ISEP INSTITUTO SUPERIOR
DE ENGENHARIA DO PORTO

[Smarter Alarm Management with LLMs]

João Sousa

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Software Engineering**

Orientador: Paulo Gandra Sousa
Supervisor: João Peixoto

Porto, 22 de novembro de 2025

Declaração de Integridade

Declaro ter conduzido este trabalho académico com integridade.

Não plagiei ou apliquei qualquer forma de uso indevido de informações ou falsificação de resultados ao longo do processo que levou à sua elaboração.

Portanto, o trabalho apresentado neste documento é original e de minha autoria, não tendo sido utilizado anteriormente para nenhum outro fim.

Declaro ainda que tenho pleno conhecimento do Código de Conduta Ética do P.PORTO.

ISEP, Porto, 22 de novembro de 2025

Dedicatória

The dedicatory is optional. Below is an example of a humorous dedication.

"To my wife Marganit and my children Ella Rose and Daniel Adam without whom this book would have been completed two years earlier."in "An Introduction To Algebraic Topology"by Joseph J. Rotman.

Resumo

This document explains the main formatting rules to apply to a TMDEI Master Dissertation work for the MSc in Computer Engineering of the Computer Engineering Department (DEI) of the School of Engineering (ISEP) of the Polytechnic of Porto (IPP).

The rules here presented are a set of recommended good practices for formatting the disseration work. Please note that this document does not have definite hard rules, and the discussion of these and other aspects of the development of the work should be discussed with the respective supervisor(s).

This document is based on a previous document prepared by Dr. Fátima Rodrigues (DEI/ISEP).

The abstract should usually not exceed 200 words, or one page. When the work is written in Portuguese, it should have an abstract in English.

Please define up to 6 keywords that better describe your work, in the *THESIS INFORMATION* block of the `main.tex` file.

Palavras-chave: Keyword1, ..., Keyword6

Abstract

Trabalhos escritos em língua Inglesa devem incluir um resumo alargado com cerca de 1000 palavras, ou duas páginas.

Se o trabalho estiver escrito em Português, este resumo deveria ser em língua Inglesa, com cerca de 200 palavras, ou uma página.

Para alterar a língua basta ir às configurações do documento no ficheiro `main.tex` e alterar para a língua desejada ('english' ou 'portuguese')[1]. Isto fará com que os cabeçalhos incluídos no template sejam traduzidos para a respetiva língua.

---

[1]Alterar a língua requer apagar alguns ficheiros temporários; O target **clean** do **Makefile** incluído pode ser utilizado para este propósito.

Agradecimentos

The optional Acknowledgment goes here... Below is an example of a humorous acknowledgment.

"I'd also like to thank the Van Allen belts for protecting us from the harmful solar wind, and the earth for being just the right distance from the sun for being conducive to life, and for the ability for water atoms to clump so efficiently, for pretty much the same reason. Finally, I'd like to thank every single one of my forebears for surviving long enough in this hostile world to procreate. Without any one of you, this book would not have been possible."in "The Woman Who Died a Lot"by Jasper Fforde.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

a	distance	m
P	power	W ($\mathrm{J\,s^{-1}}$)
ω	angular frequency	rad

Capítulo 1

Thesis Structure

1.1 Introduction

The goal of a Master dissertation is to document the work developed and highlight its importance. Firstly, the author should aim to follow the best practices in terms of writing style and organise the contents such that the message is clear. The language style used should be clear, and avoid colloquial expressions.

Please use an automated spell checker during the writing of your work.

This chapter provides some brief recommendations for the writing of your dissertation.

1.2 Structure

The structure of the document can be divided in three major parts: Introduction, Body and Conclusions.

The introduction includes, at least, a statement of what was the work developed; a brief, focused, state of the art; an explanation of how the work fits into the current state of the art, and how it contributes to it; a description of the structure of the document.

The Body of the dissertation should include a literature review (in one or more chapters) and one or more chapters that describe the work developed and the results, justifying them adequately.

The conclusions make a final balance of the work, highlighting the main aspects of the work and making critical judgments about what was accomplished, and providing suggestions for future work, if appropriate.

1.3 Formatting

The document can be written in Portuguese or English. The minimum number of pages is 60 and the maximum is 120 (not counting the Annexes). Small deviations are allowed. Please follow the margins and fonts defined in this template. The font size should be 11pt. The document should be printed double sided.

Note that the graphical aspect of the thesis is important, but does not replace a well-written and well-organised presentation of ideas.

Please refer to Chapter 2 and Chapter 3 for details about this template, how to format the document and insert citations, figures, tables, equations and other elements.

Capítulo 2

About LaTeX and How to Use This Template

2.1 Learning LaTeX

LaTeX is not a wysiwyg (What You See is What You Get) program, unlike word processors such as Microsoft Word or Apple's Pages. Instead, a document written for LaTeX is actually a simple, plain text file that contains *no formatting*. You tell LaTeX how you want the formatting in the finished document by writing in simple commands amongst the text, for example, if I want to use *italic text for emphasis*, I write the \emph{text} command and put the text I want in italics in between the curly braces. This means that LaTeX is a «mark-up» language, very much like HTML.

2.1.1 A (not so short) Introduction to LaTeX

If you are new to LaTeX, there is a very good eBook – freely available online as a PDF file – called, «The Not So Short Introduction to LaTeX». The book's title is typically shortened to just *lshort*. You can download the latest version (as it is occasionally updated) from here: http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf

It is also available in several other languages. Find yours from the list on this page: http://www.ctan.org/tex-archive/info/lshort/

It is recommended to take a little time out to learn how to use LaTeX by creating several, small 'test' documents, or having a close look at several templates on:
http://www.LaTeXTemplates.com
Making the effort now means you're not stuck learning the system when what you *really* need to be doing is writing your thesis.

2.1.2 A Short Math Guide for LaTeX

If your thesis is going to contain heavy mathematical content, be sure that LaTeX will make it look beautiful, even though it won't be able to solve the equations for you.

The «Not So Short Introduction to LaTeX» (available on http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdfCTAN) should tell you everything you need to know for most cases of typesetting mathematics. If you need more information, a much more thorough mathematical guide is available from the AMS called, «A Short Math Guide to LaTeX» and can be downloaded from: ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf

There are many different LATEX symbols to remember, luckily you can find the most common symbols in `http://ctan.org/pkg/comprehensive`The Comprehensive LATEX Symbol List.

You can write an equation, which is automatically given an equation number by LATEX like this:

```
\begin{equation}
E = mc^{2}
\label{eqn:Einstein}
\end{equation}
```

This will produce Einstein's famous energy-matter equivalence equation:

$$E = mc^2 \tag{2.1}$$

All equations you write (which are not in the middle of paragraph text) are automatically given equation numbers by LATEX. If you don't want a particular equation numbered, use the unnumbered form:

```
\[ a^{2}=4 \]
```

LATEX automatically adjusts the style of the math expressions according to the case if they are inline or not (so that they occupy more or less space), but you can set this explicitly. For instance, if you want an in-line mathematical element to display as a equation-like element put `\displaystyle` before that element. Use `\textstyle` for small, inline-like maths, before the command that generates the element.

2.1.3   Obtaining LATEX

The LATEX distribution is available for many systems including Windows, Linux and Mac OS X. Check the webpage for the LATEX project for more information: `https://latex-project.org/ftp.html`.

2.2   Getting Started with this Template

If you are familiar with LATEX, then you should explore the directory structure of the template and then proceed to place your own information into the *THESIS INFORMATION* block of the `main.tex` file. You can then modify the rest of this file to your unique specifications. Section 2.4 on page 7 will help you do this. Make sure you also read section 2.7 about thesis conventions to get the most out of this template.

If you are new to LATEX it is recommended that you carry on reading through the rest of the information in this document.

2.2.1   About this Template

This LATEX Thesis Template is originally based and created around a LATEX style file created by Steve R. Gunn from the University of Southampton (UK), department of Electronics and Computer Science. You can find his original thesis style file at his site, here: `http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/`

Steve's `ecsthesis.cls` was then taken by Sunil Patel who modified it by creating a skeleton framework and folder structure to place the thesis files in. The resulting template can be found on Sunil's site here: `http://www.sunilpatel.co.uk/thesis-template`

Sunil's template was made available through `http://www.LaTeXTemplates.com` where it was modified many times based on user requests and questions. Version 2.0 and onwards of this template represents a major modification to Sunil's template and is, in fact, hardly recognisable. The work to make version 2.0 possible was carried out by `mailto:vel@latextemplates.com` Vel and Johannes Böttcher.

Vel's and Böttcher template was adapted to fit the TMDEI/ISEP dissertation formatting specifications. This adaptation was done by Nuno Pereira (nap@isep.ipp.pt) and Paulo Baltarejo (pbs@isep.ipp.pt) in December 2015. It was also based on the specifications earlier defined by Fátima Rodrigues (mfc@isep.ipp.pt).

2.3 What this Template Includes

2.3.1 Folders

This template is composed of several files and folders. The folder names are mostly self-explanatory:

frontmatter – this is the folder holding the `frontmatter.tex` where are defined the dedicatory, abstract, acknowledgement, contents pages, list of figures, tables and others.

ch1, ch2, ... – these are the folders where you put the contents of each chapter. Each chapter should go in its own separate folder. Inside each chapter folder there is one (or more) **.tex** with the contents of the chapter, and an **assets** folder which contains the figures and other graphical elements of the chapter (such as algorithms, source code, plots, ...).

appendices – this is the folder where you put the appendices. Each appendix should go into its own separate `.tex` file. An example and template are included in the directory.

build – this is the folder where the output of building your document is put. Everytime you build your document, LATEX creates a number of auxiliary files that are written to this folder. These auxiliary files can be ignored or deleted as LATEX and BibTeX will regenerate them[1]. Note that the resulting `pdf` file is written to this folder and then copied to the root folder of the template.

2.3.2 Files

Included are also several files, most of them are plain text and you can see their contents in a text editor. After initial compilation, you will see that more auxiliary files are created by LATEX or BibTeX and which you don't need to delete or worry about[2]:

main-bibliography.bib – this is an important file that contains all the bibliographic information and references that you will be citing in the thesis for use with BibTeX. You can write it manually, but there are reference manager programs available that will create and manage it for you. Bibliographies in LATEX are a large subject and you may need to read about BibTeX

---

[1]Some changes require rebuilding temporary files again; actually, a document with citations may require up to three passes. For this reason, we recommend using the included **Makefile** which deals with such details

[2]Note that some changes (such as changing the language of the template require deleting some of these temporary files) the included **Makefile** has a **clean** rule to do this automatically

before starting with this (check, for example: `http://www.bibtex.org/Using/`). Many modern reference managers will allow you to export your references in BibTeX format which greatly eases the amount of work you have to do.

tmdei-style.cls – this is an important file. It is the class file that tells LATEX how to format the thesis.

main.pdf – this is your beautifully typeset thesis (in the PDF file format) created by LATEX. It is supplied in the PDF with the template and after you compile the template you should get an identical version.

main.tex – this is an important file. This is the file that you tell LATEX to compile to produce your thesis as a PDF file. It contains the framework and constructs that tell LATEX how to layout the thesis. It is heavily commented so you can read exactly what each line of code does and why it is there. After you put your own information into the *THESIS INFORMATION* block – you have now started your thesis!

Makefile – this is an important file. This file describes the steps that need to be taken to transform the **main.tex** into a PDF file. It uses the command **latexmk** which knows how to properly build a LATEX source file. To use it, open a command line and type **make** in the base folder of the template. You can also remove temporary files with **make clean**.

latexmk.rc – this file is required to appropriately build documents with glossaries using **latexmk**. Essentially, it tells **latexmk** that it also needs to run the command **makeglossaries** to build the document.

2.3.3   Building the Document

To build the document and create a `pdf` file from the LATEX source, you need to invoke several commands that will compile your document, bibliography, and glossaries. We recommend that you build the document automatically, using the **Makefile** provided.

The **Makefile** can be used by invoking the command **make**:

- `make` – Builds the document using the command **latexmk** which knows how to properly build a LATEX source file;

- `make clean` – Deletes auxiliary files using the command **latexmk** which knows the files created every time the document is built;

- `make clean-all` – Deletes all auxiliary files and bibliography cache.

Manually Building the Document

The general process of building the document is as follows. You can execute these commands to manually build your document. This can be convenient also for debug purposes, should you encounter any problem.

1. `pdflatex -aux-directory=build -output-directory=build main` – The command **pdflatex** writes all `\cite{...}` arguments to an auxiliary file.

2. `biber -input-directory=build -output-directory=build main` – The command **biber**, processes the previously created auxiliary file and the bibliography is produced.

3. `makeglossaries -d build main` – The command **makeglossaries**, goes through the document to process the \gls{...} commands.

4. `pdflatex -aux-directory=build -output-directory=build main` – **pdflatex** is run again to include the bibliography, and write the correct labels in the auxiliary file.

5. `pdflatex -aux-directory=build -output-directory=build main` – **pdflatex** is run once again, to include the labels and produce the final document.

The above commands should be executed in the root folder of the template. Note that some of the commands might be omitted if the necessary auxiliary files were constructed previously, and no changes that require the auxiliary files to be rebuilt were made.

2.4   Filling in Your Information in the `main.tex` File

You will need to personalise the thesis template and make it your own by filling in your own information. This is done by editing the `main.tex` file in a text editor or your favourite LaTeX environment.

Open the file and scroll down to the second large block titled *THESIS INFORMATION* where you can see the entries for *Thesis Title, Author*, etc . . .

Fill out the information about your work, yourself, your supervisor(s) and institution. You can also insert web links, if you do, make sure you use the full URL, including the `http://` for this. If you don't want these to be linked, simply remove the \href{url}{name} and only leave the name.

When you have done this, save the file and recompile `main.tex`. All the information you filled in should now be in the PDF, complete with web links. You can now begin your thesis proper!

2.5   The `main.tex` File Explained

The `main.tex` file contains the structure of the thesis. There are plenty of written comments that explain what pages, sections and formatting the LaTeX code is creating. Each major document element is divided into commented blocks with titles in all capitals to make it obvious what the following bit of code is doing. Initially there seems to be a lot of LaTeX code, but this is all formatting, and it has all been taken care of so you don't have to do it.

Begin by checking that your information on the title page is correct. It can be changed by editing the block titled *THESIS INFORMATION* in `main.tex`.

After the title page, the frontmatter is included. The frontmatter includes the dedicatory, abstract, acknowledgement, contents pages, list of figures and tables. The frontmatter is further detailed in Section 2.6.

Then, there is the block where the chapters are included. Uncomment the lines (delete the % character) as you write the chapters. Each chapter should be written in its own folder (ch1, ch2, etc. . .), in a file named `Chapter1.tex`, `Chapter2.tex`, etc. . . Similarly for the appendices, uncomment the lines as you need them. Each appendix should go into its own file and placed in the *appendices* folder.

Finally comes the bibliography. The bibliography style (called `authoryear-ibid`) is used for the bibliography as it is very similar to the recommended Harvard citation style. This is fully a featured style that will even include links to where the referenced paper can be found online. Do not underestimate how grateful your reader will be to find that a reference to a paper is just a click away. Of course, this relies on you putting the URL information into the BibTeX file in the first place.

2.6   Front Matter (`frontmatter/frontmatter.tex`)

The frontmatter starts with the Dedication Section, where you can insert your dedication. Who will you dedicate your thesis to?

Following this, the abstract page summarises your work in a condensed way and can almost be used as a standalone document to describe what you have done. The text you write will cause the heading to move up so don't worry about running out of space. All dissertations should include abstracts in English and Portuguese.

Next come the acknowledgements. On this page, write about all the people who you wish to thank (not forgetting parents, and consider including partners and/or your advisor(s)/supervisor(s)).

The contents pages, list of figures, tables, algorithms and source code are all taken care of for you and do not need to be manually created or edited. Any of the commands in this section must be removed (or commented) if there are no elements in the document to list (for example, if there are no algorithms in the document, remove the `\listofalgorithms` command; if there are no source code listings, remove the `\listoflisting` command).

Next you have a list of abbreviations used in the thesis, then a list of the physical constants and numbers you refer to and finally, a list of mathematical symbols used in any formulae. You can omit these tables, but making the effort to fill these tables means the reader has a one-stop place to refer to instead of searching the internet and references to try and find out what you meant by certain abbreviations or symbols.

The list of symbols is split into the Roman and Greek alphabets. Whereas the abbreviations and symbols ought to be listed in alphabetical order (and this is *not* done automatically for you) the list of physical constants should be grouped into similar themes.

2.7   Thesis Features and Conventions

To get the best out of this template, there are a few conventions that you may want to follow.

One of the most important (and most difficult) things to keep track of in such a long document as a thesis is consistency. Using certain conventions and ways of doing things (such as using a Todo list) makes the job easier. Of course, all of these are optional and you can adopt your own method.

2.7.1   Printing Format

This thesis template is designed for double sided printing (i.e. content on the front and back of pages) as most theses are printed and bound this way. Switching to one sided printing is

as simple as uncommenting the *oneside* option of the `documentclass` command at the top of the `main.tex` file. You may wish to use this option for drafting purposes, but remember that the final document should be double sided.

The headers for the pages contain the page number on the outer side (so it is easy to flick through to the page you want) and the chapter name on the inner side.

The text is set to 11 point by default with single line spacing, again, you can tune the text size and spacing should you want or need to using the options at the very start of `main.tex`. The spacing can be changed similarly by replacing the *singlespacing* with *onehalfspacing* or *doublespacing*. Remember again, that the default spacing and font size should be used for the final document.

2.7.2 References

The `biblatex` package is used to format the bibliography and inserts references such as this one (**Reference1**). The options used in the `main.tex` file mean that the in-text citations of references are formatted with the author(s) listed with the date of the publication. Multiple references are separated by semicolons (e.g. (**Reference2**; **Reference1**)) and references with more than three authors only show the first author with *et al.* indicating there are more authors (e.g. (**Reference3**)). This is done automatically for you. To see how you use references, have a look at the `Chapter2.tex` source file. Many reference managers allow you to simply drag the reference into the document as you type.

Scientific references should come *before* the punctuation mark if there is one (such as a comma or period). The same goes for footnotes[3]. You can change this but the most important thing is to keep the convention consistent throughout the document. Footnotes themselves should be full, descriptive sentences (beginning with a capital letter and ending with a full stop).

The bibliography is typeset with references listed in alphabetical order by the first author's last name. This is similar to the Harvard referencing style. To see how LaTeX typesets the bibliography, have a look at the very end of this document (or just click on the reference number links in in-text citations).

More citation examples

1. A citation command \parencite{Reference1} results in a citation in parentheses: (**Reference1**).

2. A citation command \textcite{Reference2}, can used in the flow of text: As **Reference2** said . . .

3. A citation command \autocite{Reference3} automatically switches style depending on location and the option setting in the package declaration. In this case, it produces a citation in parentheses: (**Reference3**).

A Note on bibtex

By default, the template uses **biber** as the backend to process references, instead of **bibtex**. This is because **bibtex** does not correctly handle unicode character encoding (i.e.

---

[3]Such as this footnote, here down at the bottom of the page.

Tabela 2.1: The effects of treatments X and Y on the four groups studied.

Groups	Treatment X	Treatment Y
1	0.2	0.8
2	0.17	0.7
3	0.24	0.75
4	0.68	0.3

"non-english"characters). You can change this back to **bibtex** by finding this in `main.tex`: *backend=biber* and changing it to *backend=bibtex*. You will then need to delete all auxiliary files and navigate to the template directory in your terminal (command prompt). Once there, simply type `bibtext main` to compile your bibliography. You can then compile `main.tex` as normal and your bibliography will be updated. Alternatively, you can invoke `make clean; make`.

2.7.3 Tables

Tables are an important way of displaying your results, Table 2.1 is an example table which was generated with this code:

```
\begin{table}
\caption{The effects of treatments X and Y on the four groups studied.}
\label{tab:treatments}
\centering
\begin{tabular}{l l l}
\toprule
\tabhead{Groups} & \tabhead{Treatment X} & \tabhead{Treatment Y} \\
\midrule
1 & 0.2 & 0.8\\
2 & 0.17 & 0.7\\
3 & 0.24 & 0.75\\
4 & 0.68 & 0.3\\
\bottomrule\\
\end{tabular}
\end{table}
```

You can reference tables with `\ref{<label>}` where the label is defined within the table environment. See `Chapter2.tex` for an example of the label and citation (e.g. Table 2.1).

2.7.4 Figures

There will hopefully be many figures in your thesis (that should be placed in the **assets** folder of the repective chapter). The way to insert figures into your thesis is to use a code template like this:

```
\begin{figure}
\centering
\includegraphics{ch2/assets/electron}
\decoRule
\caption[An Electron]{An electron (artist's impression).}
```

```
\label{fig:Electron}
\end{figure}
```

Also look in the source file. Note that the path is relative to the `main.tex` file location. Putting this code into the source file produces the picture of the electron that you can see in Figure 2.1.

Sometimes figures don't always appear where you specify in the source. The placement depends on how much space there is on the page for the figure. Sometimes there is not enough room to fit a figure directly where it should go (in relation to the text) and so LATEX puts it at the top of the next page. In this case, the argument ("[b]") of the `\begin{figure}[b]` command indicates that the Figure 2.1 should be on the bottom (for example, "t"would be used to place the figure on top). Figures consistently placed on top or bottom of the page usually make life easier for the reader. Note that positioning figures is the job of LATEX and so you should focus on making them look good!

Figures should have captions (such as in Figure 2.1). The `\caption` command contains two parts, the first part, inside the square brackets is the title that will appear in the *List of Figures*, and so should be short. The second part in the curly brackets should contain the longer and more descriptive caption text.

You should also put a label (such as `\label{fig:electron}`) in all your figures so you can refer to them (for example with `\ref{fig:electron}`). **All graphical elements in your document must have accompanying text (a paragraph, or a sentence) describing it!**

Another example illustration is included in Figure 2.2, which shows a comic about thesis writing.

LATEX is capable of using images in pdf, jpg and png format.

2.7.5 Other Useful Commands

There are some additional commands created to keep the formatting separated from the content, which we document below.

keyword – Used to highlight keywords. For example, this is **somekeyword** (the command that produces this output is `\keyword{somekeyword}`).

tabhead – Used to highlight table headers.

Figura 2.1: An electron (artist's impression).

Figura 2.2: A Thesis Writing Comic (from www.phdcomics.com).

code – Used to output code. For example, this `statement is formatted as code` (the command that produces this output is `\code{...}`).

file – Used to highlight files. For example, this is `somefilename` (the command that produces this output is `\file{somefilename}`).

option – Used to highlight options. For example, this is *someoption* (the command that produces this output is `\option{someoption}`).

2.8   Sectioning and Subsectioning

You should break your thesis up into nice, bite-sized sections and subsections. LATEX automatically builds a table of Contents by looking at all the `\chapter{}`, `\section{}` and `\subsection{}` commands you write in the source.

The Table of Contents should only list the sections to three (3) levels. A `chapter{}` is level zero (0). A `\section{}` is level one (1) and so a `\subsection{}` is level two (2). In your thesis it is likely that you will even use a `subsubsection{}`, which is level three (3). The depth to which the Table of Contents is formatted is set within `tmdei-style.cls`. If you need this changed, you can do it in `main.tex`.

2.9   In Closing

You have reached the end of this mini-guide. You can now rename or overwrite this pdf file and begin writing your own `.tex` files and the rest of your thesis. The easy work of setting up the structure and framework has been taken care of for you. It's now your job to fill it out!

Good luck and have lots of fun!

Guide written by —
Sunil Patel: `http://www.sunilpatel.co.uk`www.sunilpatel.co.uk
Vel: `http://www.LaTeXTemplates.com`LaTeXTemplates.com

Capítulo 3

Algorithms, Source Code, the Portable Graphics Format and Acronyms

3.1 Algorithms

LATEX has several packages for typesetting algorithms in form of "pseudocode". In this template, we suggest the use of the `algorithm` environment with the `algpseudocode` package. More information about algorithms can be found at `https://en.wikibooks.org/wiki/LaTeX/Algorithms`.

Algorithm 3.1 shows Euclid's algorithm that computes Greatest Common Divisor (GCD) of two integer numbers.

Here it is the LATEX text for the "pseudocode" algorithm presented in Algorithm 3.1.

```
\begin{algorithm}
\caption{Euclid's algorithm (pseudocode)}
\label{alg:euclid}
\begin{algorithmic}[1]
\scriptsize

\State \textbf{Input}: Two integer numbers, $a$ and $b$
\State \textbf{Output}:  Greatest Common Divisor (GCD) of $a$ and $b$
\State
\Procedure{euclid}{$a,b$}\Comment{The GCD of $a$ and $b$}
\State $r\gets a\bmod b$
```

Algorithm 3.1 Euclid's algorithm

1: **Input**: Two integer numbers, a and b
2: **Output**: GCD of a and b
3:
4: **procedure** Euclid(a, b)   ▷ The GCD of a and b
5:     $r \leftarrow a \bmod b$
6:     **while** $r \neq 0$ **do**   ▷ We have the answer if r is 0
7:         $a \leftarrow b$
8:         $b \leftarrow r$
9:         $r \leftarrow a \bmod b$
10:    **end while**
11:    **return** b ▷ The GCD is b
12: **end procedure**

```
\While{$r\not=0$}\Comment{We have the answer if $r$ is $0$}
\State $a\gets b$
\State $b\gets r$
\State $r\gets a\bmod b$
\EndWhile
\State \textbf{return} $b$\Comment{The GCD is $b$}
\EndProcedure

\end{algorithmic}
\end{algorithm}
```

\listofalgorithms command generates a list of all algorithms. This command is called in the frontmatter.tex file. Therefore, if there is no algorithm in the thesis, this command must be removed (or commented) from such file.

3.2   Source Code

Sometimes there is the need to present programming source code snippets. The listings package is a powerful way to get nice source code highlighting in LaTeX. It supports various programming languages, like Java (selected as the default language in this template), C, and many others.

Listing 3.1 and Listing 3.2 show the source code of the Euclid's algorithm, written in Java and C, respectively.

```java
1  //valid for positive integers.
2  public static int euclid(int a, int b)
3  {
4     int r = a % b;
5     while(r != 0)
6     {
7        a = b;
8        b = r;
9        r = a % b;
10    }
11    return b;
12 }
```

Listing 3.1: Euclid's algorithm (Java).

```c
//valid for positive integers.
int euclid(int a, int b)
{
   int r = a % b;
   while(r != 0)
   {
      a = b;
      b = r;
      r = a % b;
   }
   return b;
}
```

Listing 3.2: Euclid's algorithm (C).

Here it is the LATEX text for both listings. Note that we encapsulate the listings inside a \minipage so that the listing does note break across pages. Using the \lstinputlisting command, the source code must be written in a separate file. In these two cases, both files are in ch3\assets\ directory.

```
\begin{minipage}{\linewidth}
\lstinputlisting [caption=Euclid's algorithm (Java).,
label=lst:euclid_java]
{ch3/assets/euclid.java}
\end{minipage}


\begin{center}
\begin{minipage}{0.7\linewidth}
\lstinputlisting [language=C,
caption=Euclid's algorithm (C).,
label=lst:euclid_c,
numbers=none]
{ch3/assets/euclid.c}
\end{minipage}
\end{center}
```

As it can be seen from the text above, there are a lot of parameters that can be specified, like programming language (language), numbering, etc. More information about listings can be found at https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings and http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf.

\listoflisting command generates a list of all source code listings. This command is called in the frontmatter.tex file. Therefore, if there are no listings in the thesis, this command must be removed (or commented) from such file.

3.3 The Portable Graphics Format

The Portable Graphics Format (PGF) and a number of packages built on top of PGF (such as TiKZ and PGFPLOTS) enable producing high quality graphical elements for your document.

3.3.1 TiKZ

TikZ is built on top of PGF and allows you to create sophisticated graphics using LATEX commands. According to its author, Till Tantau [1], "*What is TikZ? Basically, it just defines a number of TEX commands that draw graphics.*". With TikZ it is possible to accurately position picture elements, use LATEX fonts, incorporate mathematical typesetting, and use other LATEX features in your drawings.

The TikZ package defines the tikzpicture environment that is required to draw a graphic. This environment must be inserted into a figure environment when numbering and caption are required. Figure 3.1 shows a simple use of TiKZ, for which the LATEX source is as follows.

```
\begin{figure}[t]
\centering
```

---

[1] Available online at ftp://ftp.di.uminho.pt/pub/ctan/graphics/pgf/base/doc/pgfmanual.pdf

```
\begin{tikzpicture}
% Define four points
\coordinate (P0) at (1,0);
\coordinate (P1) at (0,1);
\coordinate (P2) at (-1,0);
\coordinate (P3) at (0,-1);
% Draw the diamond
\draw (P0)--(P1)--(P2)--(P3)--cycle;
\end{tikzpicture}

\caption{Using TiKZ for drawing pictures.}
\label{fig:tikz}
\end{figure}
```
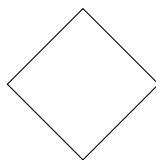
Figura 3.1: Using TiKZ for drawing pictures.

A great amount of examples are available at `http://www.texample.net/tikz/examples/`.
More information about TiKZ can be found at `https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ` and `ftp://ftp.di.uminho.pt/pub/ctan/graphics/pgf/base/doc/pgfmanual.pdf`.

3.3.2 PGFPLOTS

PGFPLOTS provides tools to draw high quality plots, and is based on TiKZ. To use PGF-PLOTS in the thesis you need to use `\usepackage{pgfplots}` (in `main.tex`). To guarantee compatibility you need to specify `\pgfplotsset{compat=<version>}`. You can choose the `version` (). In this case, it is recommended to choose `newest`. The choice `compat=newest` means "I do not care if my old figures change in appearance after the next version upgrade".

Here it is the LaTeX text for create the graph presented in Figure 3.2.

```
\begin{tikzpicture}
 \begin{axis}[ height=9cm, width=9cm, grid=major, ]
  \addplot {-x^5 - 242};
  \addlegendentry{model}
  \addplot coordinates {
   (-4.77778,2027.60977)
   (-3.55556,347.84069)
   (-2.33333,22.58953)
   (-1.11111,-493.50066)
   (0.11111,46.66082)
   (1.33333,-205.56286)
   (2.55556,-341.40638)
   (3.77778,-1169.24780)
```

```
  (5.00000,-3269.56775)
 };
 \addlegendentry{estimate}
\end{axis}
\end{tikzpicture}
```

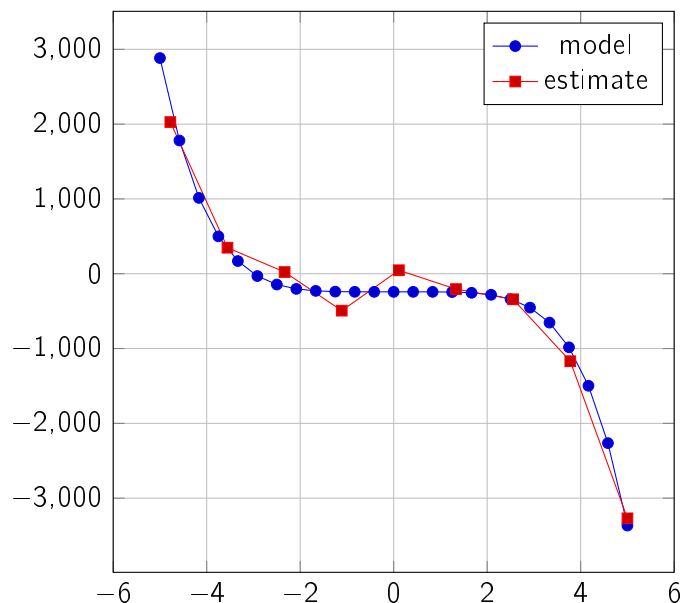Figure 3.2 shows an example of a graph created using PGFPLOTS functions.

Figura 3.2: Using PGFPLOTS for drawing a graph.

A great amount of examples are available at `http://pgfplots.sourceforge.net/gallery.html`. More information about TiKZ can be found at `http://pgfplots.sourceforge.net/pgfplots.pdf`.

3.4   Handling Acronyms Automatically

When writing a thesis you need to define acronyms. According to Wikipedia [2] *"An acronym is an abbreviation used as a word which is formed from the initial components in a phrase or a word."* and *"Acronyms are used most often to abbreviate names of organizations and long or frequently referenced terms."*. Typically, an acronym is a pronounceable word, which may already exist or it can be an invented word.

The use of acronyms imposes two rules: (i) an acronym must be defined in the text during the first appearance of the phrase or word and (ii) the document must have a list of all acronyms alphabetically sorted. In LATEX this is provided by a package called `\usepackage{glossaries}` that simplifies the use of acronyms.

Included in this thesis template there is a file called `glossary.tex` (in folder `frontmatter`), where all acronyms must be written in the form:

`\newacronym{label}{abbrv}{full}`

---

[2]Accessed in 16 of December of 2015

where `label` is the unique label identifying the acronym, `abbrv` is the abbreviated form of the acronym and `full` is the expanded text (word or phrase). This is an example of defining three acronyms:

```
\newacronym{RTS}{RTS}{Real-Time System}
\newacronym{GPOS}{GPOS}{General Purpose Operating System}
\newacronym{RTOS}{RTOS}{Real-Time Operating System}
```

In order to use the features of the `\usepackage{glossaries}`, you have only to use `\gls{label}` command in the text. Using this command the acronym will be defined in the first appearance in the text and it will be listed in a list. For instance, writing this LATEX text:

```
Linux is not a \gls{RTOS} but it is a \gls{GPOS}.
VxWorks is a \gls{RTOS}, so it is not a \gls{GPOS}.
```

outputs the following text:

Linux is not a Real-Time Operating System (RTOS) but it is a General Purpose Operating System (GPOS). VxWorks is a RTOS, so it is not a GPOS.

More information about acronyms can be found at `https://en.wikibooks.org/wiki/LaTeX/Glossary`.

Capítulo 4

Introdução

Este capítulo apresenta a organização responsável pela ideia do tema a ser desenvolvido, o contexto do seu surgimento e os problemas que a originaram. São abordados os objetivos, o planeamento e as considerações éticas tidas em conta para executar a proposta. Por último é descrita a estrutura deste documento de forma a facilitar a leitura do mesmo.

4.1 Organização e Contexto

A Critical Techworks (CTW) nasceu de um empreendimento conjunto entre a Critical Software e o grupo BMW (Bayerische Motoren Werke). Estabelecida em 2018 esta empresa desenvolve software exclusivamente para os produtos da BMW. A sua missão é a de "transformar a forma como o mundo se move"focando-se para isso na transformação digital com as melhores práticas do mercado.

A organização estrutura-se em equipas ágeis multidisciplinares, agrupadas em unidades e domínios tecnológicos, o que promove autonomia, ciclos de desenvolvimento rápidos e colaboração contínua. Esta abordagem permite alinhar o desenvolvimento de software com necessidades reais dos veículos e serviços digitais da BMW.

O presente relatório foi desenvolvido no âmbito da unidade curricular de Preparação para Dissertação (PREPD) do Instituto Superior de Engenharia do Porto (ISEP), com o intuito de demonstrar o estudo de viabilidade de uma solução para o problema em causa.

4.2 Problema

A análise e resolução de incidentes está no dia a dia das equipas que lidam com projetos potencialmente críticos. Um incidente pode ter por origem duas principais fontes: automático, através de métricas definidas pelas equipas; e manuais, criados por clientes das aplicações desenvolvidas. Um registo de incidente (ticket) é um documento, normalmente gerido por um sistema externo (como *ServiceNow* ou *Jira Service Management*), onde é possível visualizar detalhes do incidente, tais como a sua descrição e prioridade, e onde um elemento da equipa pode descrever os passos de resolução efetuados.

Equipas destes sistemas criticos lidam com muitos incidentes diariamente e muitas vezes fora de horários. - Quantidade de tickets pode ser demasiada - Analise pode ser descuidada, principalmente para o caso de acontecer de madrugada

4.3   Objetivos

4.4   Planeamento

Gant chart...

4.5   Considerações Éticas

4.6   Estrutura do Documento

Capítulo 5

Estado da Arte

5.1 Revisão de Literatura

5.1.1 Metodologia de Investigação

5.1.2 Resultados e Análise

5.2 Tecnologias Backend

Nada para já...

5.3 Tecnologias Frontend

Nada para já...

5.4 Processamento de Linguagem Natural

5.5 Sistemas Externos

5.5.1 Logs

5.5.2 Metricas

Apêndice A

Appendix Title Here

Write your Appendix content here.