

Universidade Federal do Pampa

Marcos Vinícius Treviso

# **Modelo para Trabalho de Conclusão de Curso**

Alegrete

2015



Marcos Vinícius Treviso

## **Modelo para Trabalho de Conclusão de Curso**

Projeto de Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Fábio Natanael Kepler

Alegrete

2015



Marcos Vinícius Treviso

## Modelo para Trabalho de Conclusão de Curso

Projeto de Trabalho de Conclusão de Cursoa-  
presentado ao Curso de Graduação em Ciên-  
cia da Computação da Universidade Federal  
do Pampa como requisito parcial para a ob-  
tenção do título de Bacharel em Ciência da  
Computação.

Projeto de Trabalho de Conclusão de Curso defendido e aprovado em ..... de ..... de  
.....

Banca examinadora:

---

**Fábio Natanael Kepler**  
Orientador

---

**Professor**  
sigla da instituição

---

**Professor**  
sigla da instituição



*Este trabalho é dedicado às crianças adultas que,  
quando pequenas, sonharam em se tornar cientistas.*





# Agradecimentos

O esforço para chegar até aqui é grande, e certamente não conseguiria ter feito isso sozinho, por isso gostaria de agradecer meu orientador Fábio por me ajudar em assuntos variados durante a faculdade, com o qual aprendi muito.

Não é só de estudo que concluímos uma faculdade, depende muito da nossa força de vontade, e nesse quesito gostaria de agradecer muito a minha irmã Thaís, e especialmente meu pai Gilmar e minha mãe Loeiri por todos os conselhos e apoio.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



*"Existem muitas hipóteses em ciência que estão erradas. Isso é  
perfeitamente aceitável, eles são a abertura para achar as que estão certas."  
(Carl Sagan - O Mundo Assombrado Pelos Demônios)*



# Resumo

Segundo a ??, 3.1-3.2), o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (...) As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

**Palavras-chave:** Aprendizado de máquina. Processamento de linguagem natural. Part-of-speech tagging. Redes neurais. Word embeddings.



# Abstract

This is the english abstract.

**Key-words:** Machine Learning. Natural language processing. Part-of-speech tagging. Neural networks. Word embeddings.





# Lista de ilustrações

Figura 1 – Exemplo de classificação gramatical . . . . .	25
Figura 2 – Diagrama para resultado final . . . . .	29
Figura 3 – Demonstração de regressão . . . . .	30
Figura 4 – Demonstração de classificação . . . . .	31
Figura 5 – Função sigmoide . . . . .	32
Figura 6 – Função de custo - $J(\theta_0, \theta_1)$ . . . . .	34
Figura 7 – Funcionamento do Gradiente Descendente . . . . .	35
Figura 8 – Exemplo de criação de <i>word embeddings</i> . . . . .	37
Figura 9 – t-SNE: Visualização para <i>word embeddings</i> . . . . .	38
Figura 10 – Exemplo de palavras similares . . . . .	38
Figura 11 – Árvore de recursão de Fibonacci. . . . .	46
Figura 12 – Gráfico produzido diretamente no arquivo fonte. . . . .	46
Figura 13 – Outro gráfico feito em L <sup>A</sup> T <sub>E</sub> X. . . . .	47
Figura 14 – Variação dos resultados utilizando seleção por Janela Deslizante. . . . .	48
Figura 15 – Gráfico produzido em Excel e salvo como PDF. . . . .	49
Figura 16 – Exemplo de subfiguras. . . . .	49
Figura 17 – Mesmo exemplo de subfiguras, agora em escala. . . . .	50
Figura 18 – Exemplo de tamanhos de fonte . . . . .	67
Figura 19 – Tela do JabRef para uma versão inicial do arquivo bib deste documento	69



# Lista de tabelas

Tabela 1 – Nomenclaturas . . . . .	31
Tabela 2 – Níveis de investigação . . . . .	45
Tabela 3 – Exemplo de tabela com dados de arquivo. . . . .	45
Tabela 4 – Linhas de código para inserir figura . . . . .	68



# Lista de abreviaturas

**Fig.** Figura



# Lista de siglas

**TCC** Trabalho de Conclusão de Curso





# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>25</b>
<b>1.1</b>	<b>Objetivo</b>	<b>25</b>
<b>1.2</b>	<b>Estrutura do trabalho</b>	<b>26</b>
<b>2</b>	<b>O PROBLEMA</b>	<b>27</b>
<b>3</b>	<b>FUNDAMENTOS</b>	<b>29</b>
<b>3.1</b>	<b>Aprendizado de máquina</b>	<b>29</b>
3.1.1	Classificação	30
3.1.2	Função de hipótese	31
3.1.3	Função de custo	33
3.1.4	Minimização da função de custo	34
3.1.5	Classificação multiclasse	36
3.1.6	Regularização	36
<b>3.2</b>	<b>Cópus e seu conjunto de classes gramaticais</b>	<b>36</b>
<b>3.3</b>	<b>Representação das palavras</b>	<b>37</b>
<b>3.4</b>	<b>Redes neurais profundas</b>	<b>39</b>
<b>4</b>	<b>TRABALHOS RELACIONADOS</b>	<b>41</b>
<b>5</b>	<b>DESENVOLVIMENTO</b>	<b>43</b>
<b>5.1</b>	<b>Formatação</b>	<b>43</b>
<b>5.2</b>	<b>Codificação dos arquivos: UTF8</b>	<b>43</b>
<b>5.3</b>	<b>Citações</b>	<b>43</b>
5.3.1	Referências internas	44
<b>5.4</b>	<b>Tabelas</b>	<b>44</b>
<b>5.5</b>	<b>Figuras</b>	<b>45</b>
5.5.1	Sobre a indicação da fonte de uma tabela ou figura	47
<b>5.6</b>	<b>Expressões matemáticas</b>	<b>49</b>
<b>5.7</b>	<b>Enumerações: alíneas e subalíneas</b>	<b>50</b>
<b>5.8</b>	<b>Espaçamento entre parágrafos e linhas</b>	<b>51</b>
<b>5.9</b>	<b>Inclusão de outros arquivos</b>	<b>52</b>
<b>5.10</b>	<b>Compilar o documento <math>\text{\LaTeX}</math></b>	<b>52</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>55</b>

<b>APÊNDICES</b>	<b>57</b>
------------------	-----------

<b>APÊNDICE A – PRIMEIRO APÊNDICE . . . . .</b>	<b>59</b>
---	-----------

<b>APÊNDICE B – SEGUNDO APÊNDICE . . . . .</b>	<b>61</b>
--	-----------

<b>ANEXOS</b>	<b>63</b>
---------------	-----------

<b>ANEXO A – PRIMEIRO ANEXO . . . . .</b>	<b>65</b>
---	-----------

<b>ANEXO B – LATEX PARA PRINCIPIANTES . . . . .</b>	<b>67</b>
---	-----------

<b>B.1</b>	<b>Outras Dicas . . . . .</b>	<b>68</b>
------------	-------------------------------	-----------

# 1 Introdução

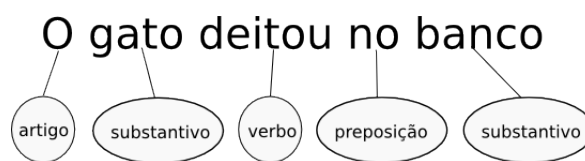
O ato de classificar uma palavra pertencente a um conjunto de textos em uma classe gramatical depende de sua estrutura morfológica, sintática e semântica, isso é conhecido no campo de Processamento de Linguagem Natural (PLN) como Part-of-speech (POS) Tagging. A [Figura 1](#) ilustra esse processo.

Esse conjunto de textos denominados *corpus* são amplamente utilizados para esse processo, e é sobre eles que é feito o treinamento do modelo de reconhecimento de padrões para que seja possível classificar uma palavra à uma certa classe gramatical.

Um dos problemas dessa classificação é justamente a eficiência com o qual cada classe gramatical é atribuída para certa palavra, nesse quesito, há vários métodos já idealizados que conseguiram uma eficiência de cerca de 97%, tais métodos são citados em X, X, onde Y diz ter conseguido o estado-da-arte com 97,42% de precisão.

Apesar de muitos desses métodos já serem utilizados em larga escala, em PLN estamos sempre buscando ganhar mais performance, já que os POS Taggers podem ser aplicados em uma grande variedade de aplicações como tradução automática [X], recuperação e extração de informação [Y], ferramentas de auxílio à leitura e escrita [Z], entre outras.

Figura 1: Exemplo de classificação gramatical



Nosso trabalho consiste em classificar palavras de acordo com seu contexto, ou seja, é feita a análise em termos das unidades primitivas que a compõem, e uma vez que ela é cumprida, podemos passar para outras análises como a sintática ou semântica.

## 1.1 Objetivo

Este trabalho tem, por fim, propor um novo método de classificação de palavras em classes gramaticais e analisar sua eficiência em relação a trabalhos já publicados que utilizam métodos já consolidados. Isso apenas para o escopo da língua portuguesa brasileira.

Para buscar uma boa eficiência será proposto um método novo que se baseia em classificar primeiramente palavras mais fáceis (*e.g verbos*), desse modo, espera-se deixar palavras ambíguas por último. Por exemplo, para a [Figura 1](#) deixaríamos a palavra *banco* por último, uma vez que não sabemos do que isso se trata semanticamente.

Como já mencionado, o estado-da-arte de atualmente tem cerca de 97% de precisão, tentaremos ultrapassar esse limite aplicando técnicas novas e utilizando características das palavras variadas, como por exemplo a escolha de diferentes características à serem levadas em consideração no momento da classificação. É importante salientar que a precisão da classificação não será a única medida levada em consideração, assim como o tempo de processamento gasto para cada cópula.

Para exemplificar brevemente, no final pretende-se que o modelo seja capaz de identificar o uso das palavras de acordo com todas suas análises, como mostrado abaixo.

$$homem + coroa = rei \tag{1.1}$$

## 1.2 Estrutura do trabalho

A fim de proporcionar uma boa interpretação, esse trabalho foi dividido nos seguintes capítulos:

[Capítulo 2](#): Será descrito o problema deste trabalho.

[Capítulo 3](#): Descreve os principais fundamentos necessários para entender o método proposto.

[Capítulo 4](#): Apresentação de trabalhos relacionados que procuram resolver o mesmo problema utilizando métodos semelhantes que o proposto.

[Capítulo 5](#): Apresentará o método proposto e os componentes utilizados.

[??](#): Fará uma comparação entre os resultados preliminares alcançados através do método proposto, assim como as discussões do mesmo.

[Capítulo 6](#): Apresentará as considerações finais.

## 2 O problema

Primeiramente, devemos informar que o problema a ser resolvido não é fácil, pois uma palavra pode possuir diferentes categorias gramaticais, e para solucionar precisamos analisar todo o contexto no qual a palavra está associada. Além disso, como já mostrado anteriormente, há muita ambiguidade no português do Brasil, segundo [X], o português brasileiro é uma das línguas com mais ambiguidade, e isso dificulta a análise morfo-sintática. No exemplo anterior, não fica claro qual é o tipo de banco que está sendo referenciado, os dois são substantivos que se encaixam semanticamente na sentença (*banco: lugar onde é guardado dinheiro.* ou *banco: objeto colocado em praças.*). Entretanto a ambiguidade pode muitas vezes ser resolvida se o contexto for analisado, como por exemplo:

"Eu canto na escola."

A palavra *canto* pode ser classificada como substantivo ou como verbo, mas levando em consideração o sentido da frase, fica evidente que a classificação correta é verbo.

Dito isso, o problema deste trabalho consiste em desenvolver um método para classificar palavras em suas respectivas classes gramaticais de modo eficiente. Para solucionar esse problema, será feito o uso de técnicas de aprendizado de máquina.

Em ordem de conseguir essa eficiência é necessário escolher um bom método computacional, esse trabalho se baseará na utilização de um método conhecido por redes neurais profundas (*deep-learning*), para que seja possível treinar modelos capazes de realizar a classificação. Utilizamos redes neurais pois elas oferecem um jeito alternativo de realizado aprendizado de máquina quando temos hipóteses complexas com muitas características diferentes.



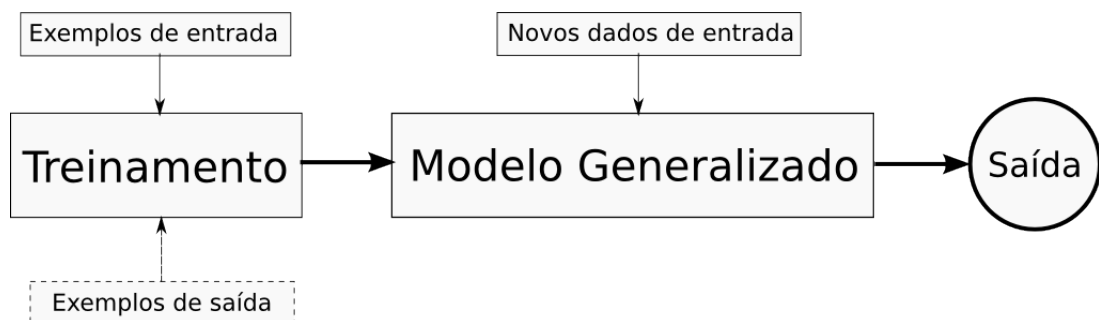
## 3 Fundamentos

Este trabalho é baseado em conceitos de aprendizado de máquina, que dedica-se na elaboração de algoritmos e técnicas que permitem um computador a aprender padrões.

### 3.1 Aprendizado de máquina

Para realizar a aprendizagem é necessário primeiramente dados de entrada que servem como exemplo para nosso modelo, com esses dados é possível treinar o modelo para que ele possa então aprender com base nesses exemplos para que depois seja possível generalizar sobre outros dados não testados ainda. O diagrama da [Figura 2](#) abaixo mostra os passos para obter o resultado final.

Figura 2: Diagrama para resultado final



O aprendizado de máquina pode ser dividido em duas abordagens: O **aprendizado supervisionado**, onde é dado um conjunto de dados de entrada e já se sabe como a saída deve parecer, tendo a ideia de que há uma relação entre a entrada e a saída; O **aprendizado não supervisionado**, que nos permite abordar problemas com pouca ou até nenhuma ideia de como os resultados devem parecer, nesse caso a caixa de exemplos de saída pode não existir.

Neste trabalho a abordagem utilizada vai ser o aprendizado supervisionado, pois já temos uma classe gramatical apropriada para cada palavra em seu contexto.

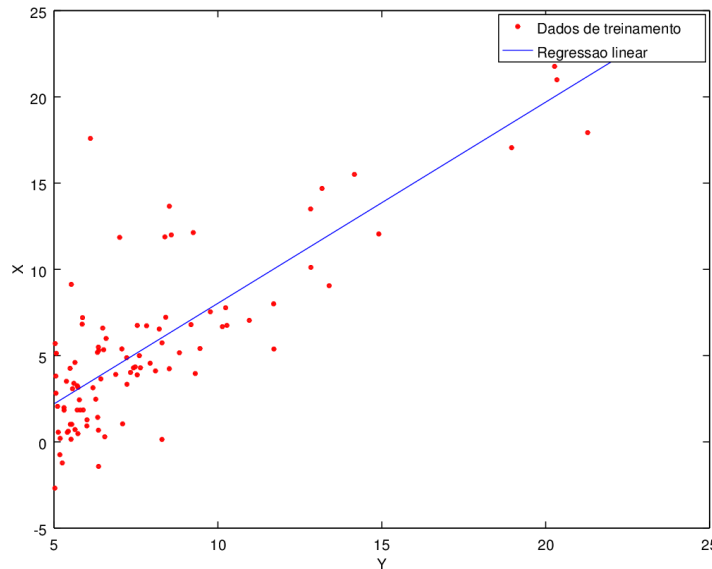
O aprendizado supervisionado permite dividir os problemas em duas categorias distintas:

- **Regressão:** Tenta-se prever resultados com uma saída contínua, significando que estamos tentando mapear variáveis de entrada em alguma função contínua. A [Fi-](#)

gura 3 demonstra esse processo. Nessa demonstração, os círculos em vermelho são dados de exemplo, já a reta azul representa a predição feita pela regressão.

- **Classificação:** Tenta-se prever resultados em uma saída discreta, ou seja, estamos tentando mapear variáveis de entrada em categoriais discretas. A Figura 4 demonstra esse processo. Aqui tem duas classes diferentes, as representadas pelos círculos amarelos e pelas cruzes vermelhas, já o contorno em verde representa o resultado da classificação sobre esses dados de exemplo.

Figura 3: Demonstração de regressão



Logo adiante será explicado os fundamentos de classificação e um método para resolver esse problema chamado de regressão logística. E em virtude de apresentar conceitos fundamentais há serem aplicados posteriormente nas redes neurais profundas, será falado um pouco mais sobre regressão logística e propriedades importantes que serão referenciadas durante todo o trabalho.

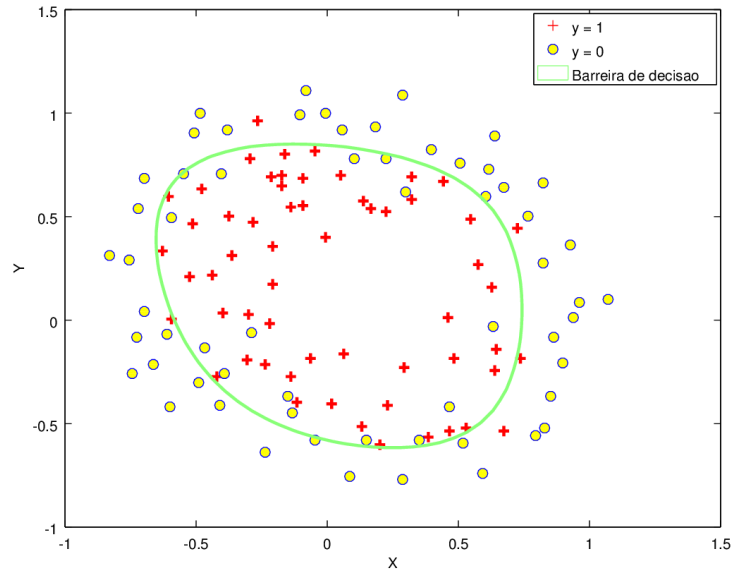
### 3.1.1 Classificação

Em um problema de classificação, nossa saída  $Y$  vai ser um vetor com valores sendo apenas zero ou um.

$$Y \in \{0, 1\} \quad (3.1)$$



Figura 4: Demonstração de classificação



A equação 3.1 está tratando apenas duas classes, esse problema é chamado de classificação binária. Para resolver esse problema, um método que pode ser utilizado é regressão logística [X].

A fim de simplificar o uso das variáveis, faremos o uso de uma notação normalmente utilizado em textos de classificação e mostrados na tabela Tabela 1.

Tabela 1: Nomenclaturas

$m$	número de exemplos de treinamento
$X$	dados de entrada ou <i>features</i>
$Y$	dados de saída
$(x, y)$	um exemplo de treinamento
$(x^{(i)}, y^{(i)})$	o $i$ -ésimo exemplo de treinamento
$\theta$	parâmetros a serem aprendidos

Notação utilizada para classificação

### 3.1.2 Função de hipótese

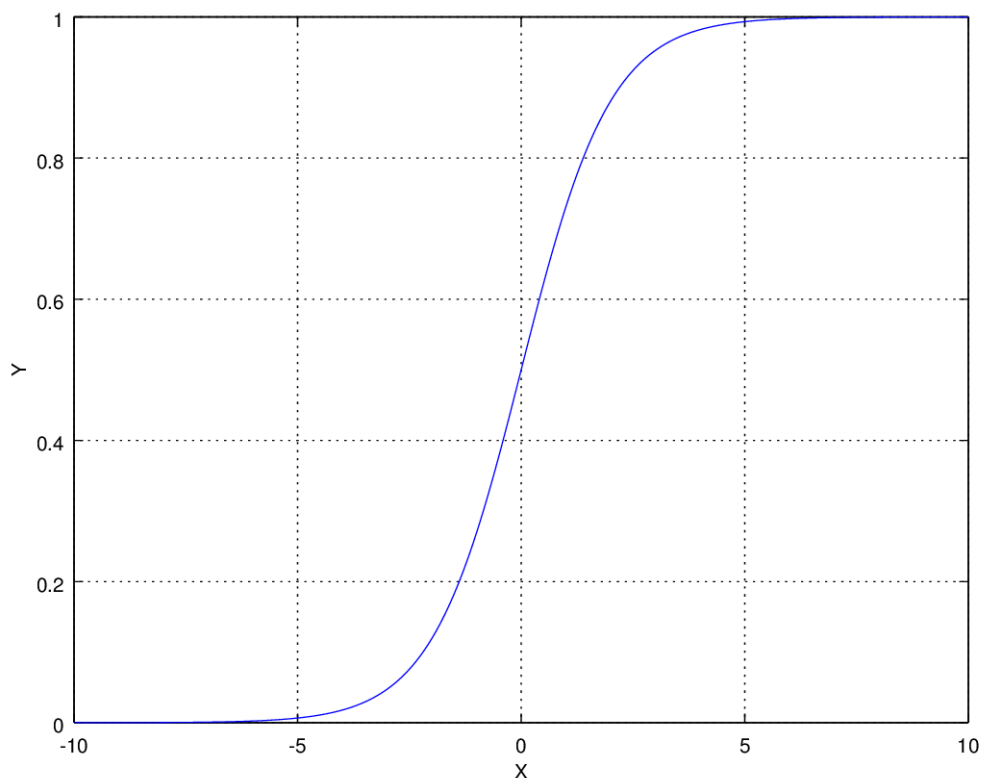
$$0 \leq h_{\theta}(x) \leq 1 \quad (3.2)$$

Uma função de hipótese tem o objetivo de mapear funções aleatórias dentro de nosso intervalo de saída  $Y$ . Portanto, nossa função de hipótese deve satisfazer a equação 3.2.

Para fazer com que uma entrada contínua fique nesse intervalo, é necessária utilizar uma função linear que faça esse mapeamento de modo linear. A nova forma da função de hipótese usa a função sigmoide, também conhecida como função logística.

$$\begin{aligned} h_{\theta}(x) &= g(\theta^T X) \\ z &= \theta^T X \\ g(z) &= \frac{1}{1 + e^{-z}} \end{aligned} \quad (3.3)$$

Figura 5: Função sigmoide



A função da Equação 3.3 representada na figura Figura 5 mapeia qualquer número no intervalo  $[0, 1]$ , fazendo isso útil para transformar qualquer valor arbitrário em uma função mais ideal para problemas de classificação.

Portanto, para calcular a função de hipótese, podemos colocar o  $\theta^T X$  na função logística. Isso nos dará a probabilidade que a saída é 1.

$$h_\theta(x) = P(y = 1|X; \theta) = 1 - P(y = 0|X; \theta)$$

Ou seja, a probabilidade de nossa predição ser 1 é o oposto da probabilidade de ser 0. Sendo assim, a soma das probabilidades para  $y = 0$  e  $y = 1$  deve ser 1.

Em ordem de ter a classificação discreta, podemos traduzir a saída da função de hipótese de acordo com a [Equação 3.4](#):

$$\begin{aligned} h_\theta(x) \geq 0.5 &\Rightarrow y = 1 \\ h_\theta(x) < 0.5 &\Rightarrow y = 0 \end{aligned} \tag{3.4}$$

Portanto, se a entrada é  $\theta^T X$ , isso significa que:

$$h_\theta(x) = g(\theta^T X) \geq 0.5 \quad \text{quando} \quad \theta^T X \geq 0$$

A partir disso, podemos dizer que:

$$\begin{aligned} \theta^T X \geq 0 &\Rightarrow y = 1 \\ \theta^T X < 0 &\Rightarrow y = 0 \end{aligned}$$

O **limite de decisão** (ou barreira de decisão) é a linha que separa a área quando  $y = 0$  e quando  $y = 1$ . Isso é criado pela função de hipótese. Uma observação importante é que a função de hipótese não precisa ser linear, pode ser até mesmo um círculo ou ter qualquer forma para ajudar os dados como mostrado na [Figura 4](#).

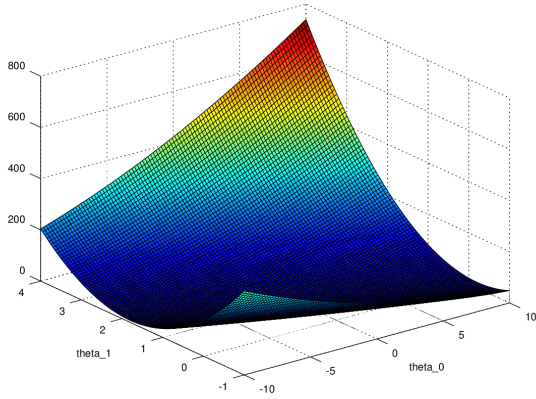
### 3.1.3 Função de custo

Para medir a precisão da função de hipótese é necessário uma função de custo. É feita uma média (elegante) de todos os resultados das hipóteses com a entrada  $X$  comparada com o valor objetivo  $Y$ . Ela pode ser expressa como na [Equação 3.5](#).

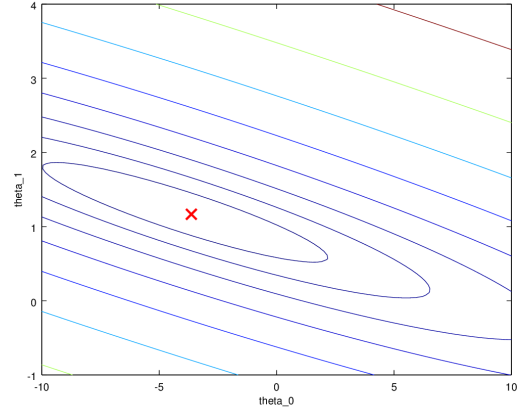
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Custo}(h_\theta(x^{(i)}), y^{(i)}) \tag{3.5}$$

Onde,

$$\text{Custo}(h_\theta(x^{(i)}), y^{(i)}) = \begin{cases} -\log(h_\theta(x)) & \text{se } y = 1 \\ -\log(1 - h_\theta(x)) & \text{se } y = 0 \end{cases} \tag{3.6}$$

Figura 6: Função de custo -  $J(\theta_0, \theta_1)$ 

((a)) Gráfico de superfície



((b)) Gráfico de contorno

O quanto mais a função de hipótese está longe de  $y$ , maior é a saída da função de custo. Se a função de hipótese é igual a  $y$ , então nosso custo é 0.

Podemos simplificar a [Equação 3.6](#) com dois casos condicionais em apenas um caso:

$$\text{Custo}(h_{\theta}(x), y) = -y \log(h_{\theta}(x)) - (1 - y) \log(1 - h_{\theta}(x)) \quad (3.7)$$

Nota-se que quando  $y$  é igual a 1, o segundo termo será 0 e não afetará o resultado. E quando  $y$  é igual a 0, o primeiro termo será 0 e não afetará o resultado. Aplicando a [Equação 3.7](#) na função de custo da equação [Equação 3.5](#), podemos reescrevê-la como a equação [Equação 3.8](#) ou na versão vetorizada da [Equação 3.9](#).

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left( y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right) \quad (3.8)$$

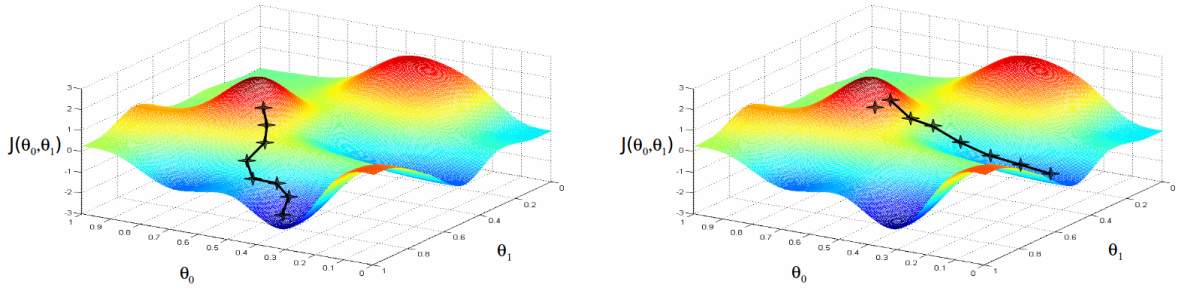
$$J(\theta) = -\frac{1}{m} \left( \log(g(X\theta))^T Y + \log(1 - g(X\theta))^T (1 - Y) \right) \quad (3.9)$$

O objetivo da regressão logística é minimizar a função de custo em relação aos parâmetros  $\theta$ s. Usando dois parâmetros é possível visualizar uma função de custo através de seu gráfico, isso pode ser observado no exemplo da [Figura 6\(a\)](#) e da [Figura 6\(b\)](#).

### 3.1.4 Minimização da função de custo

Para essa tarefa é necessário a aplicação de um algoritmo que pega a função de custo e tente minimizá-la, e por fim retorne os valores dos parâmetros  $\theta$ s aprendidos. Com isso, estaremos melhorando nossa função de hipótese.

Figura 7: Funcionamento do Gradiente Descendente



((a)) Converging para uma mínimo global

((b)) Converging para uma mínimo local

Um dos algoritmos mais utilizados para essa tarefa é o **Gradient Descendente** [X], seu funcionamento baseia-se em seguir a derivada da função de custo em relação aos parâmetros  $\theta$ s alternadamente, com isso a encosta da tangente dará a direção para seguir adiante. Além disso, é aplicada uma taxa de aprendizagem  $\alpha$  a cada iteração, para tentar controlar a convergência.

Seu funcionamento é descrito pelo algoritmo 3.10.

*repita até convergir* {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad (3.10)$$

}

Ao trabalhar sobre a derivada parcial da função de custo, podemos chegar na Equação 3.11 e adiante na versão vetorizada na Equação 3.12. A explicação da derivada pode ser encontrado em [X].

$$\theta_j := \theta_j - \alpha \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right) x_j^{(i)} \quad (3.11)$$

$$\theta := \theta - \frac{\alpha}{m} X^T (g(X\theta) - Y) \quad (3.12)$$

Na Figura 7(a) é possível visualizar esse funcionamento sobre os parâmetros. Observe que nesse caso o algoritmo consegue convergir para o mínimo global, porém em alguns casos (dependendo de nossa taxa de aprendizagem  $\alpha$ ) pode-se convergir para um mínimo local como mostrado na Figura 7(b)

O Gradiente Descendente é apenas um dos vários algoritmos que existem para minimizar a função de custo. De acordo com [X] há alternativas mais sofisticadas como o Gradiente Conjugado, BFGS e L-BFGS, mas também é sugerido que não devemos tentar codificar esses algoritmos, visto que são difíceis e requerem um bom conhecimento de cálculo numérico.

### 3.1.5 Classificação multiclasse

O POS tagging é um problema de classificação multiclasse, onde temos que etiquetar uma palavra em uma de várias categorias possíveis. É possível fazer isso ao expandir nossa definição para que a saída seja  $y = \{0, 1, \dots, n\}$ . Nesse caso, é dividido o problema em  $n + 1$  problemas de classificação binária e em cada um será feita a predizagem da probabilidade de que  $y$  é membro de uma dessas classes.

$$\begin{aligned} h_{\theta}^{(0)}(X) &= P(y = 0 | X; \theta) \\ h_{\theta}^{(1)}(X) &= P(y = 1 | X; \theta) \\ &\vdots \\ h_{\theta}^{(n)}(X) &= P(y = n | X; \theta) \end{aligned}$$

E então escolhe-se a classe com o valor de hipótese mais alto.

$$prediction = \max_i (h_{\theta}^{(i)}(X))$$

### 3.1.6 Regularização

Regularização é um conceito importante designado para resolver o problema de *overfitting*.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## 3.2 Córpus e seu conjunto de classes gramaticais

Córpus são uma coleção de textos colocados como um só, eles são os exemplos de entrada para nosso treinamento. A qualidade e o tamanho desses córpus são importantes para que o etiquetador possa generalizar sentenças ainda não vistas. Eles contêm também um conjunto de classes gramaticais associadas a cada palavra.

Esses conjuntos podem se diferenciar na sua granularidade, como por exemplo, podem ter diferentes classes gramaticais para nomes no plural e no singular, ou agrupar eles em uma única classe [X]. No inglês, há vários córpus de qualidade que são amplamente utilizados, são o Penn Treebank tagset [X], CLAWS5 [X] e CLAWS7 [X].

Já no português esses córpus estão evoluindo com o tempo, embora muitos erros são encontrados neles [X], eles ainda são a melhor opção devido a seu tamanho e qualidade,

os principais e mais utilizados para essa tarefa são o Mac-Morpho [X] com cerca de um milhão de palavras, que retrata artigos publicados na Folha de São Paulo em 1994. O Tycho Brahe [X] que também tem cerca de um milhão de palavras e retrata assuntos históricos de 66 textos diferentes, a linguagem utilizada nele é muito diferente da usada no cotidiano e por isso o seu uso é apenas a fim de curiosidade. Temos também o Bosque [X], que contém cerca de 185 mil palavras. Além desses, [X] et al. apresenta uma versão revisada do Mac-Morpho, com classes gramaticais novas e junção de outras.

Infelizmente esses *córpus* não podem ser combinados em um só, já que eles se diferenciam no conjunto de classes gramaticais e também no seu uso associado.

A aprendizagem baseada em *córpus* tem se mostrado uma estratégia atrativa, já que pode ser usado recursos criados com melhores performance [X, Y, Z]. Devido a isso, nesse trabalho vamos utilizar três *córpus*: o Mac-Morpho original; Mac-Morpho revisado e o Tycho Brahe. O conjunto de etiquetas para esses *córpus* podem ser encontrados, respectivamente, em [X], [Y] e [Z].

- *Falar mais alguma coisa sobre?*
- *Colocar alguma tabela aqui?*

### 3.3 Representação das palavras

Palavras <sup>1</sup> podem ser representadas de várias maneiras em um modelo de aprendizagem, podendo ser até mesmo feito o uso real da palavra como um conjunto de caracteres.

No entanto, a melhor abordagem e a mais indicada atualmente é a utilização de *word embeddings*, elas são representação de palavras como vetores reais valorados em um espaço multidimensional [X]. Elas podem ser geradas de maneiras diferentes dependendo da técnica utilizada, clássicas abordagens baseiam-se na frequência e co-ocorrência [X] das palavras, até mesmo via modelos de redes neurais [X]. Um dos processos de criação é mostrado na [Figura 8](#).

Figura 8: Exemplo de criação de *word embeddings*

| Brasil | possui | vulcão |  $\longrightarrow$  0.51 | 0.35 | ... 0.66 | -0.21 | ... 0.04 | 0.86 | ...

Ou seja, dada uma palavra *word*, temos a transformação  $W$ :

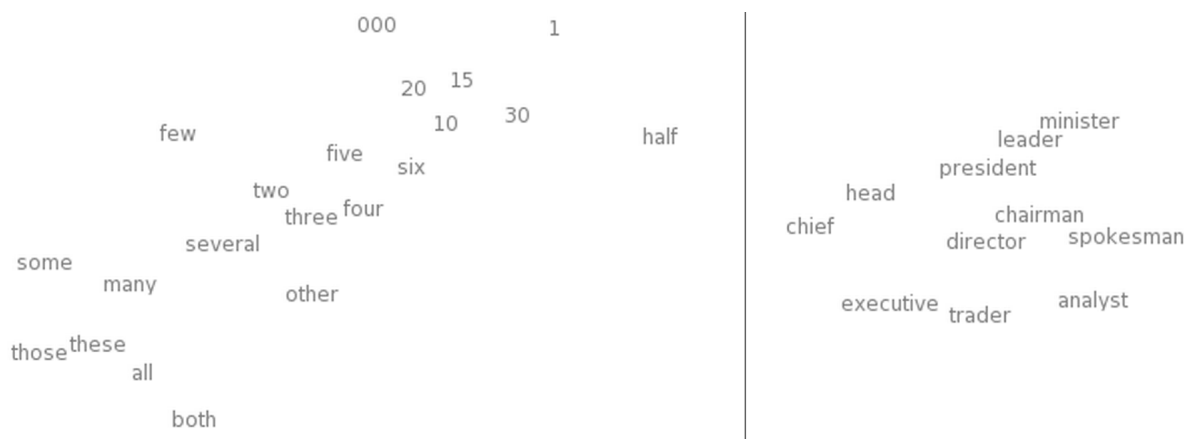
$$W : word \rightarrow \mathbb{R}^n \quad (3.13)$$

Em ordem de prever esses valores com precisão, a rede precisa aprender bons parâmetros para  $W$ .

<sup>1</sup> Uma palavra pode ser qualquer conjunto de caracteres, inclusive pontuações, números, etc.

A grande vantagem em se utilizar *word embeddings* é o fato dessa técnica conseguir capturar informações sintáticas e semânticas das palavras. Isso pode ser visualizado através do t-SNE [X], uma técnica sofisticada para visualizar dados em altas dimensões.

Figura 9: t-SNE: Visualização para *word embeddings*



Fonte: Turian et al. (2010).

Na Figura 9 podemos visualizar um tipo de mapeamento entre os sentidos intuitivos das palavras, onde na esquerda estão palavras referentes a números e na direita palavras referentes a profissões. Ou seja, palavras similares estão pertos umas das outras.

Figura 10: Exemplo de palavras similares

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

Fonte: Collobert et al. (2011)

Na Figura 10 temos mais alguns exemplos desse conceito, com isso podemos realizar tarefas semânticas mais facilmente, como o exemplo 1.1, que é o que esperamos obter.

Para mais informações sobre *word embeddings* consultar [X] e [Y].



## 3.4 Redes neurais profundas



## 4 Trabalhos relacionados

Vários métodos já foram propostos para resolver esse mesmo problema em português, apesar de nenhum deles ter um aproveitamento de 100% de precisão, muitos deles já são utilizados em escala global em diversas ferramentas formuladas por grandes empresas de tecnologia [X,Y].

É apresentado em [X] um etiquetador morfo-sintático baseado em cadeias de Markov de tamanho variável. Ele é testado sobre o corpus Tycho Brahe [X], ele apresenta uma precisão de 95.51%.

Em [Y] é apresentado um etiquetador que aprende automaticamente as features a serem usadas através de uma rede neural profunda que emprega uma camada evolutiva capaz de aprender a uma representação de caracteres das palavras. Isso foi testado sobre diferente corpus, no Mac-Morpho [X] foi alcançado uma precisão de 97.47%, no Tycho Brahe teve precisão de 97.17%, além disso, foi feito testes sobre uma versão revisada do Mac-Morpho distribuída por [Y] e obteve uma precisão de 97.17%

O mais recente etiquetador para o português brasileiro é mostrado em [X], onde é utilizado diferentes técnicas de representação das palavras e feita uma comparação entre elas. Nele é utilizado um modelo de rede neural profunda disponibilizado em [Y] e a melhor técnica obtém uma precisão de 97.47% no Mac-Morpho original.



## 5 Desenvolvimento

Alguns cuidados devem ser tomados no uso deste pacote. Leia as orientações a seguir e contate o responsável em caso de dúvidas.

### 5.1 Formatação

Embora não faça diferença no resultado final, é importante formatar adequadamente o seu código L<sup>A</sup>T<sub>E</sub>X. Da mesma forma que para outras linguagens de programação, isso aumenta a legibilidade do código e ajuda a encontrar partes específicas mais rapidamente. As principais dicas para arquivos T<sub>E</sub>Xsão:

- Indente seu código. Não só os ambientes (begin, end) mas também os parágrafos! Coloque cada sentença em uma linha, indentando a partir da segunda;
- Coloque marcações comentadas para delimitar o início de capítulos, seções, etc. Isso facilita buscar partes específicas em um arquivo.

Cuidado com abreviaturas e acrônimos. É fácil esquecer de os definir ou definir de maneira diferente em capítulos diferentes. Use os comandos do pacote **acro** para abreviaturas e acrônimos. Por exemplo, Figura (**Fig.**) é uma abreviação, então Trabalho de Conclusão de Curso (**TCC**) é um acrônimo/sigla. Eles são definidos no preâmbulo do documento.

Também vale a pena usar uma tabela de nomenclatura caso você use muitos símbolos, em especial símbolos matemáticos. Veja os comandos do pacote **nomenc1**. As definições também ficam no preâmbulo do documento.

### 5.2 Codificação dos arquivos: UTF8

A codificação de todos os arquivos deste pacote é **UTF8**. É necessário que você utilize a mesma codificação nos documentos que escrever, inclusive nos arquivos de bases bibliográficas `|.bib|`.

### 5.3 Citações

Utilize o ambiente **citacao** para incluir citações diretas com mais de três linhas:

As citações diretas, no texto, com mais de três linhas, devem ser destacadas com recuo de 4 cm da margem esquerda, com letra menor que a do texto utilizado e sem aspas. No caso de documentos datilografados, deve-se observar apenas o recuo (??, 5.3)

Citações simples, com até três linhas, devem ser incluídas com aspas. Observe que em  $\text{\LaTeX}$  as aspas iniciais são diferentes das finais: “Amor é fogo que arde sem se ver”.

Para as citações indiretas, o comando padrão, `\cite`, realiza a forma mais comum de citação (??). A outra das formas mais usadas, para citar em texto corrido, é conseguida com o comando `\citeonline`: segundo ??), na citação indireta, o número da página é opcional.

### 5.3.1 Referências internas

Usa-se o comando `\ref{}` para referenciar uma Tabela ou Figura. Por exemplo, esta é uma referência para a Tabela 2. Mas também pode-se usar o comando `\autoref{}`, que insere o tipo também. Por exemplo, esta é outra referência para a Tabela 2.

Há vários outros comandos interessantes. Eles estão no fonte do Capítulo 1, na subseção 5.3.1 <sup>1</sup> (Introdução, página 25).

## 5.4 Tabelas

A Tabela 2 é um exemplo de tabela construída em  $\text{\LaTeX}$ . Como sugestão de formatação, evite ao máximo o uso de linhas verticais. As colunas de uma tabela devem ser separadas visivelmente. O contrário indica que a tabela está mal formatada ou que certas informações não deveriam estar nela.

Da mesma forma, evite o uso de linhas horizontais para separar linhas da tabela. Use-as apenas para separar o cabeçalho e eventuais partes importantes. Para obter um resultado ainda mais elegante, use os comandos do pacote `booktabs`.

Veja essas sugestões aplicadas na Tabela 2.

Uma opção avançada para a criação de tabelas é usar o pacote `pgfplotstable`. Ele permite que os dados de um arquivo sejam lidos e colocados em uma tabela, formatando-os da maneira que se quiser. A Tabela 3 é um exemplo. Veja o arquivo `desenvolvimento.tex` para os comandos necessários.

---

<sup>1</sup> O número do capítulo indicado é 1, que se inicia à página 25.

Tabela 2: Níveis de investigação.

Nível de Inves- tigação	Insumos	Sistemas de Investigação	Produtos
Meta-nível	Filosofia da Ciência	Epistemologia	Paradigma
Nível do objeto	Paradigmas do metanível e evidências do nível inferior	Ciência	Teorias e modelos
Nível inferior	Modelos e métodos do nível do objeto e problemas do nível inferior	Prática	Solução de problemas

Fonte: ??)

Tabela 3: Exemplo de tabela com dados de arquivo.

TAMANHO	MÉTODO 1 (%)	MÉTODO 2 (%)	MÉTODO 3 (%)	MÉDIA (%)
10	30.0	36.2	28.3	31.5
20	54.8	52.5	56.8	54.7
30	65.0	59.6	74.1	66.2
40	64.5	59.6	76.7	66.9
50	64.6	59.6	76.5	66.9

## 5.5 Figuras

Figuras podem ser criadas diretamente em  $\text{\LaTeX}$ . Uma das melhores formas, por ser relativamente simples, bem documentada e gerar ótimos resultados, é com o uso do pacote `tikz`<sup>2</sup>. Ele permite gerar diagramas, árvores, fluxogramas etc. A Figura 11 mostra um exemplo simples de árvore.

Junto com o pacote `pgfplots` também é possível gerar gráficos de funções ou a partir de dados em um arquivo (como no caso da Tabela 3). As Figuras 12 e 13 mostram exemplos de gráficos de função, e a Figura 14 um exemplo de gráfico a partir dos mesmos dados que os da Tabela 3.

Figuras também podem ser incorporadas de arquivos externos, como é o caso da Figura 15. Se a figura que ser incluída se tratar de um diagrama, um gráfico ou uma ilustração que você mesmo produza, priorize o uso de imagens vetoriais no formato PDF. Com isso, o tamanho do arquivo final do trabalho será menor, e as imagens terão uma apresentação melhor, principalmente quando impressas, uma vez que imagens vetoriais são perfeitamente escaláveis para qualquer dimensão. Nesse caso, se for utilizar o Microsoft Excel para produzir gráficos, ou o Microsoft Word para produzir ilustrações, exporte-os como PDF e os incorpore ao documento conforme o exemplo abaixo. No entanto, para manter a coerência no uso de software livre (já que você está usando  $\text{\LaTeX}$  e  $\text{abnTeX2}$ ),

<sup>2</sup> Há vários exemplos em <http://www.texample.net/>.

Figura 11: Árvore de recursão de Fibonacci.

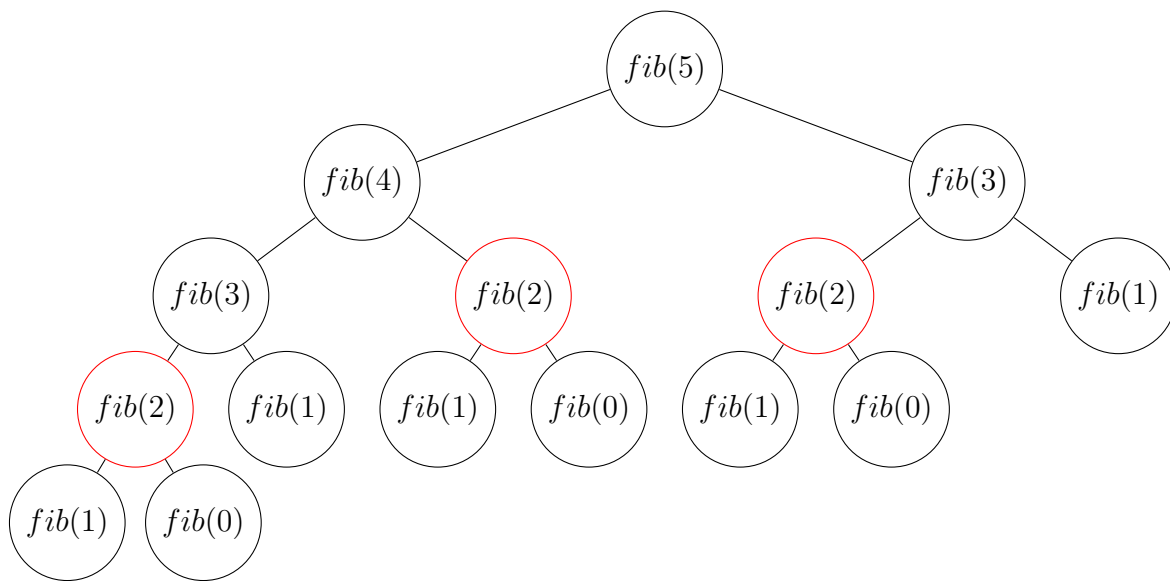
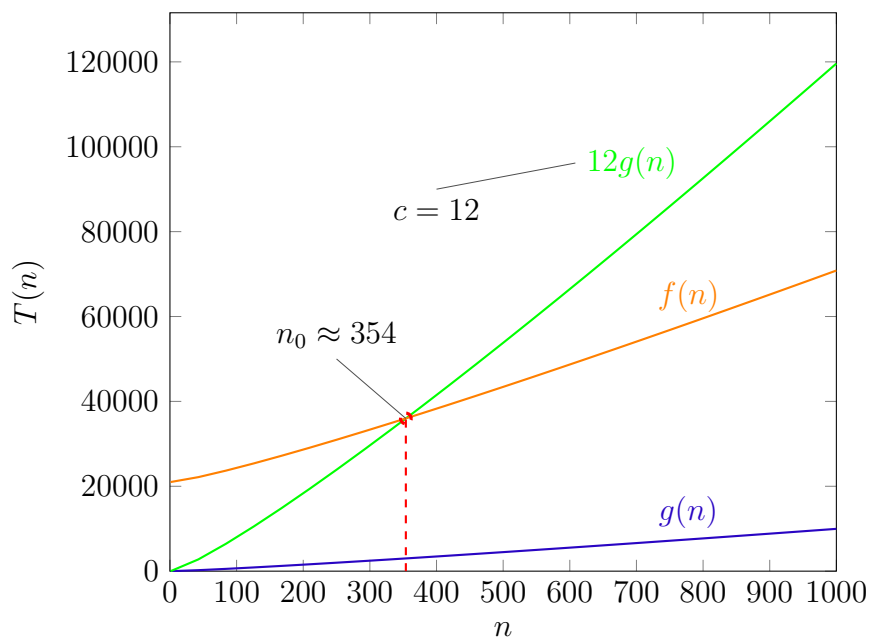


Figura 12: Gráfico produzido diretamente no arquivo fonte.

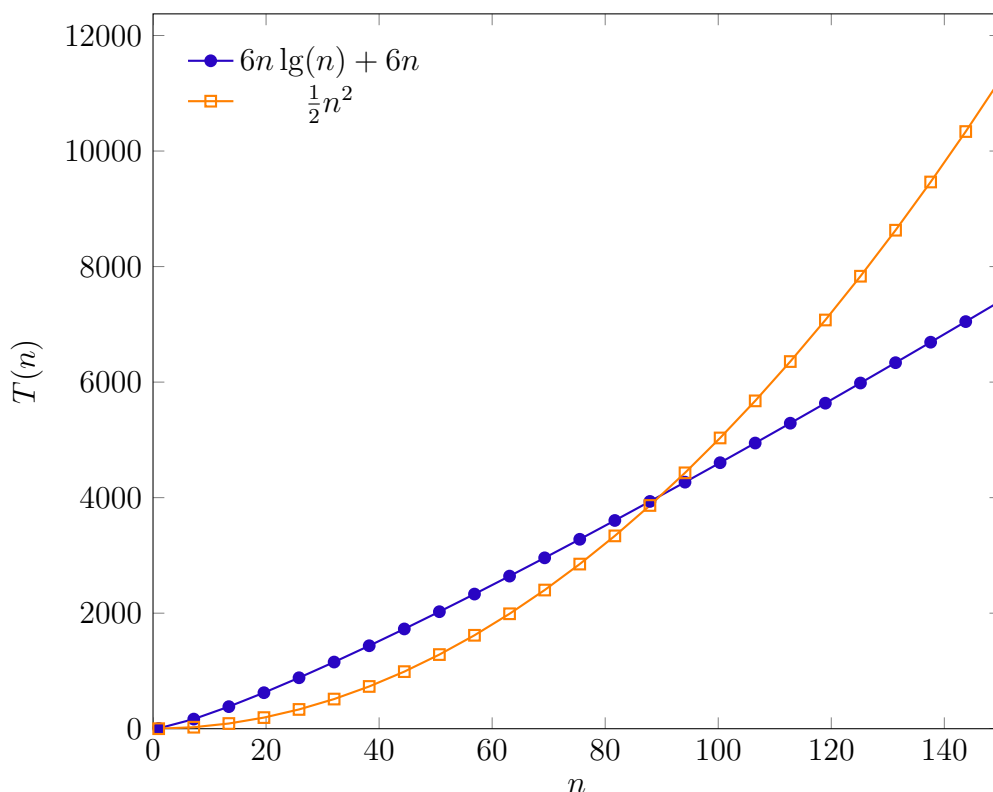


teste a ferramenta **InkScape**<sup>3</sup>. Ela é uma excelente opção de código-livre para produzir ilustrações vetoriais, similar ao CorelDraw ou ao Adobe Illustrator.

De todo modo, caso não seja possível utilizar arquivos de imagens como PDF, utilize qualquer outro formato, como PNG, JPEG, etc. Nesse caso, você pode tentar

<sup>3</sup> <<http://inkscape.org/>>



Figura 13: Outro gráfico feito em L<sup>A</sup>T<sub>E</sub>X.

aprimorar as imagens incorporadas com o software livre **Gimp**<sup>4</sup>. Ele é uma alternativa livre ao Adobe Photoshop.

A Figura 16 na página 49 contém duas subfiguras, Figura 16(a) e (b). A Figura 16(a) foi inserida de um arquivo externo, enquanto a Figura 16(b) foi escrita dentro do próprio código T<sub>E</sub>X. A Figura 17 contém o mesmo exemplo, mas usando comandos diferentes para inserir as Subfiguras 17(a) e (b).

Na Figura 16, as legendas (que indicam a fonte) para cada subfigura só funcionaram porque as figuras ficaram uma embaixo da outra. Se elas estivessem lado a lado, a inserção do comando `\legend` em cada uma faria com que elas ficassem organizadas na vertical. Uma legenda geral funcionaria, entretanto.

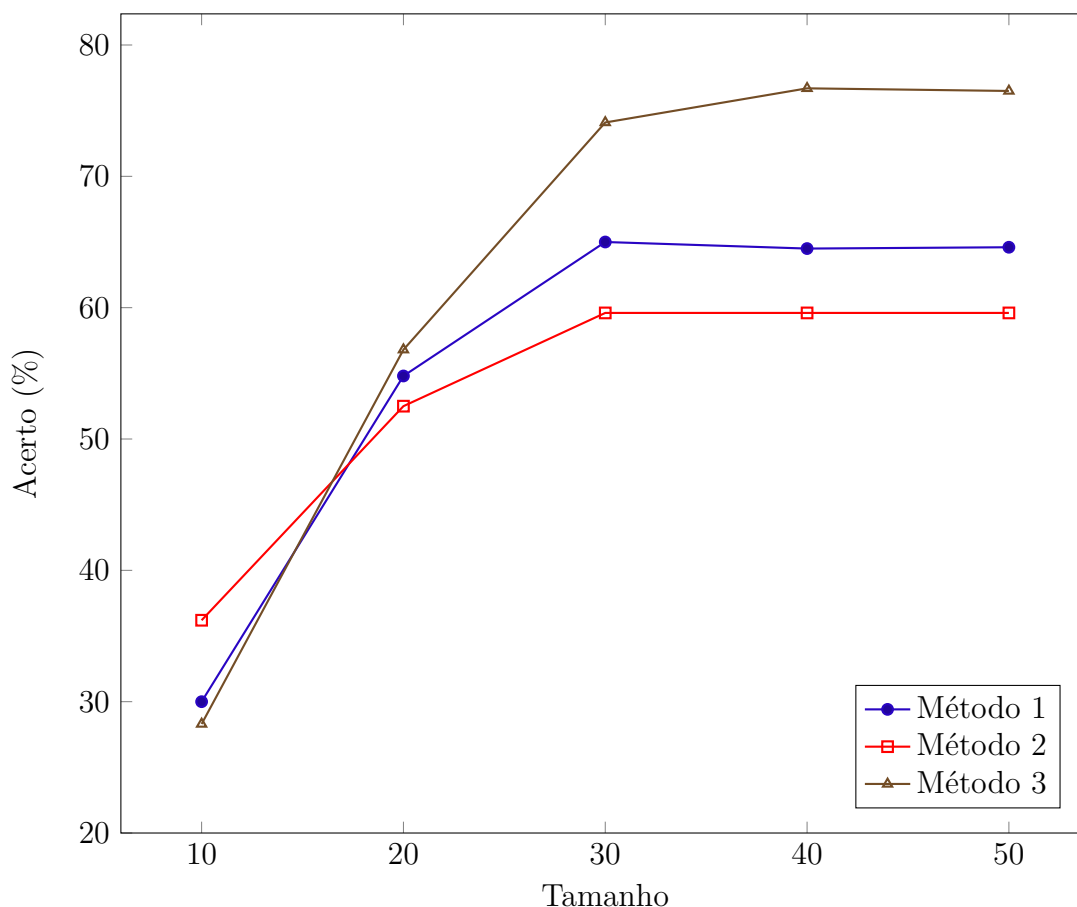
Na Figura 17, tanto legendas para subfiguras quanto uma legenda geral funcionam.

### 5.5.1 Sobre a indicação da fonte de uma tabela ou figura

As normas ??, 5.8) e o Manual de Normatização da UNIPAMPA (??) dizem para, “Após a ilustração, na parte inferior, indicar a fonte consultada (elemento obrigatório, mesmo que seja produção do próprio autor), legenda, notas e outras informações neces-

<sup>4</sup> <<http://www.gimp.org/>>

Figura 14: Variação dos resultados utilizando seleção por Janela Deslizante.

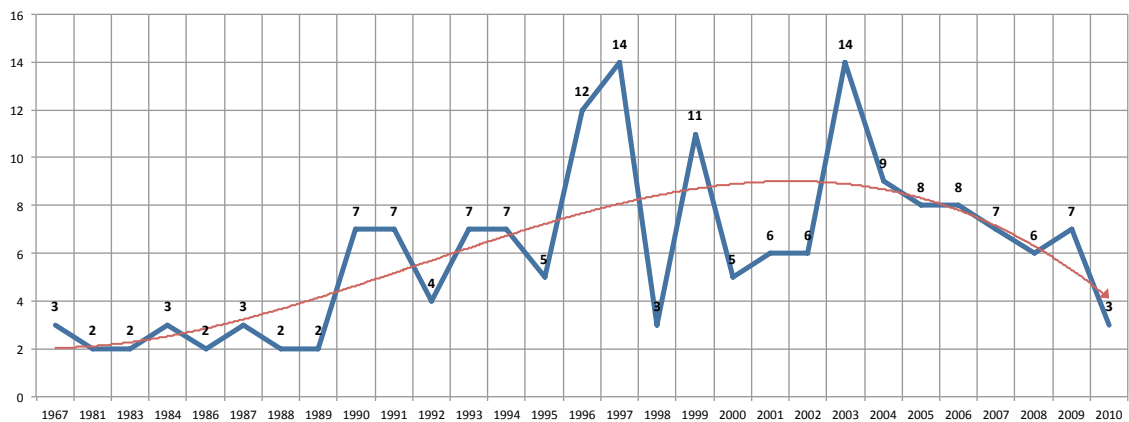


sárias à sua compreensão (se houver).” A primeira interpretação é a de que, mesmo que o autor tenha criado a figura, a fonte deverá ser indicada. Com efeito, várias outras normas, manuais e inclusive o exemplo do pacote `abnTeX22` usam “Fonte: os autores” em alguns lugares.

Entretanto, isso não está correto. Veja o trecho em destaque: “Após a ilustração, na parte inferior, indicar a fonte **consultada** (elemento obrigatório, mesmo que seja produção do próprio autor) (...).” A interpretação correta é a de que, caso a ilustração tenha sido **extraída** de um documento, a fonte deve ser indicada, ainda que esse documento pertença ao próprio autor. A sentença original das normas deveria ter sido melhor escrita para evitar a interpretação incorreta.

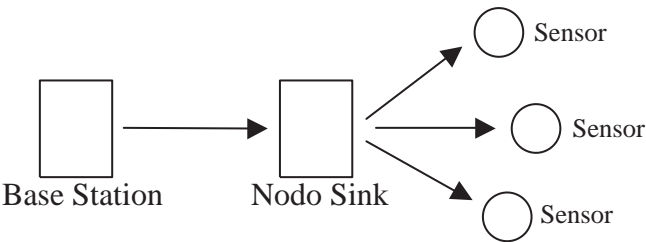
Assim, não indique a fonte se a figura ou tabela foi criada para o trabalho, ou seja, se é inédita. Caso contrário, indique a fonte. Mas cuidado: caso a figura ou tabela tenha sido adaptada de outra já publicada, então é obrigatório indicar “adaptado de” ou “acrescida de” seguido da referência da fonte de onde ela foi extraída.

Figura 15: Gráfico produzido em Excel e salvo como PDF.



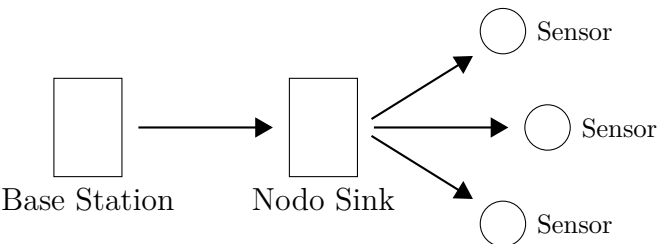
Fonte: ??, p. 24)

Figura 16: Exemplo de subfiguras.



(a) Uma figura de um arquivo.

Fonte: ??)



(b) Uma figura em puro código TikZ.

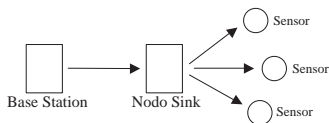
Alterado: de ??)

## 5.6 Expressões matemáticas

Use o ambiente `equation` para escrever expressões matemáticas numeradas:

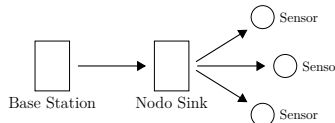
$$\forall x \in X, \quad \exists y \leq \epsilon \tag{5.1}$$

Figura 17: Mesmo exemplo de subfiguras, agora em escala.



((a)) Uma figura de um arquivo.

Fonte: ??)



((b)) Uma figura em puro código TikZ.

Alterado: de ??)

Fonte geral

Escreva expressões matemáticas entre \$ e \$, como em  $\lim_{x \rightarrow \infty} \exp(-x) = 0$ , para que fiquem na mesma linha.

Também é possível usar colchetes para indicar o início de uma expressão matemática que não é numerada.

$$\left| \sum_{i=1}^n a_i b_i \right| \leq \left( \sum_{i=1}^n a_i^2 \right)^{1/2} \left( \sum_{i=1}^n b_i^2 \right)^{1/2}$$

Consulte mais informações sobre expressões matemáticas em <http://code.google.com/p/abntex2/w/edit/Referencias>.

## 5.7 Enumerações: alíneas e subalíneas

Quando for necessário enumerar os diversos assuntos de uma seção que não possua título, esta deve ser subdividida em alíneas (??, 4.2):

- a) os diversos assuntos que não possuam título próprio, dentro de uma mesma seção, devem ser subdivididos em alíneas<sup>5</sup>;
- b) o texto que antecede as alíneas termina em dois pontos;
- c) as alíneas devem ser indicadas alfabeticamente, em letra minúscula, seguida de parêntese. Utilizam-se letras dobradas, quando esgotadas as letras do alfabeto;
- d) as letras indicativas das alíneas devem apresentar recuo em relação à margem esquerda;
- e) o texto da alínea deve começar por letra minúscula e terminar em ponto-e-vírgula, exceto a última alínea que termina em ponto final;

<sup>5</sup> As notas devem ser digitadas ou datilografadas dentro das margens, ficando separadas do texto por um espaço simples de entre as linhas e por filete de 5 cm, a partir da margem esquerda. Devem ser alinhadas, a partir da segunda linha da mesma nota, abaixo da primeira letra da primeira palavra, de forma a destacar o expoente, sem espaço entre elas e com fonte menor. ??, 5.2.1)

- f) o texto da alínea deve terminar em dois pontos, se houver subalínea;
- g) a segunda e as seguintes linhas do texto da alínea começa sob a primeira letra do texto da própria alínea;
- h) subalíneas (??, 4.3) devem ser conforme as alíneas a seguir:
  - as subalíneas devem começar por travessão seguido de espaço;
  - as subalíneas devem apresentar recuo em relação à alínea;
  - o texto da subalínea deve começar por letra minúscula e terminar em ponto-e-vírgula. A última subalínea deve terminar em ponto final, se não houver alínea subsequente;
  - a segunda e as seguintes linhas do texto da subalínea começam sob a primeira letra do texto da própria subalínea.
- i) no abnT<sub>E</sub>X2 estão disponíveis os ambientes **incisos** e **subalineas**, que em suma são o mesmo que se criar outro nível de **alineas**, como nos exemplos à seguir:
  - *Um novo inciso em itálico;*
- j) Alínea em **negrito**:
  - *Uma subalínea em itálico;*
  - *Uma subalínea em itálico e sublinhado;*
- k) Última alínea com *ênfase*.

## 5.8 Espaçamento entre parágrafos e linhas

O tamanho do parágrafo, espaço entre a margem e o início da frase do parágrafo, é definido por:

```
\setlength{\parindent}{1.3cm}
```

Por padrão, não há espaçamento no primeiro parágrafo de cada início de divisão do documento (??). Porém, você pode definir que o primeiro parágrafo também seja indentado, como é o caso deste documento. Para isso, apenas inclua o pacote **indentfirst** no preâmbulo do documento:

```
\usepackage{indentfirst}      % Indenta o primeiro parágrafo de cada seção.
```

O espaçamento entre um parágrafo e outro pode ser controlado por meio do comando:

```
\setlength{\parskip}{0.2cm} % tente também \onelineskip
```

O controle do espaçamento entre linhas é definido por:

```
\OnehalfSpacing      % espaçamento um e meio (padrão);
\DoubleSpacing       % espaçamento duplo
\SingleSpacing       % espaçamento simples
```

Para isso, também estão disponíveis os ambientes:

```
\begin{SingleSpace} ... \end{SingleSpace}
\begin{Spacing}{hfactori} ... \end{Spacing}
\begin{OnehalfSpace} ... \end{OnehalfSpace}
\begin{OnehalfSpace*} ... \end{OnehalfSpace*}
\begin{DoubleSpace} ... \end{DoubleSpace}
\begin{DoubleSpace*} ... \end{DoubleSpace*}
```

Para mais informações, consulte ??, p. 47-52 e 135).

## 5.9 Inclusão de outros arquivos

É uma boa prática dividir o seu documento em diversos arquivos, e não apenas escrever tudo em um único. Esse recurso foi utilizado neste documento. Para incluir diferentes arquivos em um arquivo principal, de modo que cada arquivo incluído fique em uma página diferente, utilize o comando:

```
\include{documento-a-ser-incluido} % sem a extensão .tex
```

Para incluir documentos sem quebra de páginas, utilize:

```
\input{documento-a-ser-incluido} % sem a extensão .tex
```

## 5.10 Compilar o documento L<sup>A</sup>T<sub>E</sub>X

Geralmente os editores L<sup>A</sup>T<sub>E</sub>X, como o TeXlipse<sup>6</sup>, o Texmaker<sup>7</sup>, entre outros, compilam os documentos automaticamente, de modo que você não precisa se preocupar com isso.

<sup>6</sup> <<http://texlipse.sourceforge.net/>>

<sup>7</sup> <<http://www.xmlmath.net/texmaker/>>

No entanto, você pode compilar os documentos  $\text{\LaTeX}$  usando os seguintes comandos, que devem ser digitados no *Prompt de Comandos* do Windows ou no *Terminal* do Mac ou do Linux:

```
pdflatex ARQUIVO_PRINCIPAL.tex
bibtex ARQUIVO_PRINCIPAL.aux
makeindex ARQUIVO_PRINCIPAL.idx
makeindex ARQUIVO_PRINCIPAL.nlo -s nomencl.ist -o ARQUIVO_PRINCIPAL.nls
pdflatex ARQUIVO_PRINCIPAL.tex
pdflatex ARQUIVO_PRINCIPAL.tex
```





## 6 Conclusão

Bom trabalho!



## Apêndices



# APÊNDICE A – Primeiro Apêndice

De acordo com a ABNT:

Apêndice (opcional): texto utilizado quando o autor pretende complementar sua argumentação. São identificados por letras maiúsculas e travessão, seguido do título. Ex.: APÊNDICE A - Avaliação de células totais aos quatro dias de evolução

Anexo (opcional): texto ou documento **não elaborado pelo autor** para comprovar ou ilustrar. São identificados por letras maiúsculas e travessão, seguido do título. Ex.: ANEXO A - Representação gráfica de contagem de células

Tais definições (e outras) podem ser encontradas na NBR 14724-2001 Informação e documentação - trabalhos acadêmicos<sup>1</sup>.

---

<sup>1</sup> <http://www.firb.br/abntmonograf.htm>



## APÊNDICE B – Segundo Apêndice

Pode ser que tenha outro...





# Anexos



## ANEXO A – Primeiro Anexo

Sendo anexo, a formatação dessa seção é livre. Ou seja: aceita-se fonte diferente e menor



## ANEXO B – LaTeX para Principiantes

TEste<sup>1</sup> Dentro dos arquivos .tex o texto pode estar organizado em partes, capítulos, seções, etc. conforme os seguintes comandos:

- `\part{Nome da Parte}`, partes do documento
- `\chapter{Nome}`, capítulos somente para arquivos do tipo *book* e *report*
- `\section{Nome}`, seções
- `\subsection{Nome}`, subseções
- `\subsubsection{Nome}`, seções dentro de subseções
- `\paragraph{Texto}`, parágrafos formatados
- `\subparagraph{Texto}`, subparágrafos

**Parágrafos** Parágrafos são definidos deixando uma linha em branco entre os mesmos. Pode-se também forçar usando `\\` bem como deixar uma linha em branco com um `~` sozinho na linha.

**Formato de texto** O tamanho do texto pode ser definido pelos comandos específicos: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `huge` e `Huge`, conforme ilustra a Figura 18.

Command	Sample	Command	Sample
<code>\tiny</code>	<code>tiny</code>	<code>\scriptsize</code>	<code>scriptsize</code>
<code>\footnotesize</code>	<code>footnotesize</code>	<code>\small</code>	<code>small</code>
<code>\normalsize</code>	<code>normalsize</code>	<code>\large</code>	<code>large</code>
<code>\Large</code>	<code>larger</code>	<code>\LARGE</code>	<code>even larger</code>
<code>\huge</code>	<code>huge</code>	<code>\Huge</code>	<code>largest</code>

Figura 18: Exemplo de tamanhos de fonte

**Referências dentro do Texto** Partes do texto podem ser referenciadas através do par de comandos `\label` e `\ref`. Por exemplo, podemos inserir uma seção no artigo utilizando o seguinte comando:

```
\section{Seção principal}\label{sec:prcpal}
```

Vejam que o título da seção é seguido do comando `\label{nome}`. Esta seção pode ser referenciada em qualquer parte do texto, como o exemplo a seguir.

Conforme explicado na Seção `\ref{sec:prcpal}`, nosso método utiliza...

<sup>1</sup> (??)

Perceba que a figura tem resolução ruim; deveria ser uma tabela

## B.1 Outras Dicas

**Caracteres Especiais** Esses não podem ser usados no texto sem a barra à frente: # \$ % ^ & \_ { } ~ e /.

**Comentários** Comentários são precedidos de % e podem estar em qualquer parte do texto. Lembrando que tudo que estiver após % será considerado como comentário e ignorado pelo processador.

**Incluir Figuras** Incluir figuras no LaTeX é relativamente fácil quando se tem um formato de arquivo pré-definido. Por exemplo, neste documento, usa-se apenas figuras do tipo *pdf*, mas também poderia-se usar do tipo *png* (e *jpeg*, mas este tipo não é recomendado). A Tabela 4 ilustra as linhas que inserem uma figura no texto.

Tabela 4: Linhas de código para inserir figura

Linha de Código	Explicação
<code>\usepackage{graphicx}</code>	<i>inclui pacote gráfico no início do documento</i>
<code>\begin{figure}[tb]</code>	<i>inicia figura, define sua posição no texto</i>
<code>\centering</code>	<i>centraliza a figura na página</i>
<code>\includegraphics[scale=.7]</code>	<i>define escala da figura</i>
<code>{img/figura}</code>	<i>inclui o arquivo da figura no texto</i>
<code>\caption{Legenda}</code>	<i>inclui a legenda da figura</i>
<code>\label{fig:ap}</code>	<i>inclui o apelido da figura</i>
<code>\end{figure}</code>	<i>termina figura</i>

**Hifenização** Às vezes aparece uma palavra cuja hifenização, divisão silábica, está errada. Para resolver esse tipo de problema, pode-se recorrer à divisão manual da palavra, acrescentando \- entre cada sílaba: `Mi\re\lla`. Se, ao invés desta solução, você quiser evitar completamente que suas palavras sejam divididas, acrescente os dois comandos no início do seu documento (ou seja, antes do `begin{document}`).

```
\hyphenpenalty=5000
\tolerance=1000
```

**BibTeX** Para editar facilmente o BibTeX, pode-se utilizar uma ferramenta própria<sup>2</sup>. A minha favorita é o JabRef<sup>3</sup>, ilustrado na Figure 19, porque:

- É de graça;
- Possui interface gráfica super intuitiva;
- Permite importar referências de bases clássicas, como ISI, Medline e RIS;

<sup>2</sup> Ferramentas para BibTeX: <http://dmoz.org/Computers/Software/Typesetting/TeX/BibTeX>

<sup>3</sup> JabRef Editor: <http://jabref.sourceforge.net/>

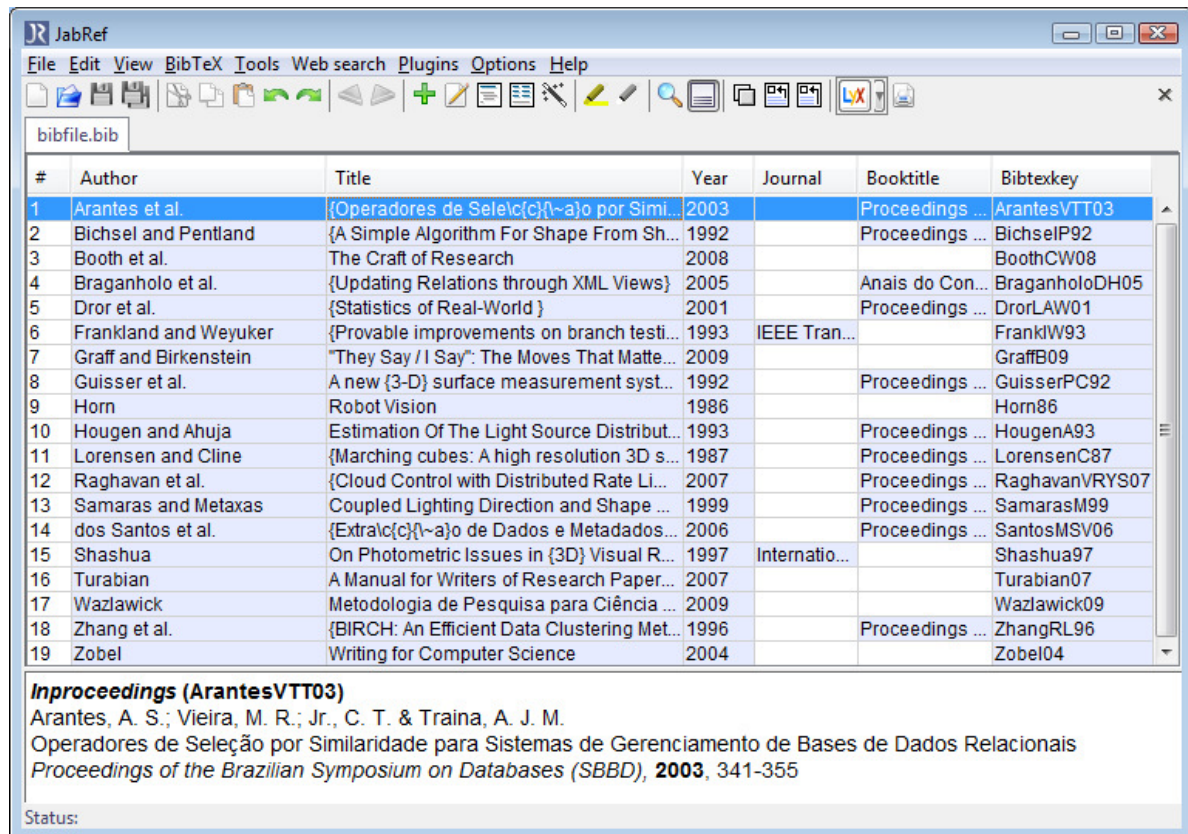


Figura 19: Tela do JabRef para uma versão inicial do arquivo bib deste documento

- Permite exportar para diferentes formatos, inclusive para um banco de dados utilizando SQL;
- Tem botão para procurar o artigo da respectiva referência e fazer o seu download;
- Permite adicionar comentários próprios para cada entrada;
- Pode-ser classificar as referências e criar grupos para as mesmas, e muito muito mais.

**Listas** Listas podem ser definidas com *bullets* ou com números, conforme os exemplos a seguir.

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

```
\begin{enumerate}
\item Item 1 numerado
\item Item 2 numerado
\end{enumerate}
```

**Fontes Coloridas** Para adicionar texto em cores (muito útil para marcar trechos do texto que estão *em trabalho*, deve-se adicionar os pacotes *graphicx* e *color* (usando o comando `\usepackage` e depois utilizar o comando `\textcolor{cor}{texto}` para colorir o *texto* com a *cor* especificada. Por exemplo `\textcolor{blue}{texto em azul}`. Outras cores comuns são *red* e *green*.

**Para Economizar Espaço** Existem alguns *dirty tricks*<sup>4</sup> pra economizar espaço, como por exemplo:

- `\usepackage{times}` Usa fonte *Times* no lugar da default.
- `\usepackage[small,compact]{titlesec}` Modifica o título e os espaços antes/depois dos mesmos.
- `\usepackage[small,it]{caption}` Reduz o tamanho das legendas de tabelas e figuras.

**WEB** A Web é repleta de páginas e documentos sobre LaTeX. Alguns exemplos incluem:

- Favorito inglês: <<http://en.wikibooks.org/wiki/LaTeX/>>
- Favorito português: <<http://linorg.usp.br/CTAN/info/lshort/portuguese/pt-lshort.pdf>>
- <<http://www.mat.ufmg.br/~regi/topicos/intlat.pdf>>
- <<http://www.duke.edu/~hg9/ctex/LaTeXManual.pdf>>
- <<http://minerva.ufpel.tche.br/~campani/cursolatex.pdf>>
- <<http://www.personal.ceu.hu/tex/words.htm>>

---

<sup>4</sup> Ou seja, eles irão alterar a formatação dada pelo estilo default do texto.