days, automatically send an email to incite his return.
- If a user tries to order without an account, show an account creation popup.

These examples serve the purpose of showing us that the options are endless and are merely dependant on who their user is. These rules can affect all sorts of business decisions, whether implemented in business or organizational solutions. Although improving the worker's workflow and overall efficiency while allowing scalability, this automation level is defied with its challenges.

The document [32] defines Business Rules as a part of business logic, treating the concept as an intermediate to support some processes, like applying to an insurance product or social benefits. That process requires workflows of validation, and Business Rules can support that action. This logic will further be developed in the study presented by this article.

2.3.1 Business Rules Management Systems

Continuing the description given in the book [15] about Business Rules, the author provides insightful context as to why Business Rules Management Systems are needed.

The author used the example of how object-oriented programming relates to why Business Rules need to integrate management systems. As stated in the book, one of the problems behind classical programming was affected business logic after maintenance or code change. Maintenance costs were estimated to gross 90% of the total project cost, as changing a part of the code would imply that other code areas were directly affected. Object-oriented programming changed because changing an object does not propagate or harm other objects, easing programming's overall efficiency. Applying the same concept to Business Rules, aggregating different business logic in similar places would also increase maintenance time upon a business rule change.

With this in mind, creating "Business Rules" objects would also be very beneficial, as they would only be related to their relative objects, improving object separation and reducing costs. This means that specific rules would be encapsulated within their related objects, and general rules would be aggregated (normally organizational policy).

According to the author, to promote integration, the separated rule sets should be maintained in a central repository, which would lead to programming abstraction and reduce the business' costs on changing/maintaining those rules.

Considering this approach, the main challenge is a system that bridges objects and rules to ensure flawless communication and efficient performance while assuring optimal maintenance levels.

2.3.2 Business Rules/Automation challenges and benefits

While scalability grants an overall increase in performance, there are some trade-offs with these solutions. In practical terms, if something is built to improve efficiency, there is a specific time when that activity is time-consuming.

In a software project, that time is usually right after the planning stage because the configuration occurs. With this, automation intends to be time-consuming at the beginning of the project to improve time-efficiency in its future stages, reusing processes. During a presentation on the Systems Configuration and Management class, the presenter [18] spoke about how this notion translates to a DevOps approach. DevOps relies on ensuring configuration automation, and that work fully replicates the sense of automating processes with Business Rules.

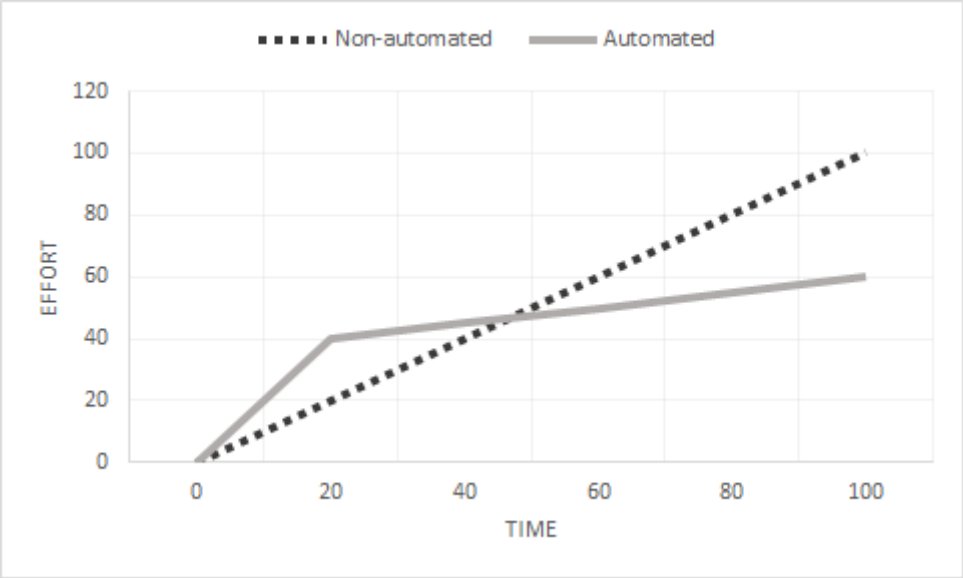

Figure 2.8: Comparison between the establishment of an automated project versus a non-automated one

Besides this graph being an empirical approach, it intends to represent the team's initial effort to make in order to achieve a more efficient solution in the future. This initial effort is beneficial for the company because it allows for a scalable solution that reduces the team's overload later.

As mentioned in the book [15], some of those benefits can be an overall faster development and maintenance and audit ability improvements. Due to its generalization, this implementation can also lead to time efficiency in reusing business logic and promoting consistency across the application.

With this, all the operational tasks will be significantly supported by this workflow improvement. This improvement is derived from the tool's simplicity, which eases maintenance, reduces costs, and improves system flexibility.

These improvements represent a big step to free up resources that can promote the company's technological progress.

2.3.3 Related applications

Business Rules are now widely used across different platforms and systems in different social areas. Some applications, such as [8] on a more theoretical approach, [5] on a Pharmaceutical Business Rules application, [16] on a more systemic approach, or even [1] on a financial credit application example, demonstrate how widely diverse the use cases can be. Business Rules are the solution to almost all business-oriented problems in which process automation is related. Every specialized area can receive Business Rules implementations, mostly because automation is a widely needed process. All businesses across the spectrum will eventually use some form of Business Rules.

Two of the four examples will now be covered in more detail.

2.3.4 Related applications - Business Rules in e-Government Applications

As stated in [8], there is already existing work on governmental applications due to its particular limitations. Those limitations are usually based on a generalized governmental resistance to changing things up. External organizations, internal policies, and law changes usually limit how much a governmental service can evolve.

The article states that some of the benefits of using business rules are the differentiation of organizations due to each peculiar business rule set, the possibility of its reuse, flexibility, and the ability to make decisions in real-time.

The proposed approach is called BRAIN (Business Rules for Adaptive Integration) and was designed to distribute e-Governmental services across local administrations.

This article also provides a case study that allows direct comparison to the SIIP application. It is a simple transaction example, but it shows how business rules interact with the system and the surroundings.

The example is based on a taxation use case that involves three entities (citizen, financial institution, and public administration) and relates to this case.

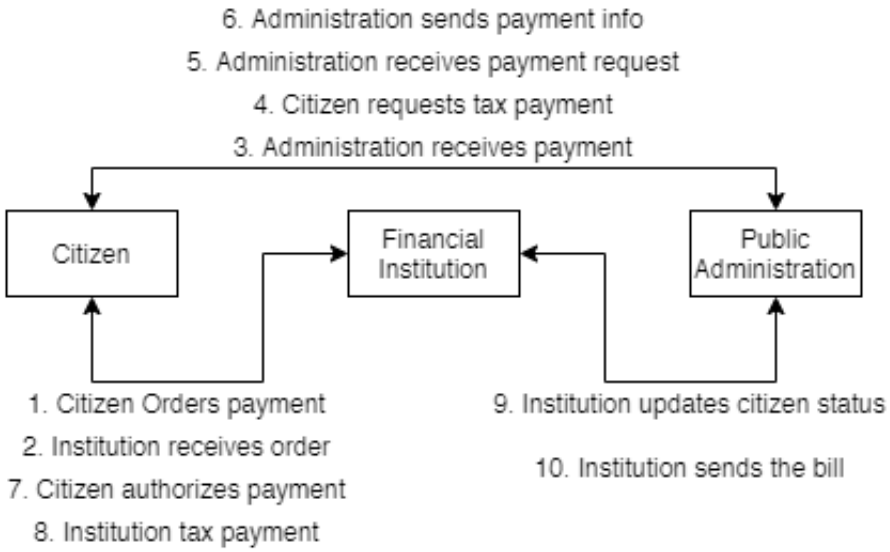

Figure 2.9: interaction between the citizen, the financial institution, and public administration on a tax payment use case

As shown in the figure above, some intermediate processes need to be checked.

With that, the need to request anything is already a significant drawback of this system, and that is where Business Rules can help, automating those intermediate processes.

In this case, given the tax payment's periodicity, an automatic request could be generated instead of forcing the citizen to take the initiative. These routines would significantly improve the system and improve workers' workflow by not dealing with these tasks.

These are the kinds of improvements desired to adapt to the SIIP application.

2.3.5 Related applications - A Business Rules Design Framework for a Pharmaceutical Validation and Alert System

A more practical approach can be found in [5], which proposes and discusses implementing a Business Rule framework that impacts the alert system that validates drug prescriptions.

Pharmaceutical validation is an active part of the prescription process, as it controls parameters like the recommended drug dosage for each case.

This publication describes the whole design process behind creating a set of Business Rules that trigger an alert if the drug dose is inappropriate for each patient. The results ended up being positive. The study shows that 24.39% (36 478) of the drug prescriptions (149 561) were triggered by the alert system, which shows the impact of using Business Rules.

As mentioned above, Business Rules are in their representative form, if-else clauses, and this pharmaceutical area is no different.

[6] demonstrate some examples of these conditions, was the following:

- Given a patient with a benign prostatic hyperplasia disease, treating him with Anticholinergics while the patient is over 65 years old automatically triggers an alert system while also providing advice for the pharmacist and the prescriber as Anticholinergics are targeted as usual developers of acute urinary retention, as seen in [14];

- Given a patient with Chronic renal failure, treating him with Nonaspirin NSAIDs while the patient is over 65 years old automatically triggers an alert system while also providing advice for the pharmacist and the prescriber as NSAIDs (non-aspirin nonsteroidal anti-inflammatory drugs) are known to cause heart attacks or strokes, which can be particularly dangerous on older people, as seen in [11];

- Given a patient with constipation, treating him with Opioid analgesics while the patient is over 65 years old automatically triggers an alert system while also providing advice for the pharmacist and the prescriber.

These conditions perfectly represent business rules' usage, as they separate a disease, the used drug, and the age target. These three parameters allow the system to build conditions between them and automate behaviours with given premises.

Conditions like these can be applied to our SIIP project. That is the primary goal of implementing Business Rules, as they provide an automatic solution to some business processes that ultimately improve the business' workflow.

2.3.6 Related applications - System and method for detecting the source of media content with application to Business Rules

[25] describes creating a workflow to determine the source of audio or video contents on a network. This information is precious in our current technological context, as digital media platforms need an efficient way to control what is published on their platforms and enforce copyright protection. This article provides insight into the usage of Business Rules in this regard.

This tool analyses audio and video elements in a piece of content by comparing the sample with known data. With this type of information, the article explains how Business Rules may act to help this cause.

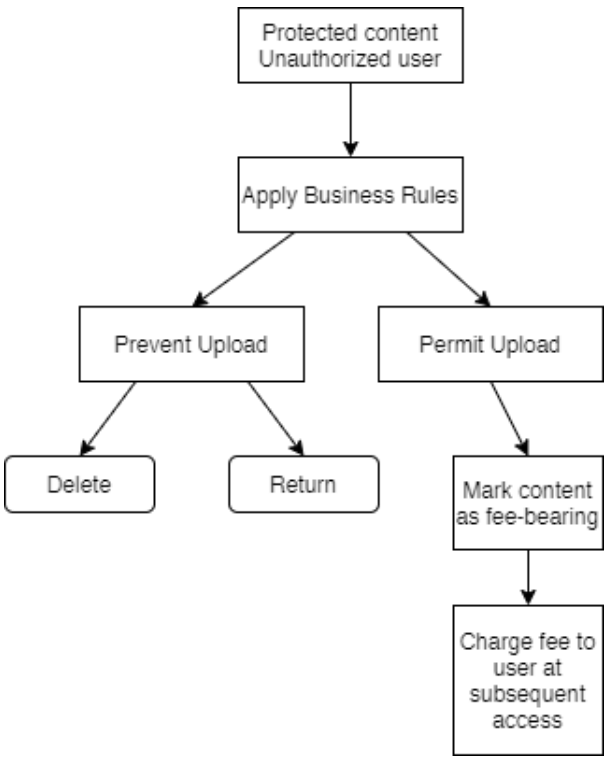

Figure 2.10: example of a Business Rules application workflow

The image shown above results from the article's presented software deciding that the piece of analyzed content violates copyright rules. With this in mind, Business Rules' implementation intends to determine how to solve copyright infringement. The available data, gathered in the previous states of this workflow, is then used to create business rules related to these actions. That evaluation will decide the outcome, prevent or allow the

upload, and decide the future steps to take on the decision. This article is essential to establish a connection to the Business Rule's implementation workflow. Business Rules should take decisions based on data, which will be done in this project.

2.3.7 Related applications - Verification Capabilities for Business Rules Management in the Dutch Governmental Context

[32] reiterates a governmental approach to the usage of Business Rules. The introductory part alludes to the increase of Business Rules' use on a business level, referring to the need to translate laws and internal policies into logically correct business decisions.

Considering that the study is intended to cover governmental needs, this might be an appropriate reference to the SIIP project's usage of Business Rules.

This study conducted a three-round focus group that led to a collection of 28 verification capabilities of Business Rules. The found verifications can be pretty helpful for this work. The actors of the article decided to fit these verifications into four distinct categories:

- Decision requirements level verification
- Decision logic level verification
- Fact level verification
- Generic verification level

These four levels split the 28 validations that can serve this work's purposes in exploring Business Rules use cases. The validations vary from documentation verifications, input verifications, or omission verifications listed in the image below.

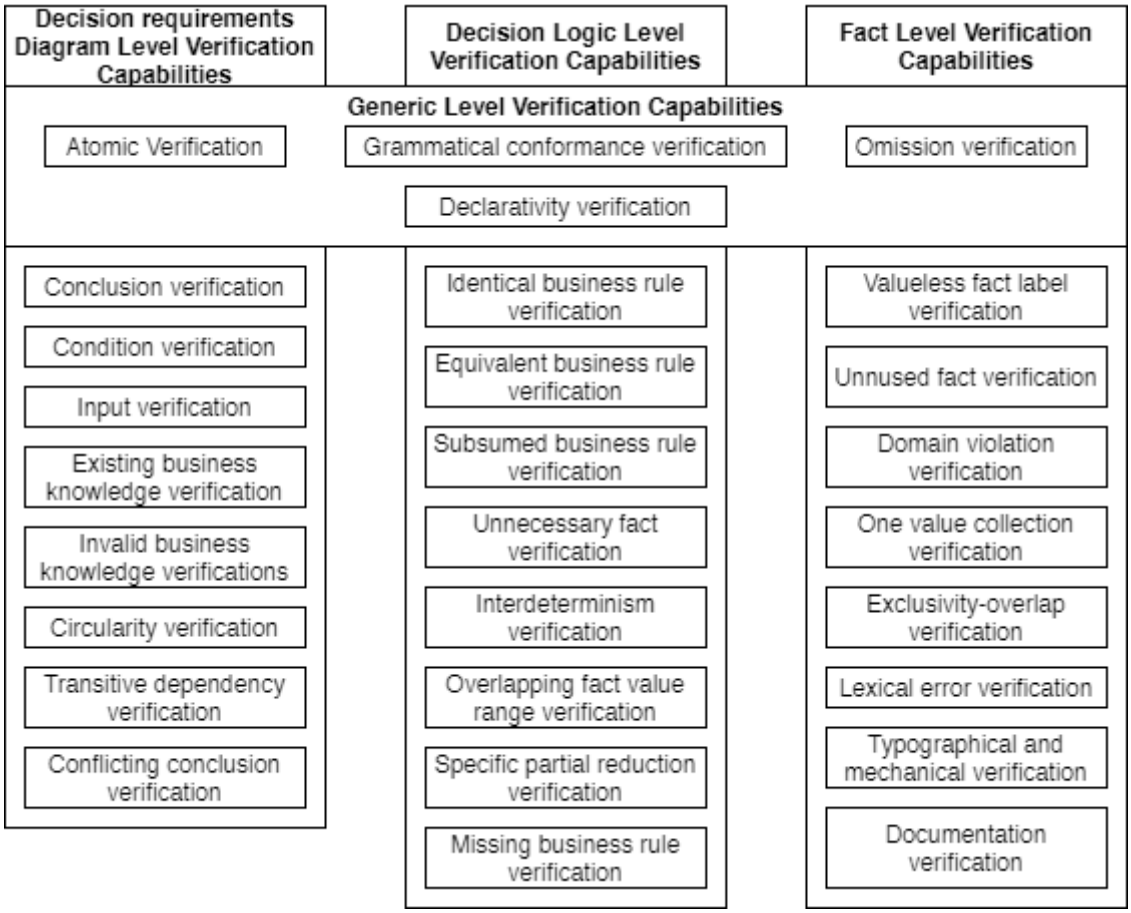

Figure 2.11: List of 28 verifications found in the presented study

Some of these verifications are valuable for the intended rules to be developed and may serve as an example for future Business Rules implementations in the SIIP application. This project involves many document management and control of values and processes, which can feature those implementations.

2.3.8 Frameworks

Frameworks are essential in this work, considering that the chosen framework will run the whole Business Rules system.

An incisive analysis must be made with that in mind, covering the project's context and requisites. To fully comply with the project's needs, the emphasis should be to maximize this implementation's performance.

The implementation of this Business Rules decision structure is an extra push to the client's initial requisites. For this exact reason, this implementation should be as invasive as possible while aligning the work's needs with the client's.

Most Business Rules frameworks are costly, as they can control a very complex system. In the SIIP project, perfectionism level is not critical, so avoiding extra costs would

greatly help this implementation.

After research, some Business Rules frameworks were analyzed, and those will be listed below. While the platform is not decided yet, this listing should put all the cards in the table and help choose the optimal framework.

Honorable Mentions

As stated earlier, this implementation is supposed to be as non-invasive as possible. With that in mind, that level of invasion also applies to financial costs.

Implementing these Business Rules is supposed to be made on relatively uncritical and straightforward areas of the project, inciting Open-Source products.

There are some complex (and expensive) alternatives on the market, such as Fico Blaze Advisor, IBM's Rule Management System, Pegasystems' PegaRules, Microsoft BizTalk Server, or Oracle's Business Rules engine.

Drools

Drools, or JBoss Drools, is one of the most used Business Rules frameworks. Its development started in 2001 and is now in its fifth version.

According to the book by [4], the service is separated into five modules: the engine, the event processor, the workflow creator, the management system, and an optional search algorithm.

The tool itself is purely programmed in Java, which is a factor to be considered, knowing that the SIIP project's base is C#. This obstacle would have to be fixed by creating an abstraction and serving the Business Rules component as a service while not fully integrating it with the SIIP project.

Being one of the lot's complete tools, the user interface is already an excellent product, with workflow and decision table systems, graphical condition editors, and complete visualization models.

Drools.NET

As stated in the Drools section, one of the main problems connecting that tool to our project is the discrepancy between programming languages.

Drools.NET intends to implement Drools to match the C# programming language with some slight syntax touches. Although looking like a viable option, this tool did not get very high development support, and its documentation is now very outdated, making it a very unviable option.

NRules

The tool's website states that NRules is an open-source rules engine for .NET, based on the Rete matching algorithm. The documentation of this tool can be found on the website [33]. This algorithm is also used on the platforms mentioned above, as it is considered a convenient approach to rule-searching. This solution may not be the most complete in terms of the graphical approach. However, its C# compatibility effectively keeps the system's coherence and easing the process of relating business rules with business logic. The tool's community is very active, which can be a very beneficial point.

CLIPS

As seen in the article by [31], CLIPS is an expert system building tool that provides a complete environment for the development and delivery of rule and object-based expert systems. CLIPS is very used in projects, such as seen in [27] and [26]. The main challenge is integrating CLIPS with the SIIP project's language, which would require extra work. One more negative point is that CLIPS is a very complex and non-graphical language, which would not improve that aspect of the project.

C# Internal Approach

The SIIP application runs on a .NET environment, which uses C# as its main programming language. Upon research, it was found that there is a programming model for Business Rules using a C# approach based on object-oriented language applications for the use of Business Rules. With this in mind, some articles such as [35] and [30]. These articles define a baseline for the implementation of Business Rules within the mentioned scope. This approach allows for a more isolated development, considering it has no real dependencies on a user interface, which is perfect for atomic integration within the SIIP application.

With all this information acquired, Parameters essential to choosing the perfect Business Rules engine to implement in SIIP are compared. That table is shown below.

	Compatibility with SIIP	Support	Cost	Complexity
Drools	Medium	Medium	Open-source	Medium-Low
Drools.NET	High	Low	Open-source	Medium
NRules	High	High	Open-source	Medium
CLIPS	Low	Medium	Open-source	High
C# Business Rules	Very High	Medium	Open-source	Medium

Table 2.1: Comparison between Open-Source Business Rules engines

Upon analyzing the options, using NRules allows for the best compatibility available. When compared to an entirely original C# implementation, NRules has the advantage of having an already-developed system capable of triggering the rules with its general implementation. The use of NRules will be mentioned in a later chapter when development related topics are covered. Other approaches are not viable due to compatibility or technology issues but serve as examples for the development section.

Chapter 3

System pre-development analysis

This chapter intends to set all the foundation stones to develop the SIIP project and its integration of business rules. As this work involved being in a project intended for an external client, the decisions made were heavily impacted by the relations between the client and the operating company. For this reason, this chapter will also explain all the personal perspectives, challenges, and reasons behind choosing the Business Rules theme and overall work explanations.

3.1 Case study: Public Sector's digital transformation

According to the context explored in the motivation section of this document, it was decided to create a Case Study to confirm the theory stating that the Public Sector is struggling more than the Private Sector in terms of digital transformation and proficiency.

The study was called "Public Sector's digital transformation.". It was created through the Google Forms platform, distributed online to various people to get a solid dispersion and result. The form was also distributed to multiple fields of the society, with the variable being a person belonging to the Public or Private sector.

This form considers four different categories that are conditioned to the individual's preliminary answers. The structure comprises demographic questions, public sector questions, private sector questions, and general questions about the public sector. The decision behind this approach allows students and private-sector workers to have an opinion about the public services from an outside perspective.

3.1.1 Demographics

The form was responded by 52 individuals, featuring 17 males and 35 females, whose distributions will be listed, along with the age distribution, in the following charts:

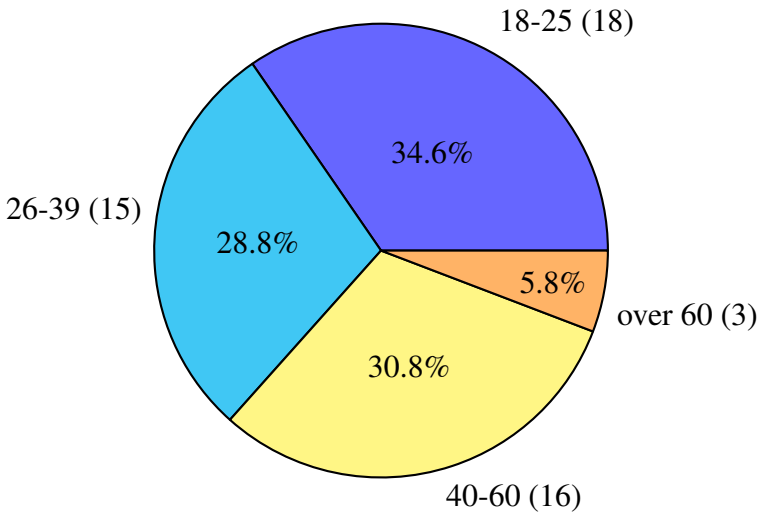

Figure 3.1: Digital transformation case study: Age distribution

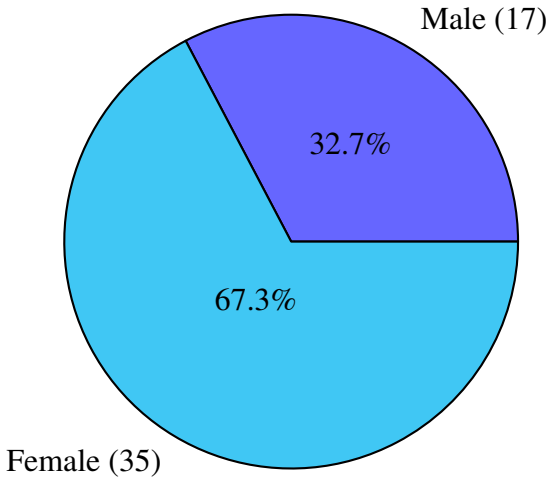

Figure 3.2: Digital transformation case study: Genre distribution

Most enquired individuals concentrate between high school (secondary school), licentiate degrees, or a master's degree in academic abilities. The last enquired parameter was if the individual was a student, a worker, or a combination of both. This resulted in the following chart:

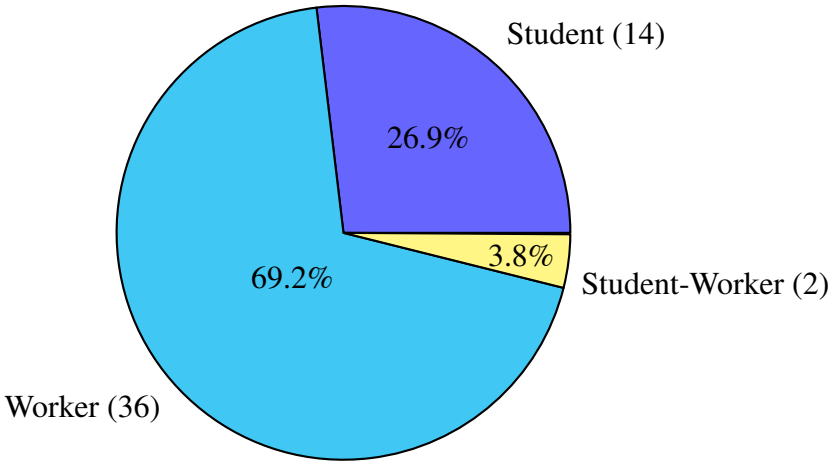

Figure 3.3: Digital transformation case study: Working status distribution

The form is programmed so that the individuals' answers condition the topics that appear to respond. Such defining parameters are the individual's occupation and whether he is part of the public or private sector. That decision is based on the form's composition, which defines whether the individual has to answer public or private sector questions, or none if he is a student, directly going to the general questions.

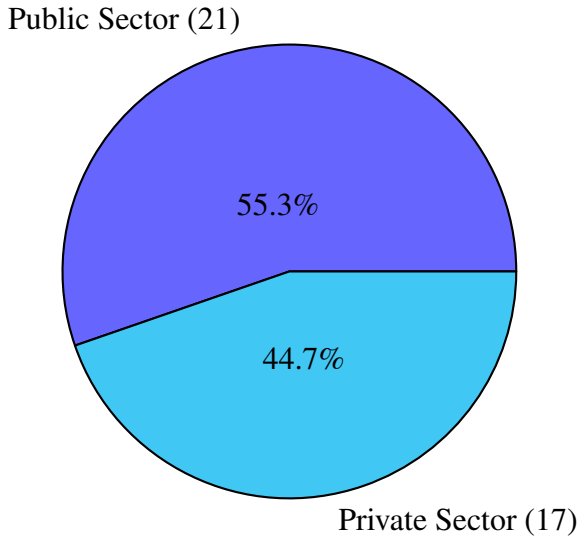

Figure 3.4: Digital transformation case study: Sector distribution

3.1.2 Comparison between public and private sectors

This subsection relates to analyzing two sections of the form and comparing them together. As stated above, the individuals filling the form had the chance to respond if they were workers from the public or private sector and were prompted to answer questions based on their choice. Those questions are the same for each sector, allowing a direct

comparison analysis. Each question will be presented and subject to graphical and numeric comparison, as listed below.

My organization tries to integrate the most recent technologies in its operations, 1 meaning a lack of agreement and 5 meaning a total agreement

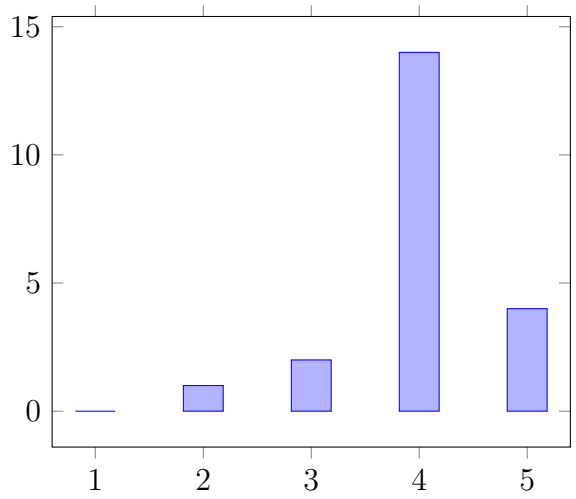

Figure 3.5: Technology integration: Public sector
1 - 0%, 2 - 4.8%, 3 - 9.5%, 4 - 66.7%, 5 - 19% **Average: 4.0**

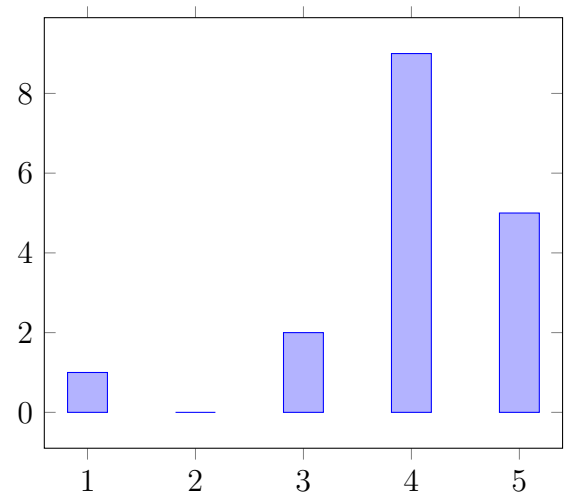

Figure 3.6: Technology integration: Private sector
1 - 5.9%, 2 - 0%, 3 - 11.8%, 4 - 52.9%, 5 - 29.4% **Average: 4.0**

In terms of this topic, both sectors appear to be even. That correlation can be seen with similar spikes in levels 4 and 5. With this, we can deduce that this question retrieved similar results due to its subjectiveness regarding what should be considered ”most recent technologies.”.

My organization is affected by the surge of new technologies in the society, 1 meaning a lack of agreement and 5 meaning a total agreement

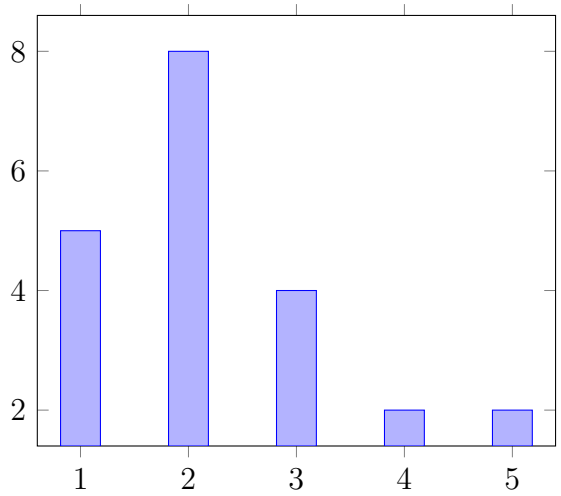

Figure 3.7: Organization affected by society's use of technology : Public sector
1 - 23.8%, 2 - 38.1%, 3 - 19%, 4 - 9.5%, 5 - 9.5% **Average: 2.43**

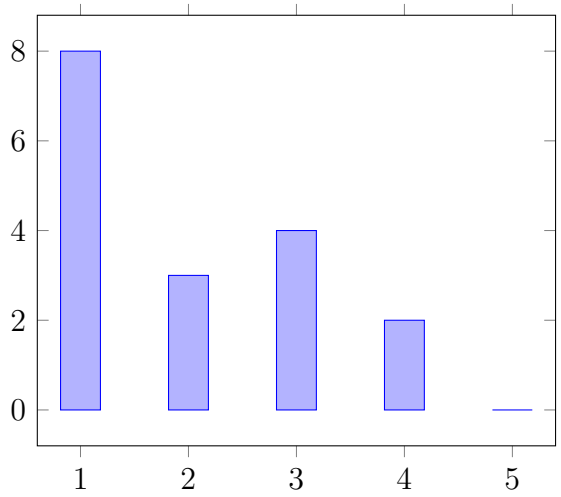

Figure 3.8: Organization affected by society's use of technology : Private sector
1 - 47.1%, 2 - 17.6%, 3 - 23.5%, 4 - 11.8%, 5 - 0% **Average: 2.0**

Although similar patterns are similar, there seems to be a clearer disagreement on the private sector, which may refer to the private sector being more prepared in terms of technological evolution, which would align with the study's purpose.

I feel difficulties with the integration of new technologies, 1 meaning a lack of agreement and 5 meaning a total agreement

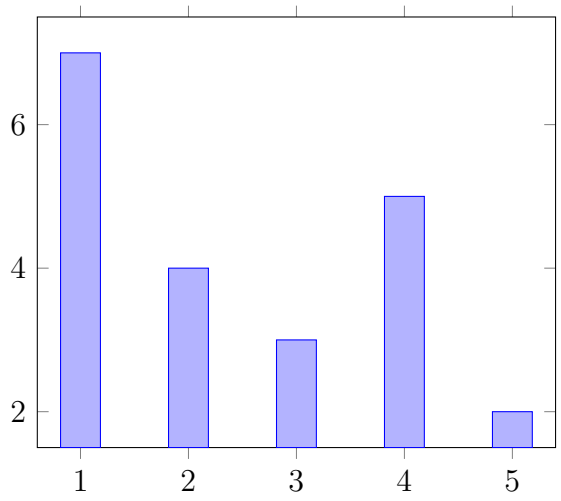

Figure 3.9: Personal difficulties with technological integration : Public sector
1 - 33.3%, 2 - 19%, 3 - 14.3%, 4 - 23.8%, 5 - 9.5% **Average: 2.57**

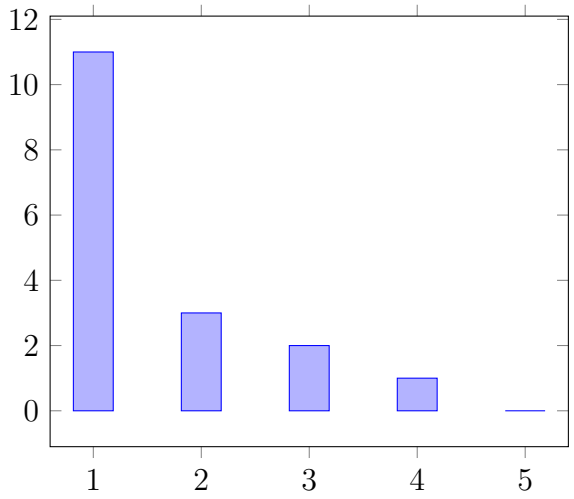

Figure 3.10: Personal difficulties with technological integration : Private sector
1 - 64.7%, 2 - 17.6%, 3 - 11.8%, 4 - 5.9%, 5 - 0% **Average: 1,59**

This comparison denotes that the private sector individuals usually embrace new technologies on a better note, feeling less difficulty with new technology integration when compared to the public sector, where some individuals noticed a greater sense of difficulty.

Others' difficulties with the integration of new technologies, 1 meaning a lack of agreement and 5 meaning a total agreement

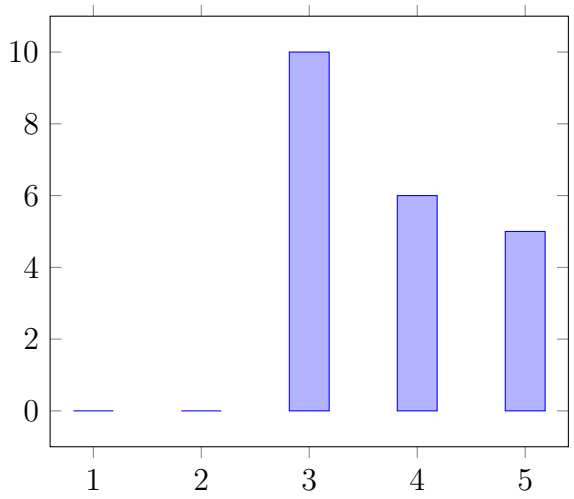

Figure 3.11: Others' difficulties with technological integration : Public sector 1 - 0%, 2 - 0%, 3 - 47.6%, 4 - 28.6%, 5 - 23.8% **Average: 3.76**

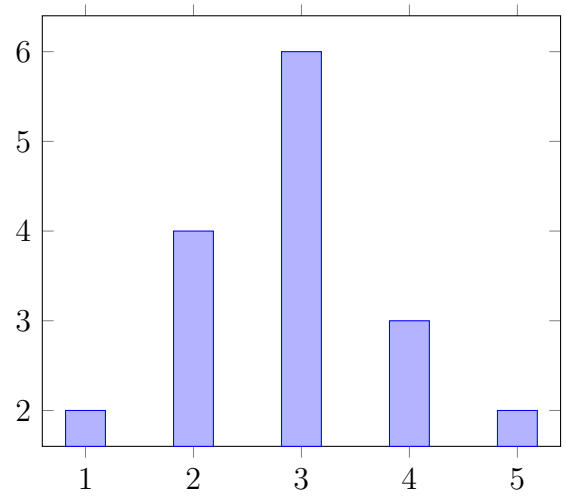

Figure 3.12: Others' difficulties with technological integration : Private sector 1 - 11.8%, 2 - 23.5%, 3 - 35.3%, 4 - 17.6%, 5 - 11.8% **Average: 2.94**

In this topic, there seems to be an apparent certainty of difficulties in the public sector. This question is exciting because it also opposes the first-person perspective. This perspective is usually more subjective, in opposition to a third-person one, because it depends on how someone contextualizes within its environment. In talking about the others, the difficulty is more objective and translates to a more significant pattern difference when comparing both paradigms.

Classification of the organization's technological level

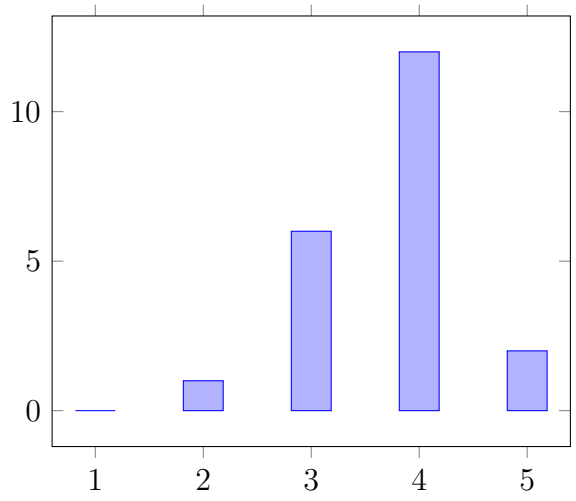

Figure 3.13: Classification of the organization's technological level : Public sector 1 - 0%, 2 - 4.8%, 3 - 28.6%, 4 - 57.1%, 5 - 9.5% **Average: 3.71**

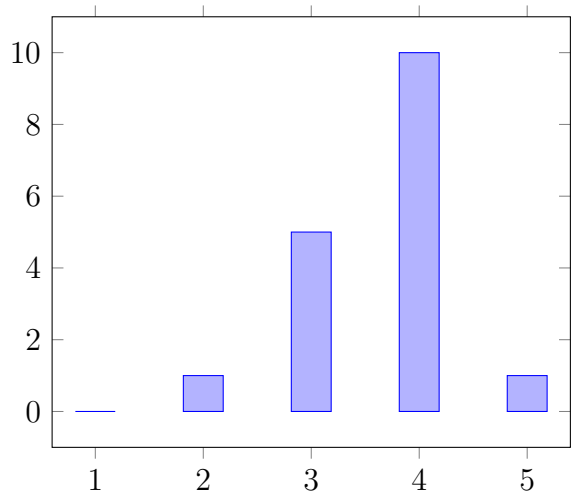

Figure 3.14: Classification of the organization's technological level : Private sector 1 - 0%, 2 - 5.9%, 3 - 29.4%, 4 - 58.8%, 5 - 5.9% **Average: 3.65**

In this question, the very similar results can, once again, be explained by subjectiveness. This subjectivity can relate to the reality of both sectors, as a public sector can be traditionally more attached to lower levels of technology, meaning that a level 4 on the public sector may be equivalent to a much lower number on the private sector.

Comparison between the individual's organization in terms of technology with other organizations

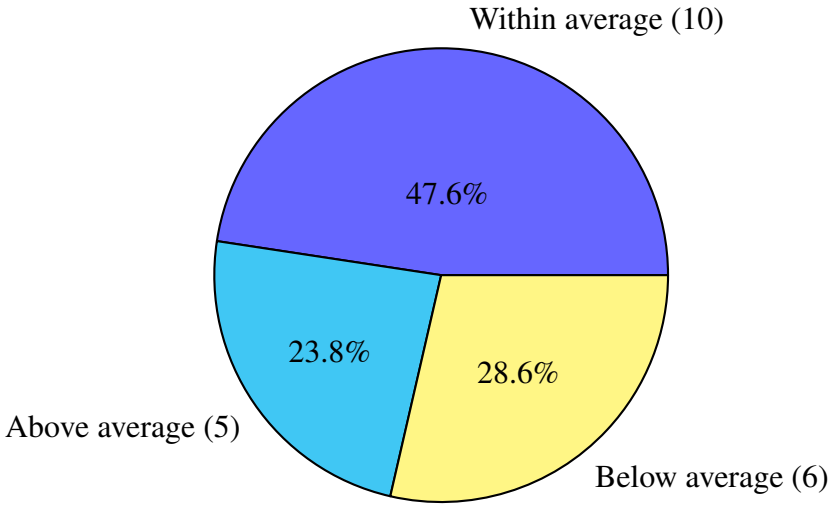

Figure 3.15: Comparison between the individual's organization in terms of technology with other organizations: Public sector

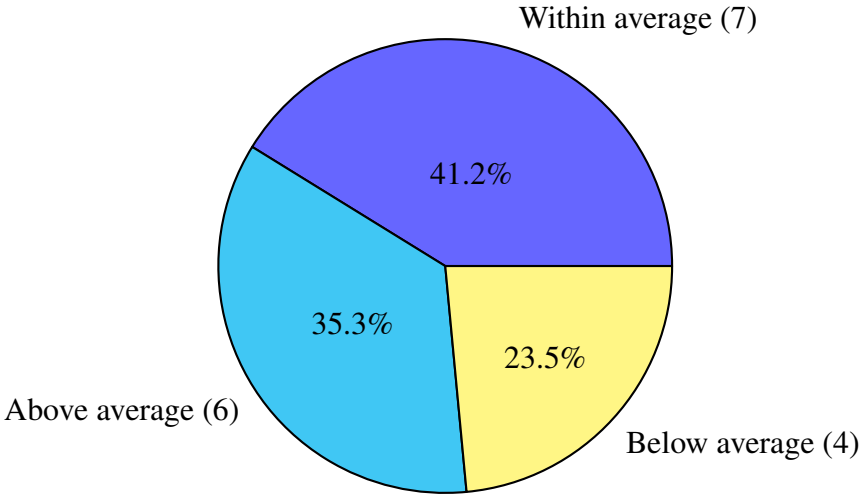

Figure 3.16: Comparison between the individual's organization in terms of technology with other organizations: Private sector

Between the two presented charts, it is visible that the private sector responses are more optimistic on comparing their organization to others. This is seen with the below-average responses being comparably lower and the above-average responses being considerably higher than the public sector, showing what this study intends to focus on.

The main catalyst to technological advances within the organization

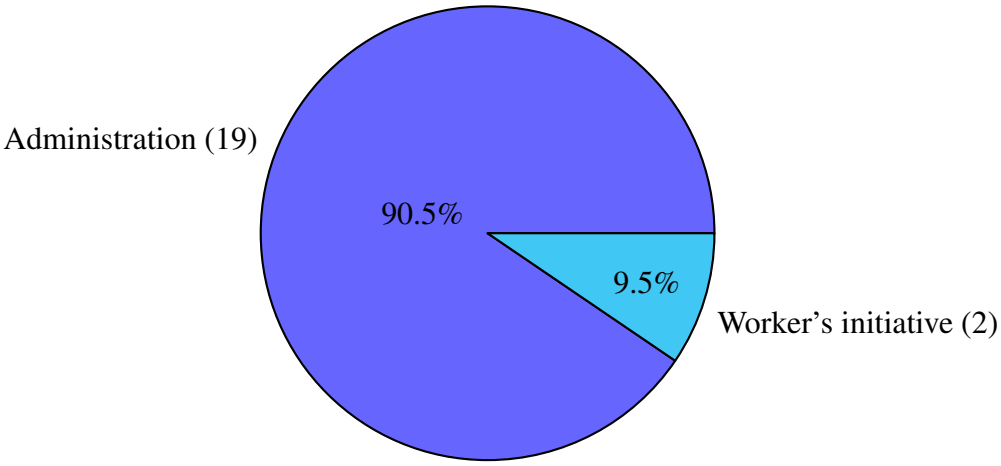

Figure 3.17: The main catalyst to technological advances within the organization: Public sector

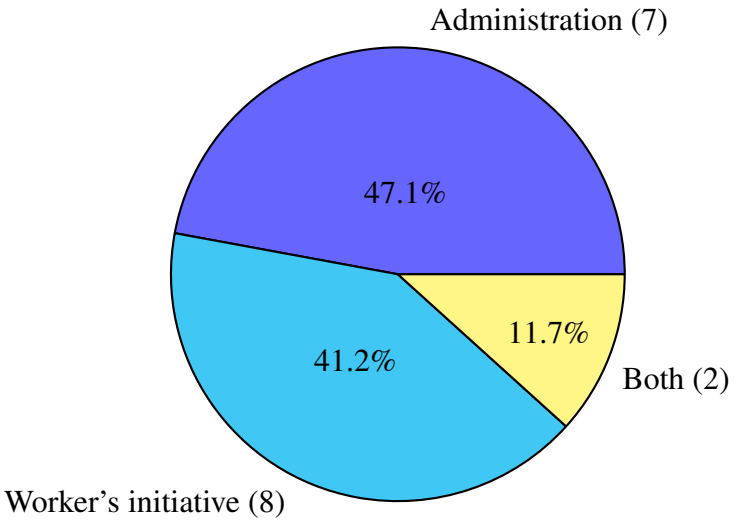

Figure 3.18: The main catalyst to technological advances within the organization: Private sector

This is one of the questions that can serve the most to this study. These results signify that the public sector relies more on administrative decisions, leading to a closed environment with limited policies and decisions. This factor can aggressively limit innovation with technological advances, with employers feeling stuck on the current approach an organization takes, with less susceptibility to change.

The pandemic improved the technological adoption of the organization, 1 meaning a lack of agreement and 5 meaning a total agreement

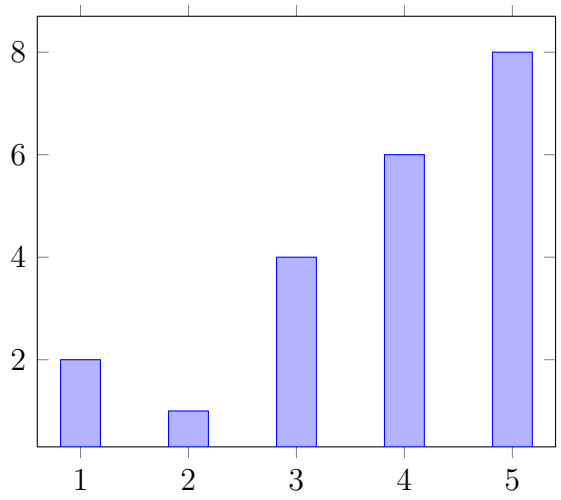

Figure 3.19: The pandemic improved the technological adoption of the organization : Public sector
1 - 9.5%, 2 - 4.8%, 3 - 19%, 4 - 28.6%, 5 - 38.1% **Average: 3.81**

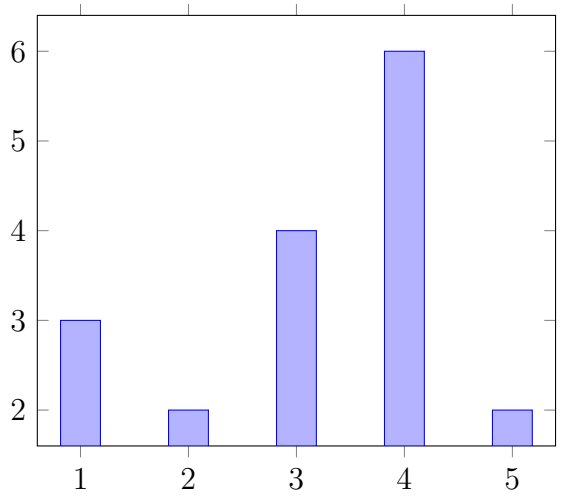

Figure 3.20: The pandemic improved the technological adoption of the organization : Private sector
1 - 17.6%, 2 - 11.8%, 3 - 23.5%, 4 - 35.3%, 5 - 11.8% **Average: 3.12**

Regarding the technological evolution with the pandemic, these chart results might be explained by how evolved the private sector already was before the pandemic, with an already defined technologically-ready structure. As for the public sector, most organizations probably did not have the required structures to embrace the current context, which triggered a need to improve the current systems.

3.1.3 Digital transformation in the public services

This subsection relates to the part of the form that was open to everyone, including students, public sector workers, and private-sector workers. It was an effort to retrieve external opinions on how public services work from citizens' perspectives. Most of the following questions were formulated based on the information presented in the Motivation section of this document to clarify the validity of the arguments presented in that section.

The public systems are well adapted to how society uses technology, 1 meaning a lack of agreement and 5 meaning a total agreement

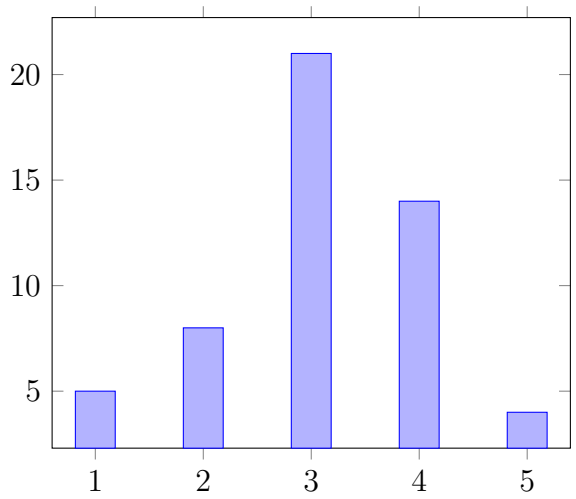

Figure 3.21: Case study: The public systems are well adapted to how society uses technology

1 - 9.8%, 2 - 15.4%, 3 - 40.4%, 4 - 26.9%, 5 - 7.7% **Average: 3.08**

The lack of digital solutions causes a distancing between the services and the citizen, 1 meaning a lack of agreement and 5 meaning a total agreement

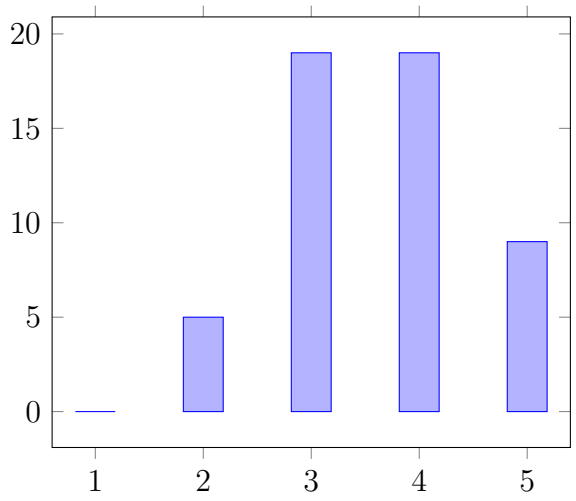

Figure 3.22: Case study: The lack of digital solutions causes a distancing between the services and the citizen

1 - 0%, 2 - 5%, 3 - 36.5%, 4 - 36.5%, 5 - 17.3% **Average: 3.62**

Feeling personally distanced from the public services due to a lack of platforms to promote communication with the services, 1 meaning a lack of agreement and 5 meaning a total agreement

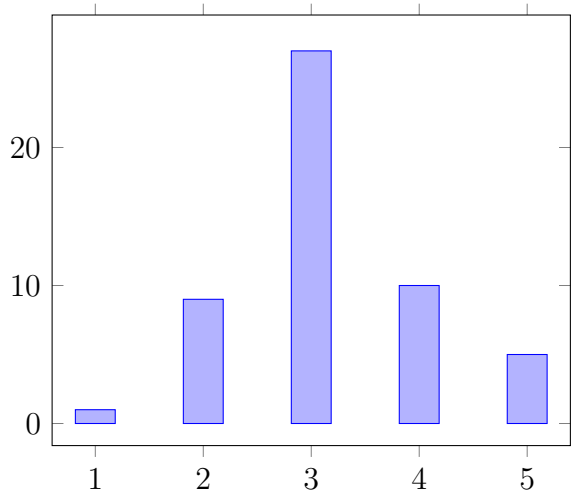

Figure 3.23: Case study: Feeling personally distanced from the public services due to a lack of platforms to promote communication with the services

1 - 1.9%, 2 - 17.3%, 3 - 51.9%, 4 - 19.2%, 5 - 9.6% **Average: 3.17**

Feeling the lack of public service digital platforms during the pandemic, 1 meaning a lack of agreement and 5 meaning a total agreement

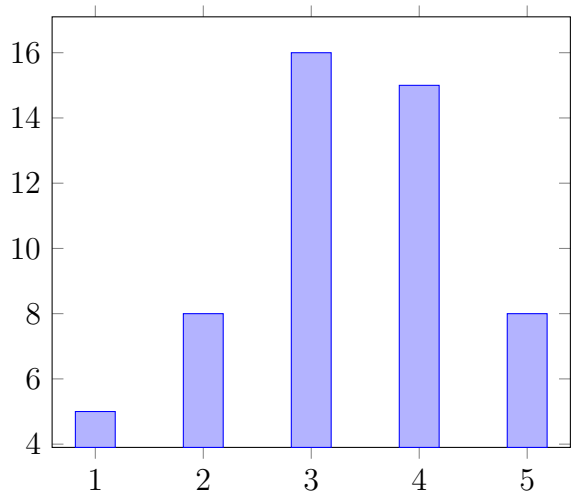

Figure 3.24: Case study: Feeling the lack of public service digital platforms during the pandemic

1 - 9.6%, 2 - 15.4%, 3 - 30.8%, 4 - 28.8%, 5 - 15.4% **Average: 3.25**

Feeling that the public system is technologically outdated when compared to private organizations, 1 meaning a lack of agreement and 5 meaning a total agreement

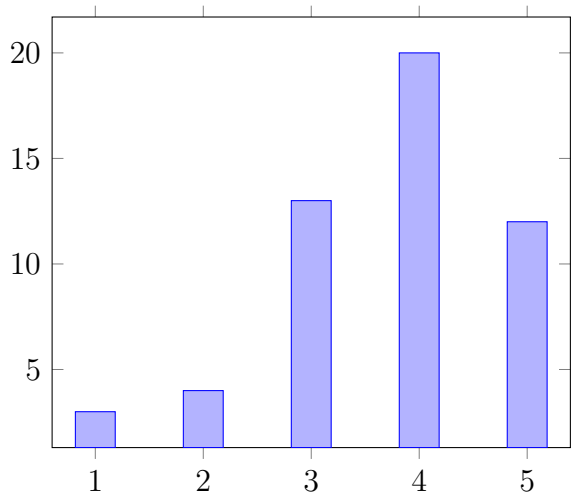

Figure 3.25: Case study: Feeling that the public system is technologically outdated when compared to private organizations

1 - 5.8%, 2 - 7.7%, 3 - 25%, 4 - 38.5%, 5 - 23.1% **Average: 3.65**

Worrying about the insurgence of new technologies and information attacks on digital platforms, 1 meaning a lack of agreement and 5 meaning a total agreement

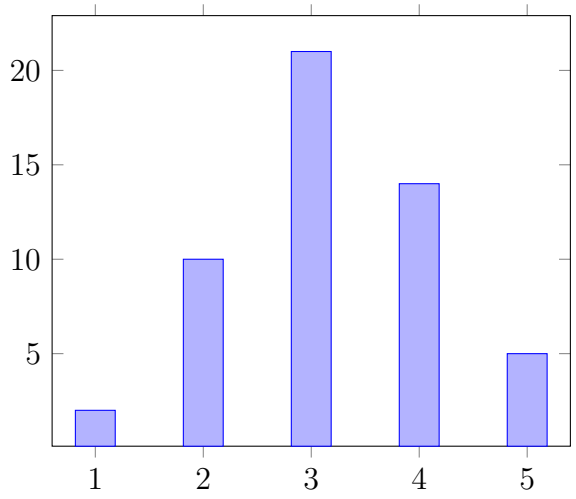

Figure 3.26: Case study: Worrying about the insurgence of new technologies and information attacks on digital platforms
1 - 3.8%, 2 - 19.2%, 3 - 40.4%, 4 - 26.9%, 5 - 9.6% **Average: 3.19**

Motives for lack of technological advances on public services

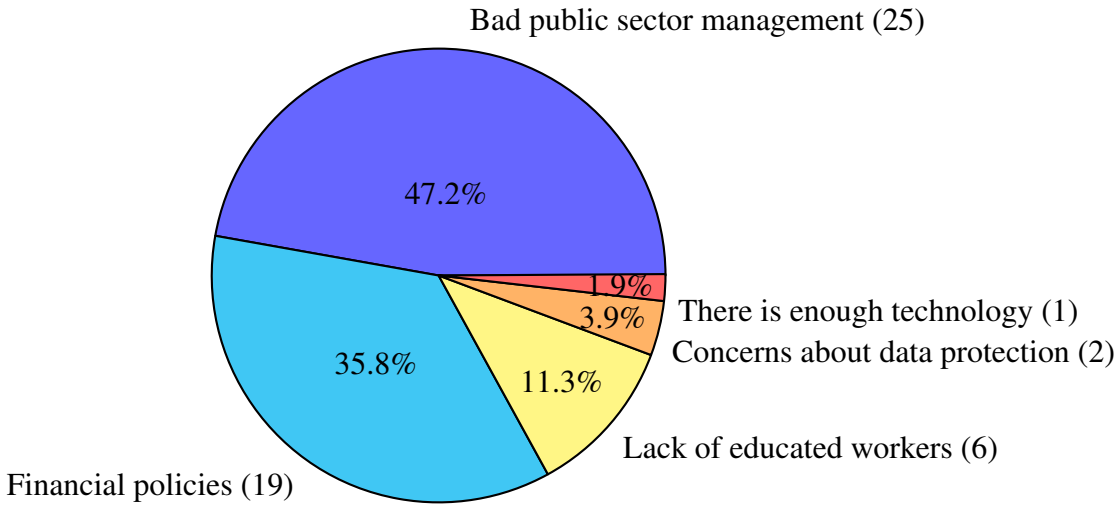

Figure 3.27: Case Study: Motives for lack of technological advances on public services

3.1.4 Case Study - Final analysis

After analyzing the form results, it can be said that all the arguments presented in the motivation section of this document are valid, just by how the individuals created a data set that points to that lack of efficiency in the public sector in terms of technological development. This form's results were essential to developing this work with these limitations in mind.

3.2 Preliminary project analysis

This section will explain some thoughts behind the first days at the SIIP project and how they influenced both the project and the work theme. That act includes client reunions and workflows analyzed and used to build the foundations of this work. All the strategic planning, excluding development-based questions, will also be listed here.

3.2.1 Initial impressions

Upon entering the SIIP project, some initial meetings took place to grasp the overall culture and context of the project. Those meetings introduced the client - Lisbon's Public Illumination Department (known as DIP) and typical communication workflows between the client and the company. Those communications seemed reasonable to a project of this magnitude. Besides this, the company culture was not a problem in the early stages, considering previous experience on a project at the same company.

Besides, the team presented some of the project's workflows, diagrams, and client desires.

3.2.2 The software

Specifically into software, the project is unique because its genesis remounts to the first contract between the client and the company around 2016. As mentioned in earlier chapters of this work, the project is based on the MVC model with variations based on client requests. The project is conducted mainly using Microsoft's frameworks for all the components of the project. The first meetings featured presentations of techniques/patterns such as Reflection and Repository. The project itself was also presented in those first meetings, and that presentation involved all the components listed in an earlier chapter that mentions system architecture. The software development will be covered in the next chapter for a better understanding of all elements.

3.2.3 The client and his impact on the work theme choice

The client, as mentioned before, is part of a Portuguese public entity responsible for the street's illumination. The project's objective is to create a software product to cover all existing workflows done by the client in a particular way in which they are currently done.

Within the first meetings with the client, some things became more pronounced, such as an evident lack of technological knowledge, an apparent lack of communication with the company, and a feeling of the two sides being on very different pages. That factor is demonstrated by the project's lifespan - around six years now, and the same lack of communication persists.

That factor was a significant barrier in choosing how to approach a theme to fulfil this work because choosing a theme that required more communication with the client would put this work at risk. Much effort was put into planning all possibilities and eventually stick to one. Without touching much into software, leaving it into its dedicated chapter, we will list some of the most critical factors that led to this work's theme choice.

Motive	Impact	Influence
Lack of technological skill	Limitations on theme choices	Negative
	Creation of something helpful	Positive
Lack of digital knowledge	Wanting to insert too much information into the software program	Negative
Lack of communication	No chance of creating personalized products for this work	Negative
	Very extensive delays in the SIIP project	Negative
Lack of organization	Company-client meetings are usually very subjective and not straight to the point	Negative
Badly defined requests	Most meetings are not effective because of the unwillingness to clear things up	Negative
Usage of older software	Inability to develop something related to modern concepts	Negative
Friendly communication	Meetings are complex but there is no animosity between both sides	Positive

Table 3.1: SIIP project's client analysis

The table above sums up all the aspects of the client during the time this work lasted. The combinations of factors that most influenced the theme choice are the technology-related topics and the communication. Those were seen as being fundamental features to which the work needed to adapt. In this case, as the communication with the client was not the best, it was needed to choose a theme that was not invasive, and that complied with the client's technological stack.

With this in mind, the optimal point of interest for a theme should be to affect the client positively, and Business Rules seem to be the perfect addition to the software. Their development is non-invasive, not requiring any communication because of its level of abstraction. This choice is significant because meetings were proof that adding an extra layer of requests to the client would delay the main project even more. This happens because the client wants to replicate their current technologies (Excel sheets) into the program, with a very complex amount of attributes for each component (especially in the storage part of the project). This part will be touched on in the next section.

Thus, the Business Rules choice tries to deliver while dodging all these limitations, building a system that does not interfere with the project's main scope, leaving both parties

with no extra workload.

3.3 Application's requisites

In the previous sections, when introducing the SIIP project, the application's different business cases were described. Those components were the Stock management, Request management, and Alert features. Those features are part of a business strategy from the client, and their requisites define the main scope of each feature. As stated before, the client is generally not that tech-friendly, meaning that the current system they used before the SIIP platform is based on Excel sheets programmed with some level of intelligence. For that reason, most of the client's requests are strictly based on things they use nowadays on their excel files because they have no real awareness of how to optimize a software project in terms of information management. This requisite list results from endless files with very complex information, especially in the Stock management area, where every item has around twenty different attributes. Some of the requirements for all the components mentioned above will be listed to better connect this theory with what needs to be done regarding software. These requisites are defined in internal documents shared between parties requested in meetings, and credit is due to those documents.

3.3.1 General requisites

These requisites are defined as general guidelines for the application's design and are the following:

- An automatic method for retrieving registries;
- Be able to adapt to different stages of the products (states or situations);
- Provide chained events;
- Be structured as a unique model that is dynamic enough to adapt to everything;
- Give users the needed advice and orientation on the application;
- Allow the editing of automated options;
- Cover the automatic production of documents and reports;
- Allow the creation of new document models;
- Allow for future evolution;
- Allow the integration of future models.

3.3.2 Request requisites

As stated before, the Requests are considered as being the core of the application. With this, the registries have to cover all the steps that make a request. This includes all the steps from its creation to its archival, such as:

- Whether the request was made by internal sources (such as supervisors or the inspect team) or external sources (such as external partner companies or citizens through the public application A Minha Rua);
- Each request much register its resolution priority that varies from Urgent-Prioritary-Common;
- Each request must register its relevancy that may vary from Important-Regular-Hypothesis;
- After each request creation, the mentioned request has to be geographically indexed to avoid false information;
- The following fields should be contemplated on the project:
  - Request promoter;
  - Request creator;
  - Contacts;
  - Numbering (Request numbers that are created following a special formula);
  - Dates;
  - Request type;
  - Request location and georeferencing;
  - Available means to solve the request;
  - Expected resolution method.
- Every request should directly reference all requests that are associated with it;
- The requests can be attributed to different moments called status or situation that should be automatically applied according to the operator's actions;
- Within each request status, there should be an automatic update on the current request phase.

Regarding the fields related to priority, relevancy, and methods, the client also proposed a categorization method discussed in the development section of this work because it was subject to a new and cleaner approach.

3.3.3 Stock management requisites

The stock management component is placed in this application to enable the management of all storage units. This placement is a very complex task due to the client's level of detail in the application. In terms of characterization, a Storage Facility should have the following attributes:

- Type;
- Name;
- Council;
- Address.

In addition to the storage facility, each facility is composed of shelves intended to store the items in inventory. The characterization of a shelf should be the following:

- Code;
- Current utilization rate;
- Capacity;
- Floor;
- Corridor;
- Section.

3.3.4 Alert requisites

Alerts are the component of this project that relates to the alert generated by the application when a user executes a certain action. Specifically, the SIIP application can be prompted on the request component. This also works the stock management component due to its utility in both fields. Characterization of the Alert component should include the following:

- Type;
- Description;
- User that triggered the alert;
- Creation date.

In terms of the stored components in these facilities, its structure is complex and too extensive to describe. These attributes requested by the client are detailed in a Microsoft Excel file. Most components have mutual parameters such as an ID, a name, or a model and add 50 attributes. Besides this, each component also has specific attributes to its characteristics. This request is a technological challenge due to the high amount of data that needs to be processed. All data is due to be manually inserted, which would also be a challenge for the client.

Chapter 4

SIIP Development

As mentioned in earlier chapters, SIIP is the base application of this work's execution. This chapter intends to show and detail the contributions to the SIIP application that eventually led to the Business Rules implementation. Business Rules is a dependent concept, meaning that its implementation only exists when connected to an application. The SIIP development chapter will feature design discussions and implementation details of the SIIP features made during this work. As stated in an earlier stage of this document, the main features of the SIIP development rely on the components such as Requests, Storage Management and Notifications. To better explain how these features work, how these workflows interact within the application should be re-stated.

The Request feature is the centre of the SIIP application, as all other features serve each request. A Request is created on two different types of occasions - external or internal submissions. The general workflow is the same for both types. The only difference is that the external type (usually submitted by citizens on external applications) is imported to the SIIP application. In contrast, the internal types are created within the application by internal staff.

After the Request analysis stage, a decision process defines the procedures needed to handle the request. Those procedures might involve the need to gather materials to fulfil the request's procedures, and those materials can be stored in the department's Storage Facilities or contracted to external entities. Controlling the Storage integration within each request led to adding a Storage Management component to the SIIP application. Besides this, the Notifications feature wraps the Requests and Storage Unit features, providing information to users upon any events.

Lastly, the Request association and disassociation features are also present in the SIIP application and will be developed from scratch as part of this work. The need to have associations comes from the client's desire to group Requests with similar root causes, especially the geographical location. For instance, if two light posts are separately reported to be broken, the assertion process of those Requests can create an Association between two requests, easing the procedure required to solve both problems. The prac-

tical result is creating an Association object consisting of two Request objects, nothing more than a link between these two entities. The Association and Disassociation features are essential to this introduction because those will be the main features developed in this chapter.

4.1 Early contributions

Early contributions to this project were made in the first weeks following the start of the work. The start of this work marked the first interaction with the SIIP application, which was then presented to increase familiarity with the project. This first overall presentation of the project was an opportunity to review the code that formed this project.

By reviewing some of the core parts of the code in the mentioned presentation, some parts were faulty, so the first small tasks were related to those changes. Reviewing the code was also an opportunity to grasp the general structure of the project, helpful in proofing this work, and the first meeting with the team to discuss possible changes. After the first steps, new project responsibilities were inducted. So the following steps on the SIIP application would be to fully develop the Requests feature of this work, both association of requests and dissociation.

4.2 Project logical structure

In order to better understand the SIIP application's structure, it is better to detail further how the project is structured in terms of code. With the amount of complexity in projects like this, splitting different components into smaller projects is recommended. That approach also eases explaining each component as the code itself is already encapsulated in a specific project.

The code that creates the SIIP application is separated into smaller modules or projects. These modules serve the purpose of separating code logic between different structures. The projects are an API project, a Database project, a Domain project, and a UMB project, composing the SIIP application in programming structure. This last one is responsible for managing the Umbraco component of the SIIP application, the content manager.

- **API Project** - This is the project area that configures and defines the access points to the use of the API technology on the SIIP application. Creating a component relies on creating two main elements in this project - A View Model and a Controller. The first relates to writing all the object's attributes, defining a constructor, and the other is responsible for establishing the routes to the API routing structure.

- **Database Project** - This project area is responsible for dealing with all the components of the database. This area does not need a new module on creating a new

component, but it is an essential part of the SIIP project. The project area comprises repository connections and database migrations that are automatically generated based on the state of the components present in the other parts of the application, allowing for a complete database rebuild at any point.

- **Domain Project** - This module is responsible for the creation of the model that represents the core of the object. This project is where database mapping is configured, with attribute tagging to constitute the attributes of the database.

- **UMB project** - This part of the project is the one who is aggregating the most components of a single object. As mentioned before, Umbraco is big responsible for integrating this project and associating the back-end and front-end components. For that reason, this is the module with the most complex structure, involving more files to create the object. There is the need to create a Controller to interact with the back-end and a set of models to ease data processing. In this case, it is recommended to construct programming objects such as a Create Model, a List Model, a Summary Model, and a Search Model to create a base object for each required activity.
  This module is also where the View structure is constructed, with a file in the ”cshtml” extension creating the web page in which the object is presented. With a direct link to that page, there is also a place for scripts, where the javascript file will provide analytical support to the web page.
  There is also a need to connect all these pieces in the Umbraco page, demonstrating the development components in detail.

4.3 Requests

As mentioned in the theoretical section of this work, requests are an essential aspect of this application. That importance derives from its centralisation within the system. Every aspect of the SIIP application is based on requests and the requests' statuses. With this, it was a maximum priority to develop this component to achieve a better project status.

The Request object, the main component, was already partially created before the start of this work. With that, there was a need to make some adjustments to the objects. Additionally, implementing some components from the objects to be created in the main component was also needed.

4.3.1 Request Association - Initial discussion

As mentioned before, the request association is the object that can store a link between two requests, which can be associated to ease the process of grouping requests that may be connected in terms of type. This component was completely new in the application's

structure, which involved a debate on the structure desired by the client. This structure, however, was not optimal, considering that the SIIP application started near 2017 and has been dragging itself since then. Therefore, a couple of team reunions were needed to conceive the optimal structure for this problem. Some questions were also made in the client reunions in order to grasp their intentions regarding this subject. The lack of communication from the client mentioned in the above sections made this decision take weeks to optimise. That factor caused multiple developed solutions to be changed until a consensual approach was reached, sometimes requiring a complete code re-writing. The following content is the final result, representing the consensual interpretation developed in conformance with the development team.

4.3.2 Request Association - Object creation

Considering the MVC pattern explained in previous chapters, this subsection will illustrate all the processes behind creating this feature, from the back-end to the front-end, with the general workflows analysed in this subsection.

Structural analysis

With the existing Request programming object, there was the need to plan how to integrate the request with the component that associates a Request with other Requests.

Creating a Request Association object was then needed to store all associations in the database efficiently. Each entry of the database would then be an association between two requests.

Database interaction

Unlike most MVC-based projects, the SIIP application has a different way of managing the database integration with the project's components. The SIIP application uses two design patterns to create the database logic without configuring the connection between the domain model and the database logic in the specific files. This connection is achieved using a programming design standard called the Repository pattern, which generalises how data is processed. In this case, the Repository pattern already has all general methods used in a database, connecting both object and database model. It is only responsible for receiving the request, knowing which entity to point and process the transaction. This repository is significant because it allows the SIIP application to manage database transactions with a single file, as all database functions are stored in that single file. The only changing aspect is the object and nothing else. This approach also allows for the database schema to be equal to the object class in the application code, as it creates a direct mapping from that code to the database.

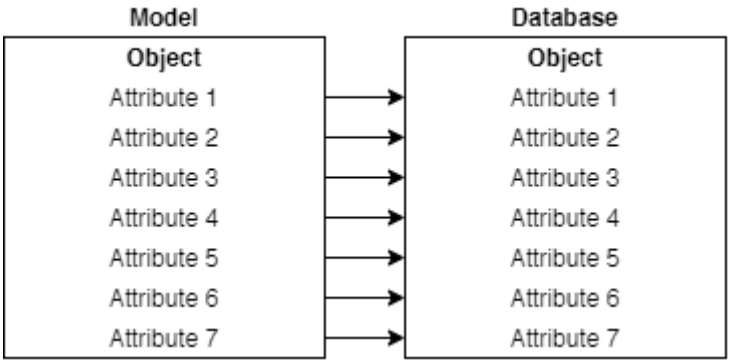

Figure 4.1: Example of the Repository pattern relation between the model and the database

4.3.3 Request association - Code development

Following the abovementioned guidelines, this subsection will detail the steps made to create a programming object in the SIIP application and integrate it within the system.

API Project

In the API Project, as mentioned above, two files need to be created.

The first one represents creating the model in the API to create a connection between the logical structure and the API. This implementation can be seen in A.1.

With the object already having a logical representation in the API project, a connection is also needed. The API works as a routing system, and the route defines an address to associate a process to that address. With that, making an API call to the specified route would do an action related to that object, as implemented in A.2.

With this in mind, using an URL similar to www.siip.com/SolicitacaoAssociacao would represent the page from this object. The representation is possible due to the direct connection between the API and serving the object page.

Analysing the code sample shows that the defined controller is connected to the original object (to be detailed later) and the View Model detailed above, connecting the object and the API.

Domain Project

This is the component responsible for defining the core logic of the object. This class is directly connected to the API in the route definition and defines how the database should handle each attribute. As mentioned above, this class defines the database schema because the repository created and directly maps all the information to the database. Implementation can be seen in A.3.

In this case, it can be seen that this object is essentially a connection between two requests. Each request is represented by its identification on the database, being applied as a foreign key. It is also essential to state that the IsDeleted attribute is part of a client requirement requested to disallow the removal of information within the system. Considering an object as deleted would be to have the IsDeleted attribute marked as True.

UMB Project

The UMB project is the most complex module of the SIIP application. As mentioned before, this component joins every component within the SIIP ecosystem.

Starting with the connection to the back-end, another set of models needs to be created. This serves of purpose properly transporting the information between the layers of the MVC pattern. These layers are responsible for receiving information from the back-end and constructing more specific elements to be applied in the front-end. The create model, by the opposition, is used to create new objects on the front-end and transport the information to the back-end. Its fundamentals are similar to the other models, as seen in A.4.

The Summary Model intends to create a specific representation of the object to trim the unneeded information from the component. This model is also crucial for table constructions because it allows naming the attributes more explicitly for display purposes, which is the primary intent of this model, seen in A.5.

In order to further create a better and more organised programming experience, there is the need to create a List Model as well, allowing the creation of sets of Requests that can directly create a table, as seen in A.6.

Fulfilling another client request, the program structure also featured a search model to save the searched information. This model is also reliable in mutating its attributes according to whether the client wants a basic or advanced search. The implementation can be seen in A.7.

With all the Models detailed, it is time to advance to the main components of the Request Association module. The front-end Controller is responsible for all the programming logic behind the other components. The controller is responsible for core functions to load the page while also gathering information from the back-end of the program in order to process the said web page, which can be seen in A.8.

In this piece of code, we have the two main deciders of this component - The indexer and the object creator. The first decider is called when the API Route gets triggered when the SolicitacaoAssociacao suffix is used on the web page. This function generates a Request Association entity to grab the object type and creates the Search Model to display the page.

The Create function is the function that takes the selected requests to associate and processes the request using an API call received on the back-end, creating all the as-

sociations between elements. These two functions directly interact with the web page, presented in A.9.

To simplify the View page, all code behaviours related to the inclusion of Association Requests were made in the javascript file listed below. This file is responsible for all logic behaviours present within the View page. It also controls the initialisation of the page components and manages the events due to page interactions. This process also includes creating the associations, which can be seen in A.10.

After the code listings, there is something additional that needs to be done in order to complete the setup process.

As mentioned in the introductory stage of this chapter, Umbraco is a Content Manager System that intends to connect all components. Although the code is all set, Umbraco needs a connection point between all files to correctly display all the information and provide an effortless level of data processing. These behaviours represented in the code will be demonstrated below with images to ease understanding of how the page works.

4.3.4 Request association - Umbraco linking

As mentioned in an earlier section, Umbraco links all the components presented in the structural part of this work regarding the Request Association object. Umbraco intends to emulate the structure presented in the coding demonstration. With this, three central components need to be inserted into the Umbraco page, which will now be detailed.

A different component must be created in a primary instance to allow the other components to be ultimately linked. A component called template was then created. That template is a direct correlation to the View that was created. This file left empty is essential to fill all requirements imposed by how Umbraco works. The graphic demonstration can be seen in A.1.

After the creation of the template, a second phase can now be processed. This phase correlates to a controller while compared to code development. This component is called a Document Type and aggregates the template within itself, allowing for linkage between elements. The configuration process relates to nothing more than choosing the suitable template to the document type, as seen in A.2.

Finally, the component connects everything - An Item, under the Content tab. This is the most important module of the Umbraco construction under the SIIP application. This item correlates to a hybrid approach to the API and the Domain modules of the MVC code development by creating the API route that connects with the programming system and allowing the document type and template choice. These attributions complete the full linking of the Umbraco program with the SIIP development code, allowing the SIIP service to work. The process of this creation can be seen in A.3.

4.3.5 Request association - Page demonstration

After demonstrating the code development and the Umbraco linking process, the last part of this phase relates to explaining the principles behind the programmed interface. Some client requirements were behind the design decisions that reflected the page design, and those requirements will be detailed in a further stage.

In essence, the screen's purpose is to allow users to have many options on searching for requests, with basic and advanced options.

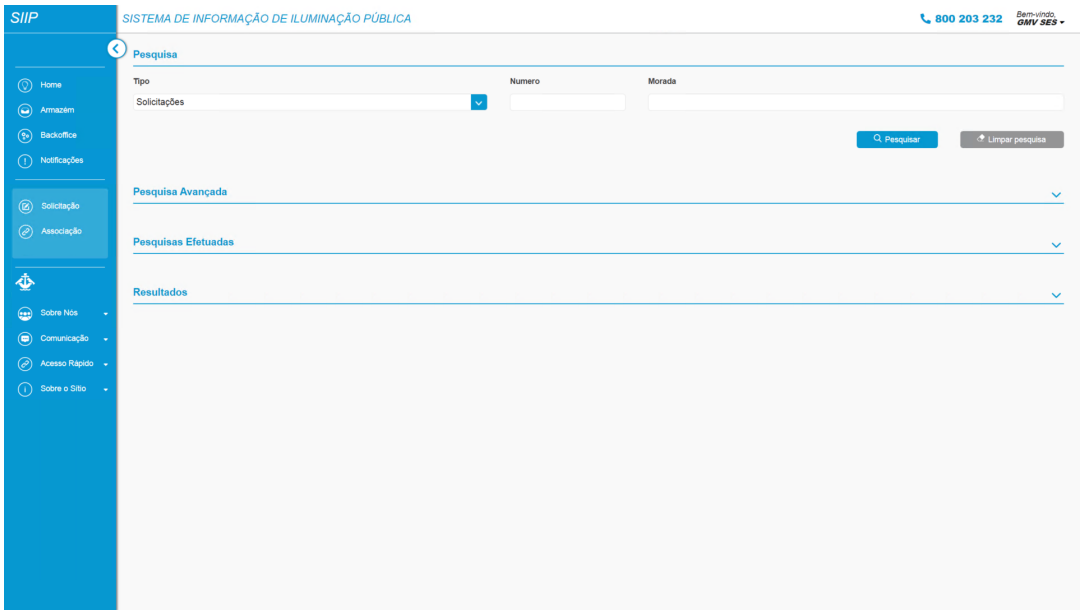

Figure 4.2: Request Association - SIIP/Page layout

After defining its search parameters, the user gets presented with all the registered requests in the system. Those include all the registry details. The list of requests, besides its information, contains the option to choose the wanted requests that get put to a secondary list that is dynamically created.

This secondary table approach was chosen to fulfil the client's functionality habits, using elements designed to simplify understanding and interacting with the interface. After choosing the requests and mapping them to the secondary table, the user then gets the option of covering another requirement - to promote flexibility of request choices. With this in mind, a personalised column was added to choose the request chosen as the primary target. In this case, the main target is the request linked with all the others in a one-to-many relation. All these workflows are highly editable, according to the client's requests. After the selection is made, the user can create associations between the requests. This is possible because the information is processed to the back-end and confirmed afterwards. The graphic representation of the tables is shown in 4.3.

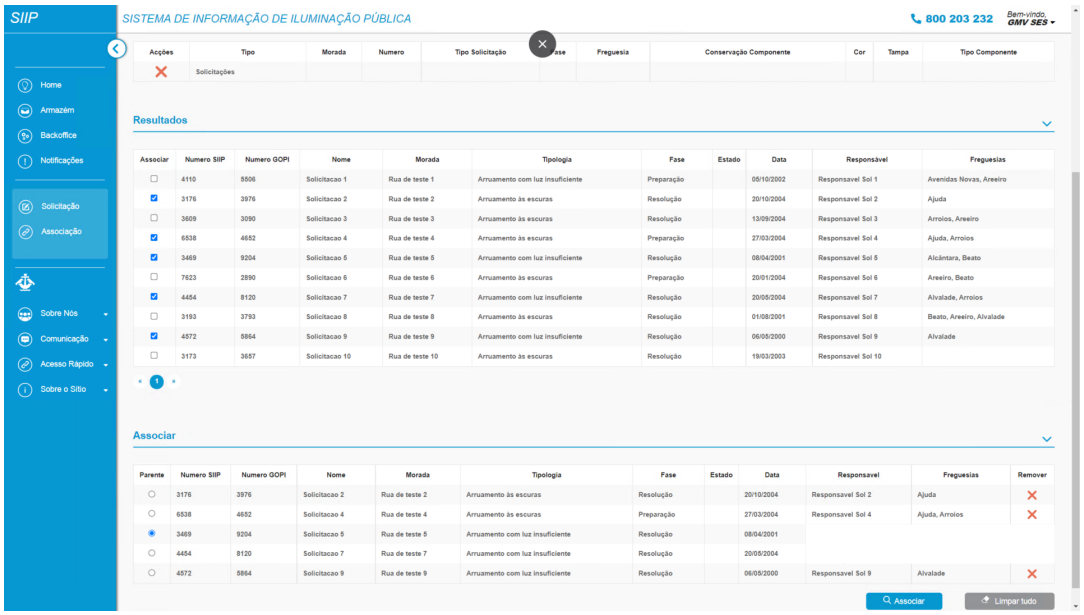

Figure 4.3: Request Association - SIIP/Associating Requests

4.3.6 Request Disassociation - Initial discussion

The Request disassociation regards the aspect of the client wanting to unlink two requests that, for some reason, were connected between themselves. This is the second prominent development feature of the work. Following the client requisites, a separate page had to be produced to disassociate requests from others. Although not as complex as the association structure, it poses a different variation of how an object works.

This difference comes from the structural needs of the project. By being an isolated object, the association page is not attached to any other class or entity, considering that the page loads with the same state for any instance (disregarding, for example, an Id pointer).

By definition, the Dissociation element is different from the Association one because it directly depends on the Request class. With this being said, the Request page in the SIIP application is unique, meaning that each request has its isolated page, differentiated by its ID, which is indicated on the page URL. For example, the Request page with the Id 1 should be located in the following URL: siip.local?id=1. The Dissociation page button is only visible through the Request page, meaning that going to the Dissociation page is always conditioned by the Request Id.

The code creation and Umbraco linkage process is more straightforward when compared to the Association object due to the Dissociation element being directly attached to the Request object. The creation of this element will now be explained.

4.3.7 Request Disassociation - Code development

By simplifying decoupling, this development mostly takes place in the UMB project mentioned above, which is the front-end part of the SIIP application. As mentioned above, the Disassociation element is directly linked with the Request class. With this, all functions and files were created according to that sense of subordination.

As the Disassociation page is created for its sole purpose, a View file had to be created in order to display the requested information from the specific request, which are:

- General information;
- Applicant information;
- Geographical Information;
- Request associations.

The code used to develop this representation can be seen in B.1.

Considering all the differences explained before between the Association and Disassociation elements, the page base is set by a single function in the Request Controller that loads the View according to a given Id. While in A.8 the Index function does not load any information related to the Id due to having a generalised approach, the opposite happens in the Disassociation implementation. That different approach can be seen in B.2. The code portion in B.3 demonstrates the REST call to delete the associations related to the associated Requests implemented on the controller to receive this call and process the information. That process can be seen in B.4, where the targeted associations are deleted, but that deletion is not permanent, as requested by the client.

4.3.8 Request Disassociation - Umbraco Linkage

An association is needed between data types and templates in Umbraco to complete the linkage between all elements. The main difference to the Association object is, as there are no objects regarding the Disassociation, everything will be linked to the Request object, similar to what has been made in the code development section of this feature.

Firstly, there is the need to create the View correspondence on Umbraco, which is the template. That creation can be seen at B.1.

By creating this template, it is now possible to link it with the needed component. As mentioned above, the Disassociation element only needs a connection to the Request object. So there is only the need to add the Disassociation template to the request's template list in the Document Types section of Umbraco, as seen in B.2. After this, the Umbraco connection is finished, and the Subdivisao template will directly connect to the Subdivisao function in the Request Controller, present in the SIIP application.

4.3.9 Request Disassociation - Page demonstration

With all the demonstrated results, the Disassociation page results in a simple and comprehensive interface, allowing users to remove Associations between Requests, as seen in the image below.

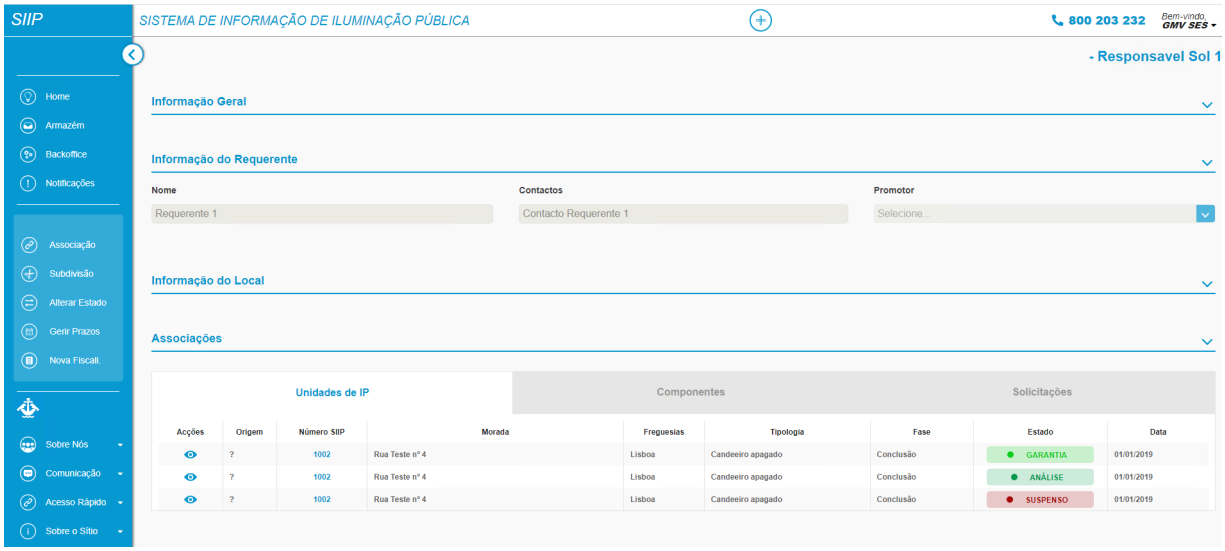

Figure 4.4: Request Disassociation - Page demonstration

4.4 Other contributions

This section represents secondary contributions that were considered relevant to advancing the project. These contributions happened on a dispersed time frame throughout the work period. Most of these contributions are made mainly based on the personal will to suggest change, creating a unique dimension to this work.

4.4.1 Priority and Relevancy Categorisation and Available means charts

As mentioned in a previous section, the priority and relevancy categorisation and available means charts were part of the proposals made by the client in one of the deliverable documents. This topic relates to the attributes present on the Request page that define the request's priority and relevancy and if the request has the proper means to be resolved.

The combination of elements generates tables with combined colour codes, defined by the images below. After reviewing this document made after a meeting and combining it with usability and design good practices, it was then decided to propose a critical change to this colour scheme.

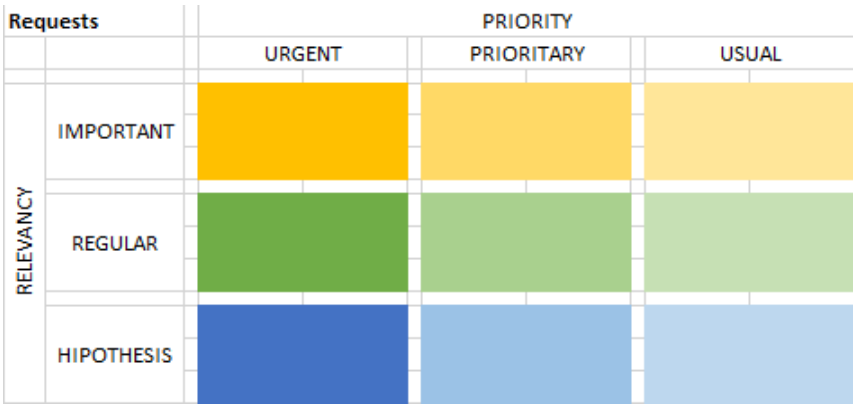

Figure 4.5: Priority and Relevancy Categorization - Client Proposal

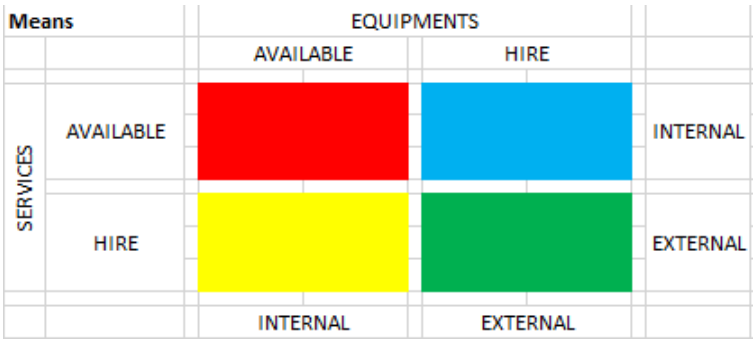

Figure 4.6: Means graph - Client Proposal

This revision mainly tried to conceive more adjusted colours to the client's needs and work better to simplify and promote semantic understanding of the situation. The colours, trying to represent a gradient of the severeness, gradually shift from hot to colder tones, from red to blue.

Regarding the means chart, there is a clear miss-conception of the colour scheme, considering that Available Service combined with Available Equipment is marked with a red tone, whose semantic value is the opposite of what it should be. The chart was also remade to represent internal and external entities and if those are present in-house or need to hire them. In this case, the colour scheme is based not only on hot and cold tones but also on the semantics represented by availability or unavailability. Both graphs are presented in 4.7 and 4.8.

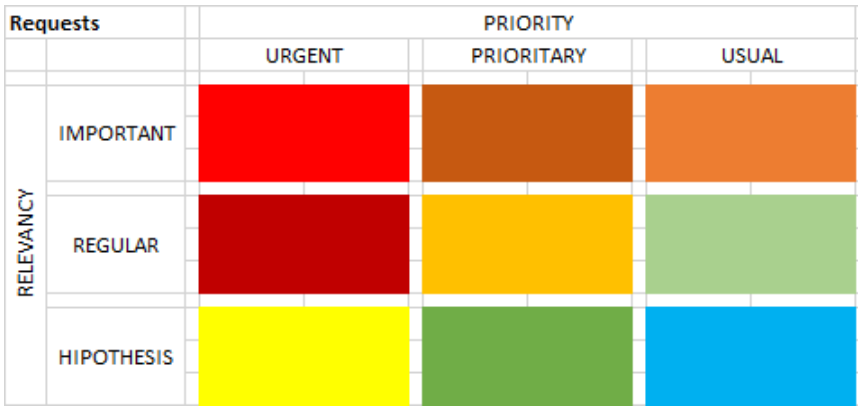

Figure 4.7: Priority and Relevancy Categorization - Own Proposal

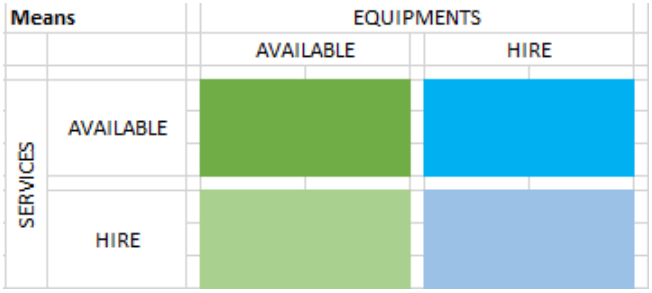

Figure 4.8: Means graph - Own Proposal

This review was considered essential because it was an exciting opportunity to further acknowledge the client on topics that are not usually debated. These topics are usability and colour science, which are considered topics that are not properly applied on most designed applications and are extremely important to improve efficiency and workflows.

4.4.2 Project Conflict Management

During the later stages of this work, a major problem occurred in the SIIP project development workflow. The project's development is staged on a version control system treated between two development branches due to different team responsibilities. These two branches covered very different project states and logic areas. While this work was being developed, the branch called Funcionalidade_Solicitacao was considered the main development branch, as that was the branch that involved the Request features, including the association and Disassociation.

One of the team members left at a point in April 2021, and a version control system error dictated the imminence of losing two months of work from the team member. That circumstance carried a full process of reorganising the project scope and content, with the detailed description of the occurrences seen in 4.9.

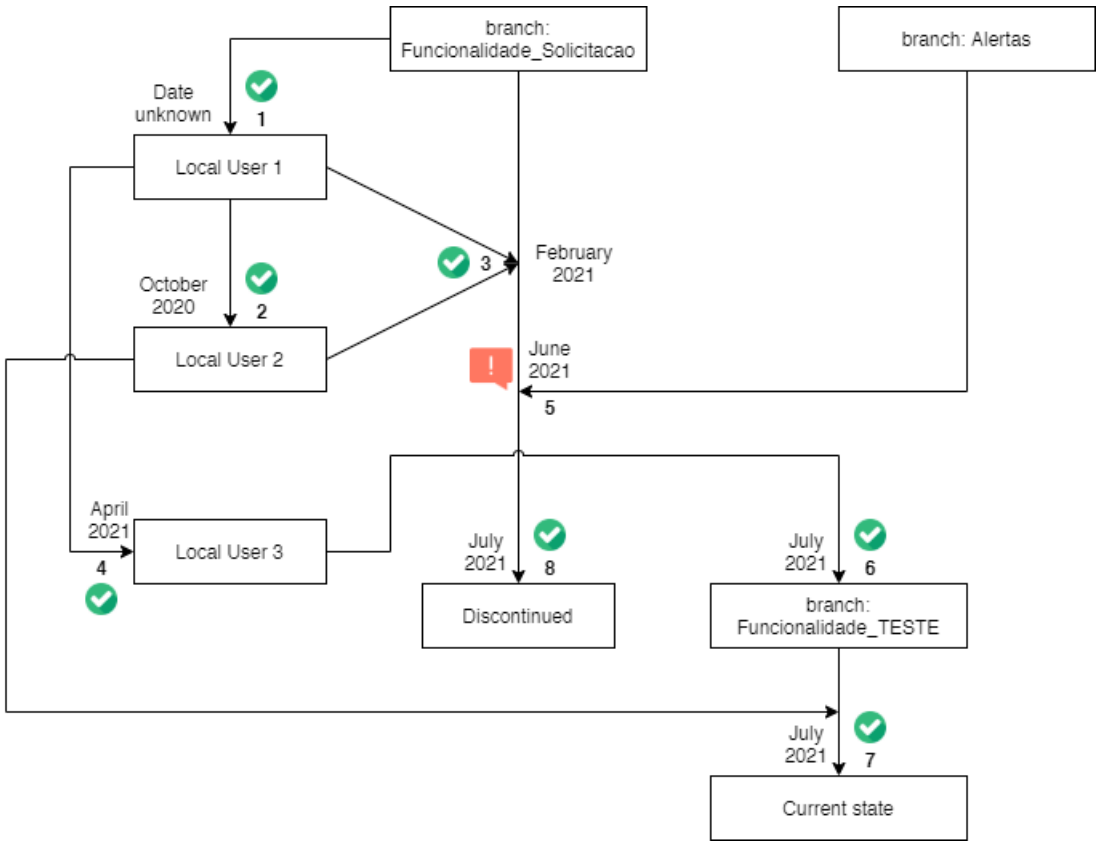

Figure 4.9: Project Conflict Management Resolution

The numeration on the image has the following descriptions:

1. Local User 1 gets a version of the Funcionalidade_Solicitacao branch on his Local Machine on an unknown date;

2. Local User 2 supplies a copy of his Virtual Machine with the same content that is on his project folder in the start date of this work, around October 2020;

3. Local Users 1 and 2 merge both contents to the branch and join all individually developed features. This action took place in early February 2021;

4. A new user (Local User 3) joins the team and gets the version from Local User 1, but with some compatibility errors with the branch, around April 2021. Local User 1 leaves the team behind two months of uncommitted work only present in Local User 3's computer. The errors described in step 4 were impeding a normal merge of content with Local User 2, and that is what forced a deeper measure;

5. To advance with the integration of Business Rules, Local User 2 tries to merge the Funcionalidade_Solicitacao and Alertas branches in June 2021 to develop the Business Rules component on top of the developed work. Unfortunately, severe compatibility issues abort that intent due to the Alertas branch being ten months old in terms of architecture;

6. After different operational approaches to recovering the unreferenced content in Local User 3's computer, Local User 2 creates a new branch called Funcionalidade_TESTE. It can now recover all the information that was at risk on Local User 3's computer. This action was taken in July 2021, and this branch now had all the project information until April 2021, which was considered essential work to keep. This was a significant step to the SIIP development because these errors caused the project to be suspended for a few weeks, and it opened a window of full operability;

7. With the hardest process completed, the main goal was to merge the most recently developed work from Local User 2 to the newly created branch, which happened after conflict resolutions were resolved. Considering this date (July 2021) marks the end of this work's contribution to the SIIP project, everything was now stable and ready for Local User 3 to be able to have a proper working environment;

8. With the new branch being fully operational, the Funcionalidade_Solicitacao was discontinued and set only to be used in cases of back-ups or urgent needs.

Lastly, it is important to mention that aborting the merge between the Funcionalidade_Solicitacao majorly impacted this work's development, as the Business Rules were supposed to be developed in that new merge. With that restriction and considering the lack of support from the client, the Business Rules integration changed to a Business Rules Integration Proposal, being developed outside the SIIP system to avoid more development issues and suspensions of work.

Chapter 5

Business Rules Integration Proposal

This chapter will describe the process of proposing Business Rules into the SIIP project, representing the work's title. As mentioned in the analysis chapter, taking this work approach was heavily influenced by the existing project's status and ways of dealing with a specific client. All the decisions from a business and logical standpoint will be listed here, along with the development workflow and proof.

This proposal aims to find an actual use case for Business Rules within the SIIP application. This proposal considers all the deciding factors considered in earlier chapters related to traits such as lack of technical knowledge or a more traditional working environment. This proposal is considered an important step to provide a knowledge base to less modern companies on improving workflows and modernising the systemic approaches and how these changes might impact work efficiency.

5.1 Business Rules Proposal - Architecture and Workflows

With the conditions presented in section 1.4, it is appropriate to define the structure that will base this Implementation Proposal. As this proposal plans to simulate a Business Rules integration within the SIIP application, some architectural changes have to be made to the core structure of how a typical Business Rules system would function, as seen in figure 2.6.

The general purpose of the Business Rules implementation on the SIIP application would be for the administration users (our clients) to create rules directly in the application. Considering the lack of technological knowledge of most SIIP users, there was a need to create an intermediary step between the rule creation and implementation. This means that a SIIP user might create a new rule without completing its implementation, which needs to be created later by the same person or someone capable of completing that task.

The opposition to 2.6 in this proposal intends to adapt the shown figure to something

usable within the SIIP application's needs and structure. The SIIP adaptation figure for a Business Rules platform would look like the figure below with the architecture mentioned above.

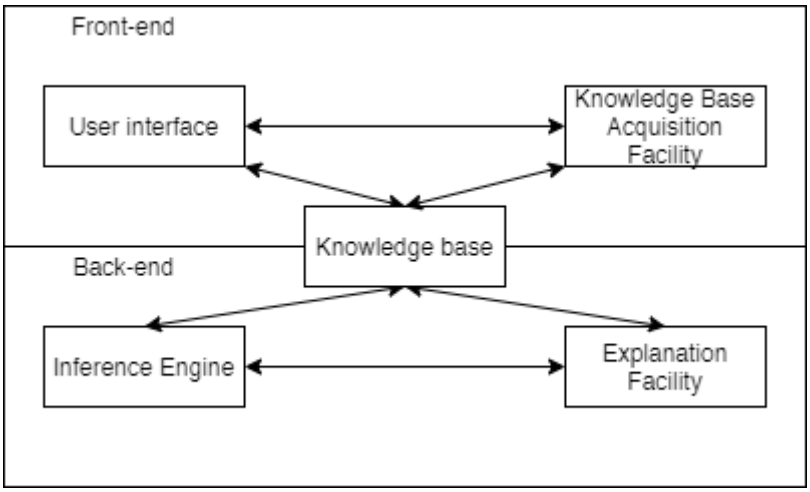

Figure 5.1: SIIP proposal of a Business Rules structure

5.1.1 Front-end components

With the given introduction in mind, the creation of the Business Rules components has to be inferred within the application's front end component. This incorporation features the acquisition of a knowledge base as well its incorporation with the user interface.

The design process behind this structure is mainly composed of integrating the user activity and the knowledge creation in a single screen. The main use case of creating such a structure is listed below.

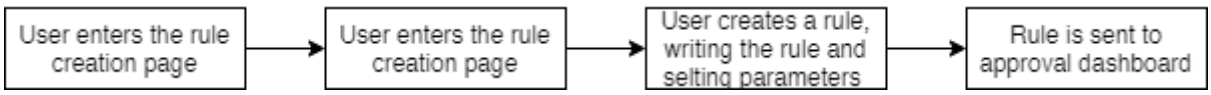

Figure 5.2: Use case for the front-end integration of creating a Business Rule

After the process demonstrated in the use case is completed, there is communication with the backend to store the requests. After that, the Business Rule enters a phase where it needs implementation based on the submitted request, which can be made by the same person that made the request or also by a more knowledgeable person. This action is concluded in a screen that lists all Business Rule requests to be implemented. The workflow demonstration is shown below.

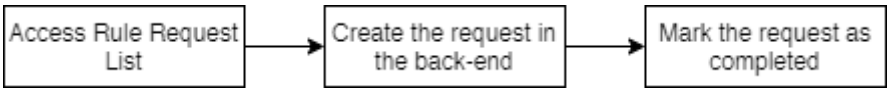

Figure 5.3: Use case for the front-end integration of dealing with Business Rule requests

This information makes it possible to connect the front-end components with the back-end, storing the requests and developing the Business Rules. The result of the development of the rules, which will be demonstrated later, is a set of working rules that need to be created, as seen in 5.2. That page can be found in the figure below.

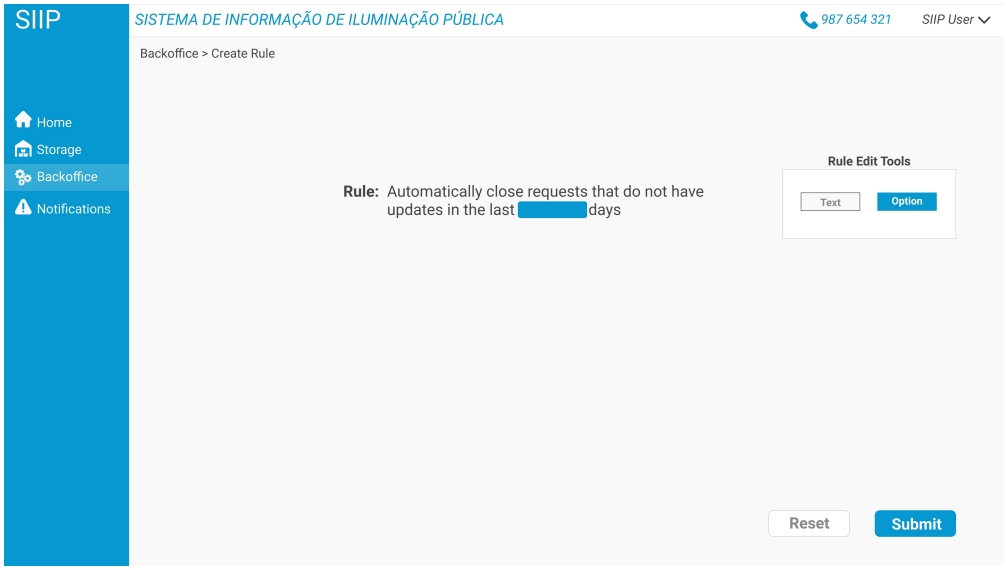

Figure 5.4: Business Rule creation screen

This page was designed to provide a low-code representation of a Business Rule for a less technologically skilled employer to provide inputs on rules he thinks the system needs. The user only needs to drag text and parameter elements to the main screen to construct the finalised rule. The parameter element is designed to provide a general solution to a property that can change its value. For example, in the figure example, that variable can be changed according to the number of specified days. With this being done, the rule request gets listed on the page containing all rule requests that need approval, listed in the figure below.

Figure 5.5: Business Rules ”to be approved” list

After using this screen, the goal is for a knowledgeable individual to pick these requests and turn them into a Business Rule. This action takes place on another page, letting the user take the rule and fill it with the desired variable parameter values before submitting the data. That event will trigger an action that will create the rule in the database, explained in a further segment. The visual representation of this acceptance page can be found in the following figure.

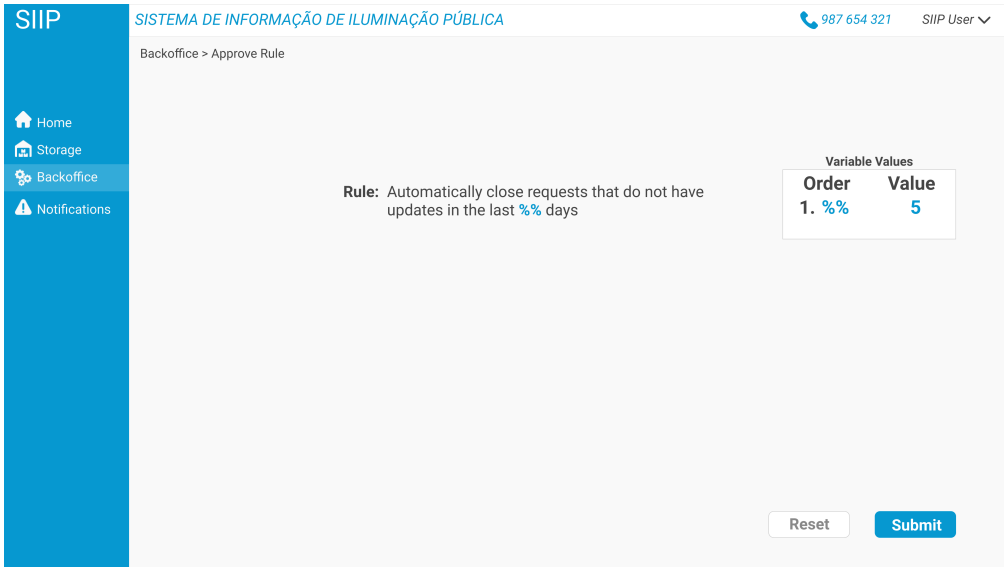

Figure 5.6: Business Rule request approval

Considering that a Business Rule would result from this action, there is also a panel where all the Business Rules that exist in the system are also designed to allow a less knowledgeable user to interact with the rules by toggling its activation or changing the

variable parameters. This approach is inspired by a low-code solution to allow less experienced users to interact with the system and propagate the changes to the backend server. This low-code proposal is demonstrated in the figure below.

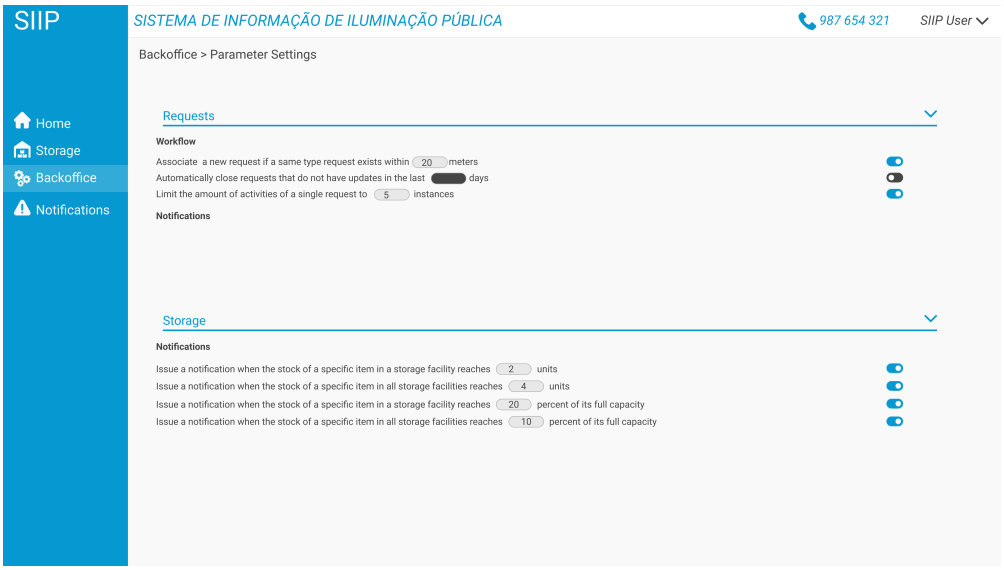

Figure 5.7: Business Rules control panel

5.1.2 Back-end components

As mentioned in the subsection 2.3.8, the general approach to incorporating Business Rules with the SIIP application would need to be a compatible, flexible and scalable alternative. This concludes that the Business Rules system must be integrated with the SIIP project, with the C# programming language.

Before representing this approach, it is essential to establish a connection with the front-end component. As mentioned in the front-end subsection, the knowledge base of the created rules must be somewhere where the front-end and backend connect. The connection between both components sends information stored in the backend and gathers information to be displayed in the front-end. The information written by the technologically savvy team member should be translated into business rules, which will be explored and demonstrated next.

The Business Rules implementation should be based on documentation articles provided in subsection 2.3.8. Those documents should serve as a guideline for adapting the Business Rules requests to the SIIP context. All the implementation processes will serve those guidelines to demonstrate the use case for this approach. Following a similar procedure to the Development chapter, the Business Rule files are separated according to each component.

5.1.3 Back-end components - Database Integration

Considering the adaptation made to fit Business Rules within the SIIP application, the rule information needs to be stored in the database to replicate the knowledge base. This database construction has to consider that a Business Rule can have variable values, which requires a dynamic database structure to recognise and effectively store the rule and its values.

It was decided to include a specific text code in the rule string passed to the database to support these requirements. This decision intends to identify a variable value with that text-code being the **%%** combination of characters. Considering that a rule can have multiple variable values, such as a rule that identifies intervals to issue a trigger, a one-to-many relation must be established, along with an association table that stores all the values from a rule. For better efficiency, the association table has a rule id reference and a parameter order attribute to differ different parameters from each rule. This factor is fundamental to better retrieving the parameters correctly, easing connecting parameters to the rule. The graphical understanding of these descriptions can be found in the below figure.

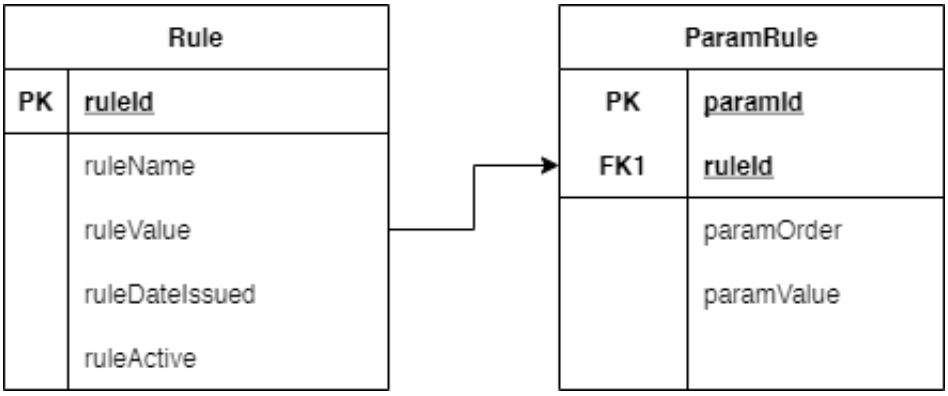

Figure 5.8: SIIP Business Rules' database design model

5.1.4 Back-end components - Use case simulation

To simulate this integration, it is better to explain all the workflow steps that need to be implemented in order to complete the process. The general workflow of this implementation can the found in the following figure.

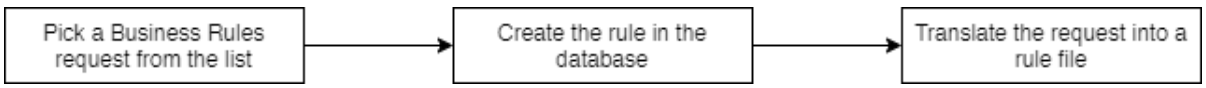

Figure 5.9: SIIP Business Rules workflow simulation

The rule database creation involves creating SQL commands within a C# file to store information about each rule. That creation is directly linked to the screen seen in the figure

5.6 that serves the purpose of generating the information needed to create the rule in the system. The information is passed from the front-end to the backend through an API call, which would convert the front-end values into a text-format request that reaches the API endpoint. By receiving the data at the API endpoint, the same data is then transferred to the database logic, which creates the programming representation and fills the structured projected in the figure 5.8. A portion of code would cover the database filling operation, which is presented below.

Listing 5.1: C# file of inserting a Business Rule in the Rule and ParamRule tables

```
using APIEndPoint.setRule
using(SqlConnection connection = new SqlConnection(_connectionString
    )){
    MyJsonDictionary<String, String> rule = new MyJsonDictionary<
        String, String>();
    rule = APIEndPoint.setRule(); //Dictionary object with JSON
        properties
    String query = "INSERT INTO dbo.Rule (ruleName,ruleValue,
        ruleDateIssued) VALUES (@ruleName,@ruleValue,@ruleDateIssued
        )";
    using(SqlCommand command = new SqlCommand(query, connection)){
        command.Parameters.AddWithValue("@ruleName", rule['name']);
        command.Parameters.AddWithValue("@ruleValue", rule['value'])
            ;
        command.Parameters.AddWithValue("@ruleDateIssued", DateTime.
            Now);
        connection.Open();
        int result = command.ExecuteNonQuery();
        int id = result.getId();
        //Get %% values from Dictionary
        int i =1;
        foreach (String param in rule['params']){
                String query2 = "INSERT INTO dbo.ParamRule (ruleId,
                    paramOrder,paramValue) VALUES (@ruleId,
                    @paramOrder,@paramValue)";
                using(SqlCommand command2 = new SqlCommand(query2,
                    connection))
                {
            command2.Parameters.AddWithValue("@ruleId", id);
            command2.Parameters.AddWithValue("@paramOrder", i);
            command2.Parameters.AddWithValue("@paramValue", param);
            connection.Open();
            int result2 = command2.ExecuteNonQuery();
            // Check Error
            if(result2 < 0)
                Console.WriteLine("Error inserting data into
                    Database!");}
        i++;}
        // Check Error
        if(result < 0)
            Console.WriteLine("Error inserting data into Database!")
                ;}
}
```

With the SQL code being ready, the rule information has conditions to be inserted into the system. After this, a user must program the Business Rule by using the information present in the database, especially the variable values. This component's code is documented in the articles mentioned above and should be structured in modules, which will now be presented.

As mentioned in the section 2.3.8, the chosen Business Rules application is called NRules. This platform requires direct installation within the development environment, but it is a relatively non-invasive method to incorporate Business Rules in a system. Installing the NRules component offers a set of features that completes the Business Rules management workflows. From rule creation to rule execution and triggering, NRules acts according to the needs of this implementation proposal.

Creating the rules implies the creation of the Data Models. In this case, they were already created, as seen in A.1. Once the object creations are done, the rule creation starts to take place. As the variable values were previously defined and sent to the database, those values will be gathered at some point. For demonstration purposes, the code related to the rule creation will represent the rule that specifies the association between same-type requests distanced within a specific interval. The rule gathers the needed parameter from the database and matches all Requests in the given distance condition while sequentially creating the association.

Listing 5.2: Rule creation example of associating requests that are close to each other

```
1 public class AssociateRequestsWithinDistance : Rule
2 {
3     List<string> params = API.GetRuleParams("distanceRule")
4     allowedDistance = params[0]
5     public override void Define()
6     {
7         Request request = default;
8
9         When()
10            .Match<Request>(() => request)
11            .Exists<Request>(r => r == request, r => r.
                  absolutePosition - request.absolutePosition <
                  allowedDistance);
12
13        Then()
14            .Do(_ => request.AssociateRequest(r));
15    }
16 }
```

This very effective implementation allows for simple generalisations that would not be possible with these types of Business Rules integration. The front-end component was constructed for a broader case demonstration to allow multiple variables because NRules can also cover more complex cases that deal with conjunction or disjunction operations. Some of those examples may represent how NRules handles interval verification, for example.

Considering a newly requested rule of issuing a notification when the stock of an item was higher than two and lower than six, the rule covering that case should follow the steps shown below. This easy data gathering is possible due to the efficient schema created in the database, which was designed to consider the order and identify each rule.

Listing 5.3: Rule creation of a stock notification between intervals

```
public class NotifyStockInterval : Rule
{
    List<string> params = API.GetRuleParams("stockMinMax-case2")
    minStock = params[0]
    maxStock = params[1]
    itemId = params[2]
    public override void Define()
    {
        StorageUnit su = default;
        Notification notification = default;

        When()
            .Match<StorageUnit>(() => su)
            .Exists<StorageUnit>(s => s == su, s => s.itemId.stock >
                minStock, s => s.itemId.stock < maxStock);

        Then()
            .Do(_ => notification.NotifyStockAtInterval(itemId));
    }
}
```

Finally, with both rules constructed, they only need to be integrated with a mechanism that manages them. As NRules is an inference engine, it means that the rules are not in a predefined order. That order is set automatically through a match-resolve-act cycle that checks rule dependencies and facts and determines which rules can fire. This cycle is repeated until there are no more rules to fire. In order to achieve this, the NRules component compiles the rules into a very well known structure called the Rete network, which allows the inference engine to know what rules can match facts. The rules are then compiled into a session factory, and then a working session is created, allowing the match-resolve-act cycle to be started.

The rule starter file loads the domain models and associates the session with the models. Right after, the session is started. This procedure can be seen below.

Listing 5.4: Rule engine start class

```
//Load rules
var repository = new RuleRepository();
repository.Load(x => x.From(typeof(PreferredCustomerDiscountRule).
    Assembly));

//Compile rules
var factory = repository.Compile();

//Create a working session
var session = factory.CreateSession();
```

```
10
11 //Load domain model
12 List<Request> requests = API.getRequests();
13 List<Storage> sotrages = API.getStorages();
14
15 foreach (Request req in requests){
16     session.Insert(req}
17 }
18
19 foreach (Storage stor in storages){
20     session.Insert(stor)
21 }
22
23 //Start match/resolve/act cycle
24 session.Fire();
```

The demonstrated code implements business rules use cases within a C# based system. This implementation was beneficial by allowing less tech-savvy users to interact with their application's workflow.

Chapter 6

Conclusion and future work

6.1 Conclusion

The work described in this document was developed to cover the Project in Information Systems required to complete the Master's in Computer Engineering at Faculdade de Ciências da Universidade de Lisboa. This work was produced during an internship at GMV Portugal.

The main goal of this work was to develop a Business Rules integration proposal that would demonstrate how helpful this tool can be in a specific set of systems, where there is space for efficiency in terms of workflows, especially in companies that are less technologically savvy. This integration required the development of the SIIP application, which is an application that is intended to serve a Public Illumination service that deals with requests across the streets of Lisboa if any replacement or technical intervention is needed.

The first phase of this project required a deep analysis of the SIIP application and its requirements to decide further how to handle the work theme. This phase allowed for a better understanding of what the best theme for the work should be.

After that, a development phase was implemented, where the features regarding request association and disassociation were completed. That task lasted until the end of the internship. The task allowed for a better understanding of the SIIP application and a better knowledge of the developed components.

There was a need to complement this work more practically during this phase, so a study was conducted. This study aimed to compare both public and private companies in terms of technological knowledge and improvement difficulties. This study concluded that there is still a gap between both sectors, which further amplifies the need to approach new business approaches for these companies, with Business Rules being a solid alternative to shift the paradigm for these companies.

The Business Rules integration was supposed to be developed on top of the request components and others. However, the client's unfortunate lack of communication did not

allow for better execution of this work. With that in mind, it was then decided that the better way to apply Business Rules within this system and still make it relevant would be to make a business proposal, covering the whole process of preparing the website and constructing the Business Rules. Although less practical, this approach allows for a broader view of the issue and how it can help companies with less technological knowledge achieve better workflow efficiency. This context also allowed for more creativity in prototyping and redesigning the SIIP system in these business areas, which was a very positive aspect.

6.2 Future work

For future work, the complete context analysis of the SIIP system should serve as an example of how to handle this specific type of client. The lack of communication led to a state where decisions had to be made, and the client was not there to communicate its will. The whole scope of this work was executed with this principle in mind, where there was a significant effort to avoid harming the process of creating this document.

Besides this, and as mentioned above, the Business Rules proposal aims to raise awareness of the urgent need for systems to improve technical complexity, considering the current digital context. This proposal was a way to contribute to this issue by providing the construction of simple processes that intend to create a foundation that allows companies to produce effective technological decisions.

The Business Rules proposal intended to cover all the construction phases and was made generalised to inspire any business areas to apply these concepts. As future work, companies are free to use these guidelines to include the developed practices within their systems, to ultimately create a better product, not just for the clients but for the workers' workflow and efficiency.

Appendix A

Request Association

A.1 Code Development

Listing A.1: API/Models/SolicitacaoAssociacaoViewModel.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace GMV.SIIP.API.Models
{
    public class SolicitacaoAssociacaoViewModel
    {
        public long Id { get; set; }

        public DateTime AssociadoEm { get; set; }

        public string AssociadoPor { get; set; }

        public long SolicitacaoId { get; set; }

        public long AlvoId { get; set; }

        public bool IsDeleted { get; set; }

    }
}
```

Listing A.2: API/Controllers/SolicitacaoAssociacaoController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using GMV.SIIP.API.Models;
using GMV.SIIP.DOMAIN.Models;
using System.Web.Http;

namespace GMV.SIIP.API.Controllers
```

```
{
    [RoutePrefix("api/SolicitacaoAssociacao")]
    public class SolicitacaoAssociacaoController : BaseController<
        SolicitacaoAssociacao,SolicitacaoAssociacaoViewModel>
    {

    }
}
```

Listing A.3: Domain/Models/SolicitacaoAssociacao.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace GMV.SIIP.DOMAIN.Models
{
    public class SolicitacaoAssociacao
    {
        [Key]
        [DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public long Id { get; set; }

        public DateTime AssociadoEm{ get; set; }

        public string AssociadoPor { get; set; }

        [ForeignKey("Solicitacao")]

        public long SolicitacaoId { get; set; }
        public virtual Solicitacao Solicitacao { get; set; }

        [ForeignKey("SolicitacaoAlvo")]
        public long AlvoId { get; set; }
        public virtual Solicitacao SolicitacaoAlvo { get; set; }

        public bool IsDeleted { get; set; }
    }
}
```

Listing A.4: UMB/Models/SolicitacaoAssociacaoCreateModel.cs

```
using GMV.SIIP.UMB.Models.Base;
using GMV.SIIP.UTILS;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Web.Mvc;
using Umbraco.Core.Models.PublishedContent;
using Umbraco.Web.Models;
```

```
11
12 namespace GMV.SIIP.UMB.Models
13 {
14     public class SolicitacaoAssociacaoCreateModel : BaseCreateModel,
           IContentModel
15     {
16         public IPublishedContent Content { get; set; }
17
18         public SolicitacaoAssociacaoCreateModel(IPublishedContent
               content)
19         {
20             ContentModel contentModel = new ContentModel(content);
21             Content = contentModel.Content;
22         }
23         public SolicitacaoAssociacaoCreateModel()
24         {
25
26         }
27
28
29         public long Id { get; set; }
30         public DateTime AssociadoEm { get; set; }
31         public string AssociadoPor { get; set; }
32         public long SolicitacaoId { get; set; }
33         public long AlvoId { get; set; }
34         public bool IsDeleted { get; set; }
35     }
36
37 }
```

Listing A.5: UMB/Models/SolicitacaoAssociacaoSummaryModel.cs

```
1 using GMV.SIIP.UTILS;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using System.Web.Mvc;
8
9 namespace GMV.SIIP.UMB.Models
10 {
11     public class SolicitacaoAssociacaoSummaryModel :
           BaseSummaryModel
12     {
13         [TableAttribute("Id", 1)]
14         public long Id { get; set; }
15         [TableAttribute("AssociadoEm", 2)]
16         public DateTime AssociadoEm { get; set; }
17
18         [TableAttribute("AssociadoPor", 3)]
19         public string AssociadoPor { get; set; }
20
21         [TableAttribute("Id Solicitacao", 4)]
22         public long SolicitacaoId { get; set; }
23
```

```
24          [TableAttribute("Id Alvo", 5)]
25          public long AlvoId { get; set; }
26
27          [TableAttribute("Tipo Alvo", 6)]
28          public string TipoAlvo { get; set; }
29
30          [TableAttribute("Estado", 7)]
31          public bool IsDeleted { get; set; }
32
33      }
34  }
```

Listing A.6: UMB/Models/SolicitacaoAssociacaoListModel.cs

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace GMV.SIIP.UMB.Models
8  {
9
10     public class SolicitacaoAssociacaoListModel : BaseListModel
11     {
12         public virtual List<SolicitacaoAssociacaoSummaryModel>
               CurrentRecords { get; set; }
13
14         public override IEnumerable<BaseSummaryModel> GetRecords()
15         {
16             return CurrentRecords ?? new List<
                   SolicitacaoAssociacaoSummaryModel>();
17         }
18     }
19 }
```

Listing A.7: UMB/Models/SolicitacaoAssociacaoSearchModel.cs

```
1  using GMV.SIIP.UTILS;
2  using Newtonsoft.Json;
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Web;
7  using System.Web.Mvc;
8  using Umbraco.Core.Models.PublishedContent;
9  using Umbraco.Web.Models;
10
11 namespace GMV.SIIP.UMB.Models
12 {
13     public class SearchAssociacaoModel : BaseModel,IContentModel
14     {
15         public IPublishedContent Content { get; set; }
16
17         public SearchAssociacaoModel(IPublishedContent content)
18         {
```

```
19              ContentModel contentModel = new ContentModel(content);
20              Content = contentModel.Content;
21
22          }
23          public SearchAssociacaoModel()
24          {
25
26          }
27
28          [JsonIgnore]
29          [FieldFrontEndOptions(Size = FieldSize.medium, Label = "Tipo
                ", Order = 1, InputName = "TipoPesquisaId", OnChange = "
                __siipSolicitacaoAssociacao.tipoPesquisa.change(this)")]
30          public List<SelectListItem> DDLTipoPesquisa { get; set; }
31
32          //[JsonIgnore]
33          [FieldFrontEndOptions(Ignore = true)]
34          public string TipoAssociacaoPesquisaId { get; set; }
35
36          public long AssociacaoId { get; set; }
37          public SearchBasicGenericModel Basic { get; set; }
38          public SearchAdvancedBaseModel Advanced { get; set; }
39          public SavedSearchModel SavedSearch { get; set; }
40      }
41  }
```

Listing A.8: UMB/Controllers/SolicitacaoAssociacaoController.cs

```
1  using GMV.SIIP.UMB.Models;
2  using GMV.SIIP.UMB.Repository.Interfaces;
3  using GMV.SIIP.UMB.Utils;
4  using GMV.SIIP.UTILS;
5  using LightInject;
6  using Newtonsoft.Json;
7  using Newtonsoft.Json.Linq;
8  using System;
9  using System.Collections.Generic;
10 using System.Linq;
11 using System.Net;
12 using System.Threading.Tasks;
13 using System.Web;
14 using System.Web.Mvc;
15 using Umbraco.Web.Models;
16
17 namespace GMV.SIIP.UMB.Controllers
18 {
19     public class SolicitacaoAssociacaoController :
           BaseRenderMvcController
20     {
21         [Inject]
22         public IApiRepository ApiRepository { get; set; }
23
24         public new ActionResult Index(ContentModel model)
25         {
26
27             var amodel = new SearchAssociacaoModel()
```

```
28              {
29                  Basic = new SearchBasicGenericModel(),
30                  Advanced = new SearchAdvancedGenericModel(),
31                  SavedSearch = new SavedSearchModel { TableExists =
                        false }
32              };
33              amodel.Setup();
34              amodel.Content = model.Content;
35
36              return PartialView("Associacao/SolicitacaoAssociacao",
                    amodel);
37          }
38
39
40
41          [Authorize]
42          public ActionResult SolicitacaoAssociacaoCreate(ContentModel
                umbracoModel)
43          {
44              SolicitacaoAssociacaoCreateModel solModel = new
                    SolicitacaoAssociacaoCreateModel(umbracoModel.
                    Content);
45              solModel.Setup();
46              return CurrentTemplate(solModel);
47          }
48
49          [HttpPost]
50          [Authorize]
51          public async Task<ActionResult> SolicitacaoAssociacaoCreate(
                SolicitacaoAssociacaoCreateModel solCreateModel)
52          {
53
54
55              ApiRepository.SetEntity("SolicitacaoAssociacao");
56              var result = await ApiRepository.Add(solCreateModel,
                    async (Sol) =>
57              {
58                  Sol.AssociadoEm = DateTime.Now;
59                  Sol.AssociadoPor = "admin";
60                  Sol.Id = Sol.Id;
61                  Sol.AlvoId = req.targetId;
62                  Sol.IsDeleted = false;
63
64              });
65
66              if (result != null)
67              {
68                  return Json(new { status = "success" });
69              }
70              else
71              {
72                  return Json(new { status = "failure" });
73              }
74          }
75
76      }
```

```
77 }
```

Listing A.9: UMB/Views/SolicitacaoAssociacao.cshtml

```
1 @inherits Umbraco.Web.Mvc.UmbracoViewPage<GMV.SIIP.UMB.Models.
      SearchAssociacaoModel>
2 @using ContentModels = Umbraco.Web.PublishedModels;
3 @using Umbraco.Core.PropertyEditors;
4 @using GMV.SIIP.UMB.Models;
5 @using System.Reflection;
6 @using GMV.SIIP.UTILS;
7 @using System.Web.Mvc.Html;
8
9 @{
10     Layout = "~/Views/master.cshtml";
11 }
12
13 @{
14     Model.DDLTipoPesquisa.RemoveAll(x => x.Text != "Solicita es")
          ;
15     var tipoPesquisaFEO = Model.GetType().GetProperty("
          DDLTipoPesquisa").GetCustomAttribute(typeof(
          FieldFrontEndOptionsAttribute)) as
          FieldFrontEndOptionsAttribute;
16 }
17 <script type="text/javascript">
18     \$(document).ready(function () {
19         \$.get("@Url.Content("~/Scripts/siip/solicitacaoassociacao.
              js")", function (result) {
20             \$('<script>' + result + '</' + 'script>').appendTo(
                  document.body);
21         }).done(function () {
22             __siipSolicitacaoAssociacao.init('#searchform', {
                  submitOnEnter : true });
23         });
24     });
25 </script>
26
27 <form id="searchform" method="post">
28     <section>
29     <div class="search">
30         <h2>Pesquisa</h2>
31         <div class="search__container">
32             <div class="search__container__fields">
33                 <div class="@Enum.GetName(typeof(FieldSize),
                      tipoPesquisaFEO.Size)">
34                     <label>@tipoPesquisaFEO.Label</label>
35                     @Html.DropDownList(tipoPesquisaFEO.InputName,
                          Model.DDLTipoPesquisa, new { id = "
                          tipoPesquisa", onchange = tipoPesquisaFEO.
                          OnChange })
36                 </div>
37                 @Html.Partial("Search/basic", Model.Basic)
38             </div>
39         </div>
40         <div class="search__container__btn">
```

```
41              <div>
42                  <a href="#" id="search_btn" class="search" onclick="
                        __siipSolicitacaoAssociacao.submit(event, 0, 10)
                        ">Pesquisar</a>
43              </div>
44              <div>
45                  <a href="#" class="clear" onclick="
                        __siipSolicitacaoAssociacao.clear()">Limpar
                        pesquisa</a>
46              </div>
47          </div>
48      </div>
49          </section>
50      <section>
51          <div class="search">
52              <h2>Pesquisa Avan ada</h2>
53              <span class="collapse-block"></span>
54              <div class="search__container">
55                  <div id="advancedsearch">
56                      @Html.Partial("Search/advanced", Model.Advanced)
57                  </div>
58              </div>
59          </div>
60          </section>
61          <section>
62              <div id="savedResults_collapsed" class="results">
63                  <h2>Pesquisas Efetuadas</h2>
64                  <span class="collapse-block"></span>
65                  <div class="results__container" id="savedSearch"
                        style="display:none">
66                      <div class="results__container__table" style="
                            padding: 0px; border-width: 1px">
67                          @Html.Partial("Search/savedSearch", Model.
                                SavedSearch)
68                      </div>
69                      <div class="results__container" id="
                            resultadosError" style="display:none;text-
                            align:center"></div>
70                  </div>
71              </div>
72          </section>
73          <section>
74              <div id="resultados_collapsed" class="results">
75                  <h2>Resultados</h2>
76                  <span class="collapse-block"></span>
77                  <div id="spinnerPesquisa" class="spinner-border"
                        style="display:none"></div>
78                  <div class="results__container" id="resultados"
                        style="display:none">
79                      @*if (Model.RecordsModel != null) { Html.Partial
                            ("table", Model.RecordsModel); }*@
80                  </div>
81                  <div class="results__container" id="resultadosError"
                         style="display:none;text-align:center">
82                  </div>
83              </div>
```

```
84          </section>
85  </form>
```

Listing A.10: UMB/Scripts/solicitacaoassociacao.js

```
1  var __siipSolicitacaoAssociacao = {};
2  __siipSolicitacaoAssociacao.options = {};
3
4  __siipSolicitacaoAssociacao.init = function (formid, options) {
5      debugger;
6      __siipSolicitacaoAssociacao.options.data = options || {};
7      __siipSolicitacaoAssociacao.form = {};
8      __siipSolicitacaoAssociacao.form.id = formid;
9
10     __siipSolicitacaoAssociacao.form.fields = {};
11
12     $(__siipSolicitacaoAssociacao.form.id).submit(
           __siipSolicitacaoAssociacao.submit);
13
14     __siipSolicitacaoAssociacao.options.apply();
15 }
16
17 __siipSolicitacaoAssociacao.tipoPesquisa = {};
18
19 __siipSolicitacaoAssociacao.options.apply = function () {
20     __siipSolicitacaoAssociacao.options.setSubmitOnEnter(
           __siipSolicitacaoAssociacao.options.data.submitOnEnter);
21 }
22 __siipSolicitacaoAssociacao.options.setSubmitOnEnter = function (
       enable) {
23     if (!!!enable) return;
24
25     $(__siipSolicitacaoAssociacao.form.id + " input, select").each(
           function () {
26         $(this).unbind('keypress');
27
28         $(this).keypress(function (e) {
29             if (e.which == 10 || e.which == 13) {
30                 e.preventDefault();
31                 $(e.target.form).submit();
32                 return false;
33             }
34         })
35     });
36 }
37
38 __siipSolicitacaoAssociacao.tipoPesquisa.change = function (evt,
       callback) {
39     debugger;
40     $("#advancedsearch").hide();
41     if (evt.selectedOptions[0].value === "") return;
42     $.ajax({
43         url: '/umbraco/surface/Search/GetAdvancedSearchForm?
               tipoPesquisa=' + evt.selectedOptions[0].text,
44         type: 'GET',
45         cache: true,
```

```
46          success: function (data) {
47              $("#advancedsearch").html(data);
48              $("#advancedsearch").show();
49
50              __siipSolicitacaoAssociacao.options.apply();
51
52              if (callback) callback();
53          }
54      })
55  }
56
57  __siipSolicitacaoAssociacao.submitWithAsyncReader = function (page,
        numRecords) {
58      $.ajax({
59          url: '/umbraco/Surface/Search/Search',
60          data: ""
61              + "tipoPesquisaId=" + ($("#tipoPesquisa").val() === "" ?
                    "" : $("#tipoPesquisa option:selected").text()) + "
                    &"
62              + "SubtipoPesquisaId=" + ($("[issearchsubtype=True]").
                    val() === "" ? "" : $("[issearchsubtype=True] option
                    :selected").text()) + "&"
63              + "page=" + page + "&numRecords=" + numRecords + "&"
64              + $("#searchform").serialize(),
65          type: 'POST',
66          cache: false
67      });
68  }
69
70  __siipSolicitacaoAssociacao.submit = function (e, page = null,
        numRecords = null) {
71      var checkTable = true;
72      if ($("#savedSearch table")[0] == undefined) checkTable = false;
73      e.preventDefault();
74      $.ajax({
75          url: '/umbraco/Surface/Search/SaveSearch',
76          data: 'checkTable=' + checkTable + '&deleteJsFunctionName=
                    __siipSolicitacaoAssociacao.DeleteFromTable(this, event)
                    &' + $("#searchform").serialize(),
77          type: 'POST',
78          cache: false,
79          beforeSend: function () {
80              $('#resultadosError').hide();
81              $('#savedSearch').hide();
82              $("#spinnerPesquisa").show();
83          },
84          success: function (data) {
85              $('#savedSearch table tbody').append(data);
86                  $('#savedSearch').show();
87              $('#savedResults_collapsed').removeClass("results
                    collapsed").addClass("results");
88
89
90
91          },
92          error: function () {
```

```
93              $("#spinnerPesquisa").hide();
94              $('#savedSearch').hide();
95              $('#resultadosError').show();
96          }
97      });
98
99      $.ajax({
100         url: '/umbraco/Surface/Search/SearchSolicitacaoAssociacao',
101         data: ""
102             + "tipoPesquisaId=" + ($("#tipoPesquisa").val() === "" ?
                    "" : $("#tipoPesquisa option:selected").text()) + "
                    &"
103             + "SubtipoPesquisaId=" + ($("[issearchsubtype=True]").
                    val() === "" ? "" : $("[issearchsubtype=True] option
                    :selected").text()) + "&"
104             + "page=" + page + "&numRecords=" + numRecords + "&"
105             + $("#searchform").serialize(),
106         type: 'POST',
107         cache: false,
108         beforeSend: function () {
109             $('#resultadosError').hide();
110             $('#resultados').hide();
111             //$("#spinnerPesquisa").show();
112         },
113         success: function (data) {
114             setTimeout(function () {
115                 $("#spinnerPesquisa").hide();
116                 $('#resultados').html(data);
117                 $('#resultados').show();
118
119                 $('#resultados_collapsed').removeClass("results
                        collapsed").addClass("results");
120
121                 addButtons();
122                 fillTable();
123             }, 500);
124         },
125         error: function (error) {
126             $("#spinnerPesquisa").hide();
127             $('#resultados').hide();
128             $('#resultadosError').show();
129         }
130     });
131 }
132
133 __siipSolicitacaoAssociacao.clear = function () {
134
135 }
136
137 __siipSolicitacaoAssociacao.delete = function (entity, id) {
138     $.ajax({
139         url: '/Delete',
140         data: "entity=" + entity + "&id=" + id,
141         dataType: 'json',
142         type: 'POST',
143         cache: false
```

```
144     }).done(function (result) {
145         if (result.status === "success") {
146             $.ajax({
147                 url: '/message?entity='+entity+'&messageType=Delete
                        ',
148                 type: 'GET',
149                 success: function (data) {
150                     $("#appModal").html(data);
151                     $("#appModal")[0].style.display = "block";
152                 }
153             })
154         }
155         else {
156             // show the error message to user
157         }
158     });
159 }
160
161 __siipSolicitacaoAssociacao.detail = function (entity, id) {
162     if(entity == "Solicitacao")
163         window.location.href = "/solicitacao?id=" + id;
164     else
165         window.location.href = "/detail?entity=" + entity +"&id=" +
            id;
166 }
167
168 __siipSolicitacaoAssociacao.DeleteFromComponentesGeridos = function
      (evt, event) {
169     event.preventDefault();
170     evt.closest("tr").remove();
171 }
172
173 __siipSolicitacaoAssociacao.ReplicateComponenteGerido = function (
      evt, event) {
174     event.preventDefault();
175     var rowToReplicate = evt.closest("tr");
176     $(rowToReplicate).clone().insertAfter(rowToReplicate);
177 }
178
179 __siipSolicitacaoAssociacao.DeleteFromTable = function (evt, event)
      {
180     event.preventDefault();
181     evt.closest("tr").remove();
182 }
183
184 __siipSolicitacaoAssociacao.ClearAll = function (evt, event) {
185     $('#addC').find('tr').remove();
186     $('#resultados .results__container__table table tbody tr td').
          find('input:checked').each(function () {
187         $(this).prop("checked", false);
188     });
189 }
190
191
192 __siipSolicitacaoAssociacao.ColapseAssoc = function (evt, event) {
193     console.log("aauaua");
```

```
194     console.log($('#addC').is(":hidden"));
195     if ($('#assocs').is(":hidden")) {
196         $('#assocs').show();
197     } else {
198         $('#assocs').hide();
199     }
200
201 }
202
203
204 function addButtons() {
205     $('#resultados .results__container__table table thead tr').
            prepend('<th class="resultsAssocCheckbox">Associar</th>');
206     $('#resultados .results__container__table table tbody tr').each(
            function () {
207         $(this).prepend('<td class="resultsAssocCheckbox" style="
                text-align: center; vertical-align: middle;">\
208             <input type="checkbox" name="query_myTextEditBox"></td>
                ');
209     });
210 }
211
212 function fillTable() {
213     $('#searchform').append('<section id="assocsTab"><section>')
214
215     $('#assocsTab').html('<div id="resultados_collapsed" class="
            results">\
216                     <h2> Associar</h2>\
217                 <span class="collapse-block" onclick="
                        __siipSolicitacaoAssociacao.ColapseAssoc()"></
                        span>\
218                 <div id="spinnerPesquisa" class="spinner-border"
                        style="display: none;"></div>\
219                 <div class="results__container" id="assocs" style=""
                        >\
220                     \
221                     <div class="results__container__table" style="
                            padding: 0px; border-width: 1px">\
222                         <table>\
223                             <thead>\
224                                 <tr>\
225                                     <th>Parente</th>\
226                                     <th>Numero SIIP</th>\
227                                     <th>Numero GOPI</th>\
228                                     <th>Nome</th>\
229                                     <th>Morada</th>\
230                                     <th>Tipologia</th>\
231                                     <th>Fase</th>\
232                                     <th>Estado</th>\
233                                     <th>Data</th>\
234                                     <th>Responsavel</th>\
235                                     <th>Freguesias</th>\
236                                     <th>Remover</th>\
237                                 </tr>\
238                             </thead>\
239                            \<tbody id="addC"></tbody>\
```

```
240                         </table>\
241         \
242                     </div >\
243         <div class="search__container__btn">\
244             <div>\
245                 <a href="#" id="search_btn" class="search doAssoc">
                        Associar</a>\
246             </div>\
247         <div>\
248         <a href = "#" class= "clear" onclick = "
                __siipSolicitacaoAssociacao.ClearAll()" > Limpar tudo</a
                >\
249             </div>\
250         </div>\
251                     \
252                     \
253                     \
254                     </div>\
255                 <div class="results__container" id="resultadosError"
                        style="display:none;text-align:center">\
256                 </div>\
257             </div >');
258
259
260
261     $('#resultados .results__container__table table tbody tr td
            input').change(function () {
262             $('#addC').find('tr').remove();
263             $('#resultados .results__container__table table tbody tr
                    td').find('input:checked').each(function () {
264
265                 var copyTr = $(this).closest('tr').clone()
266                 copyTr.find('.resultsAssocCheckbox').remove();
267                 var idSiip = copyTr.find('td:eq(0)').text()
268                 $(copyTr).attr("id", "sol" + idSiip.trim());
269
270                 if ($('#addC').find('tr').length % 2 == 0) {
271                     $(copyTr).prepend('<td>\
272                         <div class="radio">\
273                         <label><input type="radio" id="regular" name
                                ="optradio"></label>\
274                      </div>\
275                  </td >');
276                         $(copyTr).append('<td id="removeAss" style="text
                            -align:center;position:relative;background-
                            color:#f9f9f9"><img style="height:1rem; "src
                            ="/Content/siip/img/redCross.svg" alt=""></
                            td>')
277                 } else {
278                     $(copyTr).prepend('<td>\
279                         <div class="radio">\
280                         <label><input type="radio" id="regular" name
                                ="optradio"></label>\
281                      </div>\
282                  </td >');
283                         $(copyTr).append('<td id="removeAss" style="text
```

```
                        -align:center;position:relative;background-
                        color:white"><img style="height:1rem; "src="
                        /Content/siip/img/redCross.svg" alt=""></td>
                        ')
284                 }
285
286                  $('#addC').append($(copyTr));
287                 });
288         });
289
290     $(function () {
291
292         $('#assocsTab').on('click', '#removeAss', function () {
293
294             var id = $(this).closest('tr').attr('id').replace("sol",
                    "");
295             $(this).closest('tr').remove();
296             $('#resultados .results__container__table table tbody tr
                    ').each(function () {
297
298                 console.log($(this).find('td:eq(1)').text().trim());
299                 console.log(id);
300                 if ($(this).find('td:eq(1)').text().trim() == id) {
301                     $(this).closest('tr').find('input').prop("
                            checked", false);
302                 }
303             });
304         })
305
306     });
307
308
309     $(function () {
310
311         $('#assocsTab').on('click', '.doAssoc', function () {
312             if($('#addC').find('tr').length > 0){
313                 var target = $('#addC').find('tr td input:checked').
                        closest('tr');
314                 targetSolNr = target.find('td:eq(1)').text().trim();
315
316                 var createAssoc = {};
317                 createAssoc["target"] = targetSolNr;
318
319
320
321                 var i = 1;
322                 $('#addC').find('tr').each(function () {
323                     nrSol = $(this).find('td:eq(1)').text().trim();
324                     if (nrSol != targetSolNr) {
325                         createAssoc[String(i)] = nrSol;
326                         i++;
327                     }
328
329
330                     __siipAsyncRender.render.doRender({
331                         url: '/umbraco/surface/SolicitacaoAssociacao
```

```
                                /SolicitacaoAssociacaoCreate',
332                         httpMethod: 'POST',
333                         data: createAssoc,
334                         automaticallyShowTarget: false,
335                         callback: __siipAsyncRender.render.doRender
                                ({
336                             url: '/message?entity=
                                    SolicitacaoAssociacao&messageType=
                                    Create',
337                             target: 'appModal'
338                         })
339                     });
340
341                 });
342
343             }else {
344                 console.log("Empty")
345             }
346
347         })
348
349     });
350
351 }
```

A.2 Umbraco Linkage

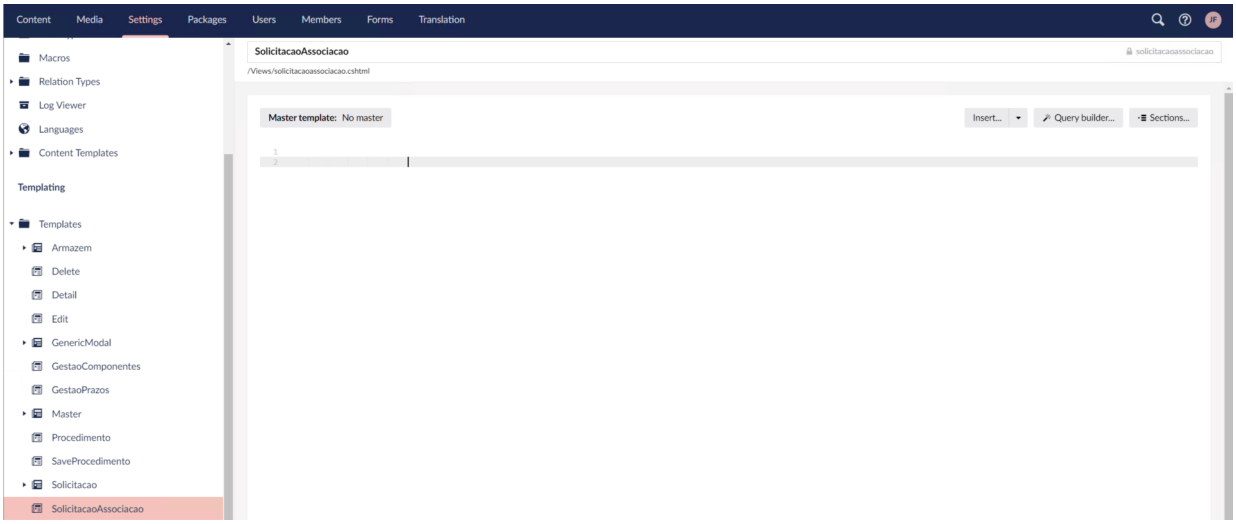

Figure A.1: Request Association - Umbraco/Template creation

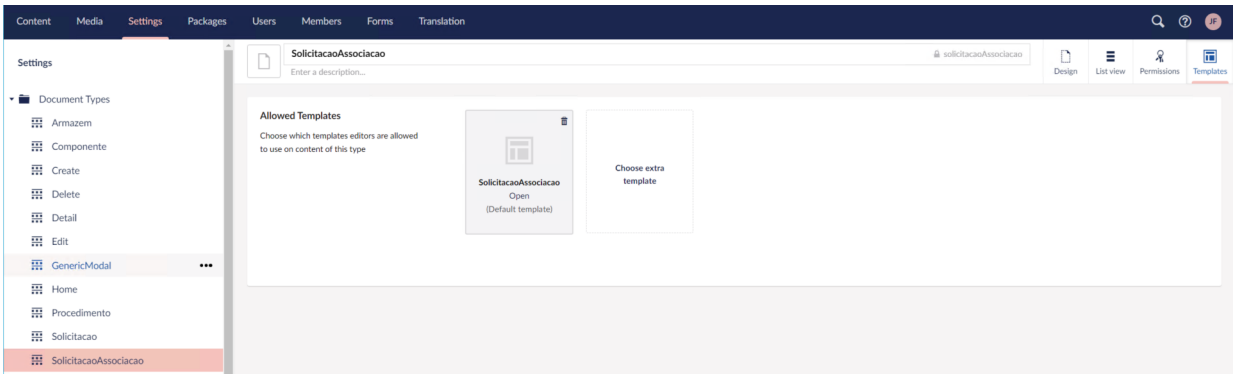

Figure A.2: Request Association - Umbraco/Document Type creation

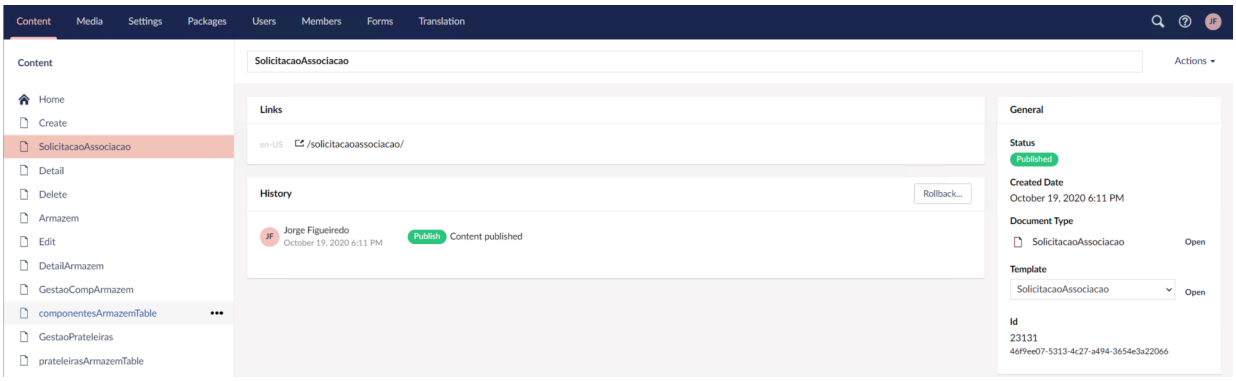

Figure A.3: Request Association - Umbraco/Item creation

Appendix B

Request Disassociation

B.1 Code Development

Listing B.1: UMB/Views/Subdivisao.cshtml

```
1 @inherits UmbracoViewPage<GMV.SIIP.UMB.Models.SolicitacaoCreateModel
      >
2 @using ContentModels = Umbraco.Web.PublishedModels;
3 @using Umbraco.Core.PropertyEditors;
4 @using GMV.SIIP.UTILS;
5 @using GMV.SIIP.UMB.Models;
6 @using System.Reflection;
7
8 @{
9     Layout = "~/Views/master.cshtml";
10 }
11
12 <script type="text/javascript">
13     $(document).ready(function () {
14         debugger;
15     $.get("@Url.Content("~/Scripts/siip/subdivisao.js")", function (
          result) {
16         $('<script>' + result + '</' + 'script>').appendTo(document.
              body);
17     }).done(function () {
18         __siipSubdivisao.init("@Model.Id", "#procedimentosForm", "#
              atividadeForm", '@Model.TipoSolicitacaoEstadoId');
19     });
20 });
21 </script>
22
23
24 <div class="modal" style="display:none">
25   <div class="gallery">
26     <div class="title">
27       <h2>Titulo</h2>
28     </div>
29     <div class="gallery__container">
30       <div class="form">
31         <form action="">
32           <div class="fields">
```

```
33              <div>
34                <label for="">Nome</label>
35                <input type="text">
36              </div>
37              <div>
38                <label for="">Adicionar nova imagem</label>
39                <span id="file">Select File</span>
40                <span id="valueFile">Nenhuma Imagem seleccionada</span
                      >
41                <input type="file">
42              </div>
43            </div>
44            <div class="btn">
45              <a href="#" type="submit">Adicionar</a>
46              <a href="#">Cancelar</a>
47            </div>
48          </form>
49        </div>
50      </div>
51    </div>
52    <div class="zoom">
53      <div class="title">
54        <h2>Titulo</h2>
55      </div>
56      <div class="zoom__container">
57        <div class="item">
58          <img src="https://picsum.photos/200/300" alt="">
59        </div>
60        <div class="item">
61          <img src="https://picsum.photos/200/300" alt="">
62        </div>
63        <div class="item">
64          <img src="https://picsum.photos/200/300" alt="">
65        </div>
66      </div>
67      <div class="zoom__container__btn">
68        <div>
69          <a href="" class="create">Predefinir esta imagem</a>
70        </div>
71        <div>
72          <a href="" class="plus">Adicionar nova imagem</a>
73        </div>
74      </div>
75    </div>
76  </div>
77  <section>
78    <h1>@Model.TipoSolicitacaoEstadoText - @Model.Responsavel</h1>
79    <div class="search collapsed">
80      <h2>Informa  o Geral</h2>
81      <span class="collapse-block"></span>
82      <div class="search__container">
83        <div class="search__container__fields">
84          <div class="small">
85            <label for="">N mero SIIP</label>
86            @Html.TextBox("NumeroSIIP", Model.NumeroSIIP ?? "", new {
                  disabled = true })
```

```
87          </div>
88          <div class="small">
89            <label for="">N mero GOPI</label>
90            @Html.TextBox("NumeroGOPI", Model.NumeroGOPI ?? "", new {
                  disabled = true })
91          </div>
92          <div class="small">
93            <label for="">Data</label>
94            @Html.TextBox("Data", Model.Data, "{0:yyyy-MM-dd}", new {
                  Type = "date", disabled = "true" })
95            <label class="calendar" for="dateFrom">
96              <img src="~/Content/siip/img/calendar.svg" alt="">
97            </label>
98          </div>
99          <div class="medium">
100           <label for="">Tipologia</label>
101           @Html.DropDownList("TipoSolicitacaoId", Model.
                  DDLTipoSolicitacao, new { disabled = "true" })
102         </div>
103       </div>
104     </div>
105     <div class="search__container__fields">
106       <div class="large">
107         <label for="">Observa  es</label>
108         <textarea name="" id="" cols="30" rows="5" disabled>@Model.
                Descricao</textarea>
109       </div>
110     </div>
111   </div>
112 </section>
113 <section>
114   @Html.Partial("SolicitacaoRequerente", Model.SolicitacaoRequerente
        )
115 </section>
116 <section>
117   <div class="search collapsed">
118     <h2>Informa  o do Local</h2>
119     <span class="collapse-block"></span>
120     <div class="search__container">
121       <div class="search__container__fields">
122         <div class="large">
123           <label for="">Local</label>
124           @Html.TextBox("Morada", Model.Morada, new { disabled =
                  true })
125         </div>
126       </div>
127       <div class="search__container__fields">
128         <div class="medium">
129           <label for="">Freguesia</label>
130           @Html.TextBox("Freguesias", Model.
                  FreguesiaTextsCommaSeparated, new { disabled = true })
131         </div>
132         <div class="medium">
133           <label for="">Refer ncia</label>
134           @Html.TextBox("Referencia", Model.Referencia, new {
                  disabled = true })
```

```
135         </div>
136       </div>
137     </div>
138   </div>
139 </section>
140 <section>
141   <div class="results collapsed">
142     <h2>Associa  es</h2>
143     <span class="collapse-block"></span>
144     <div class="results__container">
145       <div class="results__container__tabs">
146         <div class="results__container__tabs__item">
147           <a href="" class="active">Unidades de IP</a>
148         </div>
149         <div class="results__container__tabs__item">
150           <a href="">Componentes</a>
151         </div>
152         <div class="results__container__tabs__item">
153           <a href="">Solicita  es</a>
154         </div>
155       </div>
156       <div class="results__container__table">
157         <table>
158           <thead>
159             <tr>
160               <th width="100px">Ac   es</th>
161               <th>Origem</th>
162               <th>N mero SIIP</th>
163               <th width="400px">Morada</th>
164               <th>Freguesias</th>
165               <th>Tipologia</th>
166               <th>Fase</th>
167               <th>Estado</th>
168               <th>Data</th>
169             </tr>
170           </thead>
171           <tbody>
172             <tr>
173               <td>
174                 <div>
175                   <a href="">
176                     <img src="/Content/siip/img/eye.svg" alt="">
177                   </a>
178                 </div>
179               </td>
180               <td>
181                 <p>?</p>
182               </td>
183               <td>
184                 <p class="blue">1002</p>
185               </td>
186               <td>
187                 <p>Rua Teste  n   4</p>
188               </td>
189               <td>
190                 <p>Lisboa</p>
```

```
191                 </td>
192                 <td>
193                   <p>Candeeiro apagado</p>
194                 </td>
195                 <td>
196                   <p>Conclus o</p>
197                 </td>
198                 <td>
199                   <div class="state warranty">
200                     <span></span>
201                     <p>GARANTIA</p>
202                   </div>
203                 </td>
204                 <td>
205                   <p>01/01/2019</p>
206                 </td>
207               </tr>
208               <tr>
209                 <td>
210                   <div>
211                     <a href="">
212                       <img src="/Content/siip/img/eye.svg" alt="">
213                     </a>
214                   </div>
215                 </td>
216                 <td>
217                   <p>?</p>
218                 </td>
219                 <td>
220                   <p class="blue">1002</p>
221                 </td>
222                 <td>
223                   <p>Rua Teste  n   4</p>
224                 </td>
225                 <td>
226                   <p>Lisboa</p>
227                 </td>
228                 <td>
229                   <p>Candeeiro apagado</p>
230                 </td>
231                 <td>
232                   <p>Conclus o</p>
233                 </td>
234                 <td>
235                   <div class="state analyze">
236                     <span></span>
237                     <p>AN LISE</p>
238                   </div>
239                 </td>
240                 <td>
241                   <p>01/01/2019</p>
242                 </td>
243               </tr>
244               <tr>
245                 <td>
246                   <div>
```

```
247                     <a href="">
248                       <img src="/Content/siip/img/eye.svg" alt="">
249                     </a>
250                   </div>
251                 </td>
252                 <td>
253                   <p>?</p>
254                 </td>
255                 <td>
256                   <p class="blue">1002</p>
257                 </td>
258                 <td>
259                   <p>Rua Teste  n   4</p>
260                 </td>
261                 <td>
262                   <p>Lisboa</p>
263                 </td>
264                 <td>
265                   <p>Candeeiro apagado</p>
266                 </td>
267                 <td>
268                   <p>Conclus o</p>
269                 </td>
270                 <td>
271                   <div class="state suspended">
272                     <span></span>
273                     <p>SUSPENSO</p>
274                   </div>
275                 </td>
276                 <td>
277                   <p>01/01/2019</p>
278                 </td>
279               </tr>
280             </tbody>
281           </table>
282         </div>
283       </div>
284     </div>
285 </section>
```

Listing B.2: UMB/Controllers/SolicitacaoController.cs

```
1         public async Task<ActionResult> Subdivisao(string id,
              ContentModel umbracoModel)
2         {
3             if (!string.IsNullOrEmpty(id) && long.TryParse(id, out
                  long solId))
4             {
5                 var repo = ApiRepository.SetEntity("Solicitacao");
6                 SolicitacaoCreateModel solModel = await repo.GetById
                      <SolicitacaoCreateModel>(solId);
7                 solModel.Content = umbracoModel.Content;
8                 solModel.Setup();
9                 //solModel.SolicitacaoRequerente.Setup();
10                return PartialView("Subdivisao", solModel);
11            }
```

```
12              else return Redirect("/");
13          }
```

Listing B.3: UMB/Scripts/Subdivisao.js

```
1 __siipSubdivisao.deleteAssoc = function (e, formId) {
2     __siipAsyncRender.render.doRender({
3         url: '/SolicitacaoAssociacao/DeleteAssocs/' +
              __siipSolicitacao.SolicitacaoId,
4         httpMethod: 'POST',
5         data: $("#AssocForms:checked").attr('id'),
6         sourceEvent: e,
7         automaticallyShowTarget: false,
8         callback: __siipAsyncRender.render.doRender({
9             url: '/message?entity=&messageType=SolicitacaoAssociacao
                  ',
10            target: 'appModal'
11        })
12    })
13 }
```

Listing B.4: UMB/Controllers/SolicitacaoAssociacaoController.cs

```
1
2         public async Task<ActionResult> DeleteAssocs(long id,
              SolicitacaoAssociacaoCreateModel solCreateModel)
3         {
4
5             ApiRepository.SetEntity("SolicitacaoAssociacao");
6             var result = await ApiRepository.Delete(id);
7             if (result != null)
8             {
9                 return Json(new { status = "success" });
10            }
11            else
12            {
13                return Json(new { status = "failure" });
14            }
15        }
16
17    }
18
19 }
```

B.2 Umbraco Linkage

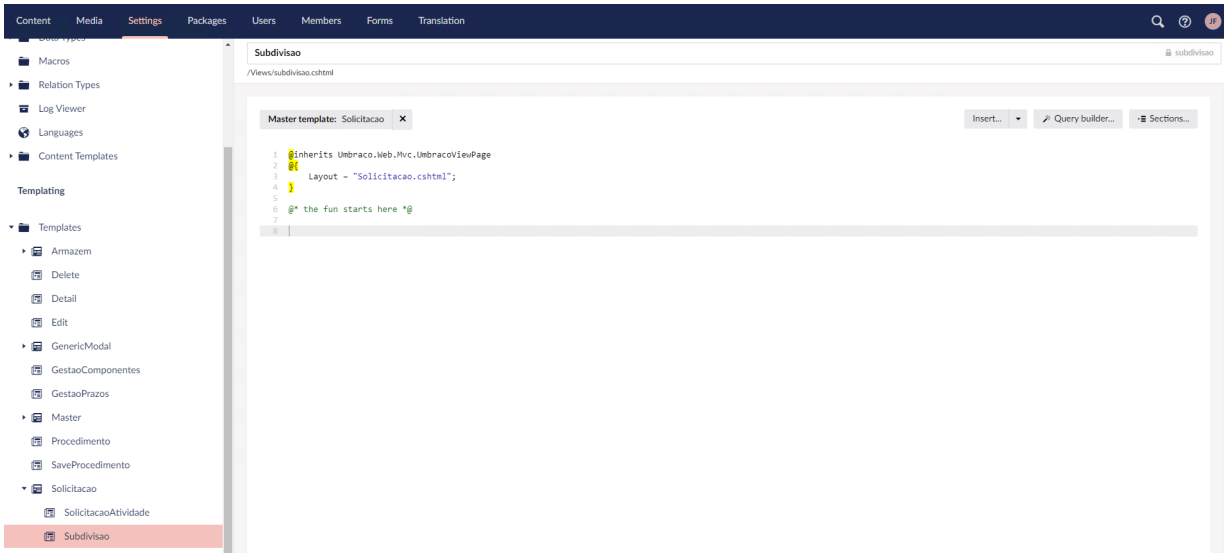

Figure B.1: Request Disassociation - Umbraco/Template Creation

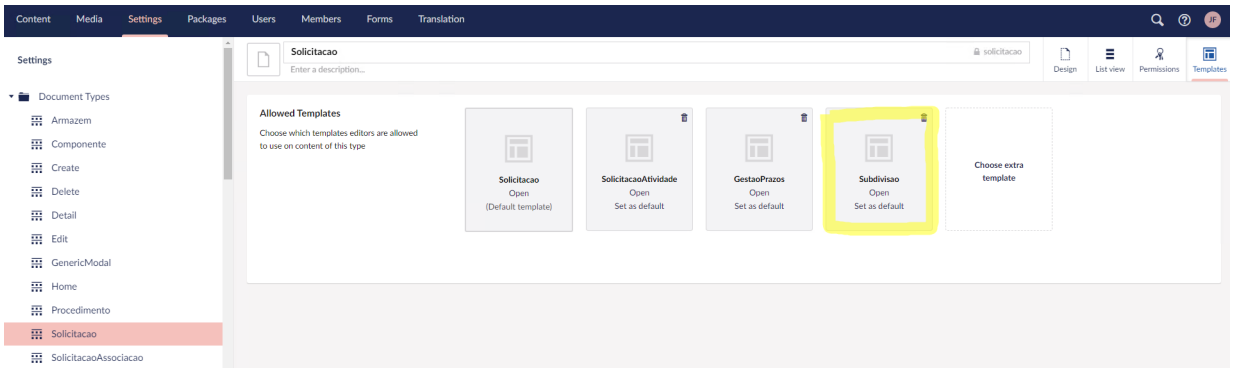

Figure B.2: Request Disassociation - Umbraco/Template Association

Bibliography

[1] Yuri Vanessa Nieto Acevedo, Jose Fernando Lopez Quintero, Carlos Enrique Montenegro Marin, and Claudio Camilo Gonzalez Clavijo. Business rules model for the automation in the receipt of credit applications by financial institutions based on archimate. 2016.
https://ieeexplore.ieee.org/abstract/document/7555258.

[2] S. Adepu, G. Mishra, and A. Mathur. Access control in water distribution networks: A case study. 2017.
https://ieeexplore.ieee.org/abstract/document/8009922.

[3] Pia Andrews. Systemic challenges for digital public sector reform. 2019.
https://www.themandarin.com.au/119660-systemic-challenges-for-digital-public-sector-reform/.

[4] Michal Bali. *BDrools JBoss Rules 5.0 Developer's Guide*. 2009.
http://www.dmi.unipg.it/bista/didattica/CSP/drools/books/Drools%20Jboss%20Rules%205.0%20Developer's%20Guide.pdf.

[5] Abdelali Boussadi, Cedric Bousquet, Pierre Durieux, and Patrice Degoulet et al. A business rules design framework for a pharmaceutical validation and alert system. 2010.
https://www.researchgate.net/publication/47510926_A_Business_Rules_Design_Framework_for_a_Pharmaceutical_Validation_and_Alert_System.

[6] Lindblad CI, Hanlon JT, Gross CR, Sloane RJ, Pieper CF, Hajjar ER, Ruby CM, and Schmader KE. Clinically important drug-disease interactions and their prevalence in older adults. 2006.
https://pubmed.ncbi.nlm.nih.gov/16982290.

[7] CML. *Organização dos serviços municipais*. Lisbon, Portugal, 2015.
https://www.am-lisboa.pt/documentos/1429803531C5oII3rq2Eh85KA5.pdf.

[8] Flavio Corradini, Alberto Polzonetti, and Oliviero Riganelli. Business rules in e-government applications. 2009.

https://www.researchgate.net/publication/251804244_Business_Rules_in_e-Government_Applications.

[9] John Deacon. Model-view-controller (mvc) architecture. 2009. http://www.johndeacon.net/?page_id=263.

[10] Doxee. 5 challenges for digital transformation in public administration. 2019. https://www.doxee.com/blog/digital-marketing/digital-transformation-in-public-administration/.

[11] FDA. Fda strengthens warning that non-aspirin nonsteroidal antiinflammatory drugs (nsaids) can cause heart attacks or strokes. 2005. https://www.fda.gov/media/92768/download.

[12] David Ferraiolo and D. Richard Kuhn. Role-based access controls. 2009. https://www.researchgate.net/publication/24164143_Role-Based_Access_Controls.

[13] David F. Ferraiolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. 2003. https://books.google.pt/books?hl=enlr=id=48AeIhQLWckCoi=fndpg=PR15dq=role+based+access+controlots=LNQJBKq_Jasig=3fukxy7dIutpiMTUhFP2SbSBBHIredir_esc=yv=onepageqf=false.

[14] Frazee LA. Gallegos PJ. Anticholinergic therapy for lower urinary tract symptoms associated with benign prostatic hyperplasia. 2008. https://pubmed.ncbi.nlm.nih.gov/18294115/.

[15] Ian Graham. *Business Rules Management and Service Oriented Architecture*. 2006. https://books.google.pt/books?id=_InzFf5XpeQCprintsec=frontcoversource=gbs_book_other_versions_rredir_esc=yv=onepageqf=false.

[16] Rahul Gudla, Amit Madhukar Bhosle, and Vijay Chandrasekaran. System and method for designing effective business policies via business rules analysis. 2014. https://patentimages.storage.googleapis.com/58/de/5c/2ae24bb00531ca/US8731983.pdf.

[17] Frederick Hayes-Roth. Rule-based systems. 1985. https://dl.acm.org/doi/abs/10.1145/4284.4286.

[18] Gustavo Homem. Presentable and unpresentable: Devops and stuff. 2020. https://moodle.ciencias.ulisboa.pt/pluginfile.php/277084/mod_resource/content/1/AS-FCUL-CGS-2020.pdf.

[19] Roman Hypský and Jitka Kreslíková. Definition of business rules using business vocabulary and semantics. 2017. http://aip.vse.cz/pdfs/aip/2017/02/01.pdf.

[20] John Kouraklis. Object-oriented programming (oop). 2020.
https://link.springer.com/chapter/10.1007/978-1-4842-6112-5_5.

[21] Han Liu, Alexander Gegov, and Mihaela Cocea. Rule-based systems: a granular computing perspective. 2016.
https://link.springer.com/article/10.1007/s41066-016-0021-6.

[22] Agência Lusa. Siza vieira destaca importância de acelerar transformação digital da administração públicatra28071996lgjj**! 2021.
https://www.tsf.pt/portugal/economia/siza-vieira-destaca-importancia-de-acelerar-transformacao-digital-da-administracao-publica-13682185.html.

[23] Naser Masri, Yousef Abu Sultan, Alaa N. Akkila, Abdelbaset Almasri, Adel Ahmed, Ahmed Y. Mahmoud, Ihab Zaqout, and Samy S. Abu-Naser. Survey of rule-based systems. 2019.
https://philpapers.org/rec/MASSOR-2.

[24] Mark Massé. Rest api design rulebook. 2012.
https://books.google.pt/books?id=eABpzyTcJNIClpg=PR3ots=vAQA00fhOAdq=api%20restlrpg=PP4v=onepageq=api%20restf=false.

[25] Satish Menon. System and method for detecting the source of media content with application to business rules. 2009.
https://patentimages.storage.googleapis.com/77/70/ba/b7e401d6625c07/US10552701.pdf.

[26] Samy S. Abu Naser and Mohanad M. Hilles. An expert system for shoulder problems using clips. 2016.
http://dstore.alazhar.edu.ps/xmlui/bitstream/handle/123456789/385/27-05-2019-06.pdf?sequence=1isAllowed=y.

[27] Samy S. Abu Naser and Abu Zaiter A. Ola. An expert system for diagnosing eye diseases using clips. 2005-2008.
http://dstore.alazhar.edu.ps/xmlui/bitstream/handle/123456789/383/27-05-2019-04.pdf?sequence=1isAllowed=y.

[28] Joshua Ofoeda, Richard Boateng, and John Effah. Application programming interface (api) research: A review of the past to inform the future. 2019.
https://www.igi-global.com/article/application-programming-interface-api-research/232166.

[29] Patrik Paetau. On the benefits and problems of the object-oriented paradigm including and finnish study. 2005.
https://helda.helsinki.fi/bitstream/handle/10227/123/151-951-555-894-8.pdf.

[30] Code Project. Object oriented way of writing business rules. https://www.codeproject.com/Articles/1100564/Object-Oriented-Way-of-Writing-Business-Rules.

[31] Gary Riley. Clips: An expert system building tool. 1991. https://books.google.pt/books?hl=enlr=id=K3S9k0Ta1Z0Coi=fndpg=PA149 dq=CLIPS:+An+expert+system+building+toolots=fXrFUrJaBzsig=0YKfW4F4 bcp41WHQ5ZgDJhbkvHkredir_esc=yv=onepageqf=false.

[32] Koen Smit, Martijn Zoet, and Matthijs Berkhout. Verification capabilities for business rules management in the dutch governmental context. 2017. https://ieeexplore.ieee.org/abstract/document/8002499.

[33] NRules team. Wiki. https://github.com/NRules/NRules/wiki.

[34] Ram Naresh Thakur and Dr. U.S. Pandey. A study focused on web application development using mvc design pattern. 2019. https://1library.net/document/zx5le2oq-study-focused-web-application-development-using-design-pattern.html.

[35] Code On Time. Learning resources - business rules. https://codeontime.com/learn/business-rules.