

The ist-thesis class

Unofficial \LaTeX class for Instituto Superior Técnico

Daniel de Schiffart

with support from

João Gonçalves

João Lourenço

<https://github.com/ekspek/ist-thesis>

Summary

This class aims to prepare and format any \LaTeX document as a Master's thesis according to Instituto Superior Técnico's regulations.

Document translated to English by João Lourenço

Contents

Contents	2
1 Quick Start – The Basics	5
2 Optional Arguments	6
3 Commands and Environments	7
3.1 Cover page and Thesis' Data	7
3.1.1 Sample Usage	8
3.2 Special sections	8
3.2.1 Sample Usage	9
3.3 Internal Commands	10
3.4 Full list of commands and environments.	10
4 Class Implementation	12
4.1 Page style	12
4.2 Colors and Hyperlinks	12
4.3 Fonts	13
4.3.1 Main font	13
4.3.2 <i>Monospacing</i>	13
4.3.3 Math typesetting	13
4.3.4 <i>Serif</i> fonts	14
4.3.5 Showcases	14
4.4 Used <i>Packages</i>	14

Introduction

I must leave a few words of mine right here.

I learned how to use \LaTeX two years ago, at the beginning of my 4th grade, and as expected, it was by writing reports to Técnico's courses. After having to spend several hours writing reports for Telecommunications and Digital Systems Design, I took a liking to it, and have been learning bit by bit the limits of what \LaTeX is capable of (which, *spoiler alert*, are a lot). Watching the semesters fly by and getting ready to carry Técnico on my back, I wanted to leave something behind written in \LaTeX for my fellow younger peers. A notebook, problems solved, anything. Time passed and laziness made me postpone any motivation for this, and the last thing anyone wants to hear about in vacation is a finished course. Therefore, I started the only thing that made sense as the end approached: spend time with the Thesis' model several months before it even started.

This project started at the end of last year as a hobby. I was starting to get used to \LaTeX , and began to notice how much certain *packages* and definitions were being reused in every report. Since \LaTeX 's classes already defined tons of things when they were loaded, I started to take that into account, looking at the classes online. I never thought I'd be able to write a class from scratch - however, they really just contain \LaTeX code. So, after some setbacks, my code started to slowly appear.

Much of this work contains knowledge learned by writing the class [ist-relatorio](#), a much more experimental project, which still proves itself to be useful when I need it. Although less versatile, this class posed several challenges in the interface between the user and the *output*, but I think that, mostly, the final product looks good, whilst still providing some versatility. In fact, to prove that versatility, I decided to implement this very documentation using the class, just to see if it worked well. I know that abusing *Arial* in every page burns the eyes, but Técnico is the one in charge `~_('_')_/\``.

The main inspiration of moving to the thesis was not only due to my deadline to deliver it, which made me reduce it more and more, but also due to observing the many thesis models available on the web, for many different universities around the world. It didn't make sense seeing Técnico standing behind, specially considering the *huge huge* technological advances in our little corner of the University of Lisbon. Special shoutouts to the TUDelft's thesis model, based on their *Huisstijl*, which I read and re-read to learn how to create classes.

Obviously, much of my inspiration was also drawn from Professor André Marta's thesis model, who still holds the *de facto* Técnico's \LaTeX model. I admire his availability to share his code without restrictions, as well how well put together it is - so much so that I considered completely dropping my idea and just use his class. I don't want in any way to compete with Professor Marta's project - I just believe there can be two different approaches to the same problem. Check out [his project](#) and make your choice.

There is not much more to add. Take a look at the final product, contact me if needed, share the project, contribute if you can, spread the word if you like what is made here, or leave suggestions to what could improve.

Good luck with your thesis!

Daniel Lopes de Schiffart

Alumnus 81479
February 2019

Chapter 1

Quick Start – The Basics

To load the class and create a document that looks like a thesis, you have to download the class file, `ist-thesis.cls`. After that, in a separate `.tex` file, the first line of the document has to be

```
\documentclass{ist-thesis}
```

This loads the class. Write any text and compile to test it. The smallest possible valid `.tex` document that compiles with no problems can be found below:

```
1 \documentclass{ist-thesis}
2
3 \begin{document}
4
5 Hello World!
6
7 \end{document}
```

More properties of this class can be found in the next chapters.

Chapter 2

Optional Arguments

When loading the class, with the command `\documentclass`, it's possible to give optional arguments to define global options. These arguments are listed here.

portuguese Sets the document's language to Portuguese, mainly the syntax of macros and keywords, such as índice (table of contents) and resumo (summary). This option is selected by default, and as such does not need to be set.

english Sets the document's language as English. Similar to the portuguese option.

bw Enables the black and white option. Replaces all occurrences of cyan on the document, such as in links and URLs.

origmath Returns the math font initially loaded by \LaTeX . Follows from the discussion on section 4.3.3, and avoids loading the package `newtxsf`.

To enable one or more arguments, insert at the beginning of the file, between square brackets, the arguments to use.

```
\documentclass[english]{ist-thesis}
```

To load more than one argument, split them by commas.

```
\documentclass[english,origmath]{ist-thesis}
```

Chapter 3

Commands and Environments

This chapter contains the official documentation for usable commands and environments implemented on the class, as well as their functions and dependencies inside the document. Also included are some modifications made to base \LaTeX documents, so they work correctly with the implemented functions.

These commands can be split into three categories: used for the cover page and document's information, on the section [3.1](#); used for the structure and special sections, on the section [3.2](#); and three internal commands defined and used by the class, on the section [3.3](#).

3.1 Cover page and Thesis' Data

The class implements a command which allows automatic cover page creation, by requesting information from the user about the document's data and placing it on the right place. To create the cover page, just use the command

```
1 \makecover
```

Since \LaTeX analyzes code from top to bottom, it is recommended that the user calls this command at the beginning of the document, right next to `\begin{document}`.

```
1 \begin{document}  
2  
3 \makecover
```

By itself, this command only returns a cover page with the filled preset values. To change these values, the class provides a set of commands, one for each cover page entry.

`\settitle{<title>}` Sets the title.

`\setsubtitle{<title>}` Sets the subtitle.

`\setauthor{<author(s)>}` Sets the author(s). To set more than one author, the class provides a command to split authors: `\tand`. Check the section [3.1.1](#) for usage examples.

`\setdegree{<course>}` Sets the course name.

`\setinstitution{<institution>}` Sets the institution. Residual command, it's not used on the cover page.

`\setdate{<date>}` Sets the date.

`\setsupervisor{<supervisor(s)>}` Sets the supervisor(s). To set more than one supervisor, check the command to set authors.

`\setchairperson{<president>}` Sets the chairperson..

`\setcommittee{<committee>}` Sets the committee. To set more than one person, check the command to set authors.

`\setcoverimage[<relative width>]{<image>}` Sets the cover page image. In this case, it's required to provide the image name, which should be found in the same directory as the document. The syntax is the same as the `\includegraphics` command, but it only accepts one argument, which is the image width relative to the text width - for example, a value of 1 would return an image which spans the cover page from margin to margin.

3.1.1 Sample Usage

To showcase the usage of the previous section's commands, here's a syntax example.

```
1 \settitle{Schiffart's Thesis}
2 \setsubtitle{About a theme}
3 \setauthor{Daniel de Schiffart}
4 \setdegree{Mestrado Integrado em Engenharia Aeroespacial}
5 \setsupervisor{Supervisor}
6 \setchairperson{President}
7 \setcommittee{Prof. A. \tand Prof. B.}
8 \setcoverimage[0.5]{example-image-a}
```

3.2 Special sections

When writing a thesis, there are certain sections written by most of the authors. Some of the sections described on the Academic Services are, therefore, implemented on the class through the use of special¹ \LaTeX environments, so they can be attributed special formatting and make any writing easier. They are three: dedication, acknowledgments, and the abstract.

The environments are defined in \LaTeX as follows.

dedication Dedication.

¹ \LaTeX environments are macros similar to commands, but instead of being called once and their content being written inside the arguments, the environments have a beginning and an end, delimited by the `\begin` e `\end` command. Check the examples of section [3.2.1](#).

acknowledgements Acknowledgements.

tabstract Abstract. This environment adds keywords to the end of the page, which can be written through a special argument. Check the section [3.2.1](#) for syntax.

fabstract Abstract. This environment is identical to the previous tabstract environment, but detects the document's language and typesets the abstract in the opposite language (portuguese when english, english when portuguese).

The text inside these environments is put into a separate page for each, and they appear in the document in the order written in the file.

3.2.1 Sample Usage

Sample usage of the provided environments. It is worth noting that, in the abstract environment (tabstract), there is an extra argument, inside square brackets, which will be printed at the bottom of the page.

```
1 \begin{dedication}
2   To dedicate to whoever you want.
3 \end{dedication}
4
5 \begin{acknowledgements}
6   Did you ever hear the tragedy of Darth Plagueis The Wise? I thought not. It's not a
   ↳ story the Jedi would tell you. It's a Sith legend. Darth Plagueis was a Dark
   ↳ Lord of the Sith, so powerful and so wise he could use the Force to influence
   ↳ the midichlorians to create life... He had such a knowledge of the dark side
   ↳ that he could even keep the ones he cared about from dying.
7 \end{acknowledgements}
8
9
10 \begin{tabstract}[Summary, Keywords, Abstract]
11   Summary and keywords (in portuguese and english). The abstract describes the goals,
   ↳ the project's content and the conclusions. It has to be written in portuguese
   ↳ and english, with at most 250 words for each, and be next to 4-6 keywords.
12 \end{tabstract}
13
14
15 \begin{fabstract}[Resumo, Palavras-Chave, Resumo Analítico]
16   Resumo e palavras-chave (em português e em inglês). O resumo analítico, também
   ↳ designado por resumo ou abstract, descreve o objectivo, o conteúdo do trabalho
   ↳ e as conclusões. Deve ser escrito em português e inglês, com um máximo de 250
   ↳ palavras cada e acompanhado de 4 a 6 palavras-chave.
17 \end{fabstract}
```

3.3 Internal Commands

These are used to create the cover page, and return the values defined by the user on section 3.1's commands. They are just used internally by the class, but can be used by the user through the document, to place their value somewhere on the text.

```
1 \tttitle{}
2 \tsubtitle{}
3 \tauthor{}
4 \tsupervisor{}
5 \tdegree{}
6 \tchairperson{}
7 \tcommittee{}
8 \tinstitution{}
9 \tdate{}
10 \tcoverimage{}
11 \tcoverimagewidth{}
```

3.4 Full list of commands and environments.

On this section, all commands available to the user can be found.

First, the commands to create the cover page.

```
1 \settitle{}
2 \setsubtitle{}
3 \setauthor{}
4 \setdegree{}
5 \setinstitution{}
6 \setdate{}
7 \setsupervisor{}
8 \setchairperson{}
9 \setcommittee{}
10 \setcoverimage[]{}
11
12 \tand
13
14 \makecover
```

Then, the available environments.

```
1 \begin{dedication}
2 \end{dedication}
3
4 \begin{acknowledgements}
5 \end{acknowledgements}
6
7 \begin{tabstract}{}
8 \end{tabstract}
9
```

```
10 \begin{fabstract}{}
11 \end{fabstract}
```

And lastly, the internal commands.

```
1 \tttitle{}
2 \tsubtitle{}
3 \tauthor{}
4 \tsupervisor{}
5 \tdegree{}
6 \tchairperson{}
7 \tcommittee{}
8 \tinstitution{}
9 \tdate{}
10 \tcoverimage{}
11 \tcoverimagewidth{}
```

Chapter 4

Class Implementation

This chapter is dedicated to describing how the class and all its functions were implemented inside the T_EX and L^AT_EX context, with special care given to the *packages* used. It has, therefore, a more technical nature, and its reading is not required for any usage of the class.

4.1 Page style

The page settings are reduced to the margin sizing. They are set to 2.5 centimeters on all sides, using the *geometry* package.

Line spacing is also set through the *setspace* package, which avoids defining the line spacing for the **entire** text, defining instead the spacing for the text body. This prevents changing spacing in, for instance, captions of images and tables. It also allows the creation of exceptions in the middle of the text, through the `\setspace` command. However, this is only recommended in special situations, in accordance to the Academic Services' regulation.

4.2 Colors and Hyperlinks

Hyperlinks found in the example thesis and in this documentation are achieved through the *hyperref* package, and are set to use colored text (instead of rectangular links surrounding the text, which is the default behaviour). This can be changed using the commands described in the official documentation for the package.

The colors used in these hyperlinks (and other places) were set using the *xcolor* package. The most frequently used color is Instituto Superior Técnico's cyan, provided by the *Identity Standards*, and is set as 1, 0, 0, 0, on the CMYK scale. The second color set is a 0.2, 0, 0, 0.8 gray, on the CMYK scale, also found in the aforementioned document.

These colors are set in the class as `ist-cyan` and `ist-gray` and can be reused according to the *xcolor* package syntax.

4.3 Fonts

Fonts have many details to write about, but the choice of each of them is tied to the Academic Services' decision to pick *Arial* as the main document's font. The discussion of each implemented font includes a relevant justification, but it is important to make clear that non-main fonts can be replaced according to the user's preference. Broadly speaking, the implementations for a document's fonts, whether the document uses the `ist-thesis` class or not, are similar and should work correctly.

4.3.1 Main font

The Academic Services' choice of font (*Arial*) is quite distinct from the root font of any \LaTeX document¹ (*Computer Modern*), but it is, perhaps, even more recognizable. However, a problem with this font is that it is way more commonly found in *Windows* systems than in others. It is, therefore, a font without a native implementation in \LaTeX .

Luckily, *Arial* is based on *Helvetica*, which is easily available only in several formats and codings. The two fonts are pretty much indistinguishable, since *Arial*'s design was heavily based on *Helvetica*'s. This second option, due to its ease of access, has found several implementations in \LaTeX .

For this project's base font, we chose the [TeX Gyre Heros](#) font, included in any \LaTeX distribution. The reasoning was that it is easy to implement, and is pretty good with regards to mixing properties, from **bold** text to *italic* and ***both at the same time***. It also provides support to foreign characters, namely characters used in Portuguese. The biggest problem in this font is the lack of math typesetting, something that will be discussed in section [4.3.3](#).

4.3.2 Monospacing

The monospaced font chosen was Inconsolata Narrow, partly due to its similarity with the main font, but mostly due to personal preference. Although this font is not extensively used in most thesis, it is important to add programming code in the middle of the text, where monospaced characters keep coherence from line to line.

4.3.3 Math typesetting

As previously mentioned in section [4.3.1](#), the font used for math is not included in the main font used, and, therefore, had to be selected.

For experienced \LaTeX users, the root font (*Computer Modern*) already includes all needed functions for a wide array of math writing needs. And although this is true, the clash between *Arial* (or *TeX Gyre Heros*, in our case) and *Computer Modern* is not the most aesthetically pleasing, which led to the search of a better alternative. Despite that, for those not worried with these details or happy with the original math font, an `origmath` option was included on the class, to keep that font.

On this project, the math font used is provided by the [newtxsf package](#).

¹In this case, we refer to documents which use the base \LaTeX classes, such as `article`, `report`, or `book`, amongst others.

4.3.4 Serif fonts

To the most perceptive readers, you may have noticed that there is a font family left to define: *serif*.

To the curious readers, an explanation follows. For each document, \LaTeX implements 2 main font families: *serif* and *sans-serif*. These terms are used, in general, to describe fonts. The former includes fonts with more details, which are better suited for paper reading. The latter includes simpler and more rounded fonts, much easier to read in screens, where details would sacrifice readability. To the more interested, [this wikipedia article](#) can clear any doubts.

By default, \LaTeX uses *serif* fonts, as the already mentioned *Computer Modern*, leaving the choice of using *sans-serif* at any point in the text to the user. However, for this project, according to the regulations of Instituto Superior Técnico, we are using a *sans-serif* text as a default, which implies that the usage of a *serif* text is available when needed.

Since this topic is of low importance, we left this font without any modifications, which implies that, when called, such as `here`, we get the root \LaTeX font.

4.3.5 Showcases

The previously mentioned fonts can be seen in this sections. In order: normal fonts (*sans-serif*), italic, small caps, *serif*, and math, all next to their bold equivalent.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 LOREM IPSUM DOLOR SIT AMET, CONSECTETUER ADIPISCING ELIT.
 LOREM IPSUM DOLOR SIT AMET, CONSECTETUER ADIPISCING ELIT.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Also included is a showcase of an equation using the font from `newtxsf`, mentioned in section 4.3.3.

$$\mathbb{P}\left(\frac{X_1 + \dots + X_n}{\sqrt{n}} \leq y\right) \rightarrow R(y) = \int_{-\infty}^y \frac{e^{-t^2/2}}{\sqrt{2\pi}} dt \quad \text{as } n \rightarrow \infty$$

4.4 Used Packages

In table 4.1, a summary of all *packages* used to implement the class can be found.

Table 4.1: *Packages* loaded by the class without optional arguments, depending on the used compiler.

pdf \LaTeX	X \LaTeX ou Lua \LaTeX
	etoolbox
	ifluatex
	ifxetex
	ifpdf
	mathtools
	graphicx
	xcolor
	hyperref
	geometry
inputenc	
fontenc	
	fontspec
	babel
tgheros	
inconsolata	
	newtxsf
	microtype
	toctibind
	setspace
	totpages