



SISTEMAS DE INFORMAÇÃO

ENTERPRISE ANALYTICS AND DATA
WAREHOUSING

PROFº FABIANO J. CURY MARQUES

ENTERPRISE ANALYTICS AND DATA WAREHOUSING

BIG DATA E NOSQL



AGENDA

- X Quem é o profissional de Big Data?
- X Conteúdo e Ferramentas
- X Introdução a NoSQL
- X Instalação da máquina virtual vagrant
- X Referências



QUEM TRABALHA EM BIG DATA?

Qual deve é o perfil do profissional de Big Data?



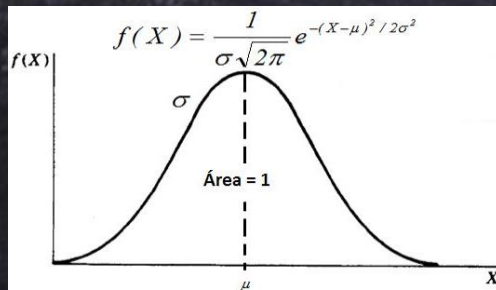
IN DATA WE TRUST!



... In God We Trust,

All others must bring data.

William Edwards Deming





“Sem dados, você é só
mais uma pessoa com
uma opinião.”

- W. EDWARDS DEMING

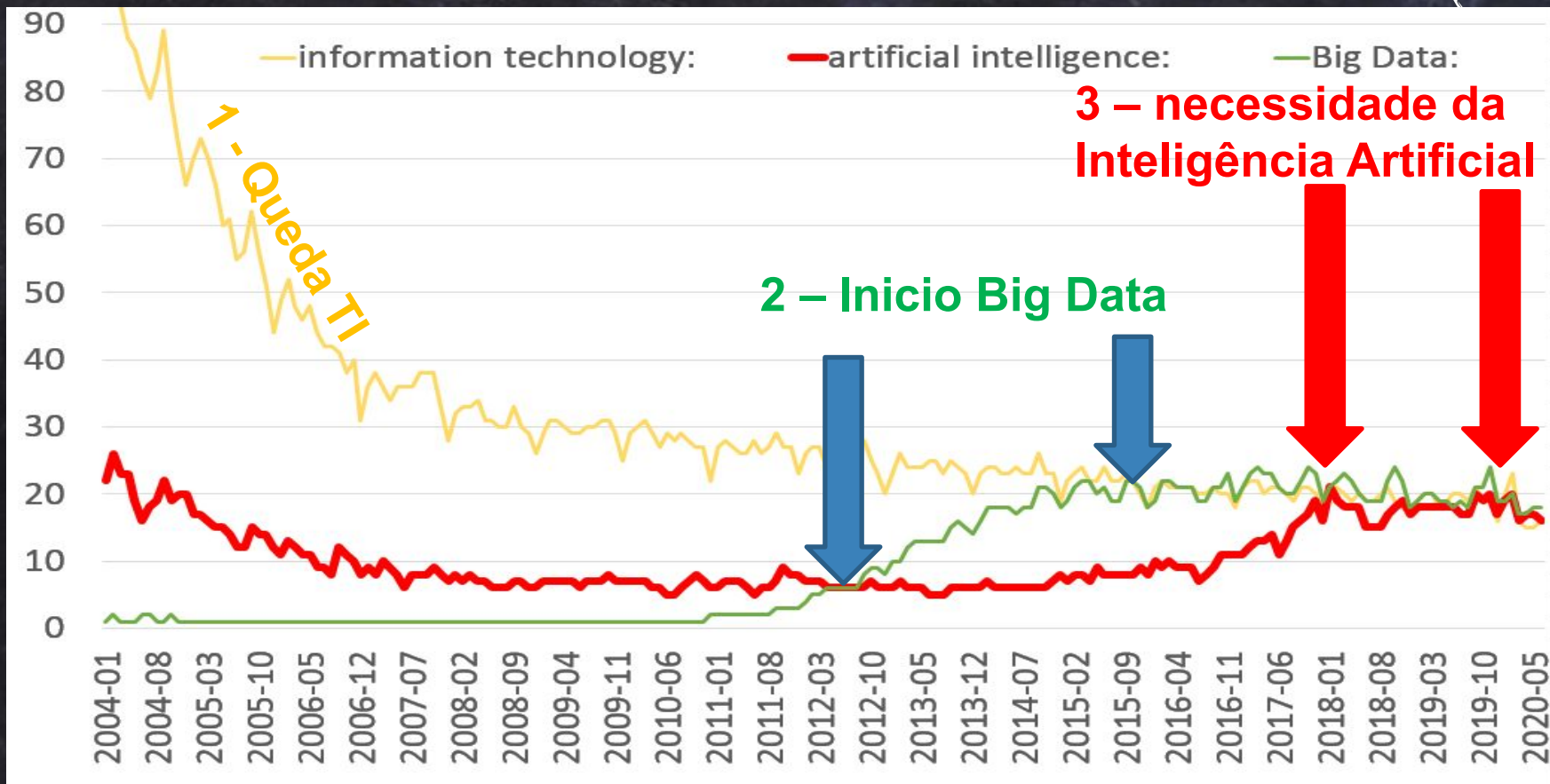
estatístico, palestrante e consultor estadunidense

RAFAELPITON.COM



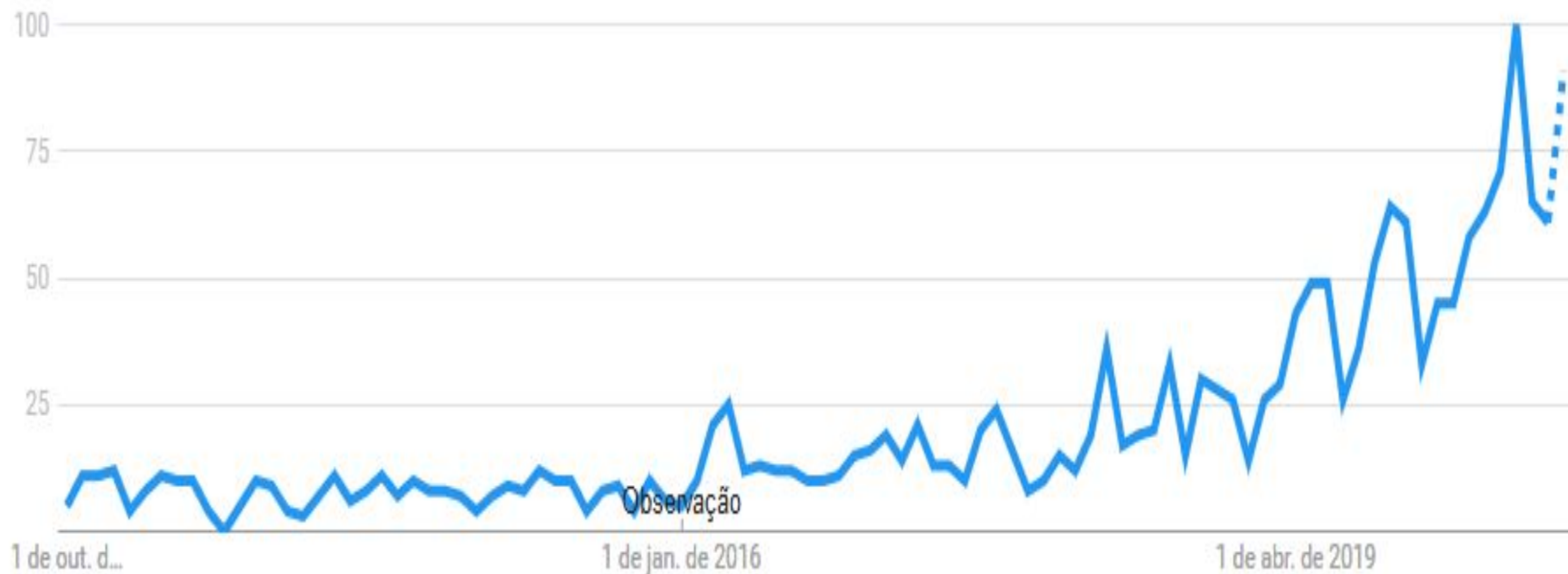
... O QUE
DEVEMOS SABER?

X Busca Big Data e IA no Google Trends:

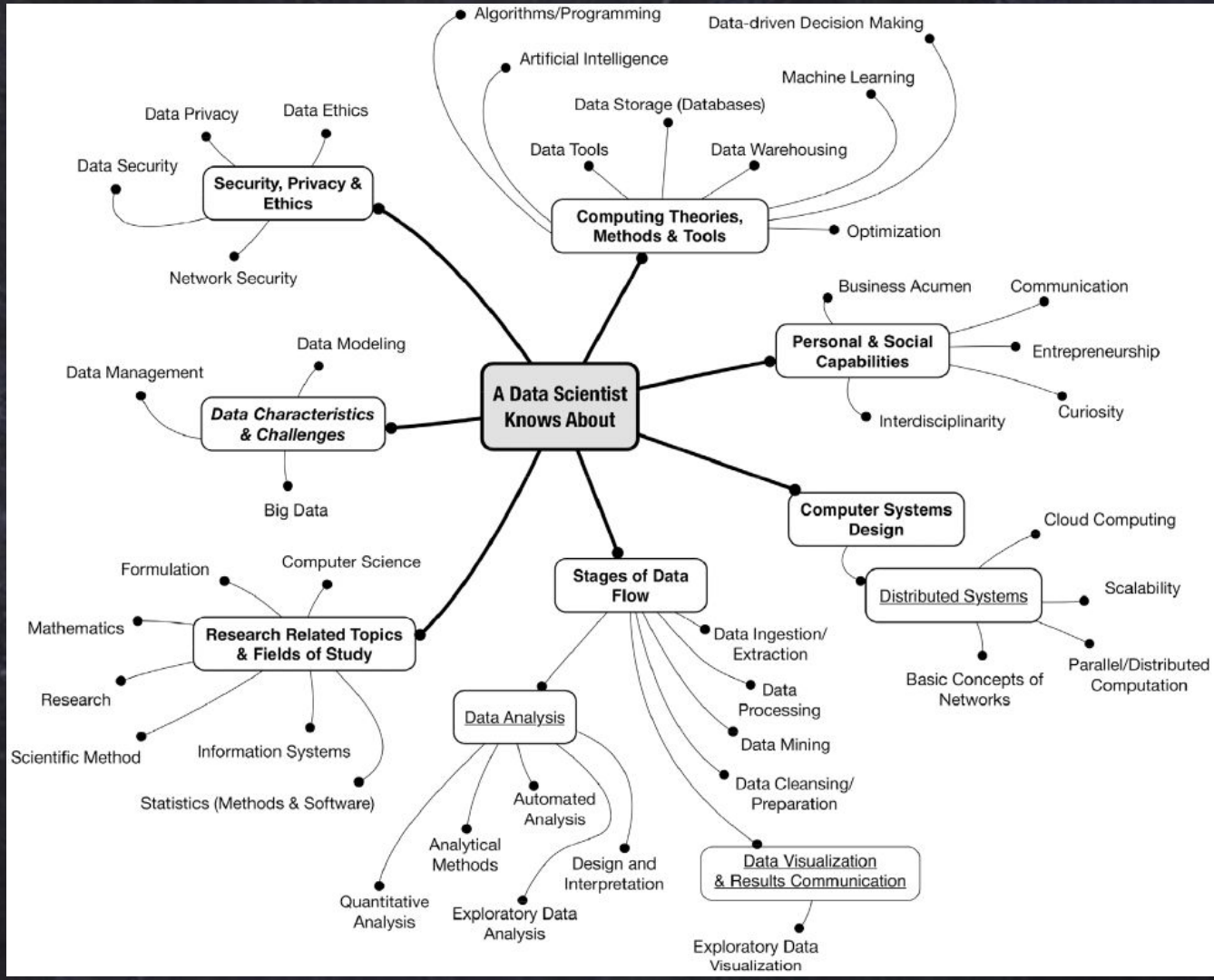




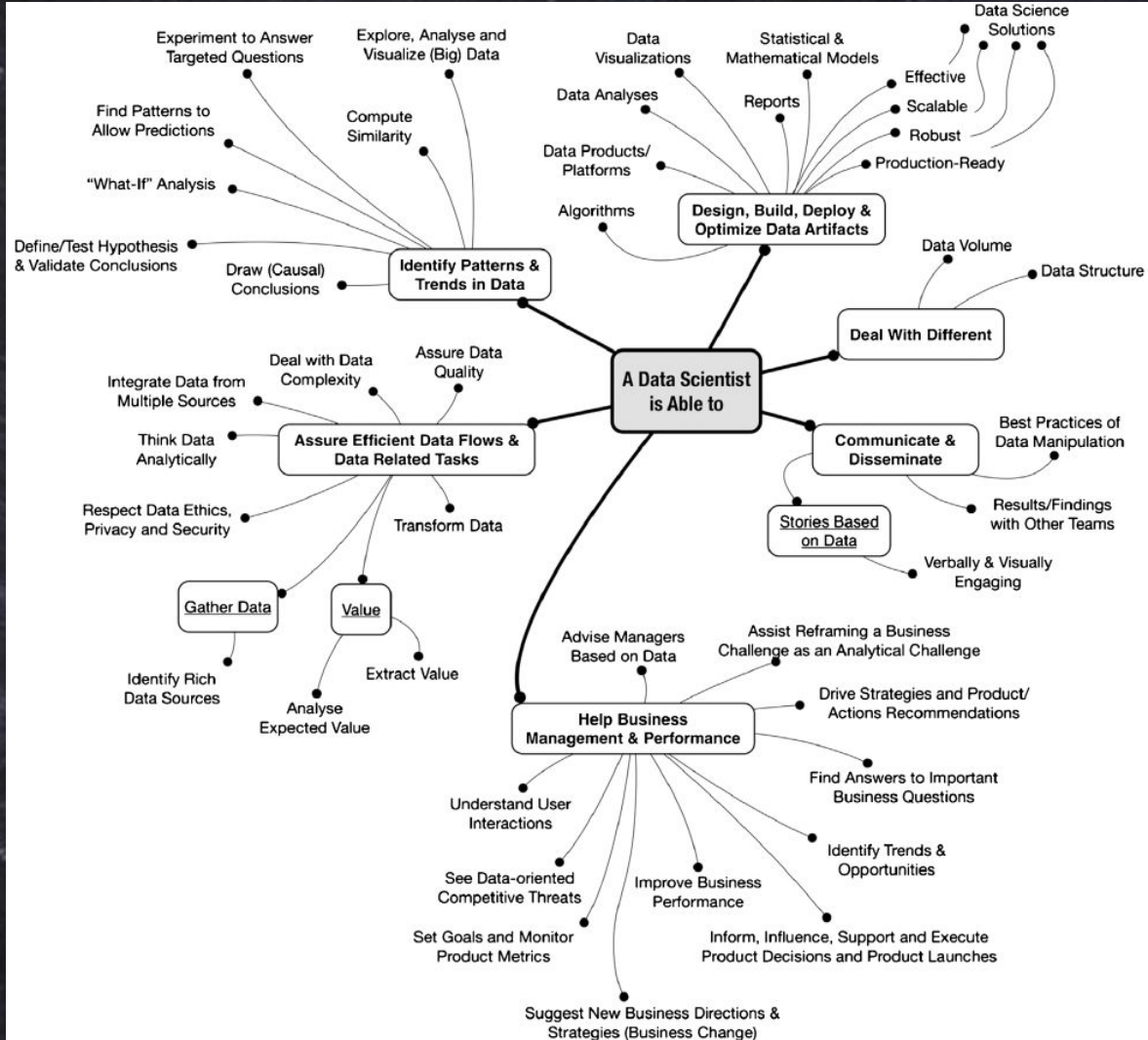
Popularidade de **Ciência de Dados** no Google Trends (2012 a 2020):



Conhecimentos necessários para um bom Cientista de dados



Habilidades necessárias para um bom Cientista de dados



MERCADO



- ✗ Entidades Lista anual de empregos emergentes do LinkedIn:
 1. Especialista em Inteligência Artificial
 2. Engenheiro de Robótica
 3. Cientista de Dados
 4. Engenheiro Full Stack
 5. Engenheiro de Confiabilidade do Local
 - ...
- ✗ Segundo o site de empregos indeed:
- ✗ Engenheiro de IA, em São Francisco, recebem entre **R\$ 497mil e R\$600mil**.
- ✗ Engenheiro de aprendizado de máquina é o melhor emprego atualmente.
- ✗ **Aumento de 344%** nas ofertas de emprego e salário base na casa **de R\$600 mil**

VIVEMOS UM MOMENTO DE MUDANÇAS!



X Em qual modelo de negócio você investiria?

Modelo de negócio

Gravadora de músicas

Locadora de vídeos

Agencias de turismo

Enciclopédias

Rede de hotéis

Operadora de telefonia

Televisão

Transporte urbano

Concorrência:

Spotify

Netflix

Booking

Google

Trivago

Whatsapp

Mídias sociais, youtube

Uber

VIVEMOS UM MOMENTO DE MUDANÇAS!



Modelo de negócio

classificado de jornais

câmeras amadoras

Locadora de veículos

montadora veículos

Correio

Fabricar GPS

Bancos

Fabricar pendrive

Balada

Concorrência:

OLX

SmartPhone

ZipCar

Tesla

Email

Waze

Fintecs

Nuvem

Tinder

... Quanto tempo vai durar o seu emprego? ... ou a empresa onde você trabalha ...



INTRODUÇÃO À NOSQL

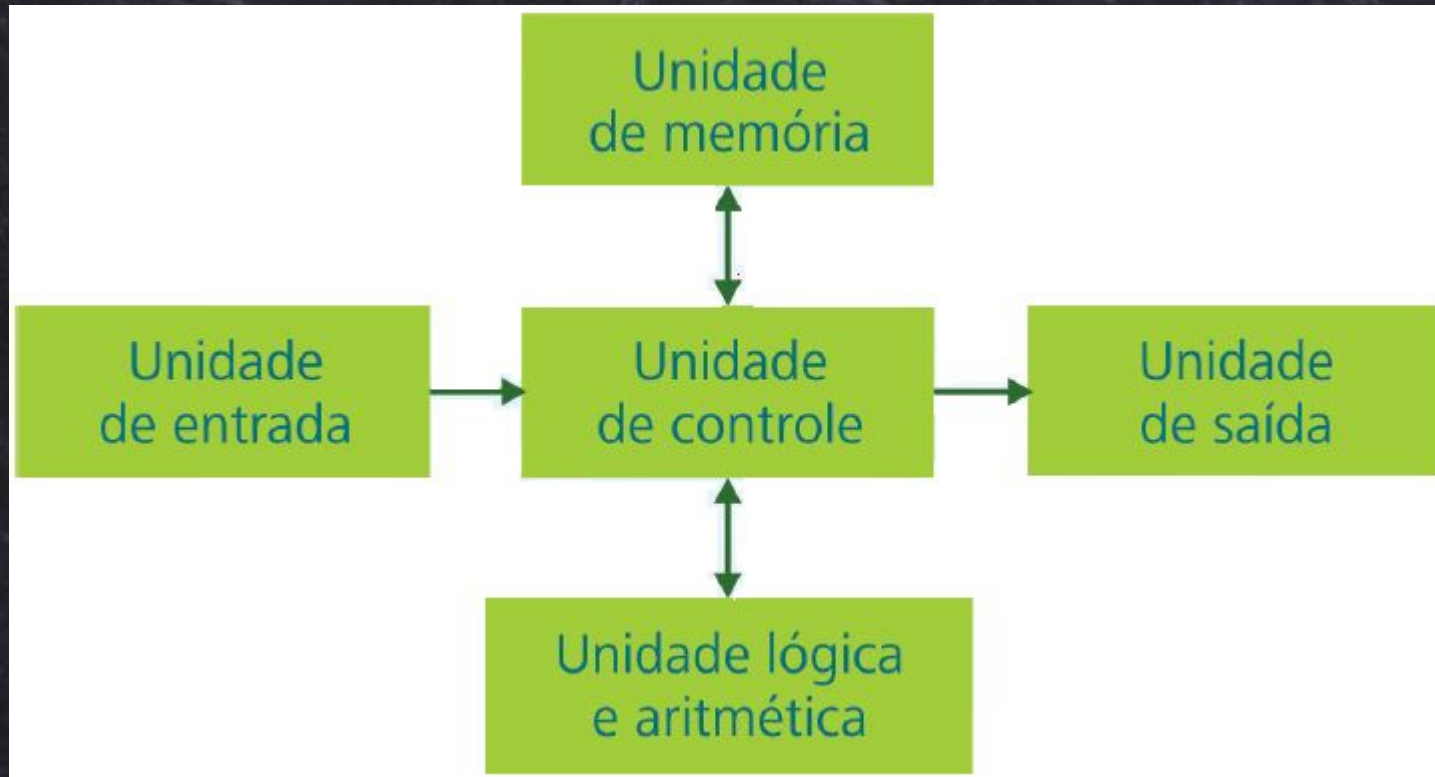
Introdução à NoSQL (Not Only SQL)

...ENTENDENDO A
NECESSIDADE
LÓGICA A
PARTIR DA
REALIDADE
FÍSICA...

ARQUITETURA DE COMPUTADORES



- ✗ Unidade de memória Principal (Volátil) separada do dispositivo de memória secundário (Não Volátil):



ARQUITETURA DE COMPUTADORES



Capacidade Velocidade Custo

Baixa

Alta

Alto

CPU



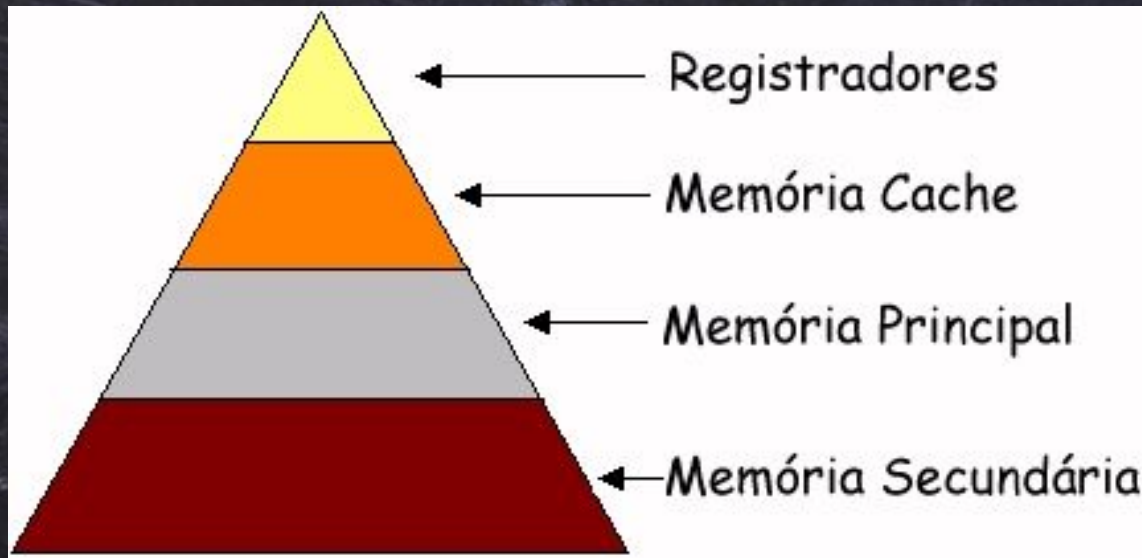
Alta



Baixa



Baixo



BANCO DE DADOS RELACIONAL



- ✗ Entidades representadas na forma de **tabela**;
- ✗ Cada tabela tem uma **chave primaria**;
- ✗ Duas ou mais tabelas podem estar **relacionadas**;
- ✗ O relacionamento se dá pela definição de um campo, denominado **chave estrangeira**, em uma das tabelas para receber o conteúdo da **chave primaria** de outra;
- ✗ Todos os dados são armazenados no formato de **registros** nas tabelas;

BANCO DE DADOS RELACIONAL

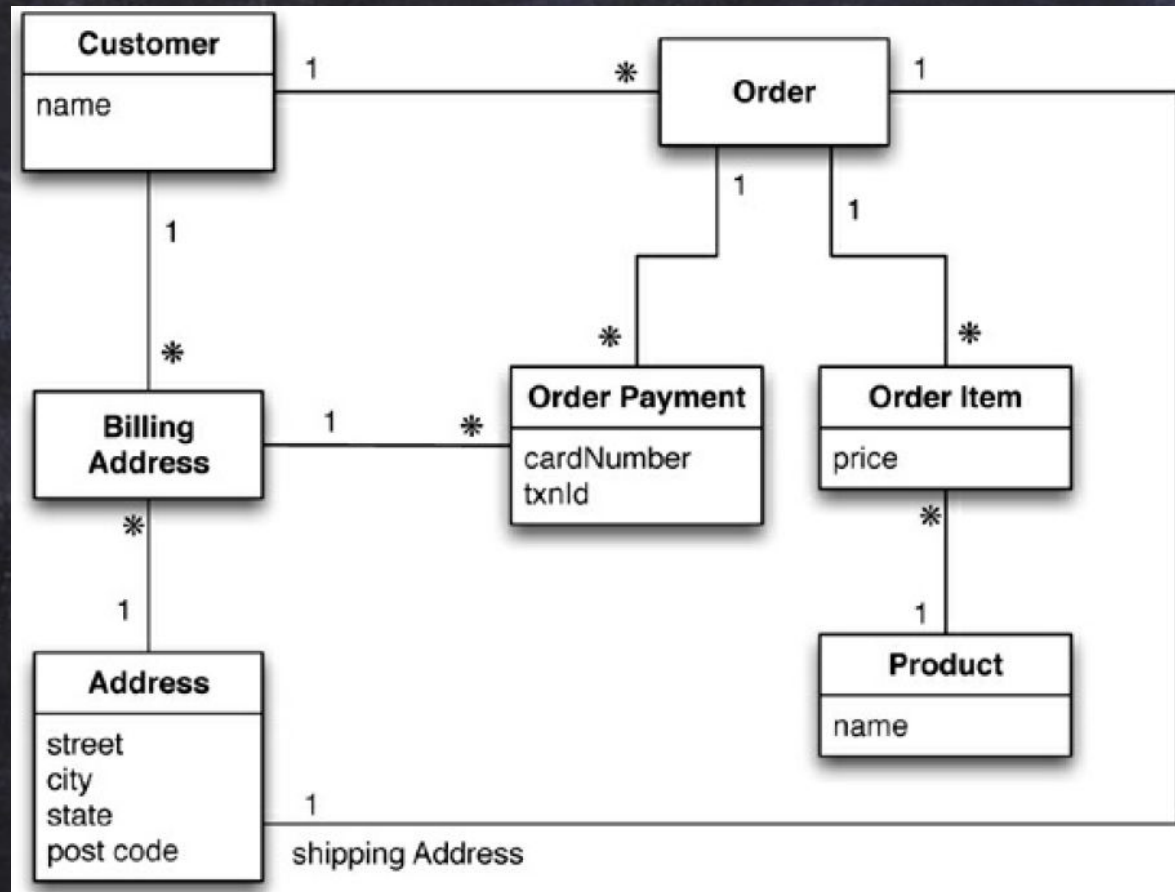


- ✗ O conceito de **cardinalidade** está associado à multiplicidade de registros associados mutuamente entre tabelas;
- ✗ A **eliminação de redundância** no banco de dados é feita por meio de um procedimento denominado **normalização**;
- ✗ Existem diferentes etapas de normalização do banco de dados;
- ✗ Cada etapa de normalização é denominada “**forma normal**”;

BANCO DE DADOS RELACIONAL



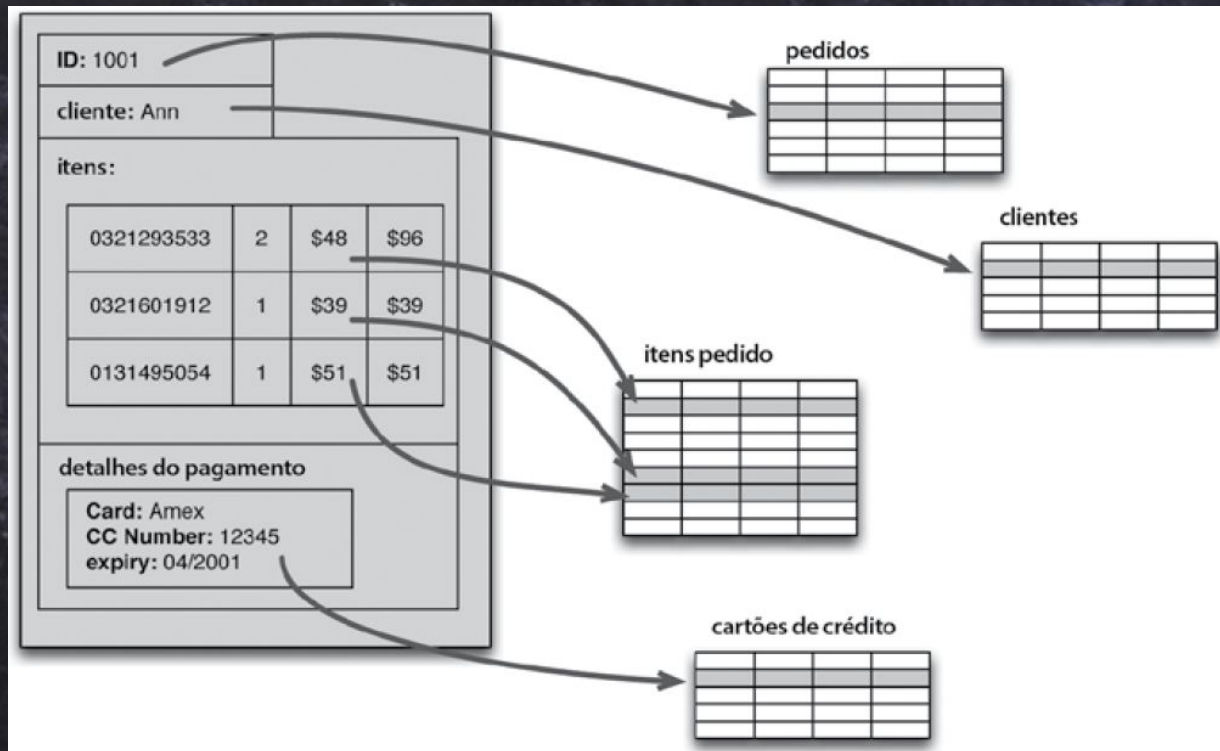
✗ Modelo de dados (UML):



BANCO DE DADOS RELACIONAL



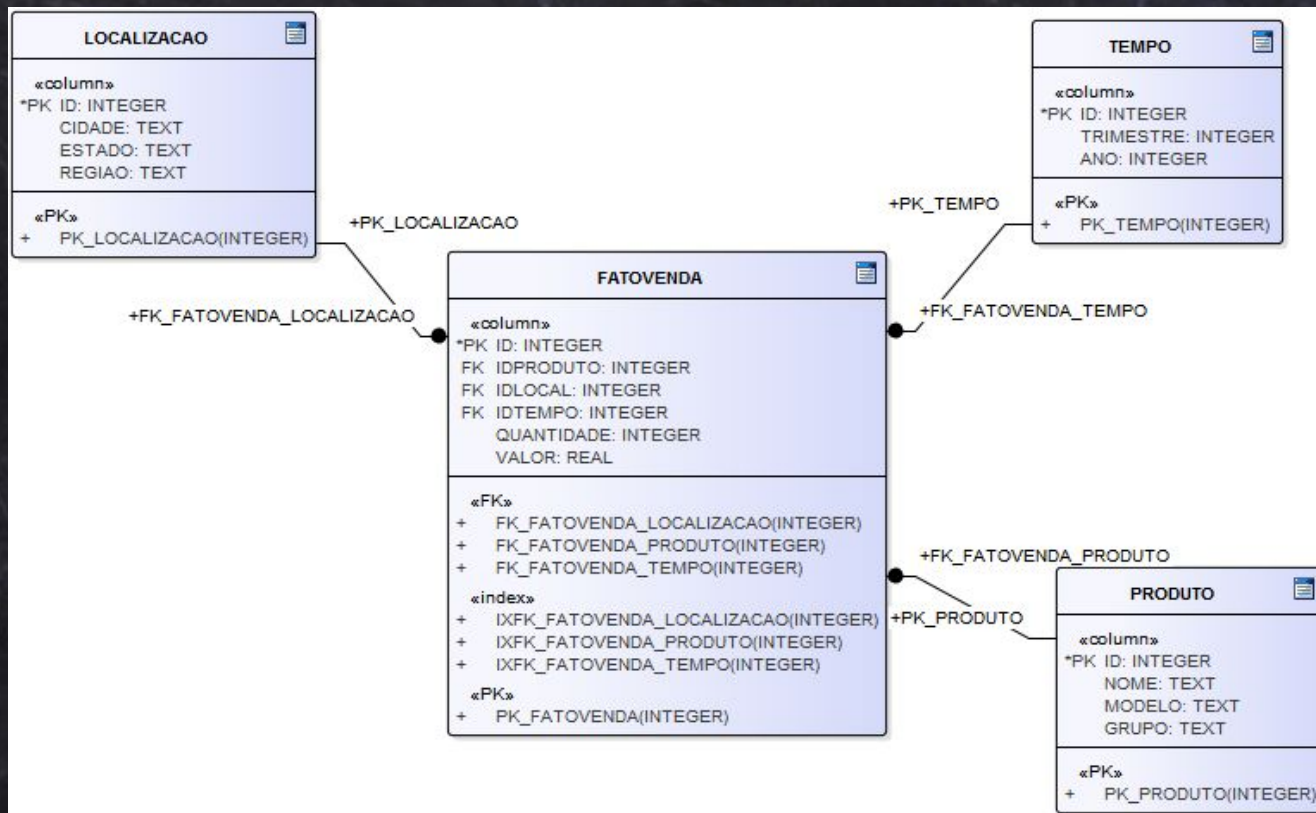
- ✗ Os dados são armazenados de forma segregada (normalizada) em um conjunto de tabelas;



MODELO RELACIONAL EM DATA WAREHOUSE



✗ Consiste na definição de um conjunto de **Data Marts** que tem uma tabela fato e um conjunto de tabelas de dimensões, como por exemplo:



MODELO RELACIONAL EM DATA WAREHOUSE



✗ O acesso aos dados em um banco de dados relacional deve ser feito por meio de operações CRUD utilizando linguagem SQL:

Consulta

```
SELECT
    L.REGIAO,
    ROUND (SUM (FV.VALOR) ) AS VENDA
FROM
    FATOVENDA FV,
    LOCALIZACAO L
WHERE
    FV.IDLOCAL = L.ID
GROUP BY
    L.REGIAO;
```

Resultado

Região	Venda
Sul	\$ 1.500,00
Sudeste	\$ 5.000,00
Nordeste	\$ 2.350,00
Norte	\$ 1.890,00
Centro-Oeste	\$ 1.732,00
Total de Vendas	\$ 11.472,00

MODELO RELACIONAL EM DATA WAREHOUSE



✗ Dependendo da complexidade da consulta, o script SQL tende a ficar maior, embora seja respeitando o padrão de escrita:

SELECT

L.REGIAO,
T.TRIMESTRE,
P.GRUPO,
ROUND (SUM (FV.VALOR)) AS VENDA

FROM

FATOVENDA FV,
TEMPO T,
LOCALIZACAO L,
PRODUTO P

WHERE

FV.IDLOCAL = L.ID
AND FV.IDTEMPO = T.ID
AND FV.IDPRODUTO = P.ID

GROUP BY

L.REGIAO, T.TRIMESTRE, P.GRUPO;

Região	Trimestre	Grupo de Produto	Venda
Sul			
	1		
		Enlatados	\$ 50,00
		Cereais	\$ 200,00
	2		
		Enlatados	\$ 300,00
		Cereais	\$ 500,00
	3		
		Enlatados	\$ 25,00
		Cereais	\$ 225,00
	4		
		Enlatados	\$ 200,00
		Cereais	\$ 100,00
Sudeste			
	1		
		Enlatados	\$ 700,00
		Cereais	\$ 300,00
	2		
		Enlatados	\$ 1.500,00

INSTALAÇÃO DO SQLLITE

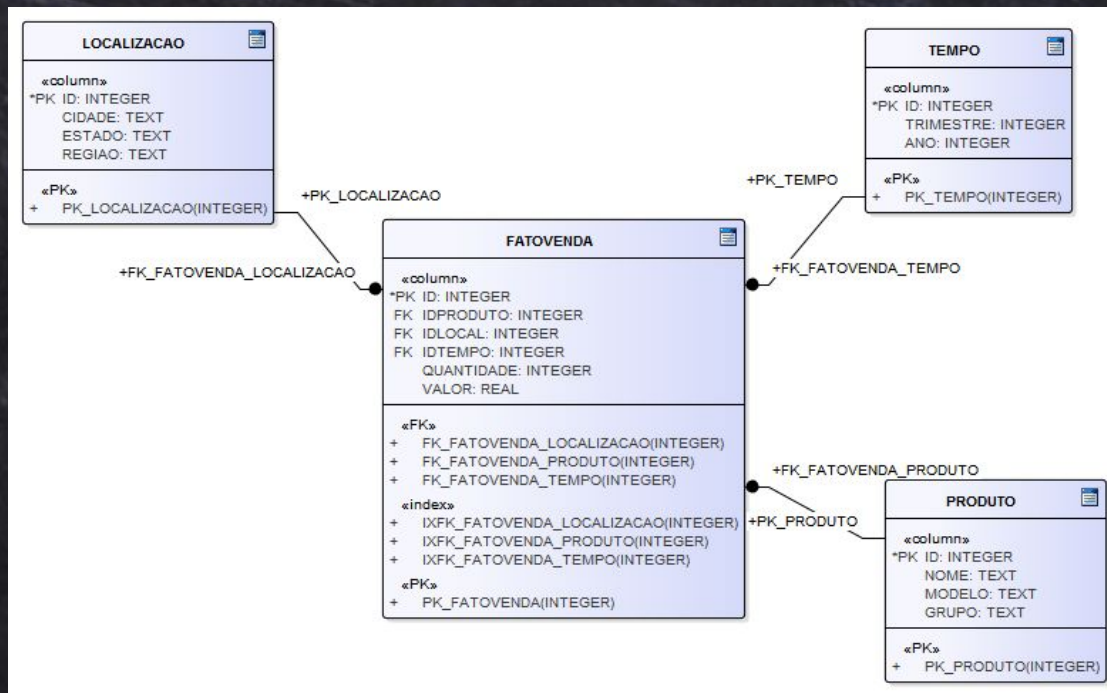


- ✗ Execute o comando abaixo para realizar a instalação do sqlite:
`sudo apt-get install sqlite3`
- ✗ Crie um DB para iniciar as atividades:
`sqlite3 meudb.db`
- ✗ Para melhorar a visualização habilite o modo coluna e a visualização de cabeçalho:
`.mode column`
`.headers on`
- ✗ O comando utilizado para sair do DB é:
`.quit`

EXERCÍCIO



1. Partindo do MER apresentado abaixo, vamos construir um Data Mart no nosso DB de testes (sqlite). Posteriormente faremos algumas consultas para a área de negócio.



CRIAÇÃO DE UM DB RELACIONAL



- ✗ Crie as tabelas dimensionais e a tabela fato utilizando os scripts abaixo:

```
CREATE TABLE LOCALIZACAO (ID INTEGER, CIDADE VARCHAR(30), ESTADO  
VARCHAR(2), REGIAO VARCHAR(10));
```

```
CREATE TABLE TEMPO (ID INTEGER, TRIMESTRE INTEGER, ANO INTEGER);
```

```
CREATE TABLE PRODUTO (ID INTEGER, NOME VARCHAR(100), MODELO INTEGER,  
GRUPO VARCHAR(10));
```

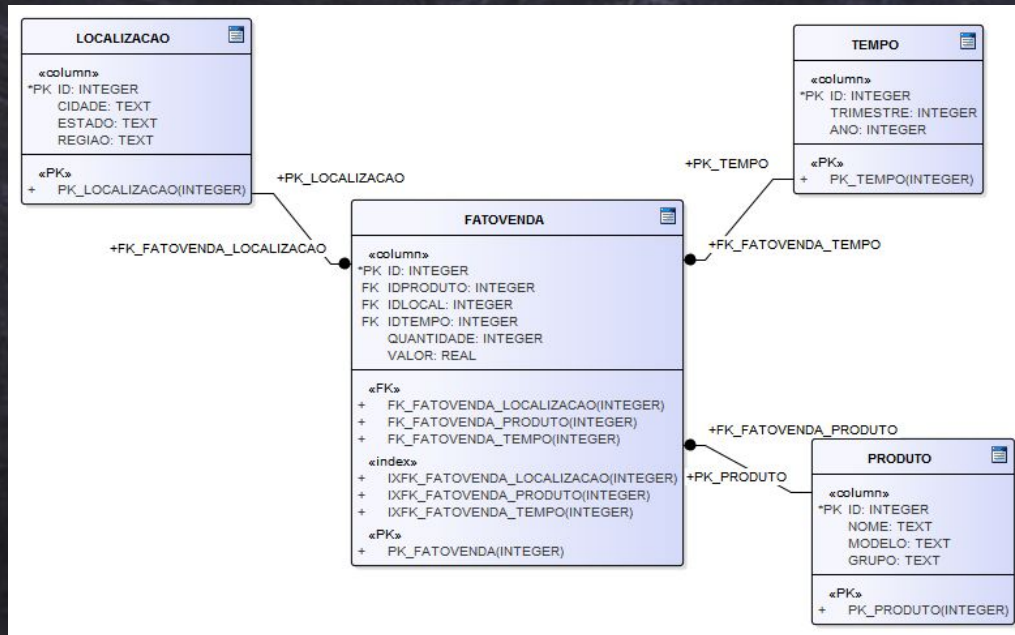
```
CREATE TABLE FATOVENDA (ID INTEGER, IDPRODUTO INTEGER, IDTEMPO  
INTEGER, IDLOCAL INTEGER, QUANTIDADE INTEGER, VALOR NUMBER(10,2));
```

- ✗ Utilize o script cargaDimensoes.sql para popular as tabelas de dimensão do Data Mart;
- ✗ Depois disso, utilize o script CargaFato.sql para dar uma carga no DW;



Depois de popular o DataMart do DW, elabore as consultas necessárias para obter:

- ✗ O relatório de vendas totais classificadas por região de vendas;
- ✗ O relatório de vendas totais classificadas por região de vendas e por trimestre;
- ✗ O relatório de vendas totais consolidadas por trimestre;
- ✗ O relatório de vendas totais, por região, por trimestre e por grupo de produto;



CONSULTAS EM UM DB RELACIONAL

```

1 SELECT
2     PEE.PESSOAIID AS ID, PEE.NOME AS NOME, PEE.TIPODOCUMENTO AS TIPODOCUMENTO, PEE.NUMERODOCUMENTO AS
3     NUMERODOCUMENTO, PEE.ATIVO AS ATIVO, PEE.PAIS AS PAIS, PEE.UF AS UF, PEE.CIDADE AS CIDADE, PEE.
4     BAIRRO AS BAIRRO, PEE.LOGRADOURO AS LOGRADOURO, PEE.NUMERO AS NUMERO, PEE.CEP AS CEP, 'CLIENTE'
5     AS TIPOCADASTRO
6
7 FROM
8 (
9     SELECT
10         PE.PESSOAIID, PE.NOME, PE.TIPODOCUMENTO, PE.NUMERODOCUMENTO, PE.ATIVO, 'Brasil' AS PAIS, END.UF,
11         END.MUNICIPIO AS CIDADE, PE.STATUSVALIDACAOID, END.BAIRRO AS BAIRRO, END.LOGRADOURO AS LOGRADOURO,
12         END.NUMERO AS NUMERO, END.CEP AS CEP
13
14 FROM
15 (
16     SELECT
17         P.ID AS PESSOAIID, PF.NOME AS NOME, 'CPF' AS TIPODOCUMENTO, PF.CPF AS NUMERODOCUMENTO, PF.SITUACAO AS
18         ATIVO, P.STATUSVALIDACAOID
19
20 FROM
21     PESSOA P,
22     PESSOA FISICA PF,
23     CLIENTE CL
24
25 WHERE
26     P.ID = PF.PESSOAIID AND
27     P.ID = CL.PESSOAIID AND
28     CL.CATEGORIACLIENTEID = 1
29
30 UNION
31
32 SELECT
33     P.ID AS PESSOAIID, PJ.NOMEFANTASIA AS NOME, 'CNPJ' AS TIPODOCUMENTO, PJ.CNPJ AS NUMERODOCUMENTO, PJ.
34     SITUACAO AS ATIVO, P.STATUSVALIDACAOID
35
36 FROM
37     PESSOA P,
38     PESSOA JURIDICA PJ,
39     CLIENTE CL
40
41 WHERE
42     P.ID = PJ.PESSOAIID AND
43     P.ID = CL.PESSOAIID AND
44     CL.CATEGORIACLIENTEID = 2
45
46 ) PE,
47 ENDEREÇO END
48
49 WHERE
50     END.PESSOAIID = PE.PESSOAIID
51
52 UNION
53
54 SELECT
55     PS.PESSOAIID, PS.NOME, PS.TIPODOCUMENTO, PS.NUMERODOCUMENTO, PS.ATIVO,
56     (SELECT P.DESCRICAO FROM PAIS P, ENDEREÇO EXTERIOR EE WHERE P.ID = EE.PAISID AND EE.PESSOAIID = PS.
57     PESSOAIID) AS PAIS,
58     (SELECT EE.ESTADOPROVINCIA FROM ENDEREÇO EXTERIOR EE WHERE EE.PESSOAIID = PS.PESSOAIID) AS UF,
59     (SELECT EE.CIDADE FROM ENDEREÇO EXTERIOR EE WHERE EE.PESSOAIID = PS.PESSOAIID) AS CIDADE, PS.
60     STATUSVALIDACAOID,
61     '' AS BAIRRO,
62     (SELECT EE.RUA FROM ENDEREÇO EXTERIOR EE WHERE EE.PESSOAIID = PS.PESSOAIID) AS LOGRADOURO,
63     (SELECT EE.NUMERO FROM ENDEREÇO EXTERIOR EE WHERE EE.PESSOAIID = PS.PESSOAIID) AS NUMERO,
64     (SELECT EE.CODIGOPOSTAL FROM ENDEREÇO EXTERIOR EE WHERE EE.PESSOAIID = PS.PESSOAIID) AS CEP
65
66 FROM
67 (SELECT P.ID AS PESSOAIID, PE.NOME AS NOME,
68     (SELECT PED.NUMERO FROM PESSOA ESTRANGEIRO DOCUMENTO PED WHERE PED.PESSOAESTRANGEIROID = PE.ID AND

```

```

46     (SELECT PED.NUMERO FROM PESSOAESTRANGEIRODOCUMENTO PED WHERE PED.PESSOAESTRANGEIROID = PE.ID AND
47     ROWNUM <= 1) AS NUMERODOCUMENTO,
48     (SELECT DE.DESCRICAO
49     FROM PESSOAESTRANGEIRODOCUMENTO PED, PAIS_DOCUMENTOESTRANGEIRO PD, DOCUMENTOESTRANGEIRO DE
50     WHERE
51     PED.PESSOAESTRANGEIROID = PE.ID AND
52     PED.PAISDOCUMENTOESTRANGEIROID = PD.ID AND
53     PD.DOCUMENTOESTRANGEIROID = DE.ID AND
54     ROWNUM <= 1) AS TIPODOCUMENTO,
55     PE.SITUACAO AS ATIVO, P.STATUSVALIDACAOID
56
57 FROM
58     PESSOA P,
59     PESSOAESTRANGEIRA PE,
60     CLIENTE CL
61
62 WHERE
63     P.ID = PE.PESSOAIID AND
64     P.ID = CL.PESSOAIID AND
65     CL.CATEGORIACLIENTEID = 3
66
67 ) PS
68
69 ) PEE
70
71 WHERE
72     PEE.STATUSVALIDACAOID = 4
73     AND PEE.ATIVO = 1
74
75 UNION
76
77 SELECT
78     PEE.PESSOAIID AS ID, PEE.NOME AS NOMEFILIAL, PEE.TIPODOCUMENTO AS TIPODOCUMENTO, PEE.
79     NUMERODOCUMENTO AS NUMERODOCUMENTO, PEE.ATIVO AS ATIVO, PEE.PAIS AS PAIS, PEE.UF AS UF, PEE.CIDADE
80     AS CIDADE, PEE.BAIRRO AS BAIRRO, PEE.LOGRADOURO AS LOGRADOURO, PEE.NUMERO AS NUMERO, PEE.CEP AS
81     CEP, 'FILIAL' AS TIPOCADASTRO
82
83 FROM
84 (
85     SELECT
86         PE.PESSOAIID, PE.NOME, 'CNPJ' AS TIPODOCUMENTO, PE.NUMERODOCUMENTO, PE.ATIVO, 'Brasil' AS PAIS, END
87         .UF, END.MUNICIPIO AS CIDADE, END.BAIRRO AS BAIRRO, END.LOGRADOURO AS LOGRADOURO, END.NUMERO AS
88         NUMERO, END.CEP AS CEP
89
90 FROM
91 (
92     SELECT
93         P.ID AS PESSOAIID, PJ.NOMEFANTASIA AS NOME, PJ.CNPJ AS NUMERODOCUMENTO, PJ.SITUACAO AS ATIVO
94
95 FROM
96     PESSOA P,
97     PESSOA JURIDICA PJ,
98     FILIAL FI
99
100 WHERE
101     P.ID = PJ.PESSOAIID AND
102     PJ.ID = FI.PESSOA JURIDICAID
103
104 ) PE,
105 ENDEREÇO END
106
107 WHERE
108     END.PESSOAIID = PE.PESSOAIID
109
110 ) PEE
111
112 WHERE
113     PEE.ATIVO = 1

```




Uma linguagem de manipulação de dados estruturada SQL (Structured Query Language) pressupõem o armazenamento de dados estruturados. E quando os dados armazenados não são estruturados?

PRINCIPAIS CARACTERÍSTICAS DE NOSQL



- ✗ Não tem uma linguagem **padronizada** para manipulação de dados;
- ✗ Cada sistema de gerenciamento de dados define sua própria linguagem;
- ✗ Tratamento particular para **Transações**;
- ✗ Não segue as premissas ACID;
- ✗ Problemas com integridade;

QUE FATORES DEFINEM O USO DE NOSQL?



- ✗ **Volume** de dados (google big table – 20 pentabytes/dia);
- ✗ **Frequência de acesso** aos dados;
- ✗ Acessos simultâneos (**Concorrência**) aos dados;
- ✗ Elasticidade do banco de dados (**Scalability**);
- ✗ **Natureza** dos dados;
- ✗ **Integridade** e **Consistência** esperadas;
- ✗ Como serão manipulados esses dados (**Tipos de operações**);

ACID



✕ Conjunto de propriedades satisfeitas por um DB transacional:

✕ **A**tomicity;

✕ **C**onsistency;

✕ **I**solation;

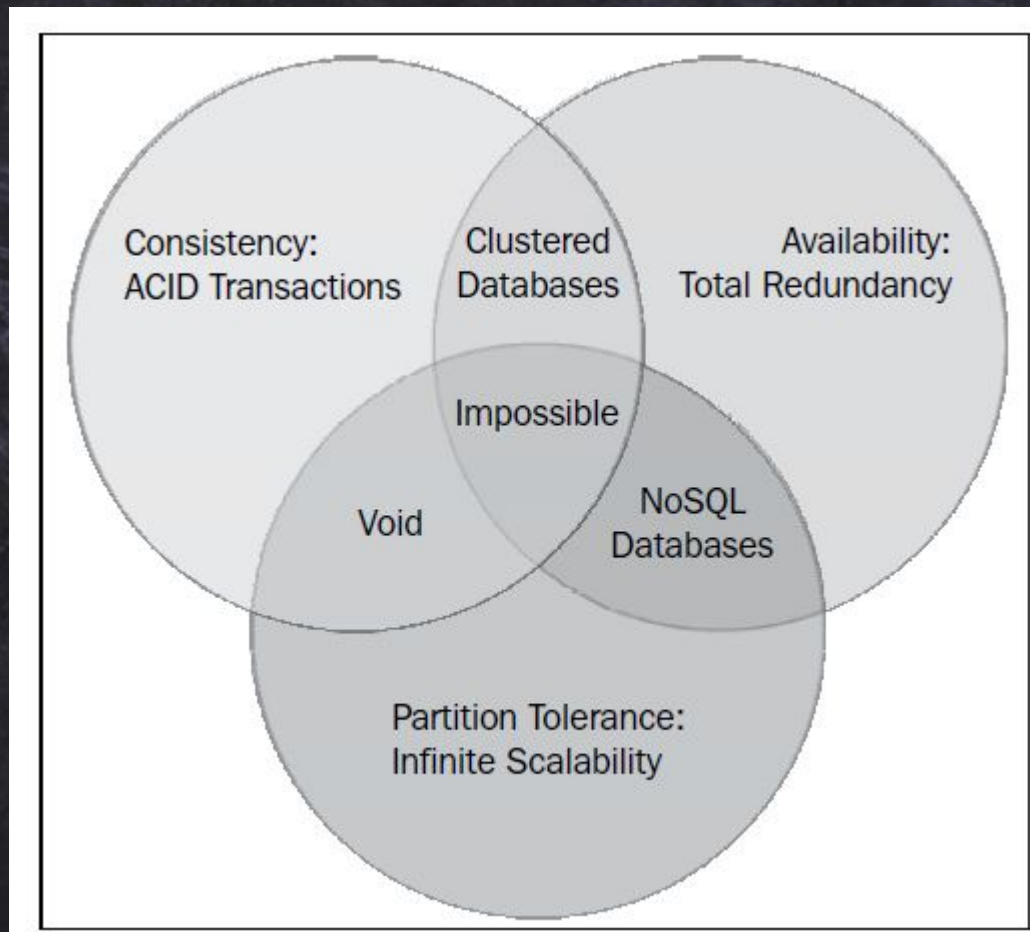
✕ **D**urability;

TEOREMA CAP (CONSIDERANDO SISTEMAS DISTRIBUIDOS)



- ✗ **Consistência (Consistency)** – A capacidade do sistema manter-se consistente depois da execução de operações sobre os dados; Para ser considerado consistente, um sistema distribuído deve refletir uma alteração, realizada em um nó, em todos os outros nós;
- ✗ **Disponibilidade (Availability)** – A capacidade do sistema continuar atendendo às suas funcionalidades, mesmo quando uma parte, software ou hardware, do sistema teve algum problema;
- ✗ **Tolerância a Partição (Partition Tolerance)** – Capacidade do sistema funcionar mesmo quando uma desconexão de partes (partição de nós em rede de computadores) do sistema ocorrer;

RELAÇIONAMENTO ENTRE CAP, ACID E NoSQL



BASE PROPRIEDADES



É Uma filosofia que **prioriza a disponibilidade em detrimento da consistência:**

- ✗ **Basicamente Disponível (Basically available)** – O Sistema garante a disponibilidade do teorema CAP, apresentando respostas para toda requisição. Respostas podem 'falhar' ou o dado pode estar inconsistente, em um estado de mudança;
- ✗ **Eventualmente Consistente (Eventualy Consistency)** – O Sistema estará eventualmente consistente uma vez que pare de receber inputs. O dado é propagado para todo lugar enquanto recebe inputs.
- ✗ **Estado Flexível (Soft state)** – O estado do sistema muda com o tempo. Mudança ocorrem durante o período sem inputs.

TIPOS DE SISTEMAS NoSQL



- ✗ Armazenamento Chave-Valor (**Key-Value**);
- ✗ Bancos de dados orientados a Colunas (**Column-Oriented**);
- ✗ Bancos de dados orientados a Documentos (**Document-Oriented**);
- ✗ Bancos de dados orientados a Grafos (**Graphs-Oriented**);

ARMAZENAMENTO CHAVE-VALOR



- ✗ Conhecidos como tabelas de hash distribuídas;
- ✗ Armazenam objetos indexados por chave;
- ✗ Permitem a busca dos objetos por meio de suas chaves;
- ✗ Alguns bancos que suportam esse tipo de armazenamento: DynamoDb, Couchbase, Riak, Azure Table Storage, Redis, Tokyo Cabinet, Berkeley DB...

ARMAZENAMENTO ORIENTADO A COLUNA



- ✘ Em um banco de dados relacional o armazenamento é feito linha a linha de uma tabela de forma contigua, exemplo: **Id1, nome1, cpf1, Id2, nome2, cpf2, Id3, . . .**
- ✘ O armazenamento orientado a coluna permite que os dados das colunas sejam guardados de forma contigua. Exemplo: **Id1, Id2, Id3; Nome1, Nome2, Nome3; Cpf1, Cpf2, Cpf3; ...**
- ✘ Banco de dados relacionais são mais adequados para OLTP, enquanto que banco de dados orientados a coluna obtém destaque para uso de OLAP;
- ✘ Sistemas que suportam esse modo de armazenamento são: Hadoop, Cassandra, Hypertable, Amazon SimpleDB, ...

ARMAZENAMENTO ORIENTADO A DOCUMENTO



- ✗ Um documento pode ser uma coleção de atributos e valores;
- ✗ Um atributo pode ser multi-valorado;
- ✗ Não possuem esquema/estrutura;
- ✗ Permitem o armazenamento de dados semi estruturados
- ✗ Alguns sistemas que suportam esse tipo de armazenamento são: MongoDB, CouchDB, RavenDb

ARMAZENAMENTO ORIENTADO A GRAFOS



- ✗ Armazenamento baseado em estruturas matemáticas denominadas grafo;
- ✗ Nesse tipo de armazenamento, prioriza-se a informação de interconectividade ou a topologia dos dados;
- ✗ Basicamente, esse modelo é representado por nós, arestas e atributos (tanto dos nós quanto das arestas);
- ✗ Bancos de dados que suportam esse modo de armazenamento são Neo4J, Infinite Graph, InforGrid, HyperGraphDB.

RANKING DOS DBs



378 systems in ranking, September 2021

Rank Sep 2021	Aug 2021	Sep 2020	DBMS	Database Model	Score		
					Sep 2021	Aug 2021	Sep 2020
1.	1.	1.	Oracle +	Relational, Multi-model	1271.55	+2.29	-97.82
2.	2.	2.	MySQL +	Relational, Multi-model	1212.52	-25.69	-51.72
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model	970.85	-2.50	-91.91
4.	4.	4.	PostgreSQL +	Relational, Multi-model	577.50	+0.45	+35.22
5.	5.	5.	MongoDB +	Document, Multi-model	496.50	-0.04	+50.02
6.	6.	7.	Redis +	Key-value, Multi-model	171.94	+2.05	+20.08
7.	7.	6.	IBM Db2	Relational, Multi-model	166.56	+1.09	+5.32
8.	8.	8.	Elasticsearch	Search engine, Multi-model	160.24	+3.16	+9.74
9.	9.	9.	SQLite +	Relational	128.65	-1.16	+1.98
10.	11.	10.	Cassandra +	Wide column	118.99	+5.33	-0.18
11.	10.	11.	Microsoft Access	Relational	116.94	+2.10	-1.51
12.	12.	12.	MariaDB +	Relational, Multi-model	100.70	+1.72	+9.09
13.	13.	13.	Splunk	Search engine	91.61	+1.01	+3.71
14.	14.	15.	Hive +	Relational	85.58	+1.64	+14.41
15.	15.	17.	Microsoft Azure SQL Database	Relational, Multi-model	78.26	+3.11	+17.81
16.	16.	16.	Amazon DynamoDB +	Multi-model	76.93	+2.03	+10.75
17.	17.	14.	Teradata +	Relational, Multi-model	69.68	+0.86	-6.72
18.	18.	21.	Neo4j +	Graph	57.63	+0.68	+7.00
19.	19.	19.	SAP HANA +	Relational, Multi-model	56.24	+0.66	+3.38
20.	21.	23.	FileMaker	Relational	52.32	+2.04	+4.75
21.	23.	107.	Snowflake +	Relational	52.07	+5.53	+49.39
22.	20.	20.	Solr	Search engine, Multi-model	49.81	-1.26	-1.81
23.	22.	18.	SAP Adaptive Server	Relational, Multi-model	47.01	-0.60	-7.00
24.	24.	22.	HBase +	Wide column	45.05	+0.41	-3.30
25.	25.	24.	Google BigQuery +	Relational	43.92	+1.89	+10.60
26.	26.	25.	Microsoft Azure Cosmos DB +	Multi-model	38.52	+1.66	+6.86
27.	27.		PostGIS	Spatial DBMS, Multi-model	30.71	+0.49	
28.	28.	29.	InfluxDB +	Time Series, Multi-model	29.50	-0.06	+6.17
29.	29.	26.	Couchbase +	Document, Multi-model	27.68	-0.14	-2.92
30.	30.	27.	Memcached	Key-value	25.66	-0.32	-0.68
31.	31.	32.	Firebird	Relational	24.83	-0.52	+3.96
32.	33.	30.	Amazon Redshift +	Relational	23.86	+1.00	+0.85

Method of calculating the scores of the DB-Engines Ranking

The DB-Engines Ranking is a list of database management systems ranked by their current popularity. We measure the popularity of a system by using the following parameters:

- **Number of mentions of the system on websites**, measured as number of results in search engines queries. At the moment, we use [Google](#) and [Bing](#) for this measurement. In order to count only relevant results, we are searching for <system name> together with the term database, e.g. "Oracle" and "database".
- **General interest in the system.** For this measurement, we use the frequency of searches in [Google Trends](#).
- **Frequency of technical discussions about the system.** We use the number of related questions and the number of interested users on the well-known IT-related Q&A sites [Stack Overflow](#) and [DBA Stack Exchange](#).
- **Number of job offers, in which the system is mentioned.** We use the number of offers on the leading job search engines [Indeed](#) and [Simply Hired](#).
- **Number of profiles in professional networks, in which the system is mentioned.** We use the internationally most popular professional network [LinkedIn](#).
- **Relevance in social networks.** We count the number of [Twitter](#) tweets, in which the system is mentioned.



ANÁLISE DE COMPLEXIDADE

Analise de Complexidade de Algoritmos



COMPLEXIDADE DE ALGORITMOS

Complexidade de da busca1:

```
1  #include <iostream>
2  #include <vector>
3  #include <string>
4  using namespace std;
5  int main ()
6  {
7      std::vector<string> dados;
8      string temp;
9      char buffer[100];
10     int n = 1000;
11
12     // CARGA DE DADOS
13     for(int i=0; i < n ; i++)
14     {
15         std::cin >> buffer;
16         dados.push_back(buffer);
17     }
18
19     // BUSCA POR DADO
20     for(int i=0; i < dados.size() ; i++)
21     {
22         if(dados[i] == "abacaxi")
23         {
24             std::cout << " ENCONTROU!!" << endl;
25             std::cout << "dados["<<i<<"] "<< dados[i] << endl;
26             break;
27         }
28     }
29     return 0;
30 }
```

Quantos acessos serão feitos para encontrar o dado “abacaxi”?

COMPLEXIDADE DE ALGORITMOS



considere o número de operações de cada um dos dois algoritmos que resolvem o mesmo problema, como função de n .

- Algoritmo 1: $f_1(n) = 2n^2 + 5n$ operações
- Algoritmo 2: $f_2(n) = 500n + 4000$ operações

Dependendo do valor de n , o Algoritmo 1 pode requerer mais ou menos operações que o Algoritmo 2.

(Compare as duas funções para $n = 10$ e $n = 100$.)

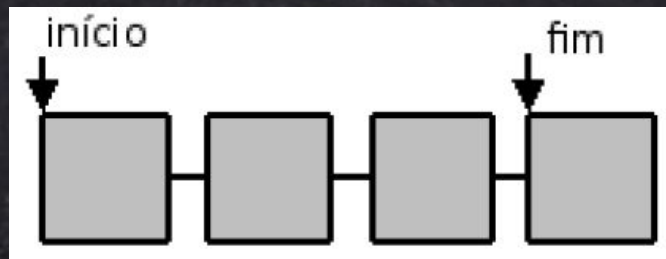
Considere 5 algoritmos com as complexidades de tempo.
Suponhamos que uma operação leve 1 ms.

n	$f_1(n) = n$	$f_2(n) = n \log n$	$f_3(n) = n^2$	$f_4(n) = n^3$	$f_5(n) = 2^n$
16	0.016s	0.064s	0.256s	4s	1m 4s
32	0.032s	0.16s	1s	33s	46 dias
512	0.512s	9s	4m 22s	1 dia 13h	10^{137} séculos

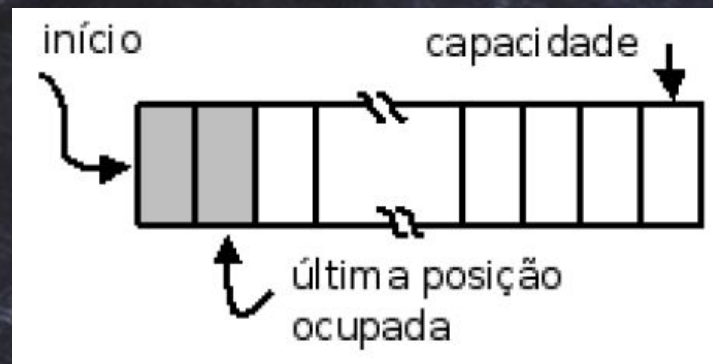
ESTRUTURA DE DADOS



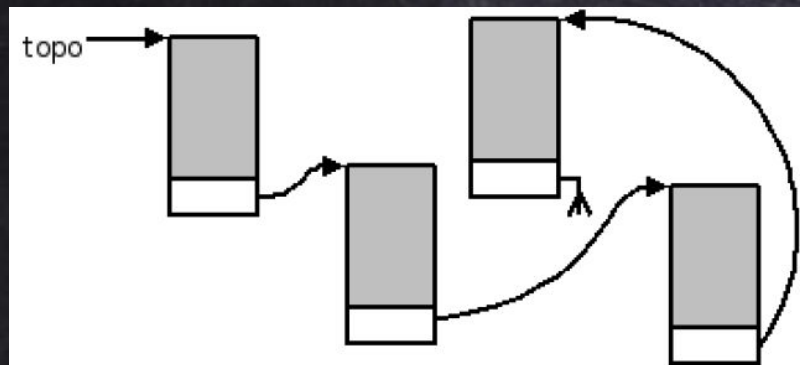
Pilha/fila



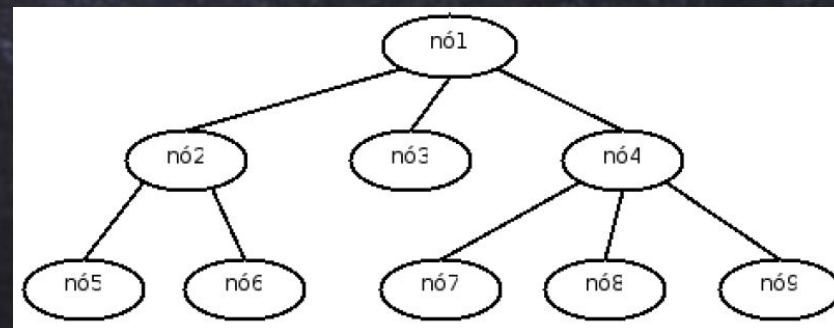
Vetor



Lista ligada/fila



Arvore



ESTRUTURAS DE DADOS



- ✗ Existe variação de tempo para realizar as operações CRUD em diferentes estruturas de dados?

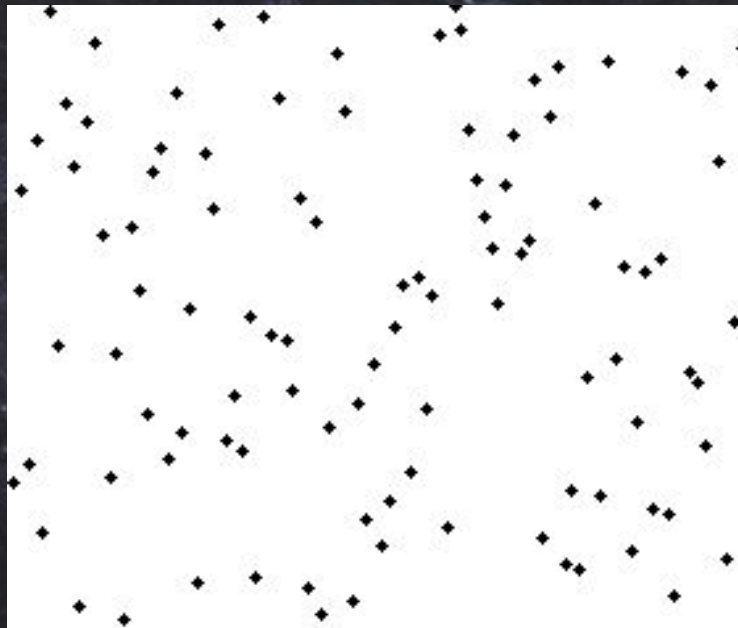
	Vetor	Pilha	Lista ligada	Arvore
Create	?	?	?	?
Read	?	?	?	?
Update	?	?	?	?
Delete	?	?	?	?

- ✗ Existe mudança na variação de tempo ao considerarmos dados com natureza aleatória ou ordenada?

COMPLEXIDADE DE ALGORITMOS



- ✗ Considerando a ordenação de um vetor de n elementos inteiros, temos:

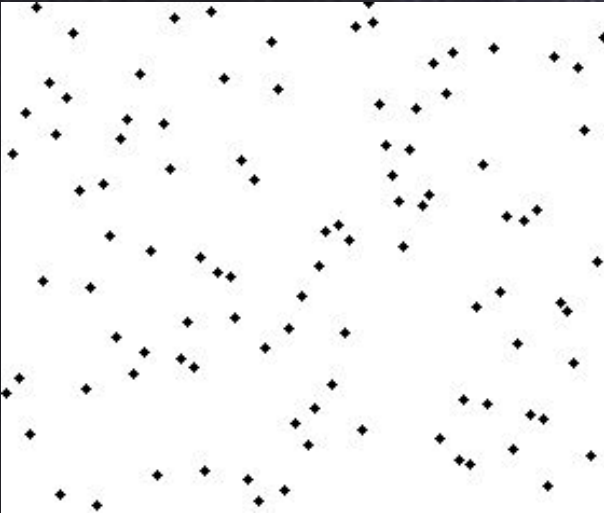


Bubble Sort

COMPLEXIDADE DE ALGORITMOS



✗ Considerando a ordenação de um vetor de n elementos inteiros, temos:



Merge Sort



COMPLEXIDADE DE ALGORITMOS



✘ Considerando a ordenação de um vetor de n elementos inteiros, temos:

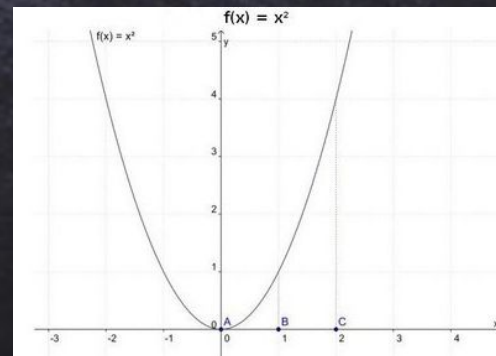
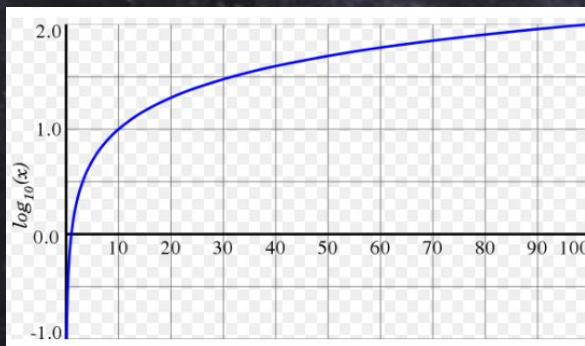
✘

Algoritmo	Pior Caso	Melhor Caso	
Quick Sort			?
Bubble Sort			?
Merge Sort			?
Heap Sort			?

COMPLEXIDADE DE ALGORITMOS



Algoritmo	Comparações			Movimentações		
	Melhor	Médio	Pior	Melhor	Médio	Pior
Bubble	$O(n^2)$			$O(n^2)$		
Selection	$O(n^2)$			$O(n)$		
Insertion	$O(n)$	$O(n^2)$		$O(n)$	$O(n^2)$	
Merge	$O(n \log n)$			–		
Quick	$O(n \log n)$		$O(n^2)$	–		





REFERÊNCIAS

- ✕ Algoritmos – <https://www.ime.usp.br/~pf/algoritmos>, acessado em 09/05/2019.
- ✕ Peter A. Boncz, Martin L. Kersten, Stefan Manegold. Breaking the memory wall in MonetDB. communications of the acm, Dec. 2008, Vol. 51, N° 12.
- ✕ Introduction toMulticore architecture Tao Zhang Oct. 21, 2010
- ✕ Conheça os principais algoritmos
<https://www.treinaweb.com.br/blog/conheca-os-principais-algoritmos-de-ordenacao>,
acessado em 11/09/2023



OBRIGADO!

Copyright © 2019 Prof. MSc. Eng. Wakim B. Saba

<https://br.linkedin.com/in/wakimsaba>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).