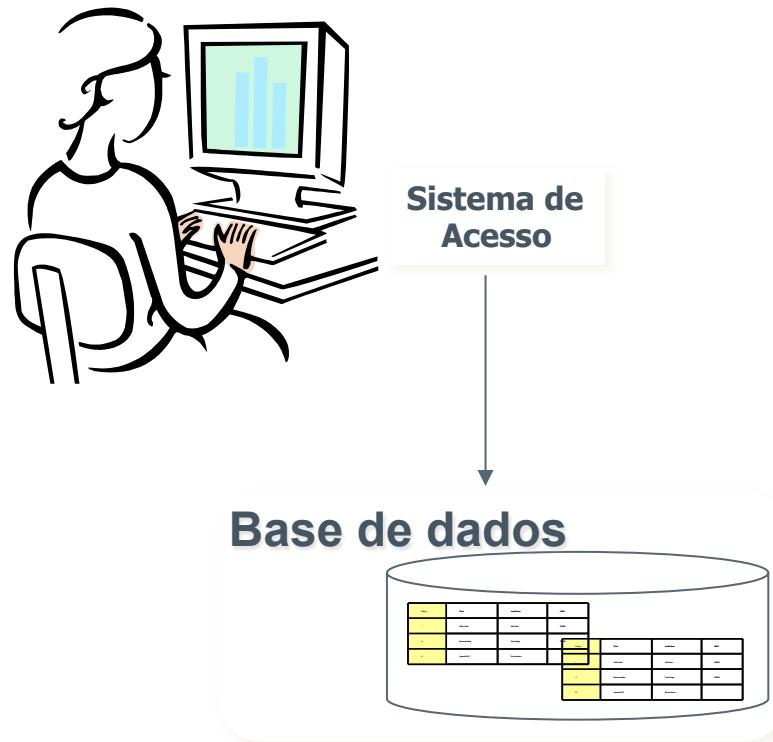


# Funções de Similaridade e algumas aplicações

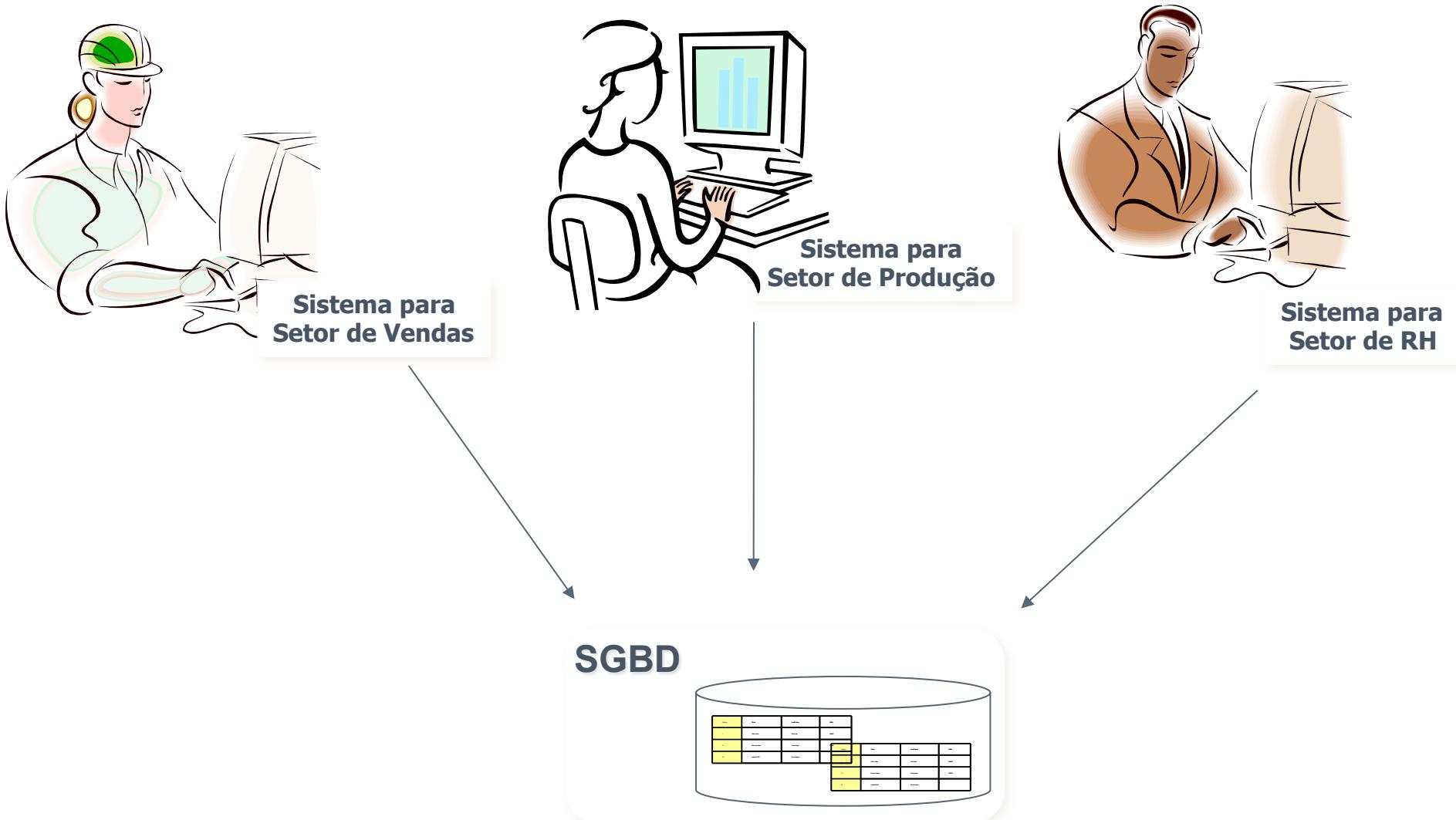
Carina F. Dorneles  
[dorneles@inf.ufsc.br](mailto:dorneles@inf.ufsc.br)

# Introdução – acesso aos dados

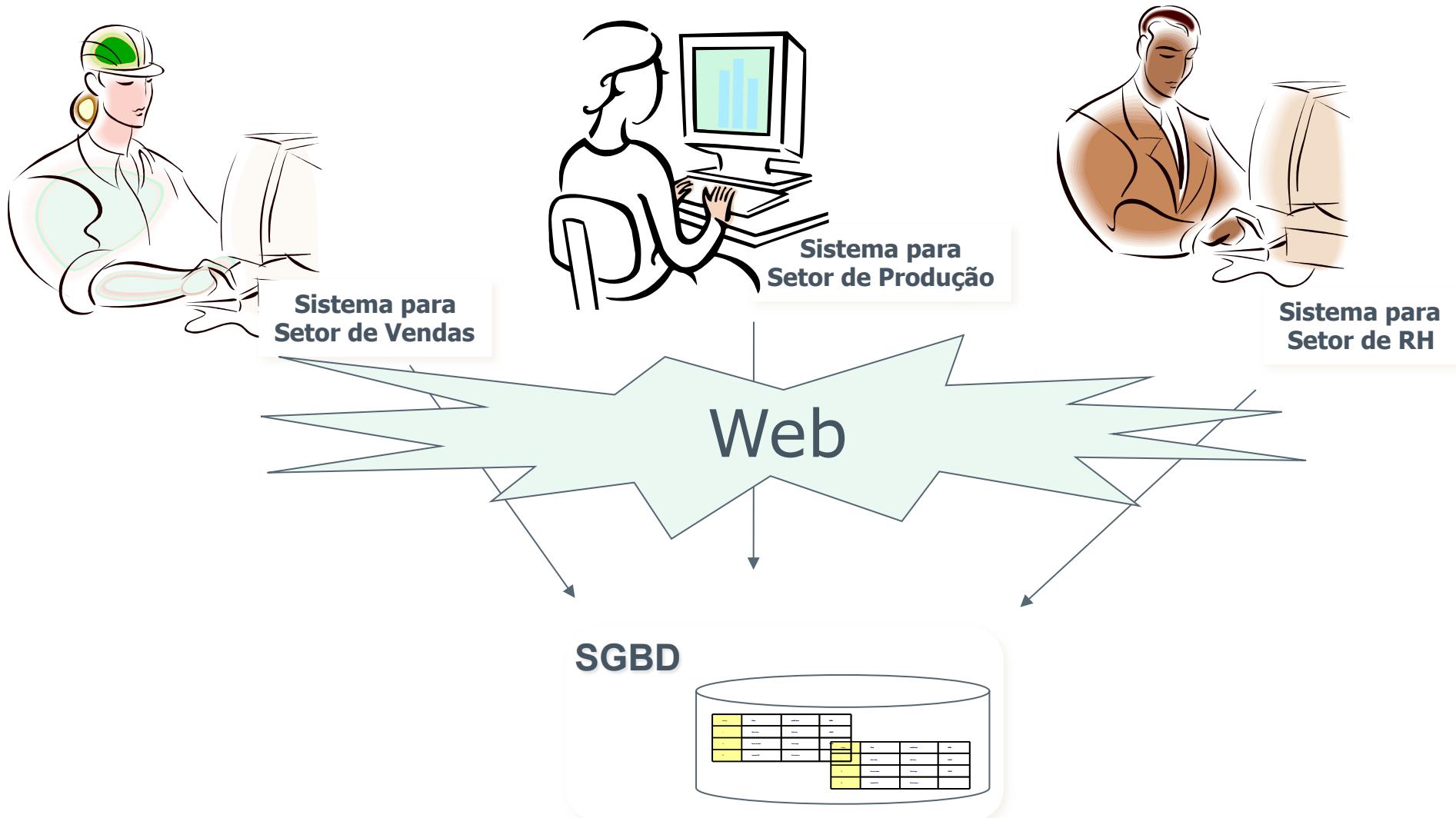
# Introdução – acesso aos dados



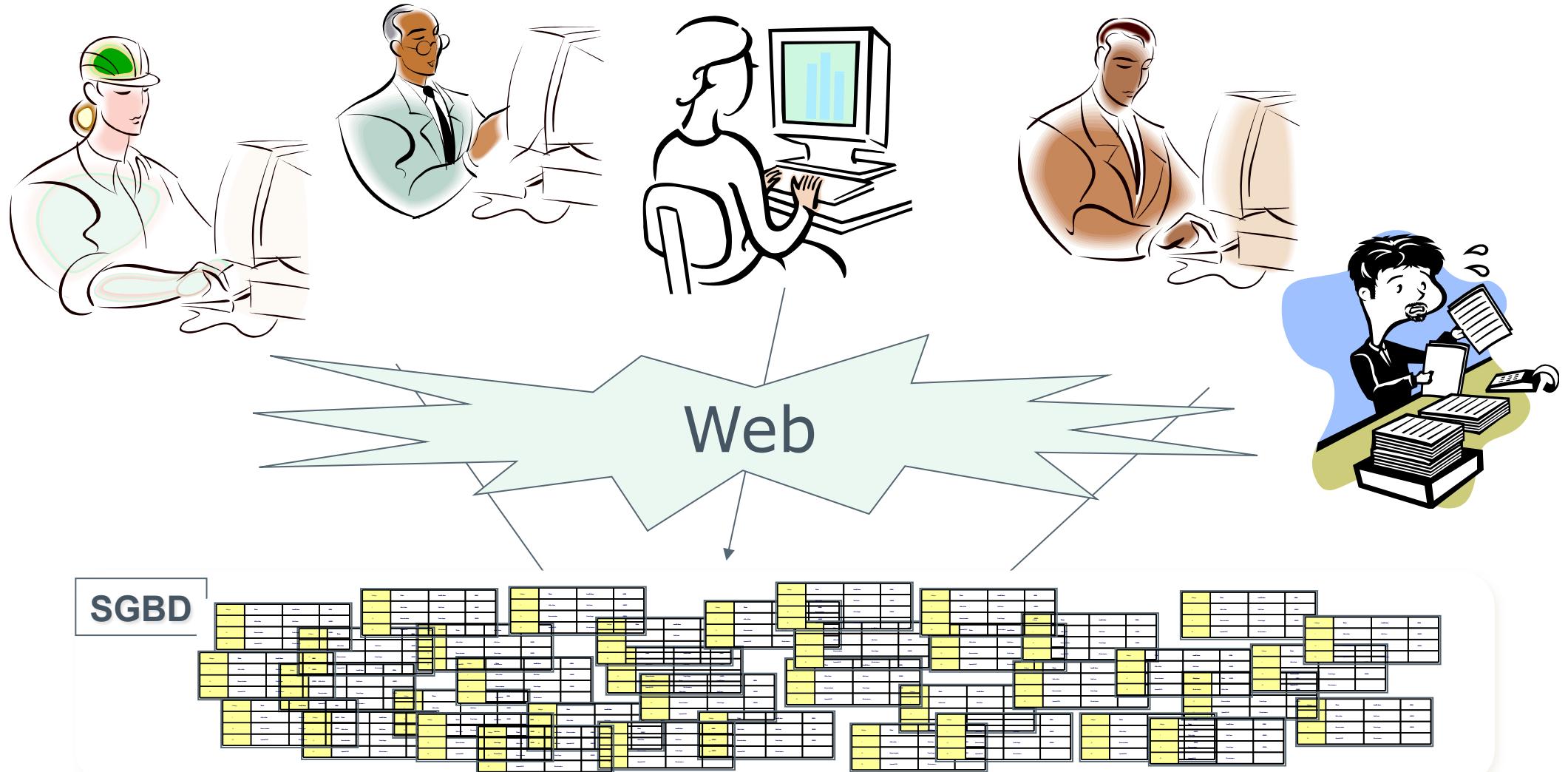
# Introdução – acesso aos dados



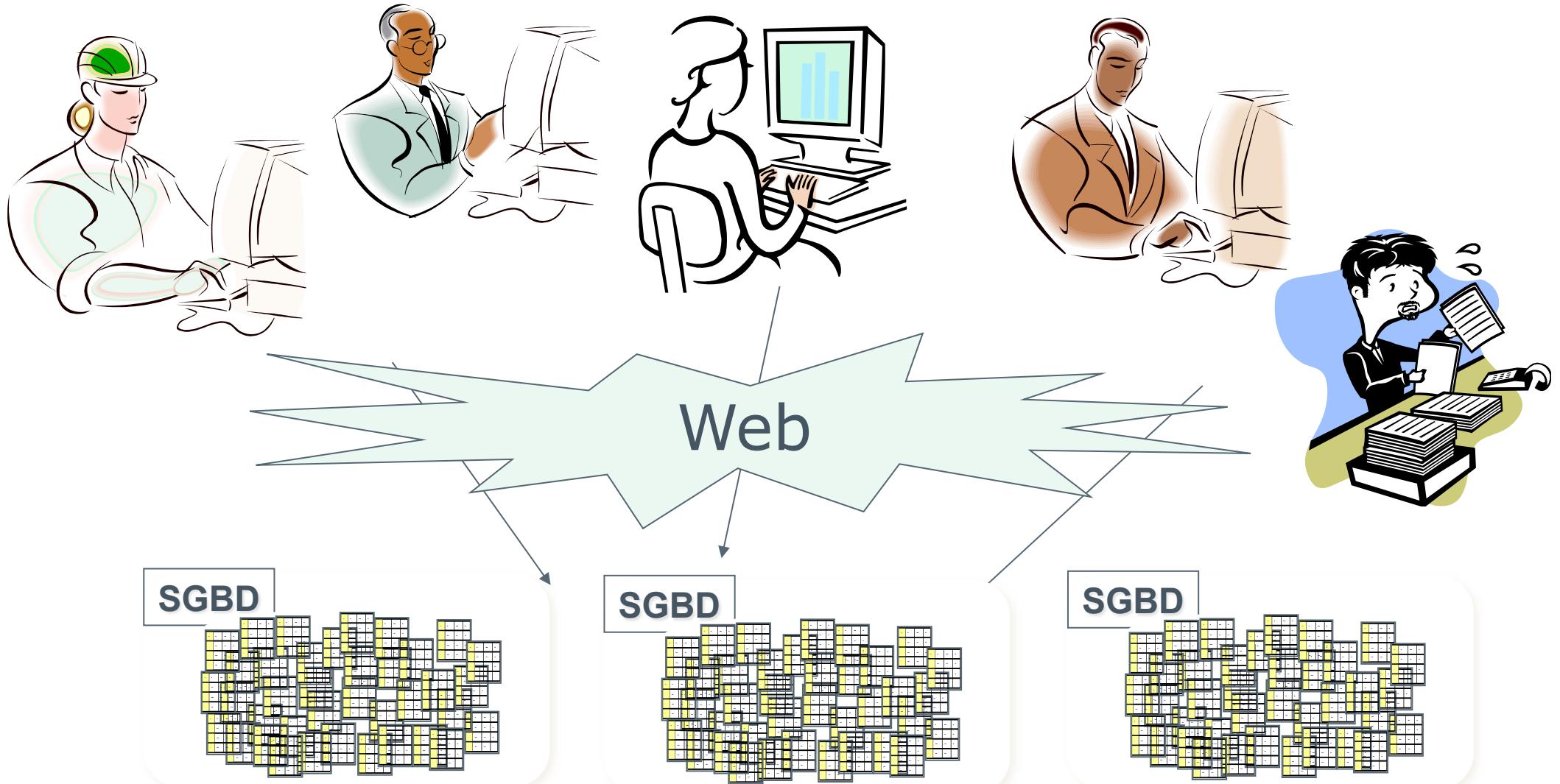
# Introdução – acesso aos dados



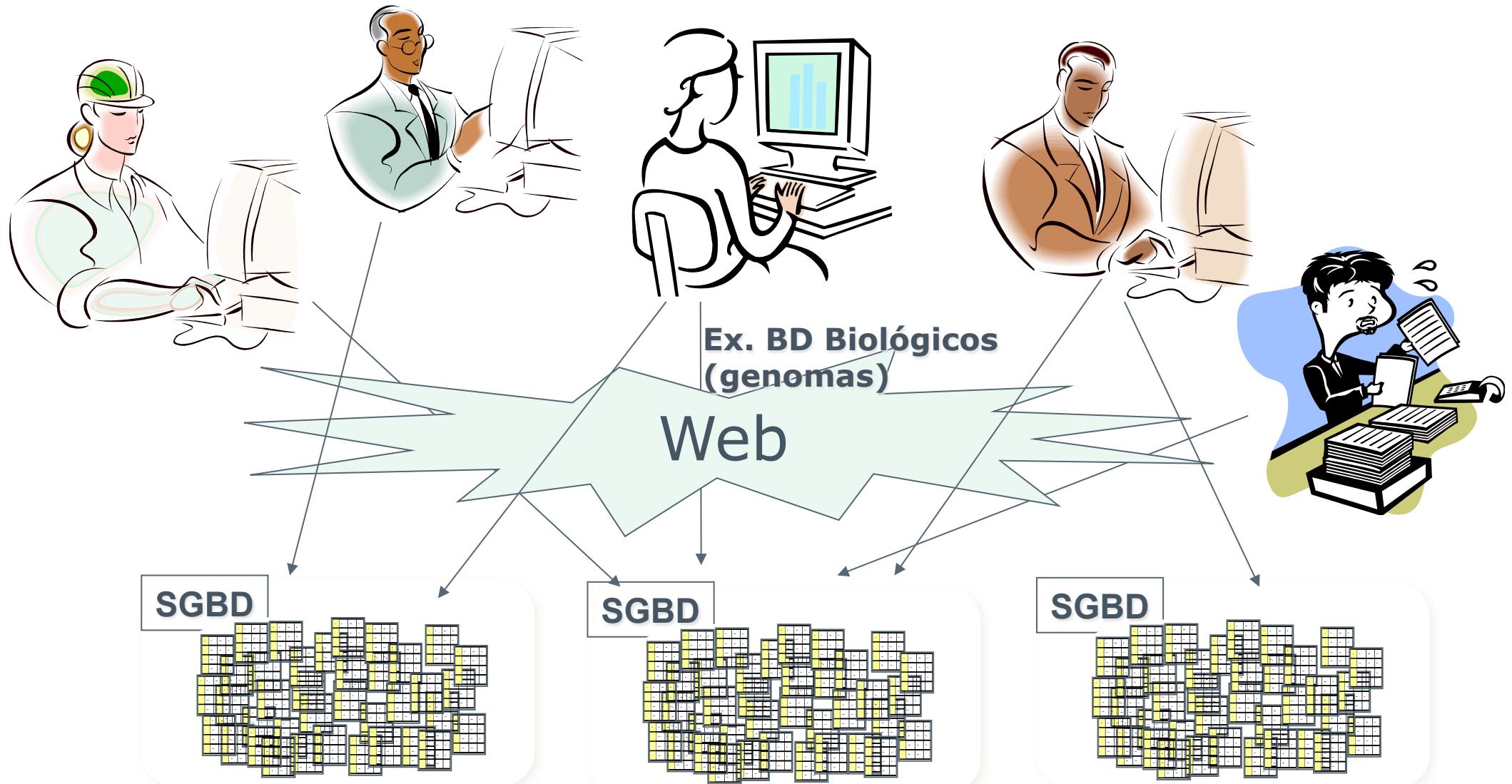
# Introdução – acesso aos dados



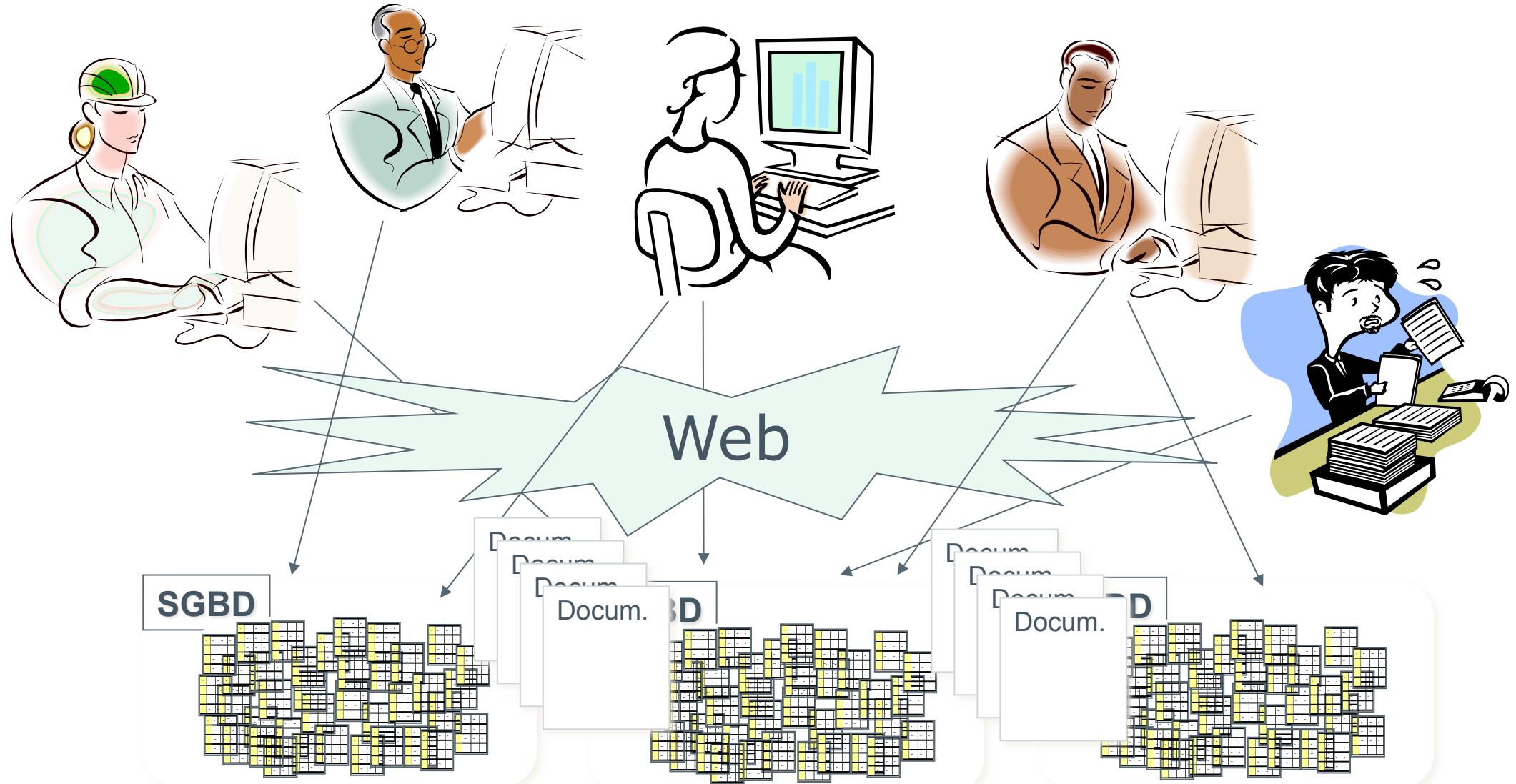
# Introdução – acesso aos dados



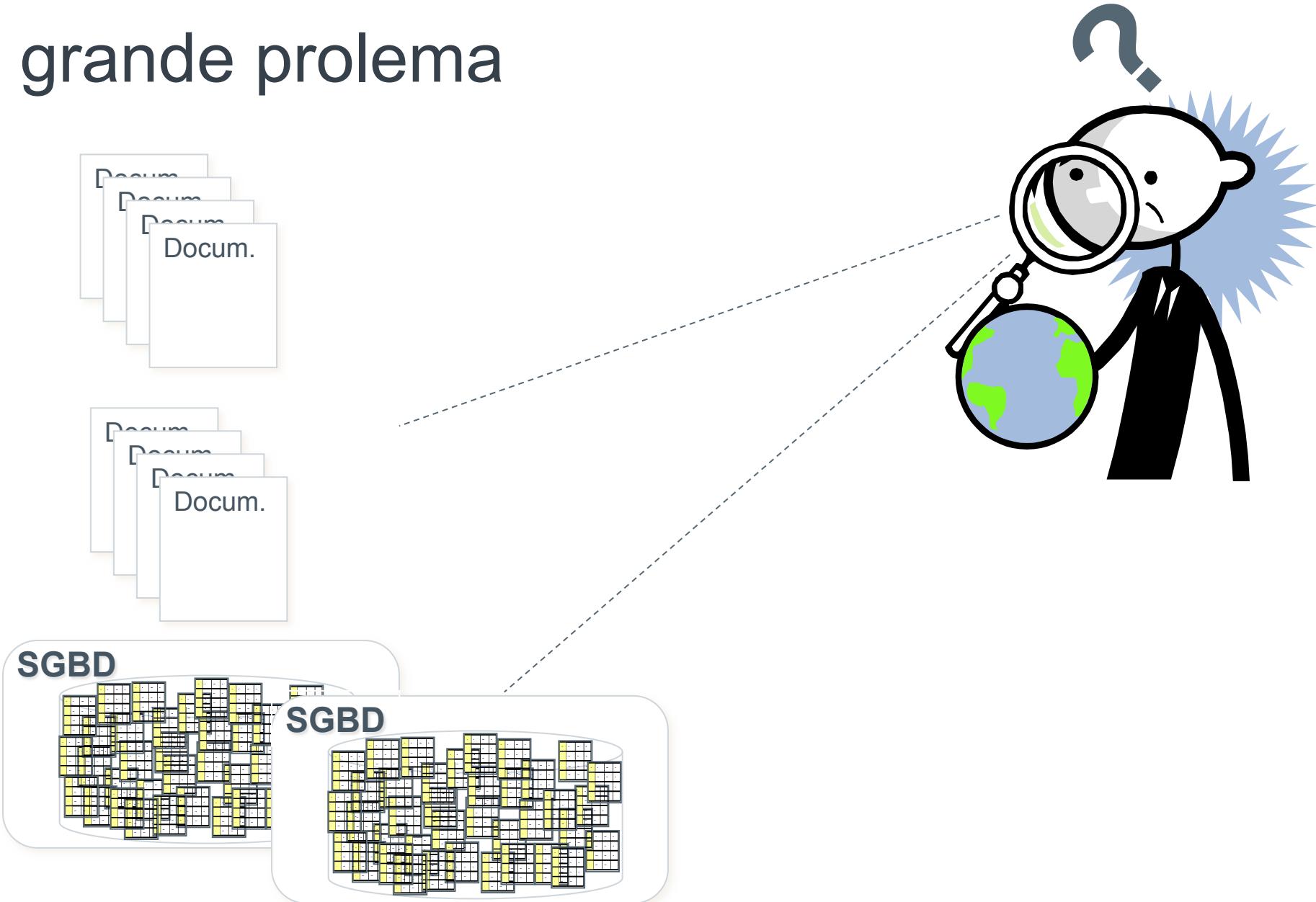
# Introdução – acesso aos dados



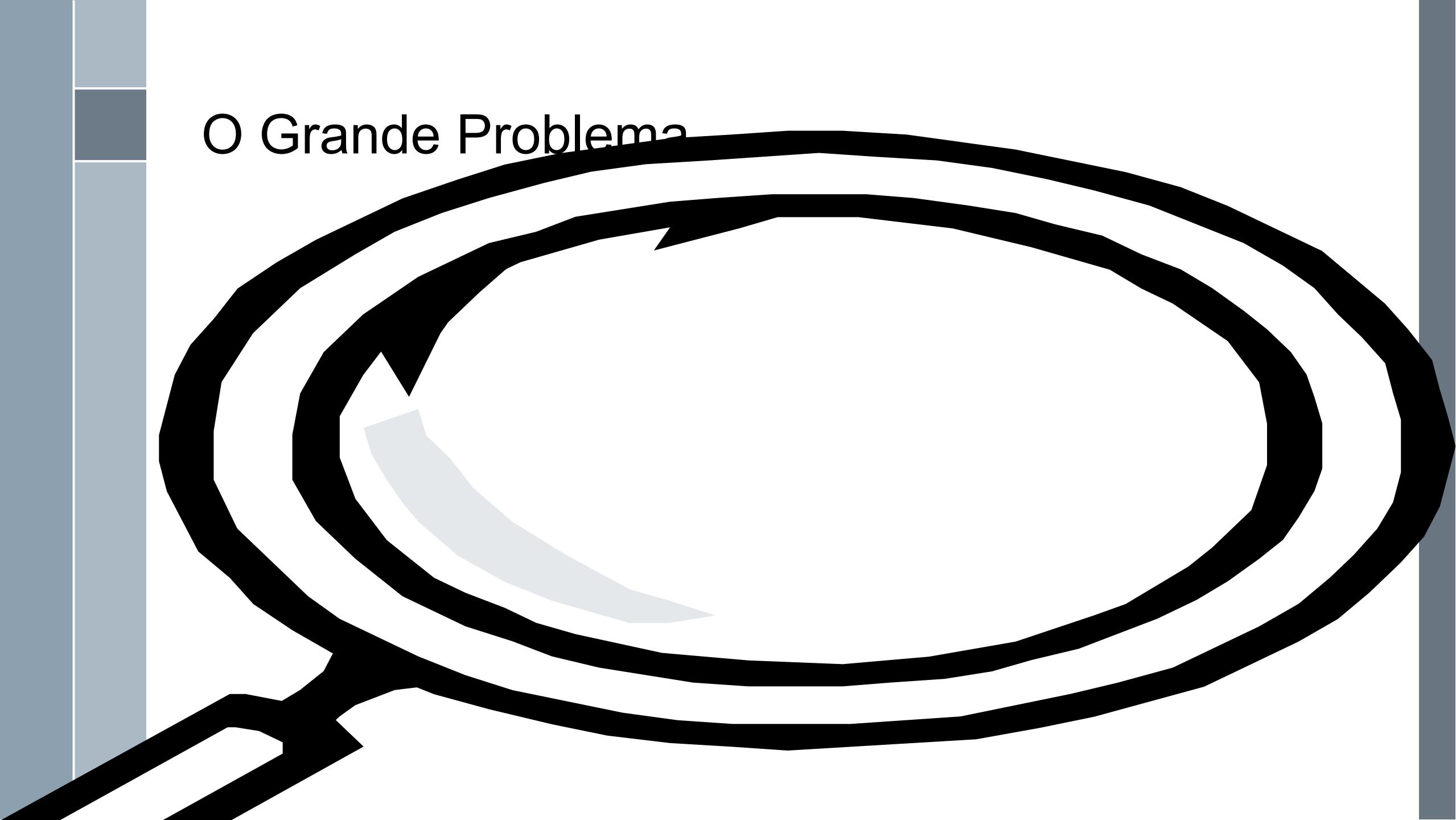
# Introdução – acesso aos dados



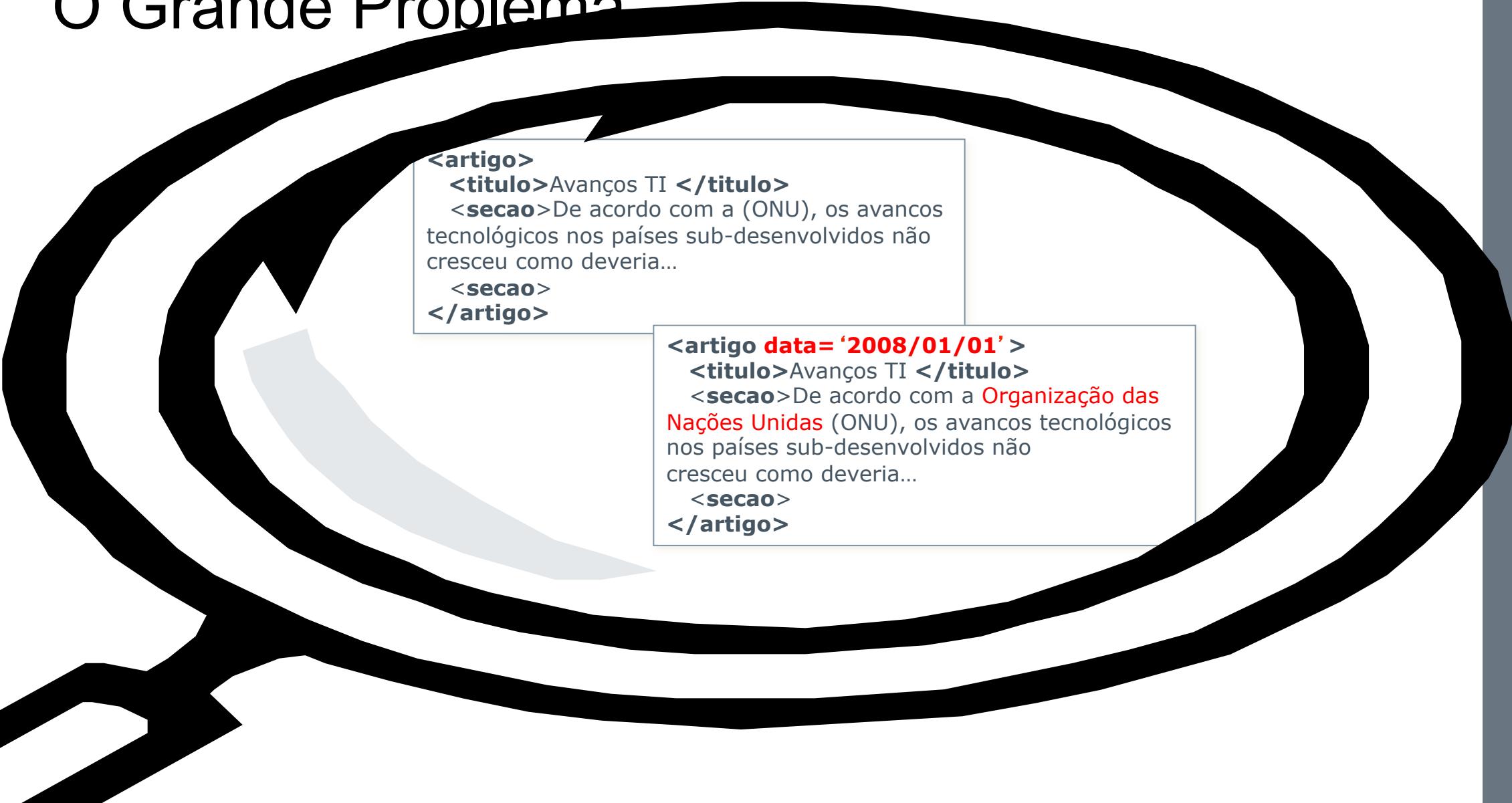
# O grande prolema



# O Grande Problema



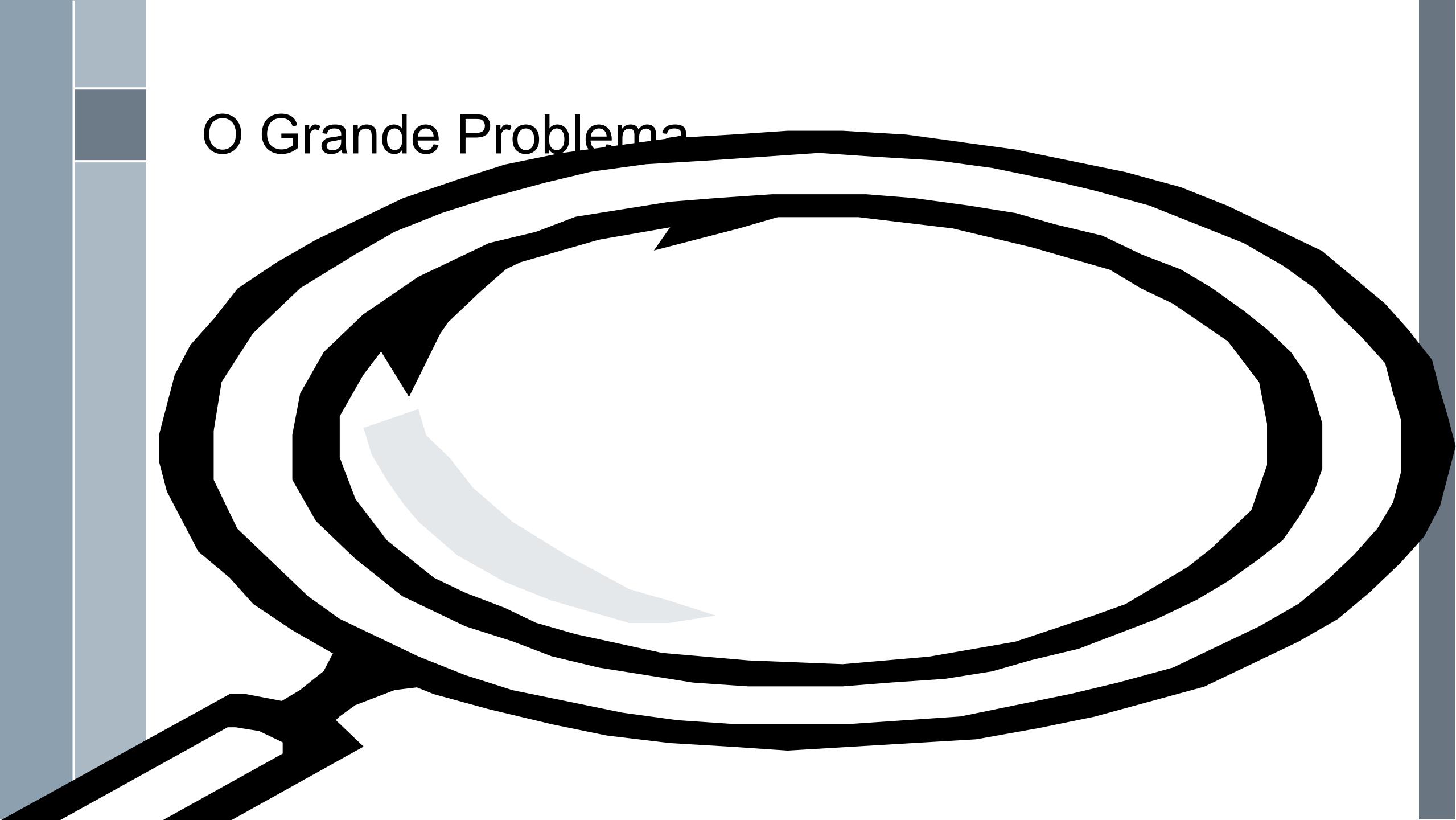
# O Grande Problema



```
<artigo>
  <título>Avanços TI </título>
  <seção>De acordo com a (ONU), os avanços tecnológicos nos países sub-desenvolvidos não cresceu como deveria...
  <seção>
</artigo>
```

```
<artigo data='2008/01/01'>
  <título>Avanços TI </título>
  <seção>De acordo com a Organização das Nações Unidas (ONU), os avanços tecnológicos nos países sub-desenvolvidos não cresceu como deveria...
  <seção>
</artigo>
```

# O Grande Problema



# O Grande Problema

*Person*

<b>Name</b>	<b>Institution</b>	<b>email</b>
Kofi Annan	ONU	kofiannan@...

*Person*

<b>Name</b>	<b>Institution</b>	<b>email</b>
Anann, Kofi	United Nations (UN)	kofi@...

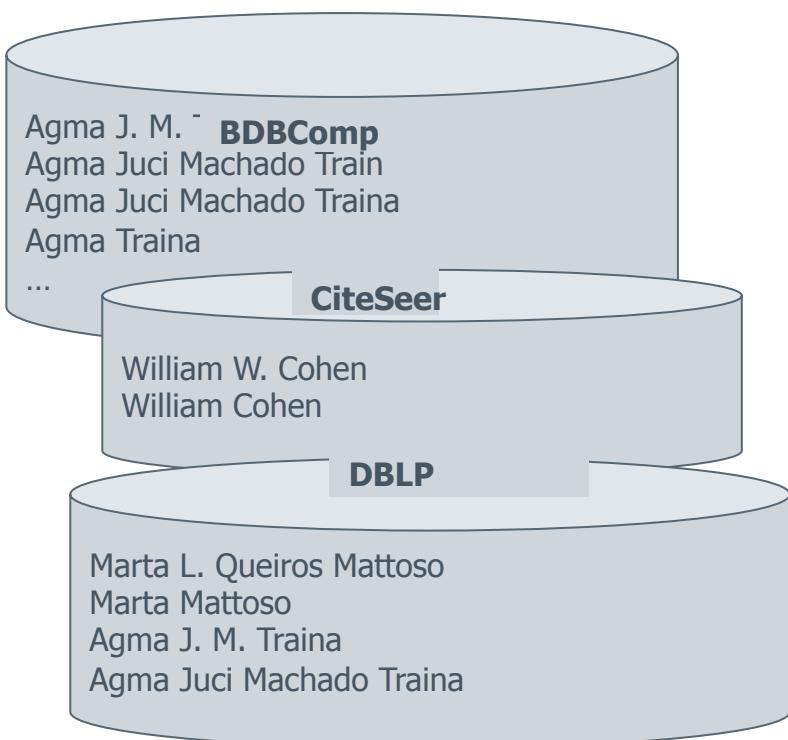
# O grande problema

- › Instâncias de um mesmo objeto do **mundo real** podem possuir diferentes representações

# O grande problema

- › Instâncias de um mesmo objeto do **mundo real** podem possuir diferentes representações

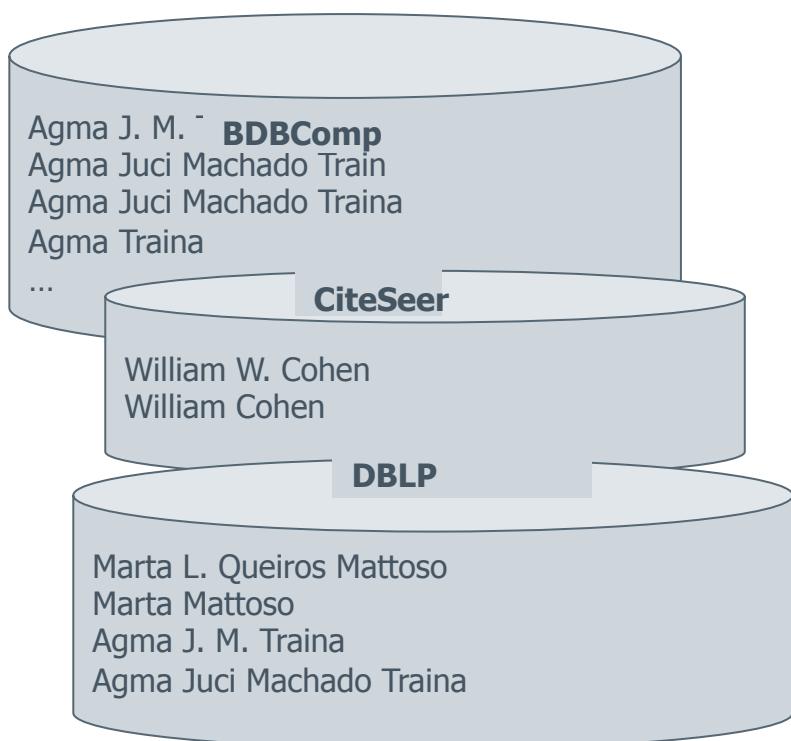
Dados de Citações Bibliográficas



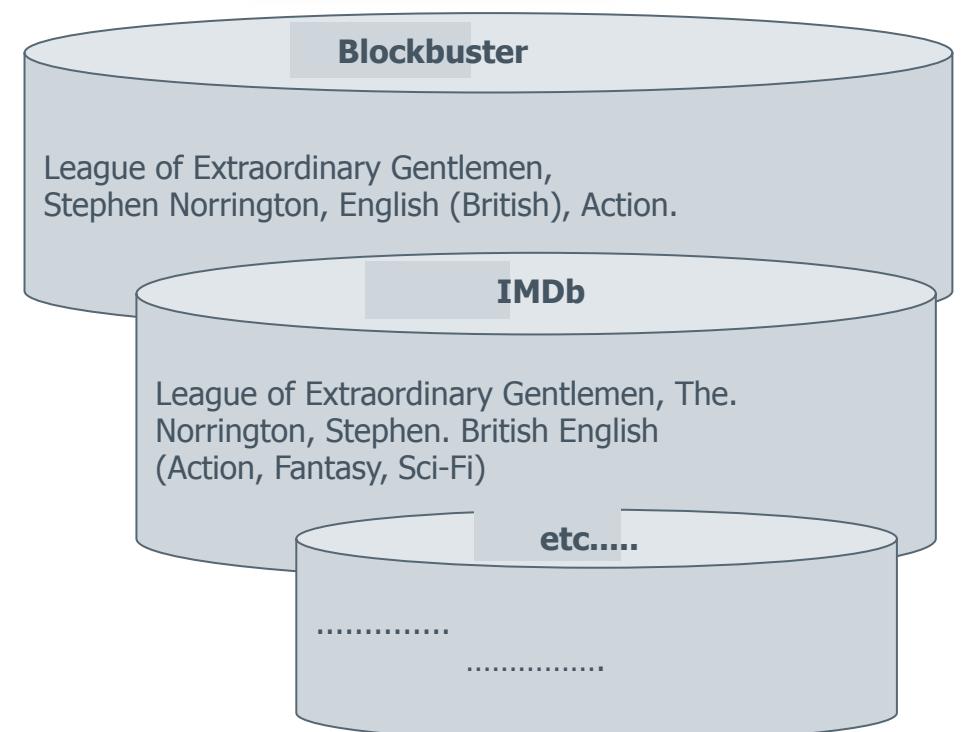
# O grande problema

- › Instâncias de um mesmo objeto do **mundo real** podem possuir diferentes representações

Dados de Citações Bibliográficas



Dados sobre Filmes



# O grande problema

## Artigos Científicos

### Artigo disponibilizado na página do autor

```
<article>
  <title>Using Similarity Functions to Query XML</title>
  <authors>
    <author>Melni, J</author>
    <author>Melni, J</author>
  </authors>
  <section>Similarity functions have been
extensively used for...
  </section>
</article>
```

### Artigo disponibilizado no repositório do evento

```
<article date="2008-01-01">
  <copyright>ACM....</copyright>
  <title>Using Similarity Functions to Query XML</title>
  <authors>
    <author>Melni, J</author>
    <author>Melni, J</author>
  </authors>
  <resumo>This paper presents...</resumo>
  <section>Similarity functions have been extensively used for...
  </section>
</article>
```

# O grande problema

## Artigos Científicos

### Artigo disponibilizado na página do autor

```
<article>
  <title>Using Similarity Functions to Query XML</title>
  <authors>
    <author>Melni, J</author>
    <author>Melni, J</author>
  </authors>
  <section>Similarity functions have been
extensively used for...
  </section>
</article>
```

### Artigo disponibilizado no repositório do evento

```
<article date="2008-01-01">
  <copyright>ACM....</copyright>
  <title>Using Similarity Functions to Query XML</title>
  <authors>
    <author>Melni, J</author>
    <author>Melni, J</author>
  </authors>
  <resumo>This paper presents...
  <section>Similarity functions have been extensively used for...
  </section>
</article>
```

## Arquivos de aulas

### Material encontrado na página do prof.

```
<aula data="2008-07-01">
  <titulo>SQL DDL</titulo>
  <secao titulo="Introdução">
    <subsecao tit="Histórico">
      A Linguagem SQL (Structured Query Language) teve
      seu primeiro padrão definido no ano de 1998.
    </subsecao>
    <secao>
  </artigo>
```

### Artigo disponibilizado no ambiente de ensino

```
<aula data="2008-05-01">
  <titulo>SQL</titulo>
  <secao titulo="Introdução">
    <subsecao tit="Histórico">
      A Linguagem SQL (Structured Query Language) teve
      seu primeiro padrão definido no ano de 1998.
    </subsecao>
    <secao>
  </artigo>
```

# Mas porque isso acontece?



# Por que ?

- Principais fatores
  - Fatores Humanos
    - Entrada de dados incorreta
    - Uso de diferentes padrões
      - Previsão do tempo: EUA usa F, Mundo usa C
  - Fatores Computacionais
    - Aplicações sem restrições populam as bases de dados
    - Projeto do BD Errado ou diferente
      - Diferentes restrições
- **Mundo real é dinâmico**

# Dados

- › Imagem e Vídeo
- › Texto
  - Não estruturados
    - › Páginas HTML
    - › Documentos PDF, PS, TXT, etc...
  - Dados estruturados
    - › Texto curto (Bancos de dados, dados XML)
    - › Texto longo (Documentos XML)

# Manipulação dos dados

- › Não pode ser com o uso do operador de igualdade
- › Deve ser com métricas que identifiquem que os dados tratam da mesma coisa
  - Funções de similaridade
  - Algoritmos de Detecção de Diferença



# Métricas dependem do tipo

- › Imagens e Vídeo
  - Uso de características extraídas dos arquivos
- › Texto
  - Manipulação das strings

# Métricas dependem do tipo

- › Imagens e Vídeo
  - Uso de características extraídas dos arquivos
- › Texto
  - Manipulação das strings

# Imagens

› Exemplo

# Imagens

## › Exemplo



# Imagens

## › Exemplo

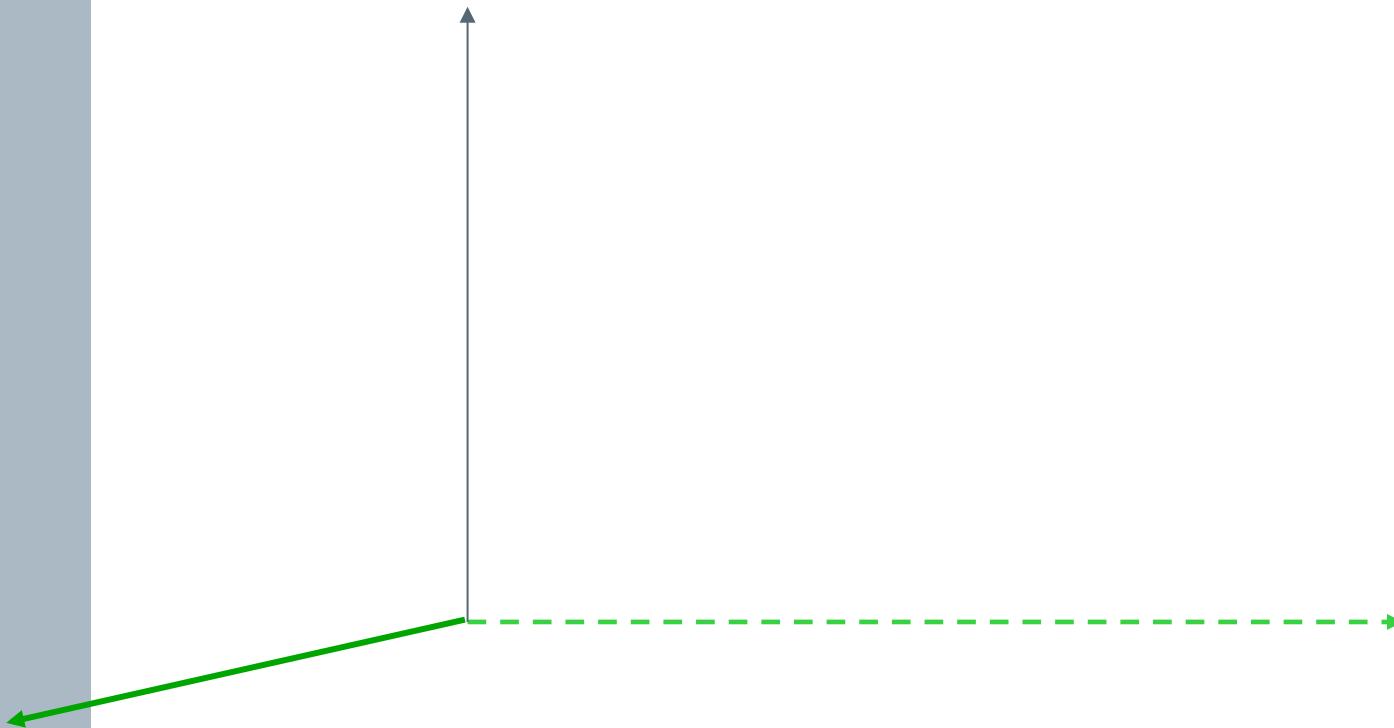


- › Extração de características
  - Cor
  - Textura
  - Formas geométrica
  - ...

# Vetor de Característica

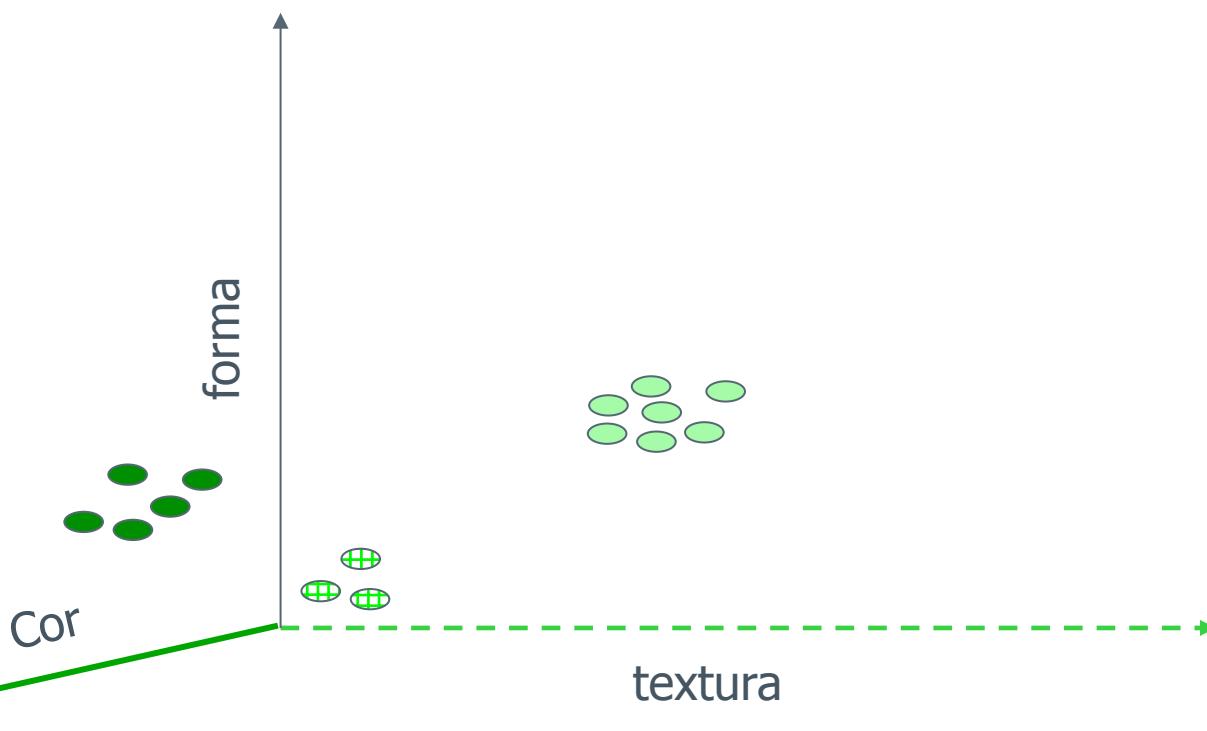
# Vetor de Característica

- Cria-se um espaço **n** dimensional
  - **n** é o número de características consideradas



# Vetor de Característica

- Cria-se um espaço **n** dimensional
  - **n** é o número de características consideradas
  - Cada imagem é transformada em um vetor de características e colocada no espaço



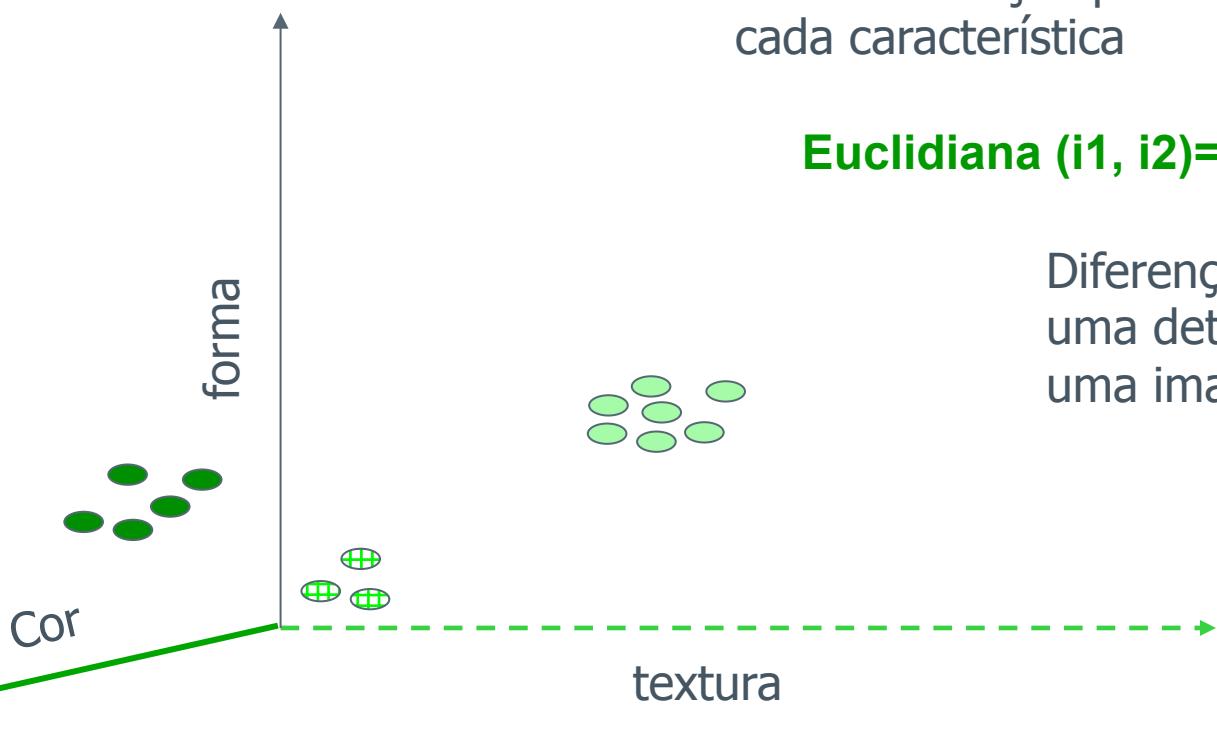
# Vetor de Característica

- Cria-se um espaço **n** dimensional
  - **n** é o número de características consideradas
  - Cada imagem é transformada em um vetor de características e colocada no espaço

Usa uma função para combinar os valores de cada característica

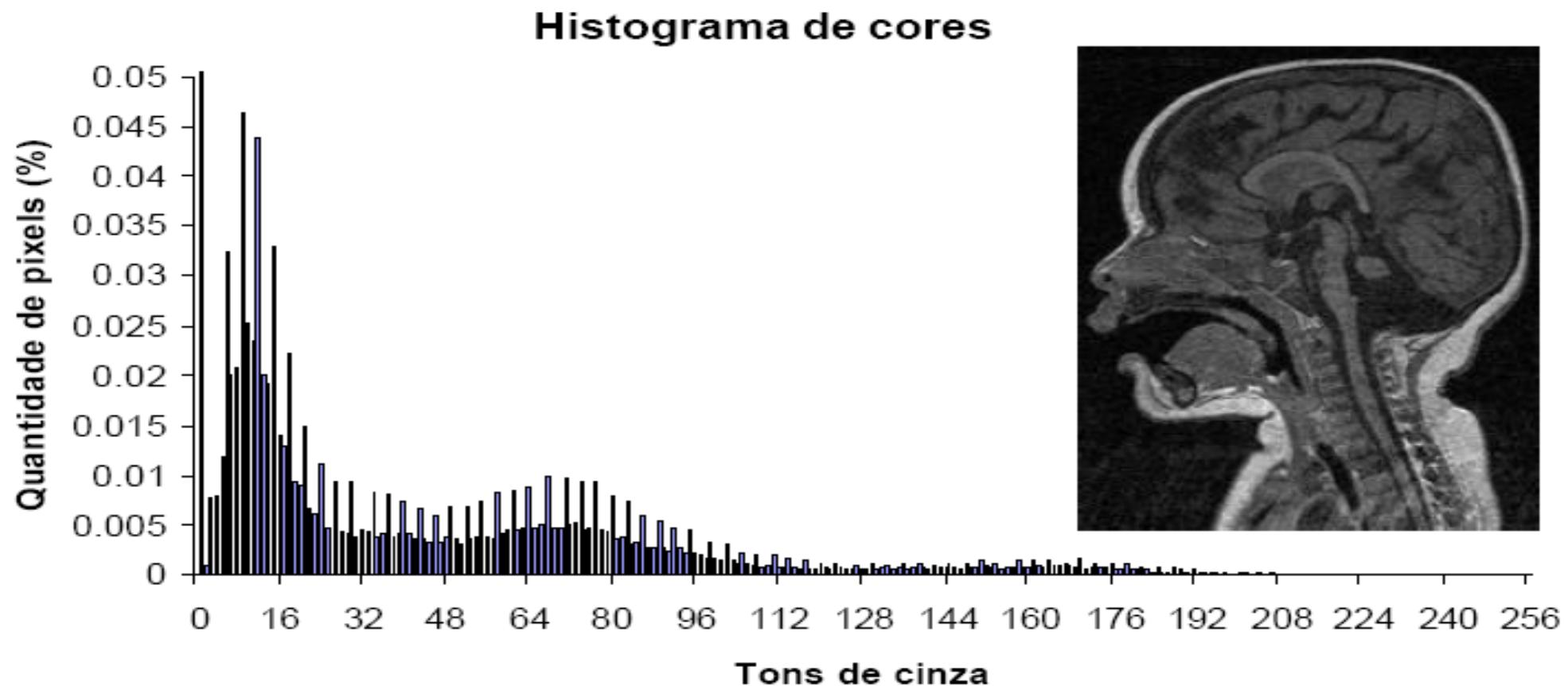
$$\text{Euclidiana } (i_1, i_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Diferença representa a distância entre uma determinada característica de uma imagem em relação a outra



# Extração das características

› Cor – uso de histograma



# Extração das características

› Forma – segmentação da imagem

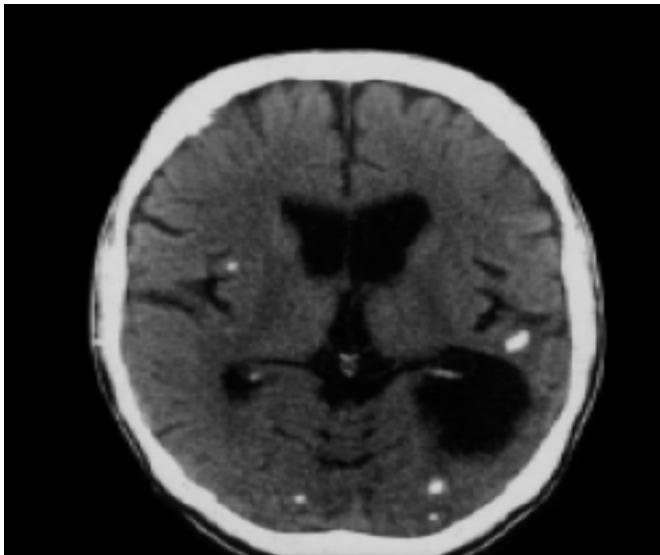


Imagen original

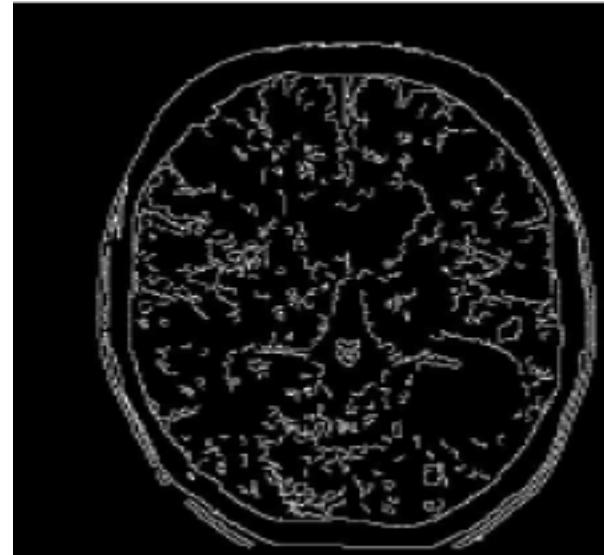


Imagen segmentada

# Extração das características

- › Textura – segmentação da imagem



Imagen original

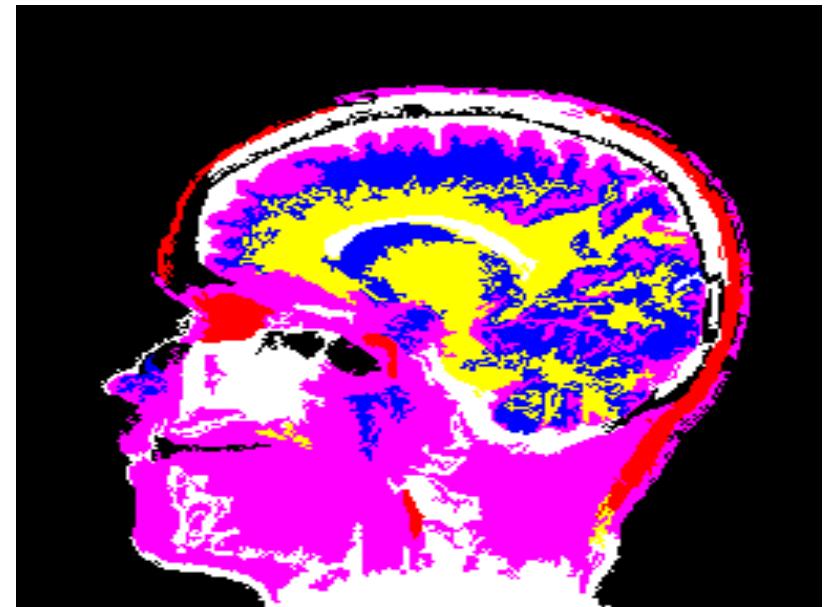


Imagen segmentada

# Métricas dependem do tipo

- › Imagens e Vídeo
  - Uso de características extraídas dos arquivos
- › Texto
  - Manipulação das strings

# Texto

- › Não estruturados ou semi-estruturados
  - › Páginas HTML
  - › Documentos PDF, PS, TXT, etc...
- › Dados estruturados
  - › Texto curto (Bancos de dados, dados XML)
  - › Texto longo (Documentos XML)

# Dados Estruturados

## › Texto curto

- Banco de dados relacionais e dados XML
  - › Strings de texto curto (nome próprio, email, endereço)
    - Nome Próprio: Maria Eduarda
    - Email: me@abc.def.ghi.br
    - Endereço: Av. Brasil 1000

## › Texto longo

- Documentos XML

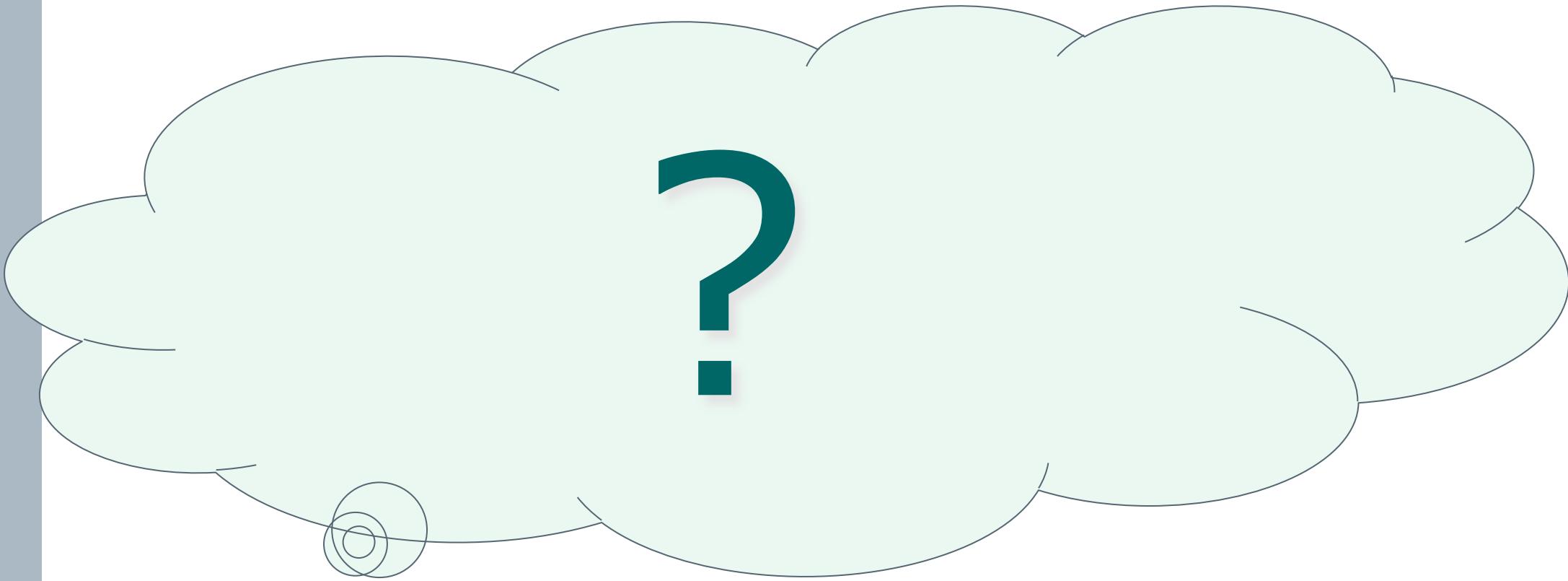
- › Strings de texto longo (Introdução, Seções)

- **Introdução:** Data cleaning is a critical element for developing effective business intelligence applications. The inability to ensure data quality can negatively affect downstream data analysis and ultimately key business decisions. A very important data cleaning operation is that of identifying records which match the same real world entity. For example, owing to various errors in data and to differences in conventions of representing data, product names in sales records may not match exactly with records in master product catalog tables. In these situations, it would be desirable to match similar records across relations. This problem of matching similar records

- **Seção 1:** As discussed earlier, several efficient similarity join algorithms have been recently proposed in the context of record matching (e.g., [22, 26, 13, 2]). Koudas et al. present an excellent tutorial of various similarity functions and similarity join algorithms in the context of record matching [27]. In contrast to our similarity join based combination of similarity function values  $\bigvee_i f_i > i$ , another natural predicate would have been a thresholded linear combination of all similarity function values being greater than a threshold :  $\sum_i w_i f_i > \theta$ . This combination is adopted by SVM models [17]. As discussed earlier, one of our design goals is to ensure that our primitive operators are efficiently executable. We are not aware of any method to efficiently perform a join between two large input relations

# *Funções de Similaridade*

# Como calcular a similaridade?



# O que é similaridade?

# O que é similaridade



# O que é similaridade



Similaridade é difícil de definir, mas...  
“sabemos quando a vemos”

O real significado da similaridade é uma questão filosófica. É necessário adotar uma visão mais pragmática.

# Diferentes áreas, diferentes focos

- › Algumas aplicações com dados textuais e/ou estruturados
  - Fazem desambiguação
    - › Identificar o que trata o **mesmo objeto do mundo real**
    - › Usa as funções de similaridade como um substituto do operador de igualdade
- › Aplicações com dados binários (imagens, sons, vídeos)
  - Consultar **objetos similares**
    - › Não necessariamente o mesmo
- › Aplicações de tomada de decisão
  - Identificar **objetos similares**
    - › Que objeto se *comporta* de forma similar ao outro

# Contexto de uso

- › Consulta
  - Executar consultas sobre uma ou várias bases de dados
- › Integração
  - Efetuar Integração de dados
- › *Data Cleaning*
  - Efetuar *Data Cleaning* em ambientes de *Data Warehouse*

# Consultas

- › Como efetuar consulta sobre uma base que possui diferentes representações do mesmo objeto?
  - Exemplo usando um dialeto similar a SQL

# Consultas

- › Como efetuar consulta sobre uma base que possui diferentes representações do mesmo objeto?
  - Exemplo usando um dialeto similar a SQL



# Consultas

- › Como efetuar consulta sobre uma base que possui diferentes representações do mesmo objeto?
  - Exemplo usando um dialeto similar a SQL



Recuperar artigos do autores 'Agma Machado Traina'



# Consultas

- › Como efetuar consulta sobre uma base que possui diferentes representações do mesmo objeto?
  - Exemplo usando um dialeto similar a SQL



```
SELECT artigo  
FROM BDBComp  
WHERE levenshtein(autor, 'Agma Machado Traina') > 0,75
```

Recuperar artigos do autores 'Agma Machado Traina'



# Integração de Dados

- › Como integrar dados que são escritos de diferentes formas?

Blockbuster

League of Extraordinary Gentlemen,  
**Stephen Norrington**, English (British), **Action**.

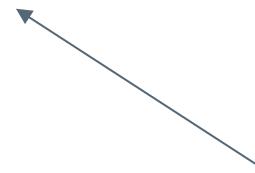
IMDb

League of Extraordinary Gentlemen, The.  
**Norrington, Stephen**. British English  
(**Action, Fantasy, Sci-Fi**)

# Integração de Dados

- › Como integrar dados que são escritos de diferentes formas?

Levenshtein ("League of ... Action", "League of ... Sci-Fi")  $\geq 0.78$



Blockbuster

League of Extraordinary Gentlemen,  
**Stephen Norrington**, English (British), **Action**.

IMDb

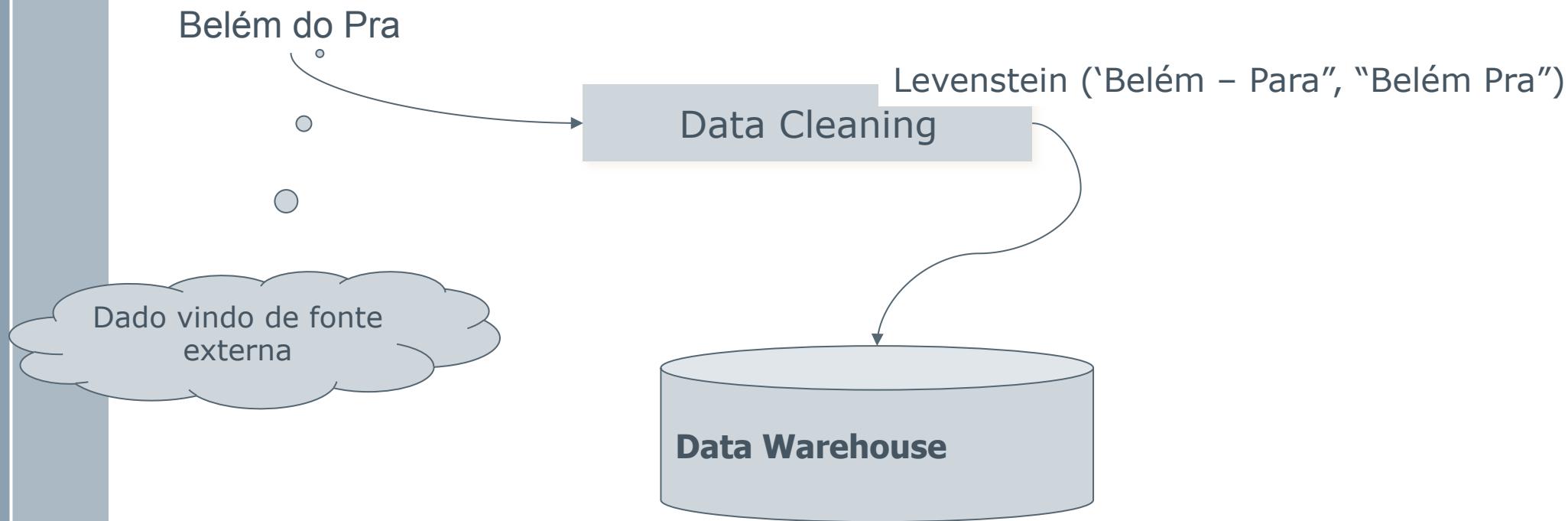
League of Extraordinary Gentlemen, The.  
**Norrington, Stephen**. British English  
(**Action, Fantasy, Sci-Fi**)

# Data Cleaning

- › Dados vindos de fontes externas podem conter inconsistências

# Data Cleaning

- › Dados vindos de fontes externas podem conter inconsistências



# Conceitos Básicos

- › Similaridade vs. Distância
- › Limiar ou ponto de corte (*threshold*)
- › Avaliação de uma função
  - Revocação
  - Precisão
  - Discernibilidade

# Similaridade Vs. Distância

# Similaridade Vs. Distância

- › Uma função de similaridade  $fs(a_1, a_2) \rightarrow s$ 
  - Escore  $s$  no intervalo  $[0, 1]$ .
  - Quanto **maior** o valor do escore, **mais similares** os dois valores  $a_1$  e  $a_2$  são entre si.

# Similaridade Vs. Distância

- › Uma função de similaridade  $fs(a1, a2) \rightarrow s$ 
  - Escore  $s$  no intervalo  $[0, 1]$ .
  - Quanto **maior** o valor do escore, **mais similares** os dois valores  $a1$  e  $a2$  são entre si.
- › Uma função de distância  $fd(a1, a2) \rightarrow s$ 
  - Escore  $s$  no intervalo  $[0, \infty]$ .
  - Quanto **menor** o valor do escore, **mais similares** os dois valores  $a1$  e  $a2$  são entre si.

# Similaridade Vs. Distância

- › Exemplo: sejam duas strings
  - $s1 = \text{"Linus B. Torvalds"}$
  - $s2 = \text{"Linus B. Tolvards"}$

# Similaridade Vs. Distância

- › Exemplo: sejam duas strings
  - $s_1 = \text{"Linus B. Torvalds"}$
  - $s_2 = \text{"Linus B. Tolvards"}$
- › Com uma função de distância  $fd()$  qualquer
  - A distância será:  $fd(s_1, s_2) = 0,1176$ .
- › Com uma função de similaridade  $fs()$  qualquer
  - O escore de similaridade será:  $fs(s_1, s_2) = 0,8824$ .

# Similaridade Vs. Distância

- › Transformando um valor de distância em similaridade
- › Similaridade =  $1 - (\text{valor de distância normalizado})$

$s_1 = \text{"Linus B. Torvalds"}$

$s_2 = \text{"Linus B. Tolvards"}$

$$fd(s_1, s_2) = 0.1176$$

$$fs(s_1, s_2) = 1 - (fd(s_1, s_2))$$

$$fs(s_1, s_2) = 1 - 0.1176$$

$$fs(s_1, s_2) = 0.8824$$

# Um problema que surge...

- Exemplo: Supondo uma aplicação de integração de dados



# Um problema que surge...

- Exemplo: Supondo uma aplicação de integração de dados



Noção de proximidade entre os valores

Dada por um valor de escore,  
gerado por uma função de similaridade.

**Intervalo de [0,1]**



**Não define com exatidão que valor indica quando duas instâncias são consideradas representação do mesmo objeto do mundo real**

# Um problema que surge...

- Exemplo: Supondo uma aplicação de integração de dados



Noção de proximidade entre os valores

Dada por um valor de escore,  
gerado por uma função de similaridade.

**Intervalo de  $[0,1]$**



Para delimitar o que é considerado  
Similar:  
valor de LIMIAR

# Uso de funções de similaridade

Valores de escores gerados pelas funções são pouco significativos ao usuário

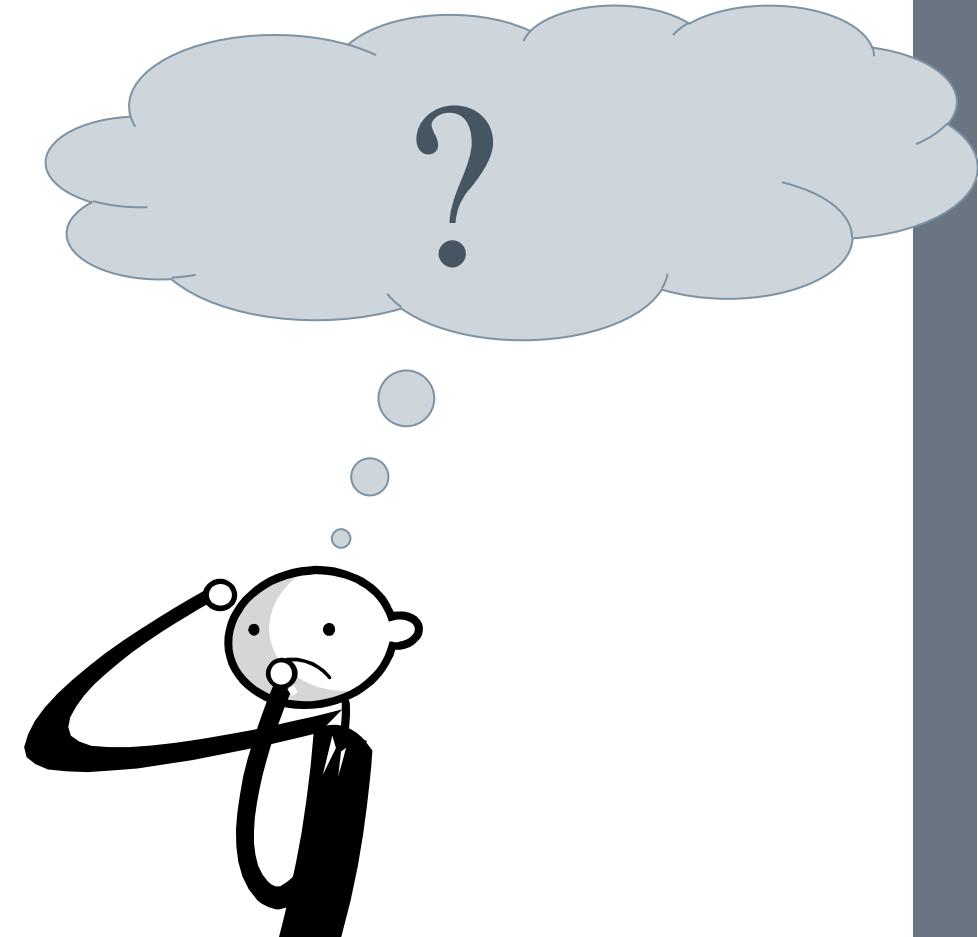
**Difícil definir um limiar adequado**



# Uso de funções de similaridade

Valores de escores gerados pelas funções são pouco significativos ao usuário

Difícil definir um limiar adequado



# Uso de funções de similaridade

Valores do escoramento  
função de similaridade

O problema ainda encontra-se  
em aberto e tem sido trabalhado  
em diversos centros de pesquisa

Magical

0.875

Magical

0.714

Magical

0.714

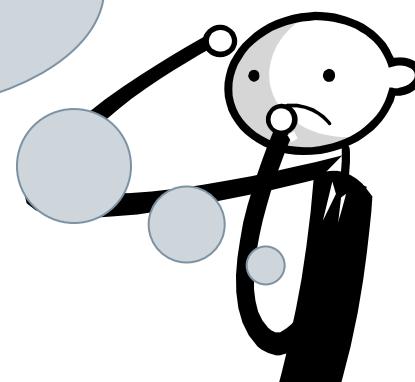
Magical

Magical

Magical

Magic

Musical



# Conceitos Básicos

- › Similaridade vs. Distância
- › Limiar ou ponto de corte (*threshold*)
- › Avaliação de uma função
  - Revocação
  - Precisão
  - Discernibilidade

# Avaliação

- › Porque avaliar:
  - funções podem não apresentar um resultado perfeito na comparação de valores
    - › um par de valores que não deveria ser considerado similar pode ter um escore de similaridade mais alto do que o limiar e portanto aparecer no resultado final (falso positivo). Da mesma forma, um par de valores que deveria ser considerado similar pode não aparecer no resultado (falso negativo)
  - Uma função pode ser boa para um domínio de valores e ruim para outro

# Avaliação

- › Revocação (*Recall*)
- › Precisão (*Precision*)
- › Discernibilidade

# Revocação e Precisão [BAE99]

- › Avaliação é feita sobre uma base de dados conhecida
- › Parâmetros
  - $N$  = conjunto de instâncias, existentes no banco, que são relevantes (identificadas por especialistas)
  - $R$  = conjunto de instâncias relevantes retornadas pelo sistema

# Parâmetros usados para avaliação

- › Considerando uma base de dados com *nomes de cidade e estado*:

# Parâmetros usados para avaliação

- › Considerando uma base de dados com *nomes de cidade e estado*:

**Cidade**

<b>nome</b>	<b>Estado</b>
Porto Alegre	RS
Bela Vista	PR
P. Alegre	RS
Porto Alegre	Rio Grande do Sul
Porto Alegre	Rio G. do Sul
Belém	PA
Porto Lucena	RS
Belém	Pará
PoA	RS
Pouso Alegre	RS

**P**A

Passo 1: Para cada consulta usada na avaliação, especialista deve marcar o que é relevante.

**avaliação**

*es de cidade e estado:*

### **Cidade**

<b>nome</b>	<b>Estado</b>
Porto Alegre	RS
Bela Vista	PR
P. Alegre	RS
Porto Alegre	Rio Grande do Sul
Porto Alegre	Rio G. do Sul
Belém	PA
Porto Lucena	RS
Belém	Pará
PoA	RS
Pouso Alegre	RS

**P**A

Passo 1: Para cada consulta usada na avaliação, especialista deve marcar o que é relevante.

**avaliação**

com nomes de cidade e

## **Cidade**

<b>nome</b>	<b>Estado</b>
Porto Alegre	RS
Bela Vista	PR
P. Alegre	RS
Porto Alegre	Rio Grande do Sul
Porto Alegre	Rio G. do Sul
Belém	PA
Porto Lucena	RS
Belém	Pará
PoA	RS
Pouso Alegre	RS

PoA

Por exemplo, considerando a consulta:  
“Porto Alegre”, “RS”

## Evolução

com nomes de cidade e

### Cidade

<b>nome</b>	<b>Estado</b>
Porto Alegre	RS
Bela Vista	PR
P. Alegre	RS
Porto Alegre	Rio Grande do Sul
Porto Alegre	Rio G. do Sul
Belém	PA
Porto Lucena	RS
Belém	Pará
PoA	RS
Pouso Alegre	RS

<b>Relevante</b>
Sim
Não
Sim
Sim
Sim
Não
Não
Não
Sim
Não

## Avaliação

Passo 2: PRECISÃO - para cada posição no ranking, calcular:  
Número de itens relevantes naquela posição / posição

Consulta.

<b>nome</b>	<b>Estado</b>
Porto Alegre	RS
P. Alegre	RS
Porto Alegre	Rio Grande do Sul
Porto Alegre	Rio G. do Sul
Pouso Alegre	RS
PoA	RS
Bela Vista	PR
Belém	PA
Porto Lucena	RS
Belém	Pará

<b>Itens Relevantes</b>	<b>Posição</b>	<b>Precisão</b>
1	1	1
2	2	1
3	3	1
4	4	1
4	5	0.8
5	6	0.8333333333
5	7	0.714285714
5	8	0.625
5	9	0.5555555556
5	10	0.5

A  
No re

Passo 3: REVOCAÇÃO - para cada posição  
no ranking, calcular:

Número de itens relevantes naquela  
posição / número total de relevan-

Aqui são  
5

Consulta: "Po

<b>nome</b>	<b>Estado</b>
Porto Alegre	RS
P. Alegre	RS
Porto Alegre	Rio Grande do Sul
Porto Alegre	Rio G. do Sul
Pouso Alegre	RS
PoA	RS
Bela Vista	PR
Belém	PA
Porto Lucena	RS
Belém	Pará

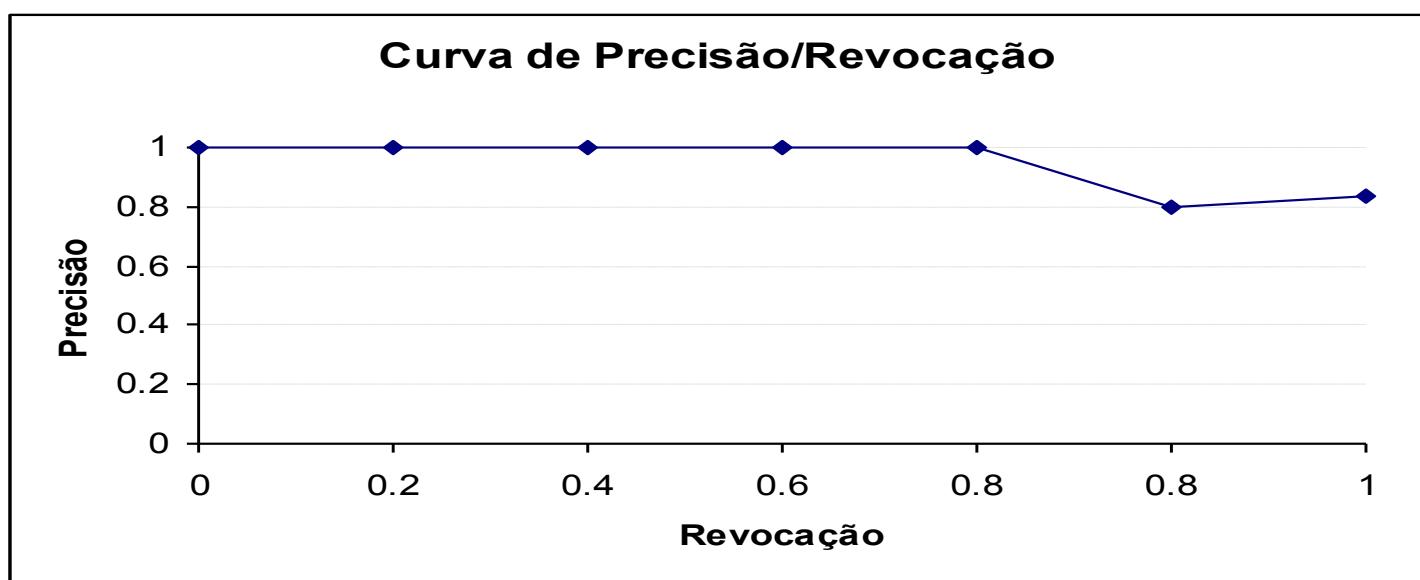
<b>Itens Relevantes</b>	<b>Revocação</b>
1	0.2
2	0.4
3	0.6
4	0.8
4	0.8
5	1
5	1
5	1
5	1
5	1

# Relação de precisão/revocação

- › Para facilitar a avaliação dos resultados:
  - Gráfico que mostra a evolução da precisão em função da revocação.
    - curva de precisão e revocação

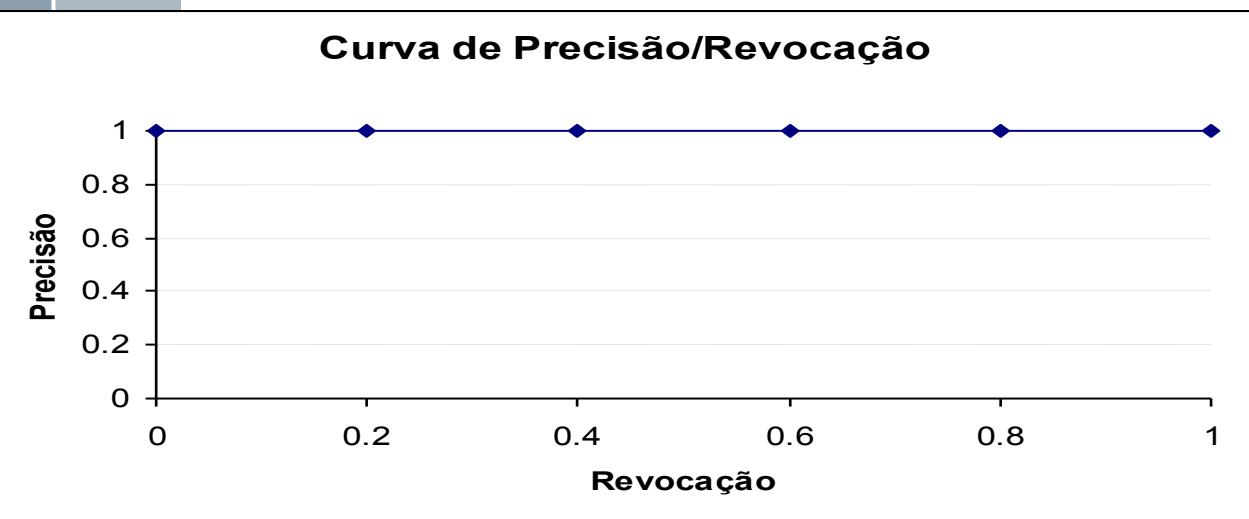
# Gráfico de precisão/revocação

<b>Nome</b>	<b>Estado</b>	<b>Precisão</b>	<b>Revocação</b>
Porto Alegre	RS	1	0.2
P. Alegre	RS	1	0.4
Porto Alegre	Rio Grande do Sul	1	0.6
Porto Alegre	Rio G. do Sul	1	0.8
Pouso Alegre	RS	0.8	0.8
PoA	RS	0.8333333333	1
Bela Vista	PR	0.714285714	1
Belém	PA	0.625	1
Porto Lucena	RS	0.5555555556	1
Belém	Pará	0.5	1



# Gráfico de precisão/revocação

## › Resultado ideal



Indica que todos os relevantes estão no topo do ranking

<b>Nome</b>	<b>Estado</b>	<b>Precisão</b>	<b>Revocação</b>
Porto Alegre	RS	1	0.2
P. Alegre	RS	1	0.4
Porto Alegre	Rio Grande do Sul	1	0.6
Porto Alegre	Rio G. do Sul	1	0.8
PoA	RS	1	1
Pouso Alegre	RS	0.8333333333	1
Bela Vista	PR	0.714285714	1
Belém	PA	0.625	1
Porto Lucena	RS	0.5555555556	1
Belém	Pará	0.5	1

# Avaliação

- › Revocação (*Recall*)
- › Precisão (*Precision*)
- › Discernibilidade

# Discernibilidade

- › Leva em conta dois pontos
  - Se a função separou os itens relevantes dos não relevantes
    - › Uma boa função atribui escores altos aos relevantes e escores baixos aos não relevantes
  - Grau de separação entre os relevantes e não relevantes
    - › Uma boa função deve separar o resultado em dois conjuntos bem distintos

# Discernibilidade

- › Leva em conta dois pontos
  - Se a função separou os itens relevantes dos não relevantes
    - › Uma boa função atribui escores altos aos relevantes e escores baixos aos não relevantes
  - Grau de separação entre os relevantes e não relevantes
    - › Uma boa função deve separar o resultado em dois conjuntos bem distintos

# Discernibilidade

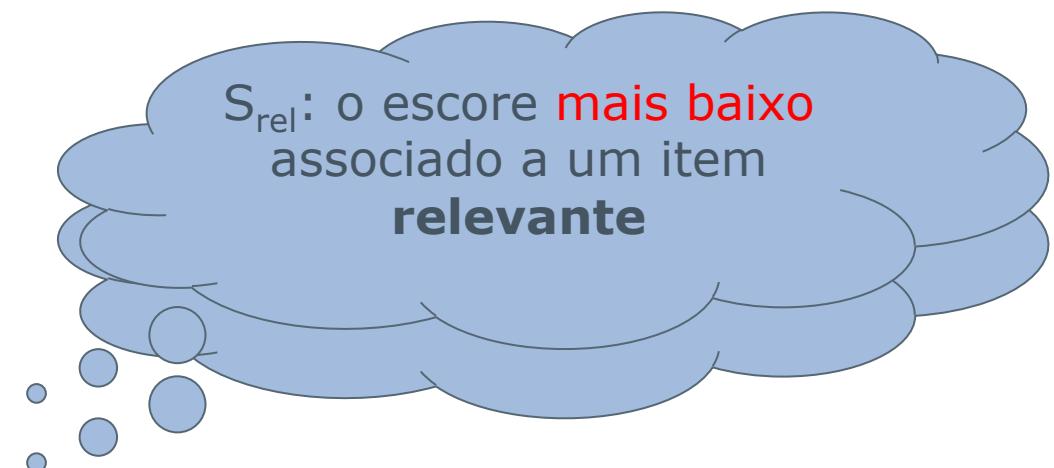
- › Habilidade de uma função de similaridade em separar relevantes e irrelevantes
- › Baseada na distância entre:
  - Escore mínimo do elemento relevante:  $S_{rel}$
  - Escore máximo do elemento irrelevante:  $S_{irrel}$

# Exemplo

- › Sendo a consulta: “Porto Alegre”
  - É um ranking gerado para um função  $\text{Sim}()$

## *Cidade*

<b>Nome</b>	<b><i>Sim()</i></b>
Porto Alegre	1
Porto Alerge	0.92
Porto Alege	0.89
Pouso Alegre	0.68
P. Alegre	0.6
Porto Lucena	0.6
PoA	0.5
Bela Vista	0.3
Belém	0.2
Belém	0.2



# Exemplo

- Executa o processo para  $n$  consultas (em torno de 40) com diferentes valores, sobre o mesmo atributo usando a mesma função.

**Cidade**

<b>Nome</b>	<b>Sim()</b>
Porto Ale	

**Cidade**

<b>Nome</b>	<b>Cidade</b>
Porto Ale	

<b>Nome</b>	<b>Cidade</b>
Pouso Al	Mato Gross

<b>Nome</b>	<b>Cidade</b>
P. Alegre	M. Grosso

<b>Nome</b>	<b>Cidade</b>
Porto Luc	M.G. do Su

<b>Nome</b>	<b>Cidade</b>
PoA	R.G. do Sul

<b>Nome</b>	<b>Cidade</b>
Bela Vista	Rio Grande

<b>Nome</b>	<b>Cidade</b>
Belém	Rio G. do S

<b>Nome</b>	<b>Cidade</b>
Belém	Mato Gross

<b>Nome</b>	<b>Cidade</b>
	MGS

<b>Nome</b>	<b>Cidade</b>
	MG

<b>Nome</b>	<b>Cidade</b>
	M. Grosso

**Cidade**

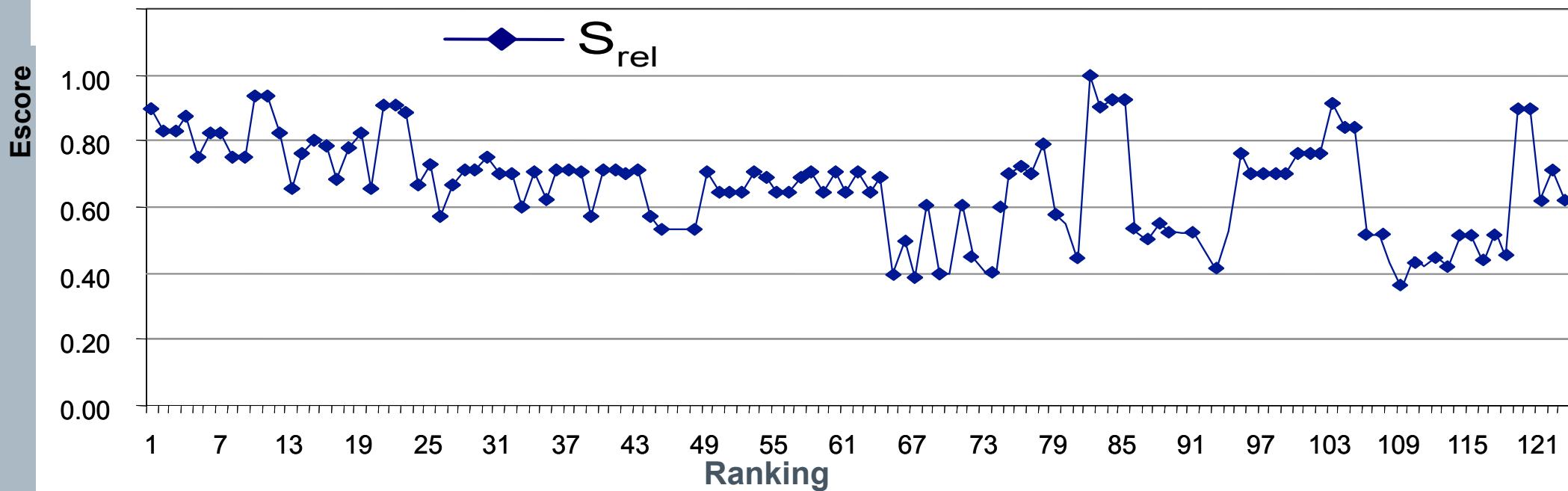
<b>Nome</b>	<b>Sim()</b>
Belém	1
Belem	0.97
Blem	0.91
Marajo	0.54
Marajó	0.56
Marjo	0.49
Manaus	0.36
Manas	0.36
Maunas	0.32
Manasu	0.31

...



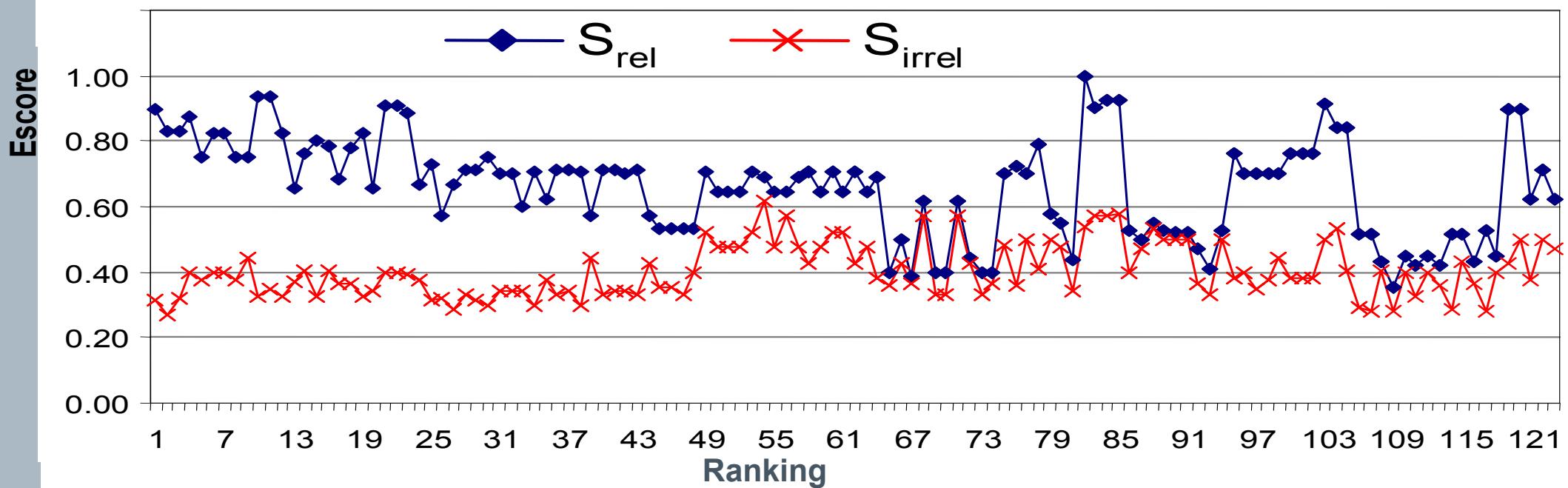
# Exemplo

› Plotar os valores em um gráfico



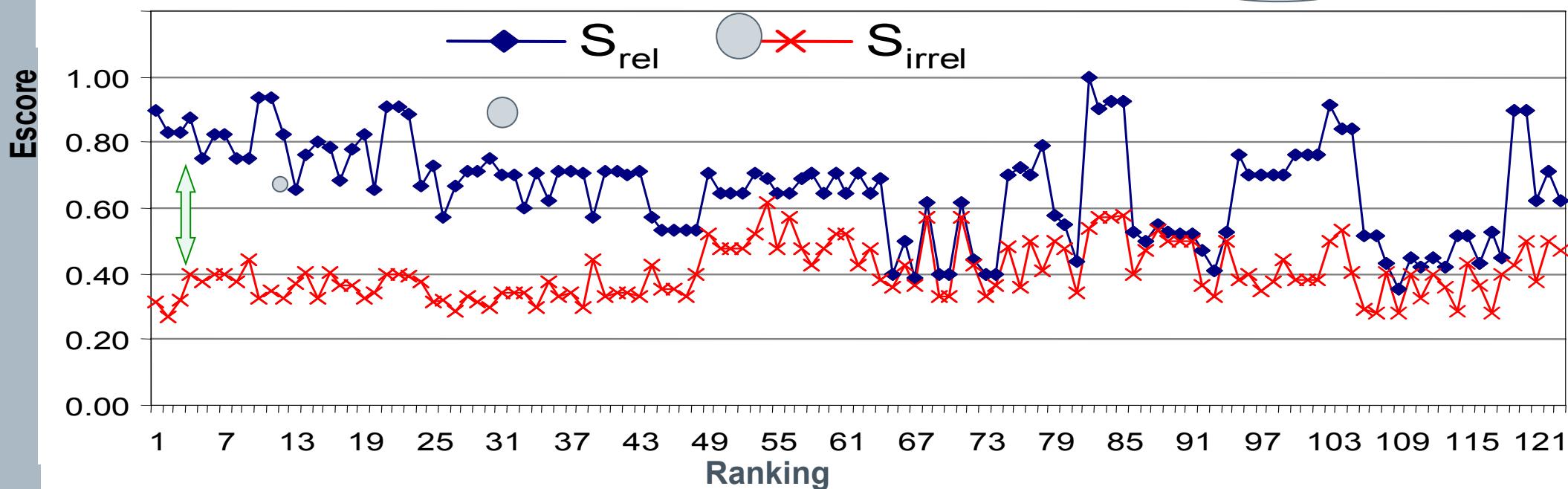
# Exemplo

› Plotar os valores em um gráfico

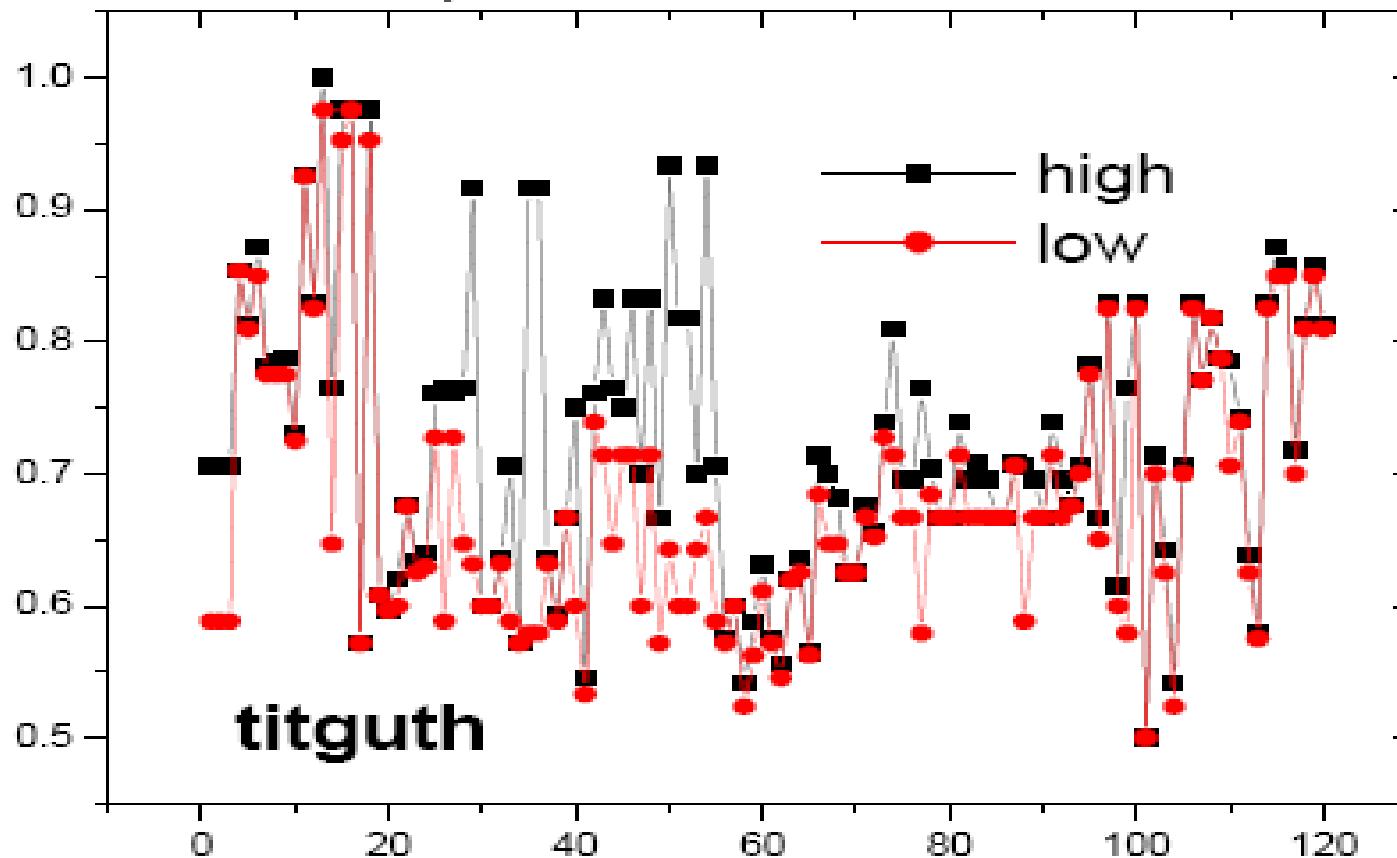


# Exemplo

Quanto maior a distância entre  
as duas linhas, melhor é a  
função de similaridade



## Exemplo – função de similaridade ruim



# Classificação

- › Valores atômicos
  - Baseadas em *Caracter*
  - Baseadas em *Token*
- › Valores agregados
  - Uso de expressões algébricas
  - Uso de algoritmos

# Valores atômicos

- › Baseadas em *caracter*
  - Efetuam a comparação caractere a caractere
- › Baseadas em *token*
  - Efetuam a comparação *token* a *token* (*token-based*), ou seja, a função analisa cada *token* individualmente em uma *string*

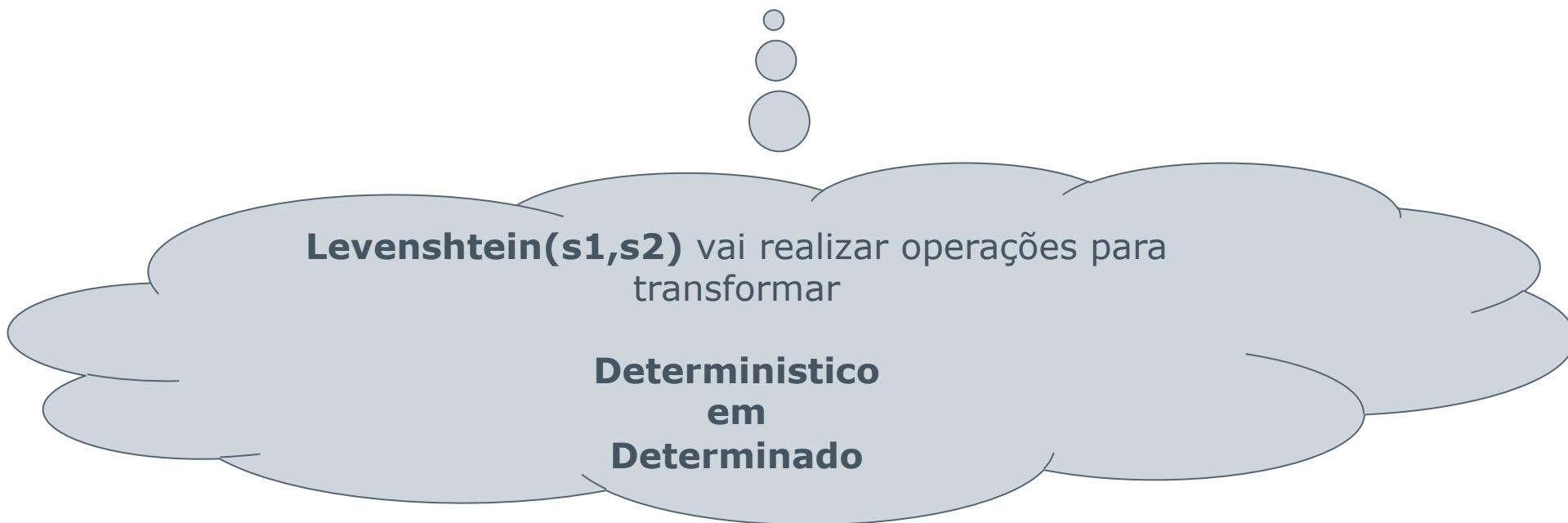
# Levenshtein

- › Função Levenshtein()

- Originalmente, é uma **função de distância** que calcula o número de operações necessárias para transformar uma *string* em outra
- Para usá-la como função de similaridade precisamos
  - › Normalizar o valor da distância
  - › Reduzir o valor de distância resultante de 1

# Levenshtein

- › SimLev = 1 - 
$$\frac{\text{Levenshtein } (s_1, s_2)}{\max (\text{size } (s_1), \text{size}(s_2))}$$
- Exemplo:  $s_1$  = deterministico e  $s_2$  = determinado



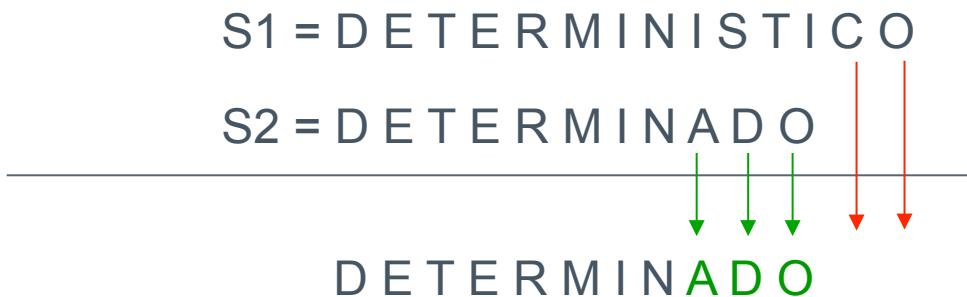
# Levenshtein

- › SimLev = 1 - 
$$\frac{\text{Levenshtein } (s_1, s_2)}{\max (\text{size } (s_1), \text{size}(s_2))}$$
- Exemplo:  $s_1 = \text{deterministico}$  e  $s_2 = \text{determinado}$

S1 = D E T E R M I N I S T I C O  
S2 = D E T E R M I N A D O

---

D E T E R M I N A D O



Operações:

replace  
delete

Operações:

- ↓ replace
- ↓ delete

# Levenshtein

- › Assim, tem-se

S1 = D E T E R M I N I S T I C O

S2 = D E T E R M I N A D O

DETERMINA D O

- › 5 operações (3 *replaces* e 2 *deletes*)
- › Levenshtein (s1, s2) = 5
- › max (size (s1), size(s2)) = 14
- ›  $\text{SimLev} = 1 - \frac{5}{14}$
- › **SimLev ('deterministico', 'determinado') = 0,64285714**

# Acronyms

## › Função *Acronyms()*

- Transforma cada string no seu respectivo acrônimo, usando iniciais ou letras maiúsculas, e em seguida usa *Levenshtein()* para retornar um valor de escore

# Acronyms

› Exemplo:

s1 = 'CSBC' e s1= 'Congresso da SBC'

S1 → SCBC

S2 → Congresso da **SBC**

SimLev (CSBC, CSBC) = 1

# Acronyms

› Exemplo:

s1 = 'VLDB' e s1= 'very large database'

S1 → VLDB

S2 → **v**ery **l**arge **d**atabase

SimLev (VLDB, vld) = 0,75

# Soundex

## › Função Soundex()

- Algoritmo fonético que indexa *strings* – apenas pronúncias em Inglês
- Idéia básica
  - › A primeira letra da palavra é seguida por três números (resultantes de codificação)

# Soundex

- › Passos
  - 1. Remove todos os W e H
  - 2. Codifica
    - › B, F, P, V como 1
    - › C,G,J,K,Q,S,X,Z como 2
    - › D,T como 3
    - › L como 4
    - › M,N como 5
    - › R como 6
  - 3. Remove vogais
  - 4. Concatena primeira letra com os numerais

# Soundex

› Exemplo:

great

**G63**

e

grate

**G63**

Ignora a primeira letra

2 - Codifica consoantes

r -> 6  
t -> 3

numerais

# Soundex

› Exemplo:

~~cheap~~

e

~~sheep~~

**C1**

**S1**

Ignora a primeira letra

2 – Codifica consoantes  
 $p \rightarrow 1$

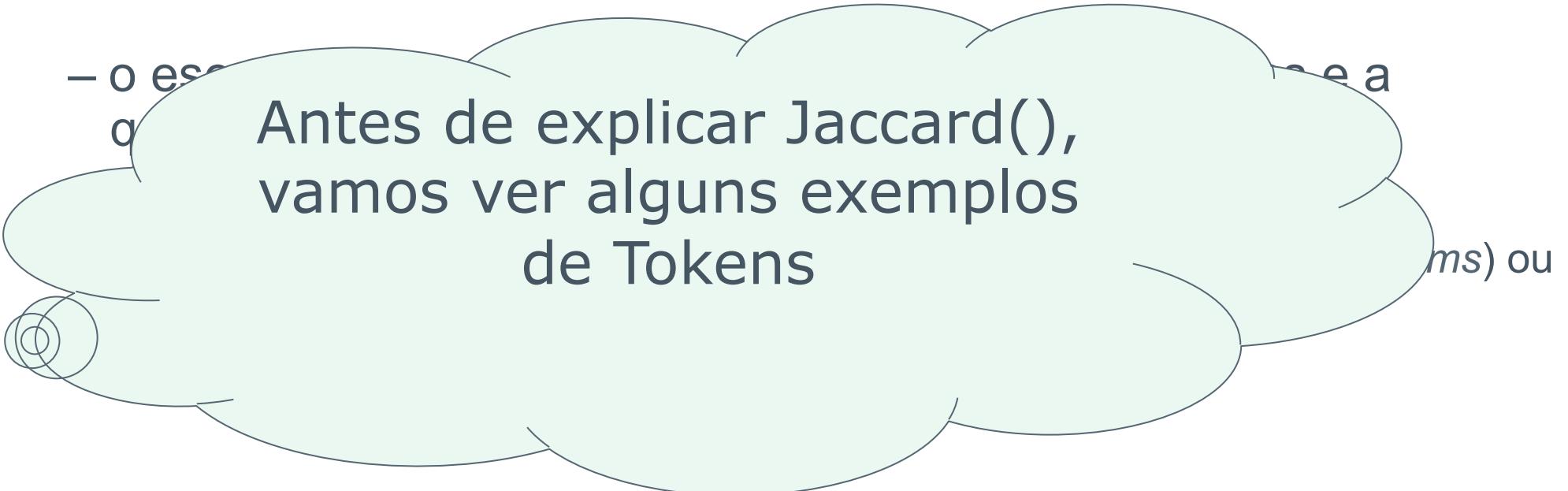
com os numerais

# Valores atômicos

- › Baseadas em *caracter*
  - Efetuam a comparação caractere a caractere
- › Baseadas em *token*
  - Efetuam a comparação *token* a *token* (*token-based*), ou seja, a função analisa cada *token* individualmente em uma *string*

# Jaccard

## › Coeficiente de *Jaccard()*



Antes de explicar *Jaccard()*,  
vamos ver alguns exemplos  
de Tokens

# Tokens

- › Sendo a frase:

“Maria gosta de ouvir música”

Tokens = {Maria, gosta, de, ouvir, música}

- Sendo a palavra:

“Maria”

Tokens = {Mar, ari, ria}

# Tokens

- ›  $T = \{\text{Mar}, \text{ari}, \text{ria}\}$ 
  - Criados através de tri-grams
- › Podem ser n-grams:
  - 2-grams  $\rightarrow T = \{\text{Ma}, \text{ar}, \text{ri}, \text{ia}\}$
  - 3-grams  $\rightarrow T = \{\text{Mar}, \text{ari}, \text{ria}\}$
  - 4-grams  $\rightarrow T = \{\text{Mari}, \text{aria}\}$
  - ***n*-grams**

# Jaccard()

- › Jaccard ( $s_1, s_2$ ) = 
$$\frac{(s_1 \cap s_2)}{(s_1 \cup s_2)}$$
- ›  $S_1$  = “Maria gosta de ouvir música” = {Maria, gosta, de, ouvir, música}
- ›  $S_2$  = “Pedro gosta de tocar música” = {Pedro, gosta, de, tocar, música}

$$\text{Jaccard } (s_1, s_2) = \frac{3}{7} \approx 0,428$$

# Jaccard()

- › Jaccard ( $s_1, s_2$ ) = 
$$\frac{(s_1 \cap s_2)}{(s_1 \cup s_2)}$$
- › S1 = “Maria Eduarda” = {MAR, ARI, RIA, EDU, DUA, UAR, ARD, RDA}
- › S1 = “Eduarda, Maria” = {EDU, DUA, UAR, ARD, RDA, MAR, ARI, RIA}

- › Jaccard ( $s_1, s_2$ ) = 
$$\frac{8}{13}$$

# Classificação

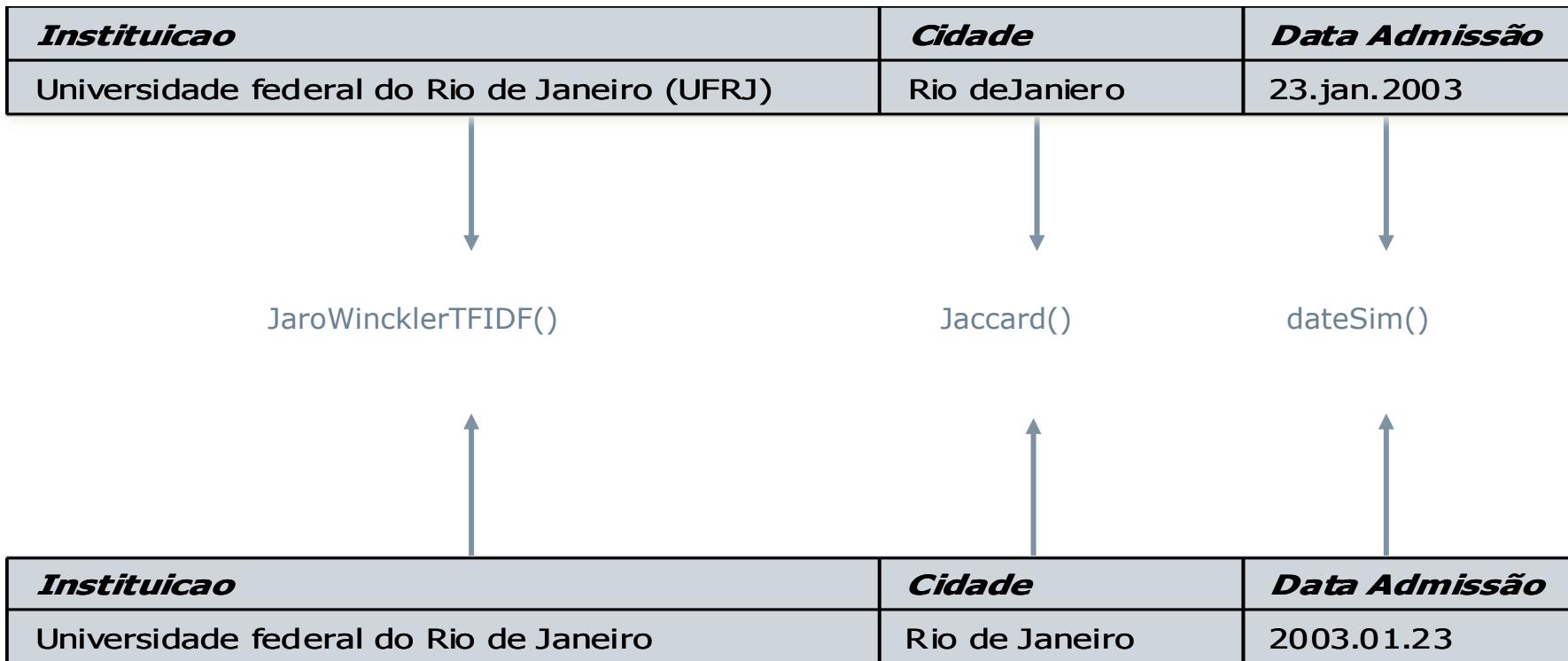
- › Valores atômicos
  - Baseadas em *Caracter*
  - Baseadas em *Token*
- › Valores agregados
  - Uso de expressões algébricas
  - Uso de algoritmos

# Valores agregados

- › Valores compostos por múltiplos campos
  - Tuplas, dados XML, registros
  - Funcionamento:
    - › Compara cada campo individualmente, depois combina

# Valores agregados

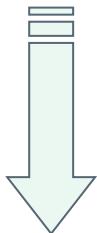
- Para um melhor resultado, cada atributo é comparado, usando uma função diferente



# Distância Euclidiana

- › Distância Euclidiana
  - Mesma função usada para imagens

$$\text{Euclidiana } (i_1, i_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

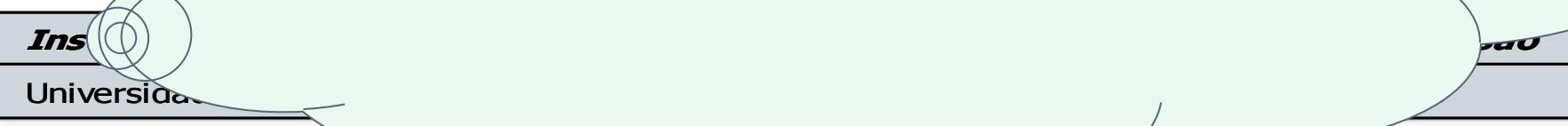


$$\text{Euclidiana } (i_1, i_2) = \sqrt{\text{sim}(o1.a1, o2.a1)^2 + \text{sim}(o1.a2, o2.a2)^2}$$

# Exemplo

Instituição	Cidade	Data Admissão
Universidade federal do Rio de Janeiro (UFRJ)	Rio de Janeiro	23.jan.2003

Não leva em conta a estrutura usada no armazenamento  
(tupla, lista, conjunto...)



$$\text{Euclidiana } (t_1, t_2) = \sqrt{(0.16667)^2 + (0.1538)^2 + (0)^2}$$

$$\text{Euclidiana } (t_1, t_2) = \sqrt{(0.02777778)^2 + (0.023669)^2 + 0}$$

$$\text{Euclidiana } (t_1, t_2) = \sqrt{0.51446417}$$

$$\text{Euclidiana } (t_1, t_2) = 0.226818026$$

$$\text{Similaridade} = 1 - 0.226818026 = \textcolor{red}{0.773181974}$$

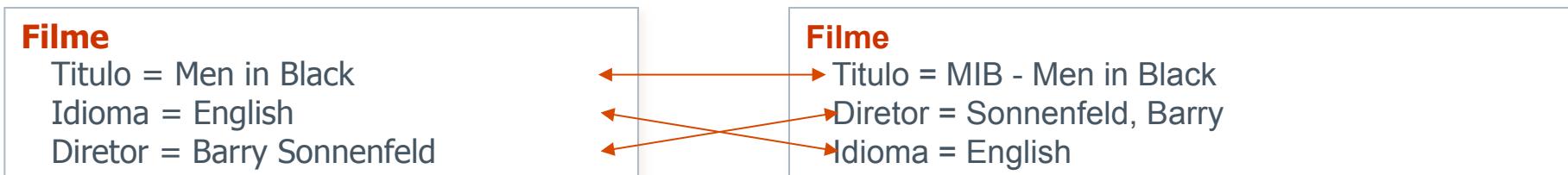
# Funções para agregados

- Funções que consideram a estrutura de armazenamento
  - Tupla
    - tupleSim()
  - Coleção
    - Lista
      - listSim()
    - Conjunto
      - setSim()

# Tuplas

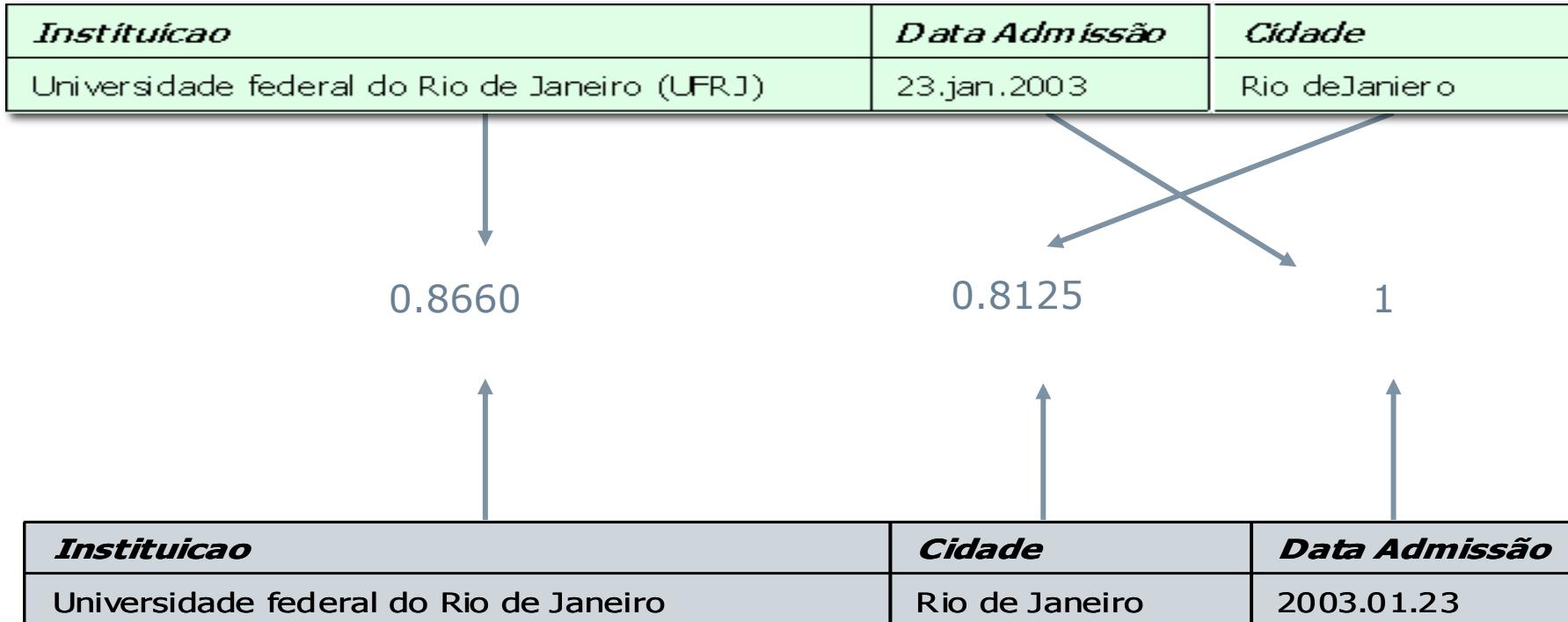
- **tupleSim()**

- Agregados compostos por atributos com diferentes campos
- Comparação dos atributos de mesmo nome.



- Média dos escores obtidos

# tupleSim()



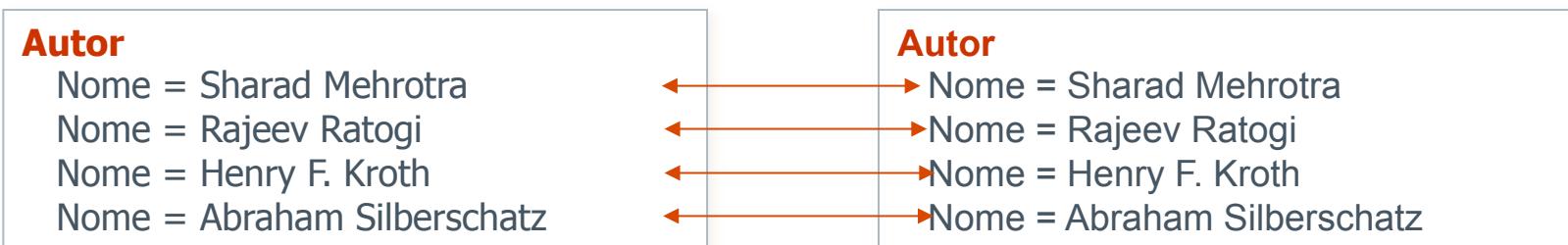
$$\text{tupleSim}(t_1, t_2) = (0.8660 + 0.8125 + 1)/3$$

$$\text{tupleSim}(t_1, t_2) = \mathbf{0.892833}$$

# Listas

- **listSim()**

- Agregados compostos por atributos com o mesmo domínio de valores
- Comparação dos atributos de mesmo nome, de mesma posição



- Média dos escores obtidos

# listSim()

<b>nome</b>	<b>nome</b>	<b>nome</b>	<b>nome</b>
Sharad Mehrotra	Rajeev Ratogi	Henry F. Kroth	Abraham Silberschatz
1	1	1	1
<b>nome</b>	<b>nome</b>	<b>nome</b>	<b>nome</b>
Mehrotra, Sharad	Ratogi, Rajeev	F. Kroth, Henry	Silberschatz, Abraham

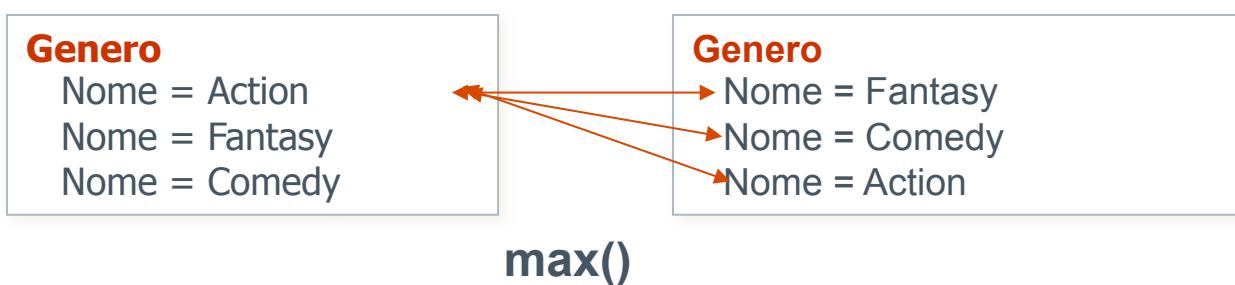
$$\text{listSim}(t1, t2) = (1 + 1 + 1 + 1)/4$$

$$\text{listSim}(t1, t2) = 1$$

# Conjunto

- **setSim()**

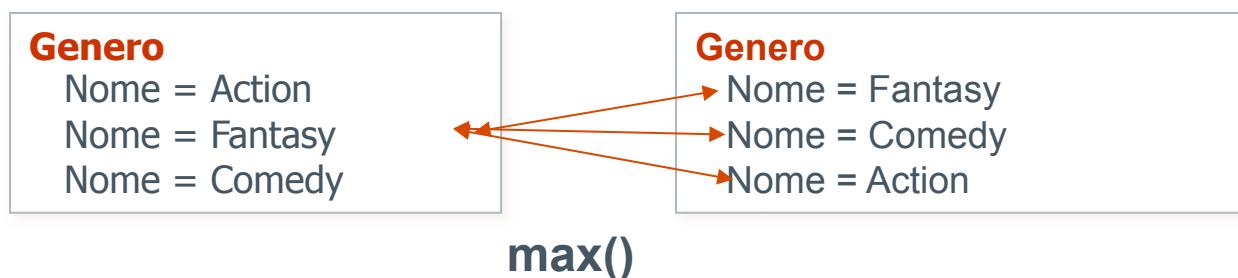
- Agregados compostos por atributos com o mesmo domínio de valores
- Comparação dos atributos de mesmo nome, independente da posição



# Conjunto

- **setSim()**

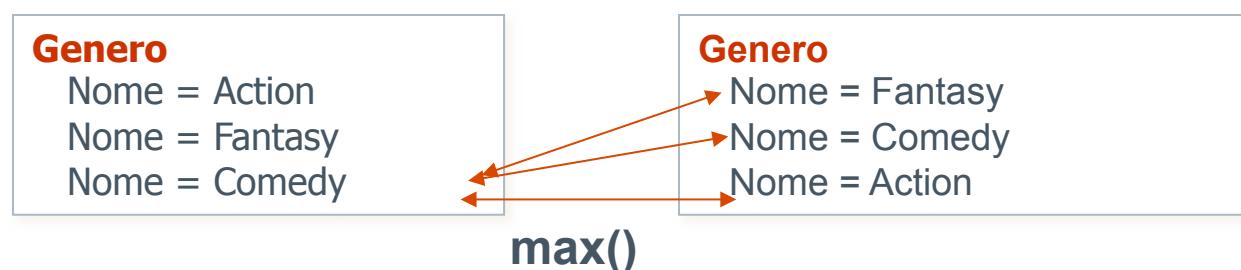
- Agregados compostos por atributos com o mesmo domínio de valores
- Comparação dos atributos de mesmo nome, independente da posição



# Conjunto

- **setSim()**

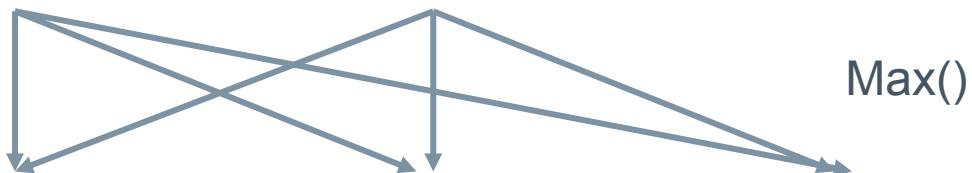
- Agregados compostos por atributos com o mesmo domínio de valores
- Comparação dos atributos de mesmo nome, independente da posição



- Média dos escores máximos obtidos

# setSim()

<b><i>gênero</i></b>	<b><i>gênero</i></b>
Comedy	Action



Max()

Max()

<b><i>gênero</i></b>	<b><i>gênero</i></b>	<b><i>gênero</i></b>
Comedy	Action	Fantasy

$$\text{listSim } (t1, t2) = (1 + 1)/3$$

$$\text{listSim } (t1, t2) = \mathbf{0.6666667}$$

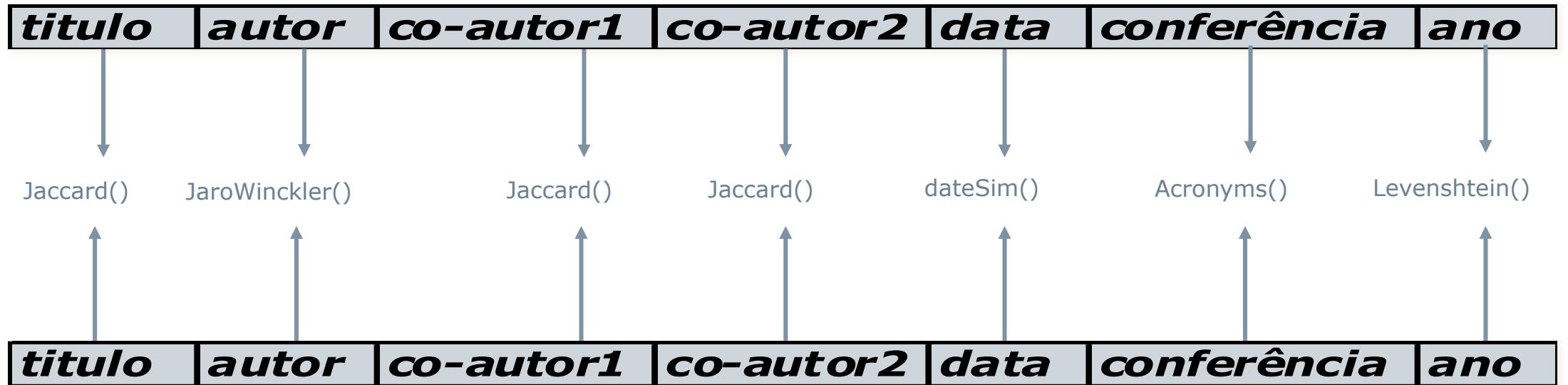
# Classificação

- › Valores atômicos
  - Baseadas em *Caracter*
  - Baseadas em *Token*
- › Valores agregados
  - Uso de expressões algébricas
  - Uso de algoritmos

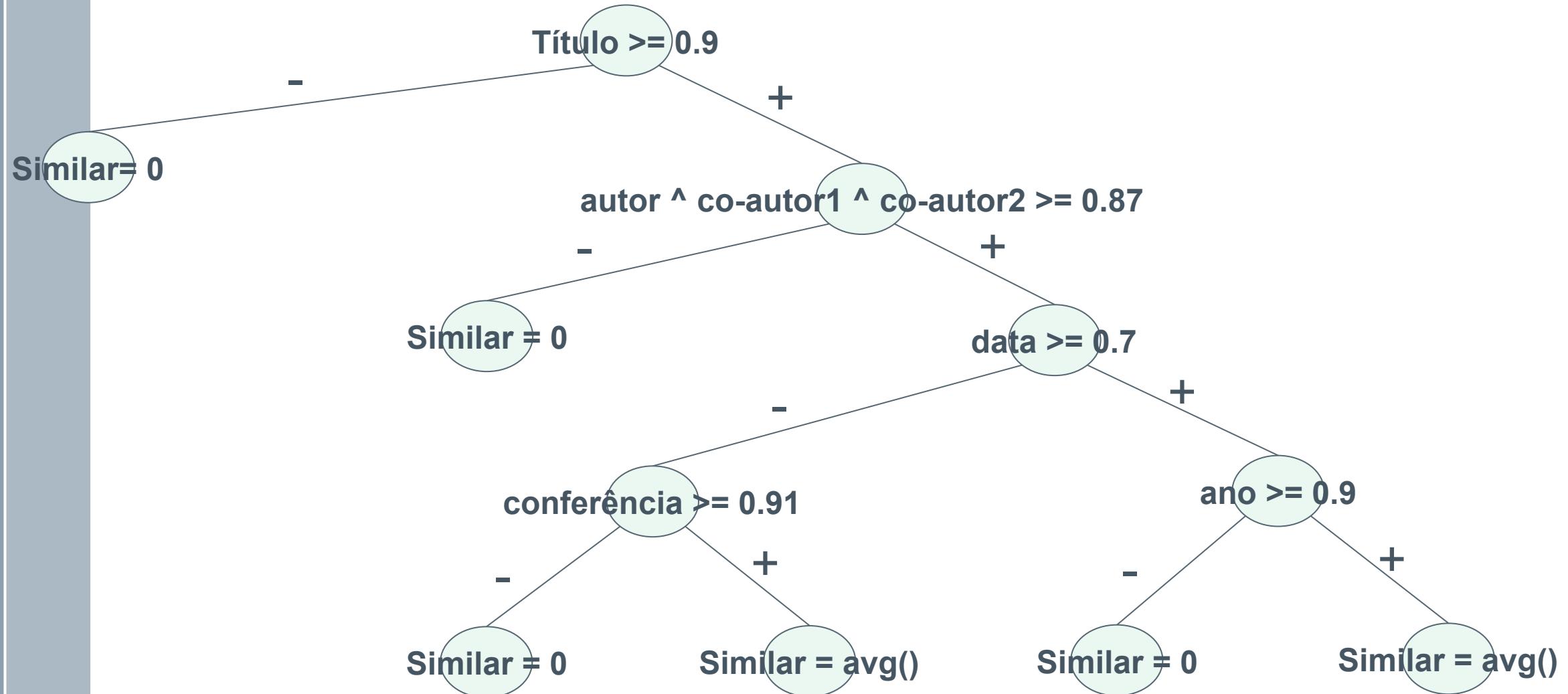
# Algoritmos

- › Grande parte dos trabalhos
  - Uso de algoritmos de *machine learning*
    - › Árvores de decisão
    - › SVM (*Support Vector Machine*)
- › Outros
  - Uso de técnicas de RI
    - › Combinação de rankings (*rank merge*)

# Usando árvores de decisão



# Árvores de decisão



# SVM

**Movie**

<b>Title</b>	<b>Director</b>	<b>Genre</b>	<b>Main Actress</b>
The Wedding Planner	Shankman, Adam	Romance	Jennifer Lopez
Wedding Planner, The	Adam Shankman	Romance	J. Lo
The Wedding Planner	A. Shankman	Romance	Lopez, Jennifer

$d1t, d2t$

$d1d, d2d$

$d1g, d2g$

$d1m, d2m$

[  $d1t, d2t ; d1d, d2d ; d1g, d2g ; d1m, d2m$  ]

SVM

Duplicadas

Não duplicadas

# Algumas Aplicações que usam Funções de Similaridade

# Roteiro

- › Consultas por similaridade
- › Integração de Dados
- › Data Cleaning
- › Mineração de Dados

# Consultas por similaridade

# Consultas

- › Como efetuar consulta sobre uma base que possui diferentes representações do mesmo objeto?
  - Exemplo usando um dialeto similar a SQL



```
SELECT artigo  
FROM BDBComp  
WHERE levenshtein(autor, 'Agma Machado Traina') > 0,75
```

Recuperar artigos do autores 'Agma Machado Traina'



# SGBDs

- › Alguns SGBDs implementam funções de similaridade
  - São usadas em consultas SQL
  - Podem ser construídas através de UDF (*User Defined Functions*)
  - Oracle
    - › Algumas funções *built-in* – podemos usar no SQL (PL/SQL)
  - DB2
    - › Não tem as funções *built-in*
    - › Criação de UDFs
  - PostgreSQL
    - › Algumas funções *built-in* – podemos usar no SQL (PgP/SQL)
    - › Criação de UDFs

# PostgreSQL

- › A partir da versão 8.0 implementa algumas funções como
  - Levenshtein
  - Soundex
  - dMetaphone (da mesma forma que o Soundex, é a implementação de um algoritmo fonético)
  - Exemplo de uma consulta possível – **executar no SGBD**

```
SELECT valor, soundex(valor)
FROM palavra
WHERE soundex(valor) = soundex('great')
```

# PostgreSQL

## › Consultas

```
SELECT valor, soundex(valor),  
       levenshtein(soundex(valor), soundex('sheep'))  
FROM palavra  
WHERE levenshtein(soundex(valor), soundex('sheep'))>=0
```

```
SELECT valor, soundex(valor), levenshtein(soundex(valor), soundex('sheep'))  
FROM palavra  
WHERE levenshtein(soundex(valor), soundex('sheep'))>=1
```

```
SELECT valor, dmetaphone(valor)  
FROM palavra  
WHERE dmetaphone(valor) = dmetaphone('great')
```

```
SELECT valor, dmetaphone(valor), soundex(valor)  
FROM palavra  
WHERE dmetaphone(valor) = dmetaphone('great')
```

# PostgreSQL

## › Consultas

```
SELECT *  
FROM cidade
```

```
SELECT *  
FROM cidade  
WHERE levenshtein(nome, 'Porto Alegre') < 5
```

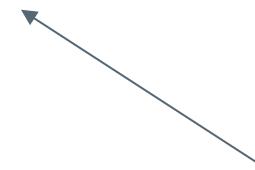
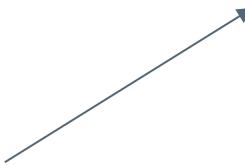
```
SELECT nome, levenshtein(nome, 'Porto Alegre')  
FROM cidade  
WHERE levenshtein(nome, 'Porto Alegre') < 5  
ORDER BY nome
```

# Integração de Dados

# Integração de Dados

› Como integrar dados que são escritos de diferentes formas?

Levenshtein ("League of ... Action", "League of ... Sci-Fi")  $\geq 0.78$



Blockbuster

League of Extraordinary Gentlemen,  
**Stephen Norrington**, English (British), **Action**.

IMDb

League of Extraordinary Gentlemen, The.  
**Norrington, Stephen**. British English  
(**Action, Fantasy, Sci-Fi**)

# Integração

- › Sistemas usam
  - Produto Cartesiano
  - Junção
- Tradicionalmente
  - Ambos são implementados usando **operador de igualdade**
- Em integração por similaridade
  - Operador de igualdade é substituído por uma **função de similaridade**

# Integração

- › Fazendo uma analogia com o que já conhecemos...

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a1	Gol GTI	VW	Prata
a1	Gol GTI	VW	Prata
a1	Gol GTI	VW	Prata

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Mareá	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Mareá	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat				
a2	Pálio ELX	Fiat				
a2	Pálio ELX	Fiat				
a2	Pálio ELX	Fiat				

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat	Branco	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a2	Pálio ELX	Fiat	Branco	a3	c1	01/02/2000
a2	Pálio ELX	Fiat	Branco	a2	c2	03/01/2000

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat	Branco	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a2	Pálio ELX	Fiat	Branco	a3	c1	01/02/2000
a2	Pálio ELX	Fiat	Branco	a2	c2	03/01/2000
a3	Corsa Wind	GM				
a3	Corsa Wind	GM				
a3	Corsa Wind	GM				
a3	Corsa Wind	GM				

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat	Branco	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a2	Pálio ELX	Fiat	Branco	a3	c1	01/02/2000
a2	Pálio ELX	Fiat	Branco	a2	c2	03/01/2000
a3	Corsa Wind	GM	Preto	a1	c1	01/01/2000
a3	Corsa Wind	GM	Preto	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a3	Corsa Wind	GM	Preto	a2	c2	03/01/2000

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat	Branco	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a2	Pálio ELX	Fiat	Branco	a3	c1	01/02/2000
a2	Pálio ELX	Fiat	Branco	a2	c2	03/01/2000
a3	Corsa Wind	GM	Preto	a1	c1	01/01/2000
a3	Corsa Wind	GM	Preto	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a3	Corsa Wind	GM	Preto	a2	c2	03/01/2000
a4	Marea	Fiat				
a4	Marea	Fiat				
a4	Marea	Fiat				
a4	Marea	Fiat				

# Integração

› Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat	Branco	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a2	Pálio ELX	Fiat	Branco	a3	c1	01/02/2000
a2	Pálio ELX	Fiat	Branco	a2	c2	03/01/2000
a3	Corsa Wind	GM	Preto	a1	c1	01/01/2000
a3	Corsa Wind	GM	Preto	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a3	Corsa Wind	GM	Preto	a2	c2	03/01/2000
a4	Marea	Fiat	Preto	a1	c1	01/01/2000
a4	Marea	Fiat	Preto	a2	c3	05/01/2000
a4	Marea	Fiat	Preto	a3	c1	01/02/2000
a4	Marea	Fiat	Preto	a2	c2	03/01/2000

# Integração

- Em BD relacional, uma consulta SQL com produto cartesiano é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a4	c2	03/01/2000

Sobre este resultado,  
as linhas indesejadas podem ser  
eliminadas usando o operador de  
igualdade:  
**carro.codCarro = reserva.codCarro**

# Integração

- Em SGBD, uma consulta SQL com produto cartesiano tradicional é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat	Branco	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a2	Pálio ELX	Fiat	Branco	a3	c1	01/02/2000
a2	Pálio ELX	Fiat	Branco	a2	c2	03/01/2000
a3	Corsa Wind	GM	Preto	a1	c1	01/01/2000
a3	Corsa Wind	GM	Preto	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a3	Corsa Wind	GM	Preto	a2	c2	03/01/2000
a4	Marea	Fiat	Preto	a1	c1	01/01/2000
a4	Marea	Fiat	Preto	a2	c3	05/01/2000
a4	Marea	Fiat	Preto	a3	c1	01/02/2000
a4	Marea	Fiat	Preto	a2	c2	03/01/2000

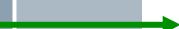
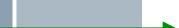
# Integração

- Em SGBD, uma consulta SQL com produto cartesiano tradicional é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a2	c2	03/01/2000

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a1	Gol GTI	VW	Prata	a2	c3	05/01/2000
a1	Gol GTI	VW	Prata	a3	c1	01/02/2000
a1	Gol GTI	VW	Prata	a2	c2	03/01/2000
a2	Pálio ELX	Fiat	Branco	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a2	Pálio ELX	Fiat	Branco	a3	c1	01/02/2000
a2	Pálio ELX	Fiat	Branco	a2	c2	03/01/2000
a3	Corsa Wind	GM	Preto	a1	c1	01/01/2000
a3	Corsa Wind	GM	Preto	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a3	Corsa Wind	GM	Preto	a2	c2	03/01/2000
a4	Marea	Fiat	Preto	a1	c1	01/01/2000
a4	Marea	Fiat	Preto	a2	c3	05/01/2000
a4	Marea	Fiat	Preto	a3	c1	01/02/2000
a4	Marea	Fiat	Preto	a2	c2	03/01/2000



# Integração

- Em SGBD, uma consulta SQL com produto cartesiano tradicional é executada assim:

CodCarro	Modelo	Marca	Cor
a1	Gol GTI	VW	Prata
a2	Pálio ELX	Fiat	Branco
a3	Corsa Wind	GM	Preto
a4	Marea	Fiat	Preto

CodCarro	CodCli	Data
a1	c1	01/01/2000
a2	c3	05/01/2000
a3	c1	01/02/2000
a4	c1	03/01/2000

Em um sistema de integração, o adm. deve fornecer uma função de similaridade e um ponto de corte (o threshold):

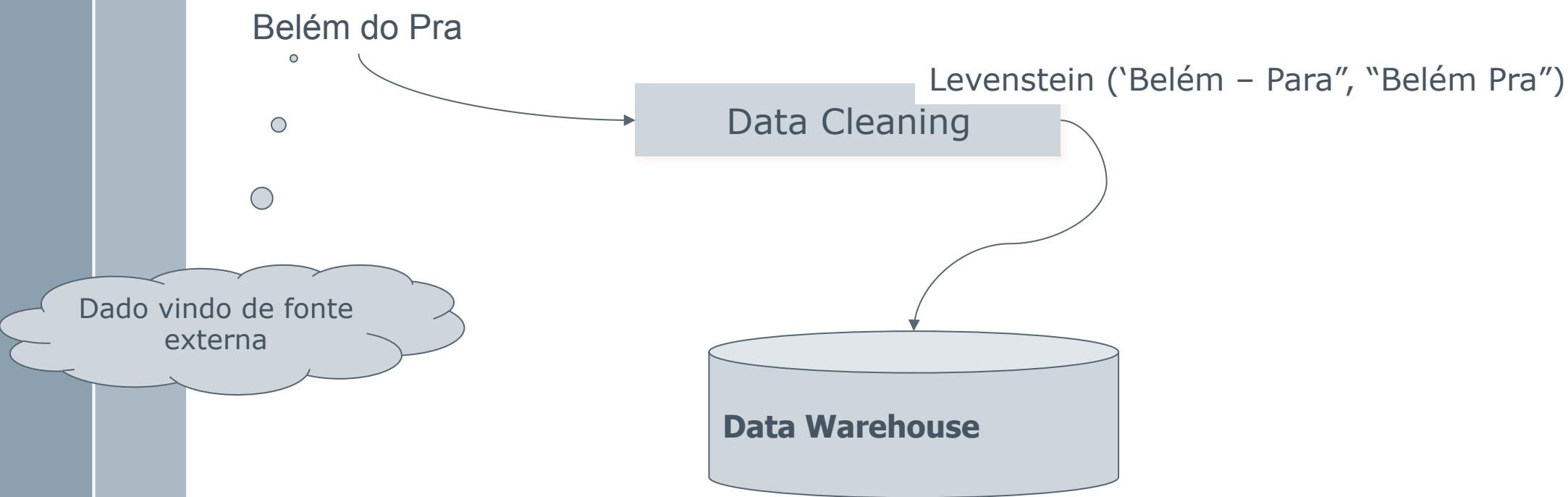
$\text{Levenshtein}(\text{carro.codCarro}, \text{reserva.codCarro}) > 0.7$

CodCarro	Modelo	Marca	Cor	CodCarro	CodCli	Data
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a4	Marea	Fiat	Preto	a4	c1	03/01/2000
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a4	Marea	Fiat	Preto	a4	c1	03/01/2000
a1	Gol GTI	VW	Prata	a1	c1	01/01/2000
a2	Pálio ELX	Fiat	Branco	a2	c3	05/01/2000
a3	Corsa Wind	GM	Preto	a3	c1	01/02/2000
a4	Marea	Fiat	Preto	a4	c2	03/01/2000

# Data Cleaning

# Data Cleaning

- › Dados vindos de fontes externas podem conter inconsistências



# Spider

› Demo: <http://queens.db.toronto.edu/project/spider/demo>

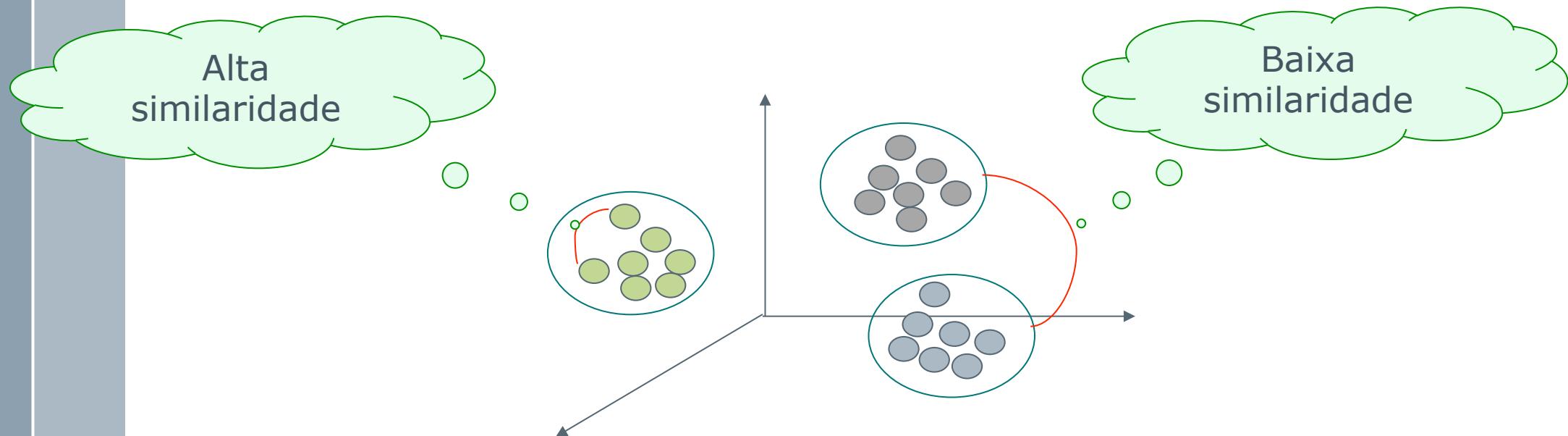
# Mineração de Dados

# Várias técnicas

- › Classificação
- › Agrupamento
- › Descoberta de Regras de Associação
- › Descoberta de Padrões Seqüenciais
- › Regressão
- › Detecção de Desvio

# Agrupamento

- › Agrupar linhas das tabelas de acordo com a similaridade existente entre elas

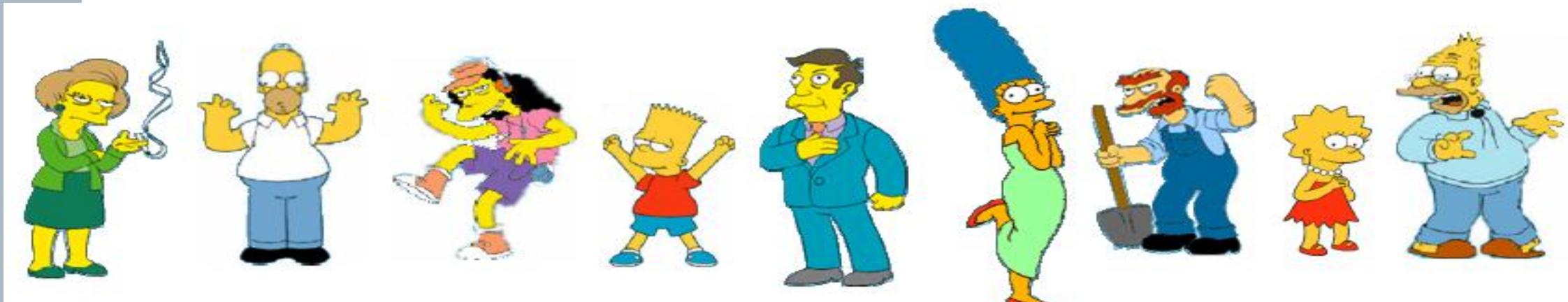


# Matriz de distância/similaridade

- › Matriz de Distância
  - Armazena uma coleção de medidas de distância (diferença) entre dois objetos quaisquer
    - › Quanto mais alto o valor, mais distantes (diferentes) eles são
  - **Valor 0 significa que são o mesmo objeto**
- › Matriz de Similaridade
  - Armazena uma coleção de medidas de similaridade entre dois objetos quaisquer
    - › Quanto mais baixo o valor, mais semelhantes eles são
  - **Valor 1 significa que são o mesmo objeto**

# Exemplo de Agrupamento de dados

- › Encontrar grupos de clientes similares



# Matriz de distância



<b>0</b>	<b>0.8</b>	<b>0.8</b>	<b>0.7</b>	<b>0.7</b>
	<b>0</b>	<b>0.2</b>	<b>0.4</b>	<b>0.4</b>
		<b>0</b>	<b>0.3</b>	<b>0.3</b>
			<b>0</b>	<b>0.1</b>
				<b>0</b>


$$D(\text{Marge}, \text{Bart}) = 0$$


$$D(\text{Marge}, \text{Bart}) = 0.8$$


$$D(\text{Marge}, \text{Marge}) = 0.1$$

# Matriz de similaridade



<b>1</b>	<b>0.2</b>	<b>0.2</b>	<b>0.3</b>	<b>0.3</b>
	<b>1</b>	<b>0.8</b>	<b>0.6</b>	<b>0.6</b>
		<b>1</b>	<b>0.7</b>	<b>0.7</b>
			<b>1</b>	<b>0.9</b>
				<b>1</b>

$$1 - D(\text{Marge}, \text{Marge}) = 0$$

$$1 - D(\text{Marge}, \text{Lisa}) = 0.2$$

$$1 - D(\text{Marge}, \text{Edna}) = 0.9$$

# Como calcular a distância

- › Uma alternativa:
  - Através de transformações



Quantas transformações são necessárias para transformar o Bart na Lisa?



Cliente

<b>nome</b>	<b>Tipo Roupa</b>	<b>Cor olhos</b>	<b>Formato cabelo</b>	<b>Tipo sapato</b>	<b>cor sapato</b>
Bart	camiseta/bermuda	castanho	ralo	tenis	azul
Lisa	vestido	castanho	curto	sandália	vermelho

# Como calcular a distância

- › Uma alternativa:
  - Através de transformações



Quantas transformações são necessárias para transformar o Bart na Lisa?



- Trocar tipo roupa – 1 ponto**
- Trocar cor dos olhos – 0 ponto**
- Trocar formato do cabelo – 1 ponto**
- Trocar tipo do sapato – 1 ponto**
- Trocar cor do sapato – 1 ponto**

Soma das transformações (4), dividido pelo total de características (5) = 0.8

**Similaridade = 1 - 0.8: 0.2**

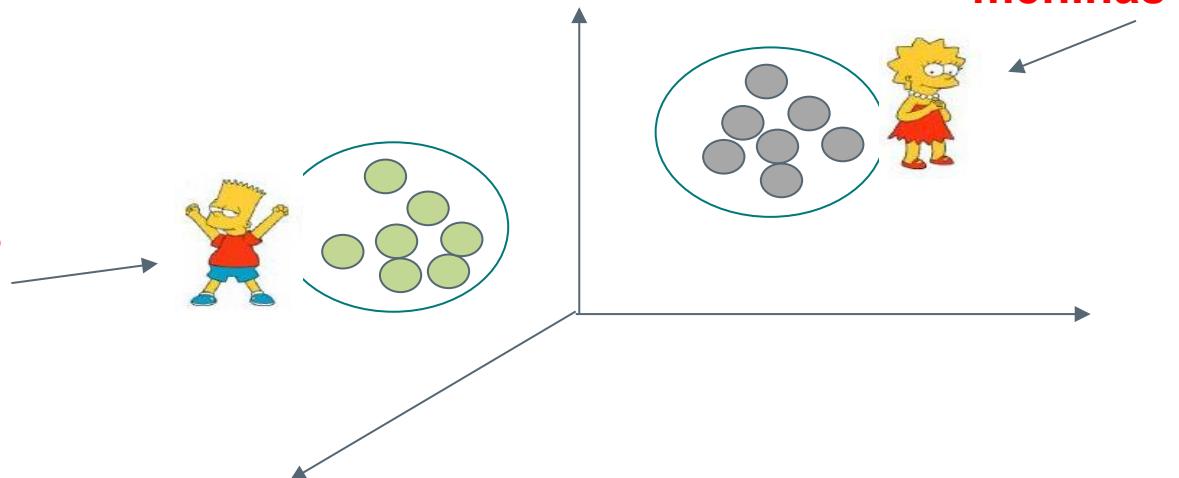
# Transformações

› Cada transformação equivale a um atributo na tabela:

- Tipo de roupa
- Cor dos olhos
- Formato do cabelo
- Tipo de sapato
- Cor do sapato

meninos

Usando estes atributos de transformação, Bart e Lisa não pertencem ao mesmo grupo



# Mas...

› Usando outros critérios de agrupamento



Quantas transformações são necessárias para transformar o Bart na Lisa?



Cliente

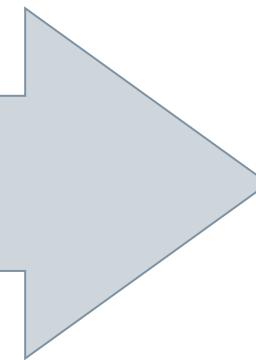
<b>nome</b>	<b>idade escolar</b>	<b>poder aquisitivo</b>	<b>nivel dep. Pais</b>	<b>ativ. Trabalho</b>	<b>ativ. Entretenimento</b>
Bart	primaria	médio	alto	nenhuma	brincar
Lisa	primaria	médio	alto	nenhuma	brincar

# Mas...

- › Usando outros critérios de agrupamento



Quantas transformações são necessárias para transformar o Bart na Lisa?



**Trocar idade escolar – 0 ponto**

**Trocar poder aquisitivo – 0 ponto**

**Trocar nível de dependência dos pais – 0 ponto**

**Trocar atividade de trabalho – 0 ponto**

**Trocar atividade de entretenimento – 0 ponto**

# Mas...

## › Usando outros critérios de agrupamento



Soma das transformações (0), dividido pelo total de características (5) = 0

$$\text{Similaridade} = 1 - 0 : 1$$



**Trocá idade escolar – 0 ponto**

**Trocá poder aquisitivo – 0 ponto**

**Trocá nível de dependência dos pais – 0 ponto**

**Trocá atividade de trabalho – 0 ponto**

**Trocá atividade de entretenimento – 0 ponto**

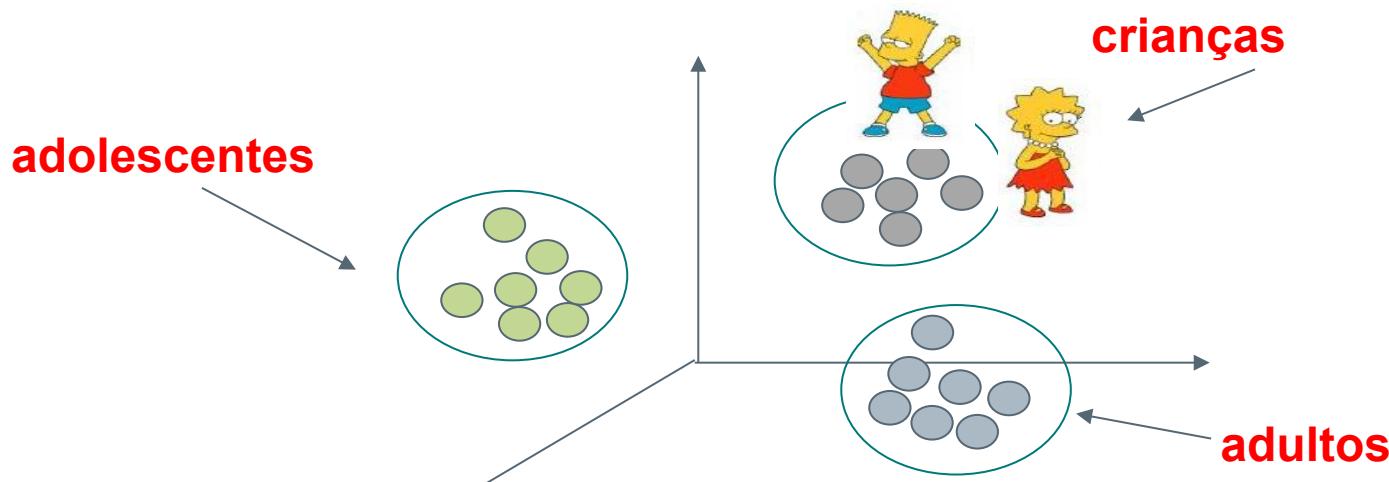
# Mas...

- › Usando outros critérios de agrupamento



Soma das transformações (0), dividido pelo número de transformações (5) = 0

$$\text{Similaridade} = 1 - 0 : 1$$



# Aplicativo

- › WEKA
- › Software Livre
  - <http://www.cs.waikato.ac.nz/ml/weka/>
  - Versão 3.4.1
- › Ferramenta de mineração
  - Vários algoritmos para técnicas mais conhecidas
  - Extensível
  - Certo apoio ao pré-processamento e exploração de dados
- › Algoritmos podem ser usados para compor outras aplicações
- › É necessário alimentar a ferramenta com dados