

Alysson Cirilo Silva

# **Template de monografia da UFMA em L<sup>A</sup>T<sub>E</sub>X**

São Luís – MA

2025

Alysson Cirilo Silva

## **Template de monografia da UFMA em L<sup>A</sup>T<sub>E</sub>X**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Maranhão

Orientador: Francisco José da Silva e Silva

São Luís – MA

2025

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Núcleo Integrado de Bibliotecas/UFMA

Silva, Alysson Cirilo.

Descoberta e desconexão de objetos inteligentes smart objects em ambientes oportunistas de IoMT / Alysson Cirilo Silva. - 2019.

40 f.

Orientador(a): Francisco José da Silva e Silva.  
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís, 2019.

1. IoMT. 2. IoT. 3. Middleware. I. Silva, Francisco José da Silva e. II. Título.

Alysson Cirilo Silva

## **Template de monografia da UFMA em L<sup>A</sup>T<sub>E</sub>X**

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. São Luís – MA, 19 de Julho de 2025:

---

**Francisco José da Silva e Silva**  
Doutor - UFMA

---

**Carlos de Salles Soares Neto**  
Doutor - UFMA

---

**Marcelo Henrique Monier Alves**  
**Júnior**  
Mestre - IFMA

São Luís – MA  
2025

# Agradecimentos

Em primeiro lugar, agradeço pela dádiva da vida e pela força para superar os desafios ao longo desta jornada.

À instituição de ensino e a todos os seus professores e colaboradores, pela oportunidade de acesso a uma educação de qualidade e pelo conhecimento compartilhado, que foi fundamental para o meu crescimento.

Ao meu orientador, pela confiança, paciência, e por todo o aprendizado proporcionado. Agradeço também pela oportunidade de participar de projetos enriquecedores e desafiadores, que certamente marcaram minha trajetória.

À minha família, pelo apoio incondicional, amor e compreensão em cada etapa. Sem vocês, nada disso seria possível.

Aos membros da banca examinadora, pela dedicação e por contribuírem com suas análises e sugestões para o aprimoramento deste trabalho.

Aos colegas de jornada, pelo companheirismo e por tornarem esta caminhada mais leve e significativa.

A todos os amigos que estiveram presentes ao longo do caminho, compartilhando desafios e alegrias, meu mais sincero agradecimento. Obrigado.

*The road to wisdom? Well, it's plain  
And simple to express:  
Err  
and err  
and err again,  
but less  
and less  
and less.  
(Piet Hein)*

# Resumo

Com a crescente demanda por formatos padronizados e a importância de facilitar o processo de formatação, a utilização de templates em L<sup>A</sup>T<sub>E</sub>X para a elaboração de monografias tem se tornado uma prática comum em diversas instituições acadêmicas. No contexto da UFMA, a adoção de um *template* específico para monografias, alinhado ao padrão ABNT, visa garantir a padronização da estrutura e a conformidade com as normas da instituição e da Associação Brasileira de Normas Técnicas. Esse template facilita a organização dos conteúdos e a inserção de elementos como referências bibliográficas, tabelas e figuras. Este trabalho tem como objetivo apresentar e detalhar o uso de um template em L<sup>A</sup>T<sub>E</sub>X para a elaboração de monografias na UFMA, abordando suas principais funcionalidades e vantagens no processo de formatação conforme as diretrizes da ABNT.

**Palavras-chave:** ABNT. UFMA.

# Abstract

With the growing demand for standardized formats and the importance of simplifying the formatting process, the use of L<sup>A</sup>T<sub>E</sub>X templates for writing theses has become a common practice in various academic institutions. In the context of UFMA, the adoption of a specific template for theses, aligned with ABNT standards, aims to ensure the standardization of structure and compliance with the institution's and the Brazilian Association of Technical Standards (ABNT) guidelines. This template helps organize content and insert elements such as bibliographic references, tables, and figures. This work aims to present and detail the use of a L<sup>A</sup>T<sub>E</sub>X template for writing theses at UFMA, highlighting its main features and advantages in the formatting process according to ABNT guidelines.

**Keywords:** ABNT. UFMA.

## **Lista de ilustrações**

Figura 1 – Diagrama de sequência do processo de entrega de mensagens no MQTT 15

# Lista de tabelas

Tabela 1 – Exemplo . . . . .	16
Tabela 2 – Dados de leitura por país . . . . .	17
Tabela 3 – Performance . . . . .	17
Tabela 4 – Uso de aspas no L <sup>A</sup> T <sub>E</sub> X . . . . .	21
Tabela 5 – Especificando unidades de media . . . . .	22

# Sumário

<b>Sumário</b>	<b>10</b>	
<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Sobre esse <i>template</i></b>	<b>11</b>
<b>1.2</b>	<b>Por onde começo?</b>	<b>11</b>
1.2.1	Tenho experiência com $\text{\LaTeX}$	11
1.2.2	Não tenho experiência com $\text{\LaTeX}$	12
<b>1.3</b>	<b>Organização do texto</b>	<b>12</b>
<b>2</b>	<b>ABNT</b>	<b>13</b>
<b>2.1</b>	<b>Alíneas</b>	<b>13</b>
<b>2.2</b>	<b>Citações</b>	<b>13</b>
2.2.1	Citação indireta	13
2.2.2	Citação direta	14
<b>2.3</b>	<b>Figuras</b>	<b>15</b>
<b>2.4</b>	<b>Tabelas</b>	<b>16</b>
<b>2.5</b>	<b>Estrangeirismos</b>	<b>16</b>
<b>3</b>	<b>MISC</b>	<b>18</b>
<b>3.1</b>	<b>Código</b>	<b>18</b>
3.1.1	Linguagens e estilos	19
<b>3.2</b>	<b>Remissões internas</b>	<b>19</b>
3.2.1	<i>Labels</i> em capítulos, seções e subseções	20
3.2.2	<i>Labels</i> em figuras	20
3.2.3	<i>Labels</i> em tabelas	20
3.2.4	<i>Labels</i> em códigos	20
<b>3.3</b>	<b>Aspas</b>	<b>20</b>
<b>3.4</b>	<b>Termos muito usados no seu documento</b>	<b>21</b>
<b>3.5</b>	<b>Unidades de media</b>	<b>22</b>
3.5.1	Unidades	22
3.5.2	Quantidades	23
<b>3.6</b>	<b>Fonte</b>	<b>23</b>
<b>REFERÊNCIAS</b>		<b>24</b>

# 1 Introdução

## 1.1 Sobre esse *template*

Esse projeto é um *template* “opinionado” de como estruturar e organizar um trabalho de conclusão de curso—focado em Ciência da Computação—utilizando a classe `abntex2`.

O projeto somente oferece um ponto de partida para criação de documento, com configurações e inclusões de pacotes que o autor considera útil, assim sendo, é recomendado a leitura da documentação oficial da classe `abntex2`. O projeto também vem configurado com automação para compilação com `latexmk` ou `make`, e com sistema de CI/CD no GitHub.

## 1.2 Por onde começo?

### 1.2.1 Tenho experiência com L<sup>A</sup>T<sub>E</sub>X

Se você já tem familiaridade com L<sup>A</sup>T<sub>E</sub>X e quer um resumo de como usar esse *template*, você pode olhar o código fonte desse documento e entender como montar sua monografia. Essas informações irão te auxiliar:

- a) edite o arquivo `metadata.tex` com os dados do seu trabalho;
- b) altere o arquivo de ficha catalográfica localizada em:  
`content/ficha-catalografica/index.pdf`, com o documento correspondente ao seu trabalho;
- c) o arquivo `monografia.tex` é o arquivo principal, ele inclui os arquivos no diretório `content/`;
- d) o *template* foi construído de forma que cada seção fica em um subdiretório com o seguinte padrão: `content/{nome-da-seção}/index.tex`:
  - lembre-se de atualizar o arquivo `monografia.tex` ao renomear o diretório ou o arquivo;
  - sinta-se a vontade de mudar a organização, como por exemplo:  
`content/section-01/index.tex`.
- e) termos que aparecem frequentemente no texto podem ser movidos para macros no arquivo `macro.tex` para garantir grafia e formatação consistentes;

- f) a inclusão e configuração de pacotes estão no arquivo `tccconfig.sty`;
- g) as referências bibliográficas estão em `bib/biblio.bib`.

O documento é compilado com X<sup>E</sup> $\text{\LaTeX}$  com o auxílio do programa `latexmk`. Um `Makefile` é usado para oferecer uma interface mais simples de compilação, os seguintes comandos são suportados:

- a) `make`: compila o projeto e gera o PDF;
- b) `make continuous`: compila o projeto e gera o PDF, adicionalmente, fica executando e compila novamente cada vez que um arquivo do projeto muda;
- c) `make clean`: deleta os arquivos auxiliares de compilação;
- d) `make cleanall`: deleta os arquivos auxiliares de compilação e o PDF gerado.

Também deve ser possível compilar o projeto com o *build system* ou IDE de sua preferência.

### 1.2.2 Não tenho experiência com $\text{\LaTeX}$

Vai ter que aprender o básico até entender o que está descrito na subseção 1.2.1. Esses recursos podem ser um bons pontos de partida:

- a) [wiki do abntex](#);
- b) [tutorial do Overleaf](#);
- c) [wikibook](#).

## 1.3 Organização do texto

O restante deste documento mostra como utilizar o *template* para as coisas mais comuns na monografia.

## 2 ABNT

Este capítulo mostra como usar os macros específicos do pacote abntex2. Isso é apenas um resumo, consulte a documentação oficial.

### 2.1 Alíneas

Alíneas são listas, devemos usar o ambiente `alineas`, desta forma:

```

1 \begin{alineas}
2   \item primeira alínea;                                % termina com ;
3   \item segunda alínea                                 % termina com :
4   \begin{alineas}
5     \item primeira subalínea da segunda alínea; % termina com ;
6     \item última subalínea da segunda alínea.    % termina com .
7   \end{alineas}
8   \item última alínea.                                % termina com .
9 \end{alineas}
```

Código 1 – Alíneas

O texto que vem antes, geralmente, deve terminar com `:', desta forma: O [Código 1](#) vai gerar o seguinte:

- a) primeira alínea;
- b) segunda alínea
  - primeira subalínea da segunda alínea;
  - última subalínea da segunda alínea.
- c) última alínea.

### 2.2 Citações

O *template* utiliza bibtex para gerenciar as referências do artigo, eles devem ser salvos no arquivo `bib/biblio.bib`, a sintaxe se assemelha ao que está descrito no [Código 2](#).

#### 2.2.1 Citação indireta

Para fazer uma citação indireta, é apenas necessário usar a macro `\cite`, passando o id da referência. Como apresentando no [Código 3](#).

```

1 @book{knuth:aocp:1997,
2     author = {Knuth, Donald E.},
3     title = {The art of computer programming, volume 1 (3rd ed.):
4         fundamental algorithms},
5     year = {1997},
6     isbn = {0201896834},
7     publisher = {Addison Wesley Longman Publishing Co., Inc.},
8     address = {USA}
9 }
10
11 @article{dijkstra:goto:1968,
12     title={Go To Statement Considered Harmful},
13     author={Dijkstra, Edsger W},
14     journal={Communications of the ACM},
15     volume={11},
16     number={3},
17     pages={147--148},
18     year={1968},
19     publisher={ACM New York, NY, USA}

```

Código 2 – Sintaxe bibtex

```
1 Programar pode ser bastante prazeroso \cite{knuth:aocp:1997}.
```

Código 3 – Citação indireta

O resultado vai ser: Programar pode ser bastante prazeroso ([KNUTH, 1997](#)).

Também pode-se usar a macro `\citeonline` para adicionar o autor sem parênteses

## 2.2.2 Citação direta

Use o ambiente `citacao` como no [Código 4](#).

```

1 \begin{citacao}
2 For a number of years I have been familiar with the observation that
3 the quality of programmers is a decreasing function of the density of
4 go to statements in the programs they produce. More recently I
5 discovered why the use of the go to statement has such disastrous
6 effects, and I became convinced that the go to statement should be
7 abolished from all \enquote{higher level} programming languages.
8 \cite{dijkstra:goto:1968}.
9 \end{citacao}

```

Código 4 – Citação direta

O resultado será:

For a number of years I have been familiar with the observation that  
the quality of programmers is a decreasing function of the density of go

to statements in the programs they produce. More recently I discovered why the use of the go to statement has such disastrous effects, and I became convinced that the go to statement should be abolished from all “higher level” programming languages. ([DIJKSTRA, 1968](#)).

Como o texto em questão está em inglês, deveríamos usar a opção `english`, desta forma: `\begin{citacao}[english]`. O que vai deixar o texto em itálico:

*For a number of years I have been familiar with the observation that the quality of programmers is a decreasing function of the density of go to statements in the programs they produce. More recently I discovered why the use of the go to statement has such disastrous effects, and I became convinced that the go to statement should be abolished from all “higher level” programming languages. ([DIJKSTRA, 1968](#)).*

## 2.3 Figuras

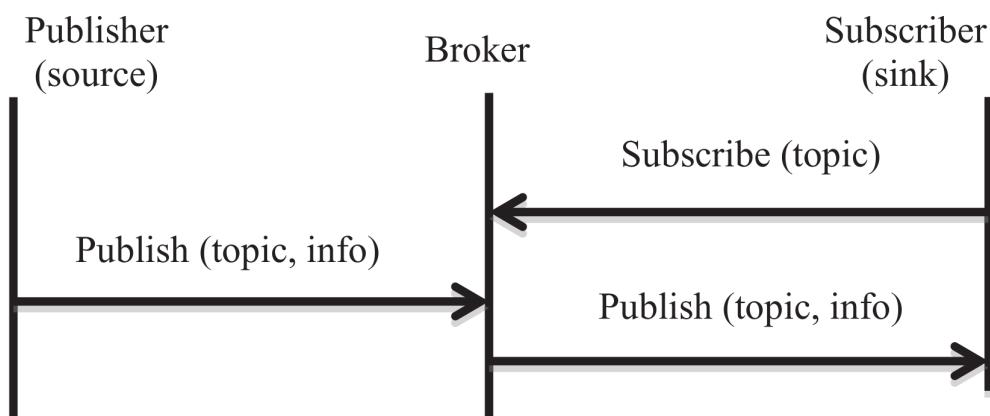
Use o ambiente `figure` como no Código 5, que produz a Figura 1.

```

1 \begin{figure}[htb]
2   \caption{Diagrama de sequência do processo de entrega de
3     mensagens no \mqtt}
4   \label{fig:mqtt}
5   \centering
6   \includegraphics[width=0.85\linewidth]{\CurrentFilePath/res/img/
7     mqtt-sequence.png}
8   \fonte{\citeonline{al-fuqaha-et-al:2015}}
9 \end{figure}
```

Código 5 – Adição de imagem

Figura 1 – Diagrama de sequência do processo de entrega de mensagens no MQTT



Fonte: [Al-Fuqaha et al. \(2015\)](#)

## 2.4 Tabelas

Tabelas no padrão ABNT têm uma sintaxe complicada, copie e cole o [Código 6](#) altere os dados, ele vai gerar a [Tabela 1](#).

```

1 \begin{table}[htb]
2   \begin{center}
3     \IBGETab{
4       \caption{Exemplo}
5       \label{tab:fake}
6     }{
7       \begin{tabular}{lcr} % left, center, right
8         \toprule
9         \textbf{Col 1} & \textbf{Col 2} & \textbf{Col 3} \\
10        \midrule
11        Texto & Outro texto & Mais texto \\
12        1.1 & $\log_2 n$ & 3.14 \\
13        \bottomrule
14      \end{tabular}
15    }{
16      \fonte{\autoriapropria}
17    }
18  \end{center}
19 \end{table}
```

Código 6 – Tabela

Tabela 1 – Exemplo

Col 1	Col 2	Col 3
Texto	Outro texto	Mais texto
1.1	$\log_2 n$	3.14

Fonte: Produzido pelo autor

Você vai encontrar outras tabelas nesse capítulo, consulte o código fonte desse documento caso queira ver como são construídas.

## 2.5 Estrangeirismos

Utilize o comando `\foreign` para deixar o termo em itálico. Tome como exemplo o [Código 7](#), ele gerará o seguinte paragrafo, com a palavra “*dataset*” em itálico:

Para a análise estatística, foi necessário utilizar um *dataset* abrangente, contendo uma grande quantidade de dados que permitiram gerar insights significativos sobre o comportamento do usuário.

Tabela 2 – Dados de leitura por país

País	Livros lidos por ano (média)	Gênero mais popular
Brasil	4,7	Ficção
Estados Unidos	10	Ficção
Japão	12	Mangás
Alemanha	12	Ficção
França	21	Romances
Índia	4,7	Ficção

Fonte: ChatGPT :)

Tabela 3 – Performance

	Média	Desvio padrão	Mínimo	Máximo	Intervalo de 95% de confiança
$\Delta t$ (ms)	20.1	3.7	7.0	78.9	20.1 ± 0.03

Fonte: Produzido pelo autor

1 Para a análise estatística, foi necessário utilizar um  
 2 `\foreign{dataset}` abrangente, contendo uma grande quantidade  
 3 de dados que permitiram gerar insights significativos  
 4 sobre o comportamento do usuário.

#### Código 7 – Estrangeirismos

Não existe uma regra, mas entende-se que algumas palavras já sofreram aportuguesamento e não necessitam de grafia especial, o [manual da SECOM](#) pode ser usado como referência.

## 3 Misc

### 3.1 Código

Inclusão de código pode ser feita através do pacote `listings`. Utilize conforme o Código 8.

```

1 \begin{lstlisting}[
2     float=htb,
3     style=javaStyle,
4     caption={Fatorial},
5     label=lst:factorial-inline-output
6 ]
7 public class Main {
8     public static void main(String[] args) {
9         System.out.println(factorial(10));
10    }
11
12    static int factorial(int n) {
13        if (n == 1) return 1;
14
15        return n * factorial(n - 1);
16    }
17 }
18 \end{lstlisting}

```

Código 8 – Inclusão de código

O conteúdo dentro do bloco `\lstlisting` será colocado dentro do seu documento, como os exemplos de código que você viu até agora, no caso do Código 8, este iria gerar como output o Código 9.

```

1 public class Main {
2     public static void main(String[] args) {
3         System.out.println(factorial(10));
4     }
5
6     static int factorial(int n) {
7         if (n == 1) return 1;
8
9         return n * factorial(n - 1);
10    }
11 }

```

Código 9 – Fatorial

Caso queira deixar seu código L<sup>A</sup>T<sub>E</sub>X mais compacto, você pode colocar o seu código em um arquivo separado e incluir em seu documento com o comando `\lstinputlisting`. Esse *template* sugere que coloque esse códigos em `content/{nome-da-seção}/res/code`. Assumindo que o código existe em um arquivo `Factorial.java` dentro desse diretório, o exemplo anterior poderia ser escrito conforme o [Código 10](#).

```

1 \lstinputlisting[
2   float=htb,
3   style=javastyle,
4   label=lst:factorial-include-output,
5   caption=Fatorial
6 ]{\CurrentFilePath/res/code/Factorial.java}
```

Código 10 – Inclusão de código através de um arquivo externo

### 3.1.1 Linguagens e estilos

Você pode especificar o estilo de código para uma determinada linguagem alterando o arquivo `tccconfig.sty`, incluindo cores e palavras reservadas adicionais, siga conforme os estilos já definidos. Consulte a documentação oficial para coisas mais complexas.

## 3.2 Remissões internas

Se você quer referenciar partes do seu documento, como: capítulos, seções, subseções, figuras, tabelas e códigos, você deve primeiramente adicionar uma *label* nesse elemento, depois deve usar o comando `\autoref` ou `\ref` no local que deseja mencioná-lo.

O comando `\autoref` adiciona um link para o elemento contendo o nome que o identifica (como: “capítulo”, “seção”, “subseção”, “tabela” e etc.) seguido do número desse elemento no documento.

O comando `\ref` adiciona um link para o elemento contendo *apenas* o número do elemento no documento.

Como exemplo, adicionei a *label* “`sec:cross-ref`” na seção atual, ao utilizar o comando `\autoref{sec:cross-ref}`, vou obter o seguinte resultado: “[seção 3.2](#)”. Entretanto, se utilizar `\ref{sec:cross-ref}`, vou obter apenas: “[3.2](#)”, que pode ser útil ao referenciar múltiplos elementos ao mesmo tempo.

A forma de adicionar *labels* é ligeiramente diferente dependendo do tipo de elemento sendo referenciado.

### 3.2.1 *Labels* em capítulos, seções e subseções

Para capítulos, seções e subseções, usamos o comando `label` seguido do id da *label* que queremos atribuir, conforme o [Código 11](#).

```

1 \chapter{Introdução} \label{chap:intro}
2
3 Blá.
4
5 \section{Panorama Geral} \label{sec:overview}
6
7 Blá.
8
9 \subsection{Questões Fundamentais} \label{sub:fundamental-questions}
10
11 Blá.

```

Código 11 – Atribuição de *labels* em capítulos, seções e subseções

Desta forma, eu posso referenciar o capítulo “Introdução” usando o comando: `\autoref{chap:intro}`.

### 3.2.2 *Labels* em figuras

Também se utiliza o comando `label`, entretanto, dentro do ambiente `figure` e após a macro `caption`, consulte o [Código 5](#).

### 3.2.3 *Labels* em tabelas

Também se utiliza o comando `label`, entretanto, dentro do primeiro bloco do comando `IBGЕtab` e após a macro `caption`, consulte o [Código 6](#).

### 3.2.4 *Labels* em códigos

*Não* utilizamos comando `label` nesse caso, utilizaremos o argumento opcional `label` presente no ambiente `lstlisting` e no comando `lstinputlisting`, consulte os códigos [8](#) e [10](#).

## 3.3 Aspas

Aspas em L<sup>A</sup>T<sub>E</sub>X são mais complicadas do que parecem. Ao tentar usar os caracteres “'” ou “""” (aspas simples e dupla, respectivamente), o resultado não será o esperado, não gerando as aspas curvas. O uso errôneo desse caracteres no [Código 12](#) irá gerar o seguinte resultado: ‘Aspas simples’ e "aspas duplas".

<sup>1</sup> 'Aspas simples' e "aspas duplas".

Código 12 – Uso incorreto de aspas

Note que ‘’’ sempre gera aspas direitas e “”” sempre gera aspas duplas retas. Para produzir aspas corretamente, o L<sup>A</sup>T<sub>E</sub>X espera que se utilize os símbolos presentes na Tabela 4.

Tabela 4 – Uso de aspas no L<sup>A</sup>T<sub>E</sub>X

Resultado desejado	Símbolo	Descrição do símbolo	Resultado
Aspa simples esquerda	'	1 acento grave	‘
Aspa simples direita	'	1 aspa simples	’
Aspa dupla esquerda	``	2 acentos grave	“
Aspa dupla direita	''	2 aspas simples	”

Fonte: Produzido pelo autor

Ou seja, o exemplo anterior deveria ser escrito conforme o Código 13, gerando: ‘Aspas simples’ e “aspas duplas”.

<sup>1</sup> 'Aspas simples' e ``aspas duplas''.

Código 13 – Uso correto de aspas

“abc” ‘abc’ Como esse método de inclusão de aspas é pouco intuitivo e fácil de errar, podemos usar o pacote `csquotes` para auxiliar. Usamos os comando `enquote*` para aspas simples e `enquote` para aspas duplas. O exemplo pode ser então reescrito conforme o Código 14, gerando: ‘Aspas simples’ e “Aspas duplas”.

<sup>1</sup> \enquote\*{Aspas simples} e \enquote{Aspas duplas}.

Código 14 – Uso do pacote csquotes

## 3.4 Termos muito usados no seu documento

Se o seu texto utiliza com bastante frequência termos que você quer que tenham tipografia e escrita consistente, elas podem ser adicionadas como macros no arquivo `macro.tex`, com o comando `newterm`. Vamos olhar alguns exemplos no Código 15.

Neste caso, foram definidos 3 macros (`\api`, `\broadcast` e `\printf`), a intenção do autor é que o termo API sempre seja escrito em maiúsculo, o termo `broadcast` sempre seja escrito como uma palavra estrangeira e o termo `printf` sempre seja formatado com fonte monoespaçada.

```

1 \newterm{\api}{API}
2 \newterm{\broadcast}{\foreign{broadcast}}
3 \newterm{\printf}{\texttt{\printf}}

```

Código 15 – Definição de macros

Para usar os macros definidos pelo autor, basta incluir no texto desejado, conforme o [Código 16](#), gerando o seguinte resultado:

Usando essa API, você pode fazer um *broadcast* de dados para vários dispositivos. Para verificar o status, pode-se utilizar `\printf` para imprimir mensagens no console.

```

1 Usando essa \api, você pode fazer um \broadcast de dados para
2 vários dispositivos. Para verificar o status, pode-se utilizar
3 \printf para imprimir mensagens no console.

```

Código 16 – Uso de macros

## 3.5 Unidades de media

Aconselho o uso do pacote `siunitx`, a leitura do manual é extremamente útil para descobrir todos os comandos que você vai precisar usar. Aqui está um resumo:

### 3.5.1 Unidades

Use `unit` para escrever unidades de medida. Esse comando aceita duas sintaxes, uma usando macros predefinidas, e outra usando strings. Por exemplo, para escrever “metros por segundo ao quadrado”, podemos usar `\unit{\meter\per\second\squared}` ou `\unit{m/s^2}`. Entretanto, esses dois comandos vão produzir resultados levemente diferente, conforme

Tabela 5 – Especificando unidades de media

Comando	Resultado
<code>\unit{\meter\per\second\squared}</code>	$\text{m s}^{-2}$
<code>\unit{m/s^2}</code>	$\text{m/s}^2$

Fonte: Produzido pelo autor

Apesar de representar a mesma unidade, por padrão, o macro `\per` usa o formato de exponencial, isso pode ser alterado passando um argumento opcional, da seguinte forma: `\unit[per-mode=symbol]{\meter\per\second\squared}`, gerando:  $\text{m/s}^2$ .

Também existem alguns macros abreviados, no caso anterior, poderíamos escrever `\unit[per-mode=symbol]{\m\per\s\squared}`, para obter:  $\text{m/s}^2$ .

Os prefixos como quilo, mega, e giga também estão definidos e podem ser usados normalmente, para obter kg, usa-se `\unit{\kilo\gram}`.

Prefixos binários também estão disponíveis (kibi, mebi, gibi, etc), assim como macros para bit e byte, por exemplo: `\unit[per-mode=symbol]{\mibi\byte\per\second}`, produz KiB/s.

### 3.5.2 Quantidades

Quantidades podem ser especificadas com a macro `qty`, essa macro aceita dois parâmetros obrigatórios, o número e a unidade. Para especificar “5 gibibytes”, você usaria `\qty{5}{\gibi\byte}`, que produz: 5 GiB.

## 3.6 Fonte

A ABNT não determina um tipo de fonte específico, mas é um equívoco comum que deve ser utilizado a fonte Arial. Se você precisar mudar a fonte, esse documento é compilado com X~~L~~AT<sub>E</sub>X então você pode remover o pacote `fontenc` e usar o pacote `fontspec` e usar qualquer fonte em seu computador.

## Referências

AL-FUQAHÀ, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, IEEE, v. 17, n. 4, p. 2347–2376, 2015. Citado na página 15.

DIJKSTRA, E. W. Go to statement considered harmful. *Communications of the ACM*, ACM New York, NY, USA, v. 11, n. 3, p. 147–148, 1968. Citado na página 15.

KNUTH, D. E. *The art of computer programming, volume 1 (3rd ed.): fundamental algorithms*. USA: Addison Wesley Longman Publishing Co., Inc., 1997. ISBN 0201896834. Citado na página 14.