Lisbon School of Economics & Management
Universidade de Lisboa

MASTER

DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK

INTERNSHIP REPORT

APPLICATION OF PROCESS MINING TECHNIQUES IN A BUSINESS CONTEXT

ANA CATARINA FERREIRA PARENTE

MARCH - 2022

Lisbon School of Economics & Management
Universidade de Lisboa

MASTER
DATA ANALYTICS FOR BUSINESS

MASTER'S FINAL WORK
INTERNSHIP REPORT

APPLICATION OF PROCESS MINING TECHNIQUES IN A BUSINESS CONTEXT

ANA CATARINA FERREIRA PARENTE

SUPERVISION:
PROF. CARLOS J. COSTA
DR. BEATRIZ BAGOIN GUIMARÃES

MARCH – 2022


ISEG
Lisbon School
of Economics
& Management
Universidade de Lisboa

ABSTRACT

As a result of the data explosion, most companies are addressing the need to deal with massive volumes of data about their processes in their information systems. Some of this data, which are commonly known as event logs, contain valuable information about the activities that happen during the execution of a process in a company. Process mining attempts to extract useful insights from these event logs. The present work aims to realize if it will be advantageous for a company to implement process mining techniques to manage document flows. Having this in mind, the intention is to understand the available technologies that allow the implementation of process mining algorithms and to compare the results obtained. Therefore, the main objective of this project is to identify the most suitable tool and algorithm that will allow us to achieve the best results, based on the needs of the company under consideration. For this purpose, we made the processes emerge through three different tools: PM4PY, ProM, and Power BI. Regarding the algorithms, we focused our analysis on Alpha Miner, Heuristic Miner, and Inductive Miner. As a result, it is expected that the process models resulting from each of these algorithms and each tool represent the complete workflow of decision and execution of each procedure so that their performance can be later evaluated according to a set of metrics. As such, the theoretical contribution of this work focuses on the study of the application of different process mining algorithms in a set of tools that have been scarcely analyzed in the literature, always aiming to meet business needs.

KEYWORDS: Process Mining; Process discovery; PM4PY; ProM; Power BI.

RESUMO

Como resultado do crescimento do fluxo de dados, surge a necessidade de as empresas conseguirem lidar eficazmente com estes grandes volumes de dados relativos aos seus processos que estão presentes nos seus sistemas de informação. Alguns desses dados, usualmente conhecidos como *event logs*, contêm importantes informações sobre as atividades que acontecem durante a execução de um processo numa empresa. As técnicas de *process mining* tentam extrair informações úteis destes *event logs*. O presente trabalho visa compreender se será benéfico para uma empresa implementar técnicas de *process mining* para gerir os fluxos documentais. Com isto em mente, o foco centra-se em perceber quais as tecnologias disponíveis que permitem a implementação de algoritmos de *process mining*, e comparar os resultados obtidos. Assim, o principal objetivo deste projeto é identificar a ferramenta mais adequada, assim como o algoritmo que nos permitirá alcançar os melhores resultados, com base nas necessidades da empresa em questão. De acordo com este domínio, emergimos os processos através de três ferramentas diferentes: PM4PY, ProM e Power BI. Relativamente aos algoritmos, focamos a nossa análise no Alpha Miner, Heuristic Miner e Inductive Miner. Como resultado, espera-se que os modelos de processos decorrentes de cada um dos algoritmos e de cada ferramenta representem o fluxo de trabalho completo de decisão e execução de cada procedimento para que o seu desempenho possa ser avaliado de acordo com um conjunto de métricas. Assim, a contribuição teórica deste trabalho centra-se no estudo dos diferentes algoritmos de *process mining* em ferramentas pouco analisadas na literatura, visando sempre corresponder às necessidades empresariais.

Lisbon School of Economics & Management
Universidade de Lisboa

TABLE OF CONTENTS

Lisbon School of Economics & Management
Universidade de Lisboa

LIST OF FIGURES

List of Tables

Acronyms and Abbreviations

BPMN – Business Process Model and Notation.

CRISP-DM – Cross Industry Standard Process for Data Mining.

CSV – Comma Separated Values.

DBMS – Database Management System.

PmBI – Process Mining by ProcessM.

PM4PY – Process Mining for Python.

XES – EXtensible Event Stream.

XLSX – Microsoft Excel Open XML Spreadsheet.

XML – Extensible Markup Language.

Lisbon School of Economics & Management
Universidade de Lisboa

ACKNOWLEDGMENTS

First of all, I would like to thank Dr. João Paulo Carvalho for receiving me at Quidgest, as without this, this work would not have been possible. Furthermore, I would also like to express my gratitude to Dr. Beatriz Guimarães for incorporating me into her department and for always being supportive.

I am also thankful to Professor Carlos J. Costa for all the support, suggestions, and advice shown throughout the process of carrying out the Master's Final Work.

I would like to thank all the collaborators and friends I had the opportunity to work with during this internship, for their help and encouragement, in particular: Ana Glória, Francisco Cavaco, João Milagaia, and Vitor Pinto.

I am also grateful for the support and patience that I received throughout this process from all my friends, with special appreciation for Leonor, and Márcia.

Finally, the biggest thanks of all goes to my family, especially to my parents and grandparents, for allowing me to reach this stage, and to my boyfriend, João, for all the love, support, and recommendations throughout these months. Without you, none of this would be possible!

1. INTRODUCTION

1.1. Theoretical Context

With the growth and development of an organization, it is expected that there will be an increase in the demand from its customers, as well as the potential problems associated with that business relationship. Allied to this, the implications for each organization's workflow arising from the increasing development of technologies must be considered, as they may or may not simplify the issues that could happen in terms of the organization's business process management. Process mining is an approach that should be considered as it can help organizations to face these challenges. Specifically, this area has somehow emerged to answer the remarkable growth of technology and the data explosion that has been emphasized in recent years (van der Aalst, 2010). Through the application of process mining techniques, it would be possible to have a full perception and a better understanding of each of the processes of a given organization and their interactions. This is achievable as long as each organization records some of the most valuable information related to its processes in their information systems, such as which activities were performed, by whom, and when. Nowadays, there is, in fact, a growing tendency for organizations to prefer to store their data in a more or less structured format in their systems, either as a result of the current legislation or to improve the organization's performance (van der Aalst et al., 2007).

According to van der Aalst et al. (2012), process mining procedures can be described as a means to extract important knowledge regarding the data available in the information systems of each company. As a result of implementing those techniques, the organization will gain valuable insights into the actual workflow of each process, since one of the main advantages of applying those methods is the possibility of obtaining the path of the processes that occur in the organization's day-to-day operations. With the analysis of these solutions, an organization would be able to find deviations from the idealized process path by comparing the results achieved from the application of process mining techniques with the desired results; as well as being able to locate the activities of a process with a lack of resources — bottlenecks —, as these are the ones that end up leading to deviations from the expected performance. As such, the key contribution of

this work is that these advantages can, in the end, optimize the company's performance in the execution of its activities.

1.2. Quidgest

Quidgest is a company that was created in Portugal, having been founded in 1988. Nowadays, the company is present in Portugal, Germany, East Timor, and Mozambique. It comprises a team of more than 100 employees, distributed across different business areas, such as the financial and the health areas, among others.

Quidgest focuses its activity in the area of consultancy and development of information systems, with special emphasis and investment in the area of Software Engineering. Its main product is the extreme low-code platform, which was developed by the company, called Genio. This platform is based on several programming languages, such as C#, C++, Java, and HTML 5. By using the Genio platform, its users are expected to be able to develop software more quickly and automatically, allowing them to obtain equally fast and efficient solutions, regardless of the user's level of knowledge of programming languages. Due to this, Genio is seen as an example of a user-friendly platform, as it allows professionals without any specific knowledge in the field of software engineering to develop any type of solution, including the most complex ones.

1.3. Objective

This work was developed within the scope of a project carried out by the Information Management and Business Processes department and by a Portuguese Public University. The department is in charge of understanding the underlying processes of each project, preparing the workflows of each one of them, and developing the respective systems. To this end, it includes in its activities a set of systems, such as the Integrated Information Management system; and the Data Protection Management system, through which they support the implementation of the General Data Protection Regulation. Therefore, the motivation for this project was focused on the integration of process mining technologies in this business environment.

In light of this context, the research question of this work is to understand to what extent process mining helps in document management. In this sense, this project aims to identify the most suitable tool and the algorithm that offer the best results, according to the company's needs.

To achieve this goal, it is first necessary to recognize the format in which the activities are being stored so that the development of the processes is achievable. Then, using the set of chosen tools, it will be possible to observe the processes from start to finish. As a result, a set of anomalous situations can be identified, through the analysis of a set of metrics, in order to improve performance.

1.4. Methodological Approach

As previously introduced, this project intends to integrate current technologies to create workflows capable of expressing processes, and then compare the information transmitted by these tools to determine which one is the most appropriate.

As a result, the methodological approach used in the development of this work was divided into distinct phases, each with its own set of objectives. Through the literature review, the first phase aimed to get an understanding of the field of process mining, specifically its most common techniques, perspectives, and algorithms. Following that, it was important to identify a set of process mining tools to develop process models, by studying the related work. Subsequently, it was also necessary to identify the set of metrics most commonly provided by the tools, which was also done through the analysis of the works related to this domain. Finally, the recommendation of the most appropriate tool for the organization's needs, as well as the algorithm with the best result, is based on the development of empirical work and the analysis of its results.

1.5. Structure of the Master's Final Work

The remainder of this work is structured as follows. The literature review is covered in Chapter 2, which includes an examination of the studied literature and its conclusions. The empirical work is developed in Chapter 3, and the results are analyzed in Chapter 4. In Chapter 5 we compare the results taken from our work with related studies. Finally, in Chapter 6, we summarize the conclusions that could be drawn from the analysis of the results and suggest some points for future research.

2. LITERATURE REVIEW

2.1. Introduction

In the sphere of the study of processes, two distinct areas intersect, data science and process management (van der Aalst, 2016). Through this, the knowledge of a less process-centric approach is combined with an approach that applies technology and management to operational processes. Process mining is thus recognized as a procedure that lowers the time required to study processes while also delivering more accurate results than standard process analysis methods.

As previously introduced, due to the impact of the growth of technologies, companies are increasingly showing the need to keep, in a structured way, huge amounts of data related to their processes in their information systems. However, apart from recording the set of activities that occur throughout the development of a given process — event logs — the real challenge is figuring out how to extract information from this data to enhance the company's processes' effectiveness. With this in mind, van der Aalst (2016) described the purpose of applying process mining techniques as a way to extract knowledge about event logs. Accordingly, the event logs are seen as essential for developing process mining models, being, therefore, described as the starting point for implementing such techniques (van der Aalst et al., 2012).

Event logs contain various information, with mandatory and optional information about the events. Concerning mandatory components, the following are included: the case, meaning, the instance to which the process is associated, and the activity, which corresponds to an event in the process. In the study from van der Aalst et al. (2012), the extra information that event logs may contain is described: an associated timestamp, that is, the time the activity started and eventually the time at which the activity ended, and a resource, which corresponds to the person who did the activity. The process models that result from this type of event data are then analyzed to identify and locate bottlenecks throughout the processes, and to identify models that can serve as a reference for similar future processes (van der Aalst, 2016).

2.2. Types of Process Mining Techniques

In the literature, three types of process mining techniques are distinguished: process discovery, conformance checking, and model enhancement.

Applying process discovery techniques makes it possible to convert an event log into a process model (van der Aalst, 2016). Figure 1 shows how this technique works: a process model is generated as an output from a set of logs. As a result, we can conclude that the main objective of using this method is to create a model that reflects all the possible paths of a given process. The model resulting from these techniques may provide different information according to the characteristics of the algorithm that is used.

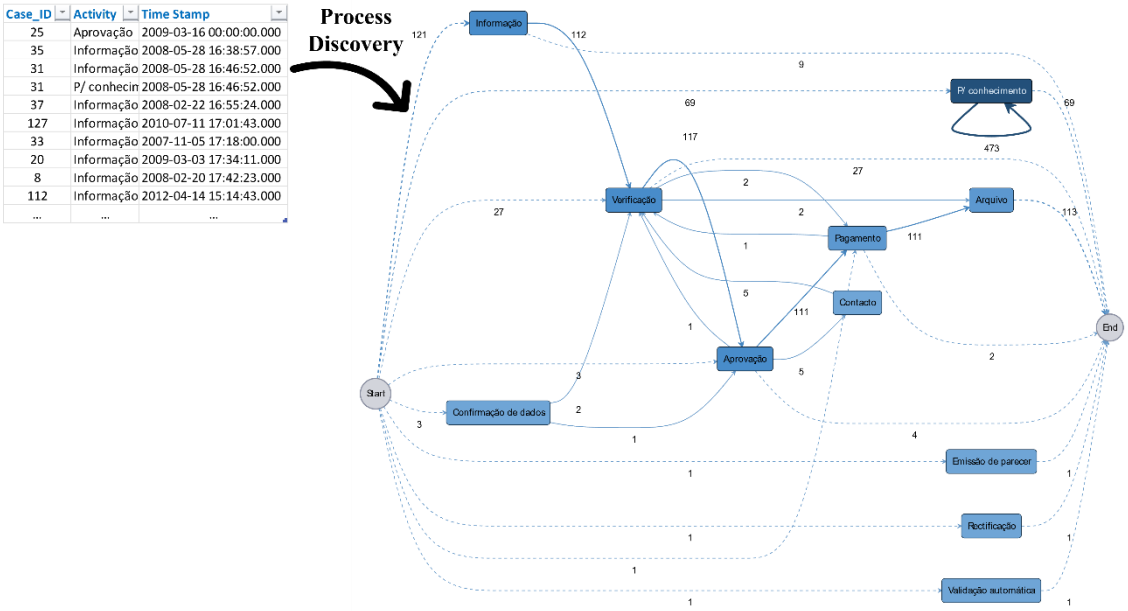

Figure 1 - Process Discovery Technique.

Although process discovery is one of the most well-known process mining approaches, there are more procedures to consider in addition to this one. In fact, process discovery is often seen as the starting point for carrying out deeper analyses (van der Aalst, 2012).

Conformance checking techniques are in charge of verifying whether the registered model corresponds to the reality, by comparing both models (van der Aalst, 2016). When considering this technique, the central question consists of: is the process model an adequate reflection of the logs? By answering this question, it is possible to understand whether or not there is a good match between the event logs and the existing model (Rozinat & van der Aalst, 2008). To determine the conformance between models, two measures are considered: fitness and appropriateness. Rozinat & van der Aalst (2008) duly address these concepts. According to the authors, fitness is the most crucial

constraint for measuring conformance. In this study, fitness is defined as a measure that tries to realize to what extent the process model fits the different paths identified by the logs (Rozinat & van der Aalst, 2008). Concerning the measurement of the fitness between a process model and the actual process, the authors suggested replaying the logs in the model to understand if there is any problem in carrying out the underlying path (Rozinat & van der Aalst, 2008). The values for this metric can range between 0 and 1, where 1 means that there is perfect fitness, and 0 means that none of the traces can be reproduced, i.e., there is no fitness (van der Aalst, 2016). As this metric can be too strict, appropriateness is also considered. This metric focuses on evaluating if the process model accurately represents the behavior of the real logs (Rozinat & van der Aalst, 2008). By applying these two measures, it is expected to achieve the practical goal of performing conformance checking techniques, that is, to detect and understand the deviations that may exist in the process that is under analysis (van der Aalst, 2010).

The last process mining technique, model enhancement, tries to change the original process model by applying the available information, with the ultimate goal of improving it. In the research from van der Aalst (2016), two types of model enhancement techniques are identified: repair and extension. The first approach aims to repair the model so that it describes more accurately what happens in reality. The extension technique allows the inherent process model to be expanded by providing new perspectives according to the information contained in the logs (van der Aalst, 2016).

2.3. Process Modeling Languages

A key element related to process mining is the language that models the systems of each business process. One of the best-known ways to represent these models is through graph-based models. A set of examples of these representation techniques are given in the literature, and the following will be highlighted: Petri nets, Business Process Model and Notation (BPMN), and Process Trees.

Petri nets have been around for a fairly long time, having been created by Carl Adam Petri in 1939 and officially proposed in 1962 (Murata, 1989). Petri nets are represented by stable and direct graphs, including a start and an end state. Each one of these networks includes two types of nodes: places and transitions (Murata, 1989). As illustrated in Figure 2, the places are represented by the circles whereas the squares represent the

transitions. A place can be seen as an input if there is an arc connected from the place to the transition, or it can be considered as an output otherwise (van der Aalst, 1998). A transition represents an action or an activity that is being modeled and is responsible for the movement of tokens from the input to the output place that it is connected to. For a transition to fire, it must have at least one token. When enabled, it produces an additional token that will be the input of the next place (van der Aalst, 1998).

Petri nets are useful for depicting two frequent circumstances in daily life: parallelism and choices. In parallelism situations, we try to represent activities that happen in parallel. This is done by connecting a transition with several output places, in the case of a parallel split. If it is a parallel join, the different places are connected to a single transition, and that transition can only fire if it receives tokens from all places. On the other hand, in the case of representing an exclusive decision situation, a place appears connected to different transitions, as is the case of the link between place **p1** and transitions **b** and **e** in Figure 2.

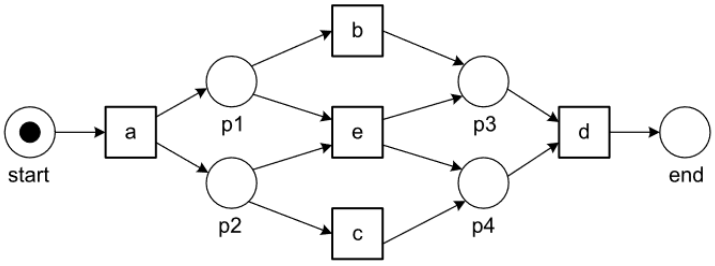

Figure 2 - Example of a Petri Net. Adapted from van der Aalst (2016).

Aiming at representing business processes, a BPMN model can also be considered. This modeling language is similar to the previous one in terms of its main goal, which consists of representing processes through a graph to help manage business procedures. Regarding the modeling elements, BPMN models include events — actions that happened during the flow of a process — which are identified by the circles in Figure 3; activities, i.e., the set of tasks that were performed, and which are associated with the squares in the same figure; and separating gateways, which can be of three types: AND, XOR, or OR (OMG, 2011). When compared with Petri nets, events are equivalent to places, and each BPMN model has a start event, a set of intermediate events, and an end event (van der Aalst, 2016).

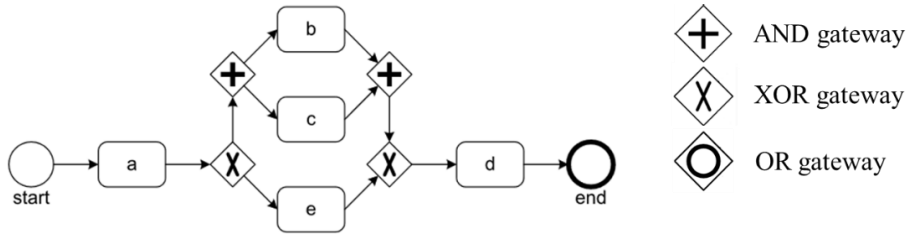

Figure 3 - Example of a process model designed using BPMN modeling language. Adapted from van der Aalst (2016).

Finally, process trees can also be a solution for model representation. Process trees are frequently affected to process discovery techniques and are specifically applied in the Inductive Miner algorithm (van der Aalst, 2016). The resulting model represents a hierarchical process model, which starts from a root node and includes the set of activities inherent to the process, according to the order in which they have to be performed. Furthermore, an interesting aspect of process trees that distinguishes them from previous modeling languages is the fact that they ensure that the resulting process models are sound (van der Aalst, 2016). In the work from van der Aalst (2016), the soundness properties that a model must check are specified: safeness, i.e., each place in a model cannot contain multiple tokens at the same time; proper completion, which guarantees that no event is executed when the process is already finished; option to complete, meaning that it is always possible to complete the process; and absence of dead parts in the model, that is, any place in the model is the result of an event. This is a crucial criterion as it ensures that all process instances are included in the model and act properly.

2.4. Types of Process Discovery Algorithms

In the literature, different process mining algorithms are often described. Concerning process discovery techniques, the Alpha Miner, Heuristic Miner, and Evolutionary Tree Miner algorithms are considered the most suitable (Nafasa et al., 2019).

Alpha Miner is seen as one of the first algorithms when talking about process discovery. The output of this algorithm usually corresponds to a Petri net representing the underlying process. In Figure 4, it is possible to see the steps that make up the implementation of this algorithm. First (1), the existing traces are identified, i.e., the sequence of activities of a given process. Subsequently (2), the order of these activities is analyzed, and a matrix is created with the respective relationships so that, finally (3), it is possible to create the final Petri net.

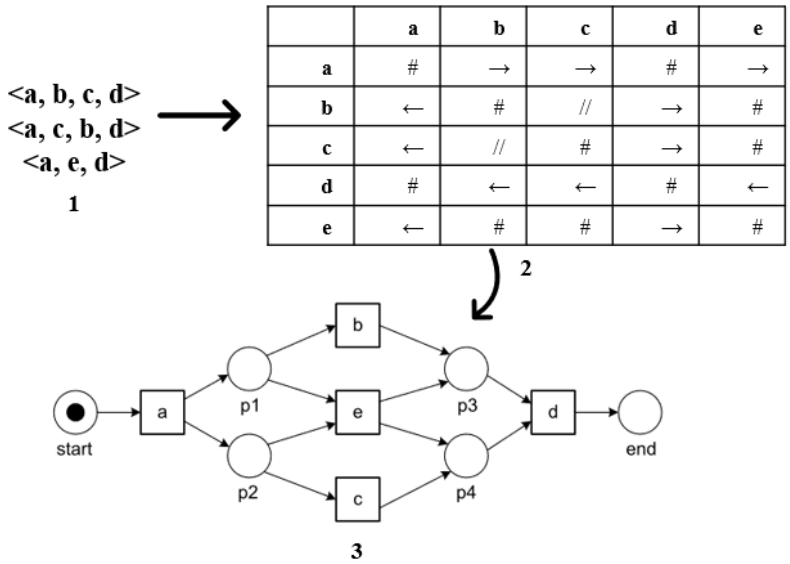

Figure 4 - Alpha Miner Algorithm. Adapted from van der Aalst (2016).

Although this algorithm has some advantages, such as its simplicity, it faces certain challenges in some situations, namely: when there is too much data, and therefore there may be some noise included[1]; in the presence of complex data; and in cases of data incompleteness (van der Aalst, 2016). Therefore, since real data may contain some of these characteristics, this algorithm would not be a good choice to represent real-life logs (van der Aalst, 2016). As a result, additional well-known algorithms, such as the Heuristic Miner and Fuzzy Miner, should be considered, as they try to minimize these limitations (Saint et al., 2021).

The Heuristic and Fuzzy Miner algorithms avoid the limitations of the Alpha Miner by creating the process map based on the frequencies of each event rather than just the order of the traces (Weijters et al., 2006). As a result, these algorithms prevent including infrequent traces in the model (van der Aalst, 2016). Consequently, this leads to more concise diagrams in the presence of outliers since only the most frequent paths are included and not those that may appear occasionally. The peculiarity of the Fuzzy Miner algorithm is focused on the construction of hierarchical models that attempt to group infrequent activities into subprocesses (van der Aalst, 2016).

The last algorithm to be introduced is the Inductive Miner. This procedure is described in van der Aalst (2016) as an algorithm that shows good results when dealing with a large

[1] In this case, it is considered that the data is noisy when it includes some outliers, i.e., when it has some records that do not represent the actual behavior.

dataset and with logs that include infrequent behavior. The output is usually a process tree, which preserves model soundness. Even so, the results can also be converted to other representations, such as Petri nets and BPMN models (van der Aalst, 2016).

2.5. Process Mining Tools

The aforementioned algorithms have been implemented in several tools which, despite serving the same purpose, have different characteristics. Thus, this subsection aims to describe some of the process mining tools most discussed in related works.

As suggested in the research from van Dongen et al. (2005), most tools tend to save the event logs in distinct formats. This characteristic can make it inefficient to apply different tools to the same dataset, which could be advantageous as they offer different analyses. It is in this sense that the use of the ProM framework is proposed. The usual input format for this tool is eXtensible Event Stream (XES) files, which corresponds to an XML-based standard format for event logs ("IEEE Standard for EXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams," 2016). The objective of creating this new format was to guarantee that users could record events through a more or less common structure that is recognized between different systems, hoping to maintain the accuracy and quality of the information contained in the logs.

ProM is often seen as a relatively intuitive and straightforward framework, not imposing significant difficulties on a user with little experience. For this reason, it is used in several works in the literature, such as in the research from Berti & van der Aalst (2019). While it may be beneficial for ProM to be considered an easy-to-use tool, other authors also see this as a limitation for drawing more specific and in-depth conclusions (Berti et al., 2019). In the work of Mans et al. (2014), the authors suggested the use of RapidProM — an extension to ProM within the RapidMiner data science platform — as a response to these challenges. As such, by connecting an advanced analytics tool with process mining, it is possible to achieve more complex and robust solutions (Berti et al., 2019). However, Berti et al. (2019) also characterized this extension by its difficult integration and lack of customization, which remains a disadvantage.

Therefore, to respond to the previous issues, Berti et al. (2019) presented Process Mining for Python (PM4PY). The authors introduced a set of advantages of using this library, such as the possibility of combining more robust and customized algorithms and

the fact that it also allows the use of algorithms from different fields. For example, process mining algorithms can be used by importing the PM4PY package and, at the same time, integrated with machine learning algorithms by importing libraries such as Scikit-learn (Pedregosa et al., 2011). In addition, Berti et al. (2019) also emphasized the documentation about each library, considering it a valuable resource to support the users' developments. For these reasons, this tool has been used and recommended by other authors in performing process mining research, as is the case with the work from van der Aalst & Berti (2020), through which they use PM4PY to try to discover Petri nets through a set of event logs.

2.6. Literature Overview

In light of what has been discussed in the previous subsections, it is important to highlight some conclusions. Firstly, we can conclude that the subject of process mining has special relevance in the literature at a scientific level. Furthermore, specifically for the case study under analysis, it is important to emphasize that there is a set of criteria that the chosen tools and algorithms must verify. These criteria are based on the literature and will be used to recommend the most suitable tool and algorithm (see Table 1).

TABLE 1 - COMPARISON CRITERIA FOR TOOLS AND ALGORITHMS

	Criteria	Description
Tools	License	The license type that the tool has.
	Documentation	Check if there is documentation available about the tools and how they work.
	Integration of process mining algorithms	Check which process mining algorithms the tool can employ.
	Integration of process mining techniques	Check which process mining techniques the tool can develop.
	Integration of metrics	Check whether the tool can provide metrics to evaluate the performance of the process.
	User-friendly tool	Check if the use of the tool is accessible, i.e., if it is considerably complex to use or not.
Algorithms	Produce sound models	Check if the resulting process models are sound.
	Deal with data incompleteness	Check if the algorithm deals well with data with few events.
	Produce good results with real-life logs	Check if the resulting process model accurately represents the actual process.

3. EMPIRICAL WORK

In the context of the underlying business process, the empirical work proposed in this section is based on the information presented in the literature review. The complete architecture is shown in Figure 5. We highlight that it was possible to conceive a method that includes the integration of programming languages and other tools to create representative process models of document flows, which is the goal of this project.

The Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology inspired the methodological approach used during this work and is organized into a few stages (Costa & Aparicio, 2020). The first phase included, at first, the understanding of the business environment, through a contextualization of the department and the project. Then, the practical part of our work starts with the extraction of data from the Database Management System (DBMS) of the company, through an SQL query. As a result of this initial phase, we obtain an Excel file including all the data, and that will be the target of the second stage of the methodology, in which the data and characteristics of the available variables are understood. The third stage, i.e., the data preparation phase, is a must to ensure that the raw data will be transformed into a consistent format. Consequently, we obtain a file where the data are in the desired format, which allows us to apply some mining techniques, and later visualize and analyze the developed models. As each tool offers different advantages, the modeling phase of the process models will be done in three different tools, as shown in Figure 5. To analyze those models, it was also required to figure out which metrics were built into the selected tools for the project. This is a crucial step as these measures allow for identifying conclusions and comparisons of different performances, which is what the final phase is all about.

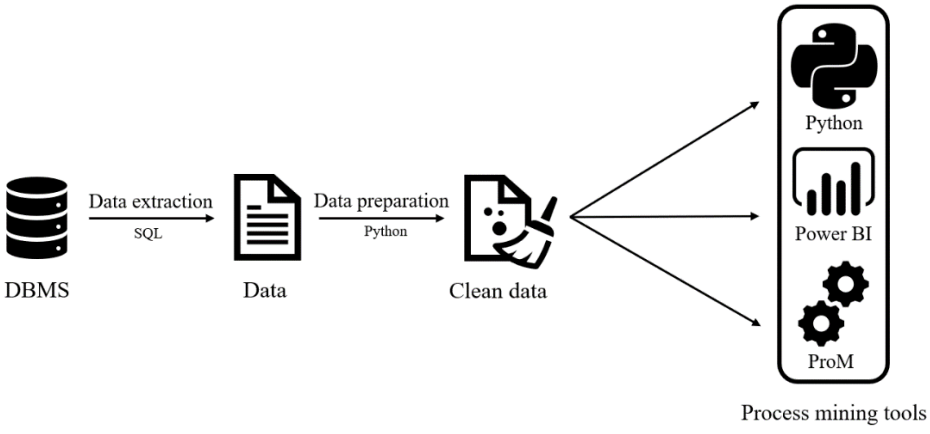

Figure 5 – Empirical work architecture.

3.1. Technologies

To decide which tools we wanted to use in this work, we complemented the description provided in Section 2.5. with additional research about which tools are additionally available in the market. In this sense, the ones that were used to perform the process models in this work are Python, Power BI, and ProM. We considered it relevant to produce models using these three tools as each one of them offers different advantages that complement the analysis, even if the final goal is the same, i.e., to produce process mining models. Thus, with the study of these tools, we expect to understand which are the most suitable and suggest some recommendations to process mining projects.

In the first phase, Python was used for data processing and cleaning through the Pandas package. Then, the PM4PY package, which is the package associated to process mining, was installed in Jupyter through the following command:

```
1.  !pip install pm4py
```

Furthermore, since the objective is to create and visualize models that represent the flow of the processes, it was also necessary to download and install the GraphViz package.

Power BI was also considered as it allows for the creation of process mining models and integrates some unique metrics, such as the average and the sum of the duration of an activity and each activity's frequency.

Finally, the use of the ProM tool to conduct analyses was also demonstrated to be essential owing to the integration of multiple process mining plug-ins, which allows the creation of models from several perspectives. Besides, the fact of providing visualizations that present the tokens running in real-time makes the analysis somewhat more interactive and gives some insights concerning how long each event takes to be executed. Finally, this tool also includes several metrics, like the total number of cases and events, which might be useful to analyze the performance of processes to subsequently identify anomalous situations.

3.2. Data Preparation

Data extraction was done through a script on Microsoft SQL Server, which is the relational database management system used by Quidgest. The extracted data were exported to an excel file, and therefore, it was imported into Python through the use of the Pandas library.

```
1.  import pandas as pd
2.  event_data = pd.read_excel("File")
```

Concerning data treatment, the first step included removing the columns that weren't necessary, the null values, and the lines with incomplete data using the same library. After that, it was necessary to check the data types of the values. This step made it possible to perceive that two of the columns containing date values were being saved as objects, which was incorrect. To solve this issue, the following piece of code was required:

```
1.  event_data['DataEnvio'] = pd.to_datetime(event_date['DataEnvio'])
2.  event_data['DataResposta'] = pd.to_datetime(event_date['DataResposta'])
```

After all the necessary data transformations, the final variables are defined in the following Table 2:

TABLE 2 - DESCRIPTION OF VARIABLES

Variable Name	Data Type	Description
GuidRegisto	Object	Unique process identifier.
TipoDocumento	Object	Description of the type of document that is associated with a given process.
DataEnvio	Object	Start date.
TipoTarefa	Datetime64[ns]	Description of the activity.
DataResposta	Object	End date.
Destino	Datetime64[ns]	Destination department.
Organização	Object	Institution's unique identifier.

3.3. Event Log Preparation

After processing the data, and since the input for implementing the process mining algorithms is an event log, the focus shifts to preparing the data according to the requirements to fit into this context.

Depending on the tool that will be used, the file format where the logs are recorded may differ. For example, the most commonly accepted standard file format in ProM is

the XES. In contrast, Python and Power BI both accept Comma-Separated Values (CSV) and Microsoft Excel Open XML Spreadsheet (XLSX) files. In this sense, ProM offers the possibility of implementing a plug-in that allows the conversion of a CSV file to XES, allowing process mining techniques to be applied to data in a format other than the one originally recognized by this tool. However, suppose this plug-in was not meant to be used. In that case, Python can be used to convert an event log object to an XES file using the following command, where the first argument is the data frame being used and the second argument is the final directory for the resulting XES file:

```
1.  from pm4py.objects.log.exporter.xes import exporter as xes_exporter
2.  xes_exporter.apply(data_frame, '<path-to-file>')
```

Additionally, to be able to apply process mining algorithms, it is necessary that each case in the event data can be distinguished by a unique identifier; the respective activities, and a timestamp associated with each of these activities. In Python, this correspondence is done through the use of PM4PY package:

```
1.  data_frame = pm4py.format_dataframe(data_frame, case_id = '<Column>', activity_key
    = '<Column>', timestamp_key = '<Column>', timest_format = '%Y-%m-%d %H:%M:%S%z')
```

With this instruction, it is possible to give the appropriate format to the data frame, within the context of process mining.

The identification of the case id, the activity, and the timestamp is also a must in Power BI. Nonetheless, before proceeding with this identification, the data processing was carried out in the Power Query Editor. In particular, the data type of columns representing dates was changed, as it was found as Text/Number and not as Date Time. Additionally, the null values and the unnecessary columns were eliminated as well. Then, Power BI offers process mining by importing visuals. As a result of the construction of these visuals, each of the three critical factors — case id, activity, and timestamp — may be matched to the query's corresponding columns.

Finally, in ProM, identifying these elements is done almost automatically by each plug-in. In this sense, it is only necessary to ensure that it is done correctly.

4. RESULTS

This work focuses on creating models for managing document flows. To that purpose, the study was separated into three stages: an analysis made in terms of the activities that are part of the global process; an analysis carried out in terms of each type of document to try to identify which activities and individuals were involved in each document; and an analysis to study the influence of each institution that makes up the university. The development of each of these analyses will be described separately in the following three subsections. As the process models are quite extensive and considering the page limit that is imposed, some models appear in Appendix A. Nonetheless, the Jupyter notebook containing the entire analysis will be shared.[2]

4.1. Analysis 1

As previously stated, the purpose of this first analysis is to study the activities that are inherent to the process. As a result of the evaluation of the process models derived from each tool, we expect to conclude which activities should receive more attention and try to understand which measures should be taken to optimize the performance.

- **Analyses made through Python:**

The first tool used was Python, and therefore, the changes to be made to the data are those already described in Section 3.3, which are related to the event logs preparation. So, at this stage, we need to give the appropriate format to the data frame under the process mining context, which is done through the use of the PM4PY package.

With the event logs prepared, the process mining algorithms can also be applied through the PM4PY package. The algorithms that will be employed are the Alpha Miner, Heuristic Miner, and Inductive Miner.

The first algorithm we attempted was Alpha Miner. As noted in Section 2.4, this algorithm does not perform well on real-life logs and is therefore not recommended. Even so, we tried to create models with this algorithm to verify these conclusions. In fact, as shown in Figure 20, the algorithm's output is a model with separate activities, i.e., there are no links between the different activities. As such, this analysis enforces the conclusion that the algorithm has serious issues when dealing with this type of data.

[2] Jupyter notebook available on: https://github.com/catparente/process-mining-dmgt

Then, we used the Heuristic Miner algorithm, which tackles some issues of the Alpha Miner. This algorithm allows the specification of some parameters, such as a threshold for the dependency factor. This parameter denotes the dependency relation between the different activities in a process and can range from -1 to 1. A value close to 1 corresponds to a positive dependence, i.e., between two consecutive activities X and Y, it means that Y always appears after X; a value close to -1 means a negative dependency, that is, activity Y never appears after X, and a value close to 0 indicates that no relationship is detected between the two activities. With this, we can see that relationships with negative values for this parameter are not interesting for our study as they do not correctly represent real-life events. PM4PY's default threshold for this factor is 0.5, but we also developed models for values equal to 0.99. This change was made since the logs used in this analysis contain real events, so it was thought that it would be interesting to understand the events that are inherent to almost all processes and not just those that happen more sporadically.

Thus, as a result of applying the Heuristic Miner algorithm to the logs, we obtain a heuristic net for activities with dependency equal to 0.5 in Figure 21 and equal to 0.99 in Figure 22. These models include the following elements: an initial and an end marking connected to the respective activities; arc labels representing the absolute frequency of each relationship; and, inside the boxes, the absolute frequency of each activity. This algorithm leads to the identification of concurrent activities through the arcs, i.e., activities that are not related to each other. For example, according to the model in Figure 22, the activity “Ofício” and the activity “Nomeação” are said to be concurrent or parallel since a specific order between them was not detected in the logs. Furthermore, the activities are shaded regarding their frequency, where the most frequent activities are associated with darker colors and less frequent activities with lighter colors. Hence, based on this pattern and confirmed by the frequency value, we can conclude that the most frequent activity in this process is the activity “Despacho”, out of which 866 occurrences were the rework of the activity. Considering the high-frequency value and the fact that it has the highest value for self-loop, we can conclude that this activity is perhaps damaging the performance. In the context of a company, the occurrence of self-loops might suggest that the individual who executed the process did not actually know how to do it, or the process was set up ambiguously. Consequently, this can mean more costs for the company, and therefore, more attention should be paid to this issue.

Comparing the models obtained for each of the values of the dependency parameter, we can conclude that there are some problems for the model with this value equal to 0.99. Specifically, some relationships appear with a frequency value equal to 0, so they could simply be omitted from the graph, as is the case of the relationship between the activity "Assinatura" and the activity "Homologação". This problem can be fixed by choosing a dependency threshold smaller than this number, as shown in the model with a parameter value of 0.5. However, it should be noted that this process model may include activities that are not as important as desired. Thus this trade-off must be adjusted to the needs of each company and specific analysis.

The last algorithm to be addressed is the Inductive Miner. As clarified in the literature, this algorithm can derive different process models, namely a process tree, a Petri net, or a BPMN model. Therefore, to understand all these modeling languages, we generated the process models in all of these options, as can be seen in the Jupyter notebook. Nonetheless, the analysis that will be discussed next will refer to the Petri net process model. Thus, similar to the process model resulting from the previous algorithm, this model also presents the absolute frequencies of activities and each relationship, as well as the same relation between the color of the box of each activity and its frequency value. Therefore, the conclusions to be drawn will be similar, as we expected. The difference between the previous model and this one is related to the fact that the latter includes additional activities, as the Inductive Miner algorithm does not incorporate the definition of the dependency threshold. In addition, we can also consider that the way this algorithm presents the activities gives a much more concrete temporal notion when compared to the previous algorithms. Thus, the models generated through this algorithm allow us to get some insights into which activity follows which.

The Petri net resulting from this algorithm also includes some black squares, which denote invisible activities. This indicates that our procedure can either begin with the activity "Validação do mapa de pessoal" or go to the other path, by skipping this event. Based on the data we have, we can conclude that this is a low-frequency activity, so it should be studied more carefully to try to understand whether it makes sense for the company to continue to perform it. Nonetheless, these data still refer to an initial phase of project adoption, so this situation could be mitigated with the inclusion of more data,

in the sense that with a larger sample we could confirm whether the inherent activity ("Validação do mapa de pessoal") persists with few observations.

- **Analyses made through Power BI:**

The analyses done through Power BI do not follow specific algorithms like in Python. In this case, it is necessary to import the available visuals that are capable of making process models.

One of the most recognized software implemented in Power BI that is related to process mining is the "PAFnow Process Mining". In addition to performing process discovery techniques, it also allows for conformance checking; however, this analysis requires a paid subscription. Creating process models in Power BI just requires assigning the particular column of the logs to the specific elements necessary for process mining — the case id, the activity name, and the timestamp. Nonetheless, to generate a model through this visual, it is also required to choose the number of variants to integrate. In this context, a variant refers to an alternative way of carrying out a process. Hence, if we start with a variant value equal to 1, we get the simplest possible model, usually including only one activity. As we increase the number of variants, we get the different paths that better reflect reality, and the model becomes more and more complex. Thus, to be consistent with the process models generated by the remaining tools, we chose a variant value of 135, far from the maximum value of 288 variants that are extractable from our event log.

The final model is shown in Figure 23, which includes a start and an end node; the different activities connected, and a set of additional information regarding not only the volume of each activity but also the average and the sum of the duration. Looking at these metrics for each of the activities, we conclude that the activity "Despacho" continues to be the most frequent. In addition, it is also the activity whose sum of durations is greater. This activity's average duration is 3 days and 9 hours; however, there are activities with higher values for this metric, such as the activity "Informação". Nonetheless, when considering the frequency of the activity "Despacho", the average duration becomes problematic even if it does not seem to be an issue by itself. Besides displaying these metrics for each activity, this visual also presents this set of metrics for the relations between events. This allows the company to have an idea of the time that is spent in the process between activities. Since time is a scarce resource and one of the highest costs

for companies, this is a critical metric for determining why so much time is spent transitioning between tasks and determining if this makes sense.

Power BI allows the importation of other visuals, like “Process Mining by ProcessM” (PmBI). For this visual, it is also necessary to choose the value of the variants to include when creating a process model, which, in this case, is not an absolute value, but a percentage. When we increase the percentage, we see that this model is not so sensitive to changes, so when we put 100% of variants, we get a reasonable model, similar to the previous ones and not too complex. Concerning the metrics, this visual displays the frequency of each activity and each relationship, as well as the total, the average, the maximum, and the minimum duration of each relationship. Furthermore, it also highlights the relationship with longer duration, increasing its thickness. As expected, the relationship with the longest duration is the self-loop in the activity “Despacho”.

In the case of the “process.science Process Analyzer Free” visual, the value of the variants ranges from 0 to 654, with 654 variants yielding to a model with a high level of complexity due to the large number of connections it contains. Therefore, we chose a value of 200 variants to be included. Similar to previous visualizations, this model also displays the shade proportional to the frequency of each activity. In this sense, the activities "Despacho" and "Para Conhecimento" have the darkest color, and hence the highest frequency, which is consistent with the preceding process models.

- **Analyses made through ProM:**

The first step to perform process models is to import the event logs, which must be in XES format. In this case, we opted to convert the file with the event logs to XES using Python since the initial analysis was already completed in this tool and we can just use the final data frame that includes the process mining requirements.

After importing the data, ProM shows us the summary of the information regarding the logs, as shown in Figure 6. We can perceive the total number of processes, cases, and events based on this data. In addition, it also provides the average, minimum and maximum number of events per case. Thus, according to this figure, we can observe that although there is a considerably high maximum value for the number of events per case, the average value for this metric is equal to 3, which leads us to conclude that the traces to perform the tasks are usually not very long. Ideally, this dashboard also displays

information regarding the number of complete events that the data contains; however since we are working with real data, there may be cases that are not complete and, therefore, this information does not appear.

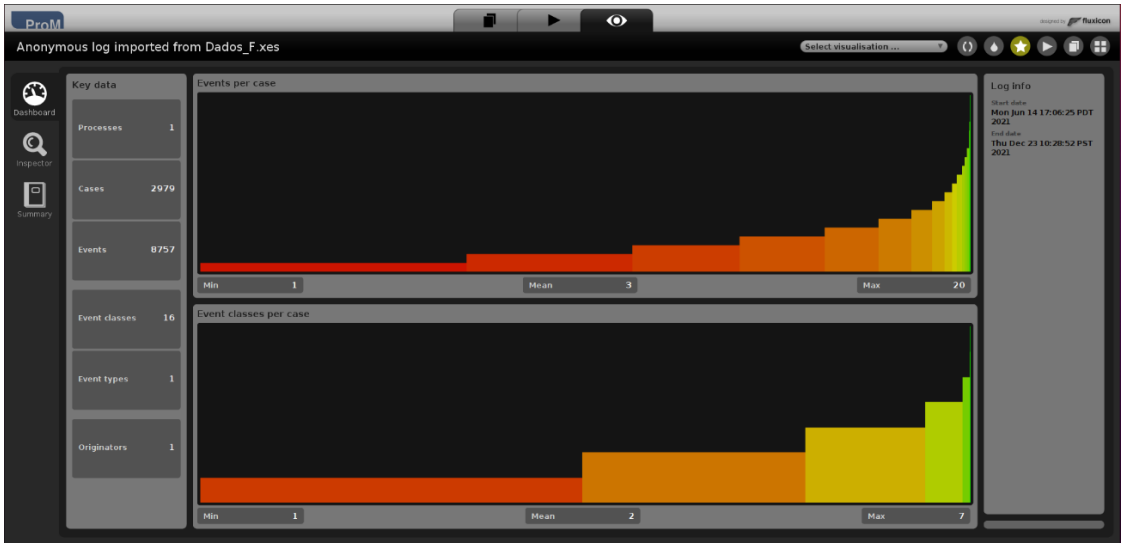

Figure 6 - Summary of information contained in the logs.

To implement the algorithms, we have to import a set of plug-ins and apply them to the input, which is the XES file that was previously imported. The first algorithm to be applied is the Alpha Miner algorithm, which is imported through the plug-in named “Alpha Miner”. The model follows the previous analysis, and the same issues are present.

We also applied the plugin "Mine for a Heuristics Net using Heuristics Miner", to create a process model based on the Heuristic Miner algorithm and the plugin "Mine Petri net with inductive miner" to obtain a process model resulting from the Inductive Miner algorithm. As predicted, both models are similar to the ones produced by the same Python algorithms. Nevertheless, Heuristic Miner's model is not in the form of a heuristic net, as seen in Figure 7.

ProM also allows us to create animated models through which we see events emerging in real-time. This is possible in the case of the Inductive Miner algorithm, through the plug-in "Mine with Inductive visual Miner". In Figure 8 we see one of the moments that captures the result of this algorithm. The yellow circles are called tokens and represent each event that emerges in real-time. This plug-in allows for an advantageous visualization, as we can observe which traces are more frequent, as well as in which places the events accumulate more, as these are the locations that may be regarded as process

bottlenecks. So, when looking at this model in ProM, we see that there is a very high concentration of events in certain areas of the process, mainly in the two final activities, "Para conhecimento" and "Despacho". This leads us to consider these two activities as bottlenecks in our process, which is consistent with the conclusions of the previous tools.

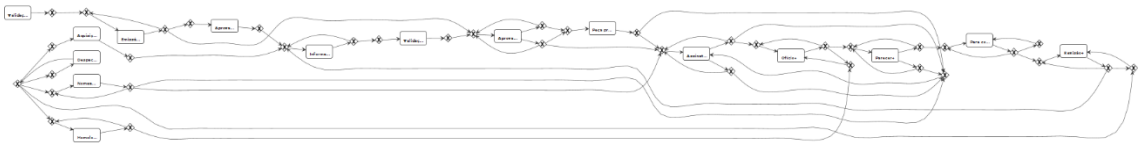

Figure 7 - "Mine for a Heuristics Net using Heuristics Miner" plug-in results using ProM.

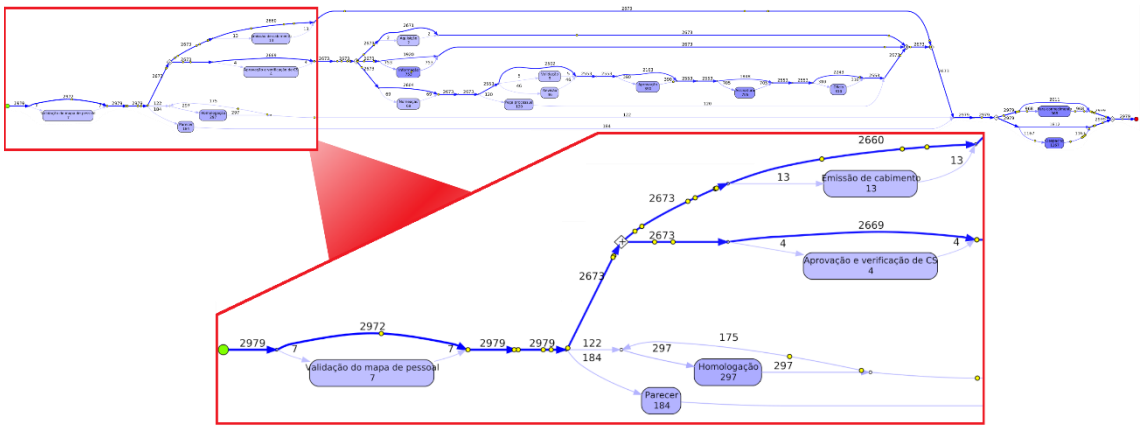

Figure 8 - “Mine with Inductive visual Miner” plug-in results using ProM.

4.2. Analysis 2

The goal inherent in this section was to study, for each document type, which activities and departments are involved in each process. As a result of implementing each algorithm for the first part of this analysis, we will end up with a set of process models displaying the tasks performed for each document type. For the second part, we will have a set of models showing the departments involved in creating each document. This will allow the company to analyze the performance and the challenges of each process independently since it will be able to see in detail which activities and intermediaries are engaged in the process, and which ones could be deteriorating the performance.

- **Analyses made through Python:**

The first step regarding Python was to understand how many document types exist in the logs to create a separate data frame for each unique document type. We carry out the

set of algorithms for each of the scopes of analysis: the first one in which we assign the column “TipoTarefa” to the activity key, and a second moment in which the activity was the department, represented by the column “Destino”.

We started by using the Alpha Miner algorithm, and, for most of the results, it generated models with problems, which was expected according to the results of Analysis 1. Surprisingly, it appeared to work correctly for four types of documents: the “Nota de crédito”, “Extrato”, “Ato” and “Certificado”. However, by checking the event logs related to each of these documents, we can confirm that for some of them there are events that do not appear in the process model. This can happen since this algorithm does not deal well with incompleteness, that is, with event logs with few events, which is what is happening here. Therefore, even if this algorithm manages to reproduce a complete model, it does not have enough data to be able to discover the true workflow. As an example, we can see Figure 9, which represents what happened in the document type “Extrato”. When comparing the set of logs to the resulting Petri net, we can observe that the activity “Para conhecimento” appears just once, and is thus excluded from the model.

TipoDocumento	DataEnvio	TipoTarefa
Extrato	2021-07-23 16:11:53.103	Informação
Extrato	2021-07-25 23:54:24.793	Despacho
Extrato	2021-11-25 09:48:13.920	Despacho
Extrato	2021-09-28 14:58:55.073	Informação
Extrato	2021-07-08 13:57:48.330	Aprovação
Extrato	2021-07-05 21:26:38.910	Informação
Extrato	2021-11-03 15:22:37.517	Informação
Extrato	2021-11-03 15:53:33.240	Informação
Extrato	2021-09-28 15:01:27.353	Para conhecimento
Extrato	2021-12-14 13:52:34.560	Informação

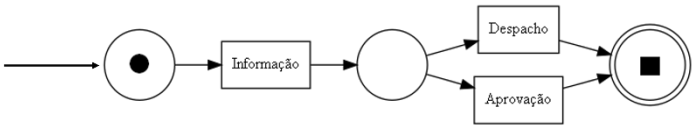

Figure 9 - Event logs and Alpha Miner results using PM4PY for the document type “Extrato”.

When developing models through the Heuristic Miner, we realize that the same issue of the previous algorithm occurs for some types of documents. In this case, there are not only models that do not include all the events that appear in the data, but there are also some models that contain activities isolated from the workflow. We can consider Figure 10 as an example of this last problem, in which it appears an activity that is not linked to

the others. This might happen because this algorithm also has problems in dealing with low-frequency events. Thus, by including more data, these issues may be addressed.

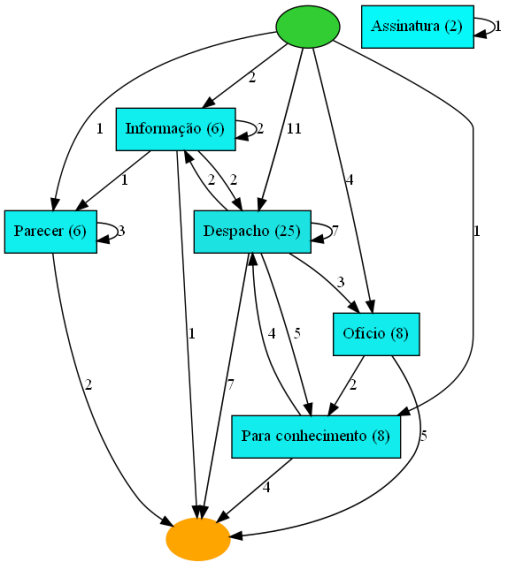

Figure 10 - Heuristic Miner results using PM4PY for the document type “Carta”.

Furthermore, for this type of analysis, setting the Heuristic Miner's dependency threshold for values above 0.5 no longer makes sense, as the activities to be included for each analysis are already quite reduced. Even so, if we make this change, we see that we obtain process models where some activities are separated from the model and others are in parallel, which means that they are seen as independent of each other. Although these problems exist for some types of documents, the algorithm continues to allow the discovery of representative models for most processes.

By analyzing the set of Petri nets obtained through the Inductive Miner, we can conclude that this algorithm leads to the best process models as it seems to include all of the information present in the logs. These process models describe the behavior of each document type in a more detailed way as it does not exclude activities from the models. In this sense, this would be the suggested algorithm for performing analyses regarding the performance of each document type.

To summarize this first part, in which we analyzed the activities that are part of creating each document type, we noticed that the activity “Despacho” is present in most of the workflows. This confirms the fact that it is one of the most frequent activities, even when looking at each document type individually. Therefore, this is an event whose metrics must be investigated, to assess whether the performance is as idealized.

Additionally, the same algorithms were implemented, but now referring to the study of the departments that are involved in the development of each document. In this way, the company can understand which departments are participating in each process, as we produced different models for each of these. By examining the resulting models for each of the algorithms, it is possible to evaluate whether all the departments involved are required and, if not, whether it would be possible to remove a specific department, and evaluate if this change enhances the process' performance. Moreover, it is possible to perceive that there are indeed very complex processes, which involve several intermediaries. This conclusion justifies the fact that there are more time-consuming processes, which should be carefully analyzed. Concerning the performance of each algorithm, we can verify that both the Alpha Miner and the Heuristic Miner have the same problems of not including the low-frequency events, so it is recommended to focus on results from the Inductive Miner.

- **Analyses made through Power BI and ProM:**

In both Power BI and ProM, this sort of analysis is not so accessible to carry out. Performing this analysis in Power BI implies creating a new query for each type of document. Likewise, in ProM, we have to import an independent log file for each document, which proves to be a time-consuming operation given the variety of documents. Therefore, these tools would not be so suitable for studying these types of variables, as it is expected that, for longer processes, there will be an even greater number of document types.

However, since these tools provide additional metrics and types of visualizations, it may be helpful to deploy some models in these tools for critical activities that need to be analyzed more deeply. In this sense, according to the results previously obtained, it could be useful to develop models for the process “Despacho”, as it could allow the company to understand another type of metrics to evaluate the performance of each department. By creating models for this process in Power BI, we can observe that there is, in fact, a relationship between two entities that takes approximately 12 days to complete. As a result, it is critical to figure out to what extent this relationship can be optimized, as it consumes a lot of resources compared to the other relationships' average duration.

4.3. Analysis 3

Finally, one last piece of information presented in the logs must be examined, namely, the fact that the university is made up of different colleges. For that reason, each event is associated with a specific intermediary. As a result, it was thought that it would be useful to know if, on the one hand, processes were transitioning between the various institutions and, on the other hand, if the workflow of each entity differed significantly.

- **Analyses made through Python:**

We resorted to the Heuristic and the Inductive Miner in Python to achieve the defined goals and complemented it with a directly-flows graph since this latter indicates the duration of each linkage, which is crucial information. Figure 11 and Figure 12 show the two final models of these algorithms. To ensure anonymity, the colleges were coded with the notation "Organização X", where X is a numerical identifier that encodes each institution.

From the two resulting models, we can see that all organizations are included in the process, and thus, the information flows between them. From Figure 11 we can conclude that Organization 3 has the highest frequency, having 5287 events, also including 3230 self-loops. Besides, from the model in Figure 12, we can perceive another important metric: the time taken to transfer information between different organizations. On average, the duration of the information flow seems not to be very long. There is only one relationship whose thickness is highlighted, as it has a duration of 3 months, which is quite considerable. Thus, this relationship should be evaluated, to determine whether it is reasonable to take so long to transfer the information from Organization 1 to Organization 4. The long duration must be assessed and can be due to multiple reasons, for example, it might be important to verify if the information systems are receiving and sending information correctly and check whether the intermediaries are carrying out their activities efficiently.

To complement the aforementioned observations, we created a single data frame for each organization, to understand the flow of tasks for each one. When we look at the process model for Organization 3, which is represented in Figure 24, we realize that it includes several tasks and relationships, being the largest model. This observation is in line with the fact that this is the organization with the highest frequency. Thus, it might

be advantageous for the organization to figure out whether any events are taking longer than planned. If so, these activities could be processed through different intermediaries to improve the overall performance of the processes.

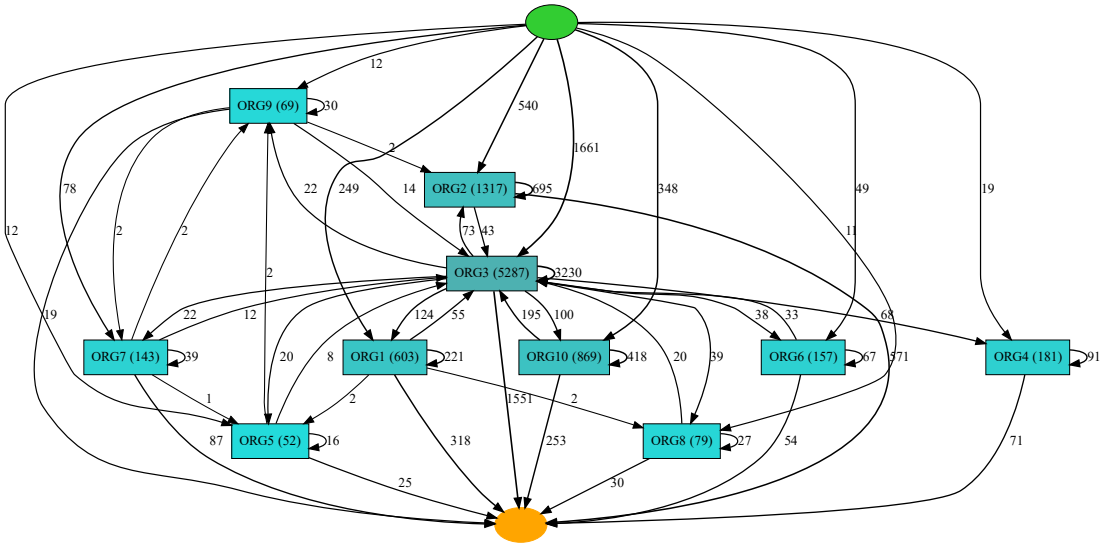

Figure 11 - Heuristic Miner results using PM4PY within the scope of Analysis 3.

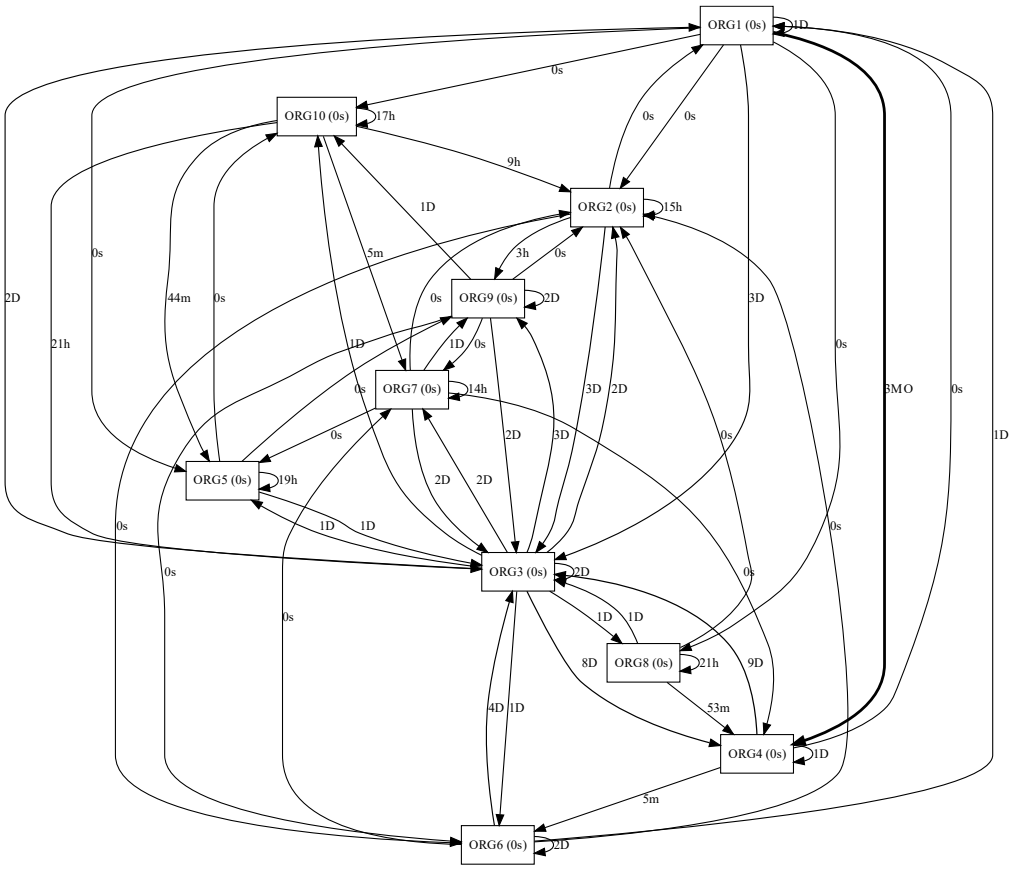

Figure 12 - Directly-Follows Graph results using PM4PY within the scope of Analysis 3.

We also made new data frames including information about the document types and the different organizations, to try to evaluate and compare the workflow of the same document made by different organizations. We created models through the Heuristic and the Inductive Miner, and we concluded that, in this situation, the first algorithm includes the same issues about not including certain activities. As a result, we focused on the Petri nets resulting from the second algorithm. When looking at these models, we can see that, in general, each organization displays a different task flow for the same document type. As such, it appears that there are no general workflows, i.e., similar workflows adopted by several organizations. So, the focus should be on assessing metrics for each of these processes, such as the average and total duration, to determine which one performs best and to generalize the adoption of all or part of the workflow.

Finally, to understand the overall process, we developed the following model, which shows the relationships between the different organizations and the tasks that each one performs. By observing this model, we see that we end up with a very complex and unstructured graph, resembling what is often called a “spaghetti diagram”. Although it is difficult to draw great conclusions from this model, we can observe a cluster of activities from Organization 2. If we observe the model in detail, we see that Organization 2 only appears in that place on the graph, with only direct relationships between them. This means that this organization handles most of the documents internally, rather than relying on external entities. As there are few intermediaries involved in the process, this might be a plus for ensuring data compliance.

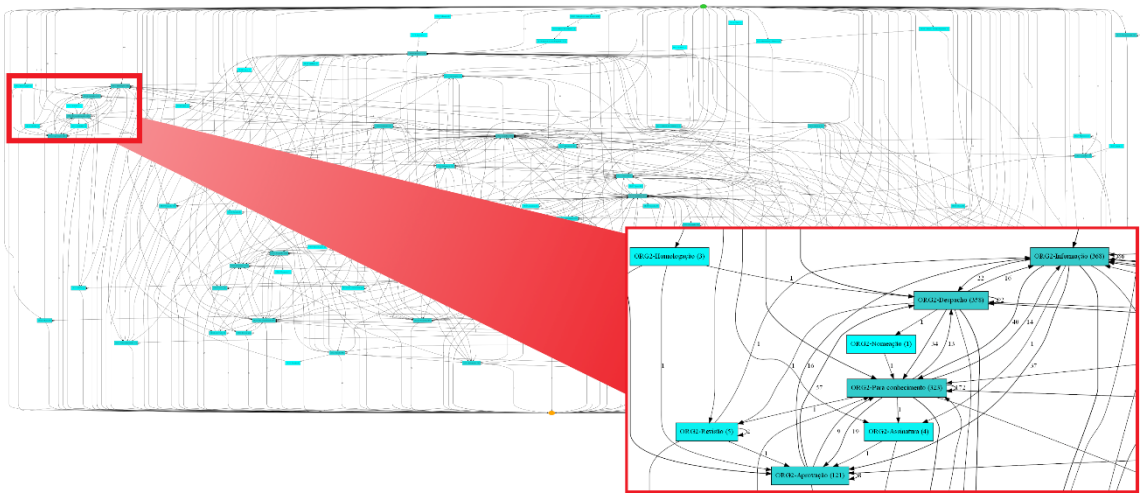

Figure 13 - Cluster visualization in the process model for all tasks and organizations.

- **Analyses made through Power BI and ProM:**

In this case, Power BI and ProM also become more limiting tools as they do not allow for the flexibility inherent to programming languages. While they allow us to complement our research with additional metrics and visualizations, they also make the analysis much more time-consuming as we have to create different inputs for each entity that we want to analyze. In this sense, and considering that the developments made using Python are much more automated, we advise that, for these types of analyses, Power BI and ProM should be reserved for exceptional processes that require more careful analyses.

Therefore, considering the prior analysis, it could be useful to develop models for Organization 3, as it is the most complex entity. Through this, we can compare the models resulting from these visuals with the ones obtained through Python, complementing the analysis with the metrics provided by each tool. With this, we can confirm which tasks are taking the most time and therefore worsening overall performance. Specifically, the most resource-intensive activities are self-loops, which could indicate that there may be a gap in the information received by the intermediaries that perform the activity.

4.4. Comparison of results for a specific process

To understand which tools performed best, we compare the generated process models related to "Acumulação de Funções" using the process mining tools, with the handmade activity diagram related to the same process. The activity diagram was made a priori, to understand which activities could be inherent to each process. Thus, it is expected that there will be differences when comparing it with the generated models, as these are based on the real logs and not on the events that are anticipated to happen. In this sense, effectively when comparing the models, we see that there is a set of events present in the activity diagram that are not present in the generated models and vice versa.

Also, by comparing the results obtained through each algorithm, we can conclude that the Inductive Miner is the one whose results seem most similar. This is explained by the fact that, with this algorithm, we can have a perception of the order in which the events take place, as in the activity diagram. As for the Heuristic Miner, this is not the case. In fact, in the model resulting from this algorithm through PM4PY, the visual perception is that the activity "Despacho" happens first, as it seems to appear first in the graph.

Additionally, the activity diagram modeling language allows to clearly differentiate the paths according to certain decisions, which is not as obvious in the obtained models.

Even so, with the Inductive Miner, it is easier to make this differentiation of paths through the use of invisible activities. Table 3 summarizes the differences in the generated diagrams.

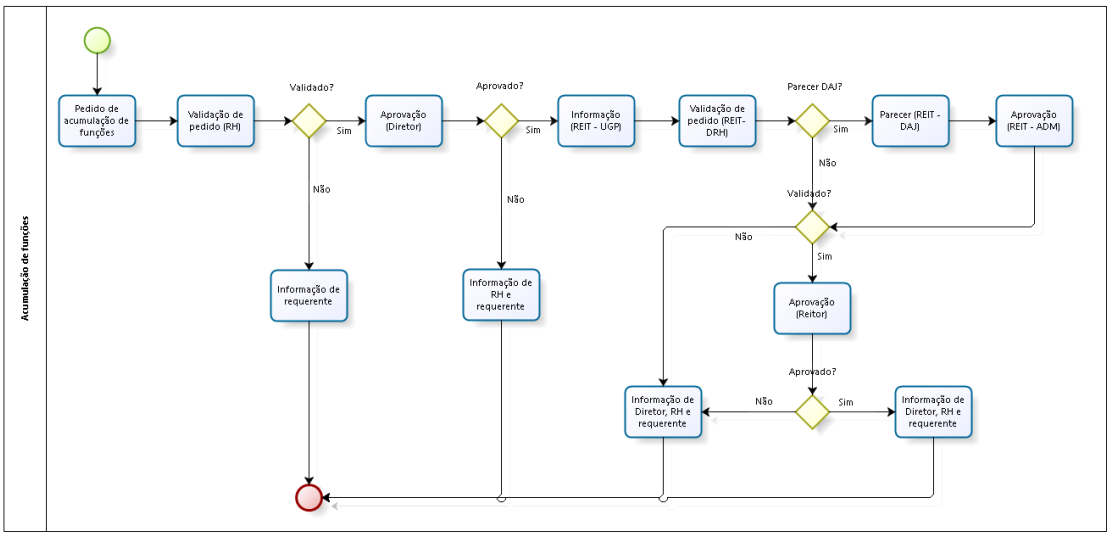

Figure 14 - Activity diagram for “Acumulação de funções”.

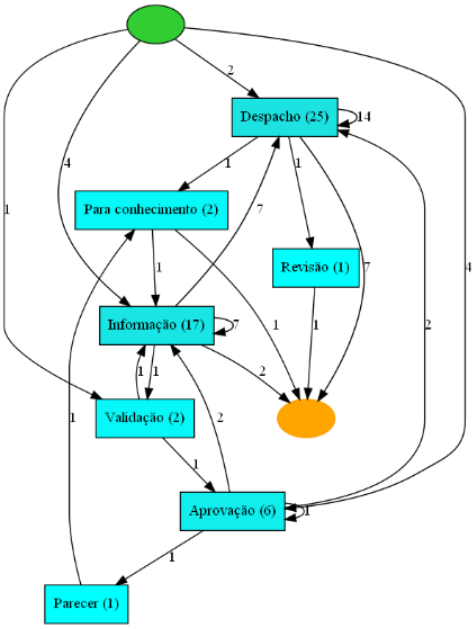

Figure 15 - Heuristic Miner results using PM4PY.

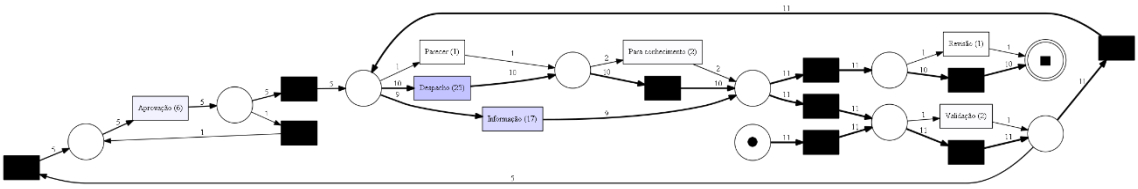

Figure 16 - Inductive Miner results using PM4PY.

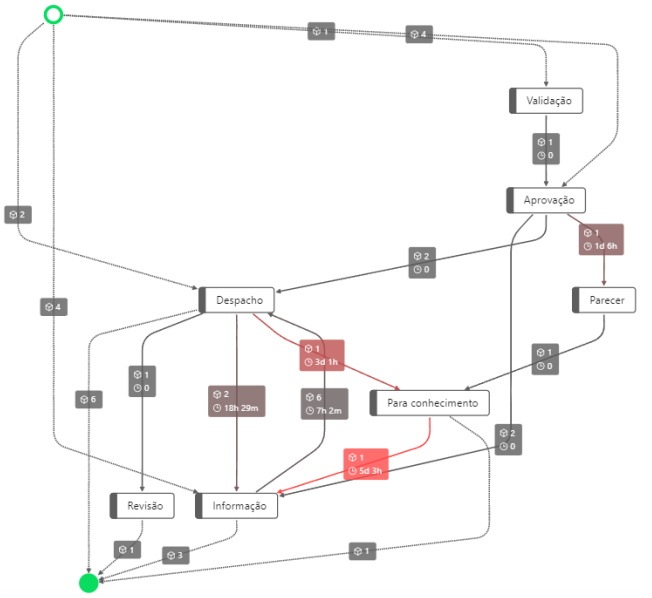

Figure 17 - PAFnow Process Mining results using Power BI.

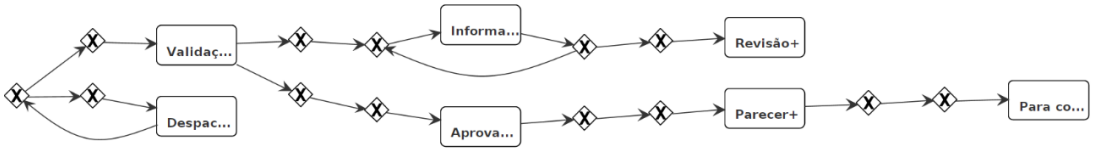

Figure 18 - "Mine for a Heuristics Net using Heuristics Miner" results using ProM.

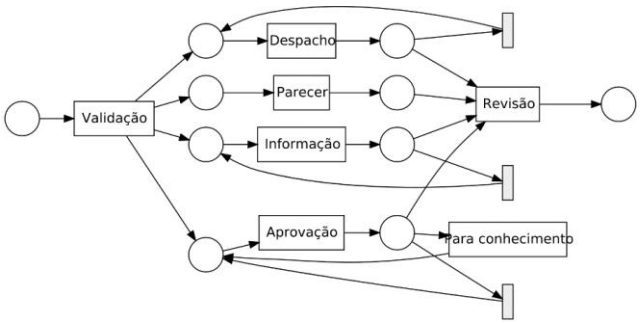

Figure 19 - "Mine Petri net with inductive miner" results using ProM.

TABLE 3 - SUMMARY OF COMPARISONS WITH THE HANDMADE ACTIVITY DIAGRAM

Figure	Algorithm	Tool	Comparison
15	Heuristic Miner	PM4PY	Different activity flow; no temporal notion of the order in which the activities took place.
16	Inductive Miner	PM4PY	Different activity flow; better perception of the order in which activities occur; differentiation of alternative paths.
17	-	Power BI	Different activity flow; inclusion of additional metrics in the generated model; without clear differentiation of alternative paths.
18	Heuristic Miner	ProM	Different activity flow; unstructured appearance model due to the inclusion of consecutive gateways.
19	Inductive Miner	ProM	Different activity flow; better perception of the order in which activities occur; differentiation of alternative paths.

5. DISCUSSION

The results described throughout this work focus on creating document management process models and their analysis and interpretation. Additionally, we evaluated and compared the performance of the process mining tools and algorithms in the construction of these models.

In this sense, when compared with other related works, such as Drakoulogkonas & Apostolou (2021), this research provides a differentiating and potentially more complete analysis, as it compares both factors, tools, and algorithms, rather than focusing on just one. In this regard, the table below summarizes the key characteristics of the tools that we analyzed during this work.

TABLE 4 - COMPARISON OF PROCESS MINING TOOLS (✓ = YES, **X** = NO)

Criteria	**PM4PY**	**ProM**	**Power BI**
License	Open-source.	Open-source.[3]	Free version.[4]
Documentation	Available.	Available.	There is information available for some visuals, but not as complete.
Integration of process mining algorithms	Alpha Miner: ✓ Heuristic Miner: ✓ Inductive Miner: ✓ Fuzzy Miner: **X**	Alpha Miner: ✓ Heuristic Miner: ✓ Inductive Miner: ✓ Fuzzy Miner: ✓	The algorithms used by each visual to generate the process models are not specified.
Integration of process mining techniques	Process Discovery: ✓ Conformance checking: ✓ Model enhancement: **X**	Process Discovery: ✓ Conformance checking: ✓ Model enhancement: ✓	Process Discovery: ✓ Conformance checking: **X** Model enhancement: **X**
Integration of metrics	✓	✓	✓
User-friendly tool	✓	**X**	✓

Considering the works already mentioned, we can conclude that there is almost a universal practice tool, namely ProM, and that the results are similar to those obtained in our study. Explicitly, both in the conclusions drawn from this work and in the research from ÇeliK & AkçetiN (2018), we see that ProM is considered a valuable tool but not

[3] However, some plug-ins may be required to be released under a different license than the one currently in use — the Lesser GNU Public License (L-GPL) —, which may not be open-source.

[4] It is important to highlight that some visuals for developing process mining models require subscription. For example, you must have a subscription to develop conformance checking analysis.

recommended for beginners due to its not very easy interface. Moreover, we did not find related works in this field that included the PM4PY or Power BI tools in their comparisons. Hence, one of the work's main contributions is the analysis of process models developed using tools that are not widely discussed in the literature.

The conclusions obtained concerning the algorithms can be compared with those obtained from the research stated in Saint et al. (2021). However, this study sought to analyze different algorithms from those evaluated in our work. In this regard, the conclusions to be derived were different. Even so, we can find some similarities in the performance of the standard algorithms, namely regarding the conclusions depicted for the Inductive Miner. As such, in both studies, this algorithm is seen as a model that preserves the soundness property, i.e., that truly reflects all possible paths and thus does not exclude infrequent behavior.

In Table 5 we seek to highlight the criteria that we evaluated throughout this work and that were decisive for the analysis of the performance of each algorithm. In this sense, we can compare the dimensions we analyzed, with the results obtained in Ajayi et al. (2019). However, the criteria used in each study differ, resulting in analyses with distinct scopes. Even so, we can conclude from our research that, in general, the Inductive Miner offers the best properties when compared to the other algorithms.

TABLE 5 - COMPARISON OF PROCESS MINING ALGORITHMS (✓ = Yes, **X** = No)

Criteria	**Alpha Miner algorithm**	**Heuristic Miner algorithm**	**Inductive Miner algorithm**
Produce sound models	**X**	**X**	✓
Deal with data incompleteness	**X**	Generally, the algorithm handled incomplete data well, but in Analysis 2 we had some cases with low-frequency activities where this did not happen.	✓
Produce good results with real-life logs	The algorithm most of the time did not produce adequate models with real-life logs.	The algorithm most of the time was able to produce adequate models with real-life logs (except some models in Analysis 2).	The algorithm was always able to produce adequate models with real-life logs.

6. CONCLUSIONS AND FUTURE WORK

The volume of data that companies must deal with has been growing exponentially. As such, these companies ought to be able to extract real value from these data. The challenge is not to collect more data but to understand existing data and be able to use it to improve performance. Process mining arises to help companies to deal with these data and realize the flow of the processes' events. Hence, this work aims to recognize to what extent process mining techniques can help a company to manage its document activities' flow. To this end, different algorithms and tools were used to extract the processes and, thus, identify which may be the most appropriate.

Considering what was explored throughout this work, the algorithm that showed the best results was the Inductive Miner. Unlike the remaining algorithms, this one did not exclude low-frequency activities. This is especially crucial when studying processes with few activities since little information is available. Furthermore, this algorithm offers a more realistic temporal perception as it allows us to understand the order of events. On the other hand, in some process models from other algorithms such as the Heuristic Miner, we see a set of events linked to the initial node that are dispersed over several regions of the model, which hinders the perception of the causality between events.

Regarding the tools, the considerations are not so clear and depend on the needs of each user. Given the scope of this work and the company's needs, the use of the PM4PY package through Python proved to be the most appropriate. As the remaining Python libraries can be integrated into the process mining analysis, the models can be created quickly and automatically for a variety of scopes. Thus, not only were we able to make a more complete analysis, but we were also able to optimize the time spent. Additionally, we can provide almost the same metrics and even complement the analysis by including several statistics, such as the event distribution over time and the rework activities, among others. Therefore, we recommend the other tools to be used as a complement to the analysis, rather than as the main resource.

The generated process models proved to be useful for understanding the flow of events and which activities and relationships were most critical and resource-consuming. These models were obtained through the application of different process discovery algorithms. Thus, for future work, it could also be interesting to extend these analyzes to

conformance checking techniques to check whether the generated process models conform with the event logs. These techniques are already developed in both PM4PY and ProM. In PM4PY this type of development can be done by employing algorithms regarding token replay and alignments, while in ProM we have to import the necessary plug-ins for that purpose.

One particularity about this work is that the project was in an early stage of adoption, so the volume of data, although already considerable, was not in massive order. However, storing all data in an excel sheet may be unfeasible for future phases. Thus, a suggestion for future work could be to try to investigate whether it would be possible to apply online learning procedures to this project, where the algorithm can use small batches of data for training instead of having to process all of the data at once.

Finally, as Genio is the company's main product, it could be equally useful to consider integrating these procedures into the tool. Since Genio creates customizable solutions for each client, it would probably be too complex to integrate process mining analyses for each process of each client. However, an alternative to consider could be to try to identify processes that are relatively generic and common to some customers, and for that, to suggest default processes models. Furthermore, it could also be of interest to provide the clients the ability to emit events from some actions, which could be used to generate process models for each of them.

To conclude, this work also resulted in the production of a paper that was presented at the 17th Iberian Conference on Technologies and Information Systems, and published in the respective proceedings (Parente & Costa, 2022).

References

Ajayi, L. K., Azeta, A. A., Owolabi, I. T., Damilola, O. O., Chidozie, F., Azeta, A. E., & Amosu, O. (2019). Current Trends in Workflow Mining. *Journal of Physics: Conference Series*, *1299*(1). https://doi.org/10.1088/1742-6596/1299/1/012036

Berti, A., & van der Aalst, W. M. (2019). Reviving Token-based Replay: Increasing Speed While Improving Diagnostics. *ATAED@Petri Nets/ACSD*.

Berti, A., Zelst, S. J., & van der Aalst, W. M. (2019). Process Mining for Python (PM4Py): Bridging the Gap Between Process- and Data Science. *ArXiv*, *abs/1905.06169*.

ÇeliK, U., & AkçetiN, E. (2018). Process Mining Tools Comparison. *AJIT-e Online Academic Journal of Information Technology*, *9*(34), 97–104. https://doi.org/10.5824/1309-1581.2018.4.007.x

Costa, C. J., & Aparicio, J. T. (2020). POST-DS: A Methodology to Boost Data Science. *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–6. https://doi.org/10.23919/CISTI49556.2020.9140932

Drakoulogkonas, P., & Apostolou, D. (2021). On the Selection of Process Mining Tools. *Electronics*, *10*(4), 451. https://doi.org/10.3390/electronics10040451

IEEE Standard for eXtensible Event Stream (XES) for Achieving Interoperability in Event Logs and Event Streams. (2016). *IEEE Std 1849-2016*, 1–50. https://doi.org/10.1109/IEEESTD.2016.7740858

Mans, R. S., van der Aalst, W. M., & Verbeek, H. M. (2014). Supporting Process Mining Workflows with RapidProM. *BPM*.

Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*.

Nafasa, P., Waspada, I., Bahtiar, N., & Wibowo, A. (2019). Implementation of Alpha Miner Algorithm in Process Mining Application Development for Online Learning Activities Based on MOODLE Event Log Data. *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, 1–6. https://doi.org/10.1109/ICICoS48119.2019.8982384

OMG. (2011). *Business Process Model and Notation (BPMN), Version 2.0*.

Parente, C., & Costa, C. J. (2022). Comparing Process Mining Tools and Algorithms. *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 1–7. https://doi.org/10.23919/CISTI54924.2022.9820570

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., & Cournapeau, D. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*

Rozinat, A., & van der Aalst, W. M. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems*, *33*, 64–95. https://doi.org/10.1016/j.is.2007.07.001

Saint, J., Fan, Y., Singh, S., Gasevic, D., & Pardo, A. (2021). Using process mining to analyse self-regulated learning: A systematic analysis of four algorithms. *LAK21: 11th International Learning Analytics and Knowledge Conference*, 333–343. https://doi.org/10.1145/3448139.3448171

van der Aalst, W. M. (1998). The application of Petri Nets to workflow management. *J. Circuits Syst. Comput.*, *08*, 21–66. https://doi.org/10.1142/S0218126698000043

van der Aalst, W. M. (2010). Process Discovery: Capturing the Invisible. *IEEE Comput. Intell. Mag.*, *5*, 28–41. https://doi.org/10.1109/MCI.2009.935307

van der Aalst, W. M. (2012). Process mining. *Communications of the ACM*, *55*, 76–83. https://doi.org/10.1145/2240236.2240257

van der Aalst, W. M. (2016). *Process Mining*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-49851-4

van der Aalst, W. M., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., van den Brand, P., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., de Leoni, M., … Wynn, M. (2012). Process Mining Manifesto. In F. Daniel, K. Barkaoui, & S. Dustdar (Eds.), *Business Process Management Workshops* (pp. 169–194). https://doi.org/10.1007/978-3-642-28108-2_19

van der Aalst, W. M., & Berti, A. (2020). Discovering Object-Centric Petri Nets. *Fundam. Informaticae*, *175*, 1–40.

van der Aalst, W. M., Reijers, H. A., Weijters, A. J., van Dongen, B. F., Medeiros, A. K., Song, M., & Verbeek, H. M. (2007). Business process mining: An industrial application. *Information Systems*, *32*, 713–732. https://doi.org/10.1016/j.is.2006.05.003

van Dongen, B. F., Medeiros, A. K., Verbeek, H. M., Weijters, A. J., & van der Aalst, W. M. (2005). The ProM Framework: A New Era in Process Mining Tool Support. In G. Ciardo & P. Darondeau (Eds.), *Applications and Theory of Petri Nets 2005* (Vol. 3536, pp. 444–454). Springer Berlin Heidelberg. https://doi.org/10.1007/11494744_25

Weijters, A. J., van der Aalst, W. M., & Medeiros, A. K. (2006). *Process mining with the HeuristicsMiner algorithm*.

APPENDICES

Appendix A – Process Models

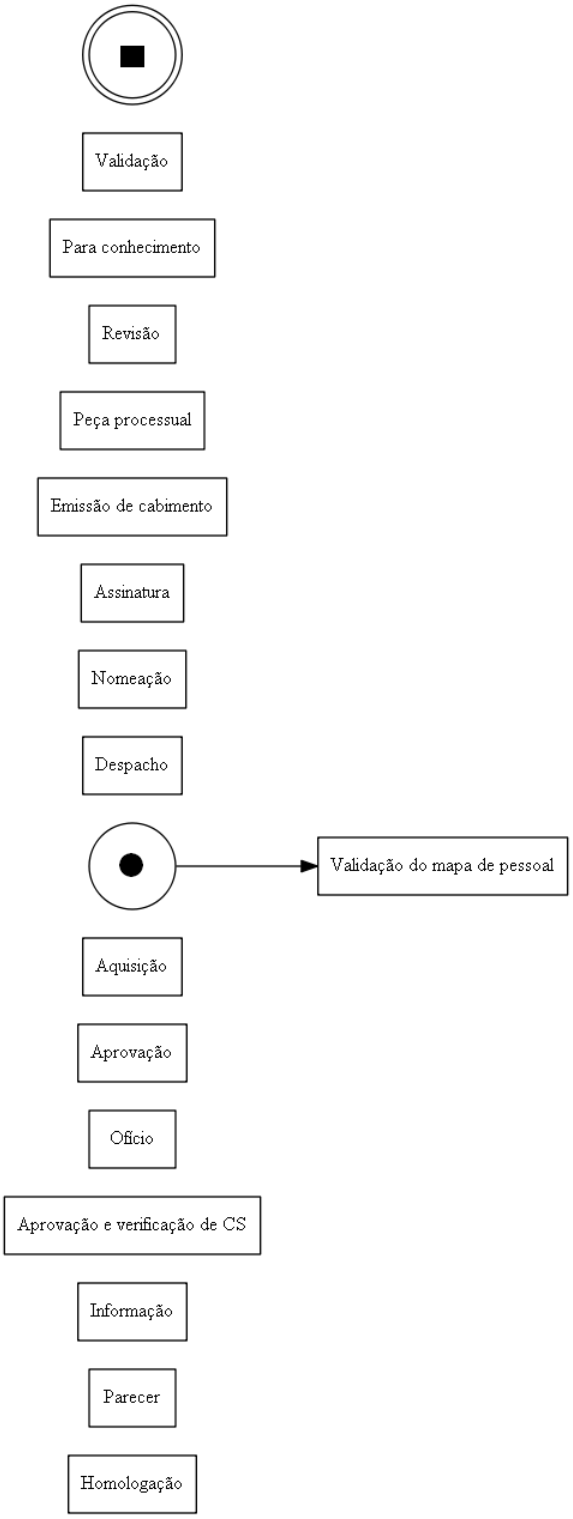

Figure 20 - Alpha Miner results using PM4PY within the scope of Analysis 1.

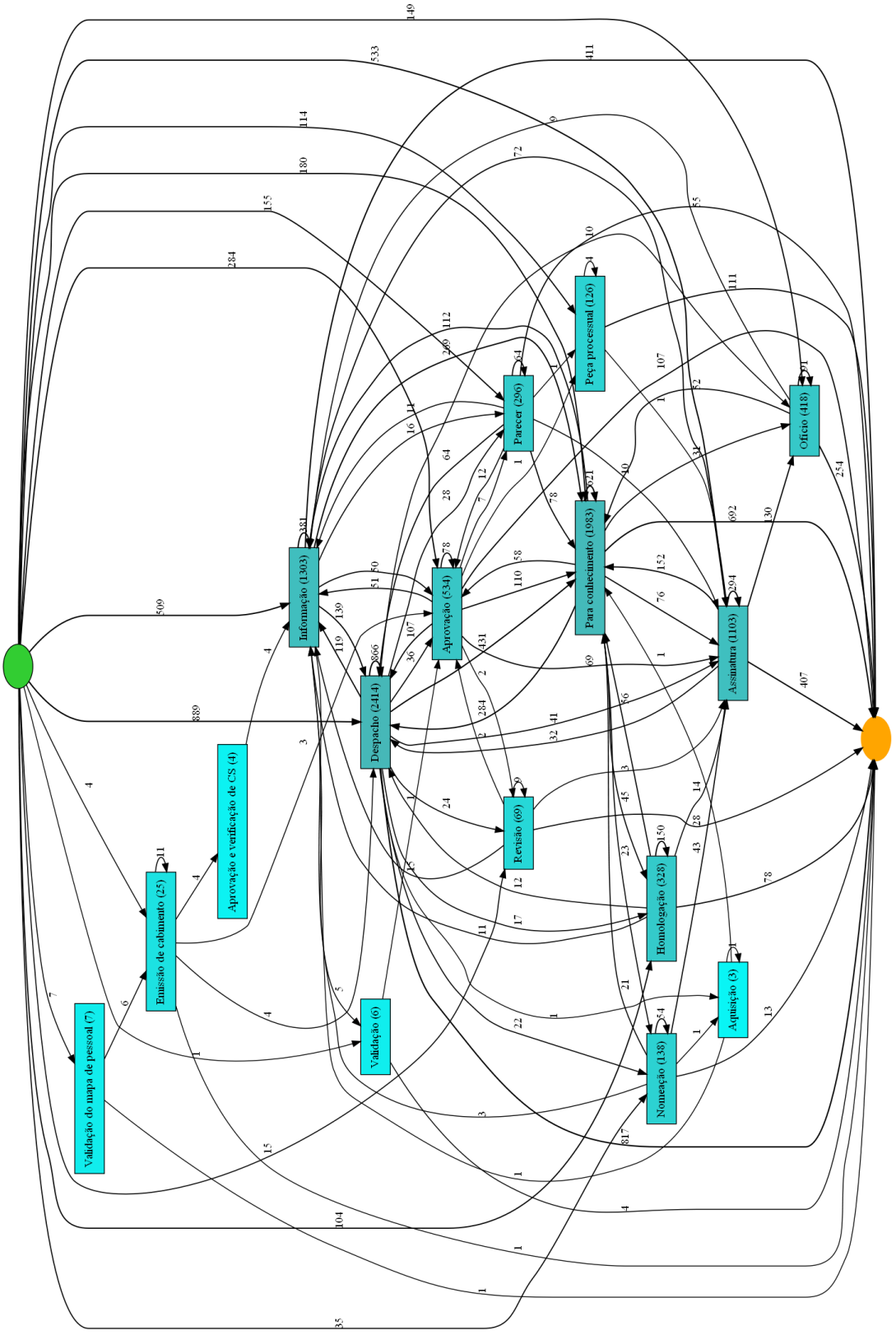

Figure 21 - Heuristic Miner results using PM4PY within the scope of Analysis 1 (dependency threshold = 0.5).

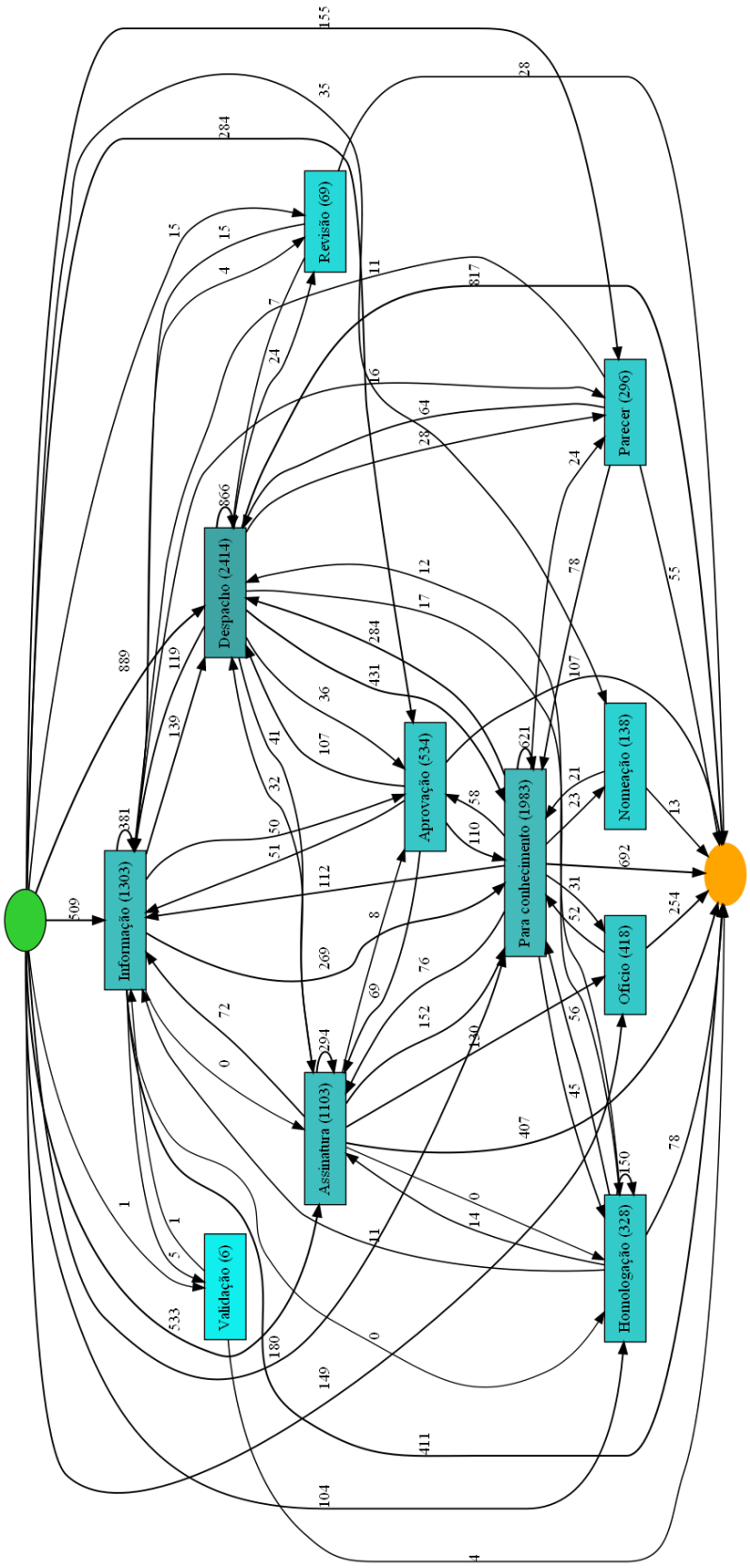

Figure 22 - Heuristic Miner results using PM4PY within the scope of Analysis 1 (dependency threshold = 0.99).

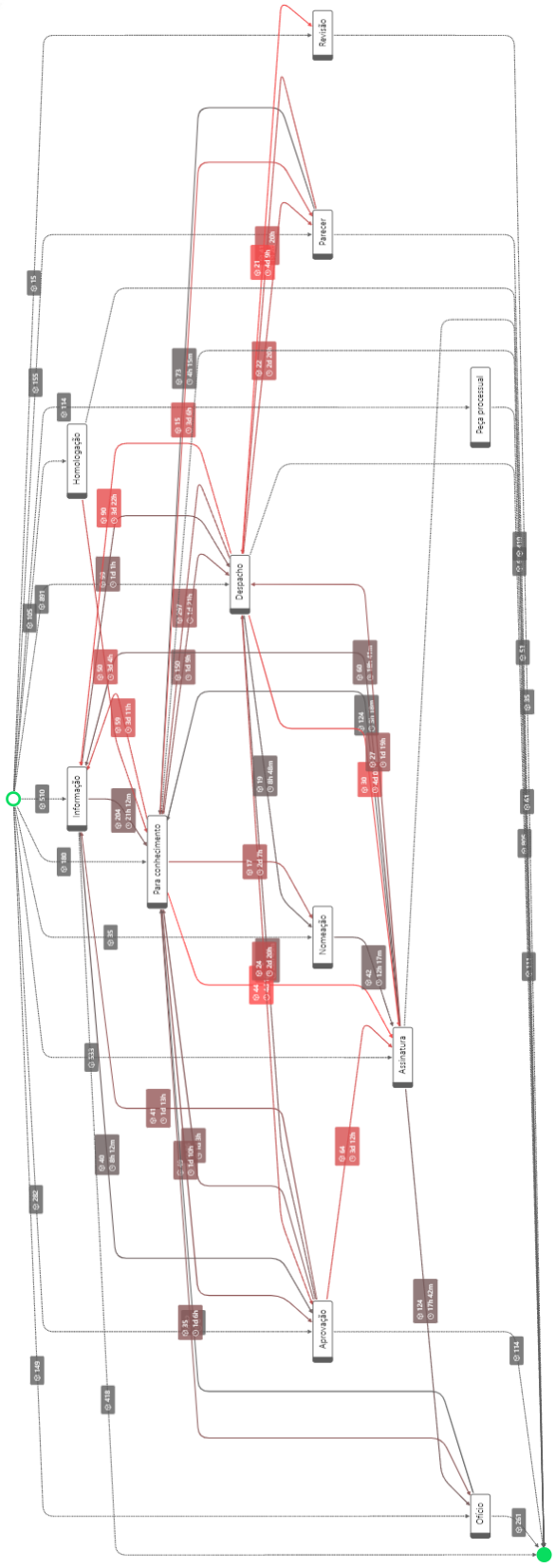

Figure 23 - PAFnow Process Mining results using Power BI within the scope of Analysis 1.

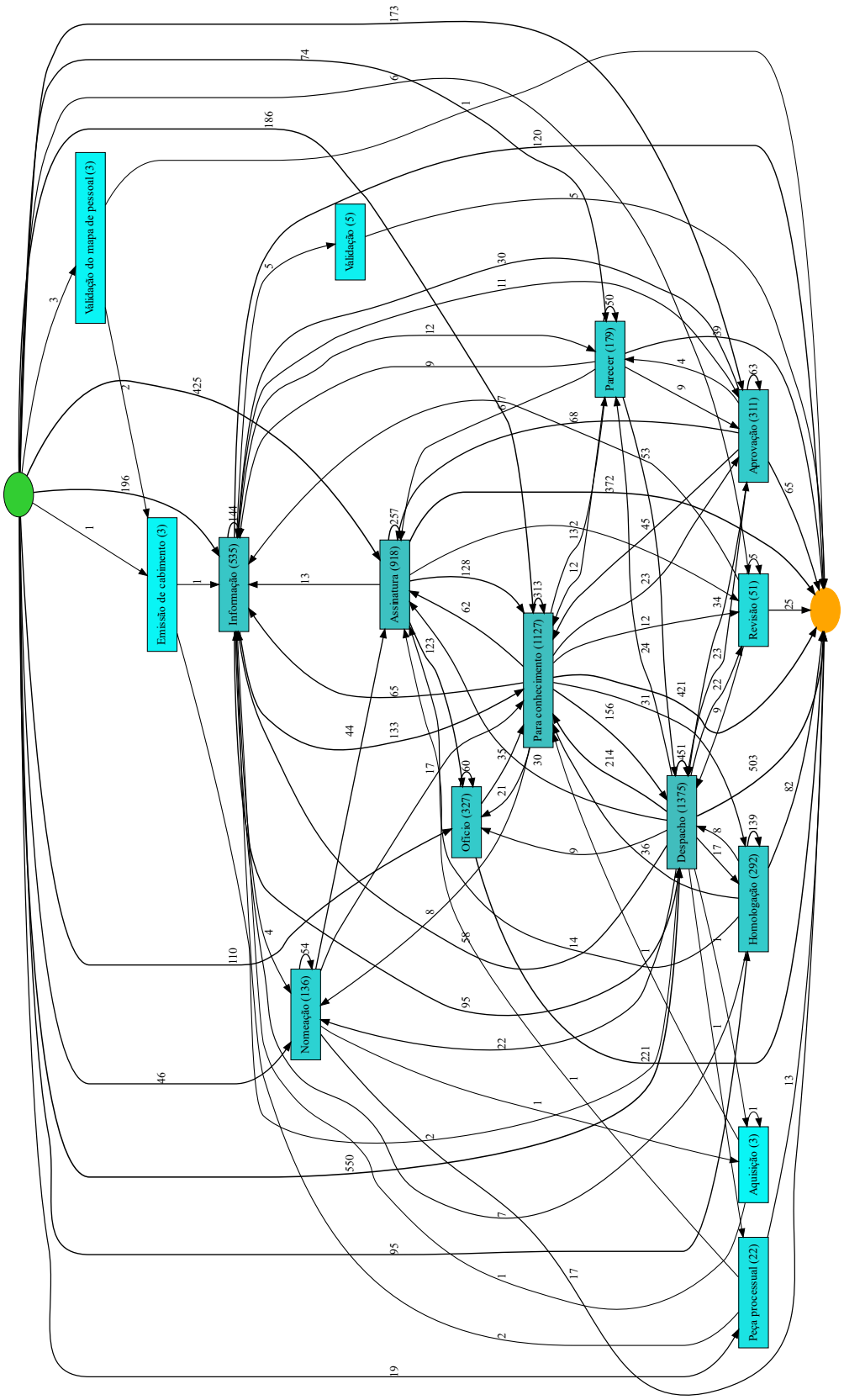

Figure 24 - Heuristic Miner results using PM4PY for the Organization 3.