



PET.COMP

# LATEX NA PRÁTICA: PRODUÇÃO PROFISSIONAL DE DOCUMENTOS ACADÊMICOS

L<sup>A</sup>T<sub>E</sub>X

Overleaf

|                                                     |           |
|-----------------------------------------------------|-----------|
| <b>1 Introdução ao LaTeX.....</b>                   | <b>2</b>  |
| 1.1 O que é o LaTeX ?.....                          | 2         |
| 1.2 Porque usar o LaTeX ?.....                      | 3         |
| 1.3 Onde usar o LaTeX ?.....                        | 3         |
| <b>2 Primeiros Passos no LaTeX (Overleaf).....</b>  | <b>4</b>  |
| 2.1 Criando sua conta.....                          | 4         |
| 2.2 Criando seu primeiro documento.....             | 5         |
| 2.3 Estrutura básica de um documento.....           | 8         |
| 2.4 Usando templates LaTeX no Overleaf.....         | 9         |
| 2.5 Primeiro Exercício - Primeiros Passos.....      | 12        |
| <b>3 Estrutura e Comandos.....</b>                  | <b>12</b> |
| 3.1 Pacotes.....                                    | 12        |
| 3.2 Títulos e Seções.....                           | 12        |
| 3.3 Formatação de Texto.....                        | 13        |
| 3.3.1 Segundo Exercício - Formatação de Texto.....  | 13        |
| 3.4 Listas.....                                     | 13        |
| 3.4.1 Terceiro Exercício - Listas.....              | 15        |
| <b>4 Equações Matemáticas.....</b>                  | <b>15</b> |
| 4.1 Inline e Destacadas.....                        | 15        |
| 4.2 Símbolos.....                                   | 17        |
| 4.3 Matrizes.....                                   | 18        |
| 4.4 Quarto Exercício - Fórmulas.....                | 20        |
| <b>5 Inserção de Imagens.....</b>                   | <b>20</b> |
| 5.1 Manipulando largura e altura de imagens.....    | 22        |
| 5.2 Quinto Exercício - Imagens.....                 | 22        |
| <b>6 Tabelas.....</b>                               | <b>22</b> |
| <b>7 Citações e Referências Bibliográficas.....</b> | <b>26</b> |
| 7.1 Estrutura do .bib.....                          | 26        |
| 7.1.1 Principais tipos de referência no BibTeX..... | 28        |
| 7.2 Inserindo Referências.....                      | 29        |
| 7.3 Citando no Texto.....                           | 29        |
| 7.4 Lista de Referências.....                       | 30        |
| 7.5 Sexto Exercício - Referências.....              | 31        |
| <b>8 Dicas Finais e Recursos.....</b>               | <b>31</b> |

# 1 Introdução ao LaTeX

## 1.1 O que é o LaTeX ?

O **LaTeX** é um sistema de preparação de documentos muito valorizado por sua qualidade tipográfica e pelo acabamento profissional que proporciona. Ele se tornou especialmente popular em áreas acadêmicas e técnicas, onde clareza e principalmente a **padronização** são fundamentais.

Ao contrário de editores de texto tradicionais e conhecidos, como o Microsoft Word ou o Google Docs, o LaTeX não se baseia em botões de formatação visual, ou seja, uma interface gráfica visual de edição. Em vez disso, utiliza comandos de marcação que descrevem a estrutura e a função de cada parte do texto. Essa abordagem separa completamente o conteúdo da aparência, assim garantindo um resultado visual uniforme e facilitando a manutenção, mesmo em trabalhos mais longos e complexos.

Entre as principais características do LaTeX, destacam-se:

- É gratuito e de código aberto.
- Produz textos com qualidade tipográfica profissional.
- É especialmente indicado para documentos extensos.
- Permite ampla personalização por meio de pacotes adicionais.

Abaixo podemos ver um artigo gerado em LaTeX:

### The Last - Jogo de sobrevivência com desafios de lógica de programação utilizando o unity 3D

Adriely Avelino de Castro<sup>1</sup>, Antônio Carlos Stephan de Souza Neto<sup>1</sup>,  
Daniel Ferreira Ferraz<sup>1</sup>, Luan Soares Oliveira<sup>1</sup>

<sup>1</sup>Centro Tecnológico – Centro de Educação Tecnológica de Minas Gerais (CEFET-MG)  
Caixa Postal 15.064 – 91.501-970 – Leopoldina – MG – Brazil

**Abstract.** *The gaming industry has become one of the world's largest markets, surpassing traditional sectors like music and film. Once limited to large companies, game development is now accessible to independent creators, thanks to tools like Unity 3D, which provide extensive libraries and community support. This shift has fueled innovation in the indie scene, with The Last as a notable example. Nearing the final stages of development, it aims to push the boundaries of national game development, highlighting the potential of independent studios.*

**Resumo.** *A indústria de jogos tornou-se um dos maiores mercados globais, superando setores tradicionais como a música e o cinema. Antes restrita a grandes empresas, o desenvolvimento de jogos agora está acessível a criadores independentes, graças a ferramentas como a Unity 3D, que oferecem amplas bibliotecas e suporte à comunidade. Essa mudança impulsionou a inovação no cenário independente, com The Last como um exemplo notável. Em seus estágios finais de desenvolvimento, o jogo busca expandir os limites da indústria nacional, destacando o potencial dos estúdios independentes.*

#### 1. Introdução

A indústria de jogos vêm se modificando desde os anos 1970. O que começou como experimentos em laboratórios universitários — como o clássico *Pong* (1972) — transformou-se em um fenômeno cultural que hoje movimenta mais dinheiro que *Hollywood* e a indústria musical juntas [Tuorv 2015]. Porém, durante muito tempo, criar jogos em um privilégio de poucos: exigia máquinas avançadas e de difícil acesso, equipes enormes e um acesso a tecnologias que só grandes estúdios podiam financiar [Zimmerman 2020].

Na virada do século, o surgimento de motores gráficos como *Unity* e *Unreal Engine* representou um marco na democratização da criação de jogos. Essas ferramentas, ao oferecerem versões gratuitas ou acessíveis, permitiram que indivíduos com equipamentos básicos pudessem experimentar a construção de ambientes 3D e 2D complexos, além de simulações físicas avançadas, sem a necessidade de investimentos iniciais significativos [Kiamos 2023] e [Technologies 2024]. No entanto, o impacto dessas tecnologias foi amplificado pelo surgimento de plataformas de compartilhamento de conhecimento, como o *YouTube*, que se tornaram espaços de aprendizado informal. Por meio de tutoriais de programação, cursos de design de níveis e comunidades online, como o *Reddit*, desenvolvedores independentes passaram a ter acesso a recursos educacionais que, anteriormente, eram restritos a ambientes acadêmicos ou corporativos [Gee 2003]. Essa combinação de

ferramentas acessíveis e conhecimento compartilhado deu origem a uma nova geração de criadores, capazes de materializar suas ideias sem depender de formação tradicional ou conexões na indústria.

Essa maior acessibilidade ao desenvolvimento reflete-se numericamente: hoje, 4 em cada 10 jogos lançados globalmente são independentes (ou indie) [Newzoo 2023]. Títulos como *Hollow Knight* e *Celeste* provaram que histórias pessoais e mecânicas ousadas podem conquistar o público tanto quanto *blockbusters* de orçamentos bilionários, consolidando-se como exemplo do potencial criativo dos jogos independentes [Pearce 2019]. Nesse contexto, temos “The Last” — um projeto brasileiro que mistura sobrevivência pós-apocalíptica com quebra-cabeças matemáticos e lógicos, produzido utilizando a *Unity Engine*. Este artigo descreve a continuidade do desenvolvimento do jogo, cujas etapas anteriores — incluindo a prototipagem de mecânicas básicas de sistemas para documentos encontrados no jogo, menu inicial e a implementação de um sistema de *minigame* — foram detalhadas em trabalhos prévios [Castro et al. 2023]. A presente fase concentra-se em otimizações técnicas, melhorias gráficas e expansão de funcionalidades, visando consolidar o projeto como produto e exemplo de inovação na área.

O Brasil, vem se tornando um protagonista no crescimento do cenário de jogos. Entre 2020 e 2023, o número de estúdios independentes no país cresceu 27%, impulsionado por editais de fomento e eventos como a *Brasil Game Show* (BGS), que em 2022 atraiu 300 mil visitantes [ABRAGAMES 2023]. O presente projeto está intimamente ligado a esse ecossistema de desenvolvimento independente, mas destaca-se por abordagens inovadoras, como mecânicas de quebra-cabeças lógico-matemáticos que estimulam a resolução de problemas.

Estudos demonstram que jogos com elementos que exigem resolução de problemas podem contribuir para o aprimoramento cognitivo. Por exemplo, adolescentes que dedicam seis horas semanais a jogos com desafios lógicos apresentam uma probabilidade 23% maior de melhorar seu desempenho em matemática [Granic et al. 2014]. Além disso, pesquisas indicam que jogos eletrônicos podem ter efeitos positivos na saúde mental. Um estudo revelou que pacientes que jogam *Tetris* por 20 minutos diários experimentaram uma redução de 34% nos níveis de cortisol, um marcador biológico associado ao estresse [Russoniello et al. 2009].

Dessa forma, projetos como “The Last” exemplificam uma transformação significativa no cenário do desenvolvimento de jogos, onde a criatividade e o acesso democratizado às ferramentas de produção não apenas redefinem os paradigmas da indústria, mas também ampliam as possibilidades do entretenimento digital como um todo. Essa evolução é marcada pela utilização de tecnologias acessíveis, que permitem a criação de experiências complexas e inovadoras com recursos mínimos, como um computador pessoal e conexão à internet, democratizando o processo de desenvolvimento e fomentando a diversidade de narrativas e mecânicas no mercado.

#### 2. Trabalhos Correlatos

Para servir de base no desenvolvimento do projeto “The Last”, foram realizadas pesquisas de projetos similares, e optou-se por analisar jogos independentes que documentam seu progresso de desenvolvimento. Essa abordagem permite analisar ferramentas e métodos utilizados na indústria, possibilitando a escolha daquelas que melhor se enquadram no

## 1.2 Porque usar o LaTeX ?

Podemos citar diversos motivos pelos quais o LaTeX se mostra como uma opção mais interessante para a escrita e construção de textos, principalmente artigos. Podemos destacar os seguintes motivos:

- Mantém a formatação uniforme em todo o documento.
- Excelente suporte para equações e expressões matemáticas complexas.
- Gera automaticamente sumários, listas e bibliografias.
- Segue padrões acadêmicos aceitos internacionalmente.

## 1.3 Onde usar o LaTeX ?

O uso do LaTeX é amplo e não é usado apenas para a escrita de artigos científicos. Ele é ideal para qualquer tipo de documento que precise de clareza, padronização e aparência profissional.

Entre as aplicações mais comuns podemos citar:

- Teses, dissertações e monografias.
- Artigos científicos em periódicos e congressos.
- Relatórios técnicos e documentos corporativos.
- Apostilas, livros e materiais de ensino.
- Apresentações de slides com o pacote Beamer.

## 2 Primeiros Passos no LaTeX (Overleaf)

O **Overleaf** é uma plataforma online que simplifica o uso do LaTeX, pois dispensa a instalação de programas e a configuração manual no computador. Todo o processo de construção do texto é feito online, pelo navegador, com salvamento automático. Além disso, o Overleaf oferece a possibilidade de colaboração em tempo real — que é ideal para trabalhos em equipe.

Outro ponto positivo é que o Overleaf oferece uma ampla variedade de modelos prontos (*templates*), integração com repositórios Git e funciona em qualquer sistema operacional (por ser diretamente no navegador).

Abaixo iremos passar pelo processo de criação de conta na plataforma Overleaf e depois a criação do seu primeiro documento.

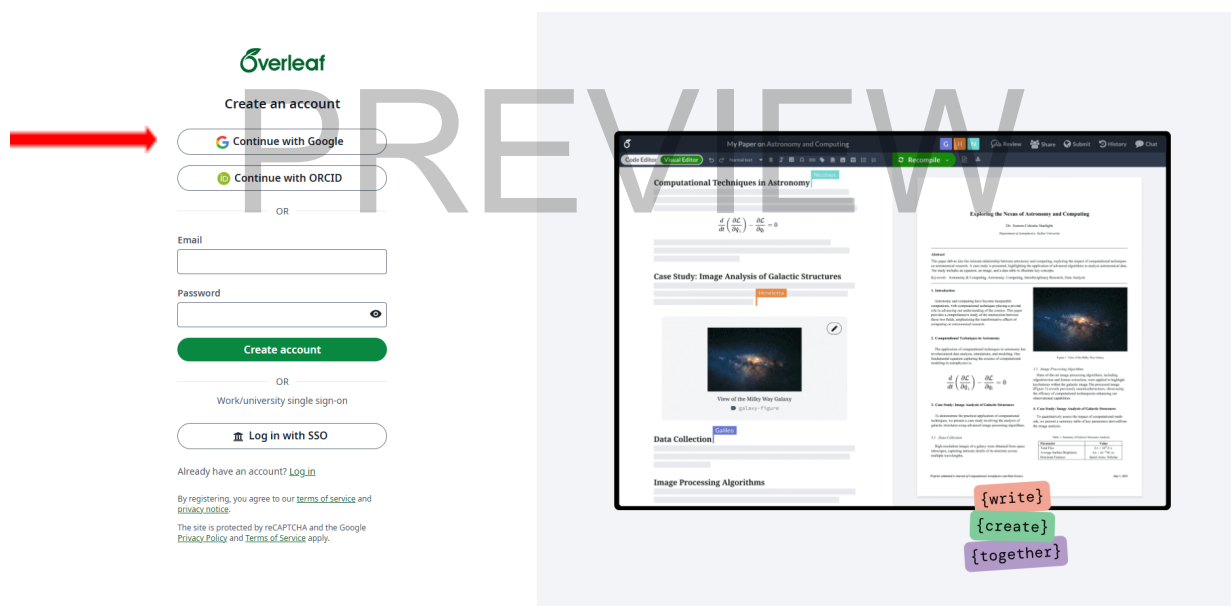
### 2.1 Criando sua conta

1. Acesse <https://www.overleaf.com>
2. Clique em **Sign Up**
3. Escolha cadastro com e-mail, Google ou ORCID
4. Confirme seu e-mail

Ao clicar no link (Passo 1) você estará no site do Overleaf (imagem abaixo), prossiga para o passo 2 clicando em “**Sign Up**”.



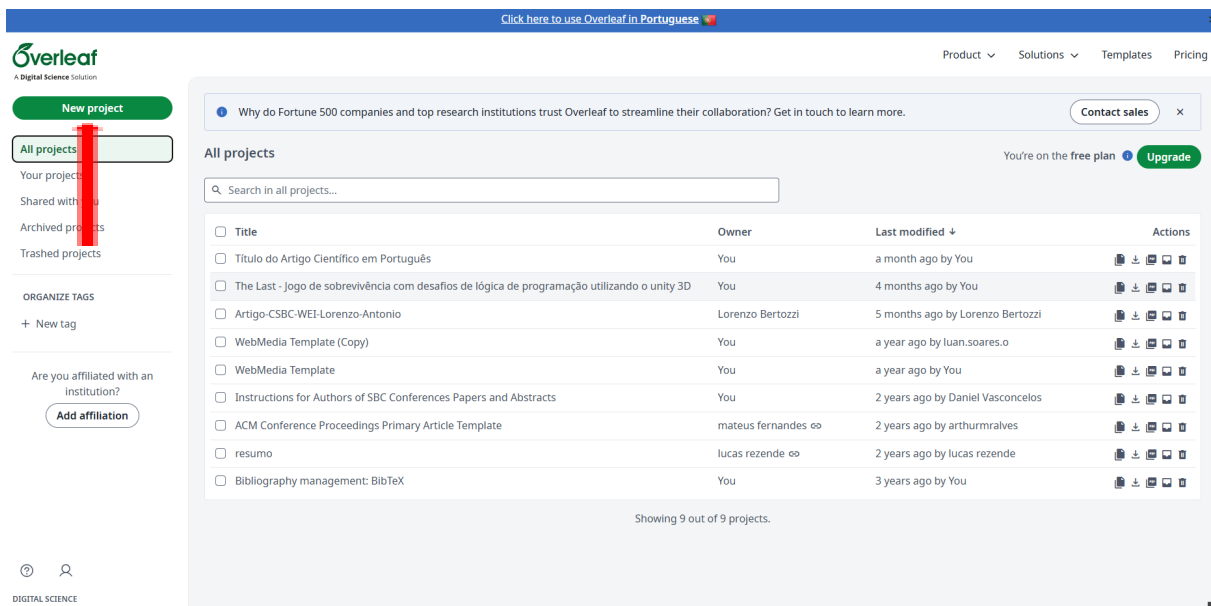
Ao clicar no botão de “**Sign Up**” o Overleaf lhe dará a possibilidade de se cadastrar com e-mail diretamente, conta do Google ou ORCID. Escolha o que for mais adequado para você (recomendamos usar sua conta do Google pela praticidade de usar a mesma). A tela de “**Sign Up**” ou “**Criação de Conta**” em português é a da imagem logo abaixo:



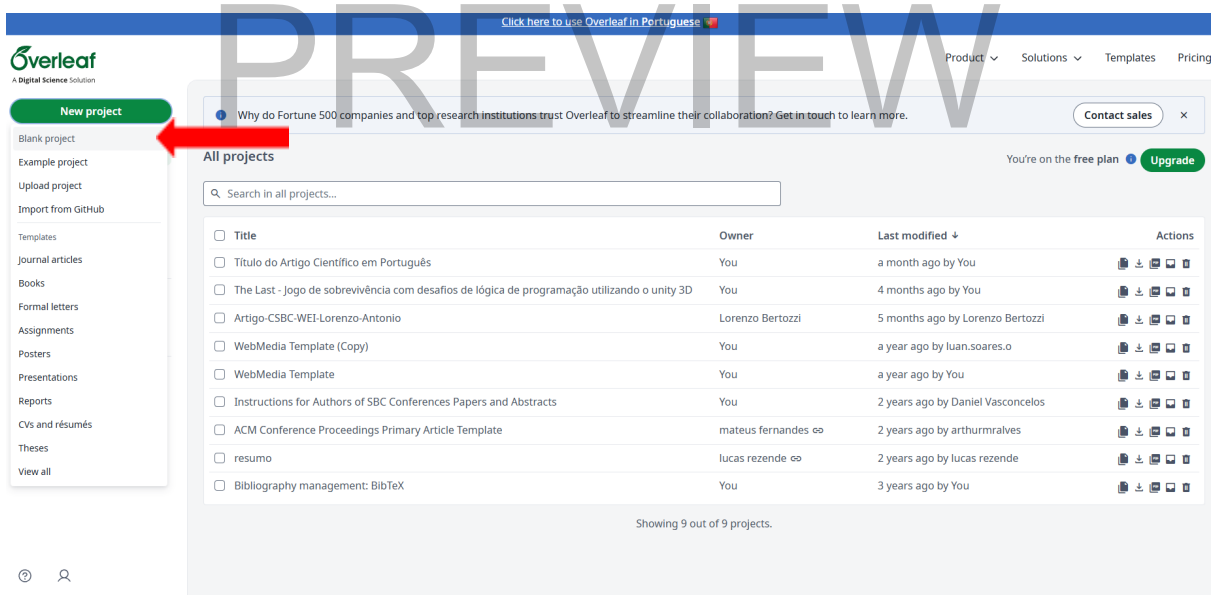
## 2.2 Criando seu primeiro documento

1. Clique em **New Project** → **Blank Project**
2. Nomeie o projeto
3. O Overleaf mostrará duas áreas:
  - Editor de código à esquerda
  - Pré-visualização do PDF à direita

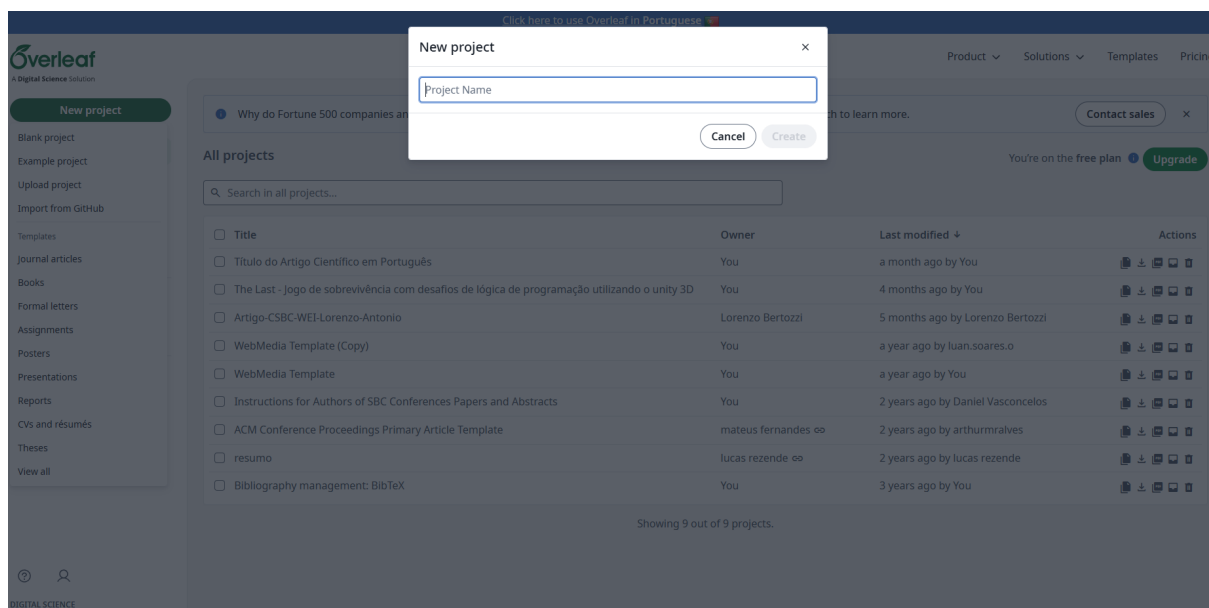
Após a conclusão de criação de conta, assim que fizer login e entrar no Overleaf, você irá se deparar com a página “**Your Projects**” ou em português, “**Seus Projetos**”. Segue na imagem abaixo essa tela.



Como descrito no passo 1, clique em **New Project** na tela acima, assim, a seguinte tela aparecerá:

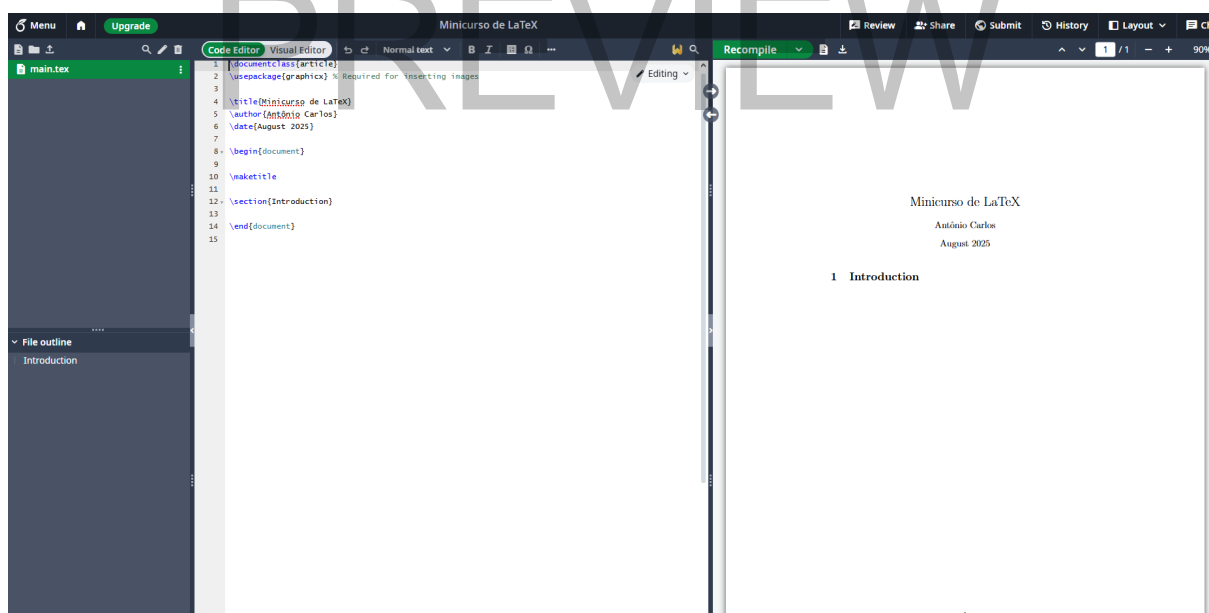


Selecione a opção **Blank Project** para prosseguir. Agora o Overleaf deve criar um projeto para você e irá pedir que você o nomeie (passo 2), segue em imagem abaixo:



Você pode nomear o projeto como preferir, mas para este minicurso iremos nomear o nosso projeto de **“Minicurso de LaTeX”**.

Com o projeto criado, o Overleaf irá automaticamente abrir o mesmo, sendo essa a tela de edição de documento do mesmo:



Perceba que temos duas áreas principais na tela (Em branco), a tela esquerda é o **editor de código**, é por ele que iremos de fato escrever o nosso texto. Já a tela à direita, é a **pré-visualização do PDF**, ou seja, como o seu texto está compilado após suas alterações no editor de código. Esta é a principal área de trabalho do Overleaf.

Agora iremos passar pela estrutura básica de um documento em LaTeX.

## 2.3 Estrutura básica de um documento

Segue abaixo um código de exemplo de um documento básico em LaTeX:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[brazil]{babel}

\title{Meu Primeiro Documento}
\author{Seu Nome}
\date{\today}

\begin{document}

\maketitle

Olá, mundo! Este é meu primeiro documento em LaTeX.
\end{document}
```

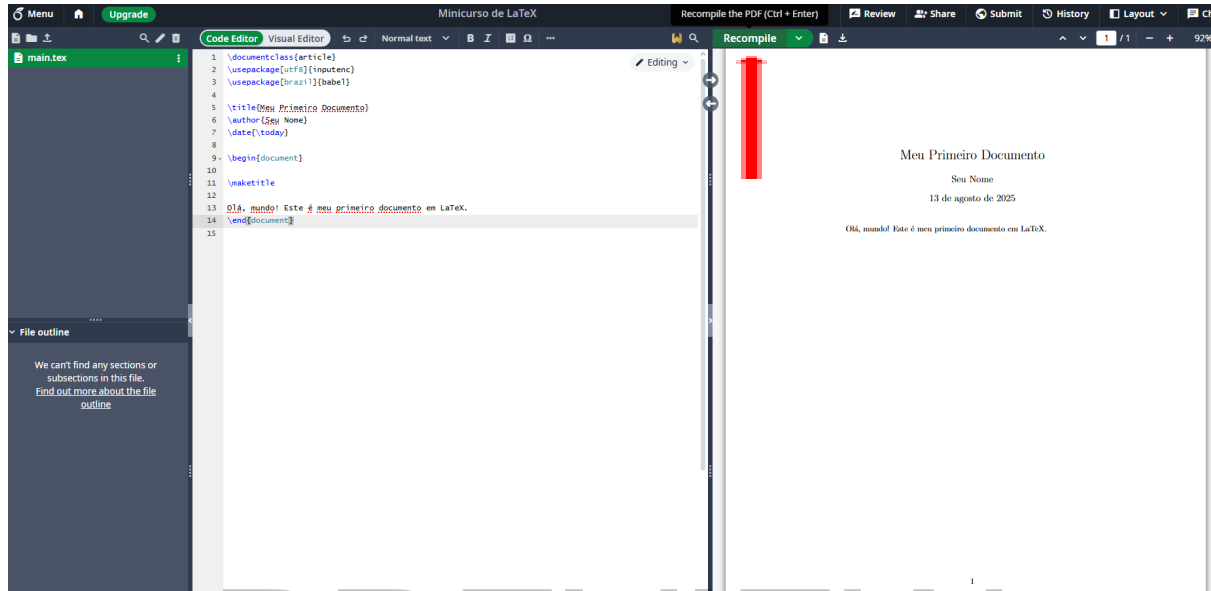
 **Observação:** Ao longo desta apostila, todo e qualquer trecho de código a ser inserido no editor de código de LaTeX será posto na cor **verde**, com o objetivo de facilitar para você leitor, o processo de encontrar o que precisa.

O código acima pode ser copiado e colado no editor de código do seu projeto recém criado, deletando o que já existe por padrão do Overleaf, caso queira. Vale a pena também aprendermos um pouco mais sobre o que cada comando utilizado faz:

- **\documentclass{article}** define o tipo de documento, neste caso um artigo simples; outros exemplos incluem **report**, **book** e **beamer**.
- **\usepackage[utf8]{inputenc}** configura a codificação de caracteres, permitindo o uso de acentos e caracteres especiais diretamente no texto.
- **\usepackage[brazil]{babel}** define o idioma do documento para português do Brasil, ajustando hifenização e termos automáticos.
- **\title{...}** especifica o título que será exibido no início do documento.
- **\author{...}** indica o(s) autor(es) do documento.
- **\date{...}** define a data de criação; pode-se usar **\today** para inserir a data atual automaticamente.
- **\begin{document}** e **\end{document}** delimitam o conteúdo do documento.

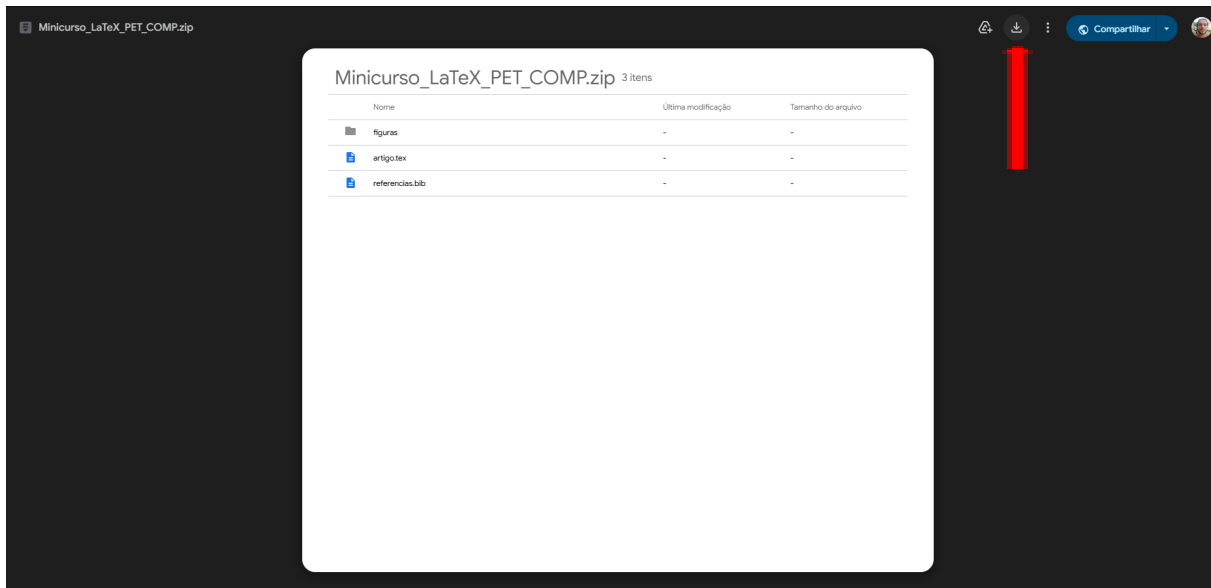
- `\maketitle` gera a página de título a partir das informações de título, autor e data fornecidas.

Com este código inserido no seu **editor de código** basta clicar em **Recompile** para que o Overleaf compile seu código LaTeX e gere uma pré-visualização do seu PDF na tela à direita. Deve ficar como a imagem logo abaixo:

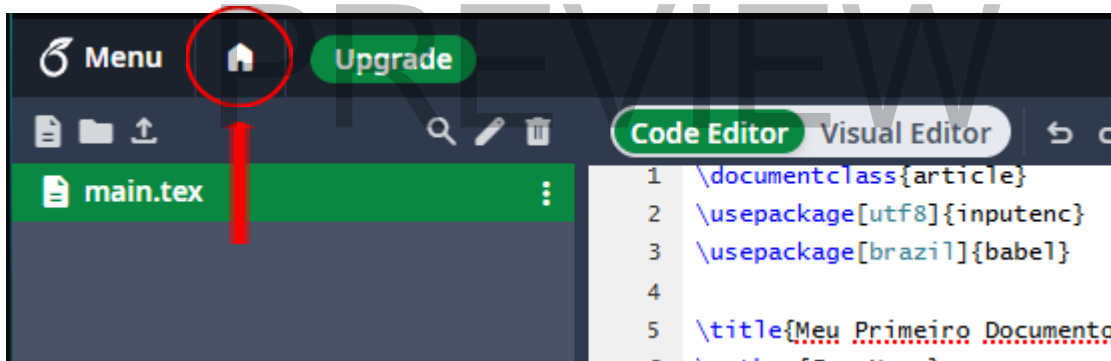


## 2.4 Usando templates LaTeX no Overleaf

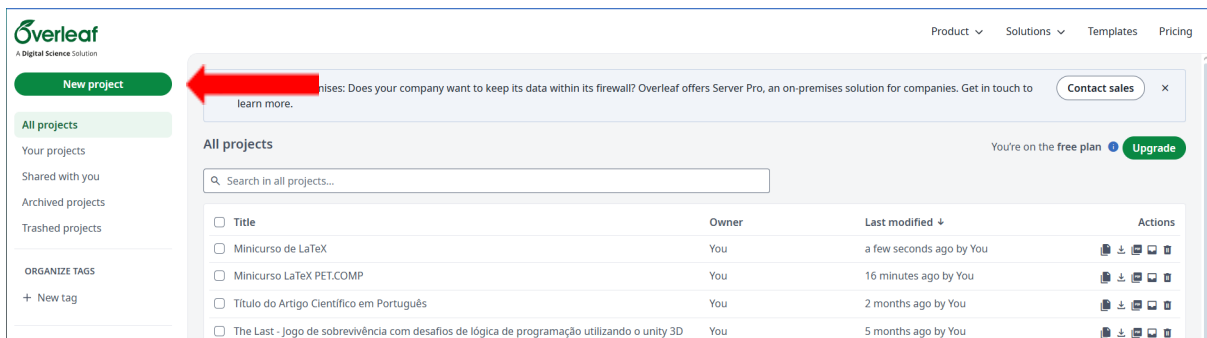
Agora que você já tem costume com o básico do LaTeX e sua interface, podemos começar a usar templates! Separamos um template para este minicurso, que contém várias dicas e exercícios para ajudar a complementar o material presente nesta apostila. O mesmo pode ser baixado neste link: [baixar template](#). Ao acessá-lo, irá lhe redirecionar para a seguinte tela:



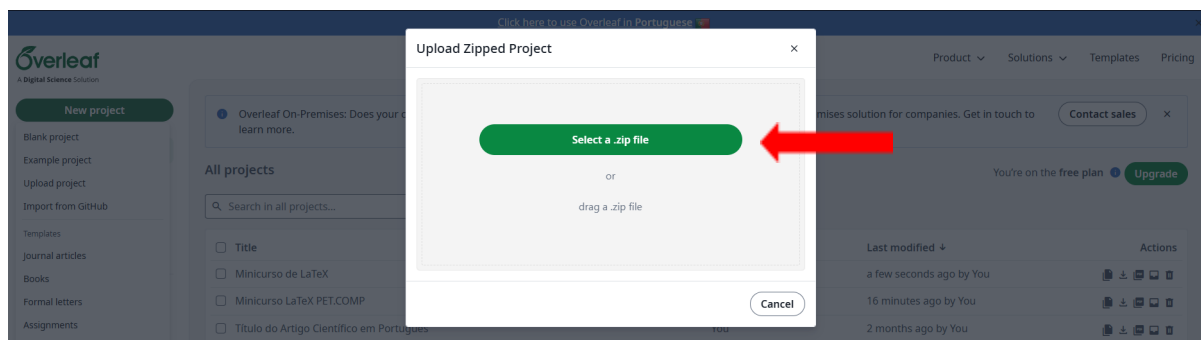
Estando nela, basta clicar na opção de baixar, no canto superior direito da tela. Isso irá baixar um .zip com todos os arquivos para o template ser usado no Overleaf. Assim, basta voltar ao Overleaf, e abrir o template. O Overleaf já aceita diretamente o arquivo .zip, assim não é necessário nem descompactar os arquivos! Segue imagem abaixo de como fazer o upload da pasta .zip:



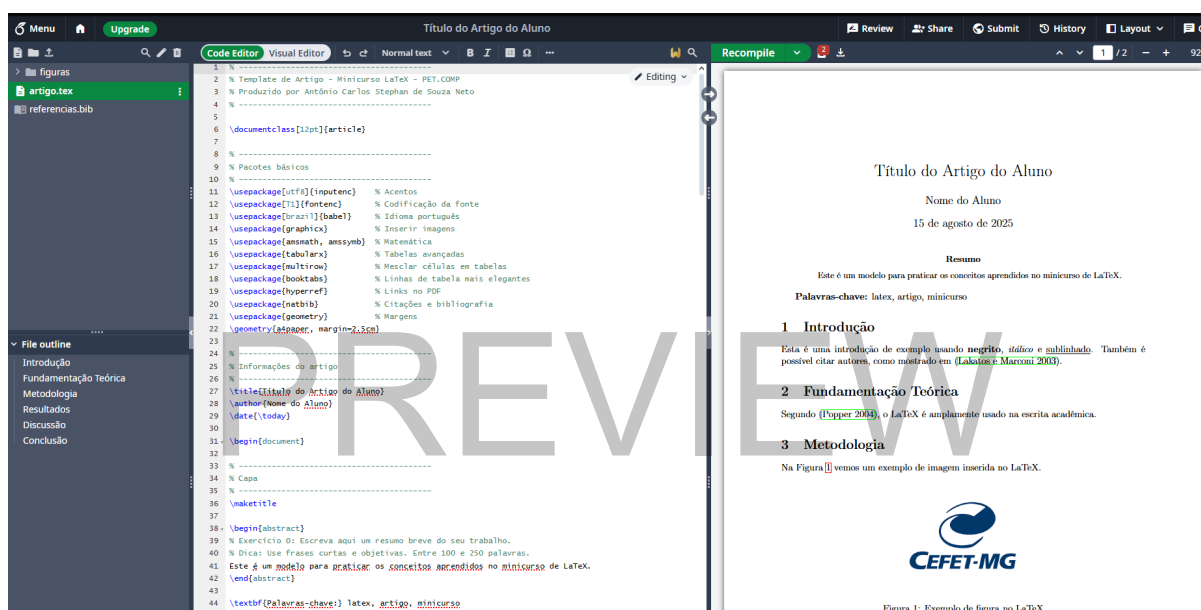
Clique na opção de **Home** e a seguinte tela irá aparecer:



Agora clique em **New Project**, e selecione a opção **Upload Project**. Com isso, a tela de escolha de pasta irá aparecer, basta arrastar a pasta .zip baixada ou a selecionar, conforme imagem abaixo:



Após selecionar a pasta .zip, o Overleaf irá automaticamente carregar e abrir o projeto do template, e você deve ver a seguinte tela:



Parabéns! Você conseguiu utilizar seu primeiro **template LaTeX**! Ao produzir artigos científicos para diversas feiras e congressos, os mesmos costumam disponibilizar templates LaTeX, já configurados com o padrão de artigo esperado pela feira/congresso. Então aprender a utilizar estes templates é de extrema utilidade, esteja você começando agora sua carreira acadêmica ou já nela há algum tempo. A partir de agora, iremos utilizar este template para acompanhamento durante o minicurso e também alguns exercícios para treinarmos enquanto aprendemos!

## 2.5 Primeiro Exercício - Primeiros Passos

Abra o arquivo **artigo.tex** e:

1. Altere o título e autor para o título do seu projeto, caso participe de algum, ou apenas “Minicurso Introdutório LaTeX”. O autor será seu nome.
2. Escreva um resumo simples no ambiente **abstract**, entre 100 e 250 palavras.
3. Inclua **3 palavras-chave** logo após o resumo.

## 3 Estrutura e Comandos

Nesta seção iremos tratar de estruturas e comandos essenciais para construção de textos, principalmente artigos, em LaTeX. Iremos tratar de pacotes úteis, formatação de texto, seções e listas.

### 3.1 Pacotes

- `\usepackage{graphicx}` → Utilizado quando queremos inserir imagens
- `\usepackage{amsmath}` → Necessário para quando queremos utilizar de matemática avançada em nosso texto (como **matrizes**, por exemplo)
- `\usepackage{hyperref}` → Possibilita que usemos links clicáveis ao longo do texto
- `\usepackage{geometry}` → Margens personalizadas para nosso texto
- `\usepackage{tabularx}` → Necessário para tabelas mais avançadas

### 3.2 Títulos e Seções

- `\section{Introdução}` → Seção de página (**Ex:** 1 Teste de Seção)
- `\subsection{Objetivos}` → Subseção de página (**Ex:** 1.1 Teste de Subseção)
- `\subsubsection{Metodologia}` → Subsubseção de página (**Ex:** 1.1.1 Teste de Subsubseção)

### 3.3 Formatação de Texto

O LaTeX permite formatar o texto usando comandos simples:

| Formatação | Código                         | Resultado    |
|------------|--------------------------------|--------------|
| Negrito    | <code>\textbf{Texto}</code>    | <b>Texto</b> |
| Itálico    | <code>\textit{Texto}</code>    | <i>Texto</i> |
| Sublinhado | <code>\underline{Texto}</code> | <u>Texto</u> |

#### 3.3.1 Segundo Exercício - Formatação de Texto

Na seção **Introdução** do `artigo.tex`:

1. Escreva um parágrafo usando cada tipo de formatação.
2. Use `\\` para quebrar uma linha propositalmente no meio do parágrafo.

### 3.4 Listas

No LaTeX, listas são criadas por meio de ambientes específicos que controlam automaticamente o espaçamento e a formatação. O ambiente `itemize` gera listas com marcadores (não numeradas), enquanto `enumerate` cria listas numeradas automaticamente. Cada item da lista é iniciado com o comando `\item`. É possível também criar listas aninhadas, colocando um ambiente de lista dentro de outro, o que é útil para subitens ou classificações hierárquicas, por exemplo.

Listas podem ser numeradas, que serão listas com números ou letras indicando ordem, como por exemplo:

1. Este é um exemplo de lista numerada
  - a. Este também é um exemplo de lista numerada

Ou podem ser listas não numeradas, que serão listas que não usam números ou letras, mas sim círculos ou quadrados por exemplo:

- Este é um exemplo de lista não numerada
  - Este também é um exemplo de lista não numerada

Tendo isto em mente, em LaTeX, quando queremos usar **listas numeradas**, podemos usar o seguinte código:

```
\begin{enumerate}
\item Passo 1
\item Passo 2
\end{enumerate}
```

Agora, caso queiramos listas não numeradas, podemos usar este:

```
\begin{itemize}
\item Item 1
\item Item 2
\end{itemize}
```

Na imagem abaixo, perceba como fica aplicado ao **editor de código** os códigos acima de listas e na próxima imagem a **pré-visualização de PDF** dos mesmos. Em **vermelho** destacamos o código e o resultado, para facilitar para você encontrar os mesmos nas imagens.

```
11
12 \begin{document}
13
14 \maketitle
15
16 Olá, mundo! Este é meu primeiro documento em LaTeX.\\
17
18
19 \begin{enumerate}
20 \item Passo 1
21 \item Passo 2
22 \end{enumerate}
23
24 \begin{itemize}
25 \item Item 1
26 \item Item 2
27 \end{itemize}
28
29
30
31 \end{document}
32
```

Recompile    120%  ...



# Meu Primeiro Documento

Seu Nome

23 de agosto de 2025

Olá, mundo! Este é meu primeiro documento em LaTeX.

1. Passo 1

2. Passo 2

- Item 1

- Item 2

### 3.4.1 Terceiro Exercício - Listas

Na seção **Conclusão**:

1. Crie uma lista numerada com 3 itens.
2. Crie uma lista com marcadores com 3 itens.

## 4 Equações Matemáticas

### 4.1 Inline e Destacadas

As equações podem ser inseridas diretamente no texto (modo inline) ou destacadas em linhas próprias para dar maior destaque visual (modo display). Para equações mais complexas, existem ambientes como **align**, **gather** e **equation**, que permitem alinhar termos, numerar fórmulas e facilitar a leitura.

Equações inline são usadas quando a fórmula faz parte de uma frase e precisa se integrar ao fluxo do texto, sendo delimitadas por `$ ... $`. Já as equações destacadas são colocadas entre `\[ ... \]` ou no ambiente **equation**, sendo exibidas centralizadas e, opcionalmente, numeradas. A escolha entre inline e destacada depende da relevância e do espaço que a expressão ocupa na explicação.

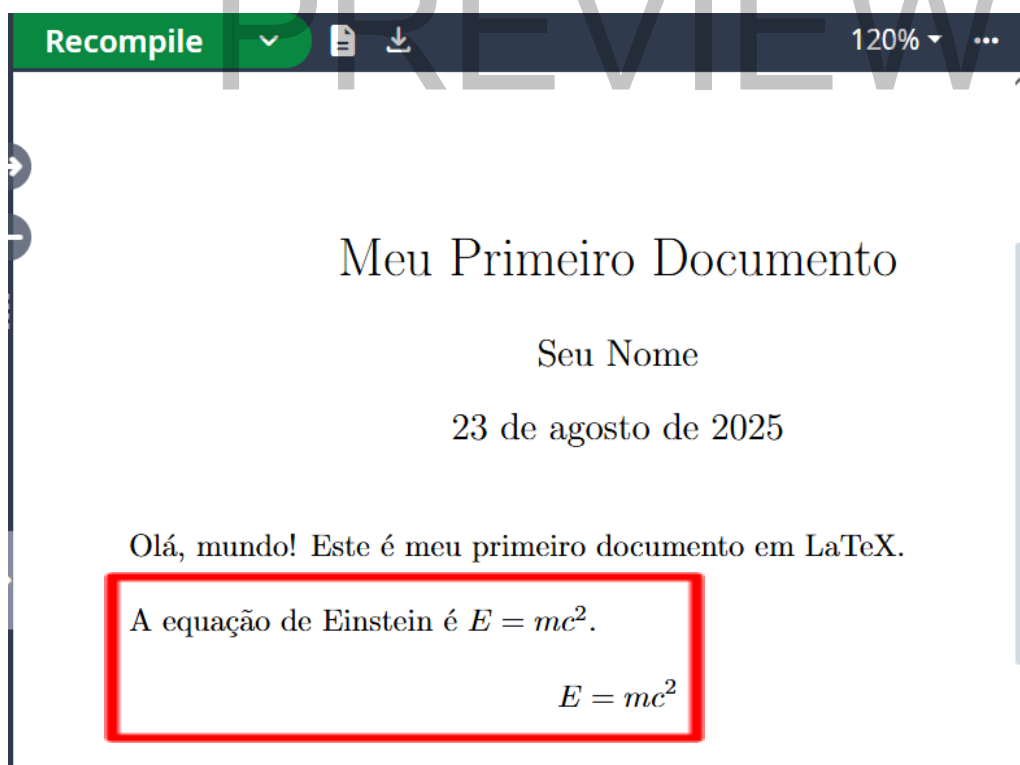
Exemplo de código:

A equação de Einstein é `$E = mc^2$`. → Inline

`\[`  
`E = mc^2` → Destacadas  
`\]`

Na imagem abaixo, perceba como fica aplicado ao **editor de código** os códigos acima de equações inline e destacadas e na próxima imagem a **pré-visualização de PDF** dos mesmos. Em **vermelho** destacamos o código e o resultado, para facilitar para você encontrar os mesmos nas imagens.

```
11
12 \begin{document}
13
14 \maketitle
15
16 Olá, mundo! Este é meu primeiro documento em LaTeX.\\
17
18
19 A equação de Einstein é  $E = mc^2$ .
20
21 \[
22 E = mc^2
23 \]
24
25
26
27 \end{document}
28
```



## 4.2 Símbolos

O LaTeX oferece uma ampla biblioteca de símbolos matemáticos, incluindo letras gregas, operadores, relações e setas. Esses símbolos são acessados por comandos específicos,

geralmente precedidos por uma barra invertida, como `\alpha` ( $\alpha$ ), `\beta` ( $\beta$ ) e `\rightarrow` ( $\rightarrow$ ). Para operações matemáticas, comandos como `\times` ( $\times$ ), `\div` ( $\div$ ) e `\sqrt{}` ( $\sqrt{\phantom{x}}$ ) garantem precisão visual e consistência tipográfica. Abaixo segue um guia rápido dos símbolos que podem ser utilizados por você em LaTeX:

| Categoria                  | Comando LaTeX            | Resultado            |
|----------------------------|--------------------------|----------------------|
| Letras gregas (minúsculas) | <code>\alpha</code>      | $\alpha$             |
|                            | <code>\beta</code>       | $\beta$              |
|                            | <code>\gamma</code>      | $\gamma$             |
|                            | <code>\lambda</code>     | $\lambda$            |
|                            | <code>\pi</code>         | $\pi$                |
|                            | <code>\sigma</code>      | $\sigma$             |
| Letras gregas (maiúsculas) | <code>\Gamma</code>      | $\Gamma$             |
|                            | <code>\Delta</code>      | $\Delta$             |
|                            | <code>\Sigma</code>      | $\Sigma$             |
|                            | <code>\Omega</code>      | $\Omega$             |
| Operadores matemáticos     | <code>\times</code>      | $\times$             |
|                            | <code>\pm</code>         | $\pm$                |
|                            | <code>\sqrt{}</code>     | $\sqrt{\phantom{x}}$ |
| Símbolos de relação        | <code>\leq</code>        | $\leq$               |
|                            | <code>\geq</code>        | $\geq$               |
|                            | <code>\neq</code>        | $\neq$               |
| Setas                      | <code>\rightarrow</code> | $\rightarrow$        |
|                            | <code>\leftarrow</code>  | $\leftarrow$         |

### 4.3 Matrizes

Matrizes em LaTeX são criadas dentro de ambientes como `matrix`, `pmatrix` (com parênteses), `bmatrix` (com colchetes) e `vmatrix` (com barras verticais). Os elementos são separados por `&` para indicar colunas e `\\` para indicar o fim de cada linha. O LaTeX garante alinhamento perfeito e espaçamento uniforme, tornando fácil a leitura de estruturas numéricas ou simbólicas.

Exemplo de código de matriz abaixo:

```

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

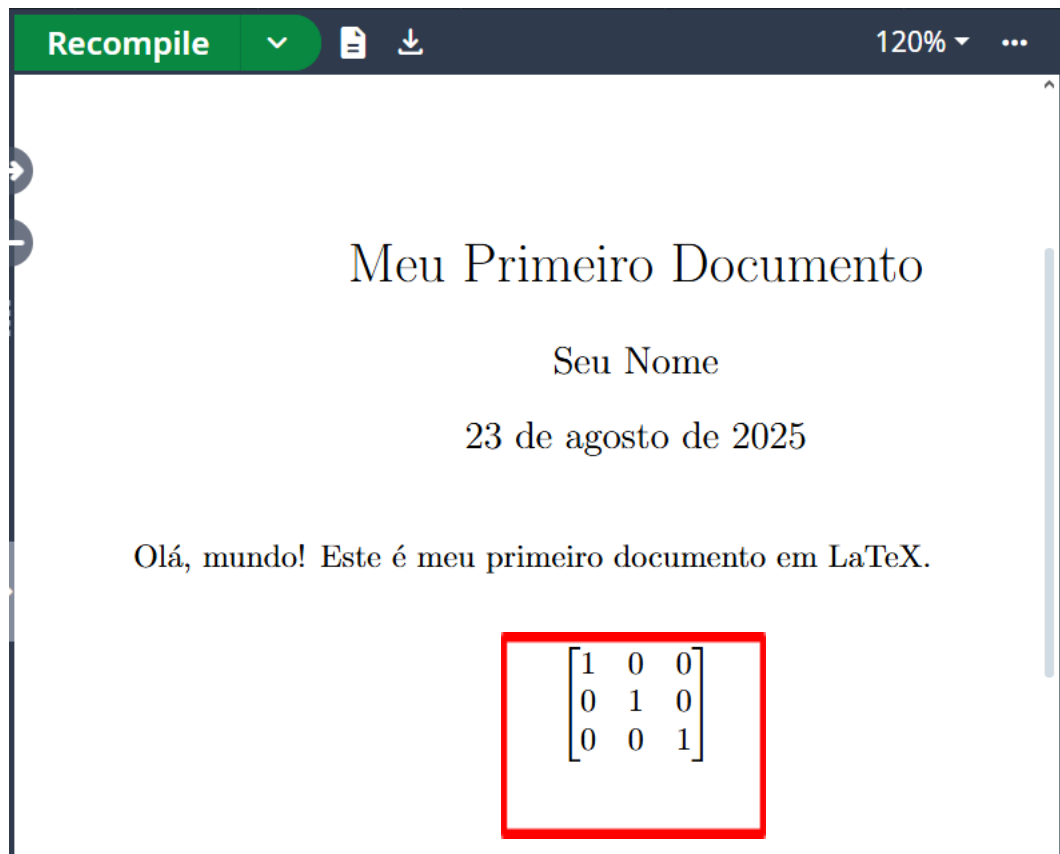
```

Na imagem abaixo, perceba como fica aplicado ao **editor de código** o código acima de matriz e na próxima imagem a **pré-visualização de PDF** da mesma. Em **vermelho** destacamos o código e o resultado, para facilitar para você encontrar os mesmos nas imagens.

```

11
12 \begin{document}
13
14 \maketitle
15
16 Olá, mundo! Este é meu primeiro documento em LaTeX.\\
17
18
19 \[
20 \begin{bmatrix}
21 1 & 0 & 0 \\
22 0 & 1 & 0 \\
23 0 & 0 & 1
24 \end{bmatrix}
25 \]
26
27
28
29 \end{document}

```



#### 4.4 Quarto Exercício - Fórmulas

Na seção **Discussão**:

1. Insira a fórmula de Bhaskara.
2. Adicione uma matriz 3x3.
3. Use pelo menos dois símbolos gregos.

### 5 Inserção de Imagens

O LaTeX permite a inclusão de imagens com alta qualidade e controle de posicionamento por meio do pacote **graphicx**. O comando **\includegraphics** insere o arquivo de imagem, enquanto parâmetros como **width** e **height** ajustam seu tamanho. Para descrever a imagem e criar referências cruzadas no texto, utiliza-se o ambiente **figure** junto de **\caption** (legenda) e **\label** (etiqueta de referência). As imagens devem estar em formatos compatíveis, como **.png**, **.jpg** ou **.pdf**.

Segue um exemplo de código a ser usado para inserção de uma imagem:

**\usepackage{graphicx}** → Pacote necessário para que possamos inserir imagens

**\begin{figure}[h]**  
**\centering**

```
\includegraphics[width=0.3\textwidth]{figuras/logo_cefet.png}
\caption{Exemplo de figura no LaTeX}
\label{fig:exemplo}
\end{figure}
```

Na imagem abaixo, perceba como fica aplicado ao **editor de código** o código acima de inserção de imagem e na próxima a **pré-visualização de PDF** do mesmo. Em **vermelho** destacamos o código e o resultado, para facilitar para você encontrar os mesmos nas imagens.



# Meu Primeiro Documento

Seu Nome

23 de agosto de 2025

Olá, mundo! Este é meu primeiro documento em LaTeX.

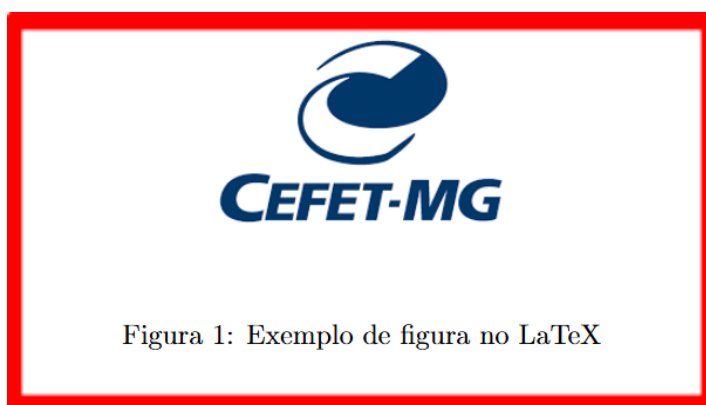


Figura 1: Exemplo de figura no LaTeX

## 5.1 Manipulando largura e altura de imagens

Ao inserir figuras no LaTeX, é comum que o tamanho original não seja o mais adequado para o documento. Para resolver isso, usamos os parâmetros **width** e **height** no comando **\includegraphics**.

- **width** controla a largura da imagem.
- **height** controla a altura da imagem.

Esses valores podem ser definidos em unidades fixas (como **cm**, **mm** ou **pt**) ou relativos ao tamanho do texto. Por exemplo, **\includegraphics[width=8cm]{figuras/exemplo.png}** define a largura em centímetros, enquanto **\includegraphics[width=0.5\textwidth]{figuras/exemplo.png}** ajusta a imagem para ocupar metade da largura do texto.

É possível também combinar **width** e **height**, mas nesse caso a imagem pode ser distorcida, já que o LaTeX vai forçar o ajuste sem manter a proporção original. Para evitar isso, existe a opção **keepaspectratio**, que preserva a proporção da imagem:

**\includegraphics[width=0.6\textwidth, keepaspectratio]{figuras/exemplo.png}**



**Dica:** Sempre prefira trabalhar com tamanhos relativos (**\textwidth**, **\linewidth**) em vez de valores fixos, pois isso garante que o documento fique mais flexível em diferentes formatos de página.

## 5.2 Quinto Exercício - Imagens

Na seção **Metodologia**:

1. Adicione uma imagem na pasta **figuras**.
2. Insira a figura no documento com legenda e etiqueta (**\label**).
3. Faça referência cruzada a ela no texto usando **\ref**.

## 6 Tabelas

O ambiente mais básico para criar tabelas em LaTeX é o **tabular**. Nele, você começa definindo quantas colunas a tabela terá e como cada uma delas será alinhada — à esquerda (**l**), ao centro (**c**) ou à direita (**r**).

Dentro desse ambiente, usamos o símbolo **&** para pular de uma coluna para outra e **\\** para dizer que aquela linha terminou. Se quiser acrescentar uma linha horizontal separando os dados, o comando é **\hline**. Já as bordas verticais, aquelas linhas que separam colunas, podem ser adicionadas usando o caractere **|** na configuração inicial da tabela.

Agora, se a sua intenção é deixar a tabela mais completa — com um título e a possibilidade de fazer referência a ela no texto —, vale a pena colocá-la dentro do ambiente **table**. Aí entram dois comandos importantes: **\caption{...}**, que insere a legenda, e **\label{...}**, que cria uma “etiqueta” para chamar essa tabela em qualquer parte do documento, a citando.

Além do **tabular**, existem outras formas de representar e formatar tabelas no LaTeX, como:

- **longtable** → para tabelas que se estendem por várias páginas
- **tabularx** → permite controle da largura das colunas
- **array** → voltado para tabelas matemáticas e matrizes
- **booktabs** → oferece linhas horizontais mais elegantes e profissionais
- **multirow** e **multicolumn** → permitem mesclar células vertical e horizontalmente

 **Observação:** Não iremos tratar neste minicurso sobre outros tipos de tabela além **tabular** e **tabularx**. Porém, caso queira estudar os outros tipos e suas implementações, recomendamos que acesse os portais de aprendizado que estão listados na seção de **Dicas Finais e Recursos** desta apostila.

Segue abaixo um código de exemplo de uma tabela simples usando **tabular**:

```
\begin{table}[h]
\centering
\caption{Exemplo de tabela simples}
```

```

\begin{tabular}{|c|c|c|}
\hline
Coluna 1 & Coluna 2 & Coluna 3 \\ \hline
A & B & C \\ \hline
1 & 2 & 3 \\ \hline
\end{tabular}
\end{table}

```

Na imagem abaixo, perceba como fica aplicado ao **editor de código** o código acima de inserção de uma tabela simples, usando **tabular** e na próxima a **pré-visualização de PDF** da mesma. Em **vermelho** destacamos o código e o resultado, para facilitar para você encontrar os mesmos nas imagens.

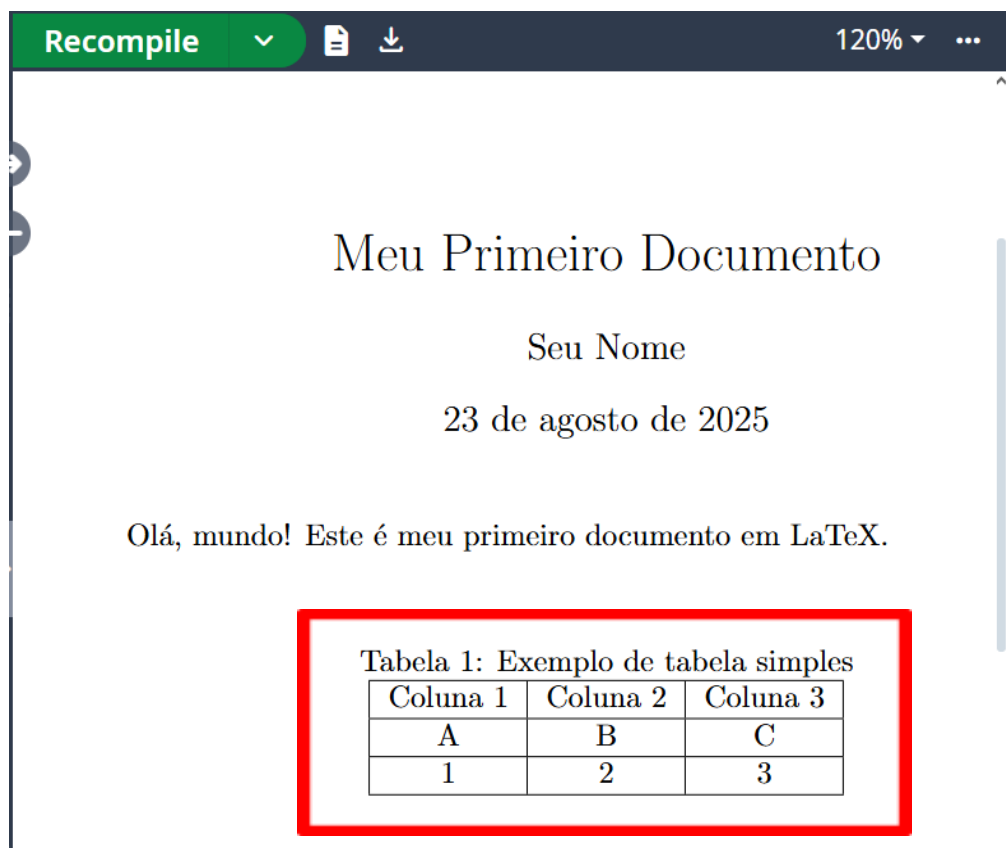
The screenshot shows a LaTeX editor interface. On the left, a list of line numbers from 11 to 30. The code editor on the right contains the following LaTeX code:

```

11
12 \begin{document}
13
14 \maketitle
15
16 Olá, mundo! Este é meu primeiro documento em LaTeX.\\
17
18
19 \begin{table}[h]
20 \centering
21 \caption{Exemplo de tabela simples}
22 \begin{tabular}{|c|c|c|}
23 \hline
24 Coluna 1 & Coluna 2 & Coluna 3 \\ \hline
25 A & B & C \\ \hline
26 1 & 2 & 3 \\ \hline
27 \end{tabular}
28 \end{table}
29
30 \end{document}

```

A red rectangular box highlights the table-related code from line 19 to line 28. To the right of the code editor is a preview window showing the rendered PDF output. The preview displays the title "Olá, mundo! Este é meu primeiro documento em LaTeX." followed by a table. The table has three columns and three rows of data, matching the code. A large, semi-transparent "PREVIEW" watermark is visible across the center of the image.



Segue abaixo um código de exemplo de uma tabela avançada usando **tabularx**:

```
\begin{table}[h]
\centering
\caption{Exemplo de tabela com tabularx}
\begin{tabularx}{\textwidth}{|X|X|X|}
\hline
\textbf{Descrição} & \textbf{Detalhes} & \textbf{Observação} \\ \hline
Texto mais longo ajustado automaticamente & Texto médio & Texto curto \\ \hline
Outro exemplo de célula & Mais conteúdo & Última coluna \\ \hline
\end{tabularx}
\end{table}
```

Na imagem abaixo, perceba como fica aplicado ao **editor de código** o código acima de inserção de uma tabela avançada, usando **tabularx** e na próxima a **pré-visualização de PDF** da mesma. Em **vermelho** destacamos o código e o resultado, para facilitar para você encontrar os mesmos nas imagens.

```
11
12 \begin{document}
13
14 \maketitle
15
16 Olá, mundo! Este é meu primeiro documento em LaTeX.\\
17
18 \begin{table}[h]
19 \centering
20 \caption{Exemplo de tabela com tabularx}
21 \begin{tabularx}{\textwidth}{|X|X|X|}
22 \hline
23 \textbf{Descrição} & \textbf{Detalhes} & \textbf{Observação} \\
24 \hline
25 Texto mais longo ajustado automaticamente & Texto médio & Texto curto \\
26 \hline
27 Outro exemplo de célula & Mais conteúdo & Última coluna \\
28 \hline
29 \end{tabularx}
30 \end{table}
31
32 \end{document}
```

Recompile 100%

# Meu Primeiro Documento

Seu Nome

23 de agosto de 2025

Olá, mundo! Este é meu primeiro documento em LaTeX.

Tabela 1: Exemplo de tabela com tabularx

| Descrição                                 | Detalhes      | Observação    |
|-------------------------------------------|---------------|---------------|
| Texto mais longo ajustado automaticamente | Texto médio   | Texto curto   |
| Outro exemplo de célula                   | Mais conteúdo | Última coluna |

 **Dica:** Lembre-se que para utilizar o pacote `\usepackage{tabularx}` para conseguir criar tabelas com **tabularx**.

## 7 Citações e Referências Bibliográficas

Uma das grandes vantagens do LaTeX é a forma como ele lida com referências bibliográficas. Em vez de misturar as citações com o texto principal, ele mantém tudo separado e organizado em um arquivo próprio, no formato **.bib** — o BibTeX.

Isso significa que você não precisa perder tempo ajustando cada referência manualmente. Basta inserir os dados no arquivo **.bib** e, quando for citar no texto, o LaTeX puxa as informações e formata automaticamente, seguindo o estilo que você escolher. E o destaque nisso é: se a instituição ou periódico (para o qual você está produzindo um artigo) pedir outro padrão, basta mudar o estilo no documento que todas as referências são atualizadas de uma só vez.

### 7.1 Estrutura do **.bib**

O arquivo **.bib** é onde você armazena todas as referências que deseja usar no seu documento. Ele funciona como um banco de dados bibliográfico, onde cada referência é identificada por uma **chave única** (ex.: **silva2020**) que será usada para citar no texto.

Segue abaixo um modelo básico de uma entrada no arquivo **.bib**:

```
@tipo{chave,  
  author   = {Nome do Autor},  
  title    = {Título da Obra},  
  year     = {Ano de Publicação},  
  publisher = {Editora ou Instituição},  
  address  = {Cidade}  
}
```

No exemplo acima:

- **@tipo** define o tipo de referência (livro, artigo, site, etc.).
- **chave** é como você irá chamar essa referência no texto.
- As linhas seguintes contêm os campos com as informações da obra.

Abaixo segue um exemplo real:

```
@book{silva2020,  
  
  author   = {João da Silva},  
  
  title    = {Introdução ao LaTeX},  
  
  year     = {2020},
```

```

publisher = {Editora Acadêmica},

address  = {São Paulo}

}

```

Na imagem abaixo, perceba como fica aplicado ao **editor de código** o código de exemplo dado acima no arquivo **.bib**.

```

1  @book{silva2020,
2      author    = {João da Silva},
3      title     = {Introdução ao LaTeX},
4      year      = {2020},
5      publisher = {Editora Acadêmica},
6      address   = {São Paulo}
7  }
8

```

### 7.1.1 Principais tipos de referência no BibTeX

| Tipo           | Descrição                                       | Campos mais comuns                                  |
|----------------|-------------------------------------------------|-----------------------------------------------------|
| @article       | Artigo de revista ou periódico                  | author, title, journal, year, volume, number, pages |
| @book          | Livro completo                                  | author ou editor, title, publisher, year            |
| @inbook        | Capítulo de livro                               | author, title, chapter, pages, publisher, year      |
| @incollection  | Parte de um livro (coleção)                     | author, title, booktitle, publisher, year, pages    |
| @inproceedings | Artigo em anais de congresso                    | author, title, booktitle, year, pages               |
| @manual        | Documentação técnica                            | title, author, organization, year                   |
| @mastersthesis | Dissertação de mestrado                         | author, title, school, year                         |
| @phdthesis     | Tese de doutorado                               | author, title, school, year                         |
| @misc          | Categoria genérica (sites, apresentações, etc.) | author, title, howpublished, year                   |
| @techreport    | Relatório técnico                               | author, title, institution, year                    |

💡 **Observação:** Essa tabela traz apenas os tipos mais usados. Há muitos outros disponíveis. Para conhecer todos, consulte a [documentação oficial](#) do BibTeX ou pesquise no CTAN.

## 7.2 Inserindo Referências

Depois de criar seu arquivo **.bib** e preenchê-lo com as entradas necessárias, é hora de conectá-lo ao seu documento LaTeX. Isso é feito no final do arquivo **.tex** usando:

```
\bibliographystyle{abbrv}
\bibliography{referencias}
```

Seguem algumas explicações sobre o código acima:

- **\bibliographystyle{abbrv}** define o estilo de formatação das referências (existem várias opções como **plain**, **abbrv**, **apalike**, entre outras).
- **\bibliography{referencias}** indica o nome do arquivo **.bib** (sem a extensão).

Se o arquivo se chamar **referencias.bib** e estiver na mesma pasta do **.tex**, o LaTeX irá encontrá-lo e gerar a lista automaticamente.

## 7.3 Citando no Texto

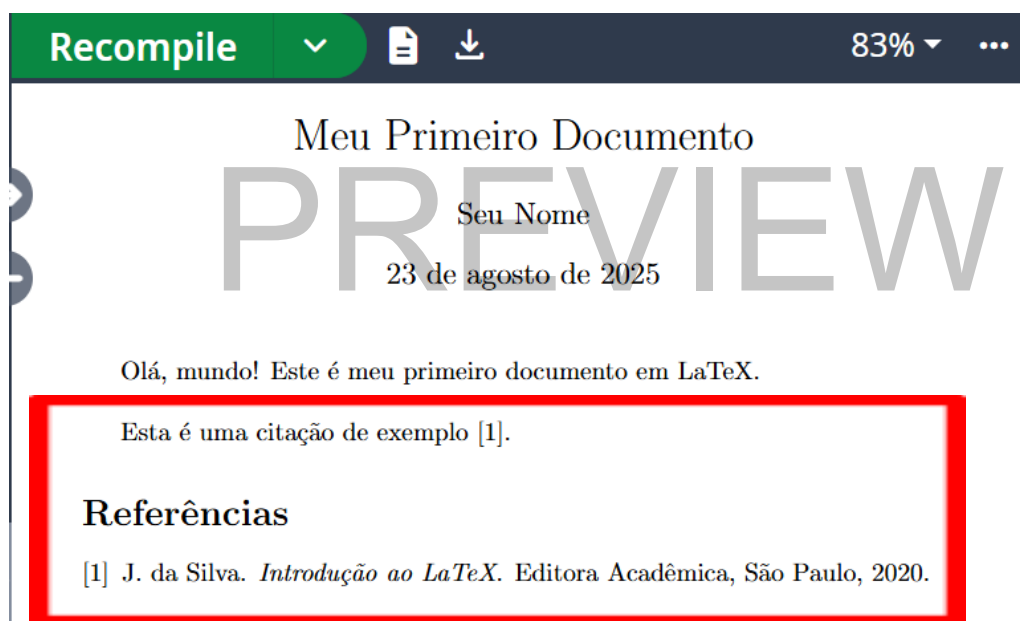
Para citar uma referência dentro do texto, basta usar a chave que você definiu no **.bib**:

- **\cite{silva2020}** → gera uma citação entre parênteses, por exemplo: **(Silva, 2020)**.
- **\citeonline{silva2020}** → gera a citação incorporada ao texto, por exemplo: **Silva (2020)** afirma que...

💡 **Dica:** Ao citar, prefira sempre usar a chave no lugar de escrever manualmente o nome e o ano. Assim, qualquer mudança no estilo ou no conteúdo da referência será aplicada automaticamente em todo o documento.

Na imagem abaixo, perceba como fica aplicado ao **editor de código** a citação sendo usada e o LaTeX reconhecendo a citação e listando a referência que foi utilizada. Na próxima imagem há a **pré-visualização de PDF** da mesma. Em **vermelho** destacamos o código e o resultado, para facilitar para você encontrar os mesmos nas imagens.

```
11
12 \begin{document}
13
14 \maketitle
15
16 Olá, mundo! Este é meu primeiro documento em
    LaTeX.\\
17
18 Esta é uma citação de exemplo \cite{silva2020}.
19
20 \bibliographystyle{abbrv}
21 \bibliography{references}
22
23
24 \end{document}
25
```



## 7.4 Lista de Referências

No momento da compilação, o LaTeX gera a lista completa de referências com base apenas nas obras que foram realmente citadas no texto. Isso evita que trabalhos fiquem poluídos com referências não utilizadas.

Se você adicionar uma referência no **.bib** mas não usá-la no texto, ela não aparecerá na lista final. Para incluir obras consultadas mas não citadas, use o comando:

`\nocite{chave}`

Ou, para incluir todas as referências do `.bib`, mesmo sem citação:

```
\nocite{*}
```

## 7.5 Sexto Exercício - Referências

1. No arquivo `referencias.bib`, adicione três novas referências de tipos diferentes (`@book`, `@article` e `@misc`).
2. Cite cada uma no texto, usando tanto `\cite` quanto `\citeonline`.
3. Compile e verifique a lista gerada automaticamente ao final do documento.

## 8 Dicas Finais e Recursos

- Compilar frequentemente para detectar erros cedo.
- Usar comentários `%` para se orientar no código.
- Recursos: [Overleaf Learn](#), [CTAN](#), [Detsexify](#).

PREVIEW