



Coleta e Obtenção de Dados

Capítulo 1: Conceitos fundamentais

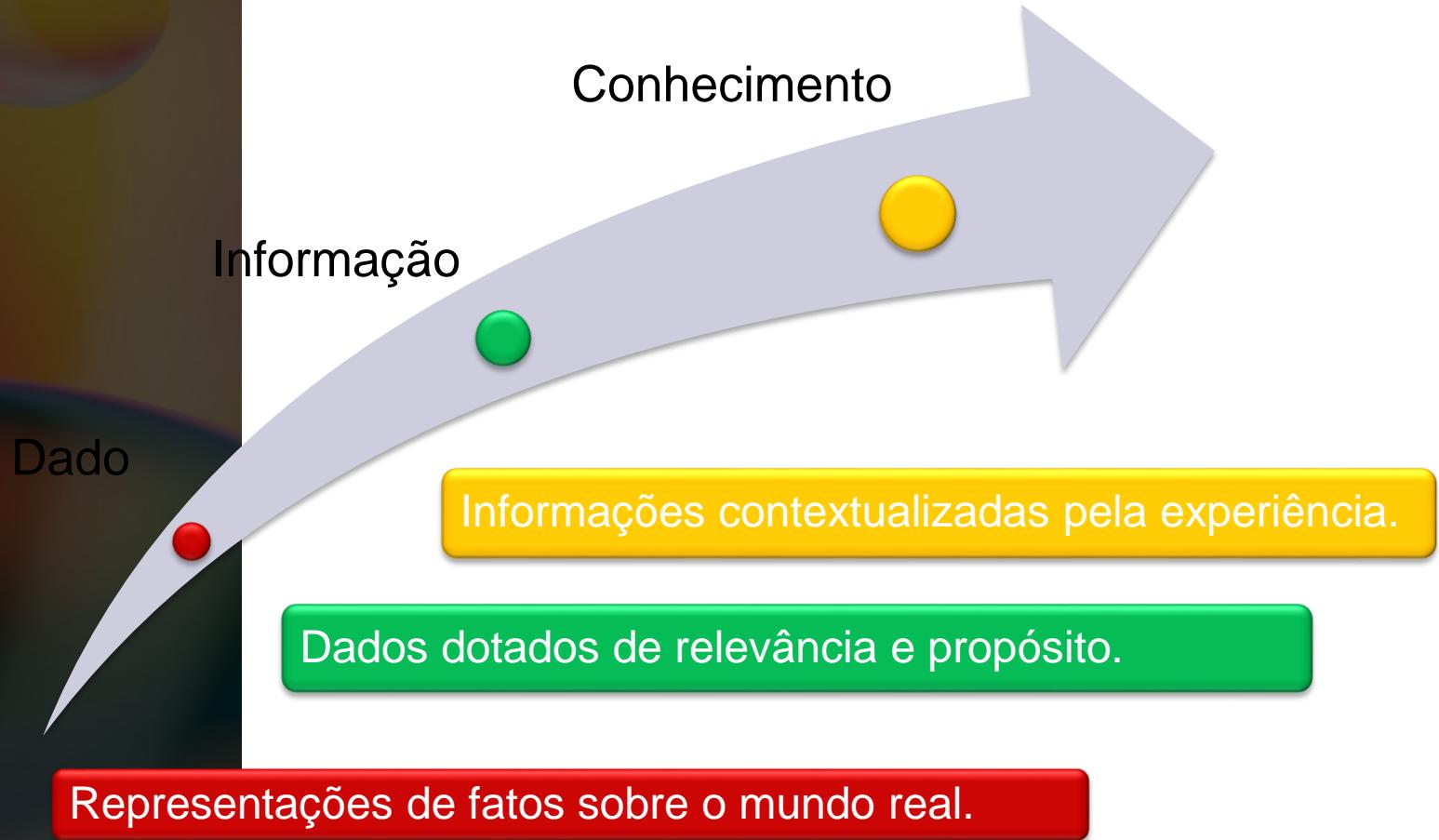
Profa. Fernanda Farinelli



Aula 1.1. Dados e fontes de Dados

Nesta aula

- Entender o conceito de dado.
- Conhecer os diferentes tipo de dados.
- Conhecer as diferentes fontes de dados.



Tipos de dados



Estruturado

Estrutura homogênea e pré-definida.

Estrutura prescritiva.

Estrutura independente dos dados.

Clara distinção entre estrutura e dados.

Fracamente evolutiva.



Semiestruturado

Esquema heterogêneo e nem sempre pré-definido.

Estrutura descritiva.

Estrutura embutida nos dados.

Distinção entre estrutura e dados pouco clara.

Fortemente evolutiva, onde a estrutura sofre mudanças com frequência.



Não estruturado

Sem esquema pré-definido.

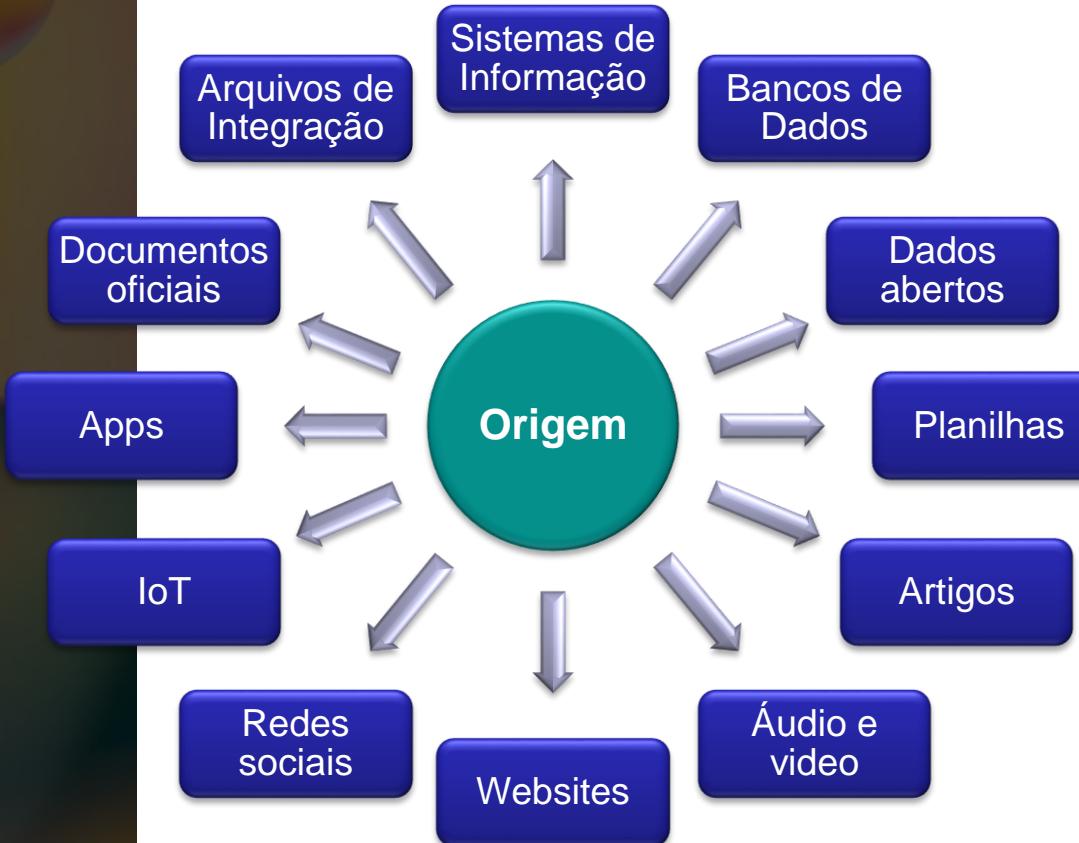
Estrutura descritiva.

Estrutura de dados irregular nem sempre presente.

Distinção entre estrutura e dados pouco clara.

Fortemente evolutiva, onde a estrutura sofre mudanças com frequência.

Fontes de Dados



Exemplos de dados de Redes Sociais

Perfis dos usuários

Avaliações ou reações

Influenciadores

Top Lists

Comentários

Listas de Favoritos

Metadados

Conclusão

- Entendemos o conceito de dado, informação e conhecimento.
- Conhecemos os diferentes tipo de dados: Estruturados, semiestruturados e não estruturados.
- Entendemos o que são fontes de dados e conhecemos algumas destas fontes.

Próxima aula

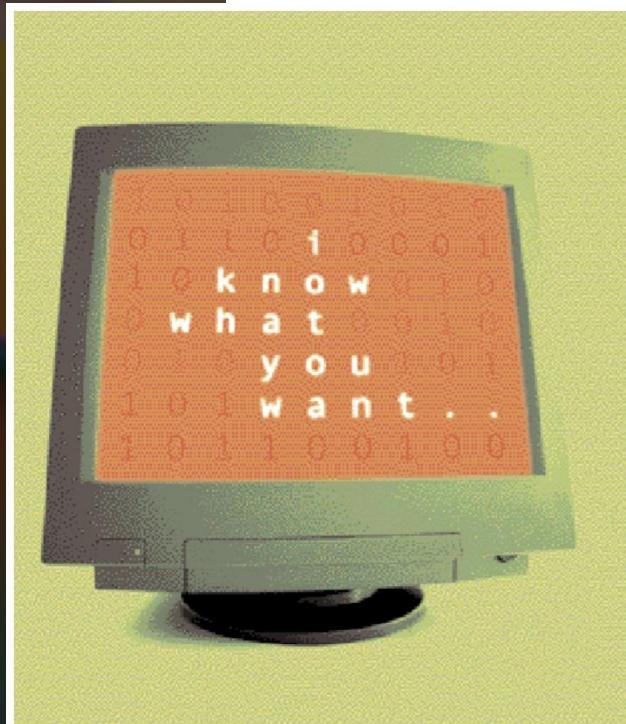
- Entender o que é a web semântica e suas principais características.



Aula 1.2. Web semântica

Nesta aula

- Entender o que é a web semântica e suas principais características.



THE SEMANTIC WEB

A new form of Web content
that is meaningful to computers
will unleash a revolution of new abilities

by

TIM BERNERS-LEE,
JAMES HENDLER and
ORA LASSILA

Hendler, J., Lassila, O., & Berners-Lee, T. (2001). The semantic web. *Scientific American*, 284(5), 34-43.

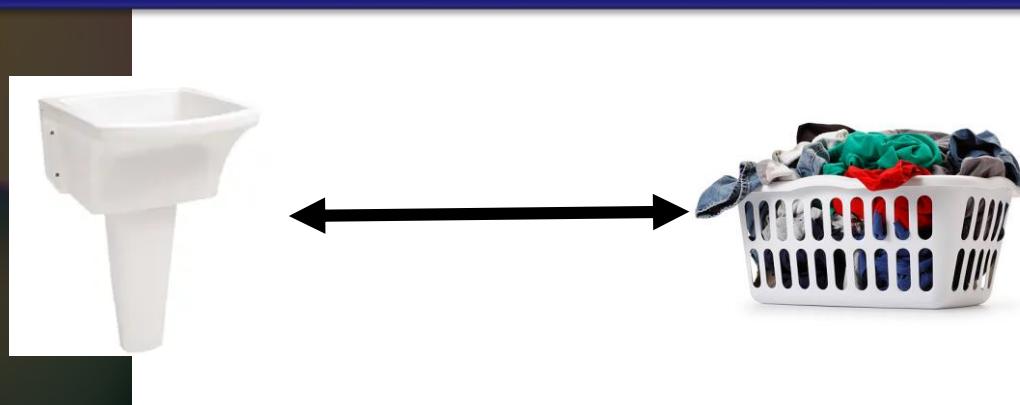
“A Web Semântica não é uma Web separada mas sim uma extensão da Web atual onde à informação é atribuído significado bem definido, permitindo que pessoas e computadores trabalhem em cooperação”.

Semântica: representa o estudo do significado.

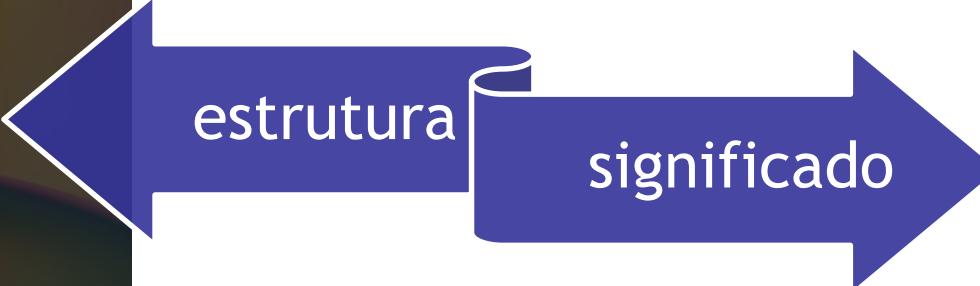
Tanque



Conjunto de dados inter-relacionados.



Visa interligar os significados das palavras, tornando perceptível tanto a humanos quanto as máquinas.



estrutura

significado

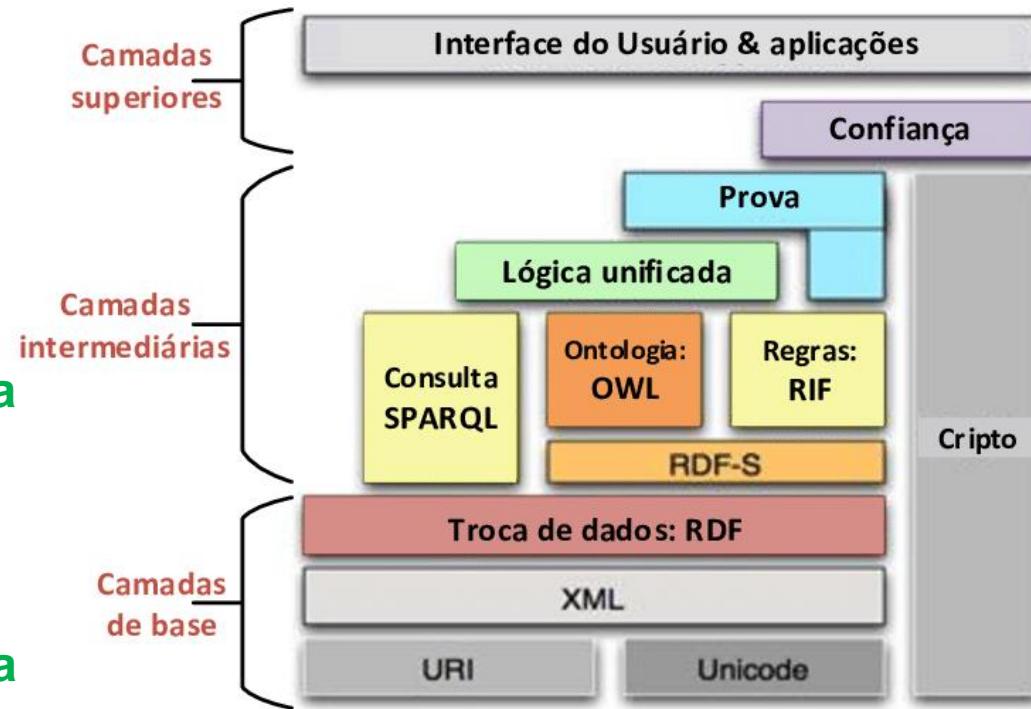
As camadas da Web Semântica

- Arquitetura Tecnológica atual da Web Semântica:



Semântica

Sintática



Identificador Uniforme de Recursos

Meio simples e extensível, formado por uma cadeia de caracteres compacta, para identificar um recurso na internet.

Permite a interação com representações do recurso através da internet (mundial), usando protocolos específicos como o HTTP.



Uniform Resource Identifier (URI)

- Exemplos:

- <ftp://ftp.is.co.za/rfc/rfc1808.txt>
- <http://www.math.uio.no/faq/compression-faq/part1.html>
- <mailto:mduerst@ifi.unizh.ch>

Linguagem de representação de informação na Internet.

Uma sintaxe padrão para representar um grafo dirigido em XML.

Recurso pode ser uma imagem, uma página, etc, tudo o que está disponível na internet precisa de um identificador único para que não seja confundido.

Um modelo em RDF é um conjunto de declarações.

declaração = (predicado,sujeito,objeto)

Sua representação segue o conceito de triplas.



declaração = (predicado,sujeito,objeto)

- Predicado é um recurso
- Sujeito é um recurso
- Objeto = Predicado(Sujeito)
- Objeto é ambos um recurso ou um literal

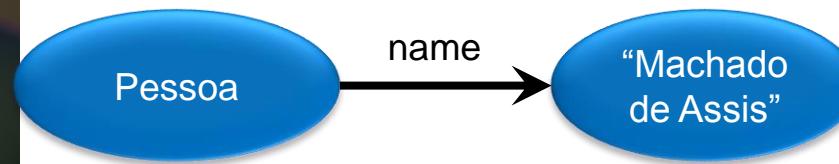
Um recurso pode ser qualquer coisa que tenha identidade.

Um recurso deve ser identificado por uma URI.

RDF (*Resource Description Framework*)

- declaração = (predicado,sujeito,objeto)

```
<rdf:Statement>  
  <rdf:subject resource="http://example.org/pessoa/25">  
  <rdf:predicate resource="http://xmlns.com/foaf.0.1/name">  
  <rdf:object>Machado de Assis</rdf:object>  
</rdf:Statement>
```



Conclusão

- Entendemos o que é a web semântica e suas principais características.

Próxima aula

- Entender o que são dados abertos.
- Entender o conceito de *linked data* e *linked open data*.



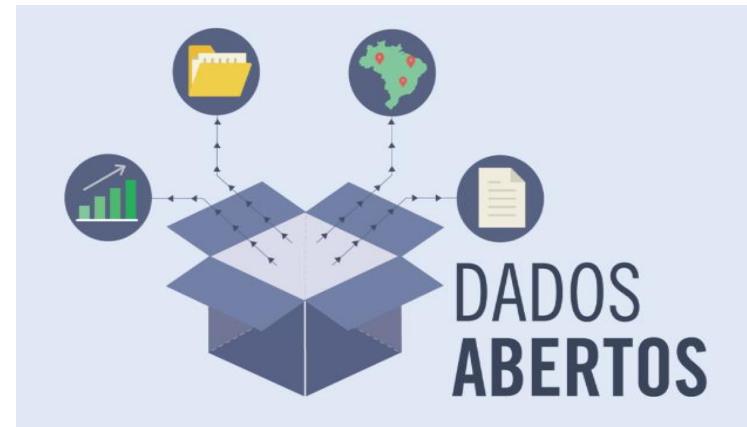
Aula 1.3. Dados abertos e *linked data*

Nesta aula

- Entender o que são dados abertos.
- Entender o conceito de *linked data* e *linked open data*.
- Visualizar exemplos de dados abertos.
- Conhecer fontes de dados para acesso a dados abertos.

Dados Abertos (*open data*)

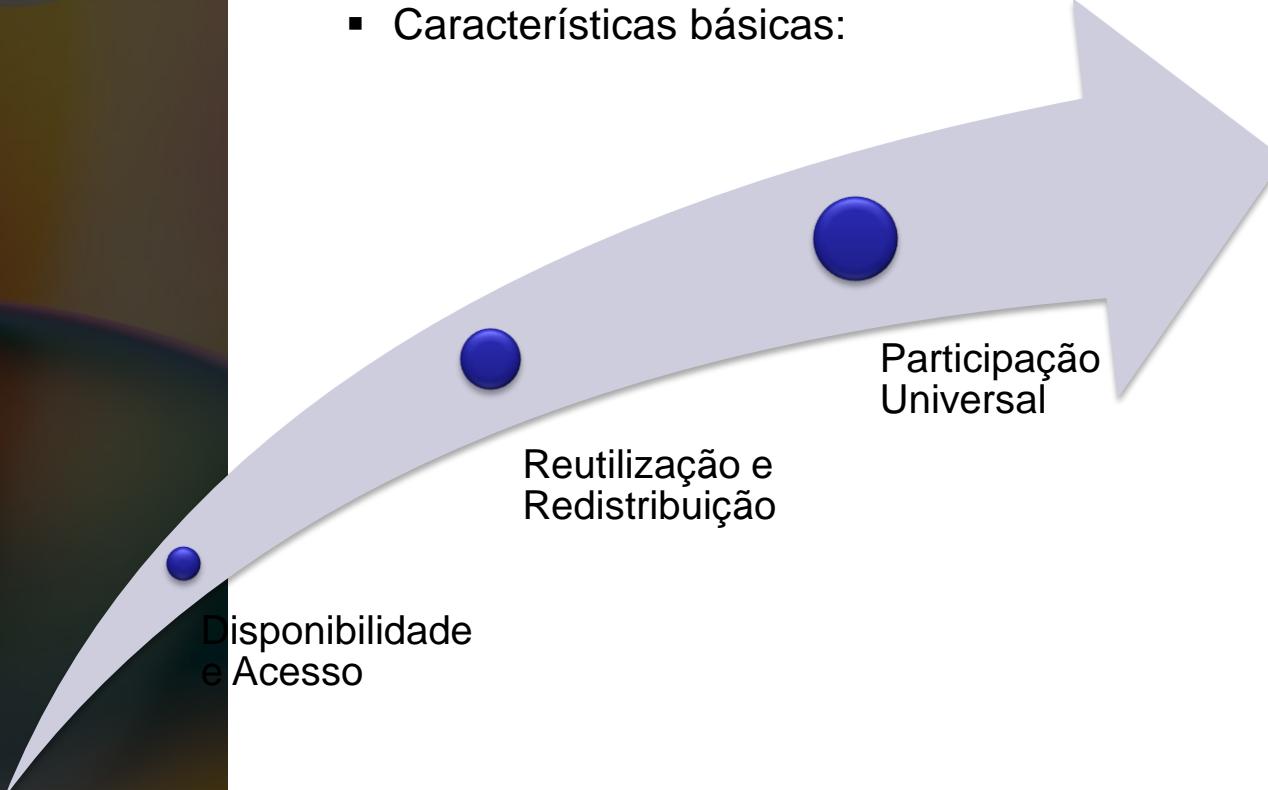
IGTI





Dados Abertos (*open data*)

- Características básicas:



Disponibilidade
e Acesso

Reutilização e
Redistribuição

Participação
Universal

Dados Abertos

- Regulamentações:
 - Lei de Acesso à Informação
 - Lei Geral de Proteção de Dados Pessoais (LGPD)
 - Regulamento Geral sobre a Proteção de Dados (GDPR)

Portal Brasileiro de Dados Abertos

IGTI

Disponível: <http://dados.gov.br>

The screenshot shows the homepage of the Brazilian Open Data Portal (dados.gov.br). The top navigation bar includes links for Simplifique!, Participe, Acesso à informação, Legislação, Canais, and a search bar. The main content area features a search bar with placeholder text "Pesquisar conjuntos de dados..." and a magnifying glass icon. Below the search bar, there are three social media icons: RSS, Twitter, and Facebook. The main content area displays a list of 6,642 data sets found, with options to "Ordenar por" (Sort by) Relevância (Relevance). There are also CSV and PDF download buttons. Other sections include "Operadoras Acreditadas" (Accredited Operators) and "Informações consolidadas de Beneficiários" (Consolidated Information of Beneficiaries), each with their own download buttons.

BRASIL Serviços

Ir para o conteúdo 1 Ir para o menu 2 Ir para a busca 3 Ir para o rodapé 4

Pesquisar conjuntos de dados...

dados.gov.br

PORTAL BRASILEIRO DE DADOS ABERTOS

Dados | Organizações | Aplicativos | Inventários | Concursos | JNDA | Perguntas frequentes | Contato | Sobre o portal

[Organizações](#)

Banco Central do Br... (3104)

Instituto Brasileir... (419)

Estado de Alagoas - AL (220)

Agência Nacional de... (205)

Distrito Federal (145)

Ministério da Saúde... (145)

Ministério da Fazen... (136)

Previdência Social (117)

Ministério do Plane... (87)

Secretaria de Gover... (83)

[Mostrar mais Organizações](#)

Buscar conjunto de dados...

6.642 conjuntos de dados encontrado(s)

Ordenar por: Relevância

Operadoras Acreditadas

Lista das operadoras de planos de saúde que, voluntariamente, passaram por processo de avaliação da adequação e eficiência dos serviços disponibilizados por essas operadoras, e...

[CSV](#) [PDF](#)

Informações consolidadas de Beneficiários

Informações consolidadas de beneficiários por competência

[zip+csv](#) [ODS](#)

Gestão de Pessoas (Executivo Federal) - Cargos Vagos

Portal da Transparéncia

IGTI

Disponível: <http://www.portaldatransparencia.gov.br/>

Ir para o conteúdo 1 | Ir para o menu 2 | Ir para a busca 3 | Ir para o rodapé 4 | A+ | A- | ACESIBILIDADE | ALTO CONTRASTE | MAPA DO SITE

Portal da Transparéncia

CONTROLDORIA-GERAL DA UNIÃO

Sobre o Portal ▾ | Painéis ▾ | Consultas Detalhadas ▾ | Controle social ▾ | Rede de Transparência | Receba Notificações | Aprenda mais ▾

Todos Busque por orgão, cidade, CNPJ, servidor... 



PANORAMA
» Governo Federal
» Pessoas jurídicas
» Pessoas físicas



Consulte os documentos diários de despesa



Sanções
» Painel
» Empresas Inidôneas e Suspensas
» Empresas Punidas
» Entidades Privadas sem Fins Lucrativos



ORÇAMENTO



DESPESAS E RECEITAS



ÓRGÃOS



ESTADOS E MUNICÍPIOS



LICITAÇÕES E CONTRATOS



CONVÊNIOS



CARTÕES DE PAGAMENTO



RECURSOS TRANSFERIDOS



BENEFÍCIOS AO CIDADÃO



SERVIDORES



VIAGENS A SERVIÇO



EMENDA PARLAMENTAR

Disponível: <http://receita.economia.gov.br/dados>

Ir para o conteúdo [1](#) Ir para o menu [2](#) Ir para a busca [3](#) Ir para o rodapé [4](#)

ACESSIBILIDADE ALTO CONTRASTE MAPA DO SITE

Receita Federal

MINISTÉRIO DA ECONOMIA

Buscar no portal 

[Twitter](#) [YouTube](#) [Facebook](#)

Perguntas Frequentes | Contato | Serviços | Dados e Estudos | Área de Imprensa | Onde Encontro | Avisos | English | Español

VOCÊ ESTÁ AQUI: PÁGINA INICIAL > DADOS ABERTOS

 Receita Federal

ACESSO RÁPIDO

Agendamento
Agenda Tributária
Dados e Estudos
e-CAC
Educação Fiscal
Idoso
Legislação
Processo e Dossiê
Tributos

Dados Econômico-Tributários e Aduaneiros da Receita Federal

ADUANA ARRECADAÇÃO ATENDIMENTO E SIMPLIFICAÇÃO

CADASTRO COBRANÇA * CONTENCIOSO ADMINISTRATIVO

ESTUDOS TRIBUTÁRIOS E ADUANEIROS FISCALIZAÇÃO MERCADORIAS APREENDIDAS

PARCELAMENTO DE DÉBITOS PESQUISA E INVESTIGAÇÃO RENÚNCIA FISCAL

RESTITUIÇÃO IRPF

- Diversos datasets que podem ser usados para aprendizado.
 - Disponível: <https://www.kaggle.com/datasets>

The screenshot shows the Kaggle Datasets page. At the top, there is a navigation bar with links for kaggle, Search, Competitions, Datasets, Notebooks, Discussion, Courses, Sign In, and Register. Below the navigation bar, the page title is "Datasets". There are two buttons: "Documentation" and "New Dataset". A search bar at the top left says "Search 23,329 datasets". To the right of the search bar are "Feedback" and "Filter" buttons. The main content area is titled "PUBLIC" and shows a list of datasets. Each dataset entry includes a thumbnail image, the dataset name, the author's name, the upload date, file size, and the number of files. The datasets listed are:

- 17K Mobile Strategy Games by Tristan (2 months ago, 8 MB, 8.2, 1 File (CSV))
- UFC-Fight historical data from 1993 to 2019 by Rajeev Warrier (4 months ago, 3 MB, 9.7, 4 Files (CSV))
- Forest Fires in Brazil by Luís Gustavo Modelli (2 months ago, 31 KB, 7.6, 1 File (CSV))
- Border Crossing Entry Data by Akhil (3 months ago, 4 MB, 8.2, 1 File (CSV))
- Food.com Recipes and Interactions by Shuyang Li (a month ago, 267 MB, 10.0, 0 Files)
- Africa Economic, Banking and Systemic Crisis Data by Chiri (4 months ago, 14 KB, 10.0, 1 File (CSV))

On the right side of the dataset list, there is a "Sort by:" dropdown set to "Hottest" and a "347" button.

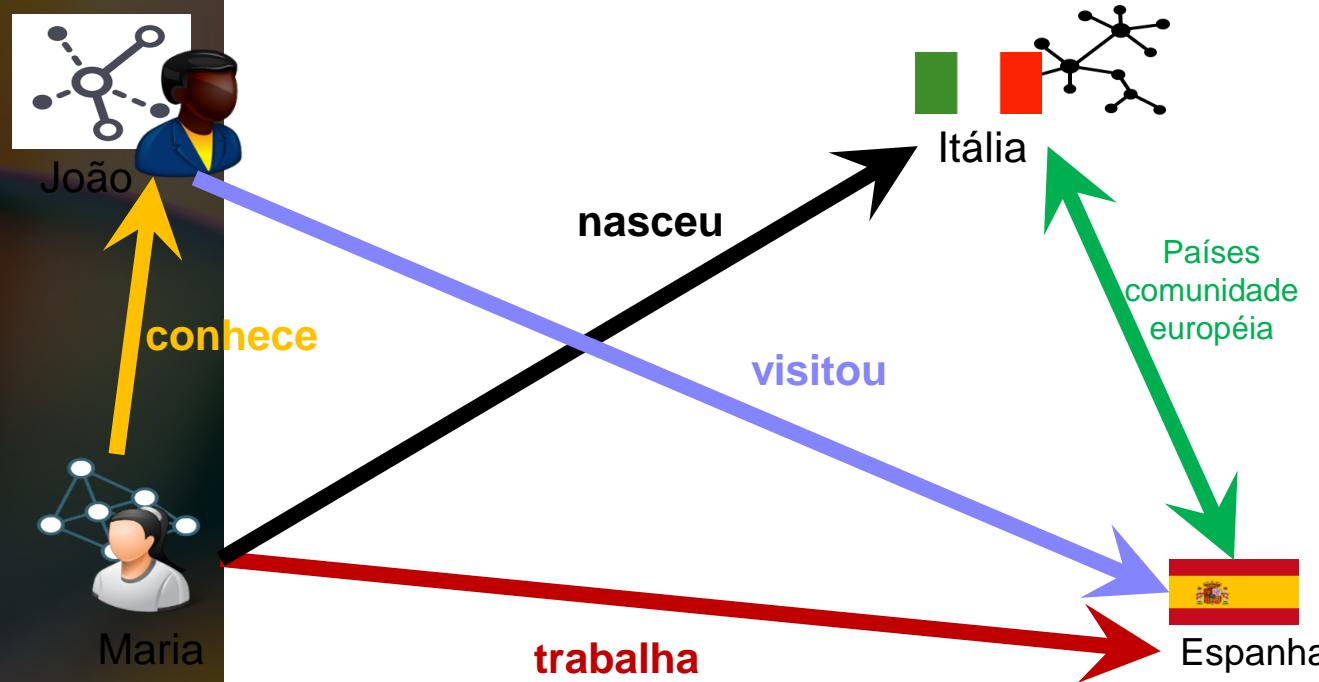
Refere-se a um estilo de se publicar e interligar dados estruturados na Web.

Descreve um conjunto de práticas para publicar e conectar dados estruturados na *web* de forma tão fácil quanto o compartilhamento de documentos.

Agrega os mesmos princípios básicos da *web* propostos por Berners-Lee: abertos, modulares e escaláveis.

Dados ligados

Ligar dados (semântica) e não documentos (estrutura).



Quanto mais um dado for interligado com outros dados maior é o seu valor e sua utilidade.

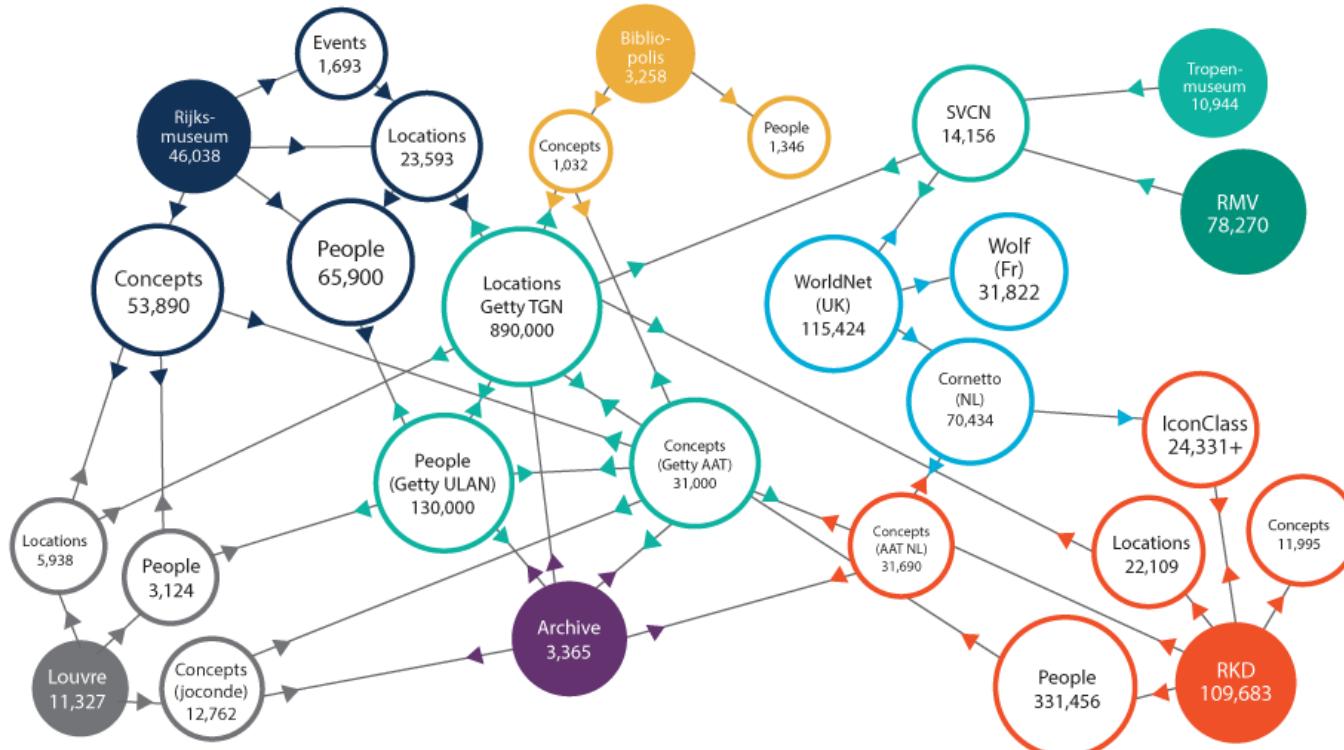


Linked Open Data (LOD), ou dados abertos ligados.



LOD é um projeto aberto comunitário mundial iniciado em 2007 e que visa à publicação de vários conjuntos de dados (*datasets*) de forma que as ligações sejam possíveis entre eles.

Dados abertos ligados

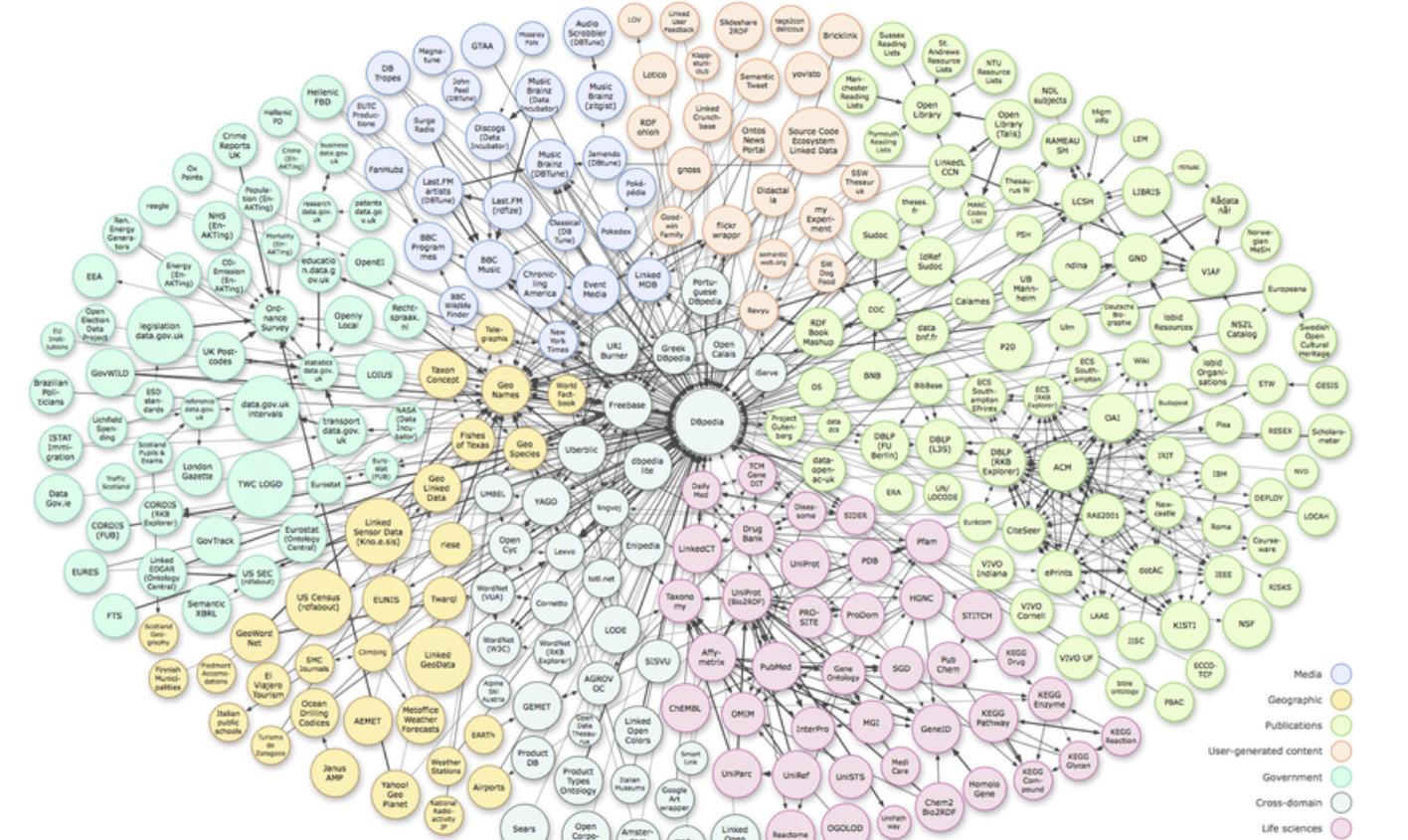


Linked open data

Exemplo no contexto Brasileiro



Nuvem do *linked open data*



Conclusão

- Entendemos o que são dados abertos.
- Entendemos o que é *linked data* e *linked open data*.
- Visualizamos alguns exemplos de dados abertos.

Próxima aula

- ❑ Compreender o que são Ontologias.

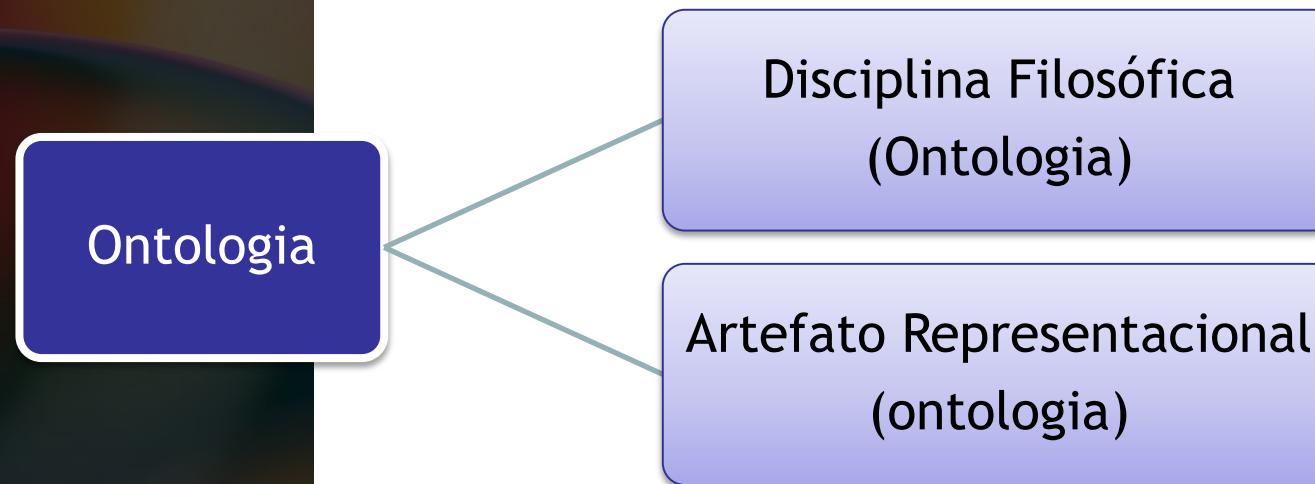


Aula 1.4.1 Ontologias (parte 1)

Nesta aula

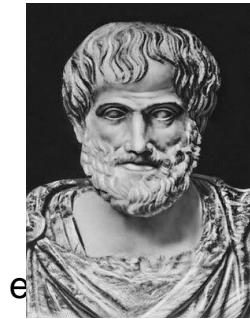
- Entender o que são ontologias.
- Conhecer os tipos de ontologias.
- Conhecer os elementos que fazem parte de uma ontologia.

- Ontologia é um termo polissêmico
 - Pesquisa: Filosofia, Ciência da Computação e Ciência da Informação
 - Domínios: Medicina, Biologia, Direito e Geografia, para citar alguns.



Ontologia como disciplina

- Origem na filosofia → “Filosofia Primeira” de Aristóteles e seus seguidores.
- Objeto de pesquisa o ser enquanto tal. Ser/Ente é tudo o que é.
- Diz respeito aquilo que existe:
 - Ontos (ser) + logia (conhecimento)
- Ciência do ser e das relações da existência.
 - Estudo da natureza e organização da realidade.



“Ontologia é a ciência do que é, dos tipos e estruturas de objetos, propriedades, e
relações em todas as áreas da realidade (SMITH, 2003).”

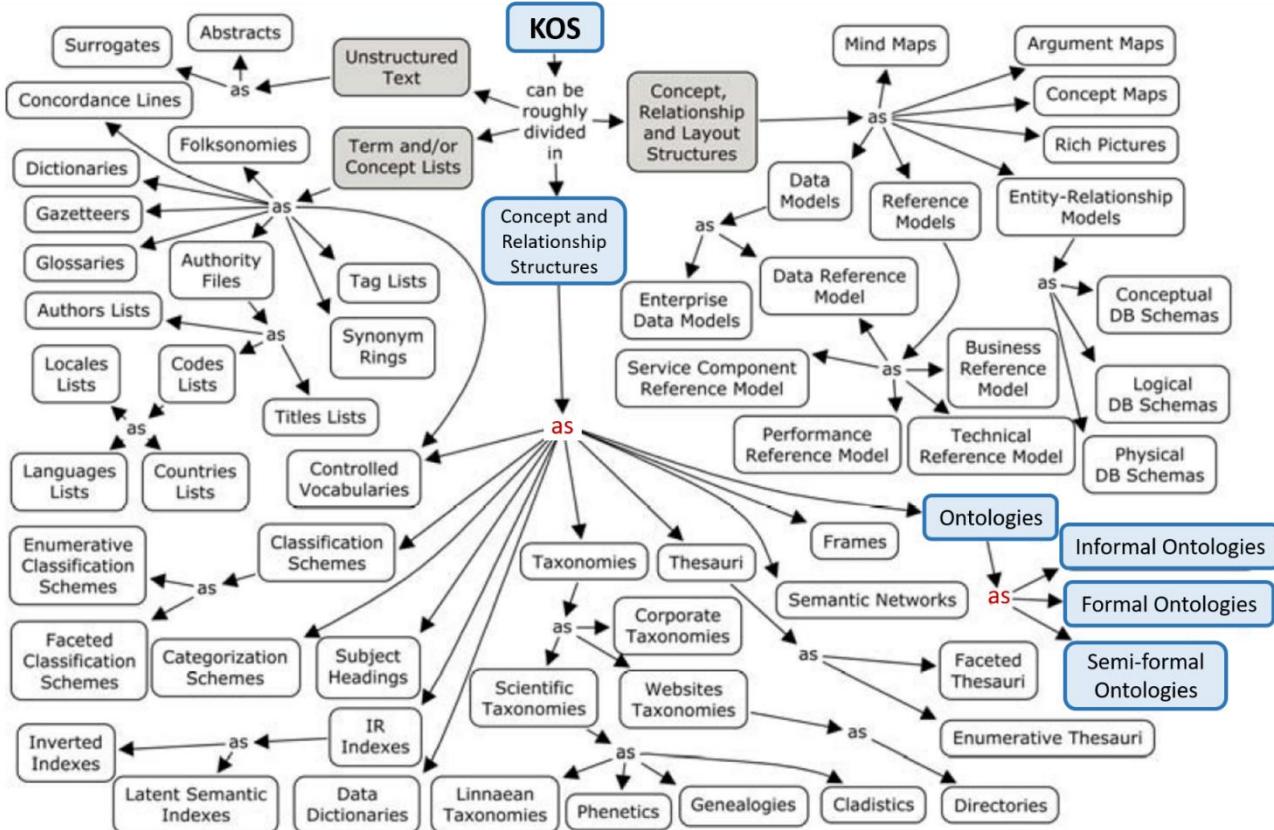
Ontologia como artefato representacional

IGTI

Ciência da
Informação

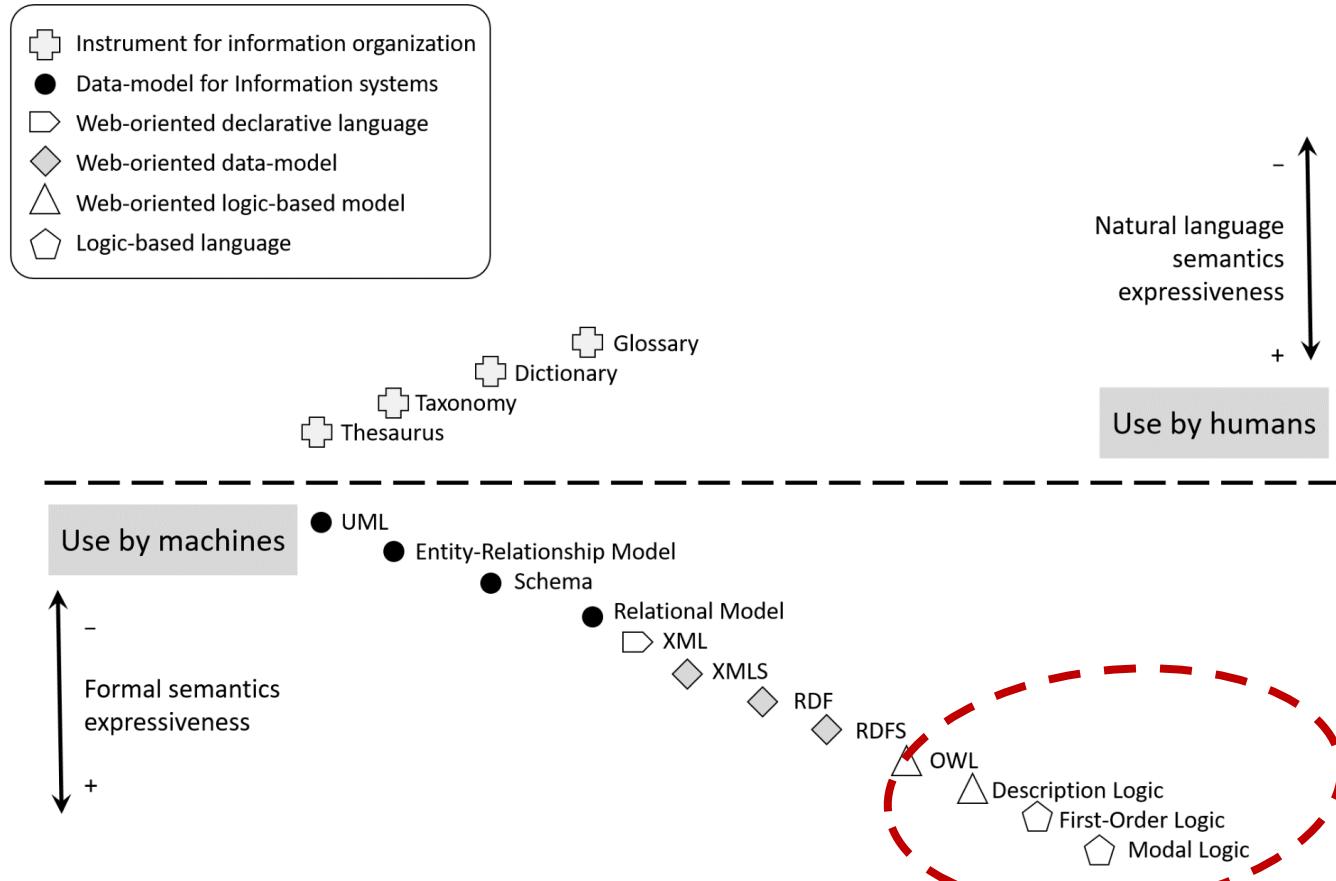


Sistemas de
organização do
conhecimento
(SOC)



- Artefato de informação que visa representar algum domínio do conhecimento.
- Ontologia usada para:
 - representação do conhecimento.
 - permitir o raciocínio computacional (Lógica descritiva/axiomas).
 - permitir a integração da informação (relações entre os entes).
 - garantir significado semântico (interoperabilidade semântica).
 - promover aplicações de processamento de linguagem natural.

Expressividade dos artefatos



Fonte: Almeida, Souza e Fonseca (2011)

Ontologias realistas e não realistas

- Não-realismo: representação de conceitos, interpretações ou representações mentais das pessoas.
- Realismo: entidades da realidade e não interpretação ou representações mentais das pessoas.

é um

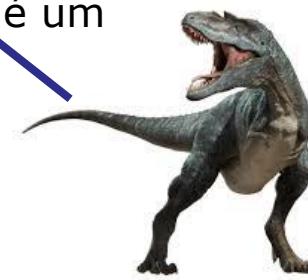


Animal

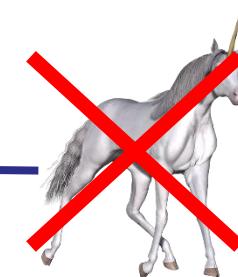
é um



é um



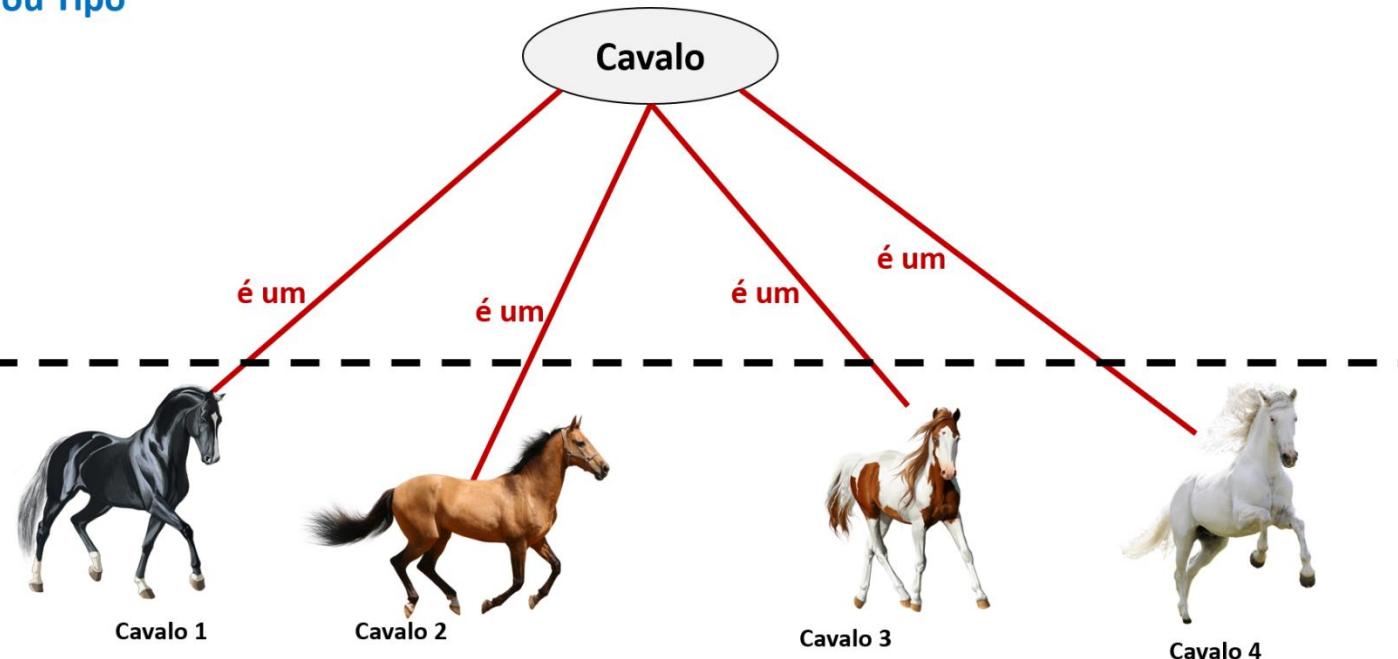
é um



- **Particulares** ou indivíduos: ocorrências únicas de algo existente na realidade.
 - Exemplo: Cada um de nós é uma única ocorrência ou indivíduo de um "homo sapiens".
- **Universais** ou tipos: entidades reais que generalizam os particulares existentes no mundo.
Existe apenas se existir pelo menos um particular desse universal.
 - Exemplo: "homo sapiens" é uma entidade geral ou universal referente aos particulares que cada um de nós é.

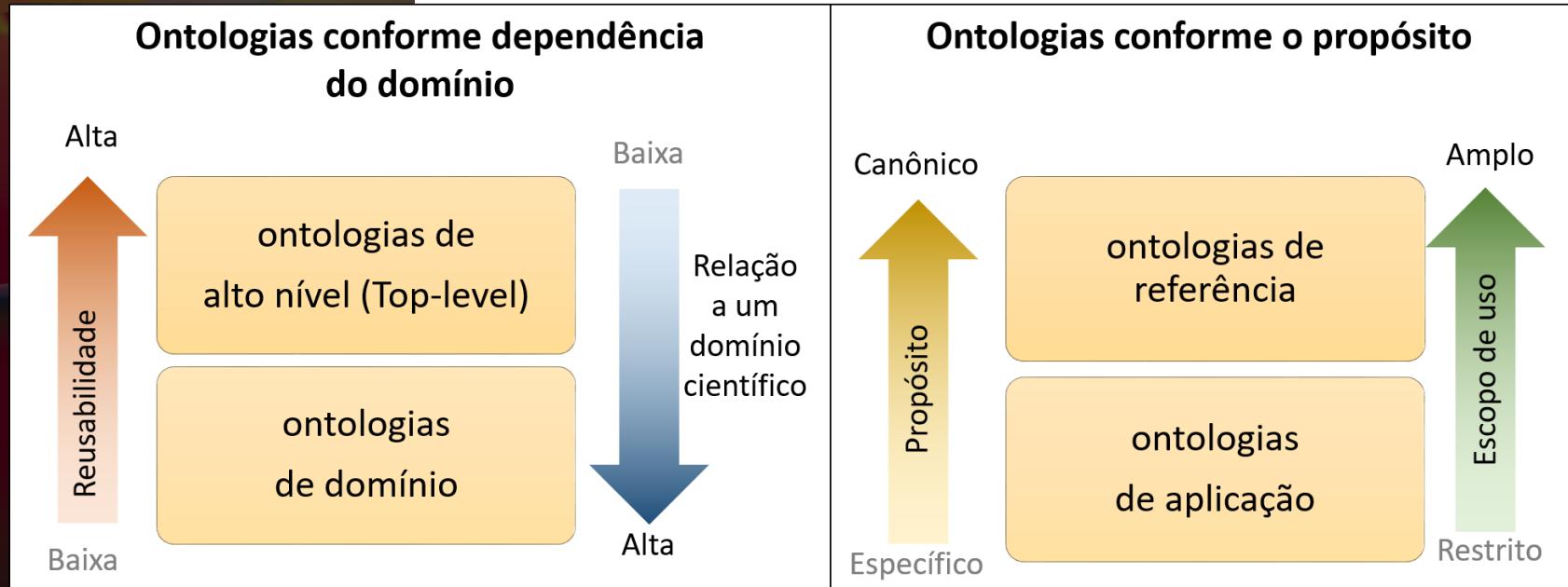
Universais e Particulares

Universal
ou Tipo



Particulares
ou Indivíduos

Classificação das ontologias



Fonte: Guarino, 1997a; Arp, Smith e Spear, 2015.

Classificação das ontologias

Baixo

↓ Acoplamento ao domínio

Alto

Ontologias de alto nível

- Ontologias neutras de domínio que lidam com termos muito gerais e básicos. Exemplo: Processo, Entidade, Qualidade.

Ontologias de referência

- Ontologias de domínio que representam termos em um determinado domínio geral. Esses termos ou entidades podem ser usados em domínios ou subdomínios específicos do domínio geral. Exemplo: Documento, Doença, Vírus.

Ontologias de domínio

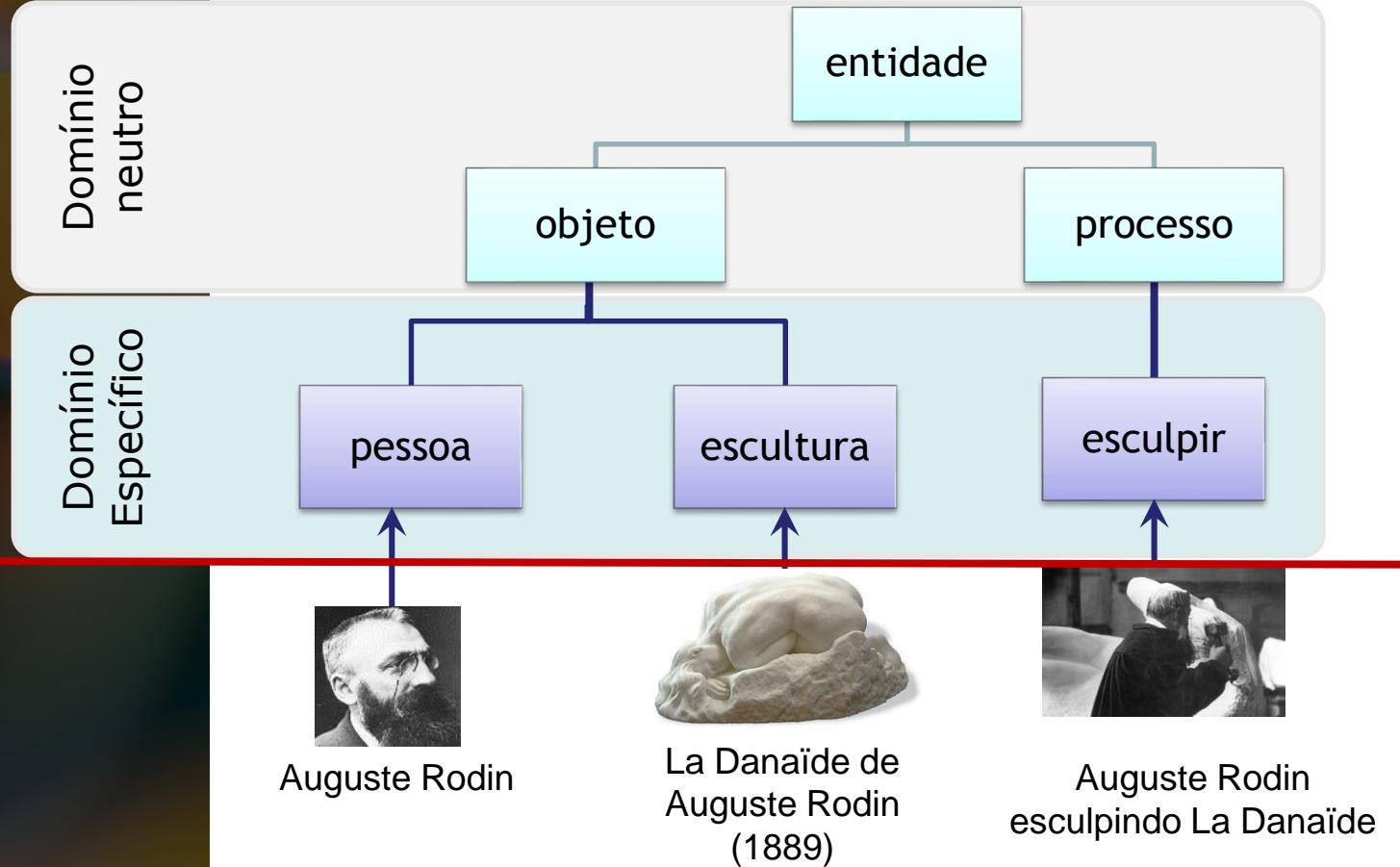
- Representam termos em um determinado domínio, que geralmente são específicos para esse domínio. Exemplo: Prontuário, Hipertensão, Zika vírus.

Particulares

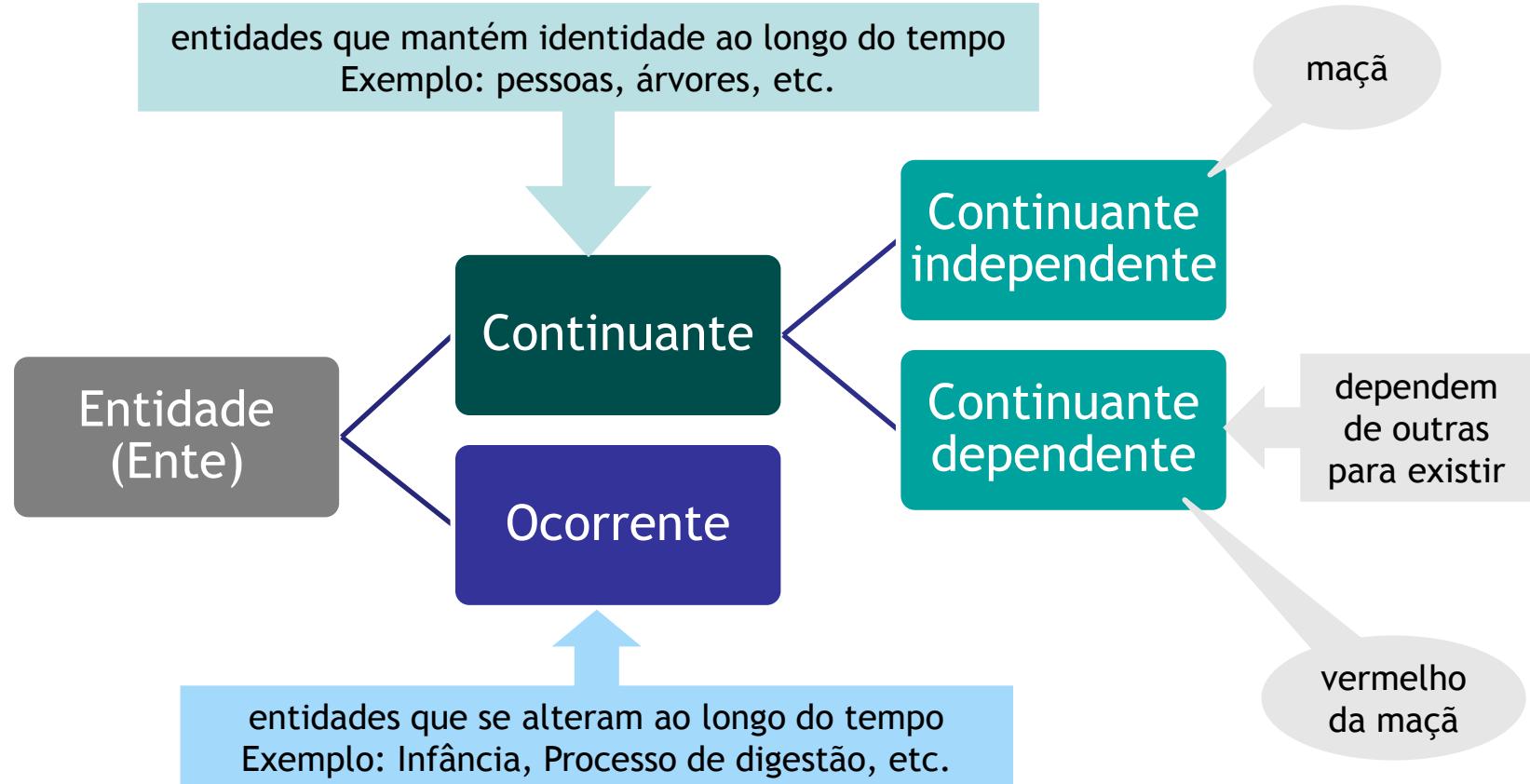
Universais

Classificação das ontologias

IGTI



Categorias básicas de modelagem

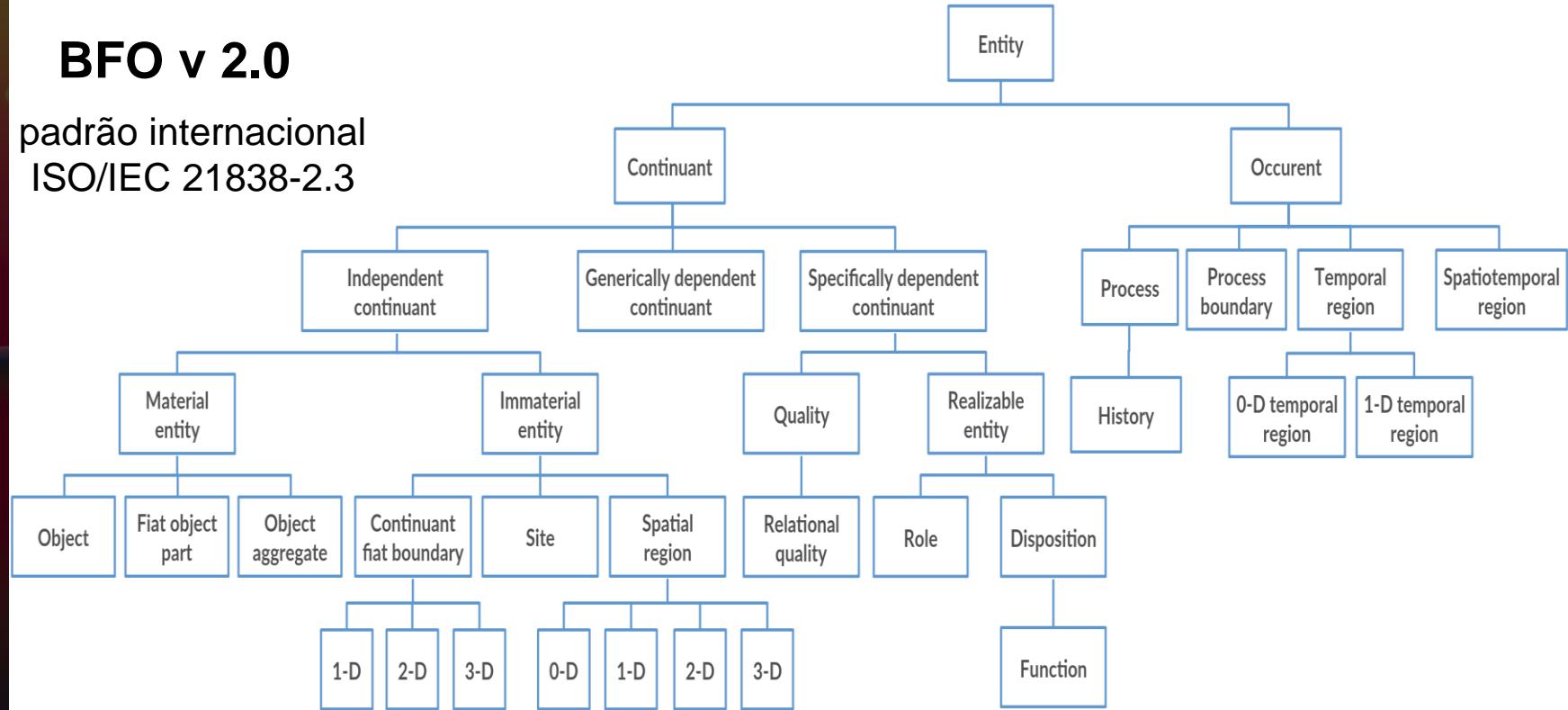


Exemplo de ontologia de alto nível

IGTI

BFO v 2.0

padrão internacional
ISO/IEC 21838-2.3



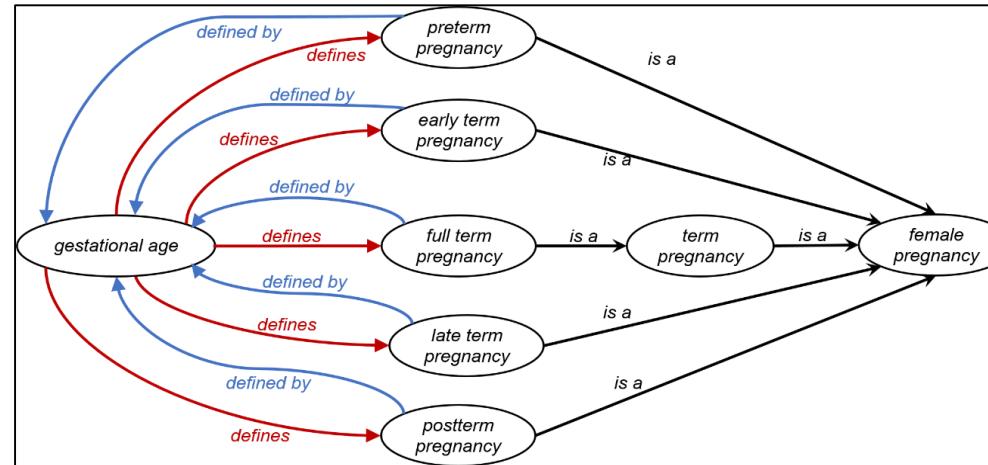
Especificação completa: <https://github.com/BFO-ontology/BFO/raw/master/docs/bfo2-reference/BFO2-Reference.pdf>

Exemplos de ontologia de referência

- Ontology for General Medical Science (OGMS)
 - Ontologia para representar o tratamento de doenças e diagnóstico.
 - Exemplos: Disease (doença), Symptom (Sintoma), diagnosis (diagnóstico), health care process (processo de cuidados de saúde).
- Ontology of Medically Related Social Entities (OMRSE)
 - Abrange o domínio de entidades sociais relacionadas à assistência à saúde.
 - Exemplos: Patient role (paciente), health care encounter (Encontro de cuidados de saúde - "consulta"), hospital facility (instalação hospitalar - "Hospital")
- Uberon multi-species anatomy ontology (UBERON)
 - Abrange anatomia dos seres vivos.

Exemplos de ontologia de domínio

- Obstetric and Neonatal Ontology (ONTONEO)
 - Representação dos dados dos prontuários eletrônicos de saúde (EHRs) envolvidos no atendimento à gestante e ao bebê.
 - Exemplo: prenatal encounter (Consulta prenatal), pregnant woman (grávida).



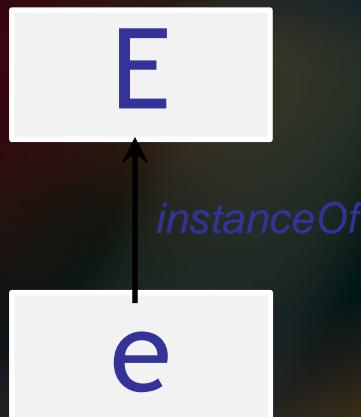
- **Entidades:** É algo que você deseja representar em um domínio particular. Qualquer coisa que exista, existiu ou irá existir. Ex: eventos, processos, objetos inanimados ou vivos, etc.
- **Classes:** Representam as entidades do domínio. O organizam as entidades de um domínio em uma taxonomia. Universais.
- **Atributos de classe:** Propriedades relevantes da classe que ajudam a descrevê-la.

- **Instância:** Representam uma unidade de objetos específicos de uma entidade, ou seja, indivíduos de um determinado universal.
- **Atributos da instância:** Essas são propriedades relevantes que descrevem as instâncias de uma entidade.
- **Relacionamento:** Descreve o tipo de interação entre duas classes, duas instâncias ou uma classe e uma instância.
- **Cardinalidade:** Uma medida do número de ocorrências de uma entidade associada a um número de ocorrências em outra.

- **Axioma:** Uma declaração ou proposição representada em um padrão lógico que é considerado verdadeiro. Restringem a interpretação e o uso das classes envolvidas na ontologia.

- Exemplo:

$$e \text{ instanceOf } E \Rightarrow \forall e \forall E \left(\text{inst}(e, E) \rightarrow p(e) \wedge u(E) \right)$$



Onde:

- e , E são variáveis para instância e classe
- inst, p , u são funções para instância, particular e universal
- o símbolo \wedge significa conjunção
- o símbolo \forall é o quantificador universal
- o símbolo \rightarrow é uma implicação

Buscadores de ontologias

- Buscadores de ontologias e termos (*search engines*):
 - Ontologias no domínio biomédico:
 - Bioportal - <https://bioportal.bioontology.org/>
 - Ontology Lookup Service - <http://www.ebi.ac.uk/ols/index>
 - Ontobee** - <http://www.ontobee.org/> (OBO Foundry)
 - Ontologias em geral:
 - Swoogle Semantic Web Search Engine - <http://swoogle.umbc.edu/2006/>
 - Watson RDF search engine - <http://watson.kmi.open.ac.uk/WatsonWUI/>
 - RDF vocabulary search and lookup - vocab.cc

Conclusão

- Entendemos o que são ontologias.
- Conhecemos os tipos de ontologias: ontologia de alto nível, ontologia de referencia e ontologia de domínio.
- Conhecer os elementos que fazem parte de uma ontologia: entidades/classes, relacionamentos, axiomas,

Próxima aula

- Conhecer metodologias e ferramentas para construção de ontologias.



Aula 1.4.2 Ontologias (parte 2)

Nesta aula

- ❑ Conhecer metodologias para construção de ontologias.
- ❑ Entender o processo de construção de ontologias.
- ❑ Conhecer a ferramenta de construção de ontologias: Protégé

Enterprise Ontology

Toronto Virtual Enterprise
(TOVE)

Methontology

101 Method

CYC Method

DILIGENTE

On-To-Knowledge Methodology

Systematic Approach for
Building Ontologies (SABiO)

Up for ONtology (UPON)

NeOn Methodology

Ontoforinfoscience

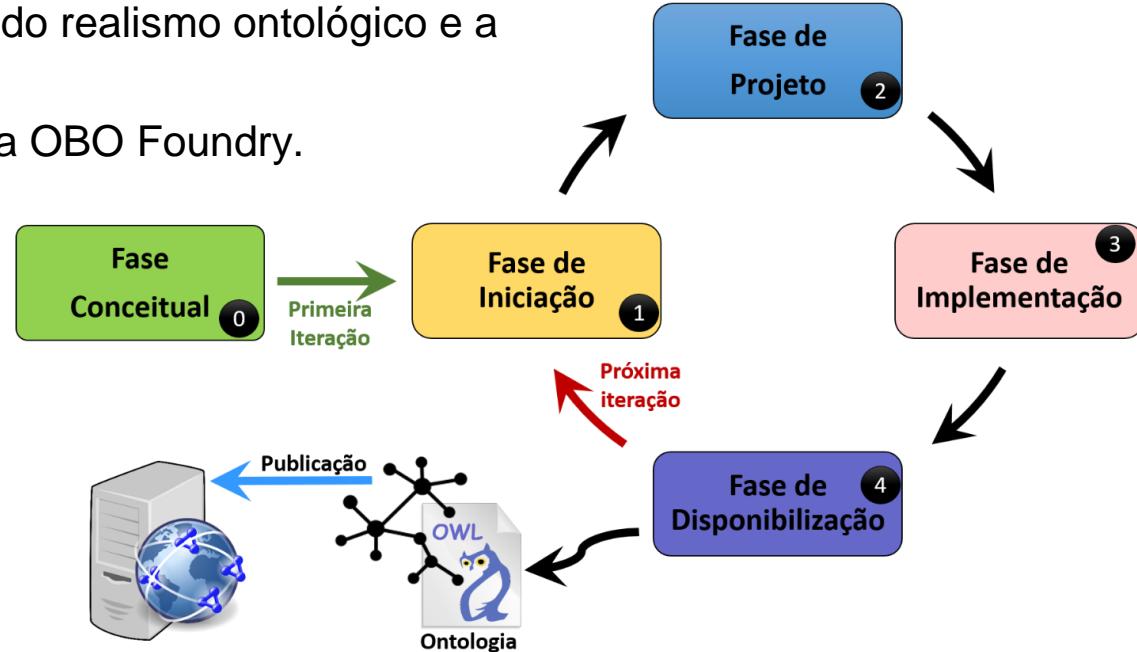
The methodology of ontological
realism

Realism-Based Ontology
engineeRing Methodology
(ReBORM)

Realism-Based Ontology engineering Methodology (ReBORM)

IGTI

- Reúne a Metodologia do realismo ontológico e a Metodologia NeOn.
- Incorpora princípios da OBO Foundry.



- Protégé: Plataforma de código aberto, desenvolvida pelo grupo de pesquisa da Stanford, que fornece um conjunto de ferramentas para construir modelos de domínio e aplicativos baseados em conhecimento com ontologias.
- Possui grande variedade de plugins para realizar visualização, consultas ou exportar a ontologia em diversas linguagens (OWL, RDF, RDFS, e XML Schema).
- Versão desktop: <http://protege.stanford.edu/products.php#desktop-protege>
- Versão webprotege: <http://webprotege.stanford.edu/>

Instalação do Protégé Desktop

- Download
 - <http://protege.stanford.edu/products.php#desktop-protege>
- Guia de Instalação em inglês:
 - https://protegewiki.stanford.edu/wiki/Install_Protege5
 - Guia para Windows:
 - https://protegewiki.stanford.edu/wiki/Instal_Protege5_Win



Interface de trabalho do Protégé

IGTI

untitled-ontology-431 (<http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431>) : [<http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431>]

File Edit View Reasoner Tools Refactor Window Help

untitled-ontology-431 (<http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431>) Search...

Annotation Properties | Property matrix | Individuals by class | Individuals matrix | OWLViz | OntoGraf | SPARQL Query | Imported Ontologies | Import | Entities | Classes | Object Properties | Data Properties | Class matrix

Ontology header:

Ontology IRI <http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431>

Ontology Version IRI e.g. <http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431/1.0.0>

Annotations +

Ontology imports | Ontology Prefixes | General class axioms

Imported ontologies:

Direct Imports +

Indirect Imports

Ontology metrics:

Metrics

Axiom	0
Logical axiom count	0
Declaration axioms count	0
Class count	0
Object property count	0
Data property count	0
Individual count	0
Annotation Property count	0
DL expressivity	AL

Class axioms

RDF/XML rendering:

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431#"
    xmlns:base="http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
    <owl:Ontology rdf:about="http://purl.obolibrary.org/obo/2017/4/untitled-ontology-431"/>
</rdf:RDF>
```

<!-- Generated by the OWL API (version 4.2.8.20170104-2310) <https://github.com/owlcs/owlapi> -->

To use the reasoner click Reasoner > Start reasoner Show Inferences

Dica para construção de ontologias

- Definir parâmetros para a implementação da ontologia:
 - Namespace, prefixo, IRI, etc.
 - [Prefix.cc](#) → serviço de busca por prefixos de URI e namespaces.
 - OBO Foundry: A IRI base extende: <http://purl.obolibrary.org/obo/>
 - Ontologia de alto nível: BFO v2 (padrão ISO)
 - Utilizando o Protégé, importar a BFO para iniciar o projeto.
- Configurar o projeto no Protégé com estas definições.

OBS: Atividade “Especificação da arquitetura da ontologia” da Fase de Projeto.

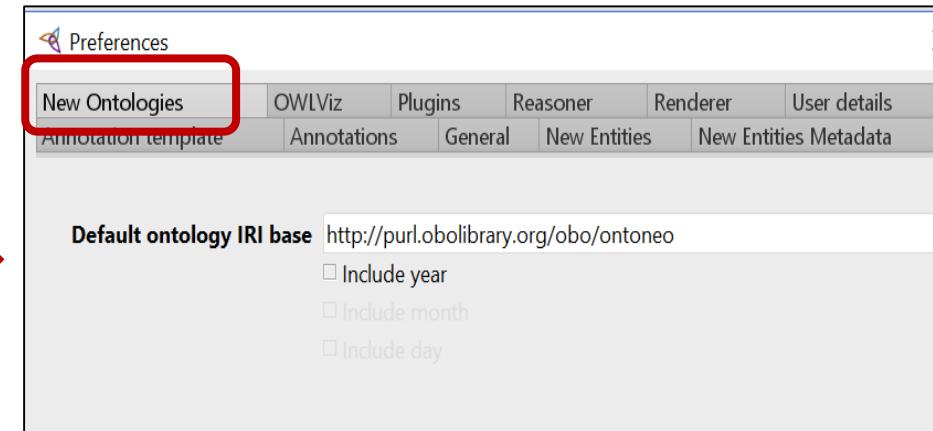
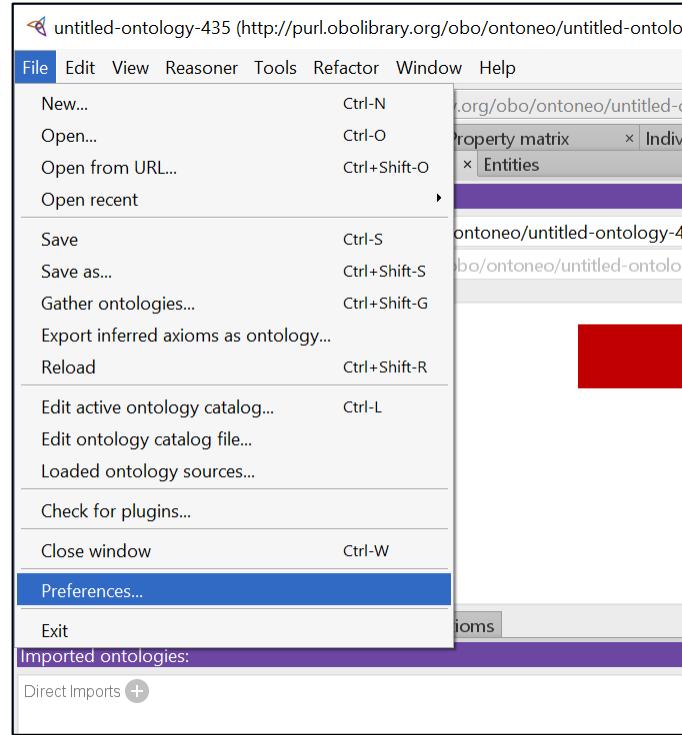
Configuração do ambiente de trabalho Protégé

- Definir URI/IRI da ontologia e de versionamento da ontologia.

The screenshot shows the Protégé ontology editor interface. At the top, the title bar displays "ontoneo (http://purl.obolibrary.org/obo/ontoneo/release/2017-30-05/ontoneo.owl) : [http://purl.obolibrary.org/obo/ontoneo/untitled-ontology-435]". The menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. Below the menu is a toolbar with back, forward, and search buttons. The main window has tabs for Active Ontology, Entities, Classes, Object Properties, Data Properties, Annotation Properties, Class matrix, Property matrix, Individuals by class, and Individuals by individual. The Active Ontology tab is selected. A purple header bar contains the text "Ontology header:" and icons for metrics and export. A red box highlights the "Ontology IRI" and "Ontology Version IRI" fields. The "Ontology IRI" field contains "http://purl.obolibrary.org/obo/ontoneo/ontoneo.owl" and the "Ontology Version IRI" field contains "http://purl.obolibrary.org/obo/ontoneo/release/2017-30-05/ontoneo.owl". To the right, a sidebar titled "Metrics" lists Axiom, Logical axiom, Declaration, Class count, and Object properties. At the bottom left, there is an "Annotations" button with a plus sign.

Configuração do ambiente de trabalho Protégé

- Definir URI/IRI base para os elementos da ontologia.



Conclusão

- Conhecemos metodologias para construção de ontologias.
- Conhecemos a ferramenta Protégé para construção de ontologias e como fazer sua configuração para iniciar um projeto.

Próxima Aula

- Prática para construção de ontologia utilizando o Protégé.



Aula 1.4.3 Ontologias (parte 3)

Nesta aula

- Entender como utilizar a ferramenta Protégé para construção de ontologias.

Criando uma ontologia



IGTI

The screenshot shows the OntoGraf interface with the following details:

- File Edit View Reasoner Tools Refactor Window Help**
- Active Ontology**: ontoneo_letras (http://purl.obolibrary.org/obo/ontoneo/release/v1/ontoneo_letras.owl)
- Annotations**: Usage
- Annotations: 'Homo sapiens'**
- Object restriction creator** dialog open for 'Homo sapiens':
 - Restricted property**: asserted 'SubClass Of' for 'organism'.
 - Description**: 'Homo sapiens'
 - Equivalent To**: none
 - SubClass Of**: asserted 'SubClass Of' for 'organism'.
 - General class axioms**: none
 - Instances**: none
 - Target for Key**: none
 - Disjoint With**: none
 - Restriction type**: Some (existential) Cardinality 1
- OWLviz** tab selected.

Conclusão

- Conhecemos um pouco mais a ferramenta Protégé.
- Entendemos como utilizar a ferramenta Protégé para construção de ontologias.

Próxima Aula

- ❑ Compreender o que é Web Mining.



Aula 1.5. Web Mining

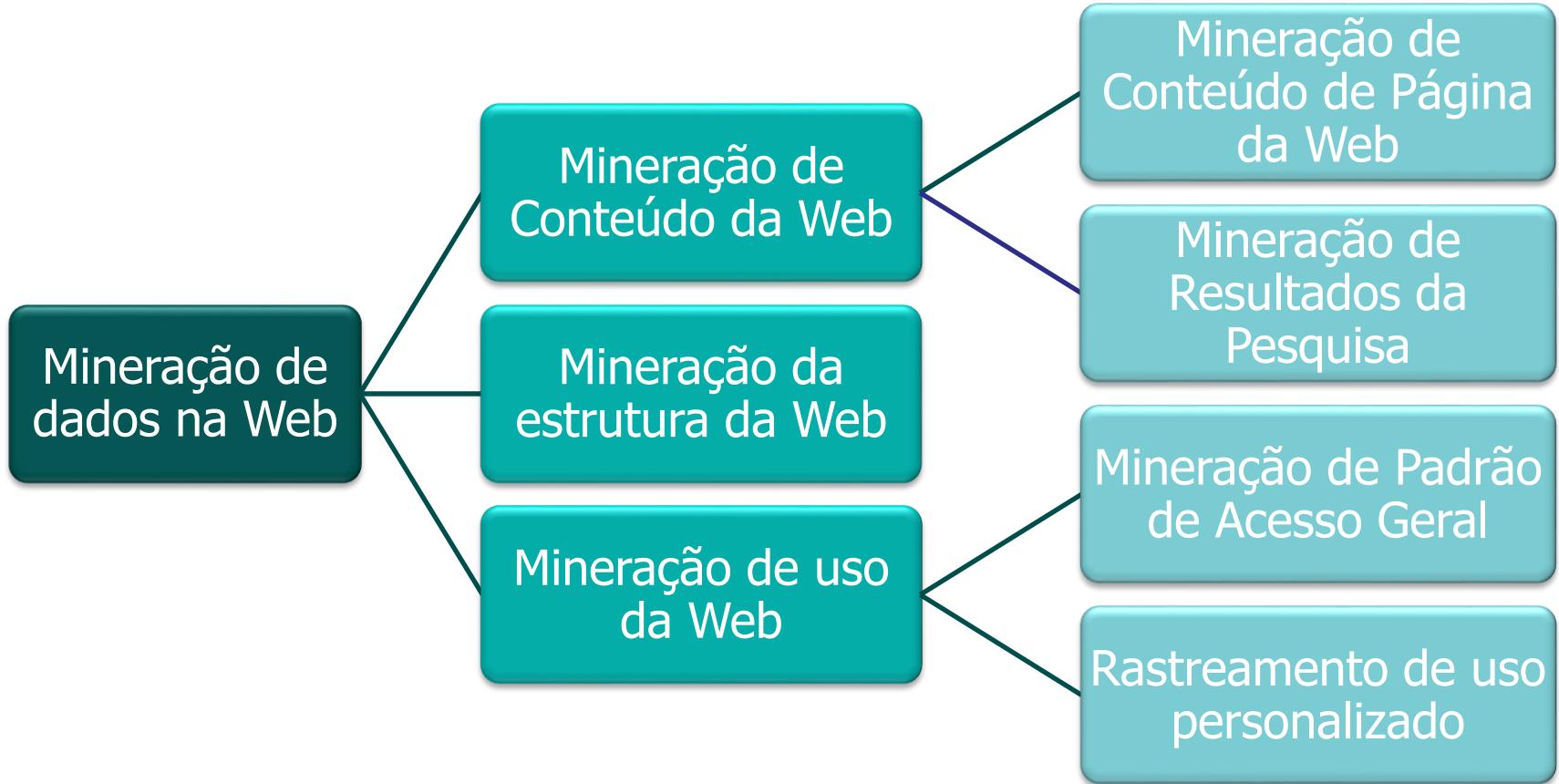
Nesta aula

- Entender o que é Web Mining ou mineração da web.

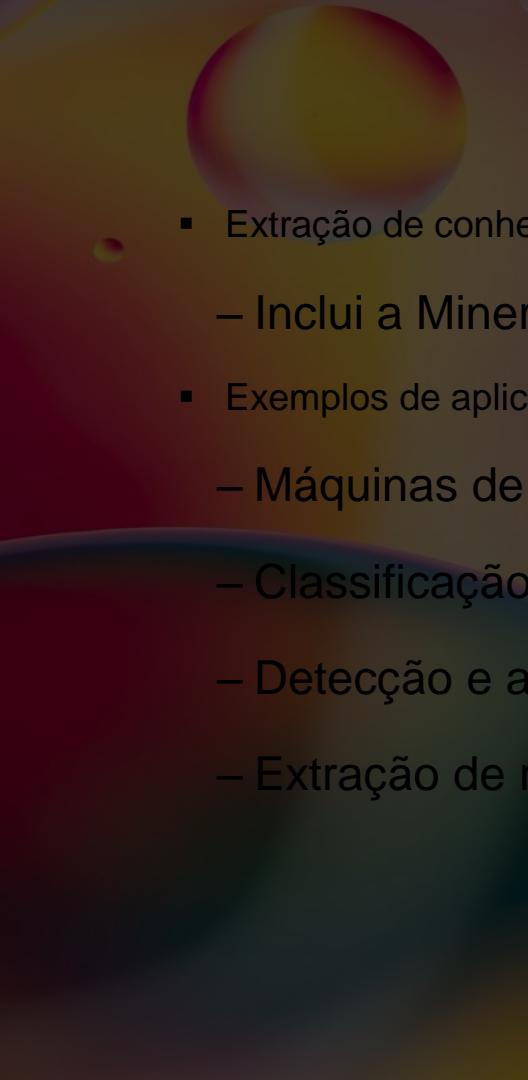
Web Mining

- Aplicação de técnicas de Mineração de Dados para descoberta de padrões na Web.
- Motivações:
 - Encontrar informação relevante
 - Gerar conhecimento a partir da informação disponível na Web
 - Personalizar a informação
 - Aprender sobre consumidores ou usuários individuais

Taxonomia do Web Mining



- Mineração de conteúdo (*Web content mining*)
 - Extração de conhecimento do conteúdo de páginas e suas descrições
- Mineração de estrutura (*Web structure mining*)
 - Obtenção de conhecimento a partir da organização da web e da referência cruzada de ligações.
- Mineração de uso (*Web usage mining*)
 - Geração de padrões interessantes com o uso dos registros de acesso da web



■ Mineração de conteúdo

- Extração de conhecimento do conteúdo de páginas e suas descrições
 - Inclui a Mineração de texto (text mining)
- Exemplos de aplicação:
 - Máquinas de busca (Google, Bing, etc)
 - Classificação de textos
 - Detecção e acompanhamento de evento
 - Extração de regras

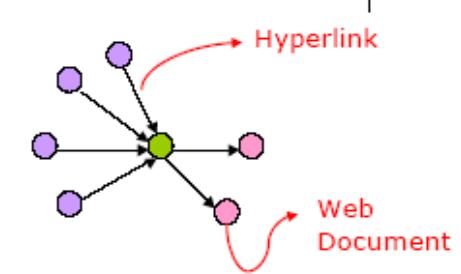


■ Mineração de estrutura

- Tenta descobrir o modelo por trás da estrutura de links na web.
- Exemplo de informação extraída:
 - Links apontando para um documento podem indicar sua popularidade
 - Links saindo de um documento podem indicar a riqueza ou variedade de tópicos tratados pelo documento

■ Mineração de estrutura

- Tenta descobrir o modelo por trás da estrutura de links na web.
- Informação coletada e analisada:
 - Autoridade de uma página.
 - Ranking de páginas.
 - Número de exibições ou retorno em buscas
 - Links apontando para um documento (indicar a variedade de tópicos tratados pelo documento)
 - Links saindo de um documento (indicar a riqueza ou variedade de tópicos tratados pelo documento)



- Visa descobrir padrões de uso a partir de registros de acesso à web.
- Fontes de dados para análises:
 - Web server log
 - Cookies
 - Dados de consultas
 - Dados do proxy

- Usado para:

- Extração de padrão geral de acesso: Analisa dados do web log file e outras fontes para descobrir padrões e tendências de acesso.
 - Aplicação:
 - Localizar pontos para propaganda mais efetiva
 - Direcionar campanhas específicas para usuários específicos
 - Customização ou personalização: Analisa tendências de usuários individuais.
 - Aplicação:
 - Personalizar dinamicamente a cada usuário, com base no padrão de acesso ao longo do tempo.

Conclusão

- Entendemos o que é o web mining.

Próxima aula

- ❑ Conhecer o que é o text mining.



Aula 1.6.1. Text Mining (Parte 1)

Nesta aula

- ❑ Entender o que é Mineração de Texto (Text Mining).

Mineração de texto (text mining)

- Mineração de dados em texto visa extrair informação relevante de uma grande base de dados textuais sem precisar de intervenção humana na leitura de cada documento
- É um processo que utiliza métodos para navegar, organizar, encontrar e descobrir informação escrita na linguagem natural.

Text mining makes information extraction from huge volumes of data easier and structures the information as important facts, key terms or persons.



- Busca de informações específicas em documentos para melhor compreensão de textos disponíveis em documentos.
- Foco em analisar dados não estruturados:
 - Redes Sociais.
 - Blogs.
 - Microblogs.
 - Compartilhadores de Vídeo.
 - Conteúdo online (New York Times, Folha, Veja, etc)

Exemplo de dado do New York Times

'Diego Maradona, once the sensation of the soccer world, was on the cover of one of Argentina's most prominent political magazines, *Noticias*, last week. He looked vastly overweight, his hair was dyed orange, and there was a huge tattoo of Che Guevara on his right arm. Beside him was the headline, ''Therapy Cuba Style.'' In recent days, Mr. Maradona, now 39, has made some of the biggest headlines of his long soap-opera career. His choice of Havana for his recovery from heart problems and cocaine addiction is viewed as something of a propaganda victory for Fidel Castro's government and its claims of superior medical services. His newest troubles have prompted a new anti-drug publicity campaign by the government here.',

Diferentes formas de mencionar um mesmo ente

'Diego Maradona, once the sensation of the soccer world, was on the cover of one of Argentina's most prominent political magazines, Noticias, last week. He looked vastly overweight, his hair was dyed orange, and there was a huge tattoo of Che Guevara on his right arm. Beside him was the headline, ''Therapy Cuba Style.'' In recent days Mr. Maradona, now 39, has made some of the biggest headlines of his long soap-opera career. His choice of Havana for his recovery from heart problems and cocaine addiction is viewed as something of a propaganda victory for Fidel Castro's government and its claims of superior medical services. His newest troubles have prompted a new anti-drug publicity campaign by the government here.',

Características

'Diego Maradona, once the sensation of the soccer world, was on the cover of one of Argentina's most prominent political magazines, Noticias, last week. He looked vastly overweight, his hair was dyed orange, and there was a huge tattoo of Che Guevara on his right arm. Beside him was the headline, ''Therapy Cuba Style.'' In recent days, Mr. Maradona now 39, has made some of the biggest headlines of his long soap-opera career. His choice of Havana for his recovery from heart problems and cocaine addiction is viewed as something of a propaganda victory for Fidel Castro's government and its claims of superior medical services. His newest troubles have prompted a new anti-drug publicity campaign by the government here.',

Compreensão de sentido

'Diego Maradona, once the sensation of the soccer world, was on the cover of one of Argentina's most prominent political magazines, Noticias, last week. He looked vastly overweight, his hair was dyed orange, and there was a huge tattoo of Che Guevara on his right arm. Beside him was the headline, ''Therapy Cuba Style.'' In recent days, Mr. Maradona now 39 has made some of the biggest headlines of his long soap-opera career. His choice of Havana for his recovery from heart problems and cocaine addiction is viewed as something of a propaganda victory for Fidel Castro's government and its claims of superior medical services. His newest troubles have prompted a new anti-drug publicity campaign by the government here.',

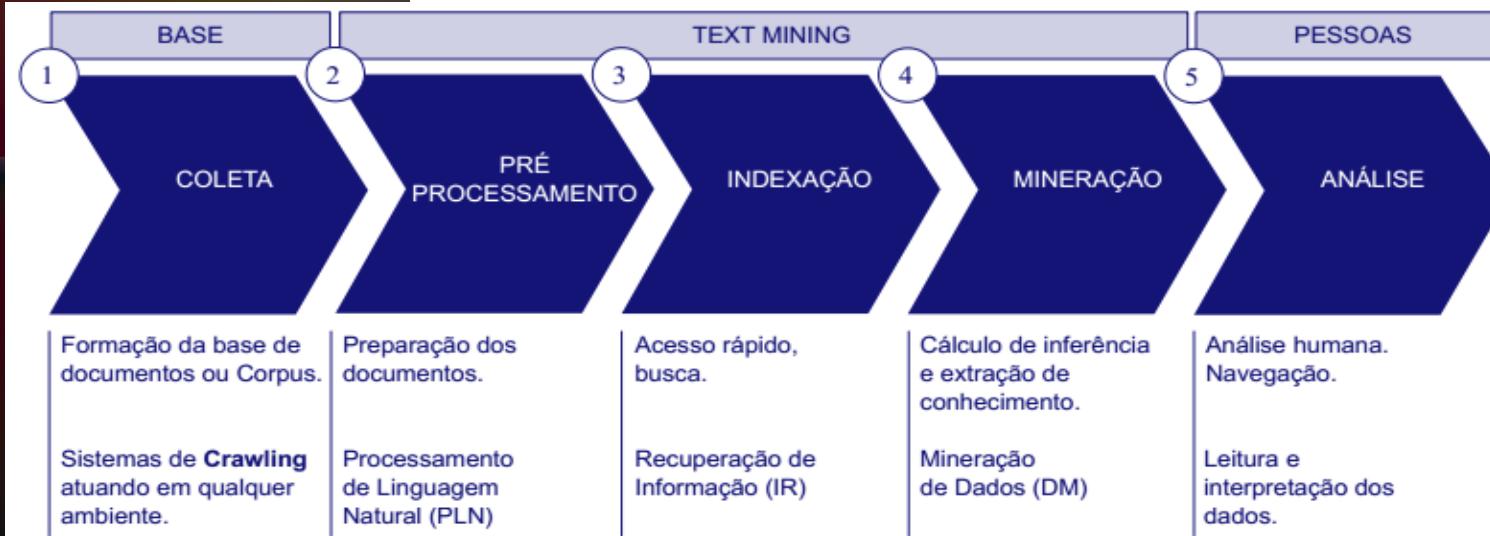


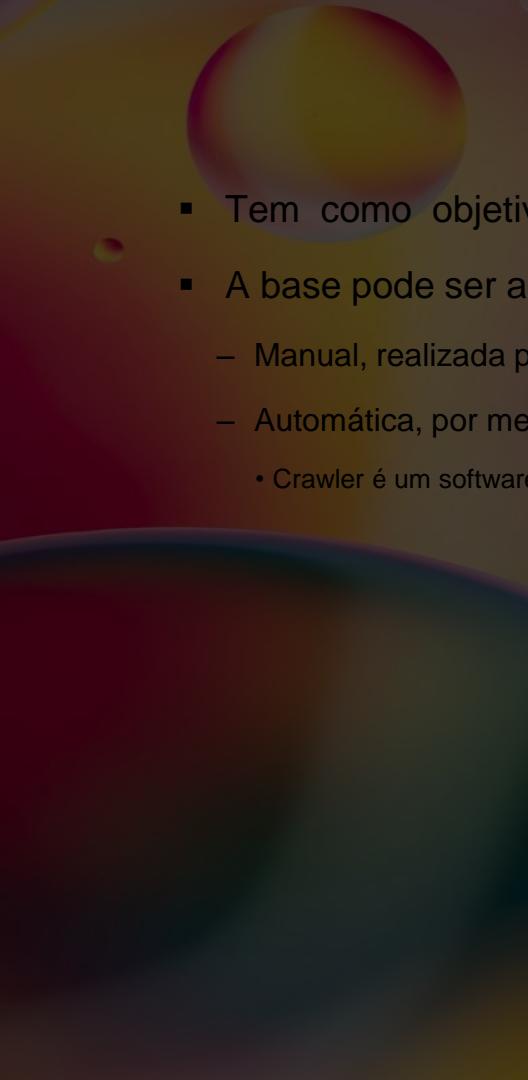
Tipos de abordagens de análise de texto

- **Análise Semântica:** envolve conhecimento morfológico, sintático, semântico, pragmático, do discurso e do mundo.
- **Análise Estatística:** envolve aprendizado estatístico a partir de dados, que normalmente inclui as etapas de codificação dos dados, estimativa dos dados e modelos de representação de documentos.

Processo de descoberta de conhecimento em textos

- *knowledge discovery in texts (KDT)*





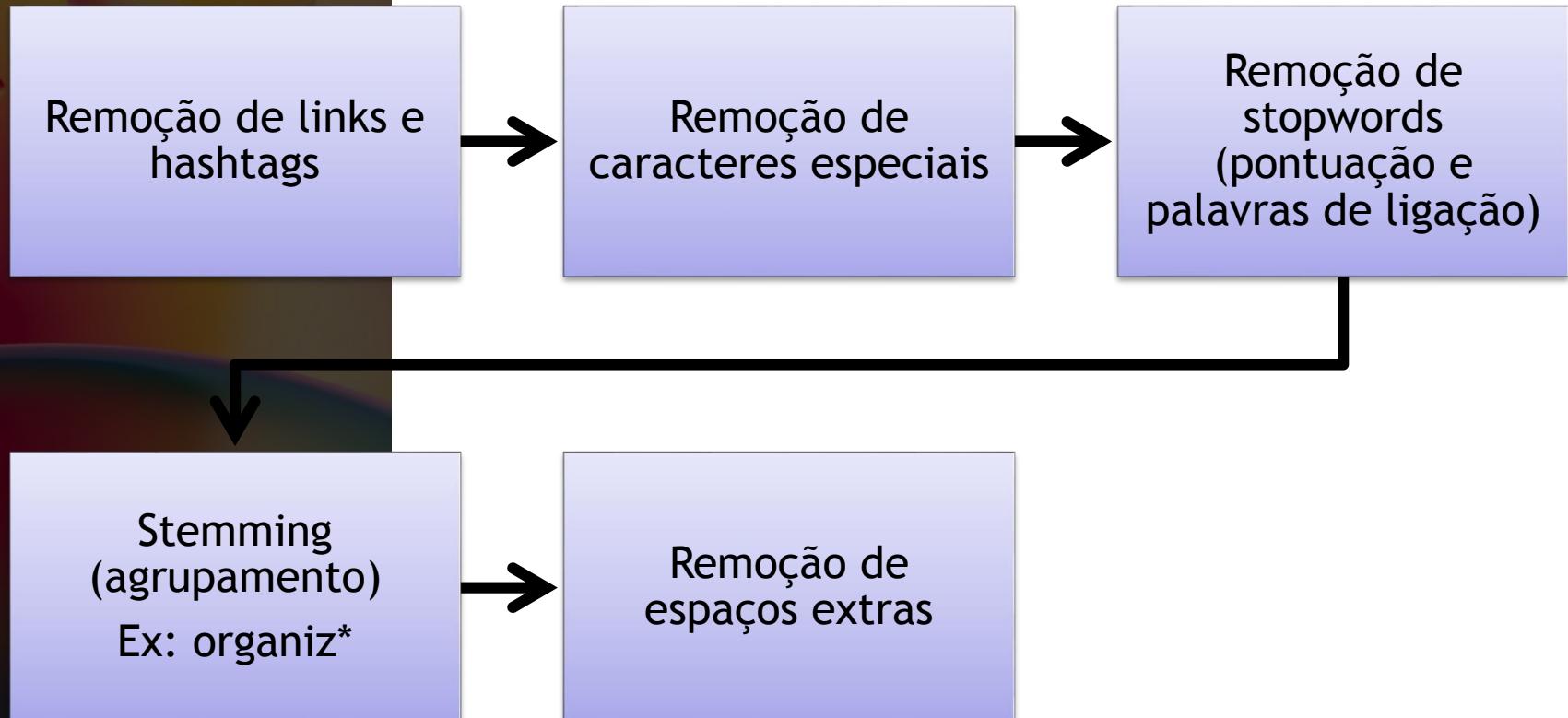
Coleta de dados

- Tem como objetivo formar a base de documentos (corpus) que irá ser analisada.
- A base pode ser adquirida de forma
 - Manual, realizada por um ser humano
 - Automática, por meio de um crawler
 - Crawler é um software que percorre sítios da internet com o intuito de coletar automaticamente os dados destes

Pré-Processamento

- Consiste em um conjunto de transformações realizadas no texto:
 - Limpeza do texto
 - dividir o texto em palavras (Tokenização)
 - Classificar palavras segundo a classe gramatical
 - *POS-tagging*
 - aplicar técnicas de *stemming*
 - remover as *stopwords*
 - Objetiva transformar os textos em uma representação estruturada adequada

Limpeza do texto



Lista reduzida de termos

Próxima aula

- ❑ Conhecer algumas técnicas de Mineração de Texto.



Aula 1.6.2. Text Mining (Parte 2)

Nesta aula

- ❑ Conhecer algumas técnicas de Mineração de Texto.

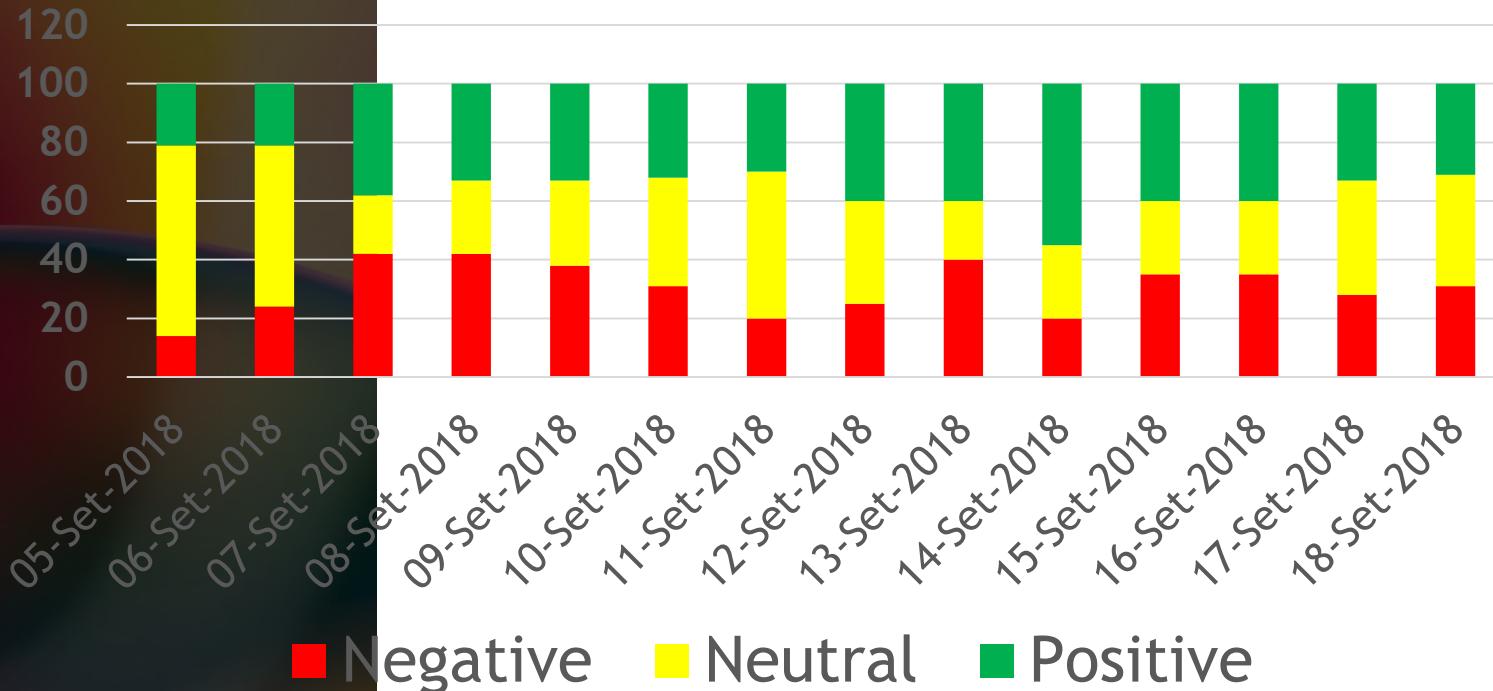
- Tipo de análise que visa estudar opiniões, sentimentos, avaliações e emoções que possam ser expressas em forma de texto.
- O foco principal é relacionar as emoções ao assunto (ou assuntos) de que se trata o documento, o que equivale a minerar suas opiniões.

Análise de Sentimento

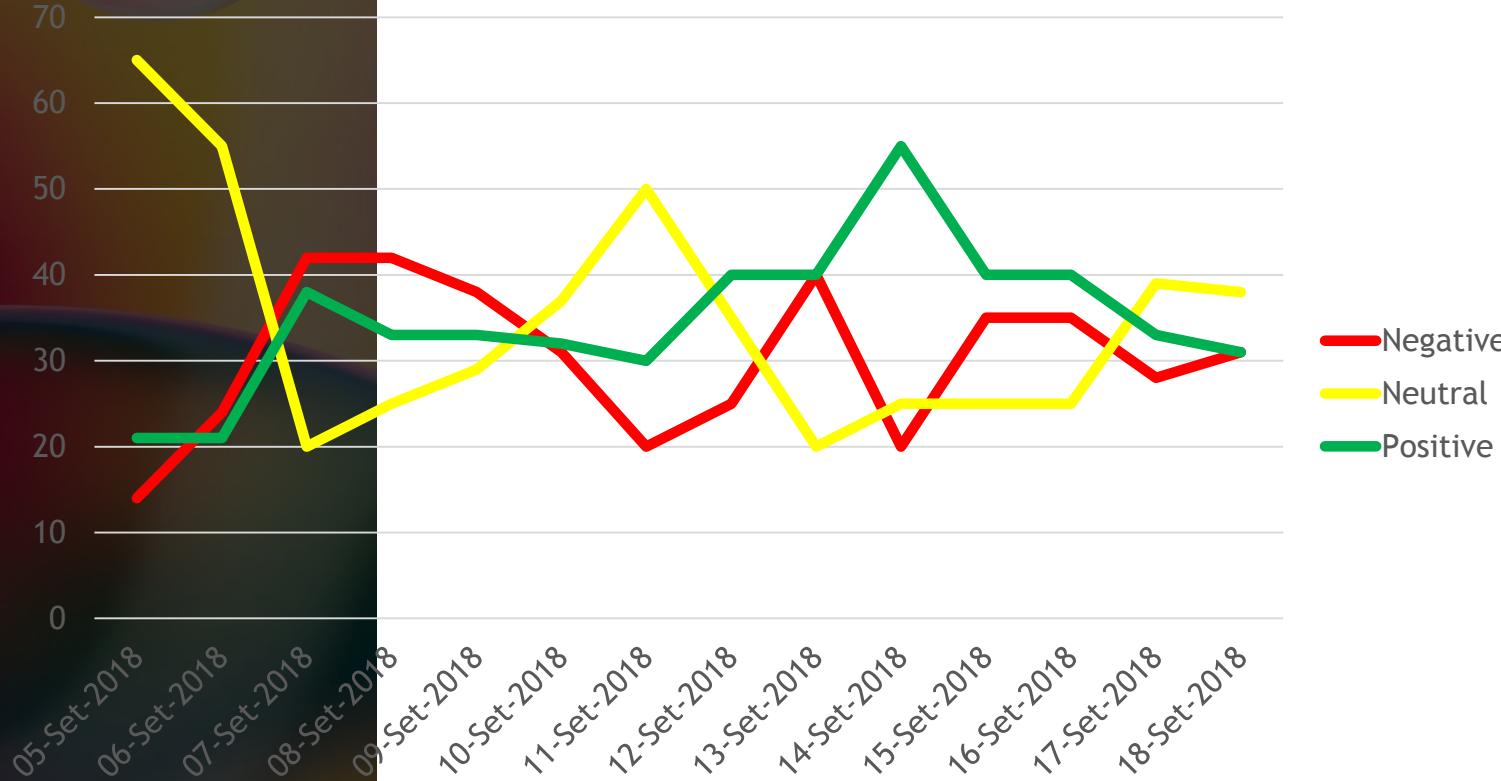
- Modelo de análise de opinião que trabalha com duas classes principais.
 - Positiva e negativa
- Em algumas análise pode-se incluir a classe neutra.



Análise de Sentimento no caso Marielle Franco



Outro modelo de visualização



- *Latent Dirichlet Allocation* (LDA) é o método padrão para modelagem de tópicos.
- Leva em consideração que:
 - Cada documento é uma mistura de tópicos.
 - Cada tópico é uma mistura de termos.
- Reorganiza a distribuição de tópicos nos documentos e a distribuição de palavras-chave nos tópicos para obter uma boa composição da distribuição de palavras-chave de tópico.

Latent Dirichlet Allocation (LDA)

IGTI

Tópicos

gene 0.04
dna 0.02
genetic 0.01
...

life 0.02
evolve 0.01
organism 0.01
...

brain 0.04
neuron 0.02
nerve 0.01
...

data 0.02
number 0.02
computer 0.01
...

Documentos

Seeking Life's Bare (Genetic) Necessities

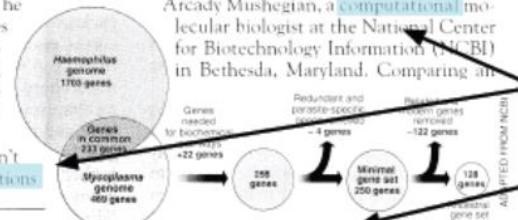
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson, a乌普萨拉 University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains

Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing all

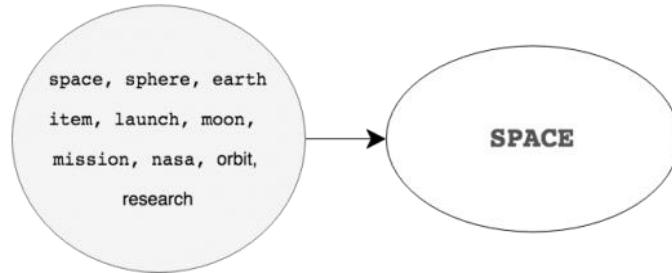
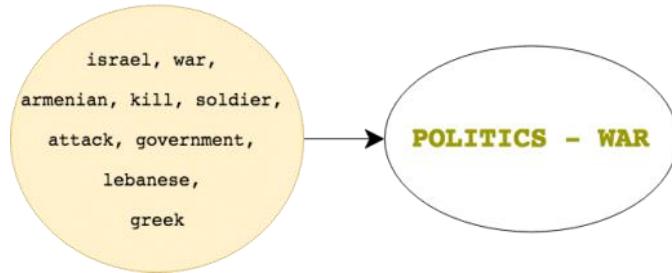
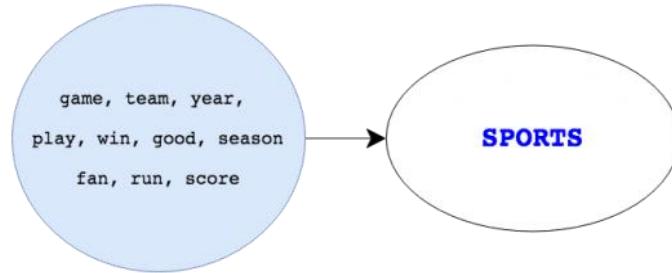
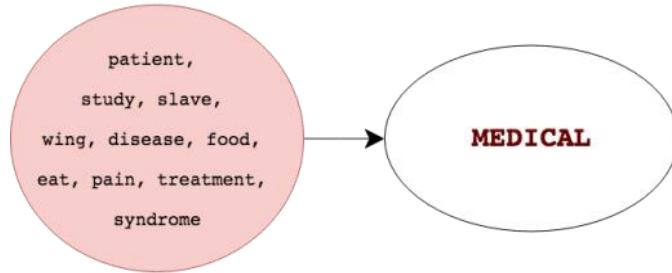
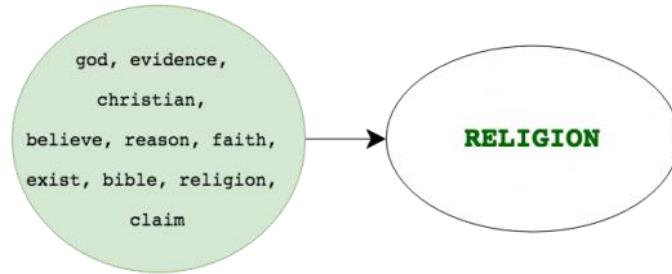
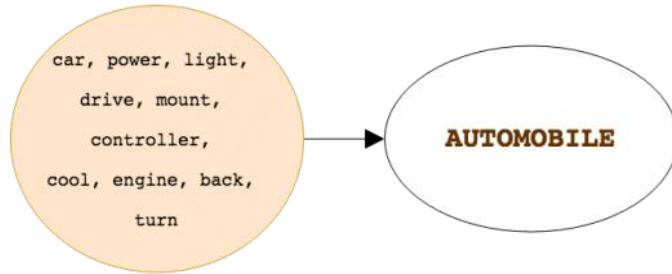


Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

Proporções da base de treinamento



Inferindo tópicos a partir das palavras chave



Bag of Words (BOW)

- É uma representação simplificada usada no processamento de linguagem natural (PLN) e na recuperação de informações (RI).
- Neste modelo, um texto (como uma frase ou um documento) é representado como o saco (*bag*) de suas palavras, desconsiderando a gramática e mesmo a ordem das palavras, mas mantendo a multiplicidade.
- Um uso clássico do BoW é para filtragem de spam.

Exemplo de Bag of Words

- Considere os seguintes documentos:

José gosta de assistir filmes. Maria também gosta de filmes.

José também gosta de assistir jogos de futebol.

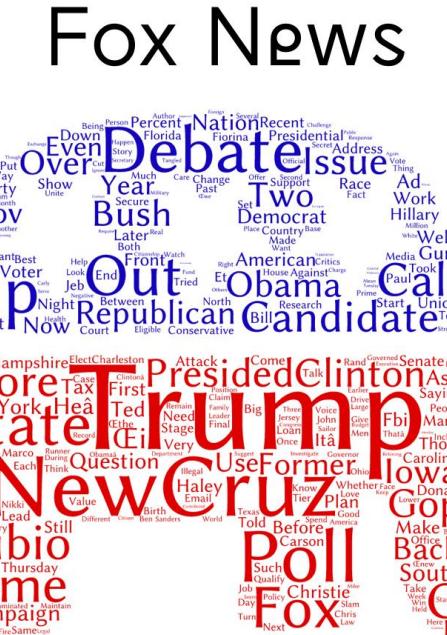
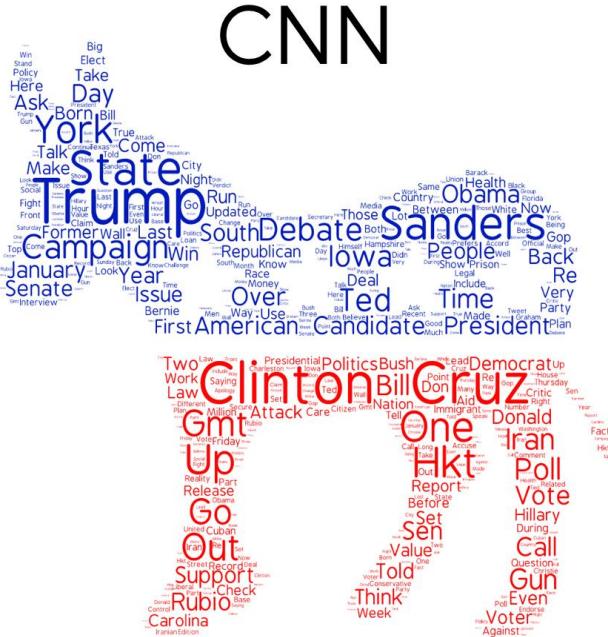
- Cria-se, uma lista com as palavras que aparece no texto dos documentos.
- Calcula-se a frequência de aparecimento das palavras.

Palavras	Frequencia
"José" ,	2
"gosta" ,	3
"de" ,	4
"assistir" ,	2
"filmes" ,	2
"Maria" ,	1
"também" ,	2
"futebol" ,	1
"jogos"	1

Bag of Words – Modelo de visualização

igt

- #### ▪ Cloud of words (Nuvem de palavras)



Palavras usadas pela imprensa durante a campanha eleitoral Norte Americana de 2016

Conclusão

- Entendemos o que é a mineração de texto.
- Conhecemos algumas técnicas usadas.

Próxima aula

- Conhecer algumas ferramentas e tecnologias para coleta de dados.
- Aprender como preparar o ambiente para realizar as práticas de coleta de dados.



Aula 1.7.1. Tecnologias usadas para coleta de dados

Nesta Aula

- Conhecer a Linguagem R e o R Studio.
- Conhecer a Linguagem Python.
- Conhecer o Framework Anaconda.
- Conhecer a plataforma Knime Analytics.

- R é uma linguagem de programação multi-paradigma (com ênfase em programação funcional), dinâmica, fracamente tipada, voltada à manipulação, análise e visualização de dados
- Utilizada para coleta, manipulação, análise e visualização de dados.
- Possui pacotes que lidam com técnicas estatísticas e gráficos.
- Pode ser instalada com o Framework Anaconda.



- O RStudio é um software com uma interface amigável para desenvolvimento integrado com a linguagem R.
- Pode ser instalado com o Framework Anaconda.



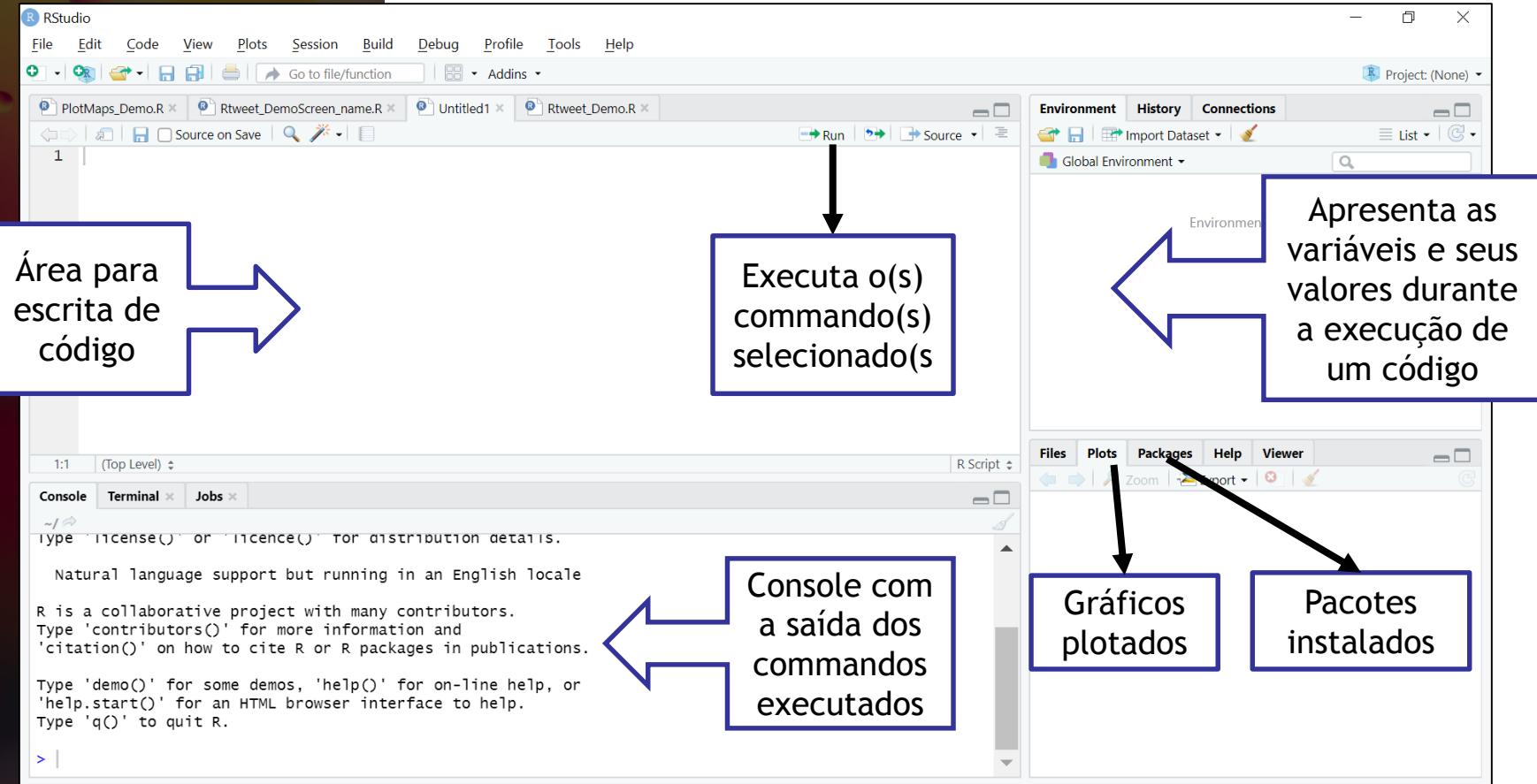
■ Download e Instalação independentes

IGTI

- Download da linguagem R:
 - Acesse: <https://cran.r-project.org/>
- Download do instalador do RStudio:
 - Acesse: <https://rstudio.com/>
- Após o download, clique com o botão direito do mouse sobre o arquivo, e selecione a opção “Executar / Executar como administrador”.
 - Para instalar basta selecionar o botão NEXT em cada janela da instalação.



Interface de trabalho do R Studio



Linguagem Python

- Criada em 1991 por Guido Van Rossumem, Python é uma linguagem de programação simples, de alto nível, interpretada e de software livre.
 - Simples (de fácil compreensão)
 - Menos linhas de código comparando como Java, C, C++
 - Documentação Gratuita e de fácil acesso
 - Alto nível:
 - Abstração elevada
 - Longe do código de máquina
 - Próximo à linguagem humana
- Sua especificação é mantida pela empresa *Python Software Foundation*.
- Pode ser instalado com o Framework Anaconda.



■ Download e Instalação independentes

- Download da linguagem Python:
 - Acesse: <https://www.python.org/downloads/>
- Download do instalador do Jupyter:
 - Acesse: <https://jupyter.org/install>
- Download do instalador do Spyder:
 - Acesse: <https://www.spyder-ide.org/>
- Para instalar basta selecionar o botão NEXT em cada janela da instalação.



Framework Anaconda

- Anaconda é uma plataforma que ajuda na preparação do ambiente necessário para análise de dados utilizando as linguagens Python e R.
- É um projeto open-source que já contém centenas de bibliotecas “embutidas” para ambas linguagens.
- A plataforma inclui diversas ferramentas e IDE's úteis para Data Science, como: Jupyter, Jupyter lab, Spyder, Orange, Microsoft Visual Code, R Studio, e outras.



ANACONDA®

Download e Instalação

- Download do instalador:
 - Acesse: <https://www.anaconda.com/products/individual>
- Para instalar basta selecionar o botão NEXT em cada janela da instalação ou siga o tutorial em uma das videoaulas seguintes.





- Ferramenta projetada para coleta, manipulação, análise e modelagem de pipelines de dados por meio de programação visual.
 - Permite análises preditivas com técnicas de aprendizado de máquina.
 - Em muitos casos não é necessário conhecer de programação.
- Oferece vários exemplos de workflows prontos para uso que podem ser facilmente parametrizados.
- Suporta à combinação de dados entre arquivos de texto simples, bancos de dados, documentos, imagens, redes e dados baseados no Hadoop em um único fluxo de trabalho visual.



Download e Instalação



Open for Innovation

KNIME

IGTI

- Download do instalador:
 - Acesse: <https://www.knime.com/knime-analytics-platform>

- Para instalar basta selecionar o botão NEXT em cada janela da instalação ou siga o tutorial em uma das videoaulas seguintes.

Conclusão

- Conhecemos algumas tecnologias que podem ser usadas na coleta de dados.
 - Linguagens R e Python.
 - Framework Anaconda.
 - Plataforma Knime Analytics.

Próxima Aula

- ❑ Aprender como instalar e criar seu ambiente de trabalho no Framework Anaconda.



Aula 1.7.2. Instalação e configuração do Framework Anaconda

Nesta Aula

- Preparar seu computador para trabalhar com o Framework Anaconda.
- Conhecer o ambiente Anaconda para trabalhar com a linguagem Python.

Criando seu ambiente com Framework Anaconda

- Framework Anaconda:
 - Acesse: <https://www.anaconda.com/>
- Download do instalador:
 - Acesse: <https://www.anaconda.com/products/individual#Downloads>
- Documentos e tutoriais:
 - Acesse: <https://docs.anaconda.com/> (em inglês).
 - Tutorial com passo a passo de instalação:
 - <https://docs.anaconda.com/anaconda/install/>



Vídeo com passo a passo da Instalação

IGTI

- Inserir o vídeo com passo a passo da Instalação

Framework Anaconda

IGTI

Anaconda Navigator

File Help

Sign in to Anaconda Cloud

Refresh

Home

Environments

Learning

Community

Documentation

Developer Blog

Twitter YouTube GitHub

ANACONDA NAVIGATOR

Applications on base (root) Channels Refresh

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated Launch	 JupyterLab 1.2.6 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch	 jupyter 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Launch	 PowerShell Prompt 0.0.1 Run a PowerShell terminal with your current environment from Navigator activated Launch	 Qt Console 4.6.0 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. Launch	 Spyder 4.0.1 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features Launch
 VS Code 1.45.1 Streamlined code editor with support for development operations like debugging, task running and version control. Launch	 Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets. Install	 Orange 3 3.23.1 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Install	 RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. Install		



Vídeo com o ambiente Python

- Inserir Vídeo com o ambiente Python

Conclusão

- Aprendemos a instalar e configurar o ambiente de trabalho utilizando o Framework Anaconda.
- Escrevemos nosso primeiro código utilizando a linguagem Python.

Próxima aula

- Configurar o ambiente Anaconda para trabalhar com a linguagem R e o R Studio.



Aula 1.7.3 Framework Anaconda: Ambiente R Studio

Nesta Aula

- Configurar o ambiente Anaconda para trabalhar com a linguagem R e o R Studio.

Criando um ambiente R no Anaconda

IGTI

Anaconda Navigator

File Help

Sign in to Anaconda Cloud

Refresh

Home

Environments

Learning

Community

Documentation

Developer Blog

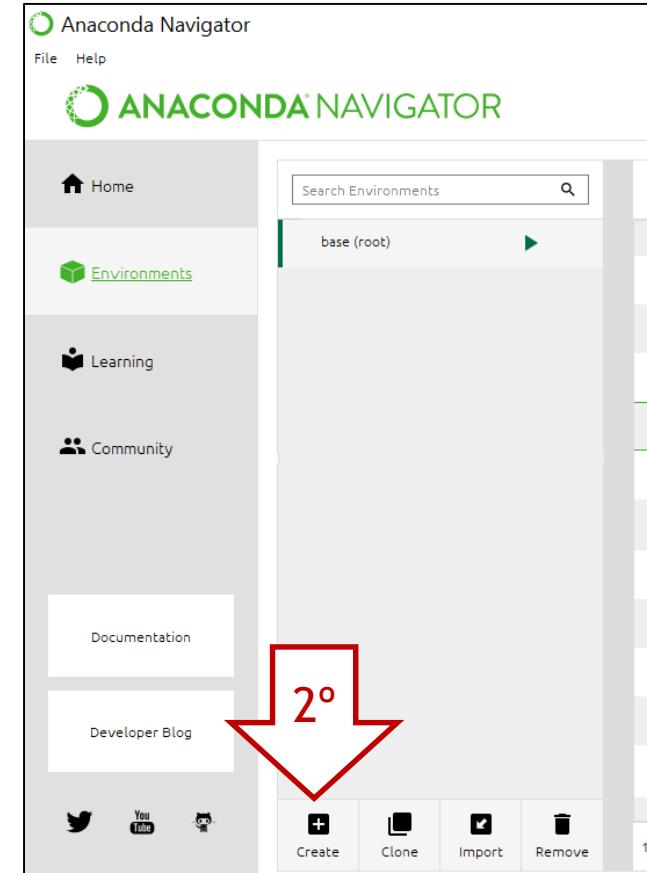
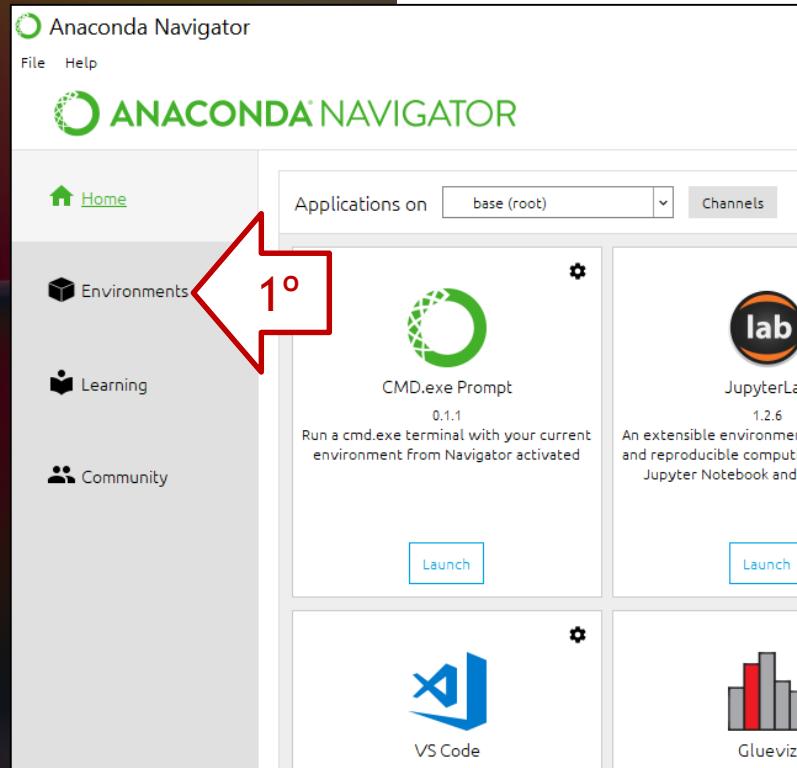
Twitter YouTube GitHub

ANACONDA NAVIGATOR

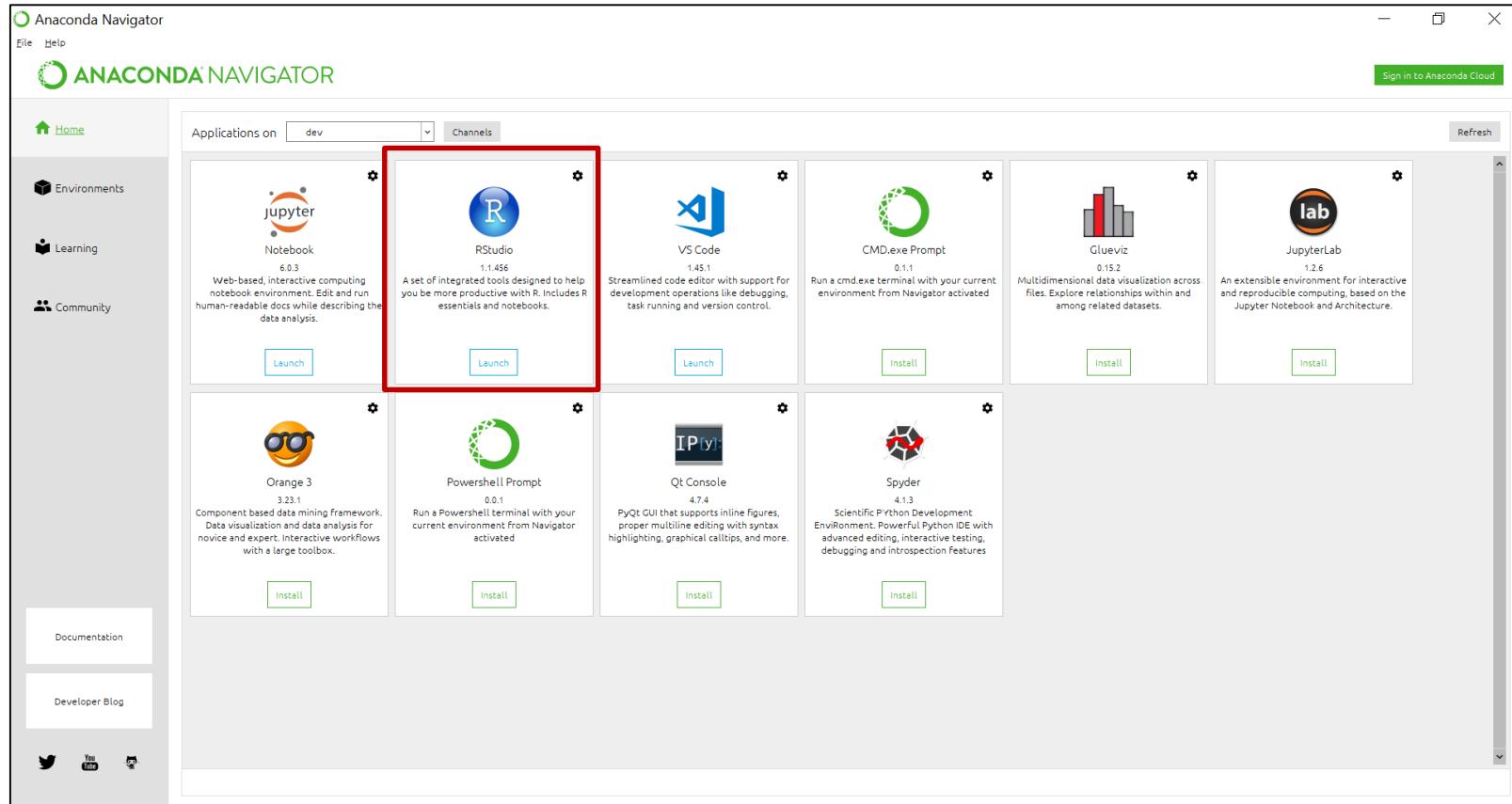
Applications on base (root) Channels Refresh

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated Launch	 JupyterLab 1.2.6 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch	 jupyter 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Launch	 PowerShell Prompt 0.0.1 Run a PowerShell terminal with your current environment from Navigator activated Launch	 Qt Console 4.6.0 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more. Launch	 Spyder 4.0.1 Scientific Python Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features Launch
 VS Code 1.45.1 Streamlined code editor with support for development operations like debugging, task running and version control. Launch	 Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets. Install	 Orange 3 3.23.1 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Install	 RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. Install		

Criando um ambiente R no Anaconda



Instalação do R Studio no Anaconda





ANACONDA®

Validando o ambiente R



IGTI

Conclusão

- Configuramos o ambiente Anaconda para trabalhar com a linguagem R e o R Studio.
- Escrevemos nosso primeiro código utilizando a linguagem R.

Próxima aula

- ❑ Instalar e configurar a Plataforma Knime Analytics.



Aula 1.7.4. Plataforma Knime Analytics (parte 1)

Nesta aula

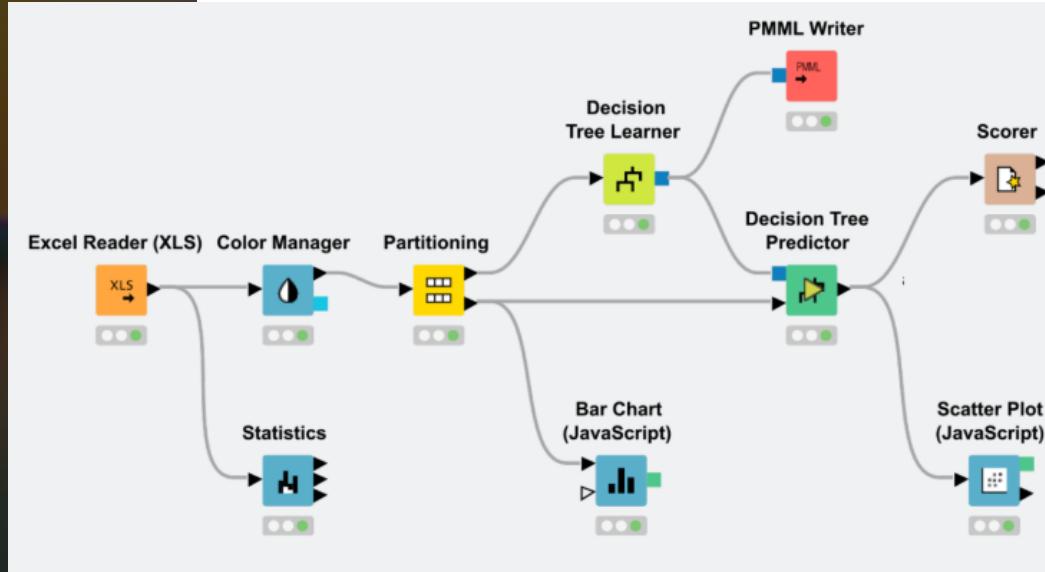
- Conhecer a plataforma KNIME Analytics.
- Aprender como instalar o Knime.



- Plataforma de código aberto para análise, relatórios e integração de dados.
- Integra vários componentes para aprendizagem de máquinas e mineração de dados através do conceito de *pipelining* de dados modular.
- Sua interface gráfica, permite a montagem de pipelines de processamento de dados (ETL), para análise e visualização de dados.
- Oferece vários exemplos prontos para uso que podem ser facilmente parametrizados.
- Suporta diversos formatos de dados e coleta dados de diferentes fontes de dados.



- O conceito de *pipelining* de dados modular:





Knime: Download e Instalação

- Link para download:
 - <https://www.knime.com/downloads>
- Vídeo com tutorial de como instalar (video em Inglês):
 - <https://www.youtube.com/watch?v=yeHbIDxakLk>
- Tutorial de instalação (em inglês):
 - https://docs.knime.com/latest/analytics_platform_installation_guide/index.html

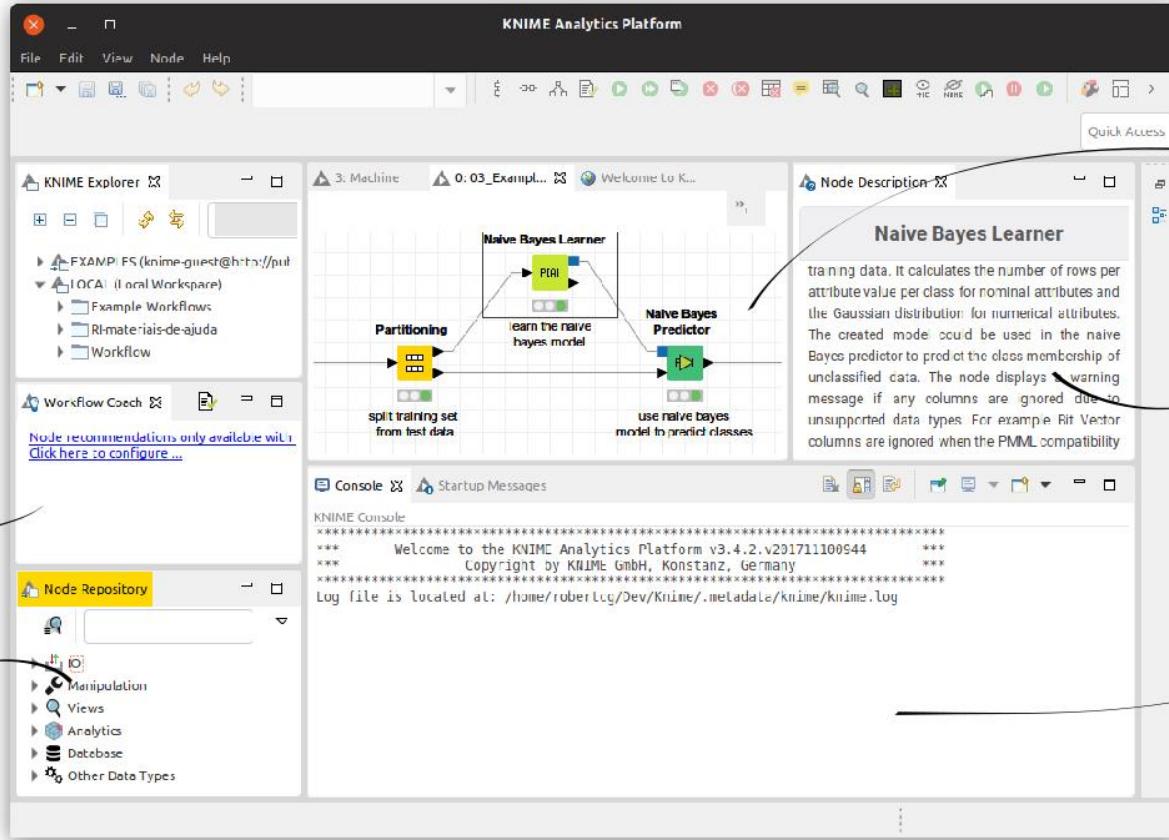
KNIME Analytics Platform

IGTI

Workflow
Projects

Favorite
Nodes

Node
Repository



Workflow
Editor

Node
Description

Console

Conclusão

- Conhecemos a Plataforma Knime Analytics em detalhes.
- Aprendemos como instalar a Plataforma Knime Analytics no sistema operacional Windows.

Próxima aula

- Aprender a instalar as extenções e integrações na Plataforma Knime Analytics.
- Conhecer algumas funcionalidades existentes na plataforma.



Aula 1.7.5. Plataforma Knime Analytics (parte 2)

Nesta aula

- ❑ Aprender a instalar as extenções e integrações na Plataforma Knime Analytics.
- ❑ Conhecer algumas funcionalidades existentes na plataforma.

- As extensões e integração são funcionalidades criadas que podem ser acopladas ao Knime para incorporar novas funcionalidades.
 - Extensões open source.
 - Extensões da comunidade.
 - Extensões de parceiros.
- Como realizar a instalação das extensões e integração ?

Conclusão

- Aprendemos a instalar as extenções e integração na Plataforma Knime Analytics.
- Conhecemos os exemplos disponíveis na plataforma.

Próxima Aula

- ❑ Criar nosso primeiro programa de coleta de dados utilizando o Framework Anaconda e as linguagens R e Python.



Aula 1.7.6. Coletando de dados em arquivos utilizando a linguagem Python

Nesta aula

- ❑ Criar um programa simples para coleta de dados em arquivos utilizando o Framework Anaconda e a linguagem Python.

Coleta de dados com Python

- **Leitura e escrita de arquivos** (txt).
- Leitura e escrita de arquivos do tipo csv.
- Leitura e escrita de arquivos do tipo json.
- Leitura de arquivos do tipo pdf.



Coleta de dados com Python

- Leitura e escrita de arquivos texto (txt).
- **Leitura e escrita de arquivos do tipo csv.**
- Leitura e escrita de arquivos do tipo json.
- Leitura de arquivos do tipo pdf.



Coleta de dados com Python

- Leitura e escrita de arquivos texto (txt).
- Leitura e escrita de arquivos do tipo csv.
- **Leitura e escrita de arquivos do tipo json.**
- Leitura de arquivos do tipo pdf.



Coleta de dados com Python

- Leitura e escrita de arquivos texto (txt).
- Leitura e escrita de arquivos do tipo csv.
- Leitura e escrita de arquivos do tipo json.
- **Leitura de arquivos do tipo pdf.**



Conclusão

- Aprendemos a fazer coleta de dados em diferentes tipos de arquivos utilizando o Framework Anaconda e a linguagem Python.
- Conhecemos alguns tipos de análises e gráficos disponíveis na linguagem Python.

Próxima Aula

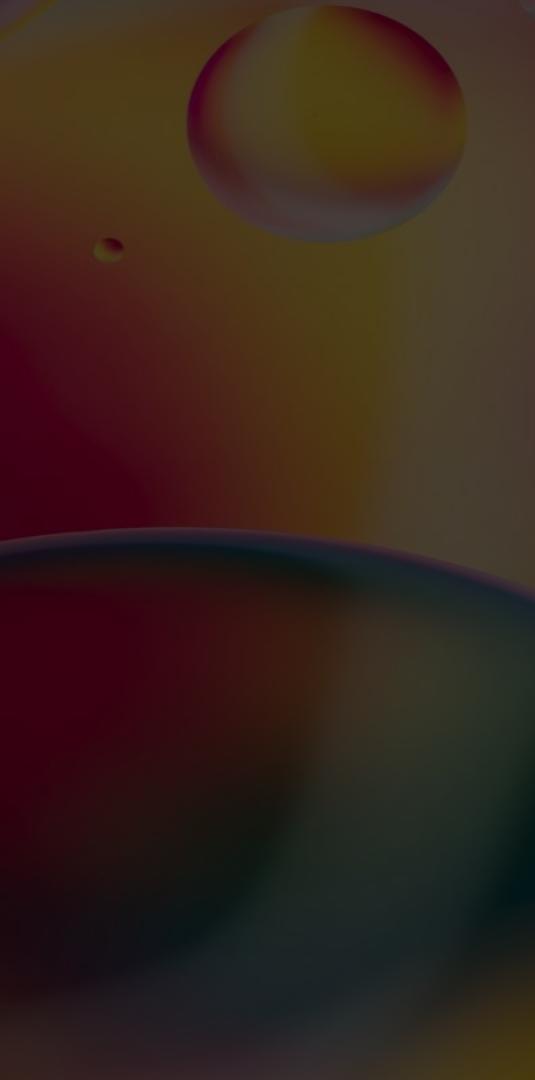
- ❑ Criar um programa simples para coleta de dados em arquivos utilizando o Framework Anaconda e a linguagem R.



Aula 1.7.7. Coletando de dados em arquivos utilizando a linguagem R

Nesta aula

- ❑ Criar um programa simples para coleta de dados em arquivos utilizando o Framework Anaconda e a linguagem R.



Vídeo da prática utilizando o R

Conclusão

- ❑ Aprendemos a fazer coleta de dados em diferentes tipos de arquivos utilizando o Framework Anaconda e a linguagem R.
- ❑ Conhecemos alguns tipos de análises e gráficos disponíveis na linguagem R.

Próxima aula

- ❑ Aprender a criar pipelines de coleta de dados na Plataforma Knime Analytics.



Aula 1.7.8. Coleta de dados em arquivos utilizando a Plataforma Knime Analytics

Nesta Aula

- ❑ Aprender a criar pipelines na Plataforma Knime Analytics para realizar a coleta de dados em arquivos.

Vídeo com a demonstração da criação do fluxo

Conclusão

- Criamos um pipeline simples para coleta de dados na Plataforma Knime Analytics.

Próxima aula

- Conhecer exemplos de API's de coleta de dados.



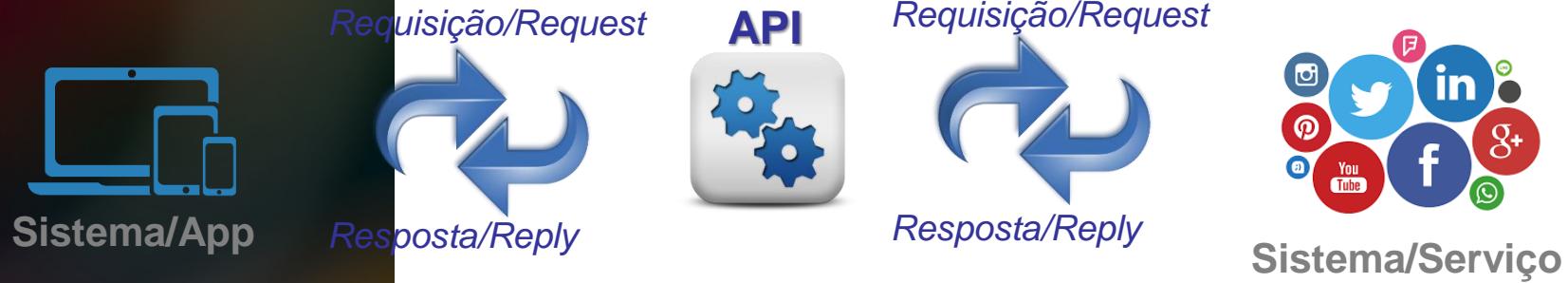
Aula 1.8. API's de coleta de dados

Nesta aula

- Entender para que servem as API's de coleta de dados.
- Conhecer exemplos de API's de coleta de dados.

O que são API's?

- API – *Application Programming Interface*.
 - Interface de Programação de Aplicações.
- Pacote que consiste em um conjunto de definições e protocolos usado no desenvolvimento e na integração de software de aplicações.



API's de coleta de dados em redes sociais

IGTI

- Podem ser usados para:
 - Aplicações externas a rede social acessam a API, permitindo criar funcionalidades externas a rede que utilizam os dados coletados na própria rede social.
 - Em geral, atendem às regras da GDPR e LGPD.

The collage consists of three screenshots:

- Left Screenshot:** A news article from exame.abril.com.br titled "Cambridge Analytica se declara culpada por uso de dados do Facebook". The article discusses how Cambridge Analytica used an app to collect data from 87 million Facebook users. It includes a small image of the Facebook logo.
- Middle Screenshot:** A quiz titled "Qual reação do facebook você seria?" by Carol Leite, posted on Medium. The quiz asks users to choose their most liked Facebook reaction. It features three emojis: a neutral face, a surprised face, and a sad face. Below the quiz is a large blue button labeled "Iniciar o Quiz".
- Right Screenshot:** A "Sign in" page for Medium. It shows several sign-in options: "Sign in with Google", "Sign in with Facebook", "Sign in with Apple", "Sign in with Twitter", and "Sign in with email". A red arrow points to the "Sign in with Facebook" button.

- Podem ser usados para:
 - Realizar o *login* à conta da rede social em sites de terceiros.



The screenshot shows a Medium article page. At the top, there's a navigation bar with a back arrow, forward arrow, refresh icon, a lock icon, the URL 'medium.com/@marlessonsantana/como-criar-apps-e-obter-os-tokens-necessarios-para-coletar-dados-do-twitter-instagram-link... ', a star icon, and a three-dot menu icon. Below the navigation is a header with a large 'M' logo, a search icon, a bookmark icon, a notification bell icon, an 'Upgrade' button, and a smiling emoji. The main title of the article is 'Como criar Apps e obter os tokens necessários para coletar dados do Twitter, Instagram, Linkedin e Facebook'. Below the title is a profile picture of the author, Marlesson Santana, with the name 'Marlesson Santana' next to it and a 'Follow' button. Below the author info is the date 'Sep 20, 2017 · 7 min read'. To the right of the author info is a red-bordered box containing social sharing icons for Twitter, LinkedIn, Facebook, and a bookmark, followed by three dots. At the bottom of the screenshot, there's a footer bar with the text 'Desenvolvedores já estão acostumados à criação de tokens em aplicações'.

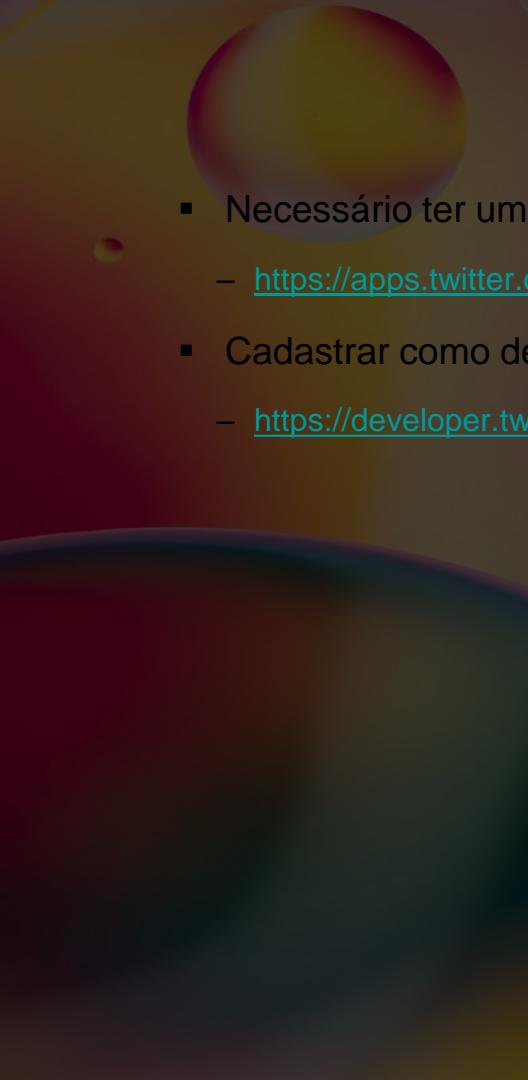
■ API's de coleta de dados em redes sociais

IGTI

- Principais API's para redes sociais:
 - Twitter - <https://developer.twitter.com/en>
 - Linkedin - <https://www.linkedin.com/developers/apps>
 - Facebook - https://developers.facebook.com/docs/graph-api?locale=pt_BR
 - Instagram - <https://www.instagram.com/developer/register/>
- Necessário ter uma conta para criar aplicativos e chaves de acesso.



Leitura recomendada: *Como criar Apps e obter os tokens necessários para coletar dados do Twitter, Instagram, Linkedin e Facebook*
Disponível em: <https://medium.com/@marlessonssantana/como-criar-apps-e-obter-os-tokens-necess%C3%A1rios-para-coletar-dados-do-twitter-instagram-linkedin-e-8f36602ea92a> - Acesso em: 12/04/2020



Twitter API: Configuração para uso

- Necessário ter uma conta no Twitter.
 - <https://apps.twitter.com/>
- Cadastrar como desenvolvedor (estudante).
 - <https://developer.twitter.com/en>

[REDACTED] Twitter API: Configuração para uso

IGTI

- Após cadastro como desenvolvedor aprovado, criar uma aplicação.
 - <https://developer.twitter.com/en>

The image shows two screenshots of the Twitter Developer website. The top screenshot displays the main dashboard with a purple header and navigation menu. A red arrow points from the 'Dashboard' button in the top header to the 'Apps' section in the sidebar. The bottom screenshot shows the 'Apps' page, which has a purple header and a 'Create an app' button. A red arrow points to this button. Both screenshots include a sidebar on the right with sections like 'Get started', 'Subscriptions', 'Apps' (with a sub-section for 'Apps'), 'Dev environments', 'Billing', and 'Team'. The URL in both browser bars is developer.twitter.com/en/apps.

developer.twitter.com/en

Developer Use cases Products Docs More Labs Dashboard IGTI Dev Q

Get started

Subscriptions

Apps Apps

Dev environments

Billing

Team

developer.twitter.com/en/apps

Developer Use cases Products Docs More Labs Dashboard IGTI Dev

Apps

Create an app

No apps here.

You'll need an app and API key in order to authenticate and integrate with most Twitter developer products. Create an app to get your API key.

Twitter API: Criação do app para uso

IGTI

The screenshot shows the 'Create an app' page in the Twitter Developer portal. The left panel contains fields for 'App details', 'Application description', 'Website URL', and 'Callback URLs'. The right panel contains fields for 'Terms of Service URL', 'Privacy policy URL', 'Organization name', 'Organization website URL', and 'Tell us how this app will be used'. Red arrows and boxes with annotations explain the purpose of each field:

- Nome:** No caso de usar o App para login, esse nome aparece.
- Descrição:** No caso de login, essa descrição aparece.
- Website do app com as informações.** Opcional, nesse caso deve-se colocar `http://localhost`
- Callback URL.** Essa é a URL de retorno utilizada quando é realizado Login
- Descrição completa**

At the bottom right, there is a large red arrow pointing down towards the 'Create' button, which is highlighted in blue.

Twitter API: Criação do app para uso

IGTI

The screenshot shows the Twitter Developer website's 'Create an app' interface. On the left, there's a sidebar with links like 'Understanding apps', 'What is an app?', 'Why register an app?', and 'Which products require an API key?'. The main form on the right has fields for 'App name' (ExemploIGTI), 'Application description' (Esta aplicação é um exemplo), 'Website URL' (https://igti.com.br), 'Allow this application to be' (checkbox checked for 'Enable Sign in with Twitter'), 'Callback URLs' (https:// or scheme://), 'Terms of Service URL' (https://), 'Privacy policy URL' (https://), 'Organization name' (empty), 'Organization website URL' (https://), and 'Tell us how this app will be used' (checkbox checked for 'O uso será exclusivamente comercial'). A large modal window titled 'Review our Developer Terms' is centered over the form. It contains a reminder about the 'Developer Agreement and Policy' and 'restricted use cases'. Three sections are listed with red circular icons and crossed-out text: 'Sensitive Information' (about potentially sensitive characteristics of users), 'Government use and surveillance' (prohibition of surveillance), and 'Automation' (about posting tweets, following accounts, or sending direct messages). At the bottom of the modal are 'Cancel' and 'Create' buttons. A large red arrow points from the bottom right towards the 'Create' button.

Review our Developer Terms

As a reminder, you have agreed to our [Developer Agreement and Policy](#). Please be mindful of the following [restricted use cases](#)

✖ Sensitive Information

Be careful about using Twitter data to derive or infer potentially sensitive characteristics about Twitter users (ie. health, political or religious affiliation, ethnic origin, sexual orientation, and more).

✖ Government use and surveillance

We prohibit the use of Twitter Data and Twitter APIs by any entity for surveillance purposes. Period.

✖ Automation

If your app will be used to post Tweets, follow accounts, or send Direct Messages, you should carefully review the [Automation Rules](#) to ensure you comply with our guidelines and never perform bulk, aggressive, or spammy actions.

Cancel Create

[REDACTED] Twitter API: Criação do app para uso

The screenshot shows the Twitter Developer Apps page at developer.twitter.com/en/apps. A purple navigation bar at the top includes links for Developer, Use cases, Products, Docs, More, Labs, Apps (selected), and IGTI Dev. A blue button labeled 'Create an app' is visible on the right. Below the bar, a section titled 'Apps' shows a single entry: 'ExemploIGTI' with an App ID of 17707047. To the right of this entry is a 'Details' button and a vertical ellipsis menu. A large red arrow points upwards from the bottom of the slide towards the 'Details' button.

- Após a criação do App, é necessário obter a chave de acesso.

Twitter API: Dados de acesso

The screenshot shows the Twitter Developer App Details page for an application named "ExemploIGTI". The page has a purple header with links for Developer, Use cases, Products, Docs, More, Labs, and Apps. Below the header, the breadcrumb navigation shows "Apps > ExemploIGTI". There are three tabs: "App details" (selected), "Keys and tokens", and "Permissions". A large red arrow points to the "App icon" section, which includes a placeholder image and a "Click edit to upload a new icon." button. The "Edit" button is located in the top right corner of the main content area. The app details listed are:

- App Name:** ExemploIGTI
- Description:** Esta aplicação é um exemplo para a disciplina do IGTI.
- Website URL:** <https://igti.com.br>
- Sign in with Twitter:** Disabled
- Callback URL:** None

Twitter API: Dados de acesso

The screenshot shows the Twitter Developer Keys and tokens page for an app named "ExemploIGTI". The "Keys and tokens" tab is selected. A red box highlights the "Consumer API keys" section, which contains fields for "API key" and "API secret key", both of which are blurred. A red arrow points from a text box on the left to this section. Another red box highlights the "Access token & access token secret" section at the bottom, which contains a note: "This is where you can create a token to auth into your Twitter account." A "Generate" button is visible in this section.

Developer Use cases Products Docs More Labs App

Apps > ExemploIGTI

App details Keys and tokens Permissions

Important notice about your access token and access token secret
Looking for your secret token? For security, API tokens are only displayed once. You will need to regenerate access tokens for previously authenticated apps. To learn more, click here.

Keys and tokens
Keys, secret keys and access tokens management.

Consumer API keys

API key: [REDACTED]
API secret key: [REDACTED]

Regenerate

Access token & access token secret

This is where you can create a token to auth into your Twitter account.

Generate

As informações de Consumer Key e Consumer Secret estarão disponíveis com a criação da APP

Twitter API: Dados de acesso

IGTI

The screenshot shows the Twitter Developer Keys and tokens page for the app "ExemploIGTI". The "Keys and tokens" tab is selected. A notice states that API tokens are only displayed once and must be regenerated. The "Consumer API keys" section shows blurred API key and secret values with a "Regenerate" button. A red arrow points to the "Generate" button in the "Access token & access token secret" section, which contains a note about creating a token to auth into the account.

Developer Use cases Products Docs More Labs App

Apps > ExemploIGTI

App details Keys and tokens Permissions

Important notice about your access token and access token secret

Looking for your secret token? For security, API tokens are only displayed once. You will need to regenerate access tokens for previously authenticated apps. To learn more, read our [FAQ](#).

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

API key: [REDACTED]

API secret key: [REDACTED]

Regenerate

Access token & access token secret

This is where you can create a token to auth into your Twitter account.

Generate

Twitter API: Dados de acesso

- Youtube - <https://developers.google.com/youtube/v3/>
- Google Data API - <https://developers.google.com/gdata/docs/directory>
- API Jornal New York Times - <https://developer.nytimes.com/>
- API do Portal Pubmed - <https://www.ncbi.nlm.nih.gov/home/develop/api/>
- APIs do Portal da Transparéncia Brasileiro:
 - <http://www.portaltransparencia.gov.br/api-de-dados>
 - <http://www.transparencia.gov.br/swagger-ui.html>

Conclusão

- Entendemos o que são API's (*Application Programming Interface*).
 - Interface de Programação de Aplicações
- Conhecemos o papel das APIs no processo de coleta de dados.
- Conhecer alguns exemplos de API's de coleta de dados.
 - Redes Sociais (Twitter, Linkedin, Facebook, Instagram).
- Aprendemos como realizar o cadastro para utilizar a API do Twitter.

Próxima aula

- ❑ Conhecer algumas técnicas para coletar dados em bases de dados relacionais.



Coleta e Obtenção de Dados

Capítulo 2: Coleta de dados em bancos de dados relacionais

Profa. Fernanda Farinelli



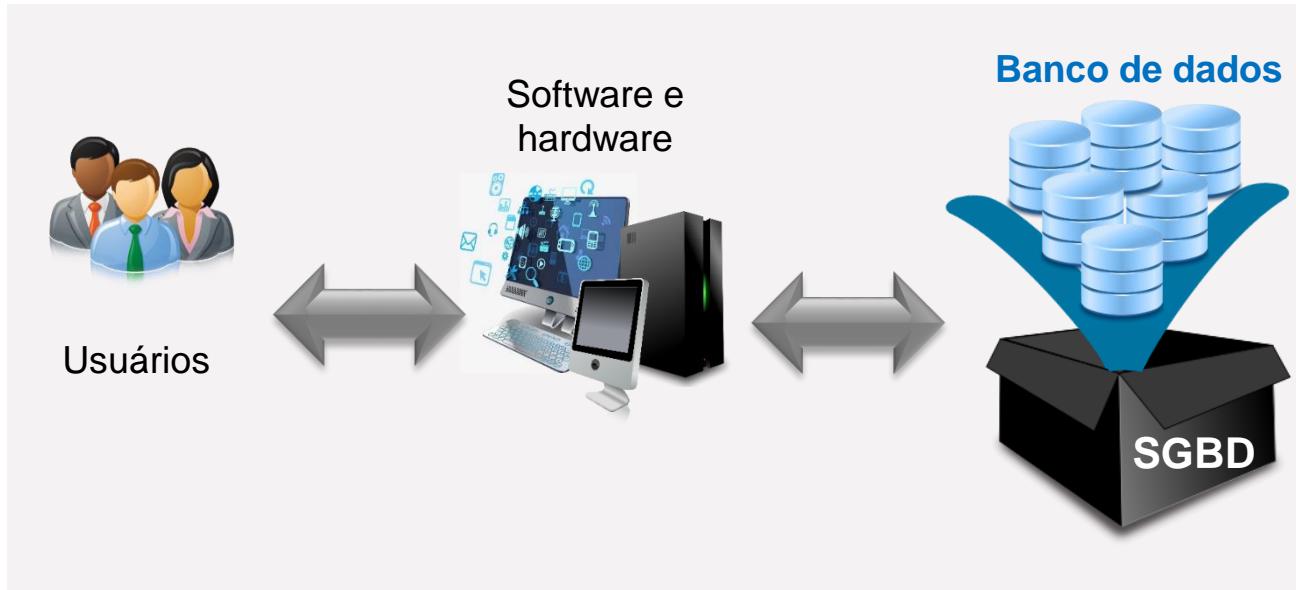
Aula 2.1. Sistemas gerenciadores de bancos de dados relacionais

- Entender o que são sistemas gerenciadores de bancos de dados relacionais.

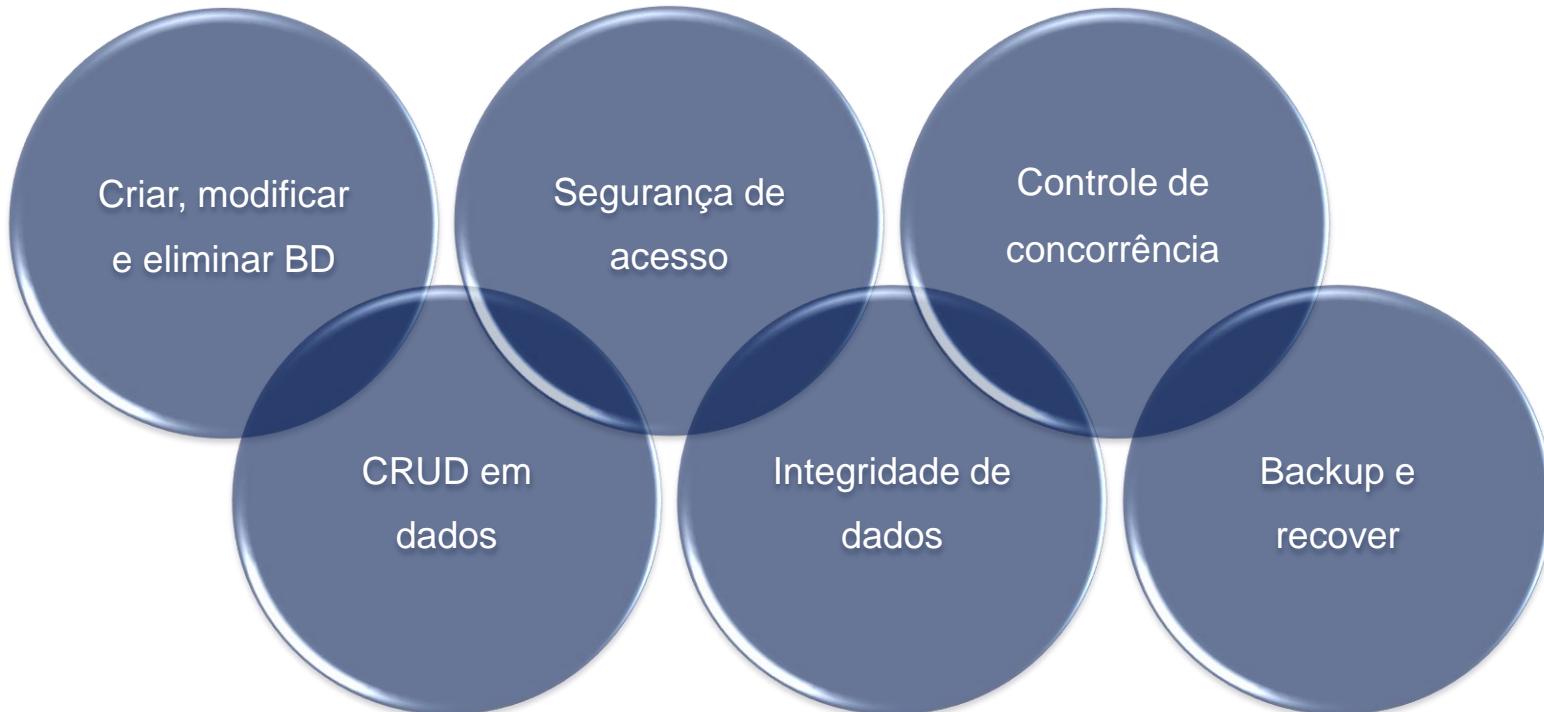
- Conjunto de dados armazenados (persistidos).
- Coleção de dados inter-relacionados.



Sistemas de banco de dados



Sistema Gerenciadores de Banco de Dados



Bancos de dados
relacionais

Bancos de dados
NoSQL

Bancos de dados
NewSQL

Implementa o modelo relacional de dados.

Fundamentado na teoria de conjuntos e nas relações entre os conjuntos.

- Operações de junção, união e seleção.

Persistência de dados mais adotados.

Persistência de dados estruturados.

Propriedades de transação ACID

Atomicidade

Transação é uma unidade atómica de processamento.

Ou se faz tudo,
ou nada.

Consistência

Os dados estarão sempre consistentes e completos.

Uma transação leva de um estado consistente a outro estado consistente.

Isolamento

Cada transação é realizada de forma isolada.

Uma transação começa após o término da outra.

Durabilidade

As alterações causadas por uma transação devem ser persistidas.

O dado deve durar, estar salvo, ele não deve ser perdido.

SGBDs Relacionais

ORACLE®



Microsoft®
SQL Server

IBM® **DB2**®

 **SQLite**

The SQLite logo features a blue feather quill pen icon next to the word "SQLite" in a serif font.

 **MySQL**®

The MySQL logo consists of the word "MySQL" in a bold, orange and blue sans-serif font, accompanied by a blue stylized wave or flame icon above the letter "M".

 **Firebird**

The Firebird logo features a stylized orange and yellow flame or bird icon to the left of the word "Firebird" in a white sans-serif font.

 **MariaDB**®

The MariaDB logo features a black line drawing of a seal or sea lion icon above the word "MariaDB" in a black sans-serif font.

- Entendemos o que são bancos de dados.
- Visão geral dos bancos de dados relacionais.
- Conhecemos as propriedades ACID que regem as transações em um banco de dados relacional.
- Conhecemos algumas ferramentas disponíveis.

■ Próxima aula

- Entender o que é modelagem de dados.

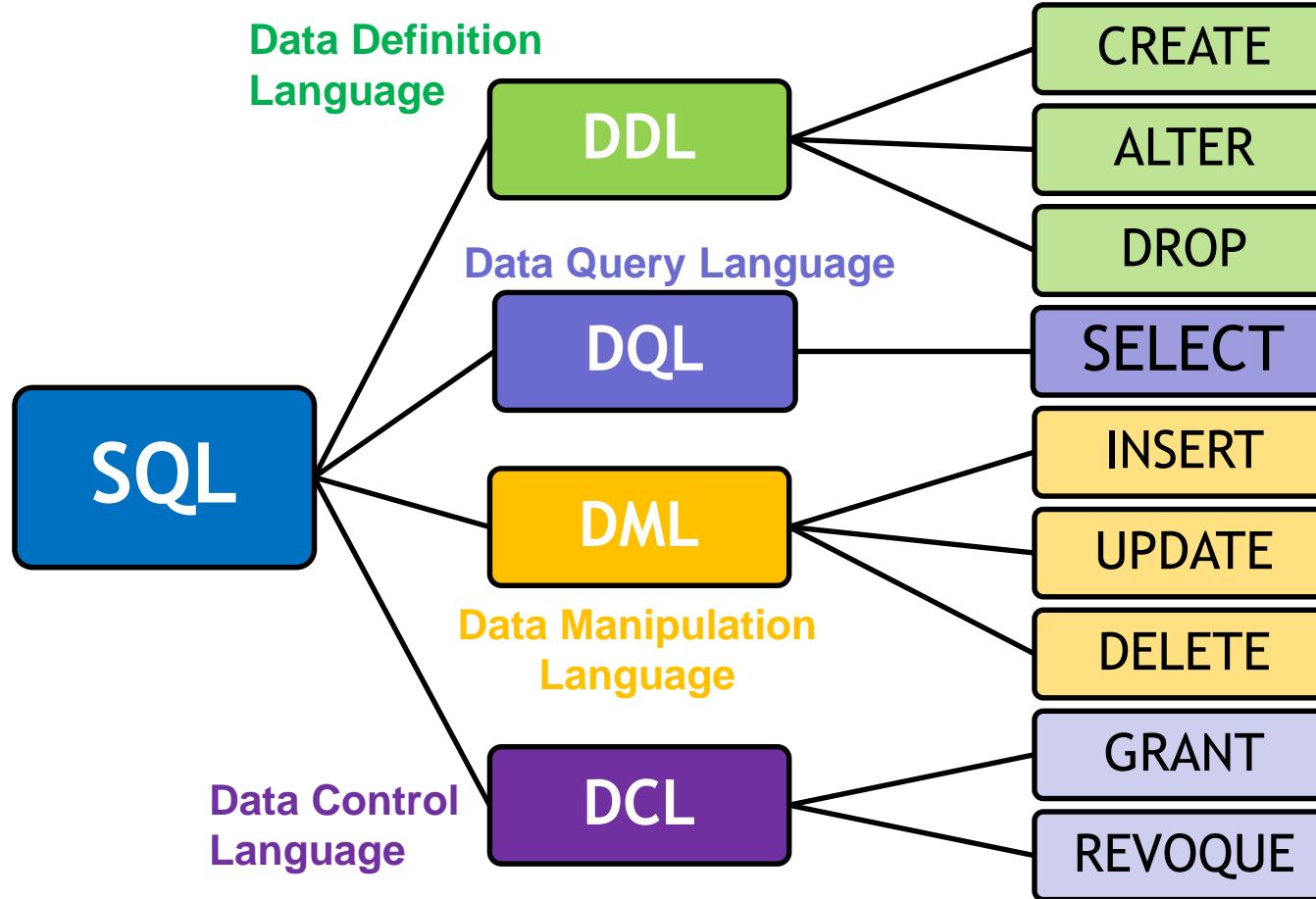


Aula 2.2. Introdução à linguagem estruturada de consulta (SQL)

- Apresentar a linguagem estruturada de consulta usada na criação e manipulação de dados em bases de dados relacionais.

- *Structured Query Language (SQL).*
- Linguagem comercial usada fundamentalmente em SGBDs relacionais:
 - Padrão ISO desde a década de 80.
- A razão do sucesso dos bancos de dados relacionais e da linguagem SQL se deve ao fato de existir um modelo matemático formal, que serviu de base para seu desenvolvimento.
 - Sua base formal é a álgebra relacional e a teoria dos conjuntos.
- Não é apenas uma linguagem de consulta.

Linguagem SQL



DDL

- **Linguagem de Definição de Dados:**
- É um conjunto de instruções usado para criar e modificar as estruturas dos objetos armazenados no banco de dados.

DQL

- **Linguagem de Consulta de Dados:**
- É um conjunto de instruções usado para consultar dados nas estruturas dos objetos armazenados no banco de dados.

DML

- **Linguagem de Manipulação de Dados:**
- É um conjunto de instruções usada nas consultas e modificações dos dados armazenados nas tabelas do banco de dados.

DCL

- **Linguagem de Controle de Dados:**
- São usados para controle de acesso e gerenciamento de permissões para usuários em no banco de dados.

DTL

- **Linguagem de Controle de Transações:**
- São usados para gerenciar as mudanças feitas por instruções DML.
- Permite as declarações serem agrupadas em transações lógicas.

Conclusão

- Visão geral da linguagem SQL.

■ Próxima aula

- Conhecer a linguagem de definição de dados.



Aula 2.3 Linguagem de Definição de Dados

Nesta aula

- Conhecer a linguagem de definição de dados do SQL.

Data Definition Language (DDL)

Subconjunto de comandos para criação e manutenção de objetos de banco de dados.

- Esquemas, tabelas, índices, chaves, colunas, visões, restrições de integridade etc.

Linguagem de definição de dados

- CREATE *objeto*:
 - Cria um objeto (uma Tabela, por exemplo) no banco de dados.
- ALTER *objeto*:
 - Altera a estrutura ou a configuração de um objeto no banco de dados.
- DROP *objeto*:
 - Exclui um objeto do banco de dados.
- Objetos podem ser: DATABASE, TABLE, INDEX, CONSTRAINT (PRIMARY KEY, FOREING KEY, UNIQUE KEY), ROLE, USER, PROCEDURE, FUNCTION, TRIGGER e VIEW.
 - Existem variações entre fabricantes de SGBDs.

- Criação de um BD:
 - SQL padrão não oferece tal comando.
 - CREATE DATABASE <nome_BD>;
 - Exemplo: CREATE DATABASE livraria;

- Exclusão de um BD:
 - SQL padrão não oferece tal comando.
 - Quando executado, exclui tudo, BD e todos os objetos de dados abaixo dele.
 - DROP DATABASE <nome_BD>;
 - Exemplo: DROP DATABASE livraria;

DDL - Criando um esquema de dados

- Criação e Exclusão de um Esquema:
 - CREATE SCHEMA <nome_esquema>;
 - Exemplo: CREATE SCHEMA Livraria;
 - DROP SCHEMA <nome_BD>;
 - Exemplo: DROP SCHEMA Livraria;

Linguagem de definição de dados

- Alguns comandos da DDL:

- **CREATE TABLE**
 - **ALTER TABLE**
 - **DROP TABLE**
 - **CREATE DOMAIN**
 - **ALTER DOMAIN**
 - **DROP DOMAIN**
 - **CREATE VIEW**
 - **ALTER VIEW**
 - **DROP VIEW**
 - **CREATE INDEX**
 - **ALTER INDEX**
 - **DROP INDEX**
 - ...

Exemplos de comandos na DDL no MySQL

```
25
26  -- CREATE
27  CREATE TABLE Person(
28      PersonID INT IDENTITY (1,1) CONSTRAINT PK_PersonID PRIMARY KEY,
29      FirstName NVARCHAR(20),
30      LastName NVARCHAR(25)
31 );
32
33  -- ALTER
34  ALTER TABLE Person
35      ADD BirthDate DATETIME
36
37  -- DROP
38  DROP TABLE Person
39
```

Comandos DDL

- **CREATE TABLE**: criar uma tabela, definir colunas e restrições

```
CREATE TABLE tabela (
    atrib1 tipo [<restrições da coluna 1>] ,
    atrib2 tipo [<restrições da coluna 2>] ,
    ....
    atribn tipo [<restrições da coluna n>] ,
    <restrições da tabela>
);
```

- **CHAR(tamanho)**: sequência de caracteres de tamanho fixo.
- **VARCHAR(tamanho)**: sequência de caracteres de tamanho variável.
- **NUMBER(total, decimais)**: valores numéricos.
- **DATE**: data e hora.

Alguns tipos de dados do MySQL

- INTEGER | SMALLINT | NUMBER
- DECIMAL [(precision, scale)]
 - precision - número total de dígitos.
 - scale - número de dígitos depois do ponto.
- CHAR(n) - tamanho fixo - n caracteres
- VARCHAR(n) - tamanho variável
 - máximo de n caracteres.
- BLOB – Binary Large Object –
- DATE | TIME | TIMESTAMP
- ...

CREATE TABLE

- Restrições de colunas:

- NOT NULL
- DEFAULT *valor*
- CHECK(*condição*)

```
CREATE TABLE tabela (  
  
    atrib1 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
  
    atrib2 tipo [(tamanho)] [NOT NULL | DEFAULT valor]  
        [CHECK (condição)],  
  
    ...
```

CREATE TABLE

- Restrições de tabela:

- PRIMARY KEY (*<atributos chave primária>*)
 - UNIQUE (*<atributos chave candidata>*)
 - FOREIGN KEY (*<atributos chave estrangeira>* REFERENCES *tabelaRef* [(*<chave primária>*)]
[*<ações>*]
 - Onde *<ações>*
 - ON DELETE | ON UPDATE
 - » CASCADE | SET NULL | SET DEFAULT

Linguagem de definição de dados

- Exemplos no MySQL:

```
CREATE TABLE `poc`.`produto` (
  `codigo` int(11) NOT NULL AUTO_INCREMENT,
  `descricao` varchar(100) NOT NULL,
  `estoque` int(11) DEFAULT NULL,
  `preco` float DEFAULT NULL,
  PRIMARY KEY (`codigo`)
);
```

```
CREATE TABLE `poc`.`fornecedor` (
  `codigo` int(11) NOT NULL AUTO_INCREMENT,
  `descricao` varchar(100) NOT NULL,
  PRIMARY KEY (`codigo`)
);
```

```
ALTER TABLE `poc`.`produto`
ADD `fornecedor` int(11);
```

```
ALTER TABLE `poc`.`produto`
ADD CONSTRAINT FK_Fornecedor
FOREIGN KEY (`fornecedor`) REFERENCES fornecedor(codigo);
```

- SQL oferece quatro instruções para definição de tabelas:

- **Create Table:**

- Define a estrutura da tabela.
 - Cria a tabela vazia.

- **Drop Table:**

- Elimina a tabela da base de dados.
 - Se a tabela já estiver populada, os dados também serão excluídos.

- **Alter Table:**

- Permite modificar a definição da tabela.

- **Truncate Table:**

- Elimina apenas os dados da tabela.

DDL - Criando uma tabela

- **CREATE TABLE** - cria a tabela, define colunas e restrições.

```
CREATE TABLE tabela (
    coluna 1 tipo [<restrições da coluna 1>] ,
    coluna 2 tipo [<restrições da coluna 2>] ,
    ....
    coluna n tipo [<restrições da coluna n>] ,
    <restrições da tabela>
) ;
```

- Restrições de colunas:

- NOT NULL.
- DEFAULT valor.
- CHECK (*condição*) .

CREATE TABLE

- Restrições de tabela:

- PRIMARY KEY (<coluna(s) da chave primária>)
- UNIQUE (<coluna(s) da chave candidata>)
- FOREIGN KEY (< coluna(s) da chave estrangeira>
REFERENCES tabelaRef [(<chave primária>)]
[<ações>]
 - <ações>
 - ON DELETE | ON UPDATE
 - » CASCADE | SET NULL | SET DEFAULT
 - CHECK (*condição*)

CREATE TABLE

- Exemplo no MySQL:

```
CREATE TABLE editora (
    ideditora int(11) NOT NULL AUTO_INCREMENT,
    nome varchar(100) NOT NULL,
    fone varchar(30) DEFAULT NULL,
    PRIMARY KEY (ideditora)
);
```

```
CREATE TABLE genero (
    idgenero int(11) NOT NULL AUTO_INCREMENT,
    descricao varchar(100) NOT NULL,
    PRIMARY KEY (idgenero)
);
```

```
CREATE TABLE livro (
    idlivro int(11) NOT NULL AUTO_INCREMENT,
    titulo varchar(100) NOT NULL,
    preco float DEFAULT NULL,
    estoque int(11) DEFAULT '0',
    idgenero int(11) NOT NULL,
    ideditora int(11) NOT NULL,
    PRIMARY KEY (idlivro,idgenero,ideditora),
    KEY fk_livro_Genero1 (idgenero),
    CONSTRAINT fk_livro_Genero1 FOREIGN KEY (idgenero) REFERENCES genero (idgenero)
);
```

- **ALTER TABLE** – incluir/alterar/remover definições de colunas e restrições.

```
ALTER TABLE tabela <ação>;
```

- **Onde <ação>:**

- ADD *novaColuna tipo* [<restrições de coluna>]
- ADD [CONSTRAINT *nome*] <restrição de tabela>
- DROP *coluna* [CASCADE | RESTRICT]
- DROP CONSTRAINT *nome*
- ALTER *coluna* DROP DEFAULT;
- ALTER *coluna* SET DEFAULT <valor>;

- **ALTER TABLE**

- **ADD** *novoAtrib tipo [<restrições de coluna>]*
- **DROP** *atributo [CASCADE | RESTRICT]*
 - **CASCADE**: todas as visões e restrições (*constraints*) que referenciam o atributo, são removidas automaticamente.
 - **RESTRICT**: atributo só é removido se não houver nenhuma visão ou restrição que o refcrcie.

ALTER TABLE

- Exemplo no MySQL:

```
ALTER TABLE editora
    ADD endereco varchar(150) NULL,
    MODIFY COLUMN nome varchar(150)
;
```

```
ALTER TABLE livro
    ADD CONSTRAINT fk_livro_Editora1 FOREIGN KEY (ideditora)
        REFERENCES editora (ideditora),
    ADD numedicao integer NULL
;
```

- **TRUNCATE TABLE:** exclui todas as linhas de uma tabela da base de dados, mantendo a tabela vazia.
- **TRUNCATE TABLE tabela;**
 - **CASCADE:** todas as visões e restrições que referenciam a tabela são removidas automaticamente
 - **RESTRICT:** a tabela é removida somente se não for referenciada em nenhuma restrição ou visão.

TRUNCATE TABLE

- Consider the Cliente table and its data, as shown in the image below:

Tabela Cliente

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	900
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	72
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700
11	Kiev Luna	Av. Dom Pedro I, 73	Centro	São Paulo	SP	NULL	600
12	Lineu Silva	Rua das Palmeiras, 55	Rudas	Campinas	SP	1994-08-18	20
13	Maria da Concei...	Rua Que Mora, 30	Morada	Rio de J...	RJ	1977-07-07	0
14	Maria da Penha	Rua das Mangas, 1222	Pomar	Niteroi	RJ	NULL	0
15	Maria da Anunci...	Rua das Peras, 1234	Pomar	Nitereio	RJ	1979-03-05	199
16	Nilo Pereira	R. Euclides Cunha, 138	Santana	Joao Pe...	PB	1945-09-01	6000
17	Otavio Mesquita	R. Olavo Bilac, 76	Poetas	Belo Hor...	MG	NULL	0
18	Pedro Perreira	R. Dom Casmurro, SN	Obras	Manaus	AM	NULL	3000
19	Quirino Silva	R. Harry Potter, 90	Livros	Londres	AM	NULL	100
20	Raquel Sinte	R. Vocal Metalizado, 765	Maquin	Porteira	RS	1983-09-26	750

TRUNCATE TABLE

- Para limpar os dados da tabela, execute o comando abaixo:

```
TRUNCATE TABLE cliente;
```

- O resultado será a tabela vazia:

Tabela Cliente

- **DROP TABLE**: exclui uma tabela da base de dados.
- **DROP TABLE**: tabela [CASCADE | RESTRICT];
 - **CASCADE**: todas as visões e restrições que referenciam a tabela são removidas automaticamente.
 - **RESTRICT**: a tabela é removida somente se não for referenciada em nenhuma restrição ou visão.

DROP TABLE

- Exemplo no MySQL:

```
DROP TABLE cliente;
```

DDL - Trabalhando com índices

- Criar um índice:

```
CREATE INDEX <nome do índice>  
ON tabela <coluna(s)>;
```

- Exemplo no MySQL:

- Considere a tabela Cliente conforme script abaixo à esquerda.

```
CREATE TABLE cliente (  
    codigo int(11) NOT NULL AUTO_INCREMENT,  
    nome varchar(100) NOT NULL,  
    endereco varchar(200) DEFAULT NULL,  
    bairro varchar(100) DEFAULT NULL,  
    cidade varchar(100) DEFAULT NULL,  
    estado varchar(2) DEFAULT NULL,  
    datanasc date DEFAULT NULL,  
    debito double DEFAULT '0',  
    PRIMARY KEY (codigo)  
) ;
```

```
CREATE INDEX cidade_idx  
ON cliente (cidade);
```

```
CREATE INDEX estado_idx  
ON cliente (estado);
```

- Excluir um índice (varia conforme o SGBD):

- MYSQL:

```
ALTER TABLE table_name DROP INDEX index_name;
```

- SQLServer:

```
DROP INDEX table_name.index_name;
```

- Oracle e DB2:

```
DROP INDEX index_name;
```

- Exemplo no MySQL:

ALTER TABLE cliente DROP INDEX cidade_idx;

DDL - Trabalhando com visões

- Criar uma visão (view):

```
CREATE VIEW <nome da view> AS  
<commando DQL para criação da visão>;
```

- Excluir uma visão (view):

```
DROP VIEW <nome da view>;
```

- Exemplo no MySQL - Considere a tabela cliente:

```
CREATE VIEW Cliente_MG AS  
SELECT codigo, nome, endereco, bairro,  
cidade, estado, datanasc, debito  
FROM cliente  
WHERE estado = "MG";
```

```
DROP VIEW Cliente_MG;
```

- Conhecemos a linguagem de definição de dados do SQL e seus principais comandos.
- Aprendemos a criar, alterar e excluir objetos do banco de dados.

■ Próxima aula

- ❑ Aprender como incluir, excluir e alterar dados nas tabelas do banco de dados.



Aula 2.4. Linguagem de manipulação de dados

- Conhecer a linguagem de manipulação de dados do SQL.
- Aprender como incluir, excluir e alterar dados nas tabelas do banco de dados.

- *Data manipulation language* (DML).
- Parte da linguagem SQL que define operações de manipulação de dados:
 - Inclusão / inserção (INSERT).
 - Alteração (UPDATE).
 - Exclusão (DELETE).
- Instruções declarativas:
 - Manipulação de conjuntos.
 - Especifica-se o que fazer e não como fazer.

- Inclusão ou inserção de dados:

```
INSERT INTO nome_tabela [(lista_atributos)]  
VALUES (lista_valores_atributos)  
[ , (lista_valores_atributos) ]
```

Linguagem de manipulação de dados

- Exemplos usando o MySQL:

```
INSERT INTO genero(idgenero,descricao)
VALUES (1,'Aventura');
```

```
INSERT INTO genero(descricao)
VALUES ('Ficção');
```

```
INSERT INTO genero(idgenero,descricao)
VALUES (3,'Suspense'),(4,'Romance'),(5,'Infantil'),(6,'Auto-Ajuda');
```

```
INSERT INTO genero
VALUES (7,'Infanto-juvenil');
```

```
CREATE TABLE genero (
    idgenero int(11) NOT NULL AUTO_INCREMENT,
    descricao varchar(100) NOT NULL,
    PRIMARY KEY (idgenero)
);
```

Linguagem de manipulação de dados

- Alteração de dados:

UPDATE *nome_tabela*

SET *nome_atributo_1* = *Valor*

[*{, nome_atributo_n* = *Valor}*]]

[**WHERE** *condição*]

- Exemplo no MySQL:

UPDATE genero

SET descricao = 'Adulto'

WHERE idgenero = 7;

- Exclusão de dados:

```
DELETE FROM nome_tabela  
[WHERE condição]
```

- Exemplo no MySQL:

DELETE FROM genero **WHERE** idgenero = 7;

DELETE FROM genero;

- ✓ Conhecemos os comando da linguagem SQL para incluir, excluir e alterar dados nas tabelas do banco de dados.

- ❑ Realizar os comandos de DDL e DML no MySQL, utilizando a ferramenta MySQL Workbench.
- ❑ Usar o MySQL Workbench para realizar alguns comandos de criação de objetos e de manipulação de dados.



Aula 2.5. MySQL Workbench: DDL e DML

Nesta aula

- Usar o MySQL Workbench para realizar alguns comandos de criação de objetos e de manipulação de dados.

- Conhecemos as funcionalidades do MySQL Workbench para criar objetos de banco de dados utilizando a linguagem DDL do SQL.
- Conhecemos as funcionalidades do MySQL Workbench para manipular (incluir, alterar e excluir) dados do banco de dados utilizando a linguagem DML do SQL.

■ Próxima aula

- Conhecer a linguagem de consulta de dados.



Aula 2.6.1. Linguagem de consulta de dados (Parte 1)

- ❑ Conhecer a Linguagem de Consulta de Dados.

Linguagem de Consulta de Dados

- *Data query language* (DQL)
- SELECT – comando de consulta
 - retorno \Rightarrow tabela resultado (**multiconjunto – potencialmente um conjunto com repetições**)
- Sintaxe básica (consulta em uma única tabela):

```
SELECT <lista de atributos> ou *
  FROM <tabela>
```

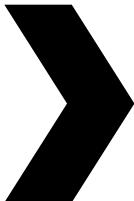
* significa que são todas as colunas

Linguagem de Consulta de Dados

- Exemplo no MySQL:
- Listar os dados da tabela genero.

```
SELECT idgenero,descricao  
FROM genero;
```

```
SELECT *  
FROM genero;
```

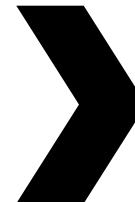


idgenero	descricao
1	Aventura
2	Ficção
3	Suspense
4	Romance
5	Infantil
6	Auto-Ajuda
7	Infanto-juvenil
8	Adulto
9	Autobiografia
10	Biografia
11	Fantasia
12	Científico
13	Poesia
14	Sátira
15	Fábula
16	Novela
17	Conto
18	Crônica
19	Ensaio
20	Autobiografia
21	Biografia

Linguagem de Consulta de Dados

- Exemplo no MySQL:
- Listar a descrição da tabela genero.

```
SELECT descricao  
FROM genero;
```



descricao
Aventura
Ficção
Suspense
Romance
Infantil
Auto-Ajuda
Infanto-juvenil
Adulto
Autobiografia
Biografia
Fantasia
Científico
Poesia
Sátira
Fábula
Novela
Conto
Crônica
Ensaio
Autobiografia
Biografia

- Valores não duplicados (distintos)

```
SELECT DISTINCT column1, column2, ...
FROM table_name;
```

- Valores ordenados

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1, column2, ... ASC|DESC;
```

Linguagem de Consulta de dados

- Exemplo no MySQL: Listar as diferentes descrições existentes na tabela genero.

```
SELECT descricao FROM genero;
```

descricao
Aventura
Ficção
Suspense
Romance
Infantil
Auto-Ajuda
Infanto-juvenil
Adulto
Autobiografia
Biografia
Fantasia
Científico
Poesia
Sátira
Fábula
Novela
Conto
Crônica
Ensaio
Autobiografia
Biografia

```
SELECT DISTINCT descricao  
FROM genero;
```

descricao
Aventura
Ficção
Suspense
Romance
Infantil
Auto-Ajuda
Infanto-juvenil
Adulto
Autobiografia
Biografia
Fantasia
Científico
Poesia
Sátira
Fábula
Novela
Conto
Crônica
Ensaio

Linguagem de Consulta de dados

- Exemplo no MySQL: Listar ordenado.

```
SELECT DISTINCT descricao  
FROM genero  
ORDER BY descricao ASC;
```

descricao
Adulto
Auto-Ajuda
Autobiografia
Aventura
Biografia
Científico
Conto
Crônica
Ensaio
Fábula
Fantasia
Ficção
Infantil
Infanto-juvenil
Novela
Poesia
Romance
Sátira
Suspense

```
SELECT DISTINCT descricao  
FROM genero  
ORDER BY descricao DESC;
```

descricao
Suspense
Sátira
Romance
Poesia
Novela
Infanto-juvenil
Infantil
Ficção
Fantasia
Fábula
Ensaio
Crônica
Conto
Científico
Biografia
Aventura
Autobiografia
Auto-Ajuda
Adulto

Linguagem de Consulta de dados

- Filtros de seleção (condição)

```
SELECT column1, column2, ...
FROM table_name
WHERE condition;
```

```
SELECT column1, column2, ...
FROM table_name
WHERE NOT condition;
```

Linguagem de Consulta de dados

- Exemplo no MySQL: Listar os dados apenas dos clientes que moram no estado do Ceará.

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100	0
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	NULL	0
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30	0
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	NULL	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0

`SELECT * FROM cliente`

`WHERE estado = 'CE';`

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	NULL	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0

Linguagem de Consulta de dados

- Exemplo no MySQL: Listar os dados apenas dos clientes que NÃO moram no estado do Ceará.

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100	0
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	NULL	0
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30	0
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	NULL	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0

```
SELECT * FROM cliente
```

```
WHERE NOT estado = 'CE';
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100	0
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	NULL	0
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30	0

Linguagem de Consulta de dados

- Filtros de valor nulo

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```

Linguagem de Consulta de dados

- Exemplo no MySQL: Listar os dados apenas dos clientes cujo debito é NULO ou ausente.

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100	0
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	NULL	0
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30	0
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	NULL	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0

`SELECT * FROM cliente WHERE debito IS NULL;`

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	NULL	0
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	NULL	0

Linguagem de Consulta de dados

- Exemplo no MySQL: Listar os dados apenas dos clientes cujo debito NÃO é NULO.

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100	0
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	NULL	0
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30	0
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	NULL	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0

```
SELECT * FROM cliente WHERE debito IS NOT NULL;
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100	0
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0

Linguagem de Consulta de dados

- Combinação de múltiplos filtros (condições)
 - Utiliza combinações dos operadores AND e OR

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 AND condition3 ...;
```

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 OR condition2 OR condition3 ...;
```

```
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2 OR condition3 ...;
```

Linguagem de Consulta de dados

- Exemplo no MySQL:
- Listar apenas os clientes cujo debito não é nulo E moram no estado do Ceará.

```
SELECT * FROM cliente
WHERE debito IS NOT NULL AND estado = 'CE';
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0

- Listar apenas os clientes cujo debito não é nulo OU moram no estado do Ceará.

```
SELECT * FROM cliente
WHERE debito IS NOT NULL OR estado = 'CE';
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
1	Alberto Carlos	Rua das Casas, 01	Centro	Mossoro	RN	1990-06-12	100	0
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
4	Diego Alexandre	Rua da Estrela, 50	Orla	Natal	RN	1988-12-10	30	0
5	Ester Moreira	Av. do Trabalhador, 99	Cafu	Fortaleza	CE	1965-01-01	NULL	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0

Linguagem de Consulta de dados

- Funções de agregação

Mínimo

```
SELECT MIN(column_name)  
FROM table_name  
WHERE condition;
```

Máximo

```
SELECT MAX(column_name)  
FROM table_name  
WHERE condition;
```

Contagem

```
SELECT COUNT(column_name)  
FROM table_name  
WHERE condition;
```

Média

```
SELECT AVG(column_name)  
FROM table_name  
WHERE condition;
```

Somatório

```
SELECT SUM(column_name)  
FROM table_name  
WHERE condition;
```

Linguagem de Consulta de dados

- Exemplo no MySQL:

1. Existem quantos clientes cadastrados?
2. Existem quantos clientes cadastrados onde o valor do débito foi informado?

`SELECT count(*) FROM cliente;`

`SELECT count(debito) FROM cliente;`

1

count(*)
10

count(debito)
8

2

3. Listar o valor do menor e maior débito, da média do débito e o somatório do débito.

`SELECT min(debito), max(debito), avg(debito), sum(debito) FROM cliente;`

3

min(debito)	max(debito)	avg(debito)	sum(debito)
0	700	225	1800

- Filtro de funções de agrupamento

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition;
```

- Exemplo no MySQL:
- Listar o estado, o valor máximo e o valor mínimo do débito para cada estado.

```
SELECT estado, max(debito) , min(debito)
FROM cliente
GROUP BY estado;
```

estado	max(debito)	min(debito)
CE	500	170
PE	700	0
RN	100	0

Linguagem de Consulta de dados

- Exemplo no MySQL:

1. Listar o estado, media do valor do débito, e o somatório do débito para cada estado.
2. Em cada estado, existem quantos clientes cadastrados? Para quantos clientes o valor do débito foi informado?

1

```
SELECT estado, avg(debito), sum(debito)
FROM cliente
GROUP BY estado;
```

estado	avg(debito)	sum(debito)
CE	323.3333333333333	970
PE	350	700
RN	43.33333333333336	130

2

```
SELECT estado, count(*), count(debito)
FROM cliente
GROUP BY estado;
```

estado	count(*)	count(debito)
CE	4	3
PE	2	2
RN	4	3

Linguagem de Consulta de dados

- Exemplo no MySQL:
 1. Listar o estado e a media do valor do débito para cada estado.
 2. Listar o estado e a media do valor do débito para cada estado quando esta media for superior a 100.

1

```
SELECT estado, avg(debito)  
FROM cliente  
GROUP BY estado;
```

estado	avg(debito)
CE	323.3333333333333
PE	350
RN	43.33333333333336

2

```
SELECT estado, avg(debito)  
FROM cliente  
GROUP BY estado  
HAVING avg(debito) > 100;
```

estado	avg(debito)
CE	323.3333333333333
PE	350

Linguagem de Consulta de dados

- Exemplo no MySQL:

1. Listar o estado e o valor máximo do débito para cada estado onde este valor máximo for inferior a 700.
2. Listar o estado e o valor mínimo do débito para cada estado onde este valor mínimo for superior a 0.

```
SELECT estado, max(debito), min(debito)
FROM cliente
GROUP BY estado;
```

estado	max(debito)	min(debito)
CE	500	170
PE	700	0
RN	100	0

1

```
SELECT estado, max(debito)
FROM cliente
GROUP BY estado
HAVING max(debito) < 700;
```

2

```
SELECT estado, min(debito)
FROM cliente
GROUP BY estado
HAVING min(debito) > 0;
```

estado	max(debito)
CE	500
RN	100

estado	min(debito)
CE	170

Linguagem de Consulta de dados

- **Operador BETWEEN**
 - seleciona valores dentro de um intervalo.
- Exemplo no MySQL:
- Listar os clients cujo debito esta entre 500 e 1000.

```
SELECT * FROM cliente  
WHERE debito BETWEEN 500 AND 1000;
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0
10	Julio Jales	Av. Alberto Maranhao, 83	Kiwi	Olinda	PE	NULL	700	0

- **Operador LIKE:**
 - compara partes de uma sequência de caracteres
 - atributo LIKE ‘%string%’
 - % compara qualquer substring (número arbitrário de caracteres)
 - atributo LIKE ‘_string_’
 - _ compara qualquer caractere
 - comparação case-sensitive
- Observação: Alguns SGBDs usam * ao invés de %

Linguagem de Consulta de dados

- Exemplo no MySQL:
 - Listar os dados onde o nome da cidade que inicia com a letra ‘A’.

```
SELECT * FROM cliente  
WHERE cidade LIKE 'A%':
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
3	Catarina Pontes	Rua do Sol, 04	Baixo	Apodi	RN	1987-06-15	0	0
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
2	Breno Nunes	Rua da Lua, 02	Alto	Assu	RN	1975-08-10	NULL	0

- Listar os dados onde o nome do cliente termina com ‘Silva’.

```
SELECT * FROM cliente  
WHERE nome LIKE '% Silva';
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
6	Fabricio Silva	Tv. Noite Escura, 99	Pelotas	Aracati	CE	1995-03-04	300	0
7	Gabriel Silva	Rua Projetada, SN	Centro	Piriqui	CE	1980-02-08	500	0

Linguagem de Controle de dados

- Exemplo no MySQL:
 - Listar os dados onde no endereço exista a preposição ‘de’

```
SELECT * FROM cliente  
WHERE endereco LIKE '%de%';
```

codigo	nome	endereco	bairro	cidade	estado	datanasc	debito	ativo
8	Helio Cunha	Rua de Cima, 875	NULL	Lagoinha	CE	NULL	170	0
9	Ivo Ilha	Rua de Baixo, 900	Ofeia	Recife	PE	1955-08-06	0	0

- **Operadores Aritméticos:** São responsáveis pela execução de operações matemáticas simples.

+	Adição
-	Subtração
*	Multiplicação
/	Divisão

Linguagem de Controle de dados

- Exemplo no MySQL:

```
SELECT debito, debito + 10, debito - 5, debito * 0.10, debito / 10  
FROM cliente  
WHERE debito IS NOT NULL;
```

debito	debito + 10	debito - 5	debito * 0.10	debito / 10
100	110	95	10	10
0	10	-5	0	0
30	40	25	3	3
300	310	295	30	30
500	510	495	50	50
170	180	165	17	17
0	10	-5	0	0
700	710	695	70	70

- **Operadores Relacionais:** São utilizados quando precisamos fazer comparações entre dois valores.

>	Maior que
<	Menor que
=	Igual a
<>	Diferente de
>=	Maior ou igual a
<=	Menor ou igual a

Conclusão

- Conhecemos a Linguagem de Consulta de Dados.
- Aprendemos alguns dos principais parâmetros e funções que podem ser usados para realizar a consulta de dados nas tabelas do banco de dados.

■ Próxima aula

- Aprender a realizar consultas em múltiplas tabelas.



Aula 2.6.2. Linguagem de consulta de dados (Parte 2)

- ❑ Aprender a realizar consultas em múltiplas tabelas.

Consultas Aninhadas

- *Nested Queries*
- Pode-se desenvolver comandos sofisticados a partir de comandos simples, utilizando-se subconsulta (subqueries).
- Sintaxe:

```
SELECT <lista de colunas>  
  
FROM <tabela ou lista de tabelas>  
  
WHERE <coluna> operador subquery;
```

- Operadores: IN, NOT IN, EXISTS, NOT EXIST, ALL, SOME, ANY, e os operadores aritimético (>, <, =, etc).

Consultas Aninhadas

- Considere as tabelas a seguir:

DEPARTAMENTO

cod_dept	nome_dept
1	Recursos humanos
2	Informática
3	Marketing
4	Infraestrutura
5	Jurídico
6	Compras

FUNCIONARIO

id_func	nome_func	nascimento	salario	depto
1	José Silva	1979-03-12	5450.00	2
2	Maria Souza	1982-05-28	5600.00	3
3	João Nogueira	1975-04-15	6500.00	4
4	Paulo Silva Amorim	1985-10-19	5500.00	1
5	Ana Paula Almeida	1979-11-25	6500.00	4
6	Silvana Abdala	1979-05-04	5450.00	2

Consultas Aninhadas

- Exemplo no MySQL:

1. Quais os departamentos que não possuem funcionários cadastrados?

```
SELECT *
FROM departamento
WHERE cod_depto NOT IN (SELECT DISTINCT depto
                           FROM funcionario)
```

cod_depto	nome_depto
5	Jurídico
6	Compras

2. Quais os departamentos que possuem funcionários cadastrados?

```
SELECT *
FROM departamento
WHERE cod_depto IN (SELECT DISTINCT depto
                       FROM funcionario);
```

cod_depto	nome_depto
1	Recursos humanos
2	Informática
3	Marketing
4	Infraestrutura

Consultas Aninhadas

- Exemplo no MySQL:

1. Quais os funcionários que possuem salário acima da média do salário de todos os funcionários?

```
SELECT *
FROM funcionario
WHERE salario > (SELECT avg(salario)
                  FROM funcionario);
```

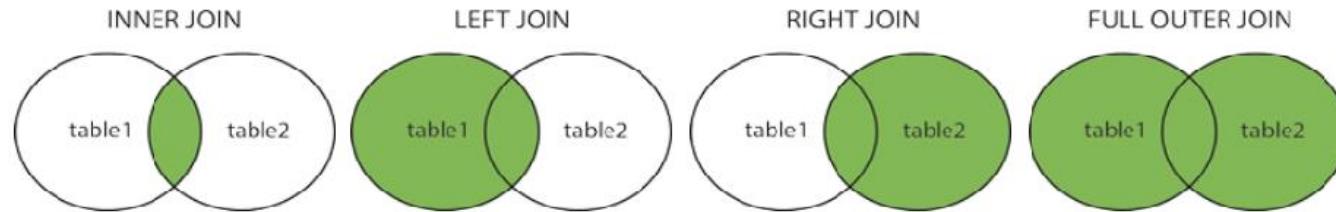
id_func	nome_func	nascimento	salario	depto
3	João Nogueira	1975-04-15	6500.00	4
5	Ana Paula Almeida	1979-11-25	6500.00	4

Junções (joins, em inglês) são consultas SQL usadas para recuperar dados de várias tabelas.

As operações de junção tomam duas relações e retornam como resultado outra relação.

Está relacionado ao conceito de FK, pois em geral, a junção se dá por meio dela.

- Existem diferentes tipos de JOINS no SQL:
 - (INNER) JOIN: Retorna registros que possuem valores correspondentes nas duas tabelas.
 - LEFT (OUTER) JOIN: Retorna todos os registros da tabela a esquerda e registros correspondentes da tabela a direita.
 - RIGHT (OUTER) JOIN: Retorna todos os registros da tabela da direita e os registros correspondentes da tabela da esquerda
 - FULL (OUTER) JOIN: Retorna todos os registros quando houver uma correspondência na tabela esquerda ou direita.



- Considere as tabelas a seguir:

DEPARTAMENTO

cod_dept	nome_dept
1	Recursos humanos
2	Informática
3	Marketing
4	Infraestrutura
5	Jurídico
6	Compras

FUNCIONARIO

id_func	nome_func	nascimento	salario	depto
1	José Silva	1979-03-12	5450.00	NULL
2	Maria Souza	1982-05-28	5600.00	3
3	João Nogueira	1975-04-15	6500.00	4
4	Paulo Silva Amorim	1985-10-19	5500.00	1
5	Ana Paula Almeida	1979-11-25	6500.00	4
6	Silvana Abdala	1979-05-04	5450.00	2
7	Jair Nogueira	1989-11-15	5300.00	NULL
8	Almir Nunes	1990-01-01	5450.00	2

(INNER) JOIN

- Retorna apenas as linhas em ambas as tabelas que possuam o campo de relacionamento com o mesmo valor.

```
SELECT column_name(s)  
FROM table1 INNER JOIN table2  
ON table1.column_name = table2.column_name
```

```
SELECT *  
FROM departamento,funcionario  
WHERE cod_depto = depto;
```

```
SELECT *  
FROM departamento INNER JOIN funcionario  
ON cod_depto = depto;
```

cod_dept	nome_dept	id_func	nome_func	nascimento	salario	depto
3	Marketing	2	Maria Souza	1982-05-28	5600.00	3
4	Infraestrutura	3	João Nogueira	1975-04-15	6500.00	4
1	Recursos humanos	4	Paulo Silva Amorim	1985-10-19	5500.00	1
4	Infraestrutura	5	Ana Paula Almeida	1979-11-25	6500.00	4
2	Informática	6	Silvana Abdala	1979-05-04	5450.00	2
2	Informática	8	Almir Nunes	1990-01-01	5450.00	2

LEFT (OUTER) JOIN

- Retorna todas as linhas da primeira tabela relacionada no join (esquerda). Quando a linha listada não possuir equivalência na tabela a direita, tais colunas aparecerão com valores nulos.

```
SELECT column_name(s)
FROM table1 LEFT JOIN table2
ON table1.column_name = table2.column_name
```

```
SELECT *
FROM departamento LEFT JOIN funcionario
ON cod_depto = depto;
```

cod_depto	nome_depto	id_func	nome_func	nascimento	salario	depto
1	Recursos humanos	4	Paulo Silva Amorim	1985-10-19	5500.00	1
2	Informática	6	Silvana Abdala	1979-05-04	5450.00	2
2	Informática	8	Almir Nunes	1990-01-01	5450.00	2
3	Marketing	2	Maria Souza	1982-05-28	5600.00	3
4	Infraestrutura	3	João Nogueira	1975-04-15	6500.00	4
4	Infraestrutura	5	Ana Paula Almeida	1979-11-25	6500.00	4
5	Jurídico	NULL	NULL	NULL	NULL	NULL
6	Compras	NULL	NULL	NULL	NULL	NULL

RIGHT (OUTER) JOIN

- Retorna todas as linhas da tabela relacionada no join (direita). Quando a linha listada não possuir equivalência na tabela a esquerda, tais colunas aparecerão com valores nulos.

```
SELECT column_name(s)
FROM table1 RIGHT JOIN table2
ON table1.column_name = table2.column_name
```

```
SELECT *
FROM departamento RIGHT JOIN funcionario
ON cod_depto = depto;
```

cod_depto	nome_depto	id_func	nome_func	nascimento	salario	depto
NULL	NULL	1	José Silva	1979-03-12	5450.00	NULL
3	Marketing	2	Maria Souza	1982-05-28	5600.00	3
4	Infraestrutura	3	João Nogueira	1975-04-15	6500.00	4
1	Recursos humanos	4	Paulo Silva Amorim	1985-10-19	5500.00	1
4	Infraestrutura	5	Ana Paula Almeida	1979-11-25	6500.00	4
2	Informática	6	Silvana Abdala	1979-05-04	5450.00	2
NULL	NULL	7	Jair Nogueira	1989-11-15	5300.00	NULL
2	Informática	8	Almir Nunes	1990-01-01	5450.00	2

FULL (OUTER) JOIN

- Retorna todas as linhas quando há uma correspondência nos registros da tabela esquerda ou direita.

```
SELECT column_name(s)
FROM table1 FULL JOIN table2
ON table1.column_name = table2.column_name
```

SELECT *

```
FROM departamento FULL JOIN funcionario
ON cod_depto = depto;
```

cod_depto	nome_depto	id_func	nome_func	nascimento	salario	depto
3	Marketing	2	Maria Souza	1982-05-28	5600.00	3
4	Infraestrutura	3	João Nogueira	1975-04-15	6500.00	4
1	Recursos humanos	4	Paulo Silva Amorim	1985-10-19	5500.00	1
4	Infraestrutura	5	Ana Paula Almeida	1979-11-25	6500.00	4
2	Informática	6	Silvana Abdala	1979-05-04	5450.00	2
2	Informática	8	Almir Nunes	1990-01-01	5450.00	2

- ✓ Aprendemos a realizar consultas em múltiplas tabelas.
 - Conhecemos os diferentes tipos de junções (JOINS).
 - Entendemos como podemos usar subconsultas ou consultas aninhadas.

- Usar o MySQL Workbench para realizar algumas consultas a dados utilizando a linguagem DQL da SQL.



Aula 2.7. MySQL Workbench: DQL

- ❑ Usar o MySQL Workbench para realizar alguns comandos de consulta de dados.

Conclusão

- ✓ Conhecemos as funcionalidades do MySQL Workbench para consultar dados do banco de dados utilizando a linguagem DQL do SQL.

■ Próxima aula

- Conhecer a linguagem de controle de acesso a dados.



Aula 3.13. Linguagem de controle de acesso a dados

- ❑ Conhecer a linguagem de controle de acesso a dados.

- *Data Control Language (DCL)*
- Manter a segurança interna do banco de dados e seus objetos de dados protegendo contra acessos não autorizados.
- Permite conceder e revogar permissões de acesso a objetos do banco de dados para usuários.
 - GRANT - conceder
 - REVOKE - revogar
- Criar usuário e papéis (roles).
 - CREATE USER/ROLE
 - ALTER USER/ROLE

- CREATE ROLE: Cria uma nova role (papel).

```
CREATE ROLE <role>[,<lista de roles>];  
  
CREATE ROLE 'developer';  
CREATE ROLE 'dba', 'read', 'write','read_write';
```

- DROP ROLE: Exclui uma role.

```
DROP ROLE <role>[,<lista de roles>];  
  
DROP ROLE 'developer';  
DROP ROLE 'dba', 'read', 'write','read_write';
```

- CREATE USER: Cria um novo usuário.

```
CREATE USER <usuario>[@<hostname>]
```

```
IDENTIFIED BY P@ssw0rd;  
CREATE USER 'usuario1' IDENTIFIED BY 'userpw1';
```

```
CREATE USER 'usuario2'@'localhost' IDENTIFIED BY 'userpw2';
```

- DROP USER: Exclui um usuário.

```
DROP USER <usuario>;  
DROP USER 'usuario1';
```

```
DROP USER 'usuario2'@'localhost';
```

- GRANT: concede privilégios para usuário/role.

```
GRANT <lista de privilegios> ON <objeto>
TO <lista de usuários/roles>
[WITH GRANT OPTION]
[GRANTED BY grantor];
```

- Lista de usuários e roles: pode ser um conjunto separado por vírgulas ou PUBLIC.
- O garantidor (grantor) é o usuário corrente (CURRENT_USER) ou o papel (CURRENT_ROLE), é opcional no comando.
- WITH GRANT OPTION → Pode propagar a permissão.

- REVOKE: revoga privilégios para usuário/role.

```
REVOKE [GRANT OPTION FOR]
<lista de privilegios> ON objeto
FROM <usuários/roles> [RESTRICT | CASCADE] ;
```

DCL – Permissões

- Exemplo no MySQL:
 - Concedendo permissões para roles e usuários:

```
GRANT ALL PRIVILEGES ON exemplo TO 'dba';
GRANT SELECT ON exemple.* TO 'read';
GRANT INSERT, UPDATE, DELETE ON exemple.* TO 'write';
GRANT SELECT, INSERT, UPDATE, DELETE ON exemple.* TO 'read_write';

GRANT 'read_write' TO 'developer';
GRANT CREATE, DROP, REFERENCES, ALTER, EXECUTE, CREATE VIEW, TRIGGER ON *.* TO developer;

SHOW GRANTS FOR 'developer';

GRANT SELECT, INSERT, UPDATE, DELETE ON exemple.* TO 'usuario1';
GRANT 'developer' TO 'usuario2'@'localhost';

GRANT INSERT ON example.* TO 'usuario3'@'localhost';
GRANT 'read' TO 'usuario3'@'localhost';
```

- Exemplo no MySQL:
 - Revogando permissões de roles e usuários:

```
REVOKE DELETE FROM 'read_write';
```

```
REVOKE 'read' FROM 'usuario3'@'localhost';
```

```
REVOKE INSERT, UPDATE, DELETE ON exemplo.* FROM 'usuario1';
```

- ✓ Conhecemos os principais comandos da linguagem de controle de acesso a dados.

■ Próxima aula

- ❑ Conhecer os conceitos fundamentais de data warehouse.



Aula 2.9. Coleta de dados estruturados: Exemplo utilizando a linguagem R

- ❑ Exemplo prático de coleta de dados estruturados utilizando a linguagem R e o SGBD MySQL.

Coleta de dados Estruturados com R

- Leitura e escrita de dados no SGBD MySQL utilizando a linguagem R.
 - MySQL versão 8
 - Pacotes R:



- ✓ Exemplo prático de coleta de dados estruturados utilizando a linguagem R e o SGBD MySQL.

- ❑ Exemplo prático de coleta de dados estruturados utilizando a linguagem Python e o SGBD MySQL.



Aula 2.10. Coleta de dados estruturados: Exemplo utilizando a linguagem Python

- ❑ Exemplo prático de coleta de dados estruturados utilizando a linguagem Python e o SGBD MySQL.

Coleta de dados Estruturados com Python

- Leitura e escrita de dados no SGBD MySQL utilizando a linguagem Python.
 - MySQL versão 8
 - Pacotes Python:



- ✓ Exemplo prático de coleta de dados estruturados utilizando a linguagem Python e o SGBD MySQL.

- ❑ Exemplo prático de coleta de dados estruturados utilizando a Plataforma Knime e o SGBD MySQL.



Aula 2.11. Coleta de dados estruturados: Exemplo utilizando a Plataforma Knime

- ❑ Exemplo prático de coleta de dados estruturados utilizando a Plataforma Knime e o SGBD MySQL.

Coleta de dados Estruturados com Knime

IGTI

- Leitura e escrita de dados no SGBD MySQL utilizando a Plataforma Knime Analytics.
 - MySQL versão 8



Conclusão

- ✓ Exemplo prático de coleta de dados estruturados utilizando a Plataforma Knime e o SGBD MySQL.

- ❑ Exemplo prático de coleta de dados estruturados utilizando a linguagem Python e o SGBD MySQL.



Coleta e Obtenção de Dados

Capítulo 3: Coleta de Dados em Bancos de Dados NoSQL

Profa. Fernanda Farinelli



Aula 3.1. Fundamentos dos SGBDs NoSQL

- Entender o que são bancos de dados NoSQL.
- Conhecer algumas ferramentas disponíveis.

Motivação ao NoSQL

Volume



Variedade



Velocidade



Escalabilidade de
armazenamento e
processamento

Modelos flexíveis de
armazenamento

Elevado grau de
disponibilidade

Consulta com baixa
latência

NoSQL é um termo genérico que define bancos de dados não-relacionais

NoSQL
Not Only SQL

Fornecem recursos eficientes para armazenamento de grandes volumes de dados estruturados e não-estruturados

Características

Escalabilidade Horizontal

- Consiste em aumentar o número de máquinas disponíveis.
- Inviável em modelos relacionais devido a concorrência.
- Nos modelos NoSQL não existe bloqueios, esse tipo de escalabilidade é a mais viável.

Esquema flexível

- Permite uma fácil aplicação da escalabilidade e também um aumento na disponibilidade dos dados.
- Devido a essa ausência, não há garantia da integridade dos dados.

Características

Suporte a replicação

- Permitem a replicação de uma forma nativa o que provém uma escalabilidade maior e também uma diminuição do tempo gasto para a recuperação de informações.

Nem sempre consistentes

- Os bancos de dados NoSQL nem sempre conseguem se manter consistentes

Teorema CAP

Consistência (consistency)

- todos os clientes enxergam os mesmos dados

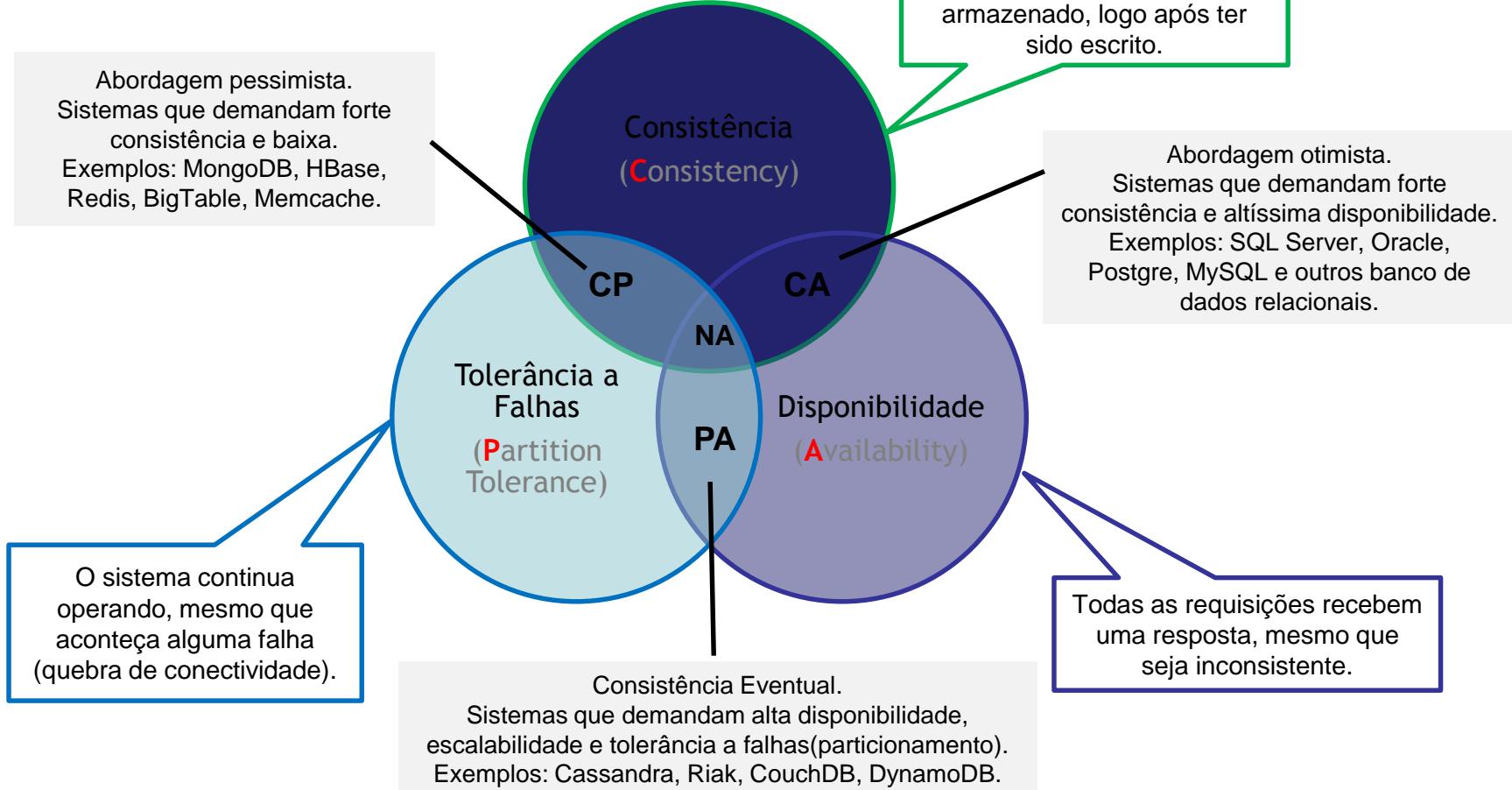
Disponibilidade
(availability)

- todos os clientes acessam alguma versão dos dados

Tolerância a Partição
(partition tolerance)

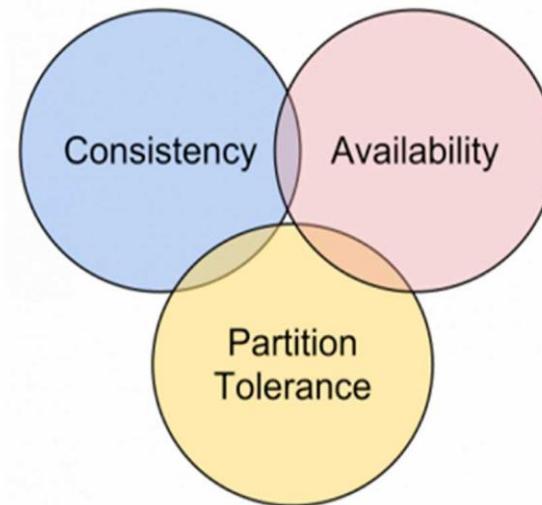
- o banco pode ser dividido em múltiplos servidores

Teorema CAP

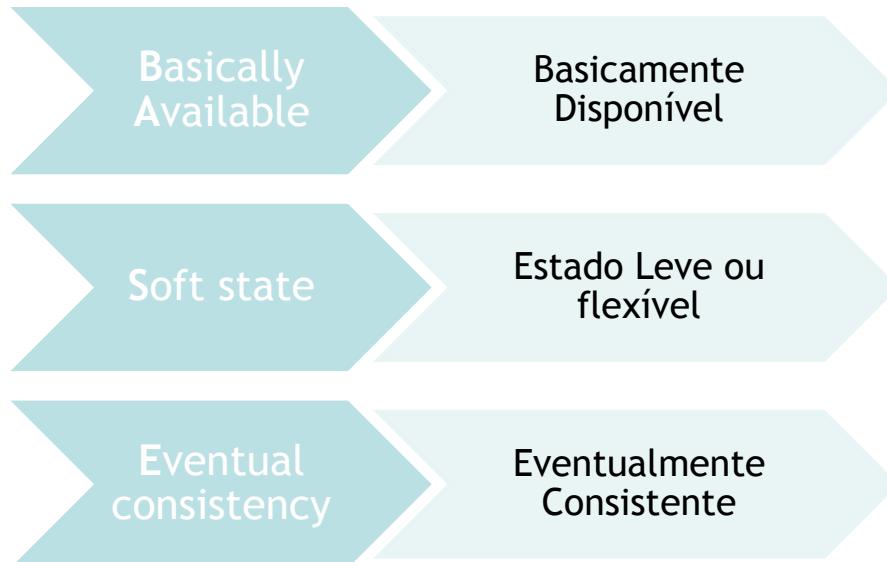


Teorema CAP

- Não é possível garantir essas três propriedades ao mesmo tempo.
- Só possível garantir quaisquer duas dessas propriedades ao mesmo tempo.



Propriedades BASE



Uma aplicação funciona **basicamente todo o tempo, não tem de ser consistente todo o tempo** e o sistema torna-se **consistente no momento devido**.

ACID versus BASE

ACID	BASE
Consistência forte	Fraca consistência
Isolamento	Foco em Disponibilidade
Concentra-se em "commit"	Melhor esforço
Transações aninhadas	Respostas aproximadas
Disponibilidade	Mais simples e mais rápido
Conservador (pessimista)	Agressivo (otimista)
Evolução difícil (por exemplo, esquema)	Evolução mais fácil

Relacional versus NoSQL

Use RDBMS quando...	Use NoSQL quando...
Suas aplicações forem centralizadas (ERP, CRM)	Suas aplicações forem descentralizadas (Web, Mobile, Big Data, IoT)
Alta disponibilidade moderada for necessária	Quando a disponibilidade tiver que ser contínua, sem interrupção
Dados gerados em velocidade moderada	Dados gerados em alta velocidade (sensores)
Dados forem gerados a partir de poucas fontes	Dados forem gerados a partir de múltiplas fontes
Dados forem estruturados	Dados forem semi ou não-estruturados
Transações complexas	Transações simples
For necessário manter moderado volume de dados	For necessário manter alto volume de dados

Fonte: <http://datascienceacademy.com.br/blog/quando-utilizar-rdbms-ou-nosql/>

- Entendemos o que é o SGBD NoSQL.
- Conhecemos as propriedades BASE (Basically Available, Soft state, Eventual consistency) e o teorema CAP (consistency, availability, partition tolerance).

■ Próxima aula

- ❑ Conhecer os tipos ou categorias de SGBDs NoSQL.

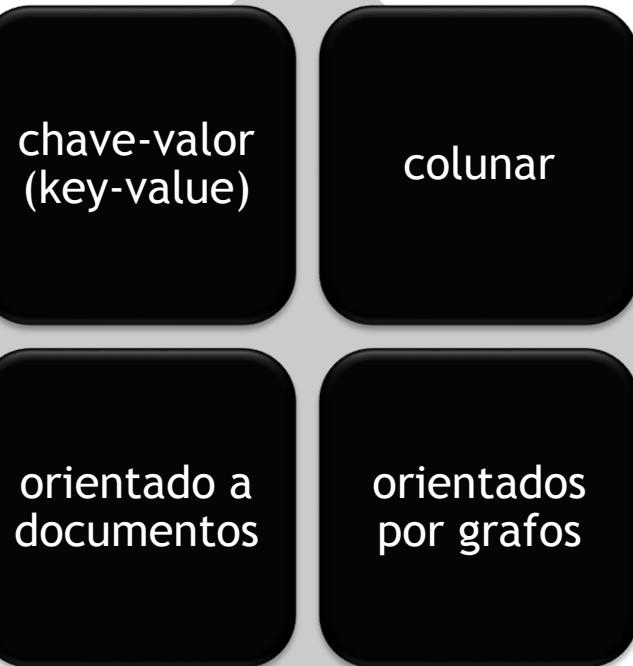


Aula 3.2. Categorias de SGBDs NoSQL

- Conhecer os tipos de SGBDs NoSQL.
- Conhecer algumas ferramentas disponíveis.

Categorias de Bancos de Dados NoSQL

Seu esquema não é orientado a tabelas

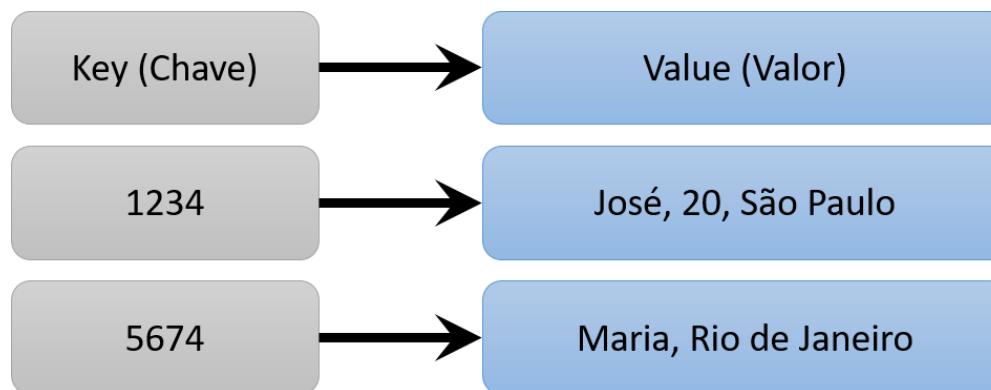


Armazenamento por chave-valor (*key-value*)

Armazenamento sem esquema definido

Dados não-estruturados ou estruturados

Dado ou valor acessado por chave única



SGBDs NoSQL chave-valor



Amazon DynamoDB



ORACLE®
NOSQL
DATABASE

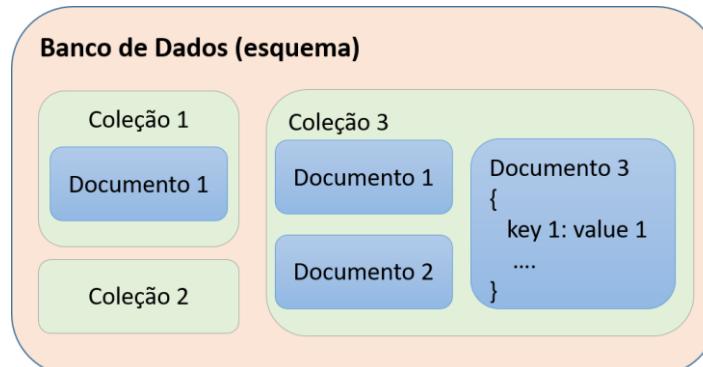


Armazenamento orientado a documentos

Estrutura de dados como coleções de documentos

Documentos são conjuntos de atributos e valores

São baseados em JSON ou XML.



SGBDs NoSQL orientado a documentos

IGTI

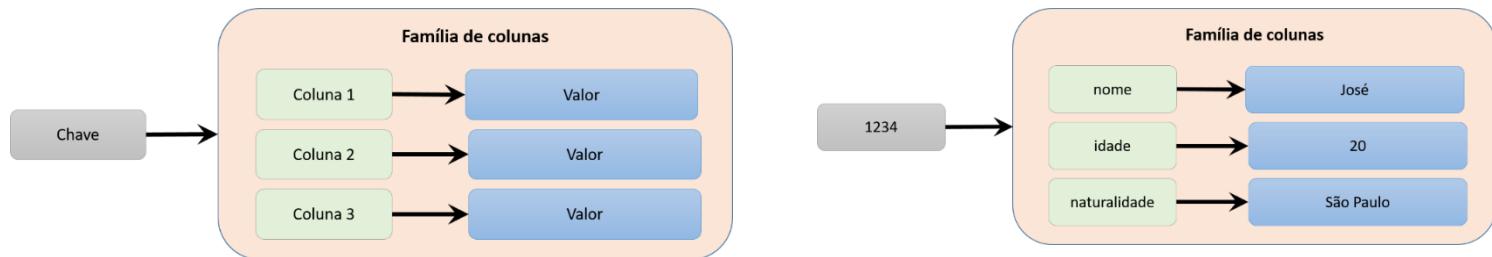


Armazenamento em coluna ou colunar

Estrutura de dados como colunas pré-definidas

A chave aponta para um conjunto de colunas

Famílias de colunas e arrays de colunas



SGBDs NoSQL colunar

IGTI



HYPERTABLE^{INC}



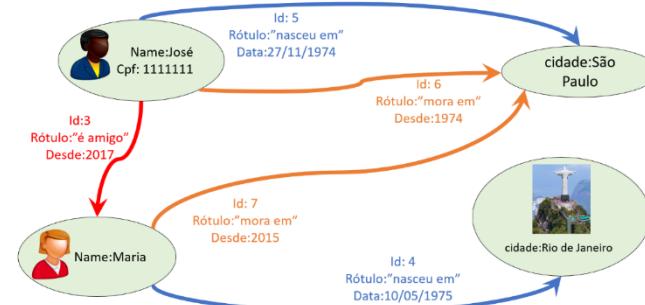
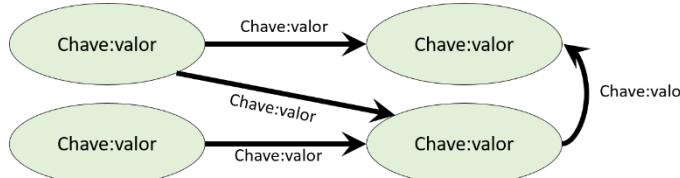
BigTable

Armazenamento orientados por grafos

Estruturas de dados conforme a teoria dos grafos

Os nós armazena dados, as arestas armazenam relacionamentos e ambos possuem atributos

Atributos format chave-valor

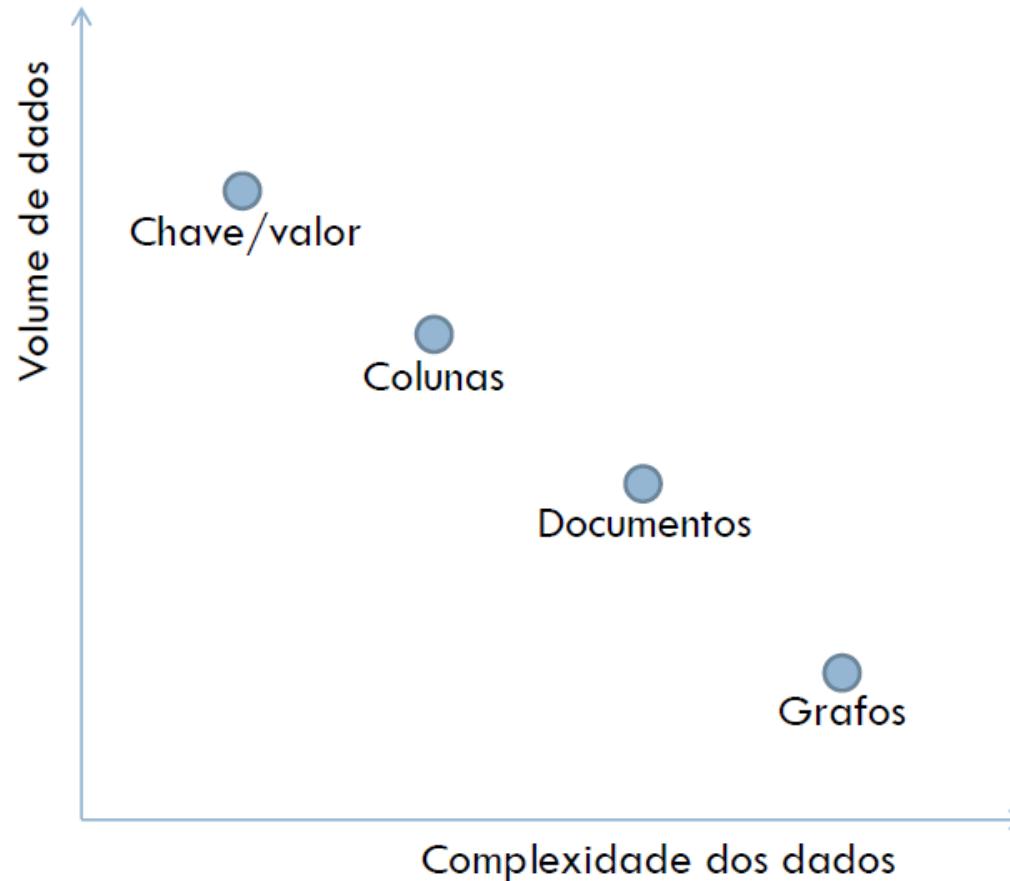


SGBDs NoSQL orientados por grafos

IGTI



Quando utilizar cada modelo?



- ✓ Conhecemos as diferentes categorias de SGBD NoSQL e os SGBDs disponíveis para cada categoria.

■ Próxima aula

- ❑ Conhecer um SGBD NoSQL orientado a documentos.



Aula 3.3. Prática: Bancos de dados documento

- Conhecer um banco de dados orientado por documentos:
 - MongoDB.

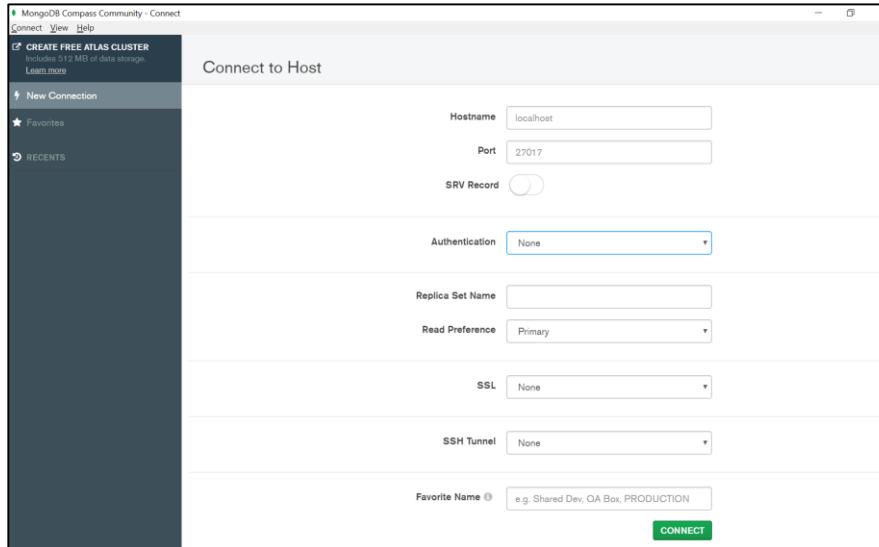
MongoDB Server

- Download:
 - <https://www.mongodb.com/try/download/community>
- Manual de instalação:
 - <https://docs.mongodb.com/manual/installation/>
- Para conhecer todos os commandos CRUD acesse:
 - <https://docs.mongodb.com/manual/crud/>
 - CRUD (Create, Read, Update, Delete)
 - Em português é Inserir, Consultar, Alterar e Excluir
- Comparação comandos MongoDB X comandos SQL:
 - <https://docs.mongodb.com/manual/reference/sql-comparison/>

- Ver disciplina: Persistência de Dados em Bancos NoSQL - NSQ201A
 - Professor: Charles Wellington de Oliveira Fortes
- Aula 2.11
 - Início: 0:00:11
 - Fim: 0:18:08

MongoDB Compass

- Gerenciador gráfico.
- Download e instalação:
 - <https://www.mongodb.com/products/compass>



- ✓ Conhecemos na prática um bancos de dados orientado por documentos: MongoDB.

- Prática: Bancos de dados chave/valor



Aula 3.4. Prática: Bancos de dados chave/valor

- Conhecer um banco de dados NoSQL do tipo chave/valor:
 - Redis.

- Ver disciplina: Persistência de Dados em Bancos NoSQL - NSQ201A
 - Professor: Charles Wellington de Oliveira Fortes
- Aula 2.10
 - Início: 0:01:44
 - Fim: 0:22:13

Conclusão

- ✓ Conhecemos na prática um bancos de dados NoSQL do tipo chave/valor: Redis.

- Prática: Bancos de dados colunar



Aula 3.5. Prática: Bancos de dados colunar

- Conhecer um bancos de dados NoSQL do tipo colunar:
 - Cassandra.

- Ver disciplina: Persistência de Dados em Bancos NoSQL - NSQ201A
 - Professor: Charles Wellington de Oliveira Fortes
- Aula 2.12
 - Início: 0:00:23
 - Fim: 0:22:19

Conclusão

- ✓ Conhecemos na prática um bancos de dados NoSQL do tipo colunar: Cassandra.

- Prática: Bancos de dados de grafos



Aula 3.6. Prática: Bancos de dados de grafos

- Conhecer um banco de dados NoSQL do tipo orientado por grafos:
 - Neo4J.

- Ver disciplina: Persistência de Dados em Bancos NoSQL - NSQ201A
 - Professor: Charles Wellington de Oliveira Fortes
- Aula 2.12
 - Início: 0:00:23
 - Fim: 0:22:19

- ✓ Conhecemos na prática um bancos de dados NoSQL do tipo orientado por grafos: Neo4J.

- Coleta de Dados no MongoDB: Exemplo utilizando a linguagem R



Aula 3.7. Coleta de dados NoSQL: Exemplo utilizando a linguagem R

- ❑ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a linguagem R e o SGBD MongoDB.

- Leitura e escrita de dados no SGBD MongoDB utilizando a linguagem R.
 - Pacotes R:



Conclusão

- ✓ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a linguagem R e o SGBD MongoDB.

- ❑ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a linguagem Python e o SGBD MongoDB.



Aula 3.8. Coleta de dados NoSQL: Exemplo utilizando a linguagem Python

- ❑ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a linguagem Python e o SGBD MongoDB.

Coleta de dados NoSQL com Python

IGTI

- Leitura e escrita de dados no SGBD MongoDB utilizando a linguagem Python.
 - Pacotes Python:



- ✓ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a linguagem Python e o SGBD MongoDB.

- ❑ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a Plataforma Knime e o SGBD MongoDB.



Aula 3.9. Coleta de dados estruturados: Exemplo utilizando a Plataforma Knime

- ❑ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a Plataforma Knime e o SGBD MongoDB.

- Leitura e escrita de dados no SGBD MongoDB utilizando a Plataforma Knime Analytics.



- ✓ Exemplo prático de coleta de dados em bases de dados NoSQL utilizando a Plataforma Knime e o SGBD MongoDB.

■ Próxima aula

- Coleta e obtenção de dados na web.



Coleta e Obtenção de Dados

Capítulo 4: Coleta de Dados na web

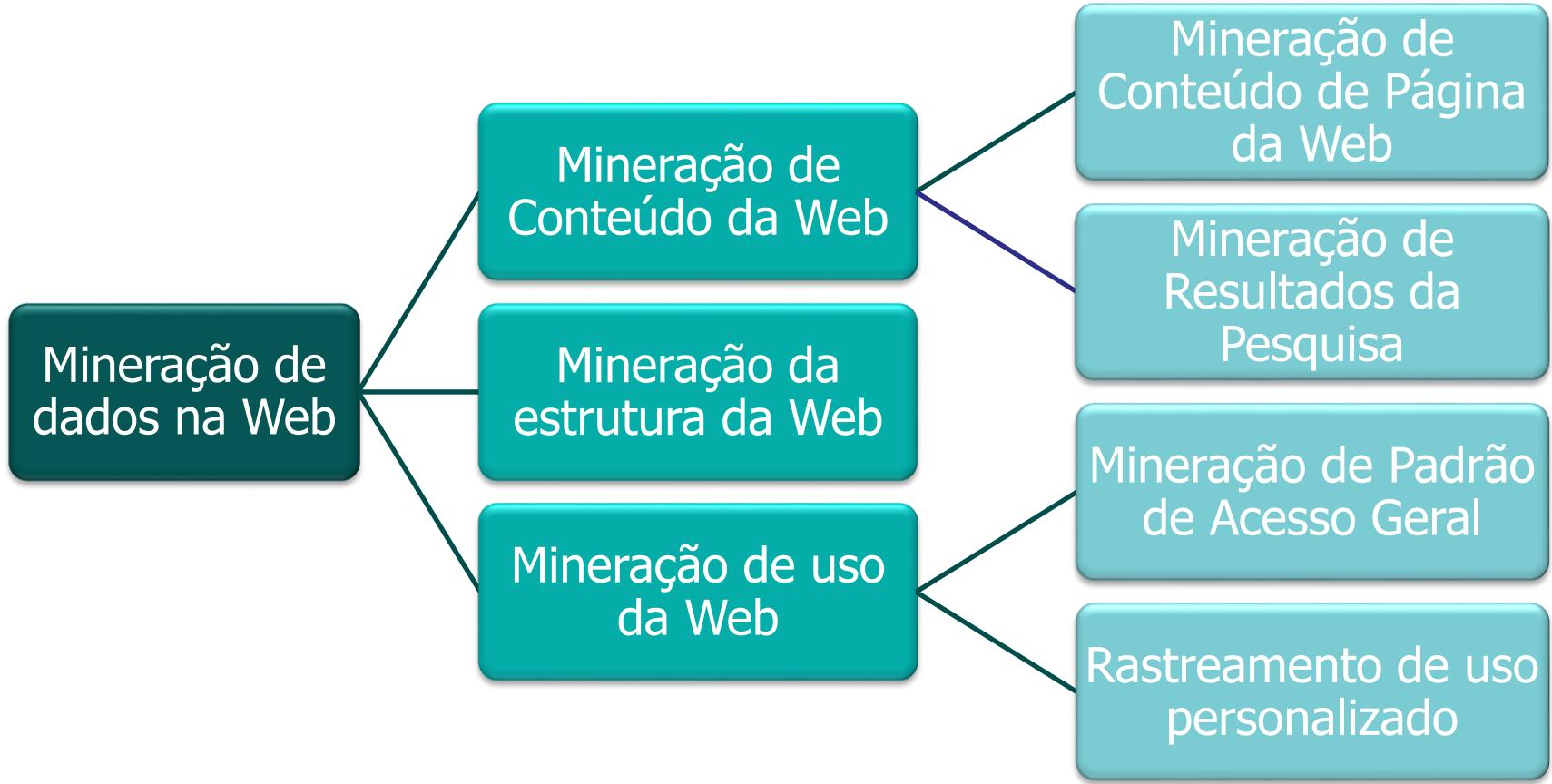
Profa. Fernanda Farinelli



Aula 4.1. Fundamentos sobre coleta de dados na web

- Compreender a importância da coleta de dados da web.
- Entender as principais abordagens de análises de dados aplicadas aos dados coletados na web.

- Aplicação de técnicas de mineração que permite a extração de dados de sites da web.
 - Web: Enorme quantidade de dados, mas pouco conhecimento
 - Web Mining: Descobrir este conhecimento para melhorar eficiência de diversas aplicações da Web.
- Desafios:
 - Diversidade de fontes de documentos (portais de conteúdo, linked open data, mídias sociais, etc).
 - Diversidade dos documentos (tamanho, formato, estrutura).
 - Documentos dinâmicos.



- Algumas abordagens de análise de dados aplicadas aos dados coletados da web:
 - Clusterização ou agrupamento
 - Regras de Associação
 - Classificação
 - Filtragem
 - Análise estatística
 - Padrões sequenciais

Clusterização ou Agrupamento

- Características:
 - Grupos de usuários: identifica grupos de usuários com padrões de navegação semelhantes.
 - Grupos de páginas: identifica grupos de páginas tendo conteúdo semelhante.
- Utilidade:
 - Inferir dados demográficos para segmentação de mercado em e-commerce.
 - Sites de busca: resultados de acordo com as consultas do usuário ou históricos de necessidades de informação.
 - Personalização.



Regras de Associação

- Pode relacionar páginas que são acessadas com mais frequência em uma mesma sessão.
 - Exemplos:
 - Relação entre usuários que visitaram uma página de produtos eletrônicos e aqueles que visitaram uma página de equipamentos esportivos.
- Utilidade:
 - Aplicações de negócios e comércio.
 - Reestruturação de web sites.
 - Pré-recuperação de documentos para reduzir a latência percebida pelo usuário para carregar página de um site remoto.

Regras de Associação

- Exemplo:
 - X% dos usuários que pesquisaram determinado assunto/produto também visitaram uma página de e-commerce específica.
 - X% dos clientes que acessam o link download também acessam o link de preços de produtos.

Se (visitou página A) **então** (visitou página B)

Se (visitou página X) **e** (visitou página Y) **então** (visitou página Z)

Se (visitou página X) **então** (visitou página Y) **e** (visitou página Z)

Se (acessou do Brasil) **então** (visitou página Y)

- Utilizando a representação adequada:

página A => página B

página X, página Y => página Z

página X => página Y, página Z

Brasil => página Y

Análises estatísticas

- Método mais comum.
- Análises estatísticas descritivas: média, frequência, mediana..
- Variáveis de análise: visões de páginas, tempo de permanência, comprimento do caminho de navegação, itens clicados.
- Utilidade:
 - Melhorar desempenho e navegabilidade do sistema.
 - Facilitar tarefas de modificação do site.
 - Apoiar decisões de mercado.
- Exemplo:
 - Qual a área do site mais acessada por usuários por turno: manhã, tarde e noite?



Classificação

- Características:
 - Identifica atributos que melhor descrevem as características de uma dada classe.
- Exemplo:
 - 30% dos usuários que compraram um ebook estão na faixa de 25-35 anos e moram na região sudeste.
 - X% dos usuários do facebook curtiram um post do assunto Y são fãs da banda W.
- Utilidade:
 - Permite definir perfis de usuários que pertencem a uma classe ou categoria.
 - Ao entrar no site o cliente é classificado em um dos possíveis padrões de página, baseado no seu perfil de acesso.

Conclusão

- Percebemos a importância do dados disponíveis na web.
- Conhecemos algumas abordagens de análises de dados para dados coletados na web.

■ Próxima aula

- Discutir as características das APIs para Mídias Sociais.



Aula 4.2. APIs para Mídias Sociais

Nesta aula

- ❑ Discutir a utilização de APIs para obtenção de conteúdo de mídias sociais.

Uso de APIs para coleta de dados

- APIs → forma mais indicada para a obtenção de dados a partir das mídias sociais.
- APIs → permite a criação de aplicações, em diferentes linguagens e sistemas operacionais, para acessar um conjunto de serviços disponibilizados pela plataforma:
 - publicar conteúdo
 - coletar conteúdo
 - coletar dados dos usuários: compartilhamentos, interesses, dados pessoais
 - outras ações específicas de cada mídia social

Uso de APIs para coleta de dados

Cada plataforma de mídia social possui APIs com características próprias, e que podem variar ao longo do tempo.

Alguns serviços podem também estar disponíveis apenas para contas com pagamento de assinatura.

Devem ser respeitadas as políticas da plataforma, para a utilização dos dados obtidos.

Atualmente as APIs foram modificadas para atender aos requisitos de privacidade da GDPR e LGPD.

API's de coleta de dados em mídias sociais

IGTI

- Principais API's para redes sociais:

- Twitter - <https://developer.twitter.com/en>



- Linkedin - <https://www.linkedin.com/developers/apps>



- Facebook - https://developers.facebook.com/docs/graph-api?locale=pt_BR

- Instagram - <https://www.instagram.com/developer/register/>

- Necessário ter uma conta para criar aplicativos e chaves de acesso.

Leitura recomendada: *Como criar Apps e obter os tokens necessários para coletar dados do Twitter, Instagram, Linkedin e Facebook*
Disponível em: <https://medium.com/@marlessonssantana/como-criar-apps-e-obter-os-tokens-necess%C3%A1rios-para-coletar-dados-do-twitter-instagram-linkedin-e-8f36602ea92a> - Acesso em: 12/04/2020

- O Twitter pode ser descrito como um serviço de microblog em tempo real, fácil de usar, e que permite aos usuários publicar atualizações curtas de *status*, chamadas *Tweets*.
- O Twitter disponibiliza APIs para publicação e acesso a conteúdo, monitoramento de Tweets em tempo real, e integração com anúncios, dentre vários outros serviços.
- O protocolo OAuth é utilizado para a conexão com o Twitter para que seja possível envio de requisições à API.



Requisitos para usar a API do Twitter

- A API do Twitter é intuitiva e fácil de usar. A documentação completa pode ser encontrada em:

<https://dev.twitter.com/rest/public>

- Para utilizar a API, inicialmente é necessário:
 - possuir uma conta de usuário do Twitter.
 - realizar o cadastro como desenvolvedor.
 - criar uma aplicação no Twitter – ela terá a autorização para o acesso aos dados de nossa conta através da API através do protocolo OAuth.



- **Name** → nome (de exibição ou da conta) é um identificador pessoal exibido na página do perfil e usado para identificar uma conta.
- **username** → nome de usuário (ou identificador) é como você é identificado no Twitter. É sempre precedido pelo símbolo @.
- **Tweet** → conteúdo que pode incluir texto, entidades (menções, *hashtags*, URL e mídia) e lugares (localização no mundo real). Tudo isto sem ultrapassar 140 caracteres.
- **retweet (s.)** → Um Tweet que você encaminha para seus seguidores é conhecido como Retweet.
- **Timeline** → é uma coleção de tweets ordenada cronologicamente. Ela pode ser de um usuário particular, ou corresponder ao resultado de um filtro.

Terminologia do Twitter

- **Trends** (ou Trending Topics – TT) → são os assuntos que estão sendo falados no momento atual, mais do que no passado, e são obtidos automaticamente por um algoritmo do Twitter.
- **# (hashtag)** → qualquer palavra ou frase imediatamente precedida pelo símbolo #. Quando você clica ou toca em uma hashtag, vê outros Tweets contendo a mesma palavra-chave ou tópico.
- **@** → sinal usado para destacar nomes de usuários em Tweets.
- **Menção** → O ato de mencionar outras contas no seu Tweet incluindo o sinal @ seguido diretamente pelo nome de usuário é chamado de "menção".
- Para mais termos acesse: <https://help.twitter.com/pt/glossary>

Conclusão

- Discutimos a utilização de APIs para obtenção de conteúdo de mídias sociais.
- Identificamos algumas APIs de mídias sociais.

- Prática de coleta de dados no twitter utilizando a linguagem R e a api do twitter.



Aula 4.3. Coleta de dados no twitter: Exemplo utilizando a linguagem R

- ❑ Exemplo prático de coleta de dados no twitter utilizando a linguagem R.

Credenciais para utilização da API do Twitter

- As informações que serão necessárias para a autorização na utilização da API do Twitter são:
 - Consumer Key (API Key)
 - Consumer Secret (API Secret)
 - Access Token
 - Access Token Secret
- Estas informações não devem ser disponibilizadas a terceiros.
- A versão grátila da API impõe limitações, como por exemplo o número de tweets que pode ser recuperados por chamada e dentro de 15 minutos.



Twitter API: Dados de acesso

The screenshot shows the Twitter Developer Keys and tokens page for an app named 'ExemploIGTI'. The 'Keys and tokens' tab is selected. A red box highlights the 'Consumer API keys' section, which contains fields for 'API key:' and 'API secret key:', both of which are blurred. A red arrow points from a text box on the left to this section. Another red box highlights the 'Access token & access token secret' section at the bottom, which contains a note: 'This is where you can create a token to auth into your Twitter account.' A 'Generate' button is visible in this section.

As informações de Consumer Key e Consumer Secret estarão disponíveis com a criação da APP

Important notice about your access token and access token secret

Looking for your secret token? For security, API tokens are only displayed once. You will need to regenerate access tokens for previously authenticated apps. To learn more, click here.

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

Regenerate

API key:

API secret key:

Access token & access token secret

Generate

This is where you can create a token to auth into your Twitter account.

Twitter API: Dados de acesso

IGTI

The screenshot shows the Twitter Developer Keys and tokens page for the app "ExemploIGTI". The "Keys and tokens" tab is selected. A notice states that API tokens are only displayed once and must be regenerated. The "Consumer API keys" section shows blurred API key and secret values with a "Regenerate" button. A red arrow points to the "Generate" button in the "Access token & access token secret" section, which contains a note about creating a token to auth into your Twitter account.

Developer Use cases Products Docs More Labs App

Apps > ExemploIGTI

App details Keys and tokens Permissions

Important notice about your access token and access token secret

Looking for your secret token? For security, API tokens are only displayed once. You will need to regenerate access tokens for previously authenticated apps. To learn more, read our [FAQ](#).

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys

API key: [REDACTED]

API secret key: [REDACTED]

Regenerate

Access token & access token secret

This is where you can create a token to auth into your Twitter account.

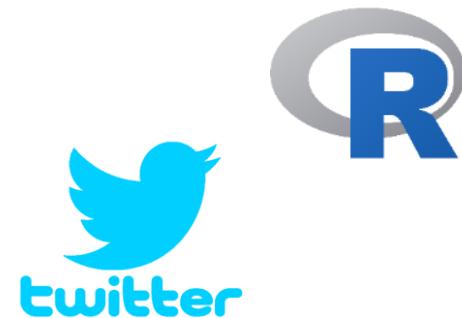
Generate

Coleta de dados no twitter com R

- Existem 2 bibliotecas no R para isso:
 - Rtweet: <https://cran.r-project.org/web/packages/rtweet/index.html>
 - twitteR: <https://cran.r-project.org/web/packages/twitteR/index.html>
 - Está sendo descontinuado
- Instalar os pacotes no R Studio usando os comandos abaixo:

```
install.packages("rtweet")
```

```
install.packages("twitteR")
```



Coleta de Dados no Twitter com R (por usuário)

IGTI

The screenshot shows an RStudio interface with a red box highlighting the code editor area. The code is written in R and demonstrates how to authenticate with the Twitter API, search for tweets by a specific user, and plot the frequency of tweets over time.

```
## Documentação do rtweet https://cran.r-project.org/web/packages/rtweet/rtweet.pdf
library(rtweet) #Usa a API do twitter

setwd("C:\\Users\\[REDACTED]\\Desktop\\TwitterData") #Define diretório de trabalho

#Autenticação
token <- create_token(app = "Primeiro Exemplo IGTI", consumer_key = "[REDACTED]",
                      consumer_secret = "[REDACTED]",
                      access_token = "[REDACTED]",
                      access_secret = "[REDACTED]")

#Parametros de busca
screen_name <- "jairbolsonaro"

#Buscando
tweets <- get_timeline(screen_name, n=1000, include_rts=TRUE,exclude_replies=TRUE)

#Salvando o vetor de tweets como CSV e apenas o Texto em TXT na codificação do português
write_as_csv(tweets, "TweetsByNameRawData.csv", fileEncoding = "latin1//TRANSLIT")

## plota a serie temporal dos tweets
ts_plot(tweets, "3 hours") +
  ggplot2::theme_minimal() +
  ggplot2::theme(plot.title = ggplot2::element_text (face = "bold")) +
  ggplot2::labs (
    x = NULL, y = NULL,
    title = "Frequencia de tweets da conta 'jairbolsonaro' nos ultimos 9 dias",
    subtitle = "Contagem de tweets agregados em intervalos de 3 horas",
    caption = "\nFonte: Dados coletados do Twitter com o pacote rtweet"
)
```

The RStudio interface includes a top menu bar, a sidebar with tabs for PlotMaps, Project, List, and Variables, and a bottom pane showing a time-series plot of tweet frequency for the account 'jairbolsonaro' over a 9-day period, with major ticks for March and April.

Coleta de Dados no Twitter com R (por hashtags)

IGTI

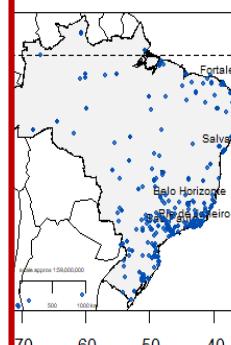
RStudio

```
1  ### Documentação do rtweet https://cran.r-project.org/web/packages/rtweet/rtweet.pdf
2
3  library(rtweet) #Usa a API do twitter
4  library(maps) #plotar mapas
5
6  setwd("C:\\\\Users\\\\[REDACTED]\\\\Desktop\\\\TwitterData") #Define diretório de trabalho
7
8  #Autenticação _____
9  token <- create_token(app = "Primeiro Exemplo IGTI", consumer_key = "[REDACTED]", consumer_secret = "[REDACTED]", access_token = "[REDACTED]", access_secret = "[REDACTED]")
10
11
12
13
14  #Parametros de busca _____
15  search.string <- c("#ficaemcasa OR #coronavirus OR #covid OR #covid-19 OR #covid19")
16  type = "mixed"      #recent", "mixed" ou "popular"
17  # include_rts (TRUE ou FALSE) - usado para indicar se inclui retweets ou não na pesquisa
18  # retryonratelimit (TRUE ou FALSE) - usado para indicar se continua ou não depois do limite de 18000 tweets p
19
20  #Buscando
21  tweets <- search_tweets(search.string, n=18000, lang="pt",type=type
22          ,include_rts = FALSE,retryonratelimit = TRUE)
23
24  #Salvando o vetor de tweets como CSV e apenas o Texto em TXT na codificação do portugues
25  write_as_csv(tweets, "TweetsRawData.csv", fileEncoding = "latin1//TRANSLIT")
26  write.table(tweets$text,"TweetsRawData.txt", fileEncoding = "latin1//TRANSLIT")
27
28  tweets <- lat_lng(tweets) ##cria lat/lgn variaveis usando todos os tweets disponiveis
29
30  ## plota o mapa do Brasil
31  par(mar=c(0,0,0,0)) #Função par define ou ajusta os parâmetros de plotagem. Parametro mar ajusta as margens.
32  map("world", "brazil",lwd = 0.3,fill=T, col="grey95")
33  map(.,add=T) #
34  map.axes()
35  map.scale(ratio=T, cex=0.3)
36  abline(h=0, lty = 2)
37  map.cities(country = "Brazil",minpop = 2000000,pch=15, cex=0.9)# pacote maps
38
39  ## Adicina os tweets ao mapa
40  with(tweets, points(lng, lat, pch = 20, cex = .75, col = rgb(0, .3, .7, .75)))
41
42  with(tweets, points(lng, lat, pch = 20, cex = .75, col = rgb(0, .3, .7, .75)))
```

Environment
17956 obs. of 92 variables

18000
"aficaemcasa OR #coronavirus OR #covid OR
"mixed"

Viewer

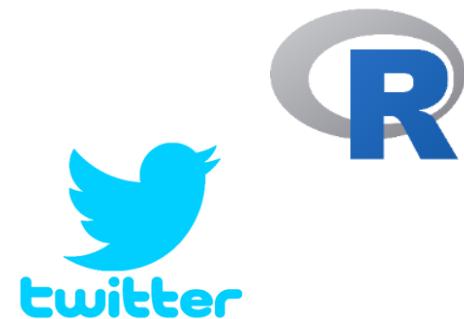


Coleta de dados no twitter com R

- Existem 2 bibliotecas no R para isso:
 - Rtweet: <https://cran.r-project.org/web/packages/rtweet/index.html>
 - twitteR: <https://cran.r-project.org/web/packages/twitteR/index.html>
 - Está sendo descontinuado
- Instalar os pacotes no R Studio usando os comandos abaixo:

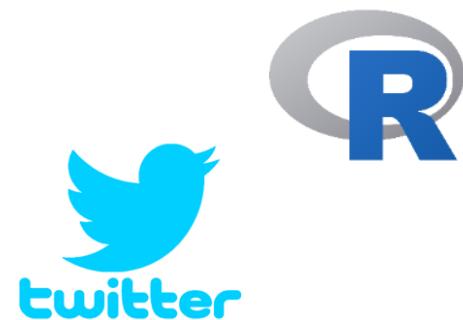
```
install.packages("rtweet")
```

```
install.packages("twitteR")
```



Inserir video da aula 1.5 de AND

- A partir do segundo 00:07:40



Conclusão

- ✓ Exemplo prático de coleta de dados no twitter utilizando a linguagem R e a api do twitter.

- Prática de coleta de dados no twitter utilizando a linguagem Python e a api do twitter.



**Aula 4.4. Coleta de dados no twitter: Exemplo
utilizando a linguagem python**

- ❑ Exemplo prático de coleta de dados no twitter utilizando a linguagem python e a API do twitter.

Credenciais para utilização da API do Twitter

- Consumer Key (API Key)
- Consumer Secret (API Secret)
- Access Token
- Access Token Secret



Coleta de dados no twitter com Python

IGTI

- Existem várias bibliotecas para isso:
 - Tweepy: <http://docs.tweepy.org/en/latest/>
 - Twython: <https://pypi.org/project/twython/>
 - Python-twitter: <https://pypi.org/project/python-twitter/>
- Veja mais opções em <https://pypi.org/search/?q=twitter>



Instalação do pacotes Tweepy

- Instalar os pacotes via Anaconda (prompt) usando os comandos abaixo:

```
conda install -c conda-forge tweepy
```

```
C:\Windows\system32\cmd.exe - conda install scipy --channel conda-forge - conda install scipy --channel bioconda - conda install -c conda-forge tweepy
https://anaconda.org

and use the search bar at the top of the page.

(dev) C:\Users\FERNANDA>conda install -c conda-forge tweepy
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

environment location: C:\Users\FERNANDA\.conda\envs\dev

added / updated specs:
- tweepy

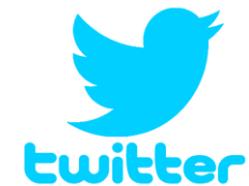
The following packages will be downloaded:

| package          | build
|-----|
| blinker-1.4     | py_1      13 KB  conda-forge
| oauthlib-3.0.1   | py_0      82 KB  conda-forge
| pyjwt-1.7.1      | py_0      17 KB  conda-forge
| requests-oauthlib-1.2.0 | py_0      19 KB  conda-forge
| tweepy-3.8.0      | py_0      26 KB  conda-forge
|-----|
Total:           158 KB

The following NEW packages will be INSTALLED:

blinker            conda-forge/noarch::blinker-1.4-py_1
oauthlib           conda-forge/noarch::oauthlib-3.0.1-py_0
pyjwt              conda-forge/noarch::pyjwt-1.7.1-py_0
requests-oauthlib conda-forge/noarch::requests-oauthlib-1.2.0-py_0
tweepy             conda-forge/noarch::tweepy-3.8.0-py_0

Proceed ([y]/n)? y
```



Coleta de dados no twitter com Tweepy

IGTI

- Tweepy: <http://docs.tweepy.org/en/latest/>

Hello Tweepy

```
import tweepy

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(auth)

public_tweets = api.home_timeline()
for tweet in public_tweets:
    print(tweet.text)
```



- Coleta de dados no twitter com Tweepy



Conclusão

- ✓ Exemplo prático de coleta de dados no twitter utilizando a linguagem python e a API do twitter.

- Prática de coleta de dados no twitter utilizando a plataforma Knime e a api do twitter.

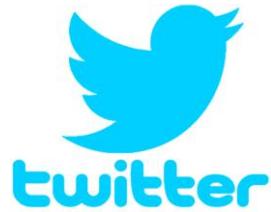


**Aula 4.4. Coleta de dados no twitter: Exemplo
utilizando a plataforma Knime**

- ❑ Exemplo prático de coleta de dados no twitter utilizando a plataforma Knime analytics e a api do twitter.

Coleta de dados no twitter com Knime

IGTI



- Desde de o inicio trocando os slides de capa e Nesta Aula.
- Fim no minuto 5:55.

- ✓ Exemplo prático de coleta de dados no twitter utilizando a plataforma Knime analytics e a api do twitter.

■ Próxima aula

- Entender o que é web crawling.



Aula 4.6. Web Crawling: Exemplo utilizando a linguagem R

- ❑ Exemplo prático de coleta de dados na web utilizando a linguagem R.

- Web crawler em Python

Conclusão

- ✓ Exemplo prático de coleta de dados na web utilizando a linguagem R.

■ Próxima aula

- Prática de coleta de dados na web utilizando a linguagem Python.



Aula 4.7. Web Crawling: Exemplo utilizando a linguagem Python

- ❑ Exemplo prático de coleta de dados na web utilizando a linguagem Python.

- Web crawler em Python

Conclusão

- ✓ Exemplo prático de coleta de dados na web utilizando a linguagem Python.

■ Próxima aula

- Prática de coleta de dados na web utilizando a plataforma Knime.



Aula 4.6. Web Crawling: Exemplo utilizando a plataforma Knime

- ❑ Exemplo prático de coleta de dados na web utilizando a plataforma Knime.

- Web crawler na plataforma Knime

Conclusão

- ✓ Exemplo prático de coleta de dados na web utilizando a a plataforma Knime.