

# Preparação de Documentos Acadêmicos com $\text{\LaTeX}$

## Aula 01 - $\text{\LaTeX}$ básico, Ambiente Overleaf, Formatação, Tabelas, Imagens e Posicionamento de Elementos

Prof. Jean Paul Barddal

Programa de Pós-Graduação em Informática (PPGIa)

Pontifícia Universidade Católica do Paraná (PUCPR)



# Agenda

- 1 Introdução
- 2 Ambiente Overleaf
- 3 Formatando Arquivos com  $\text{\LaTeX}$
- 4 Imagens
- 5 Tabelas
- 6 Posicionamento de Elementos

# Agenda

## 1 Introdução

## 2 Ambiente Overleaf

## 3 Formatando Arquivos com $\text{\LaTeX}$

## 4 Imagens

## 5 Tabelas

## 6 Posicionamento de Elementos

# Jean Paul Barddal

- Professor do **Programa de Pós-Graduação em Informática (PPGIA)** - <https://www.ppgia.pucpr.br/pt/> e do curso de **Ciência da Computação** (Escola Politécnica)
- Pesquisador na área de Aprendizagem de Máquina, Ciência de Dados e Inteligência Artificial
- <http://www.jpbarddal.com.br> (*en-us*)
- Material do Curso: Home > teaching > short duration & in-company > L<sup>A</sup>T<sub>E</sub>X

# Funcionamento do Curso

## Aulas

- Majoritariamente práticas
- Em muitos momentos, os slides servirão de apoio e consulta pós-aula
- *Check-in* (**NÃO** é necessário realizar *check-out*!)
- *Chat*: Suporte das profas. Anna e Kamila

## Agenda

- 22/05/2025 - 14h às 17h
- 29/05/2025 - 14h às 17h
- 05/06/2025 - 14h às 17h
- 12/06/2025 - 14h às 17h

# Mentimeter

- Antes de iniciarmos, vamos responder duas perguntas rapidamente no mentimeter
- Acesse usando seu smartphone:



# Avisos

- 1 Esse **NÃO** é um curso voltado à preparação de documentos usando ABNT
  - Neste momento, ao usar  $\text{\LaTeX}$ , presume-se que o documento que desejamos preparar já possui um formato (*template*) disponível
  - Isso é muito comum na maior parte de conferências, periódicos, livros, etc.
- 2 É nosso objetivo também perceber as diferenças do  $\text{\LaTeX}$  e outras ferramentas de texto, como o Word, identificando prós e contras de cada alternativa

- Sistema de preparação de documentos
- Criado por cientistas, para cientistas
- Amplamente usado para confecção de documentos científicos (artigos, apresentações, dissertações, teses, provas, etc)
  - Essa apresentação, por exemplo, foi feita em  $\text{\LaTeX}$



# Benefícios do $\text{\LaTeX}$

- Robusto, poucos problemas de versão
- Separação de conteúdo e estilo: foque no conteúdo e não em como o conteúdo vai aparecer. Deixe o  $\text{\LaTeX}$  trabalhar.
- Como usuário, não deve ser sua responsabilidade definir o estilo (*template*), mas sim usar um template disponível
- Flexibilidade para alternar entre estilos
- Qualidade Gráfica
- Equações, Algoritmos, Fórmulas Químicas
- Referências, Citações, Numeração, etc



# As vezes, nem é questão de benefício...

Não é incomum se deparar com situações como essa:

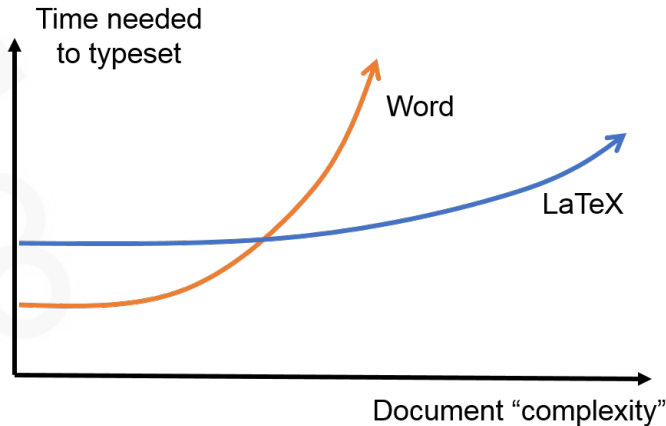
MSWord: The journal does not provide MsWord templates for your manuscript.

Authors must prepare their manuscript in LaTeX and use the bibliography style file sn-basic.bst which is included in the [Springer Nature Article Template](#).

# Comparativo com MS Word

Ferramenta	Word	LaTeX
O que é?	Processador de palavras	Linguagem de Marcação
Formatação	Flexível, versátil e amigável	Controle preciso (normalmente <b>não</b> cabe ao usuário)
Customização	Limitado	Grande gama de pacotes
Edição	WYSIWYG ( <i>What you see is what you get</i> )	Dois momentos (Edição e compilação)
Saída	.DOC, .DOCX, .DOT, ...	PDF (alta qualidade)

# Complexidade e Tempo de Confecção





# Formas de Instalação e Uso

O uso de  $\text{\LaTeX}$  requer a instalação do “core”, bem como de eventuais pacotes necessários para funcionalidades específicas. A instalação e o uso podem ser feitos das seguintes formas:

## 1 Local

- Windows: Texlive - <https://www.tug.org/texlive>
- MacOS: MacTeX - <https://www.tug.org/mactex/>
- Linux: Texlive - <https://www.tug.org/texlive/>

## 2 Remoto

- Overleaf: <https://overleaf.com>
- Texpage: <https://www.texpage.com>

# Ambiente Overleaf

## Funcionalidades:

- Escrita colaborativa (sincronização, comentários, chat)
- Editor “Visual”
- Histórico de Alterações
- Sincronização com Dropbox, Git, Mendeley, Zotero, Papers, etc
- (Muitos) Templates



# Ambiente Overleaf - Versão Gratuita

O ambiente Overleaf fornece uma versão gratuita. Ela basta para grande parte dos projetos, contudo, apresenta algumas limitações:

- Um colaborador por projeto
- Timeout de compilação
- Não há busca de referências, histórico de alterações
- Não há integrações (Dropbox, Git, etc)
- Paleta de símbolos, etc



# Agenda

- 1 Introdução
- 2 Ambiente Overleaf
- 3 Formatando Arquivos com  $\text{\LaTeX}$
- 4 Imagens
- 5 Tabelas
- 6 Posicionamento de Elementos

# Começando...

- Um documento em  $\text{\LaTeX}$  possui a seguinte estrutura básica:

```
1 \documentclass{article} % tipo do documento
2 \begin{document}
3   Oi, mundo! % vamos colocar nosso conteudo aqui
4 \end{document}
```

- Todo documento inicia com `\documentclass`
- Comandos iniciam com barra invertida `\`
- O argumento dentro de chaves `{ }` diz o tipo de documento que vamos preparar (article)
- Um símbolo de percentual `%` denota um comentário -  $\text{\LaTeX}$  vai ignorar o restante da linha

# Vamos ao Overleaf

- Em breve, vamos ao Overleaf
- Lá, vamos criar nosso primeiro documento (artigo) usando  $\text{\LaTeX}$
- Aproveitaremos e trabalharemos com diversos comandos, incluindo:
  - Estruturação (seções, parágrafos, etc)
  - Formatação (fontes, espaçamento)
  - Imagem
  - Tabelas
  - Posicionamento de elementos
- Os slides servirão de apoio e estudo pós-aula
- Esse primeiro momento pode ser chocante, mas acredite, existirão muitos benefícios no longo prazo

# Lidando com Erros

- O L<sup>A</sup>T<sub>E</sub>X pode ter dificuldade ao compilar o seu documento. Caso isso ocorra e ele pare, aparecerá um erro, que você precisará corrigir
- Um exemplo comum é a digitação incorreta de comandos, por exemplo: `meph` ao invés de `emph` resultará em **undefined control sequence**, que denota que o comando usado não foi reconhecido

## Nota sobre Erros

- 1 Não entre em pânico, erros acontecem
- 2 Arrume erros tão logo eles apareçam
- 3 Caso apareçam múltiplos erros, vá ao primeiro pois os demais podem ser causados pelo primeiro

# Estruturação

- Dentro de documentos em L<sup>A</sup>T<sub>E</sub>X , podemos estruturar um documento considerando a seguinte taxonomia:

Nível	Comando
-1	<code>\part{nome}</code>
0	<code>\chapter{nome}</code>
1	<code>\section{nome}</code>
2	<code>\subsection{nome}</code>
3	<code>\subsubsection{nome}</code>
4	<code>\paragraph{nome}</code>
5	<code>\subparagraph{nome}</code>

- Essa taxonomia é genérica, portanto, diferentes templates podem usar apenas parte dessa estrutura ou ter comandos diferentes para estruturas específicas (raro)

# Listas

- No  $\text{\LaTeX}$ , existem duas formas principais de preparar listas:
  - Itemize: listas tradicionais
  - Enumerate: listas numeradas sequencialmente
- Permite **customizações de marcador e aninhamento**

# Listas com *itemize*

- Permite criar listas com *bullets*

```
1 \begin{itemize}
2   \item Primeiro item
3   \item Segundo item
4   \item[$\rightarrow$] Terceiro item, agora customizando o
      estilo do bullet
5   \item[!] Quarto item, também mudando o estilo do bullet
6 \end{itemize}
```

# Listas com *itemize*

- Primeiro item
- Segundo item
- Terceiro item, agora customizando o estilo do bullet
- ! Quarto item, também mudando o estilo do bullet



# Listas com *enumerate*

- Funcionamento similar ao do `itemize`, contudo, os pontos são enumerados

```
1 \begin{enumerate}
2   \item Primeiro passo
3   \item Segundo passo
4   \item Terceiro passo
5 \end{enumerate}
```

Resultado:

- 1 Primeiro passo
- 2 Segundo passo
- 3 Terceiro passo

# Exemplo de Aninhamento

- As construções de *itemize* e *enumerate* podem ser usadas de forma aninhada

```
1 \begin{itemize}
2   \item Word
3   \begin{itemize}
4     \item Processador de palavras (WYSIWYG)
5     \item Pago
6   \end{itemize}
7   \item \LaTeX
8   \begin{itemize}
9     \item Edição e Compilação
10    \item Foco em documentos acadêmico-científicos
11    \item Gratuito
12  \end{itemize}
13 \end{itemize}
```

# Exemplo de Aninhamento

- Word
  - Processador de palavras (WYSIWYG)
  - Pago
- L<sup>A</sup>T<sub>E</sub>X
  - Edição e Compilação
  - Foco em documentos acadêmico-científicos
  - Gratuito

# Exemplo de Aninhamento

## ■ Outro exemplo:

```
1 \begin{enumerate}
2   \item Word
3   \begin{itemize}
4     \item Processador de palavras (WYSIWYG)
5     \item Pago
6   \end{itemize}
7   \item \LaTeX
8   \begin{enumerate}
9     \item Edição e Compilação
10    \item Foco em documentos acadêmico-científicos
11    \item Gratuito
12  \end{enumerate}
13 \end{enumerate}
```

# Exemplo de Aninhamento

## 1 Word

- Processador de palavras (WYSIWYG)
- Pago

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

- 1 Edição e Compilação
- 2 Foco em documentos acadêmico-científicos
- 3 Gratuito

# Numeração de Categorias

- Todos os comandos anteriores podem ser adaptados para que eles não sejam numerados
- Essa adaptação é feita ao usar um asterisco \*, como nos exemplos a seguir:
  - `\section*{Nome}`
  - `\subsection*{Nome}`
  - `\paragraph*{Nome}`
- Importante: a depender do template, é possível que algumas partes da taxonomia já não forneçam numeração

# Estilos

- **Negrito:** `\textbf{Negrito}`
- *Itálico:* `\textit{Itálico}`
- Tachado: `\sout{Tachado}`
  - Necessário usar o comando `\usepackage{ulem}` no preâmbulo do arquivo
- É possível combinar essas opções. Exemplos:
  - ***Negrito e Itálico:*** `\textbf{\textit{Negrito e Itálico}}`
  - ~~**Itálico e Tachado:**~~ `\textbf{\sout{Itálico e Tachado}}`

# Tamanhos de Fonte

- É possível customizar textos para que eles usem tamanhos diferentes de fontes

Comando	Saída	Comando	Saída
<code>\tiny{oi}</code>	oi	<code>\large{oi}</code>	oi
<code>\scriptsize{oi}</code>	oi	<code>\Large{oi}</code>	oi
<code>\footnotesize{oi}</code>	oi	<code>\LARGE{oi}</code>	oi
<code>\small{oi}</code>	oi	<code>\huge{oi}</code>	oi
<code>\normalsize{oi}</code>	oi	<code>\Huge{oi}</code>	oi



# Espaçamento

- As vezes é necessário forçar alterações em documentos ao considerar espaçamento
  - Quebra de linha: `\\` ou `\newline` ou `\hfill \break`
  - Quebra de página: `\newpage`
  - Espaçamento horizontal: `\hfill` ou `\hspace`
  - Espaçamento vertical: `\vfill` ou `\vspace`

# Agenda

- 1 Introdução
- 2 Ambiente Overleaf
- 3 Formatando Arquivos com  $\text{\LaTeX}$
- 4 Imagens**
- 5 Tabelas
- 6 Posicionamento de Elementos

# Imagens

- $\text{\LaTeX}$  dá suporte à inserção de imagens em documentos
- Imagens podem ser fornecidas em diferentes formatos (.jpg, .png, .svg, .pdf, etc)
- Uma imagem pode ser colocada livremente no documento ou pode ser colocada em um “ambiente”
- Um ambiente permite **posicionamento, rotulação e referência**
- Para inserir figuras, usaremos o comando `\includegraphics`
- Nos exemplos a seguir, o comando `\textwidth` retorna a largura da página (que pode variar de acordo com o *template*)



# Imagens

```
1 \includegraphics[width=0.25\textwidth]{  
  imgs/presenter.jpg}
```

```
1 \includegraphics[height=1cm]{imgs/  
  presenter.jpg}
```

```
1 \includegraphics[width=0.25\textwidth,  
  angle=45]{imgs/presenter.jpg}
```



# Imagens

- A forma mais comum de se trabalhar com imagens é usar o ambiente

`\begin{figure} ... \end{figure}` com `\caption`

```
1 Abaixo temos uma figura dentro de
  um ambiente \textit{figure}:
2 \begin{figure}
3   \includegraphics[width=0.15\
    textwidth]{imgs/presenter.
      jpg}
4   \caption{Um apresentador}
5 \end{figure}
```

Abaixo temos uma figura dentro de um ambiente *figure*:



Figure 1: Um apresentador

- Note que essa figura agora está **numerada** e possui **rótulo**!
- Via de regra, o **rótulo** (*caption*) de uma figura é posicionada abaixo da imagem

# Sub-figuras

- Em alguns casos, é comum que uma figura possua sub-figuras
- Uma solução interessante é usar o pacote `subfig`: `\usepackage{subfig}`
- Outra opção é usar o pacote `subcaption`, contudo, ele não funciona com Beamer (slides)

## Exemplo: Sub-figuras

```
1 \begin{figure}
2   \centering
3   \subfloat[Cachorro\label{fig:cachorro}]{
4     \includegraphics[width=0.3\textwidth]{imgs/cachorro.jpeg}
5   }
6   \hfill
7   \subfloat[Gato\label{fig:gato}]{
8     \includegraphics[width=0.3\textwidth]{imgs/gato.png}
9   }
10  \hfill
11  \subfloat[Pato\label{fig:pato}]{
12    \includegraphics[width=0.3\textwidth]{imgs/pato.jpg}
13  }
14  \caption{Três animais ilustrados}
15  \label{fig:animais}
16 \end{figure}
```



# Animais em Subfiguras



(a) Cachorro



(b) Gato



(c) Pato

Figure 2: Três animais ilustrados

# Figuras em documentos de múltiplas colunas

- É possível fazer com que uma figura desconsidere a limitação do tamanho de uma coluna usando:
- `\begin{figure*} ... \end{figure*}`

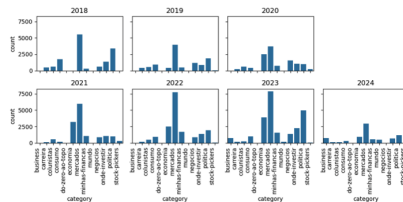


Fig. 2. News distribution according to categories across the analyzed period.

## D. Language Models (LMs)

Since the dataset used is written in Brazilian Portuguese, we used BERTimbau [13] as the base L.M. BERTimbau is a BERT model pre-trained using the brWaC dataset [20], which was the most extensive open Portuguese corpus up to its development, containing 3.53 million web pages and 17.5 GB of raw text [13].

- **SBERT2019<sup>3</sup>**: SBERT2018 model fine-tuned with sampled news from 2019;
- **SBERT2020<sup>4</sup>**: SBERT2019 model fine-tuned with sampled news from 2020;
- **SBERT2021<sup>5</sup>**: SBERT2020 model fine-tuned with sampled news from 2021;
- **SBERT2022<sup>6</sup>**: SBERT2021 model fine-tuned with sampled news from 2022;

# Textos pós-figuras

- Podemos também adicionar texto após um `\caption`

```
1 \begin{figure}[th]
2   \centering
3   \includegraphics[width=\
   textwidth]{cachorro.jpeg}
4   \caption{A foto de um cachorro
   fofinho.}
5   {\footnotesize Fonte: O autor.}
6 \end{figure}
```



Figure 3: A foto de um cachorro fofinho.

Fonte: O autor.

# Agenda

- 1 Introdução
- 2 Ambiente Overleaf
- 3 Formatando Arquivos com  $\text{\LaTeX}$
- 4 Imagens
- 5 Tabelas**
- 6 Posicionamento de Elementos

# Tabelas

- Uma das partes mais “difíceis” de se aprender L<sup>A</sup>T<sub>E</sub>X é a construção de tabelas
- O comando `\begin{tabular} ... \end{tabular}` cria tabelas em si, enquanto o comando `\begin{table}... \end{table}` cria um *float*, similar ao que vimos para *figure*

# Criando Tabelas - *tabular*

## Sintaxe básica

```
1 \begin{tabular}{c|c}
2 Coluna 1 & Coluna 2 \\
3 \hline
4 A & B \\
5 C & D \\
6 \end{tabular}
```

## Resultado

Coluna 1	Coluna 2
A	B
C	D

- A parametrização `{c|c}` nos diz que a tabela possui duas colunas centralizadas e divididas por uma barra no meio.

# Criando Tabelas - *tabular*

## Sintaxe básica

```
1 \begin{tabular}{c|c}  
2 Coluna 1 & Coluna 2 \\  
3 \hline  
4 A & B \\  
5 C & D \\  
6 \end{tabular}
```

## Resultado

Coluna 1	Coluna 2
A	B
C	D

- A linha `Coluna 1 & Coluna 2 \\\` popula a primeira linha da tabela, sendo que:
- `&` é o divisor das colunas
- `\\` é o comando para determinar o final de uma linha

# Criando Tabelas - *tabular*

## Sintaxe básica

```
1 \begin{tabular}{c|c}  
2 Coluna 1 & Coluna 2 \\  
3 \hline  
4 A & B \\  
5 C & D \\  
6 \end{tabular}
```

## Resultado

Coluna 1	Coluna 2
A	B
C	D

- O comando `\hline` aplica uma linha horizontal que percorre todas as colunas
- Uma alternativa seria usar `\cline{i-f}`, onde `i` e `f` são os índices das colunas para qual a linha deve aparecer



# Tipos de colunas

- Além da opção centralizada ( $\textcircled{c}$ ), temos outras opções para definir colunas:
  - $\textcircled{l}$ : alinhado à esquerda
  - $\textcircled{r}$ : alinhado à direita
  - $\textcircled{p}$ : *parbox* - permite definição de um tamanho

## Exemplo: tipos de colunas

```
1 \begin{tabular}{r||c|l|p{7cm}}
2   \hline
3   Nome                & Nota & Faltas & Comentário   \\
4   \hline
5   Angela dos Santos  & 10,0 & 2      & --\\
6   Rômulo Ricardo    & 8,0  & 0      & --\\
7   Thales Alves      & 9,0  & 5      & Serviço militar \\
8   \hline
9 \end{tabular}
```

Nome	Nota	Faltas	Comentário
Angela dos Santos	10,0	2	–
Rômulo Ricardo	8,0	0	–
Thales Alves	9,0	5	Serviço militar

## Colocando tabelas em *floats* - *table*

- O objetivo do uso do ambiente `table` é construir um float, visando adicionar um **identificador (número)** e um **rótulo**
- Via de regra, o **rótulo** (*caption*) de uma tabela é apresentado no topo

```
1 \begin{table}
2   \caption{Nome da tabela}
3   \begin{tabular}{|c|c|}
4     \hline
5     Nome & Nota \\
6     \hline
7     Angela & 10,0 \\
8     Rômulo & 8,0 \\
9     Thales & 9,0 \\
10    \hline
11  \end{tabular}
12 \end{table}
```

# Colocando tabelas em *floats* - *table*

Resultado:

Table 1: Nome da tabela

Nome	Nota
Angela	10,0
Rômulo	8,0
Thales	9,0

# Customizações e Sugestão

- Para tabelas, é interessante conhecer o pacote `multirow` (para mesclar células de uma coluna) e o comando `multicolumn` (para mesclar células de uma linha)
- Para construir tabelas em  $\text{\LaTeX}$ , recomenda-se o uso do site <http://tablesgenerator.com>

# Tabelas em documentos de múltiplas colunas

- Similarmente ao que ocorre com figuras, é possível que uma tabela desconsidere a limitação do tamanho de uma coluna usando:

```
\begin{table*}...\end{table*}
```

TABLE II  
LM EVALUATION OVER TIME, IN TERMS OF MACRO F1 (%), OVER TEN EXECUTIONS. THE BEST VALUES PER NEWS SUBSET ARE IN BOLD.

Model/News	News 2018	News 2019	News 2020	News 2021	News 2022	News 2023	News 2024
SBERTimbau	<b>55.02 ± 0.47</b>	61.39 ± 0.62	60.61 ± 0.64	57.64 ± 0.52	60.01 ± 0.87	55.66 ± 0.54	59.11 ± 0.88
SBERT2018		55.42 ± 0.51	62.86 ± 0.9	59.97 ± 1.22	62.20 ± 0.53	57.76 ± 0.61	58.55 ± 2.68
SBERT2019			62.34 ± 0.51	66.20 ± 0.49	69.30 ± 0.54	59.06 ± 0.89	62.24 ± 2.12
SBERT2020				67.00 ± 0.53	<b>73.47 ± 0.62</b>	<b>60.63 ± 0.89</b>	58.92 ± 1.86
SBERT2021					62.42 ± 0.62	58.77 ± 0.98	56.29 ± 2.77
SBERT2022						53.79 ± 0.66	61.45 ± 2.69
SBERT2023							60.71 ± 0.62
Upd. SBERT	<b>55.02 ± 0.47</b>	<b>61.57 ± 0.63</b>	<b>69.05 ± 0.43</b>	<b>74.52 ± 0.64</b>	71.84 ± 0.75	60.24 ± 1.10	<b>64.80 ± 2.82</b>

SBERT using the regular BERTimbau, while *Updated SBERT* is the SBERT model fine-tuned with sampled news from the previous year. We observe that only in 2022 and 2023, using the Updated SBERT does not favor the macro F1-score. This behavior is correlated with the appearance of new categories (classes), as depicted in Fig. 5.

of macro F1 with SBERTimbau, SBERT2018, and Updated SBERT in 2019.

We evaluated statistically the results in terms of macro F1-score using the Friedman test and the Nemenyi test, following the procedure presented in [33]. The Friedman test assigns the existence of a statistical difference between the methods

# Agenda

- 1 Introdução
- 2 Ambiente Overleaf
- 3 Formatando Arquivos com  $\text{\LaTeX}$
- 4 Imagens
- 5 Tabelas
- 6 Posicionamento de Elementos**

# Posicionamento de *Floats*

- Apesar do  $\text{\LaTeX}$  se preocupar com a formatação de um documento, as vezes é necessário controlar o posicionamento de *floats*
- *Floats* são elementos “flutuantes”, tradicionalmente imagens e tabelas
- O posicionamento de um *float* é controlado usando um argumento opcional, passado dentro de  $[ ]$  e  $( )$



# Posicionamento de *Floats*

Parâmetro	Descrição
<b>h</b>	Posicione “aqui”, i.e., aproximadamente no mesmo local que o item ocorre em relação ao texto
<b>t</b>	Posicione no topo da página
<b>b</b>	Posicione no final da página
<b>p</b>	Coloque em uma página apenas com <i>floats</i>
<b>!</b>	Ignore as configurações do L <sup>A</sup> T <sub>E</sub> X do que é uma posição “boa” para um <i>float</i>
<b>H</b>	Posiciona exatamente “aqui”, similar ao uso de <b>h!</b>

- É comum também passar vários argumentos em sequência
- Exemplo: **[htb!]** significa: tente colocar aqui, tente colocar no topo; e tente colocar no final (em sequência)

# Posicionamento Horizontal

- Dentro de um float, também é comum definirmos o posicionamento horizontal de objetos (tabelas, figuras, etc)
- Centralização:
  - `\centering`
  - `\begin{center} ... \end{center}`
- À direita:
  - Usar `\begin{flushleft} ... \end{flushleft}`
- À esquerda:
  - Usar `\begin{flushright} ... \end{flushright}`

# Posicionamento Horizontal - Centralização

## Código

```
1 \begin{figure}
2   \centering
3   \includegraphics[width=0.5\
      linewidth]{imgs/cachorro.
      jpeg}
4   \caption{Um cachorro
      centralizado usando \textit
      {centering}}
5   \label{fig:cachorro_
      centralizado}
6 \end{figure}
```

## Resultado



Figure 4: Um cachorro centralizado usando *centering*

# Posicionamento Horizontal - Centralização

## Código

```
1 \begin{figure}
2   \begin{center}
3     \includegraphics[width=0.5\
4       linewidth]{imgs/cachorro.
5       jpeg}
6   \end{center}
7   \caption{Um cachorro
8     centralizado usando \textit
9     {center}}
10  \label{fig:cachorro_
11    centralizado}
12 \end{figure}
```

## Resultado



Figure 5: Um cachorro centralizado usando *center*

# Posicionamento Horizontal - Esquerda

## Código

```
1 \begin{figure}
2   \begin{flushleft}
3     \includegraphics[width=0.5\
4       linewidth]{imgs/
5       cachorro.jpeg}
6   \end{flushleft}
7   \caption{Um cachorro do lado
8     esquerdo}
9   \label{fig:cachorro_esquerda}
10 \end{figure}
```

## Resultado



Figure 6: Um cachorro do lado esquerdo



# Posicionamento Horizontal

- Apesar dos exemplos anteriores terem sido feitos com imagens (*figure*), o mesmo processo pode ser feito com tabelas (*tabular* e *table*)

# Espaçamento

- As vezes é necessário forçar alterações em documentos ao considerar espaçamento
  - Quebra de linha: `\\` ou `\newline` ou `\hfill \break`
  - Quebra de página: `\newpage`
  - Espaçamento horizontal: `\hfill` ou `\hspace`
  - Espaçamento vertical: `\vfill` ou `\vspace`