



Um Sistema de Recomendação Semântico Baseado em Conteúdo

Por

Lucas Lara Marotta

Trabalho de Graduação



Universidade Federal da Bahia
wiki.dcc.ufba.br/DCC/

SALVADOR, Setembro/2018



Universidade Federal da Bahia
Departamento de Ciência da Computação

Lucas Lara Marotta

Um Sistema de Recomendação Semântico Baseado em Conteúdo

Trabalho apresentado ao Departamento de Ciência da Computação da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Frederico Araujo Durão*

SALVADOR, Setembro/2018

*Dedico esta dissertação à minha família, amigos e
professores que me deram todo o apoio necessário para
chegar até aqui.*

*It matters not how strait the gate, how charged with punishments the
scroll, I am the master of my fate, I am the captain of my soul*

—WILLIAM ERNEST HENLEY

Resumo

TODO

Palavras-chave: Sistema de Recomendação, Recomendação Baseada em Conteúdo, Recomendação Semântica, Web Semântica

Abstract

TODO

Keywords: Recommender Systems, Content-Based Recommendation, Semantic Recommendation, Semantic Web

Sumário

Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Acrônimos	xix
Lista de Códigos Fonte	xxi
1 Introdução	1
1.1 Motivação	2
1.2 Problema	4
1.3 Objetivos da Solução Proposta	5
1.4 Estrutura	6
2 Sistemas de Recomendação	7
2.1 Histórico	7
2.2 Conceitos	9
2.3 Tarefas de um Sistema de Recomendação	10
2.4 Técnicas de Recomendação	12
2.4.1 Filtragem Colaborativa	13
2.4.2 Filtragem Baseada em Conteúdo	14
2.4.3 Comparação das Técnicas de Recomendação	15
2.5 Aplicações de Sistemas de Recomendação	16
2.5.1 Netflix	16
2.5.2 Skoob	18
2.6 Sumário	19
3 Web Semântica	21
3.1 Arquitetura e formato de dados	22
3.1.1 RDF	23
3.1.2 SPARQL	24
3.1.3 OWL	24
Estrutura de um documento:	26
3.1.4 Estrutura na rede semântica	29
3.2 Dados ligados	29

3.2.1	Linked Open Data	31
3.3	Similaridade Semântica	31
3.3.1	Medidas de Similaridade Semântica	34
	Baseadas em estrutura:	34
	Baseadas em conteúdo:	35
	Baseadas em características ou recursos:	36
3.4	Projetos na Web Semântica	37
3.4.1	DBPedia	37
3.4.2	Google Knowledge Graph	39
3.5	Sumário	40
4	Um Sistema de Recomendação Semântico baseado em conteúdo	41
4.1	Requisitos	42
4.1.1	Requisitos funcionais	42
4.1.2	Requisitos não funcionais	44
4.2	Arquitetura	44
4.3	Tecnologias	46
4.3.1	JAVA	46
4.3.2	Spring Boot	47
4.3.3	HTML, CSS, Javascript	48
4.3.4	MySQL	48
4.3.5	Apache Jena	49
4.3.6	Apache OpenNLP	49
4.3.7	Apache Lucene	50
4.4	Funcionamento	50
4.4.1	Integração entre camadas	52
4.4.2	Modelo de dados	52
4.4.3	Algoritmo de similaridade e recomendação	52
4.5	Sistema em ação	52
4.6	Sumário	52
5	Avaliação	53
	Referências Bibliográficas	54

Lista de Figuras

2.1	Exemplo de lista de vídeos em alta no YouTube (2017)	8
2.2	Recomendação de Filmes no serviço Netflix. Figura elaborada pelo autor (2017).	17
2.3	Página de avaliação do livro no Skoob. Figura elaborada pelo autor (2017).	18
3.1	Exemplo do grafo RDF (RDF, 2017)	23
3.2	Exemplo do grafo da tripla sujeito predicado objeto (Web, 2009)	24
3.3	Camadas na rede semântica. (Júnio César de Lima, 2005)	28
3.4	Camadas na rede semântica. Figura elaborado pelo autor de acordo com a publicação de Berners-Lee (2008)	29
3.5	Sistema de avaliação do LOD (Berners-Lee, 2008)	32
3.6	Diagrama da nuvem dos dados ligados (Andrejs Abele, 2017)	33
3.7	Recorte da tabela de dados de triplas de entidades mapeadas no DBPedia. (DBPedia, 2014)	38
3.8	Ilustração da arquitetura do DBPèdia (DBPedia, 2017)	38
3.9	Ilustração do sumário de dados mapeados no Google Knowledge Graph.	39
4.1	Organização das camadas do padrão MVC (Sommerville, 2010)	45
4.2	Segmentação de tarefas no NLP. (Guts, 2016)	49
4.3	Fluxo das camadas do sistema de recomendação	51

Lista de Tabelas

4.1	Requisitos funcionais do sistema.	43
4.2	Requisitos não funcionais do sistema.	44

Lista de Acrônimos

NFC	Need For Cognition
API	Application Programming Interface
SR	Sistema de Recomendação
CF	Collaborative Filtering
CBF	Content Based Filtering
DVD	Digital Video Disc
RMSE	Root Mean Square Error
WWW	World Wide Web
W3C	World Wide Web Consortium
XML	eXtensible Markup Language
RDF	Resource Description Framework
SPARQL	SPARQL Protocol and RDF Query Language
OWL	Ontology Web Language
URI	Universal Resource Identifier
SQL	Structured Query Language
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
LOD	Linked Open Data
MIS	Most Informative Subsume
MVC	Model View Controller
VM	Virtual Machine
SO	Sistema Operacional

OOP	Object Oriented Programming
IOC	Inversion of Control
ORM	Object Relational Mapping
CSS	Cascading Style Sheets
SGBD	Sistema de Gerenciamento de Banco de Dados
NER	Name Entity Recognition
NLP	Natural Language Processing

Lista de Códigos Fonte

3.1	Exemplo de consulta na linguagem SPARQL	25
3.2	Exemplo do topo de um documento OWL	26
3.3	Exemplo do cabeçalho XML de um documento OWL	27
3.4	Exemplo de propriedades transitivas no OWL	27
3.5	Exemplo do cabeçalho de uma ontologia	28

1

Introdução

The computer is my favourite invention. I feel lucky to be part of the global village. I don't mean to brag, but I'm so fast with technology. People think it all seems too much, but we'll get used to it. I'm sure it all seemed too much when we were learning to walk.

—YOKO ONO

A expansão dos meios de comunicação através da Internet possibilitou o rápido acesso a todo tipo de informação de diversas áreas do mundo a todo lugar. Consumir conteúdo digital tornou-se atividade comum no dia das pessoas. Conforme mais se expande o acesso as mídias digitais mais conteúdo é gerado e mais está disponível para ler, ver, ouvir e interagir. Segundo [Walker \(2014\)](#) chegamos a uma era em que trafegamos uma quantidade enorme de dados que rapidamente perde-se a escala e cognição para o humano. Qual o significado de 400 milhões de tweets¹ por dia? Usar o pensamento empírico de grandes matemáticos como “to measure is to know” (William Thomson) torna-se especialmente difícil com o volume de informações produzidas neste século. Com a quantidade de dados disponíveis não é irônico ouvir “não sei qual filme assistir”, pois apesar do fácil acesso existe uma grande sobrecarga a qual expõe o usuário a um mar de dados ([Wellman, 2013](#)), dificultando o acesso ao conteúdo que seja mais relevante.

O volume de informações apresenta-se como um obstáculo ao usuário que deseja consumir algum tipo conteúdo. Compras online possuem milhares de opções e nem todos estão dispostos a passar um grande tempo olhando o catálogo disponível. Uma das razões pela preferência de compra pela Internet é justamente a “falta de tempo”, conforme revela

¹Tweet é o nome utilizado para designar as publicações feitas na rede social do Twitter (<https://www.merriam-webster.com/dictionary/tweet>)

análise de [Baubonienė and Gulevičiūtė \(2015\)](#). Dessa forma, é natural que o usuário recorra a alternativas para se guiar pelas informações e encontrar mais facilmente aquilo que lhe é mais útil. Para minimizar o obstáculo que o volume de informações se opõem, é comum apelar para ajuda de conhecidos, parentes, amigos, como apontado pela pesquisa de [Baubonienė and Gulevičiūtė \(2015\)](#), onde um dos fatores relacionados ao consumidor que influenciam a opção pela compra pela Internet são as recomendações de outros usuários.

A larga difusão da Internet, principalmente pela Web, também cria um desafio pela busca de informação. Sistemas populares de recuperação de informação, como Google, amenizam o problema ([Isinkaye et al., 2015](#)), mas são deficientes quanto a personalização e priorização da informação em relação as preferências e interesses do usuário. Essa é uma das razões pelo grande aumento do desenvolvimento e procura por sistemas de recomendação. Sistemas de recomendação são sistemas de filtragem de itens que possuem objetivo de prever a avaliação e preferência do usuário ([Ricci et al., 2011](#)). Tais soluções contribuem ainda mais com a experiência do usuário no que diz ao conceito do Need For Cognition (NFC) que reflete na tendência de indivíduos em se engajar e aproveitar numa atividade ([Baubonienė and Gulevičiūtė, 2015](#)). Esses sistemas filtram os dados para reduzir o problema da sobrecarga de informação ([Konstan and Riedl, 2012](#)), podendo ser utilizados em diversos domínios como livros, filmes, músicas até para construir experiências em jogos online ([Crecente, 2017](#)).

Os sistemas de recomendação tipicamente possuem três tipos de abordagens para as sugestões: filtragem colaborativa, filtragem baseada em conteúdo e filtragem híbrida que leva em consideração as duas anteriores. Filtragem baseada em conteúdo são fundamentadas na descrição dos dados e nas preferências dos usuários ([Aggarwal, 2016b](#)). Desse modo, um dos objetivos deste trabalho é modelar um sistema com métricas que realizem a filtragem baseada em conteúdo, além de utilizar dados de serviços da web semântica para expandir as possibilidades de recomendação.

1.1 Motivação

Com a crescente popularização do acesso e uso da Web no mundo, cada vez é mais comum que pessoas escolham este ambiente para fazer compras, o comércio eletrônico. No Brasil, em 2015, movimentou R\$ 41,3 bilhões com o e-commerce² segundo estudos da E-bit³ como aponta o [Seb \(2016\)](#). O estudo também levanta que livros e revistas estão

²Modalidade de comércio que realiza suas transações financeiras por meio de dispositivos e plataformas eletrônicas (<https://ecommercenews.com.br/o-que-e-e-commerce/>)

³<https://www.ebit.com.br/>

em 5º lugar como o tipo de item mais procurado. O crescimento do uso de dispositivos eletrônicos para realizar compras online, mostra que cada vez mais pessoas utilizam a Internet, especialmente para as redes sociais. Somente o Facebook⁴ já registrou em 2017 2 bilhões de usuários ativos (Sta, 2017). O tamanho da plataforma mostra que existe uma quantidade enorme de dados sobre usuários da Internet de todo o mundo, podendo ser fácil de encontrar preferências e relações de amizade. Esses dados servem como uma excelente fonte de busca para montar um perfil.

A grande quantidade de informação sobre os usuários presentes nessas redes sociais, é de amplo valor para construção de sistemas de recomendação. Em muitas dessas plataformas, é disponibilizado para terceiros uma Application Programming Interface (API) para que por exemplo o usuário possa acessar em uma aplicação utilizando as credenciais dessa rede, o que pode facilitar a adesão de novos serviços. Assim, é possível construir um sistema de recomendação baseado em conteúdo já com uma infraestrutura de dados conhecida e amplamente difundida e aceita pelos usuários. A utilização do Sistema de Recomendação (SR) com filtragem baseada em conteúdo, aprende e recomenda itens que sejam similares aos que o usuário já demonstrou interesse (Ricci *et al.*, 2011).

Na similaridade em termos associados aos itens em comparação, é comum em domínios como de livros e filmes seja comparado termos como gênero e autor. Nesse caso, é analisando se já foi demonstrado interesse em filmes com esses termos, para que assim o sistema aprenda e recomende novos filmes com esses mesmos. Entretanto, pode ser interessante para o usuário encontrar filmes que não sejam necessariamente do mesmo gênero ou autor, mas que possuam narrativas mais similares ou relacionadas. Como exemplo considere os filmes *Sucker Punch* (Snyder and Snyder, 2011) e *Labirinto do Fauno* (del Toro *et al.*, 2006), possuem diferentes diretores e apesar terem um tema de fantasia, se diferem bastante pois, o primeiro é um filme mais de ação enquanto que o segundo é um drama que se passa num período de guerra. Na narrativa dos filmes é possível encontrar novos pontos de similaridade, como os dois tratarem de jovens garotas que entram em mundo fantasioso e irão precisar vencer uma série de desafios para superar dificuldades em tempos difíceis. Nesse sentido, analisar a similaridade de conteúdo da descrição de um filme que contenha um trecho da sua narrativa, possa levar ao usuário a sair do seu círculo tradicional de preferência, podendo contribuir com o NFC no uso de um sistema. Uma das propostas desse trabalho é explorar os resultados analisando esse termo.

Além de analisar a similaridade de filmes também observando a descrição da narrativa,

⁴<https://www.facebook.com>

será utilizado o serviço da web semântica DBPedia⁵, para obter mais informações das descrições dos filmes extraíndo relações semânticas de entidades presentes nos textos. Para o SR prover as informações personalizadas é necessário criar um perfil do usuário para indicar o tipo de conteúdo, baseando-se em itens que sejam similares que aos que usuário gostou no passado. Expandindo o alcance do SR será proposto e avaliado utilizando-se o domínio de filmes, um modelo que leve em consideração nas métricas de avaliação, a relação semântica das entidades presentes nas descrições das narrativas. O objetivo é explorar o relacionamento das ontologias presentes na sinopse do filme, pelos dados ligados (apresentados no Capítulo 3.5) oferecidos no serviço da web semântica. Com os dados ligados é possível estabelecer uma relação entre diferentes fontes de dados para formar um único espaço global. É importante ressaltar que os trabalhos apresentados aqui não possuem características especificamente voltadas ao domínio de filmes, mas este é apenas usado como motivador para criação de um SR. A DBPedia possui mapeada milhares de instâncias de dados de pessoas, artistas, filmes, livros entre muitos outros em diversas línguas.

1.2 Problema

O problema deste trabalho trata-se da deficiência e dificuldade quanto a sistemas de recomendação sugerir itens quando apenas o conteúdo sintático é analisado desprezando relações semânticas presentes do conteúdo. É tradicional construir um SR apenas observando as características discretas dos itens, como propriedades de categoria, mas existe uma lacuna de informações que são desprezadas que podem ser extraídas analisando-as numa rede semântica de relações que as envolva.

Um problema também muito comum ao montar o perfil do usuário, é a falta de informação sobre suas preferências. O sistema ainda não obteve interações suficientes para montar um perfil, afetando diretamente a qualidade das recomendações. Com o serviço do Facebook⁶ existe a possibilidade de extrair dados das preferências para um grande número de pessoas de forma automática e transparente, uma vez que já é amplamente aceito pelos usuários. Dessa forma, além de facilitar a montagem do perfil do usuário, de imediato diminui a sobrecarga de informação que ainda passaria para poder usufruir de um SR.

Outra questão trata-se de como esses algoritmos de filtragem e personalização afetam

⁵<http://wiki.dbpedia.org>

⁶<https://www.facebook.com>

as pessoas. O livro “The Filter Bubble” Par (2011) levanta preocupações sobre tais sistemas, onde o usuário fica fortemente sujeito a apenas ao mesmo tipo de conteúdo, ou informação que não venha criar conflitos de ponto de visão, o efeito bolha. Assim, utilizando um SR que apenas analisasse termos de gênero e título poderia deixar o usuário “preso” no círculo tradicional de preferência. Essa preocupação pode também ter um impacto negativo no sistema, já que é possível que os usuários venham a encontrar outros conteúdos que poderiam ter interesse, mas são apenas encorajados a aqueles mais tradicionais.

A busca tradicional de informação em sistemas de recuperação, como o Google⁷, possui dados dispersos e por muitas vezes desorganizado, além da carência de dados personalizados e priorizados que considere os interesses do usuário para encontrar o item desejado. Somando a isso, propondo um sistema em que também seja possível extrair a similaridade da descrição das narrativas dos filmes, analisando e buscando outras relações semânticas com as entidades presentes, pode trazer resultados que amenizem o efeito bolha. Esse trabalho tem um dos objetivos de explorar que decorrências podem ser obtidas levando em consideração essa abordagem.

1.3 Objetivos da Solução Proposta

Este trabalho propõem a criação de um SR baseado em conteúdo que também utilize uma análise da similaridade semântica (ver capítulo 3.5) entre os itens envolvidos. Para isso será proposto um modelo de usuário que leve em consideração a descrição da narrativa do item. O objetivo é explorar que resultados podem ser obtidos realizando consultas ao serviço DBPedia⁸. Para a construção do SR foi escolhido o domínio de filmes, como motivador e exemplo de aplicação que tire proveito desse sistema. Através de uma pequena análise empírica na rede de relacionamento do autor, percebeu-se que as pessoas tendem a informar mais das preferências de filmes do que de livros, outro fator para a escolha do domínio.

Com o acesso a esse serviço da web semântica, serão analisadas entidades procurando ontologias e relações presentes nas sinopses dos filmes, através dos dados ligados na DBPedia. Assim, pode ser comparada à similaridade de dois filmes através da presença ou relação de ontologias presentes na descrição. Como exemplo, caso um filme possua na sinopse o termo *Morfeu* e o outro não, mas possua outras entidades sobre deuses

⁷<https://www.google.com>

⁸<http://wiki.dbpedia.org>

mitológicos, como *Zeus*, poderá ser criado um nível de similaridade e relevância com o novo filme.

Os filmes de preferência do usuário serão obtidos através do Facebook⁹. Inicialmente o usuário se registrará na aplicação desenvolvida por este trabalho com sua conta do Facebook. Em seguida o sistema irá coletar as informações do perfil do usuário referentes a filmes que ele avaliou ou tenha marcado. Após coletar esses dados será necessário complementar as informações dos filmes, já que o Facebook não possui informações satisfatórias e consistentes sobre o domínio. Para isso, será usado a plataforma do DBPedia para obter demais dados.

1.4 Estrutura

Neste capítulo foi motivado e introduzido o problema deste trabalho. Os próximos capítulos estão organizados da seguinte maneira: O Capítulo 2 apresenta os conceitos teóricos usados neste trabalho referentes a SR. O Capítulo 3.5 apresenta conceitos sobre a web semântica. O Capítulo 4.6 apresenta a proposta do SR com a resolução de um modelo de usuário que leve em consideração a descrição de itens, discutindo sua implementação. O Capítulo 5 apresenta a avaliação do sistema, conclusões e considerações finais.

⁹<https://www.facebook.com>

2

Sistemas de Recomendação

A Internet disponibiliza um enorme volume de informação para o usuário, o que cria um desafio pela busca de informação. Por esse problema, empresas cresceram construindo sistemas de recuperação e filtragem, para superação da sobrecarga de informação, como é o caso do Google¹. Neste capítulo será apresentado um panorama sobre SR, introduzindo os principais conceitos, tarefas e processos que o caracterizam.

2.1 Histórico

Em razão da crescente dificuldade de usuários administrar a quantidade de informação, é comum decidir baseado em opiniões e recomendações de outros, especialmente quando há pouca experiência no assunto (Resnick and Varian, 1997). Conforme mais se expandia a tendência do uso de meios digitais de comunicação, mais rapidamente pessoas migraram de cartas para e-mails. A grande quantidade de e-mails acabava deixando o usuário imerso em documentos, dificultando o consumo do conteúdo. Em 1992, Xerox Palo Alto Research Center apresentou o sistema Tapestry (Goldberg *et al.*, 1992) na revista mensal ACM Communications², como proposta para lidar com o problema quantidade de e-mails.

O objetivo do sistema era prover listas de e-mails permitindo a inscrição dos usuários naquelas que fossem mais importantes. Alguns sistemas daquela época suportavam filtragem de e-mails baseado no seu conteúdo, mas os autores acreditavam que uma maneira mais eficiente seria com ajuda da avaliação de outros usuários. Interessante ressaltar que o termo “filtragem colaborativa” apresentado no artigo tornou-se comum, e só alguns anos depois surgiu a defesa do termo sistemas de recomendação, mais genérico,

¹<https://www.google.com>

²<https://cacm.acm.org/>

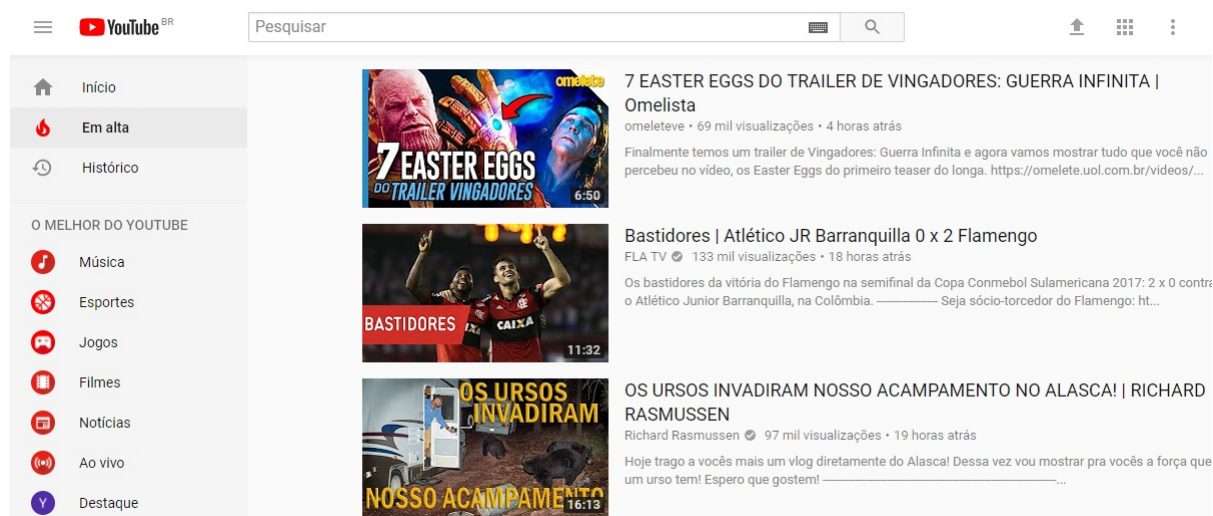


Figura 2.1 Exemplo de lista de vídeos em alta no YouTube (2017)

como defende [Resnick and Varian \(1997\)](#) em seu artigo.

O sistema do Tapestry foi concebido para a filtragem colaborativa, onde colaborações de outras pessoas auxiliam a outros filtrarem, gravando suas avaliações dos itens. Uma das vantagens da aplicação da filtragem colaborativa é que não depende da análise do conteúdo o que é especialmente útil para a análise itens complexos como vídeos, amplamente usado em serviços como o YouTube³. Um exemplo das recomendações no YouTube é na página “em alta” que mostra os vídeos em alta tendência baseada no feedback e visualizações. Em geral, as recomendações personalizadas são dispostas como uma lista de itens ranqueados. O termo “item” é o mais comum a ser denotado por SR para usuários, o que pode designar para diversos tipos, como filmes, livros, músicas etc.

Para construir o ranque os SRs tentam prever qual é o item mais adequado àquele usuário ([Ricci et al., 2011](#)). Para realizar a tarefa o SR coleta dos usuários suas preferências que podem ser informadas de forma explícita, como avaliação de produtos, ou implícita interpretando suas ações como o histórico de navegação. O princípio do SR é da dependência existente entre o usuário e sua atividade em torno dos itens ([Aggarwal, 2016a](#)). Como exemplo, se um usuário comprou um filme de ficção científica, é mais provável que também tenha interesse em outro filme de ficção científica. Dessa forma, o sistema lida com o problema da sobrecarga por filtrar itens que sejam menos prováveis do usuário gostar, baseando-se nas demonstrações do interesse prévio em outros itens, seja por outros usuários ou não.

O aumento da importância da Web como meio eletrônico, especialmente para o

³<https://www.youtube.com>

e-commerce, também se mostrou como força para o desenvolvimento de sistemas de recomendação. Na Web o usuário pode informar o seu feedback de produtos sobre o que gostou ou não. Nesse contexto, a aplicação do SR não somente beneficia o usuário, mas também para aqueles que o provem (Isinkaye *et al.*, 2015). Estudos (Baubonienė and Gulevičiūtė, 2015) demonstram que usuários optam por realizar compras online para poupar tempo. Contudo, com a explosão da variedade de informação disponível, em vez de agir em benefício começa a denigrir a experiência, diminuindo a experiência de uso. É bem aceito que ter escolha é bom, mas ter mais nem sempre é melhor (Ricci *et al.*, 2011).

É importante ressaltar que por fornecer uma informação individualizada, que esteja mais alinhada com o perfil do usuário é o que diferencia os sistemas de recomendação de sistemas de recuperação de informação. Tradicionalmente o motor de buscas deve retornar tudo correspondente a um termo de pesquisa, porém cada vez mais o usuário entra no fator desses sistemas (Burke, 2002). Sistemas como o Google⁴, vão além de retornar termos que batem com a consulta, mas também com a quantidade de outras páginas referentes, histórico de buscas, localização, compatibilidade com dispositivos móveis, além de introduzir informações extra a busca, com os quadros do knowledge graph⁵.

2.2 Conceitos

Sistemas de recomendação são sistemas de processamento de informação que lidam com diversos tipos de dados para construir recomendações que tentam prever a preferência do usuário (Ricci *et al.*, 2011). Os dados tratam-se de basicamente de itens que serão apresentados a usuários na forma de recomendações. Técnicas de recomendação variam com dependência do tipo de conhecimento que pode ser extraído de um dado (Ricci *et al.*, 2011). Dados de avaliações possuem pouca informação, o que resulta em técnicas diferentes em relação daquelas que dependem mais da descrição de um item ou relações com as atividades do usuário. Generalizando, SRs referem-se a três tipos de objetos: itens, usuários e transações que são as relações entre usuários e itens.

- **Itens:** Objetos que são recomendados. Podem ser caracterizados pela complexidade valor ou utilidade. O valor de um item pode ser positivo se é útil para o usuário, ou negativo se não é apropriado ou foi uma decisão errada de seleção por parte do mesmo. O usuário pode ser modelado e representado de diferentes formas, variando

⁴<https://www.google.com>

⁵ <https://www.google.com/intl/bn/insidesearch/features/search/knowledge.html>

bastante em relação do domínio operado pelo SR. Toda vez que um usuário interage com um item constrói-se um custo cognitivo, o que pode entrar na relevância na construção do sistema, mesmo se o usuário não chega a adquirir o item interagido. Alguns exemplos de itens são: livros, notícias (baixa complexidade), computadores, viagens, vagas de trabalho (alta complexidade).

- **Usuários:** Usuários de um SR, podendo ter uma variedade de objetivos e características. São explorados uma série de informações variadas para personalizar as recomendações. A informação pode ser estruturada de diversas formas de acordo com o seu tipo, e a seleção de um modelo depende das técnicas a serem utilizadas. Modelos para sistemas de filtragem colaborativa pode usar apenas listas de avaliações de itens por usuários. O modelo de usuário cria o seu perfil, ou seja, armazena suas preferências e necessidades. Usuários também podem ser descritos baseados num padrão de comportamento, como o histórico de navegação na Web sua ou localização.
- **Transações:** Genericamente refere-se a transações gravadas das interações entre usuários e o SR. Transações podem ser vistas como um histórico de registros, um log de dados que armazena importantes informações geradas das interações com o sistema. Um registro pode conter a descrição do que foi consultado para uma recomendação particular de um item.

2.3 Tarefas de um Sistema de Recomendação

Sistemas de recomendação são vistos como mais do que uma ferramenta de prover sugestões de itens que o usuário possa desejar. (Ricci *et al.*, 2011) em seu artigo introduziu uma série de funções que podem ser aplicadas a SRs.

- **Aumento do número de itens vendidos:** Uma das funções mais importantes para aplicações comerciais. O objetivo é ser capaz de vender outros itens comparados àqueles que são vendidos sem qualquer tipo de recomendação. O objetivo é geralmente alcançado devido a itens que são prováveis de serem úteis a necessidade do usuário.
- **Vender itens mais diversos:** Também outra função de alta importância, na qual permite o usuário a selecionar itens que podem ser difíceis de encontrar. Num

serviço de recomendações de filmes, como o Netflix⁶, o provedor estará interessado que os usuários encontrem conteúdos diversos, não somente os mais populares.

- **Aumentar a satisfação do usuário:** Quando um usuário encontra recomendações que sejam de seu interesse, impacta na experiência com o sistema. Um SR bem desenvolvido permite uma combinação precisa de recomendações que juntos a uma interface com boa operabilidade, pode aumentar a noção subjetiva da avaliação de um sistema.
- **Aumentar a fidelidade:** Um usuário costuma ser leal a um site que, quando visitado, o reconhece como um consumidor reincidente e o trata como um visitante de valor. É muito como para um SR levar em consideração as informações obtidas em prévias interações com o usuário. Consequentemente, por quanto mais tempo o usuário interage com o site, mais refinado seu modelo torna, ficando cada vez mais efetivo e customizado o resultado da recomendação.
- **Melhor entendimento do que o usuário quer:** Outra função importante, na qual pode ser influenciada por outras aplicações, é a descrição das preferências do usuário, seja coletada de forma explícita ou prevista pelo sistema. Um serviço pode decidir reutilizar esses dados do usuário para anunciar um produto em específico, derivado da coleta das informações de transações do SR.

Usuários também podem desejar um SR quando oferece suporte a suas tarefas ou objetivos. [Herlocker *et al.* \(2004\)](#) é uma clássica referência no assunto, e define onze tarefas comuns que SR podem ajudar a implementar.

- **Encontrar bons itens:** Recomendar a usuários alguns itens em ranque, junto a uma predição de o quão o usuário possa gostar deles. Também comum no uso em sistema comerciais.
- **Encontrar todo os bons itens:** Recomendar todos os itens que satisfazem as preferências do usuário. Neste caso é insuficiente apenas encontrar alguns bons itens. Esta função torna-se útil quando existe um número reduzido de itens, ou quando há uma razão crítica para fornecer informação, como em contextos de uso médico ou financeiro.
- **Anotações em contexto:** Dado um contexto, enfatizar alguns itens de uma lista a depender das preferências do usuário.

⁶<https://www.netflix.com>

- **Recomendar uma sequência:** Recomendar uma sequência de itens invés de gerar uma única recomendação.
- **Recomendar um grupo:** Sugerir grupos de itens bem relacionados que possam ser da preferência do usuário.
- **Apenas navegando:** Mesmo que o usuário não possua a intenção de comprar um item, o SR deverá ajudá-lo a navegar pelos catálogo de maneira que encaixe no escopo de interesse do usuário.
- **Encontrar um sistema de recomendação confiável:** Nem todos os usuários podem confiar no sistema, dessa forma é importante oferecer testes de suas funcionalidades.
- **Melhorar o perfil:** Relativo a capacidade de o usuário prover dados ao SR sobre suas preferências. Tarefa fundamental para personalizar o sistema, caso contrário apenas seria possível oferecer recomendações que fosse relativa ao usuário comum.
- **Expressar-se:** Usuários podem não se importar com as recomendações, mas o sistema pode permiti-lo a contribuir com as avaliações e expressão de suas opiniões.
- **Ajudar outros:** Para alguns é importante contribuir com informações de suas opiniões e avaliações, pois compartilhando sua experiência pode ajudar outros formarem uma opinião.
- **Influenciar outros:** Alguns usuários podem ter apenas o objetivo de influenciar outros, ou até usar o SR para denegrir a imagem de alguns itens.

2.4 Técnicas de Recomendação

As recomendações utilizadas no sistema são alcançadas através de algumas técnicas que possuem o objetivo de prever informações sobre itens e preferências de usuários. O SR irá produzir recomendações individualizadas como saída, ou será capaz de guiar o indivíduo de forma personalizada a modo de encontrar itens úteis (Burke, 2002). Apresentadas não somente como técnicas de filtragem colaborativa, Resnick and Varian (1997), introduz o termo mais genérico de sistema de recomendação, uma vez que tais sistemas podem explicitamente não utilizar recipientes que talvez sejam desconhecidos uns aos outros.

Para alcançar as principais funções de um SR, é necessário que o sistema seja capaz de identificar que itens possuem alguma utilidade para o usuário (Ricci *et al.*, 2011).

O sistema deve prever ou comparar a utilidade de itens, para decidir como recomendá-los. Dessa forma, as recomendações podem variar conforme os dados conhecidos de usuários e itens, podendo ter maior ou menor influência em uma função específica. Como exemplo, durante a etapa da predição pode ser considerado uma informação que não seja necessariamente personalizada, como apenas recomendar itens mais populares. De posse de poucas informações, ou não conclusivas, a premissa é basear-se num item que tem boa aceitação, ou seja, que é útil para muitos, com uma recomendação provável ao usuário genérico.

Ampliando ao já apresentado Tapestry (Goldberg *et al.*, 1992), nem todas as técnicas precisam ser baseadas nas informações de preferências de outros usuários. Na literatura já foram discutidos diversas técnicas, como as apresentadas nos trabalhos de (Ricci *et al.*, 2011) e (Burke, 2002). Dentre essas abordagens estão:

- **Filtragem Colaborativa:** O sistema agrega avaliações ou recomendações, reconhecendo características comuns entre usuários baseando-se nos itens de suas avaliações.
- **Baseada em conteúdo:** Objetos de interesse são definidos pela associação de suas características. O sistema aprende e recomenda itens similares ao que usuário demonstrou interesse no passado.
- **Demográfico:** Objetivam categorizar o usuário baseado nas informações pessoais dos usuários. Recomendações são baseadas nas classes demográficas dos usuários.
- **Baseada em conhecimento:** Realizam sugestões de itens baseadas em inferências das preferências do usuário.

Abaixo será apresentado em maiores detalhes o funcionamento das técnicas de filtragem colaborativa e baseada em conteúdo.

2.4.1 Filtragem Colaborativa

Recomendação com Collaborative Filtering (CF) é uma das técnicas mais familiares e já implementadas (Ricci *et al.*, 2011). A similaridade das preferências e desejos de dois usuários é calculada baseada na similaridade do histórico de avaliações dos usuários. A premissa do método é de que a opinião de outros usuários pode ser selecionada e agregada de forma a prover predições razoáveis ao usuário alvo (Ekstrand *et al.*, 2011). Como exemplo, intuitivamente assume-se que usuários que concordam sobre a qualidade de

um filme que João gosta, então João provavelmente gostará de outros filmes que outros usuários avaliaram, mas não assistiu.

O perfil de um usuário na CF pode ser continuamente aprimorado conforme o usuário interage com sistema, podendo levar o tempo de uso como fator de avaliação. Em alguns casos a avaliação pode ser apenas binária (gostei ou não), ou então de valor real que determina um grau de utilidade. Nesse caso, nas avaliações do usuário, o sistema deverá modelar uma função $R(u, i)$ representando o grau de utilidade do item i para o usuário u . Basicamente, a tarefa do sistema é estimar um valor de R baseado nos pares de usuário e item. Dessa forma, avaliando os dados dessas previsões de R para o usuário alvo, o sistema recomendará uma quantidade de itens com as maiores utilidades previstas.

Tipicamente, conforme apresentado por [Burke \(2002\)](#), CF divide-se em dois métodos principais: vizinhança e baseados em modelo. No método da vizinhança o foco é no relacionamento entre itens ou usuários, conhecidos como de *item-item* ou *usuário-usuário* ([Ricci et al., 2011](#)), utilizando informações armazenadas com o tempo. O método aborda modelos através da análise da preferência armazenada das classificações de usuário-item, pela avaliação de outros itens similares. Já o método baseado em modelo é criado diretamente do histórico das avaliações para aprender as preferências do usuário, podendo-se usar uma quantidade diversa de técnicas para o aprendizado, como redes neurais. O objetivo é compreender e extrair das interações usuário-item características de destaque para o sistema, podendo criar classes de preferências dos itens.

2.4.2 Filtragem Baseada em Conteúdo

Ao contrário da filtragem colaborativa, sistemas de recomendação baseados em Content Based Filtering ([CBF](#)), seleciona itens baseados entre as relações de seus conteúdos e as preferências do usuário. A CBF é uma continuação natural das pesquisas nos sistemas de filtragem de informação, [Burke \(2002\)](#). O método utiliza-se da intuição de que se o usuário demonstrou interesse em certos itens com determinados atributos, é provável de também ter interesse em outros itens de mesmo atributo ou semelhante. Como exemplo, se João gostou dos filmes com o ator *Tom Cruise*, é provável que vá gostar de outros filmes com o mesmo ator. Os sistemas de CBF foram desenhados para explorar cenários com itens que podem ser descritos com um conjunto de propriedades ou atributos ([Aggarwal, 2016b](#)).

Nessa abordagem, o sistema deverá aprender do perfil do usuário seus interesses baseados na combinação das características presentes nos objetos que ele avaliou ou marcou. O tipo do perfil utilizado no sistema dependerá do método aplicado. A informação das

preferências do usuário pode manifestar-se de forma explícita, onde existem avaliações ou indicações dos itens favoritos, ou de forma implícita como itens que o usuário comprou. Nos métodos aplicados na CBF, as descrições dos itens avaliados são usadas como dados de treinamento para criar uma classificação específica para o usuário (Aggarwal, 2016b). Os perfis da filtragem baseada no conteúdo são modelos de longo prazo, onde mais dados são atualizados conforme mais evidências do usuário são observadas, Burke (2002).

Apesar da descrição do conteúdo, ou seja, atributos particulares dos itens, sejam o centro da análise da utilidade de novos itens para recomendação, a avaliação de outros usuários tem significativo impacto no sistema (Aggarwal, 2016b). Essa característica apresenta tanto vantagens como desvantagens. Por um lado, num contexto da *partida a frio*, onde há pouca informação disponível sobre as avaliações dos usuários, há margem de utilização enquanto houver outras suficientes informações das preferências do usuário. Mesmo quando um item é novo ou desconhecido, o sistema ainda pode aproveitar suas características para recomendar novos itens, algo que não é possível apenas baseando-se nas avaliações de outros usuários.

Assim, sistemas de CBF são tipicamente utilizados quando há suficiente informação das preferências do usuário disponíveis. Particularmente, são de mais fácil utilização quando usados em domínios com dados não estruturados e ricos em textos, como páginas da Web.

2.4.3 Comparação das Técnicas de Recomendação

Todas as abordagens dos SR possuem vantagens e desvantagens, dependendo de questões como novos itens, usuários, e quantidade de informação disponível sobre os dois. Em relação a novos usuários, como recomendações partem da comparação de informações do usuário alvo e outros usuários, quanto menos avaliações o sistema possuir, mais difícil será a classificação. Já para novos itens, o problema surge em domínios em constante atualização e novas informações e onde cada usuário pouco avalia. Também pode ser visto como o problema do *early rater*, uma vez que a pessoa que avalia primeiro, pouco se beneficia.

Burke (2002) apresentou alguns pontos comuns das diferenças desses sistemas:

- **Sistemas baseados em filtragem colaborativa:** Dependem da sobreposição de avaliações através dos usuários e possuem dificuldades quando há escassez dessas avaliações dos itens. O problema ressalta que as técnicas colaborativas melhor servem quando a densidade de interesses de usuários é alta através de um universo

de itens que não mudam rapidamente.

- **Sistemas baseados em conteúdo:** Possuem o problema da partida a frio, onde o sistema não acumulou dados suficientes para construir uma recomendação confiável. Também são limitados pela quantidade de informações disponíveis e associadas aos itens. Isto acaba colocando a técnica muito dependente da descrição dos dados. Uma grande desvantagem em relação a abordagem colaborativa é que a abrangência de gêneros, onde deixa o usuário sujeito ao mesmo tipo de conteúdo. A depender da CF, pela a avaliação de outros usuários é possível recomendar itens “fora da caixa”.

2.5 Aplicações de Sistemas de Recomendação

O sistema Tapestry (Goldberg *et al.*, 1992) foi um marco inicial no desenvolvimento de aplicações, introduzindo a filtragem colaborativa. Hoje, SR são quase que obrigatórios para muitas lojas online e serviços de entretenimento, tornou-se algo comum e já disseminado entre usuários. A seguir será apresentado algumas aplicações em destaque que usam sistemas de recomendação.

2.5.1 Netflix

Com a evolução da Internet, as mídias físicas para consumo de entretenimento começaram a decair, especialmente para filmes. O avanço na conexão da banda larga trouxe o modelo do *streaming*⁷ que possibilita o usuário a assistir o conteúdo a qualquer momento, lugar, sem ter que necessariamente sair de sua residência para ir à uma locadora, por exemplo. Embora o Netflix⁸, tenha iniciado no ramo de aluguel de Digital Video Disc (DVD)s (Keating, 2012), a companhia rapidamente abandonou este modelo e partiu para a transmissão de filmes e em seguida para produção de seus próprios filmes e séries. Dessa forma, o serviço de filmes e séries cresceu, ocupou espaço das televisões, cinemas e alcançou diversos países.

Com a crescente quantidade de títulos disponíveis na plataforma e também de usuários, logo o serviço desenvolveu seu próprio sistema de recomendações de vídeos, baseado nas avaliações de usuários. Em outubro de 2006 a companhia publicou um concurso pelo melhor sistema de filtragem colaborativa que poderia superar a precisão de seu SR, o

⁷Transmissão contínua de mídia pela Internet, (<https://directradios.com/streaming>)

⁸<https://www.netflix.com>

2.5. APLICAÇÕES DE SISTEMAS DE RECOMENDAÇÃO



Figura 2.2 Recomendação de Filmes no serviço Netflix. Figura elaborada pelo autor (2017).

Cinematch ([Bennett et al., 2007](#)). Neste ponto o serviço já tinha lançado um banco de dados contendo 100 milhões de avaliações de usuários e 18 mil títulos. O Cinematch analisava as avaliações acumuladas dos usuários semanalmente usando uma variante da correlação de Pearson, com todos os outros filmes para determinar uma lista de filmes similares. Sendo assim, conforme o usuário provia avaliações, o sistema computava uma regressão baseada nessa correlação para determinar uma predição única personalizada. Caso não houvesse nenhuma predição personalizada a média de todas as avaliações é usada. As predições eram apresentadas como conjunto de 5 estrelas.

O desempenho do Cinematch é medido principalmente pelo cálculo da raiz do erro quadrático médio, Root Mean Square Error (**RMSE**) ([Herlocker et al., 2004](#)), das predições do sistema contra as avaliações que os usuários informam. Com os sistemas propostos no concurso, a companhia propôs um prêmio para aqueles que conseguissem melhorar a precisão em 10%. Nesse ano de 2017, a companhia migrou seu sistema de avaliação das tradicionais 5 estrelas para uma avaliação binária, o *Like* e *Dislike* ([Var, 2017](#)). Segundo a companhia, os usuários confundiam a avaliação de 5 estrelas, pois na verdade eram sempre as predições avaliadas para o filme, assim agora as predições aparecem no formato de porcentagem de relevância e a avaliação do usuário é indicada pelos símbolos do gostei ou não gostei. Também as predições passaram a serem baseadas apenas no histórico e comportamento do usuário e não mais na média em relação às outras pessoas.

skoob Busque por título, autor, editora, ISBN... [Explorar](#) [Entrar](#)

O Código Da Vinci (Robert Langdon #2)
(The Da Vinci Code)

4.1 ★★★★★
69.418 avaliações

LERAM	LENDO	QUEREM LER	RELENDO	ABANDONOS	RESENHAS
222.178	1.984	50.665	163	4.047	880

♥ Favoritos (8.433) ❌ Desgostados (4.370) 🔄 Trocam (1.547) ★ Avaliaram (69.418)

Um assassinato dentro do Museu do Louvre, em Paris, traz à tona uma sinistra conspiração para revelar um segredo que foi protegido por uma sociedade secreta desde os tempos de Jesus Cristo. A vítima é o respeitado curador do museu, Jacques Saunière, um dos líderes dessa antiga fraternidade, o Priorado de Sião, que já teve como membros Leonardo da Vinci, Victor Hugo e Isaac Newton. Momentos antes de morrer, Saunière consegue deixar uma mensagem cifrada na cena do crime que apenas sua neta, a criptógrafa francesa Sophie Neveu, e Robert Langdon.

[ver mais](#)

O Código Da Vinci
(The Da Vinci Code)
Robert Langdon #2
Dan Brown

🛒 R\$ 25,94 até R\$ 38,16

ISBN-13: 9788575421130
ISBN-10: 8575421131
Ano: 2004 / Páginas: 432
Idioma: português
Editora: Sextante

★★★★★
[Adicionar](#)

📖 📖 📖 📖 📖

Sinopse
Edições (8)
Vídeos (7)
Grupos (32)
Resenhas (880)
Leitores (281.053)
Similares (22)

Edições (8) [ver mais](#)

Similares (22) [ver mais](#)

Resenhas para O Código Da Vinci (880) [ver mais](#)

Malandragem
Marcos Bassini ★★★★★
on 26/1/09

1- Nada de estilo, escreva fácil: não esqueça q o Homer Simpson pode querer ler você; 2 Faça como a Madonna: escolha um tema bem polêmico, tipo sexo, política ou religião; 3 Termine o capítulo sem dizer o q aconteceu, só dê a resposta dois capítulos depois e repita a malandragem no capítulo seguinte...ei, tá fazendo o q, aí? Vai lá virar best-seller!... [leia mais](#)

Vídeos O Código Da Vinci (7) [ver mais](#)

Figura 2.3 Página de avaliação do livro no Skoob. Figura elaborada pelo autor (2017).

2.5.2 Skoob

Em janeiro de 2009, o analista de sistemas Lindeberg Moreira realizou sua ideia de criar uma plataforma em que pessoas socializassem o ato da leitura (Sko, 2009), o Skoob⁹. O sistema criado trata-se de uma rede social para leitores no Brasil (Sko, 2017). Na plataforma, o usuário montará uma estante virtual realizando buscas pelos livros e em seguida indicar o que já fez com o livro, se já leu, se lerá ou está relendo. Após a seleção dos livros os usuários poderão avaliar seus livros, podendo até escrever resenhas completas ou de capítulos dos livros, compartilhando com outras pessoas na plataforma.

A rede social, conta com algumas mecânicas para ajudar usuários a encontrar livros, com um sistema busca de livros, recomendação com filtragem colaborativa baseada nas avaliações de usuários, dos marcados como mais lidos, lendo, quero ler entre outros. A plataforma também conta com um sistema que indica livros similares. Todos esses processos não somente levam a questão da socialização da leitura e escrita entre indivíduos

⁹ <https://www.skoob.com.br>

que compartilham interesses, surgidas a partir da aplicação, mas passam a influenciar a forma como usuários passam a tratar a leitura fora do ambiente da comunidade virtual, é o que aponta [VIANA NETO \(2010\)](#).

2.6 Sumário

Neste capítulo, foi apresentado um panorama geral sobre os sistemas de recomendação. Inicialmente abordando o histórico envolvido e motivações na criação dos conceitos envolvidos do tema. Em sequência foi aprofundado e explicado os conceitos utilizados nesses sistemas. Então, foi apresentado as tarefas e técnicas utilizadas. Também foi aprofundado algumas diferenças e dificuldades entre as principais técnicas de recomendação. Por fim, foi mostrado exemplos de aplicações que utilizam esses sistemas de recomendação. No capítulo [3.5](#) será discutido sobre os conceitos envolvidos na Web Semântica, bem como os princípio dos dados ligados e o serviço da DBPedia¹⁰.

¹⁰<http://wiki.dbpedia.org>

3

Web Semântica

I found myself answering the same questions asked frequently of me by different people. It would be so much easier if everyone could just read my database

—TIM BERNERS-LEE

A introdução e expansão da *World Wide Web* possibilitou acessar e publicar uma grande variedade de conteúdo, seja para o consumo de entretenimento, exposição de opiniões, compras online etc. O crescimento da rede tornou-se tão grande que é latente a necessidade para seus usuários encontrar informações. Para eles, foram criados e desenvolvidos os indexadores de páginas, como o Google¹, Yahoo², Bing³. Tais sistemas facilitam encontrar informações em serviços populares na Internet. Entretanto, e se quiséssemos encontrar algum médico de confiança para marcar uma consulta, levando em consideração uma agenda de compromissos? Ou então se estamos realizando um trabalho escolar e queremos encontrar os reis do século XV? Essas pesquisas certamente são mais complicadas, e resultados de buscas tradicionais levam a informações fragmentadas, com uma série de outras buscas separadas para alinhar todo o conhecimento e semântica envolvidos nessas tarefas. É nesse ponto que entra o conceito da Web Semântica, como uma extensão da já existente.

O conteúdo da Web tradicional é fundamentalmente desenvolvido para humanos lerem, não para máquinas manipularem de forma produtiva e significativa (Berners-Lee *et al.*, 2001). Originalmente desenvolvida para compartilhar e apresentar conteúdo de forma que fosse possível interagir e navegar entre hipertextos e hipermídia, a World

¹<https://www.google.com>

²<https://www.yahoo.com>

³<https://www.bing.com>

Wide Web ([WWW](#)) torna fácil a apresentação de layouts. É possível estruturar um documento com um cabeçalho, um link para outra página, entretanto, dificilmente as máquinas poderão processar semanticamente que informações estão disponíveis e que podem ser organizadas naquela página ou site. Como exemplo, uma página de João com link para seu currículo informando que possui especialização em cardiologia. Todas essas informações podem até serem compreendidas por humanos ao associar a semântica das entidades presentes numa página e analisando links relacionados, mas para a máquina não há uma estrutura comum e eficiente que leve a essas mesmas conclusões.

O objetivo da Web Semântica é de estender a WWW, aproveitando a enorme variedade de dados já existente, mas agregando uma nova camada de metadados que possibilitem o processamento pela máquina e agentes de forma a compreender a semântica das informações apresentadas. Assim, a Web Semântica trata-se de prover formatos para integração de dados de diferentes fontes ([W3C, 2001](#)), onde a Web tradicional mantém-se como o meio de publicação e interconexão de documentos, e na contraparte semântica, armazena-se dados que se relacionam com objetos e coisas do mundo real. Um agente pode se deparar com uma página de clínica na Web e não apenas compreenderá que possui palavras como “tratamento, terapia, remédios, médicos”, como tipicamente é encontrado na Web tradicional, mas também saber o Dr João trabalha nessa clínica nas segundas e quartas com horários no formato *dd/mm/YYYY*.

3.1 Arquitetura e formato de dados

O funcionamento da Web Semântica depende da capacidade de máquinas acessar coleções estruturadas de informações, dados e regras de inferência para executar raciocínio automatizado ([Berners-Lee et al., 2001](#)). O desafio é de como representar conhecimento. Inicialmente o desenvolvimento desses sistemas utilizaram uma abordagem centralizadora, requerendo que as partes envolvidas compartilhem exatamente as mesmas definições de conceitos comuns ou hierárquicos. Entretanto, com a quantidade de conteúdo existente hoje em diferentes línguas, controle centralizado é desafiador. Contrastando essa visão inicial, na Web Semântica cria-se linguagens para regras as quais são tão expressivas quanto o necessário para que a Web seja ampla como desejado ([Berners-Lee et al., 2001](#)). Com um sistema que não seja centralizado é possível que não se responda todas as perguntas, ou seja, encontrado todas as informações, mas permite que regras sejam usadas para criar inferências e escolher o curso de ações para poder ou tentar responder tais perguntas.

Com esses fundamentos os pesquisadores da Web Semântica, em especial o *World*

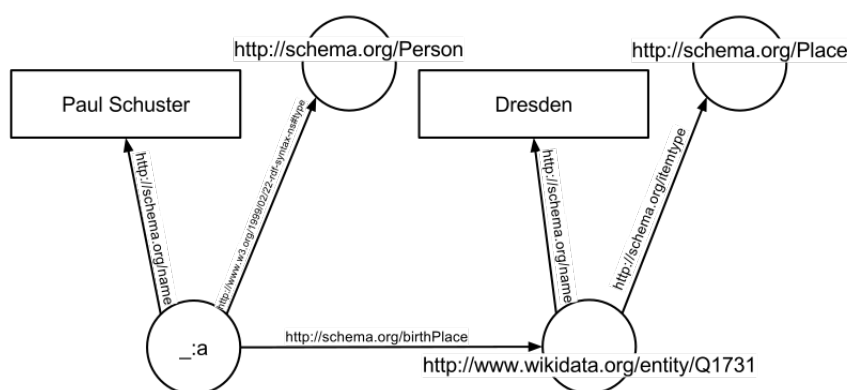


Figura 3.1 Exemplo do grafo RDF ([RDF](#), 2017)

Wide Web Consortium, desenvolveram uma série de padrões e formatos de dados para o uso na Web. O intuito é possibilitar máquinas compreenderem documentos com dados semânticos e não discursos e textos criados pelo homem. Uma tecnologia muito importante para o desenvolvimento da representação do conhecimento e protocolo de comunicação entre máquinas, foi a eXtensible Markup Language ([XML](#)). Com a XML é possível que qualquer um seja capaz de criar suas próprias *tags* e estruturas de um documento com definição de cada termo presente de forma arbitrária. Desse ponto de vista o XML é fundamental como um padrão de comunicação entre máquinas. Anos seguintes, a W3C introduziu outras três importantes tecnologias presentes no cenário atual da Web Semântica: Resource Description Framework ([RDF](#)), SPARQL Protocol and RDF Query Language ([SPARQL](#)), Ontology Web Language ([OWL](#)).

3.1.1 RDF

Resource Descripton Framework é um modelo de dado para a Web que facilita a junção de dados mesmo que seu *schema* difira, além de permitir a sua evolução sem requerer que seus consumidores tenham que se adaptar ([W3C](#), 2014). No RDF a estrutura da *web* de links é estendida para usar os Universal Resource Indentifier ([URI](#))s para nomear a relação entre qualquer coisa, com ambas as pontas, formando o que é conhecido como a tripla.

O uso da URI é especialmente notável para a Web, uma vez que não é possível apenas se basear em valores literais, mesmo para representar um atributo de algo, já que é desejado ter a definição e estrutura podendo considerar um domínio em específico. Como exemplo, com uma URI é possível identificar de forma única o predicado “título” que se refere ao título da função em uma empresa, e não um título de filme. Então, a tripla

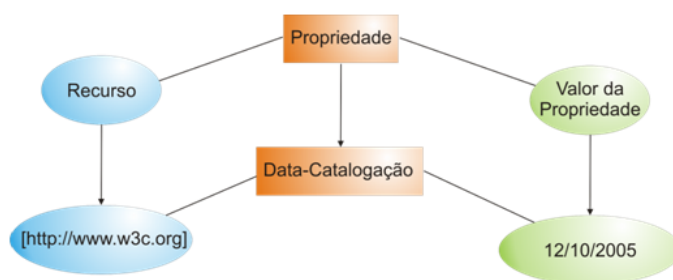


Figura 3.2 Exemplo do grafo da tripla sujeito predicado objeto (Web, 2009)

forma um grupo de três entidades que expressam uma declaração sobre o dado semântico, na forma de “sujeito, predicado, objeto”. Com essa estrutura de links é formado um grafo direcionado, com *labels*, aonde suas arestas representam o link nomeado entre dois recursos representados pelos seus nós (W3C, 2014).

3.1.2 SPARQL

O SPARQL é uma linguagem de consulta para o grafo do RDF (W3C, 2013). Dessa forma, pode-se criar *queries* através de diversos conjuntos de dados de triplas, podendo ser aplicado uma série de filtros para limitar e ordenar os resultados retornados. Diferentemente das linguagens de consulta de banco de dados relacionais, o objeto da coluna não é homogêneo, ou seja, o tipo dado da célula da tabela de resultados é implicado ou definido pelo predicado informado através da URI. O sujeito do RDF pode ser classificado com um análogo a uma entidade nos bancos de Structured Query Language (SQL), diferindo onde os campos (ou atributos) são representados como predicados e/ou objetos separados.

O exemplo do Código Fonte 3.1 demonstra a consulta de dados de uma ontologia *foaf*⁴, conhecida como "friend of a friend".

3.1.3 OWL

Ontology Web Language é uma linguagem para definir e instanciar ontologias na Web (W3C, 2009). Um programa que deseja comparar ou combinar informações entre dois bancos de dados com URIs distintas, deve saber se termos podem ser usados para descrever o significado da mesma coisa (Berners-Lee *et al.*, 2001). O objetivo é que um programa descubra o significado comum seja para o que for encontrado entre os conjuntos de dados. A solução proposta na Web Semântica para esse problema é a utilização de uma

⁴<http://xmlns.com/foaf/spec/>

```
1 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
2 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3 SELECT ?name
4         ?email
5 WHERE
6 {
7     ?person a foaf:Person .
8     ?person foaf:name ?name .
9     ?person foaf:mbox ?email .
10 }
11 }
```

Código Fonte 3.1 Exemplo de consulta na linguagem SPARQL

coleção de informações denominadas de ontologias. Na filosofia uma ontologia tem por objeto o estudo das propriedades do ser, tratando da natureza da existência. Entretanto, no campo da inteligência artificial e na Web, é definido como os termos básicos e relações que compreendem um vocabulário de um domínio, bem como regras para combiná-los junto com relações para definir extensões desse vocabulário ([Patil et al., 1992](#)).

Em essência a ontologia é um documento que define formalmente as relações entre termos. As ontologias podem ser vistas de forma semelhante à hierarquia de classes na programação orientada a objetos. Tipicamente uma ontologia para a Web possui uma taxonomia e um conjunto de regras de inferência. A taxonomia define classes (ou conceitos) de objetos e suas relações, sendo assim, um endereço pode ser definido como um tipo de localidade e o código de uma cidade pode ser definido para ser aplicado apenas a localizações, entre outros exemplos.

A linguagem OWL provê três sublinguagens, OWL Lite, OWL DL, OWL Full como apresentado pela [W3C \(2009\)](#).

- **OWL Lite:** Para primariamente a criação hierárquica e simples de limitações de *features*. Como exemplo, é possível oferecer suporte a limitações de cardinalidade que só permitam valores de 0 ou 1. É mais simples de prover suporte.
- **OWL DL (descrição lógica):** Oferece suporte a uma expressividade máxima sem perder a completude computacional (todas as implicações são garantidas para serem computadas), decidibilidade (todos os cálculos finalizaram em um tempo finito). Inclui todas as construções com restrições e separação de tipos (uma classe também não pode ser indivíduo ou propriedade, uma propriedade também não pode ser um indivíduo ou uma classe).

```
1 <rdf:RDF
2   xmlns = "http://www.w3.org/TR/2004/REC-owl-guide
   ↪ -20040210/wine#"
3   xmlns:vin = "http://www.w3.org/TR/2004/REC-owl-guide
   ↪ -20040210/wine#"
4   xml:base = "http://www.w3.org/TR/2004/REC-owl-guide
   ↪ -20040210/wine#"
5   xmlns:food="http://www.w3.org/TR/2004/REC-owl-guide
   ↪ -20040210/food#"
6   xmlns:owl = "http://www.w3.org/2002/07/owl#"
7   xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#
   ↪ "
8   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
9   xmlns:xsd = "http://www.w3.org/2001/XMLSchema#">
```

Código Fonte 3.2 Exemplo do topo de um documento OWL

- **OWL Full:** Oferece o máximo de expressividade e é sintaticamente livre do RDF sem garantias computacionais. Nessa linguagem uma classe pode ser tratada simultaneamente como uma coleção de indivíduos ou indivíduo como todo. Então, a OWL Full permite uma ontologia ter seu significado ampliado ao pré-definido (RDF ou OWL) vocabulário.

Todas as sub-linguagens são extensões de sua predecessora, sendo assim cada ontologia válida em OWL Lite é uma ontologia válida em OWL DL que por sua vez é uma ontologia válida em OWL Full ([W3C, 2009](#)). É notável destacar que o inverso das relações não é verdadeiro. Completando, todo documento OWL é um documento em XML construído com o RDF.

Estrutura de um documento:

Com a OWL é possível descrever de forma natural classes e relacionamentos entre documentos e aplicações na Web ([Júnio César de Lima, 2005](#)). Os termos descritos devem estar dispostos de tal maneira que não cause ambiguidade, assim é necessário que seja informado quais vocabulários serão empregados. Para o uso de vocabulários a [W3C \(2009\)](#) informa que deve-se definir no topo do documento os *xml namespaces* ⁵, conforme mostrado no código fonte [3.2](#).

⁵No XML, os namespaces são nomes únicos para elementos e atributos no documento. Para resolver as ambiguidades e facilitar as referências antes dos nomes são utilizados prefixos


```
1 <!DOCTYPE rdf:RDF [  
2   <!ENTITY vin "http://www.w3.org/TR/2004/REC-owl-guide  
   ↪ -20040210/wine#" >  
3   <!ENTITY food "http://www.w3.org/TR/2004/REC-owl-guide  
   ↪ -20040210/food#" > ]>
```

Código Fonte 3.3 Exemplo do cabeçalho XML de um documento OWL

```
1 <owl:ObjectProperty rdf:ID="subordinate">  
2   <rdf:type rdf:resource="#owl:TransitiveProperty"/>  
3   <rdfs:domain rdf:resource="#Agent"/>  
4   <rdfs:range rdf:resource="#Agent"/>  
5 </owl:ObjectProperty>  
6  
7 <Agent rdf:ID="Joao">  
8   <subordinate rdf:resource="#Pedro"/>  
9 </Agent>  
10  
11 <Agent rdf:ID="Pedro">  
12   <subordinate rdf:resource="#Maria"/>  
13 </Agent>
```

Código Fonte 3.4 Exemplo de propriedades transitivas no OWL

Acrescentando, a World Wide Web Consortium ([W3C](#)) recomenda incluir no documento um cabeçalho XML que preceda as definições das ontologias como apresentado no código fonte [3.3](#)

Por último será informado o cabeçalho da ontologia junto a suas propriedades. Nesse cabeçalho é importante fornecer informações sobre ela própria. Para descrevê-las utiliza-se as propriedades do OWL, uma vez que a ontologia é um recurso, assim demonstrado no código fonte [3.5](#)

Dentro da definição da ontologia poderão ser informados as classes e indivíduos relacionados como as propriedades e suas relações. As propriedades podem ser descritas como transitivas, simétricas, funcionais ou inversamente funcional. Como exemplo, numa propriedade transitiva de subordinado, se é dito que João é subordinado de Pedro e Pedro é subordinado de Maria, portanto João é subordinado de Maria. No código fonte [3.4](#) é demonstrado a declaração desse tipo de propriedade.

```
1 <owl:Ontology rdf:about="">
2   <rdfs:comment>An example OWL ontology</rdfs:comment>
3   <owl:priorVersion rdf:resource="http://www.w3.org/TR
   ↪ /2003/PR-owl-guide-20031215/wine"/>
4   <owl:imports rdf:resource="http://www.w3.org/TR/2004/REC-
   ↪ owl-guide-20040210/food"/>
5   <rdfs:label>Wine Ontology</rdfs:label>
6   ...
```

Código Fonte 3.5 Exemplo do cabeçalho de uma ontologia

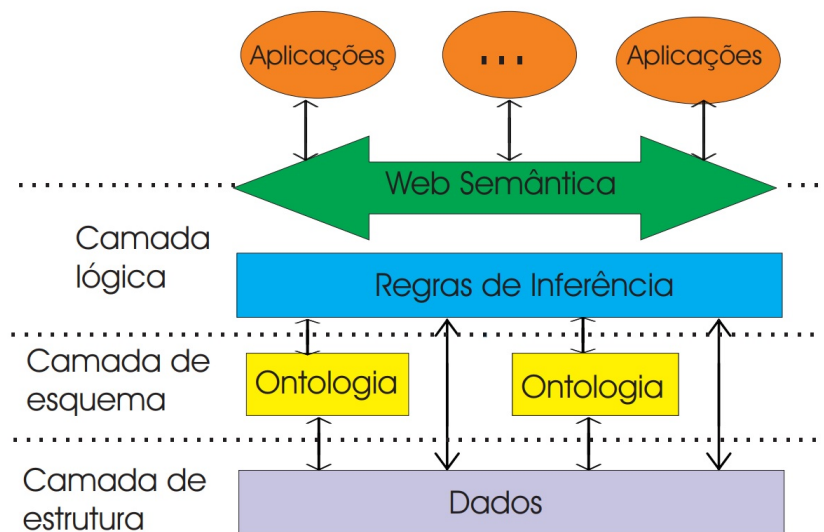


Figura 3.3 Camadas na rede semântica. (Júnio César de Lima, 2005)

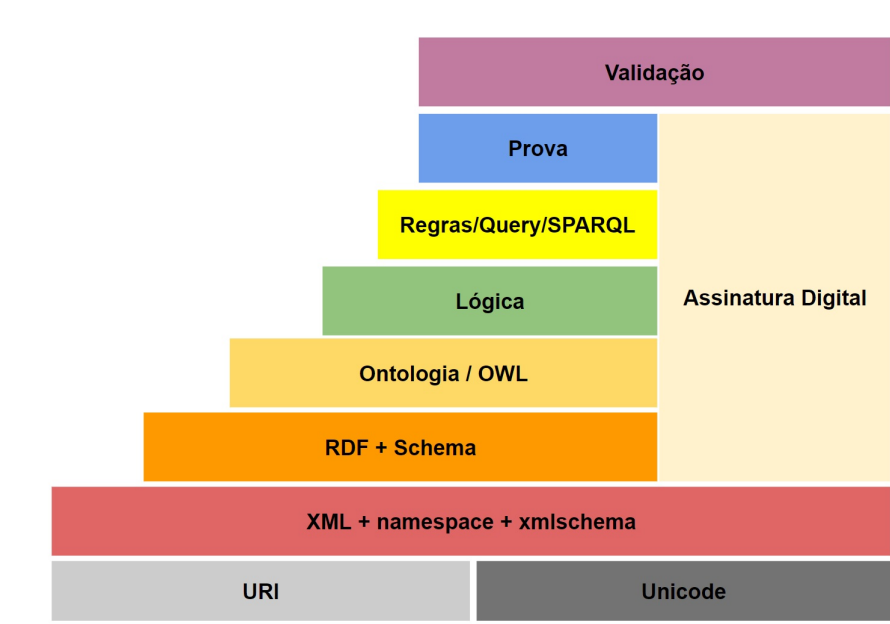


Figura 3.4 Camadas na rede semântica. Figura elaborado pelo autor de acordo com a publicação de [Berners-Lee \(2008\)](#)

3.1.4 Estrutura na rede semântica

A introdução das tecnologias para alcançar os princípios idealizados na Web Semântica são implantados em camadas. De acordo com [Berners-Lee *et al.* \(2001\)](#) é possível dividir esses serviços em três grandes camadas, como demonstrado na figura 3.3. Na camada de estrutura os dados são organizados e definidos seus significados, na qual utiliza-se as triplas do RDF. A camada com os esquemas estão as ontologias, utilizando-se o OWL para a representação de conceitos, inferências através das taxonomias e conjunto de regras. Por último na camada lógica são definidos para fazer inferência sobre os dados. Dessa forma, o desenvolvimento dessas tecnologias (ainda em andamento) e padronização dos formatos foi formulado pela W3C uma pilha das camadas ([Berners-Lee, 2008](#)) da Web Semântica confiável, conforme mostrado na figura 3.4.

3.2 Dados ligados

A evolução da WWW tornou cada vez mais acessível a publicação e acesso a documentos pela navegação no espaço global, através links dos hipertextos ([Bizer *et al.*, 2009](#)). Com os navegadores da Web pode-se passear pelos links nesse espaço e em especial com o uso dos buscadores, que indexam páginas para facilitar a recuperação. Tais mecanismos

já estão amplamente difundidos na publicação de documentos, mas quando comparados aos dados⁶ em si, esses princípios ainda foram timidamente aplicados. Assim, com o crescimento da Web Semântica trouxe-se a ênfase em criar uma Web para os dados, capaz descrever entidades individuais presentes nos documentos, conectando-se por links categorizados para relacionar tais entidades. O objetivo não é somente colocar dados na Web, mas utilizar links que ambas máquinas (principalmente) e humanos possam navegar.

Suportando essa evolução da Web (Berners-Lee, 2006) introduziu um conjunto de melhores práticas para a publicação e conexão de dados estruturados na Web, denominado de *Linked Data* (dados ligados). A adoção dessas práticas permite a extensão da Web com um espaço de dados global conectado de diversos domínios, desde pessoas, livros, publicações até dados governamentais dos mais variados assuntos. Com essa Web de dados surge a oportunidade para novos tipos de aplicações (Bizer *et al.*, 2009), como navegadores customizados para um determinado domínio podendo saltar entre diferentes fontes de dados.

Resumidamente, a W3C (2006) define que para a Web dados ser uma realidade é necessário que os dados estejam disponíveis em padrões de formatos que sejam buscáveis e manipuláveis pelas ferramentas e tecnologias da Web Semântica. Complementando, é preciso também ter acesso ao relacionamento de dados. O conjunto de *datasets* inter-relacionados na Web, para criar links tipificados entre dados de diferentes fontes é o que se denomina de dados ligados.

Ao contrário dos documentos HyperText Markup Language (HTML) na Web dos hipertextos, os dados ligados se baseiam-se nos documentos contendo dados em RDF. Assim são construídos links que são tipificados para realizar declarações sobre coisas arbitrárias no mundo. Berners-Lee (2006) enumerou um conjunto das regras para a publicação e conexão dos dados, conhecidos como os princípios dos dados ligados:

1. Usar URIs para nomear coisas
2. Usar HyperText Transfer Protocol (HTTP) URIs para que pessoas possam procurar seus nomes.
3. Quando alguém procura uma URI, forneça informação útil, utilizando os padrões como RDF e SPARQL.
4. Inclua links para outras URIs, para que assim eles possam descobrir mais coisas.

⁶Note que embora os termos "dados" e "documentos" possam ser análogos, no contexto da Web, documentos tratam-se das páginas dos sites e dados, de fato a informação em si. Assim, na Web os documentos apenas objetivam o aspecto da apresentação perdendo a semântica dos dados presentes.

Um exemplo notável do uso dos dados ligados, é o projeto da DBPedia⁷ que essencialmente torna o conteúdo da Wikipedia⁸ disponível em RDF.

3.2.1 Linked Open Data

Posteriormente em 2010, para incentivar o uso dos dados ligados no meio governamental, Berners-Lee (2006) desenvolveu um "sistema de avaliação" dos dados ligados. O objetivo era expandir o termo introduzindo os dados abertos, onde fossem publicados sob uma licença que não impede o livre reuso. No sistema de avaliação consta um esquema de pontuação em estrelas de 1 a 5, onde cada estrela também acumula as definições das estrelas anteriores, conforme consta na figura 3.5.

O Linked Open Data (LOD) tornou-se o projeto de maior adoção dos princípios dos dados ligados (Bizer *et al.*, 2009), sendo um esforço colaborativo iniciado em 2007 para suportar as definições e tecnologias da Web Semântica introduzidas pela W3C. O motivo para o início da colaboração era de mapear os dados da Web identificando os conjuntos que já estavam disponíveis sob licença aberta. O projeto inclui dados de várias fontes, como a Wikipedia⁹, Geonames¹⁰, Wordnet¹¹ entre diversos outros de múltiplos domínios, alcançando um impressionante diagrama como mostrado na figura 3.6.

3.3 Similaridade Semântica

A similaridade semântica entre dois termos, recursos, itens ou documentos é uma métrica para medir a distância de seus significados ou semântica, dado suas ontologias (Slimani, 2013). O objetivo é estabelecer características em comum entre dois conceitos. A distância entre dois conceitos para humanos pode não ter uma definição formal, já que se pode criar um juízo de valor diferente no relacionamento entre eles. Como exemplo, para uma pessoa a maçã e a banana podem estar mais relacionadas do que a maçã e a pera para outra. A similaridade e relação semântica podem por vezes determinadas como a mesma coisa, ambas como métricas de distâncias entre termos, contudo a similaridade semântica é mais específica (Slimani, 2013). A relação semântica é calculada usando um modelo de espaço vetorial e uma métrica de similaridade, como a similaridade do

⁷ <http://wiki.dbpedia.org>

⁸ <https://www.wikipedia.org>

⁹ <https://www.wikipedia.org>

¹⁰ <http://www.geonames.org>

¹¹ <https://wordnet.princeton.edu>

★	Disponível na Web (qualquer formato), mas com licença aberta para ser <i>Open Data</i> .
★★	Disponível como dado estruturado legível por máquinas (exemplo, <u>excel</u> invés de imagem capturada de uma tabela)
★★★	Como o (2), além do formato não proprietário (exemplo, CSV invés de <u>excel</u>).
★★★★	Todos os anteriores, além do uso de padrões abertos da W3C (RDF e SPARQL) para identificar coisas, para que assim pessoas possam apontar para seus dados
★★★★★	Todos os anteriores, além de ligar seus dados à outros dados de pessoas para prover um contexto

Figura 3.5 Sistema de avaliação do LOD (Berners-Lee, 2008)

cosseno 3.2 que dado dois vetores A e B como uma representação de dois documentos e A_i e B_i seus componentes, seja calculado o produto vetorial euclidiano. (Singhal, 2001).

$$A \cdot B = \|A\|_2 \|B\|_2 \cos(\theta) \quad (3.1)$$

$$similaridade = \cos(\theta) = \frac{A \cdot B}{\|A\|_2 \cdot \|B\|_2} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (3.2)$$

Entretanto, para a similaridade semântica é levado em consideração relações léxicas de sinonímia e hiperonímia onde o significado é abrangido pelo outro termo mais geral (como carro e veículo) (Gracia and Mena, 2008). Na prática, a similaridade semântica pode ser medida pelo menor caminho entre dois termos utilizando suas ontologias associadas. Para calcular a similaridade podem ser usadas diversos tipos de ontologias. Slimani (2013) descreve dois principais tipos de ontologias usadas para medir similaridade.

- **Propósito genérico:** *Wordnet*¹² é um banco de dados que modela o conhecimento léxico da língua inglesa. Nomes, verbos, adjetivos e advérbios são agrupados em conjuntos sinônimos, onde cada um expressa um conceito distinto. Essa ontologia

¹²<https://wordnet.princeton.edu>

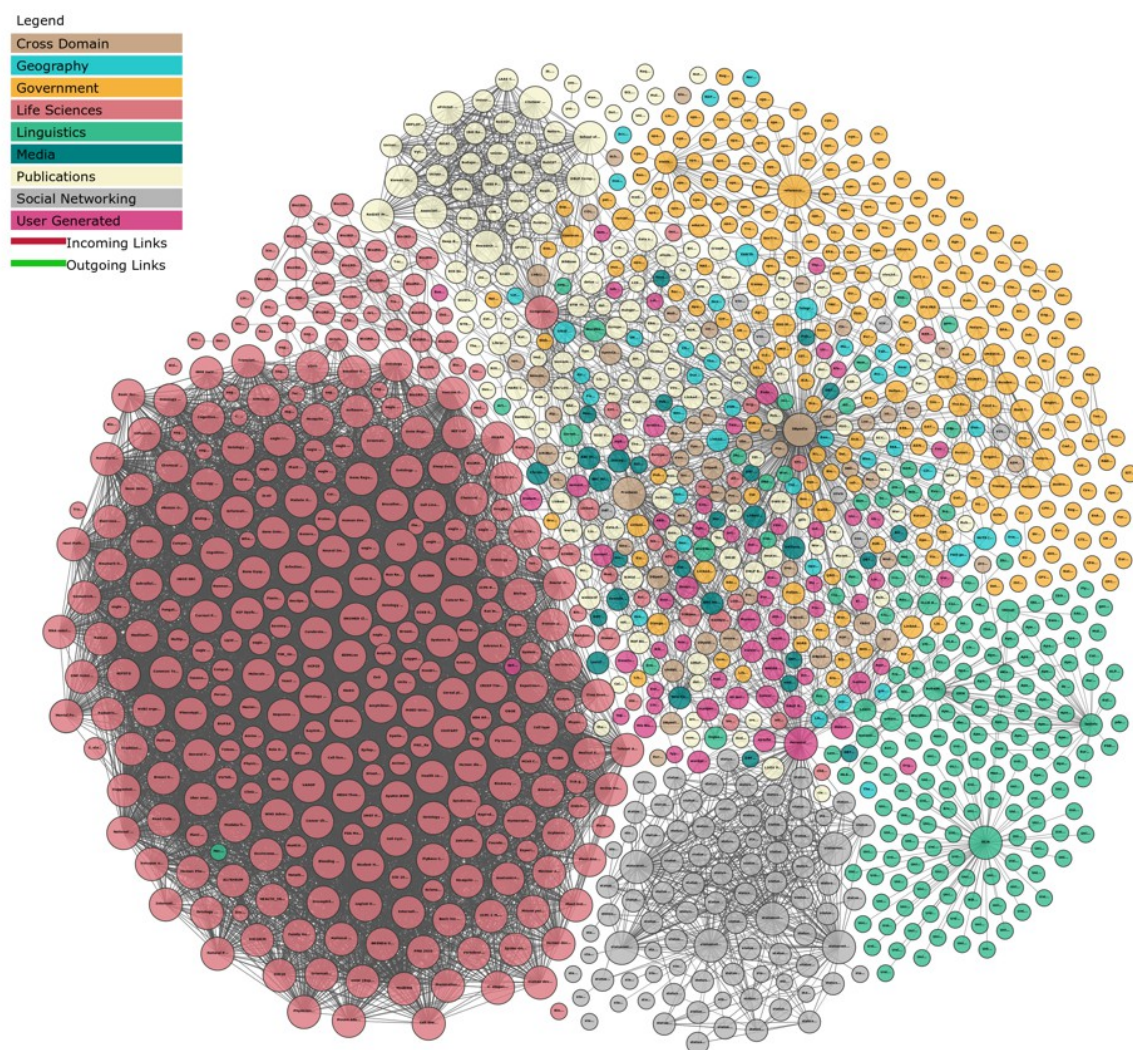


Figura 3.6 Diagrama da nuvem dos dados ligados ([Andrejs Abele, 2017](#))

pode ser utilizada para criar um *score* de similaridade. Pode ser considerada um ontologia para termos de linguagem natural.

- **De domínio específico:** *ULMS*¹³ é um sistema de linguagem médica com uma rede semântica de ontologias de multiuso, multilíngue para biomedicina, conceitos e assuntos relacionados à saúde. O banco de dados do sistema possui uma coleção de vocabulários de conceitos e termos e seus relacionamentos que são denominados de *Metathesaurus*. Cada *Metathesaurus* é classificado como pelo menos uma categoria semântica.

3.3.1 Medidas de Similaridade Semântica

Na literatura já foram apresentadas algumas medidas de similaridade semântica, mas comumente existem três fatores principais (Slimani, 2013) que podem ser associados na topologia (i.e. nós do grafo direcionado) das ontologias: *path length*, *depth*, *density*. Todos esses fatores afetam a medida da distância semântica, assim como as características entre dois termos, que podem aumentar ou diminuir as medidas de acordo com suas semelhanças. Quanto a densidade entre dois termos trata-se do número de filhos dos quais pertencem ao menor caminho (*path*) da raiz ao mais específico conceito entre esses termos. Os fatores que influenciam nas medidas levam a definição de uma classificação que podem ser divididas em quatro principais (Slimani, 2013): baseadas em estrutura, conteúdo, recursos ou características e as híbridas que combinam as características estruturais (*path length*, *depth*, *density*) e alguma outra abordagem.

Baseadas em estrutura:

As medidas baseadas em estrutura (*Structured-based* ou *Path-based*, utilizam funções que computam a similaridade baseada na hierarquia e estrutura da ontologia, ou seja, onde um conceito é definido como “é parte de”, “é um” etc. A função calcula o tamanho do caminho que liga os termos e seus posicionamentos no grafo direcionado da ontologia. Quanto mais dois conceitos são similares, mais *links* existem entre eles. Dentre as medidas baseadas em estrutura se destacam:

- **Shortest Path** (Rada et al., 1989): A medida do menor caminho é um tipo de medida de distância que é primariamente voltada para lidar com hierarquias em

¹³<https://www.nlm.nih.gov/research/umls>

redes semânticas. A função da similaridade entre conceitos C_1 e C_2 é definida como:

$$Sim(C_1, C_2) = 2 * Max(C_1, C_2) - SP \quad (3.3)$$

A função Max é o maior tamanho do caminho entre C_1 e C_2 , quanto a SP é menor caminho relacionando os dois conceitos.

- **Weighted Links**: Similar a medida do menor caminho, contudo é introduzido um conceito de pesos para os links entre os conceitos a serem comparados.
- **Wu and Palmer (1994)**: Para essa medida sejam dois conceitos C_1 e C_2 , é levado em consideração a noção intuitiva de que quanto maior a profundidade, mais similares os conceitos são. Na função tem-se que N_1 e N_2 são a quantidade de links da forma "é um" de C_1 e C_2 , onde o conceito mais específico é o mais próximo ancestral C entre eles.

$$Sim_{W\&P}(C_1, C_2) = \frac{2H}{N_1 + N_2 + 2H} \quad (3.4)$$

Baseadas em conteúdo:

As medidas baseadas no conteúdo, são aquelas que utilizam a informação do conteúdo para medir similaridade. O conteúdo de um conceito é definido pela frequência de termos dado uma coleção de documentos. Grande parte das medidas deste tipo utilizam a informação compartilhada de dois conceitos pais C_1 e C_2 , dos qual $S(C_1; C_2)$ é o conjunto de conceitos que os engloba, conforme a equação 3.5. O menor $p(C)$ é utilizado quando há mais de um pai em comum que C é o Most Informative Subsume (**MIS**), ou seja, o conceito mais informacional que os engloba.

$$P_{mis}(C_1, C_2) = \min_{C \in S(C_1; C_2)} \{p(C)\} \quad (3.5)$$

Algumas das medidas deste tipo são:

- **Resnik (1999)**: O principio desta medida define que dois conceitos são mais similares se eles possuem mais informações compartilhadas. A informação compartilhada entre C_1 e C_2 é o conteúdo de conceitos que os engloba no grafo. A definição de *Resnik* define a medida como a seguinte equação:

$$Sim_{Resnik}(C_1, C_2) = -\ln(p_{mis}(C_1, C_2)) \quad (3.6)$$

- **Lin (1993)**: A proposta é incorporar o vetor semântico e a ordem das palavras para calcular a similaridade. A medida combina o menor caminho SP entre dois conceitos e a profundidade N da taxonomia em relação ao conceito C mais em comum. A definição da equação segue conforme abaixo:

$$Sim_{Li}(C_1, C_2) = e^{-\alpha * SP} * \frac{e^{\beta * N} - e^{-\beta * N}}{e^{\beta * N} + e^{-\beta * N}} \quad (3.7)$$

Baseadas em características ou recursos:

Baseia-se em características ou recursos (*Featured-based*), que partem do princípio de valorizar informações importantes em relação ao conhecimento sobre um termo. A medida assume que os conceitos são descritos por termos indicando suas propriedades ou *features*. A similaridade entre dois conceitos é definida por uma função (3.8) que relaciona suas propriedades ou relacionamentos a outros termos similares na hierarquia da ontologia. **Tversky (1977)** apresenta uma medida *Feature-based* de termos para calcular a similaridade entre diferentes conceitos, contudo o posicionamento desses termos na taxonomia e a informação do conteúdo não são levadas em consideração. A proposta é de que com termos descritos por um conjunto de palavras como propriedades do conceito, então as que são em comum tendem a aumentar a similaridade, enquanto as que não são em comum tendem a diminuí-la. Dessa forma, é definida uma equação onde C_1 e C_2 representam o conjunto de descrições dos termos e $\alpha \in [0, 1]$ é a relação de relevância das características que não são em comum. O valor de α aumenta o quão mais em comum dois conceitos são, e decresce com suas diferenças, e não é necessariamente uma relação de simetria, mas mais baseada na similaridade (**Slimani, 2013**).

$$Sim_{Tversky}(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cap C_2| + \alpha |C_1 - C_2| + (\alpha - 1) |C_2 - C_1|} \quad (3.8)$$

3.4 Projetos na Web Semântica

3.4.1 DBPedia

A DBPedia (DB para *database*) é um esforço colaborativo para a extração de dados do Wikipedia para publicação de dados essencialmente em RDF (Auer *et al.*, 2007). Um dos objetivos é possibilitar que outros explorem a criar uma experiência da enciclopédia mais abrangente, utilizando serviços e aplicações na Web Semântica. O projeto é um dos mais famosos que aplica os conceitos de dados ligados, onde sua importância não somente é dada pela publicação dos dados da Wikipedia, mas também da incorporação de links de outros *datasets*. De fato, o DBpedia, por muitas vezes é considerado um núcleo dentro da iniciativa do LOD.

O projeto tem o foco em converter o conteúdo presente do Wikipedia em conhecimento estruturado utilizando as tecnologias da Web Semântica, para que outros agentes possam explorar realizando consultas e ligando a outros conjuntos de dados (Auer *et al.*, 2007). Assim, o projeto cobre uma das limitações da Wikipedia que é a dependência de apenas ter a busca em texto livre para encontrar informação. Desse papel, o projeto promove três importantes contribuições:

- Desenvolvimento de um *framework* para extração de informação, o qual converte o conteúdo da Wikipedia em RDF.
- Prover o conteúdo da Wikipedia como um largo, multi-domínio *dataset* de RDF. São mais de 100 milhões de triplas já mapeadas. A figura 3.7 mostra um recorte das entidades mapeadas do DBPedia.
- Interligar o DBpedia com outros conjuntos de dados abertos, o que expande a contagem das triplas RDF para mais bilhão.
- Desenvolvimento de uma série de interfaces e módulos de acesso para que tal *dataset* possa ser acessado por serviços da Web ligado a outros sites.

Como a iniciativa compõe o movimento do LOD, os dados do DBpedia podem ser importados por aplicações *third party* utilizando a licença aberta. O projeto faz uso da plataforma do *Virtuoso Universal Server*¹⁴ para prover os dados de RDFs através de uma interface e um *endpoint* em SPARQL. Na figura 3.8 uma panorama da arquitetura do projeto.

¹⁴<https://virtuoso.openlinksw.com>

Amostra de dados do DBPedia						
	Línguas					
Instâncias	en	es	pt	fr	de	ru
Pessoas	1.445.104	99.147	60.056	134.749	179.421	86.269
Atores	6.501	13.831	7.546	14.019	0	0
Artistas	96.282	34.898	14.603	32.562	0	30.266
Políticos	40.343	7.460	4.110	11.461	0	0
Lugares	735.062	156.377	123.114	148.586	168.082	91.099
Instuições de ensino	49.172	1.709	514	2.943	2.600	1.418
Filmes	87.282	12.140	11.643	15.669	18.707	14.912
Livros	31.029	2.217	1.343	3.549	0	18.491
Software	31.401	6.284	4.245	8.980	5.286	0

Figura 3.7 Recorte da tabela de dados de triplas de entidades mapeadas no DBPedia. (DBPedia, 2014)

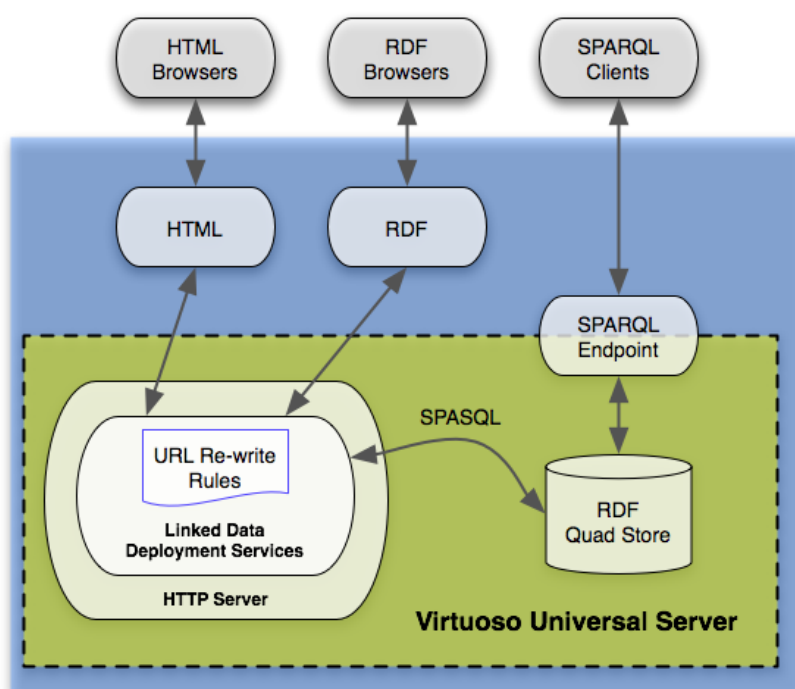


Figura 3.8 Ilustração da arquitetura do DBPedia (DBPedia, 2017)



Figura 3.9 Ilustração do sumário de dados mapeados no Google Knowledge Graph.

3.4.2 Google Knowledge Graph

O *Knowledge Graph* é uma base de conhecimento desenvolvida pelo Google¹⁵ para melhorar e ampliar seu mecanismo de busca (Singhal, 2012). Alinhado aos objetivos da Web Semântica, o projeto pretende expandir o buscador com um grafo de conhecimento, onde foi mapeado entidades de dados legíveis para máquinas com o intuito recuperar a informação semântica nos termos buscados. De forma simples, o intuito é ter a informação de coisas e não de *strings*. Como exemplo, ao ser buscado a palavra "leão", não será apenas retornado uma lista de sites que possuem referências a palavra, mas também prover a semântica e taxonomias envolvidas com a ontologia e relacionados.

Com *Knowledge Graph* é possível pesquisar por pessoas, lugares, esportes, filmes e diversas informações que o Google mapeou no grafo do conhecimento. O serviço já conta com mais de 500 milhões de objetos e 3.5 bilhões de fatos sobre o relacionamento entre diferentes objetos. O objetivo do Google é ampliar o mecanismo de busca em

¹⁵<https://www.google.com>

três sentidos: Possibilitar encontrar o item certo, obter um melhor sumário, ser mais amplo e profundo. Um dos primeiros passos da companhia para atingir os objetivos do projeto é a construção do painel do sumário. Quando pesquisamos por *Leonardo Da Vinci*, procurando seja pelas suas pinturas ou por pintores da renascença, o sistema montará um quadro de dados, conforme a figura 3.9 com as informações, além trazer itens com relações próximas, como seus quadros e outros artistas relacionados. Com esse tratamento Singhal (2012) alega que será possível melhor compreender o que os usuários buscam, além de dar importantes passos para migrar de um motor informação para um de conhecimento, algo importante para o uso em seus assistentes virtuais.

3.5 Sumário

Neste capítulo, foi apresentado os conceitos que fundamentam o desenvolvimento de sistemas para representação de conhecimentos, assim como a abordagem e objetivos da Web Semântica sobre o tema. Em sequência foram introduzidas as principais tecnologias e princípios que são utilizados e o aprofundamento do significado das ontologias. Também foi abordado um panorama sobre a estrutura das tecnologias utilizadas na Web Semântica. Ainda foi introduzido o princípio e prática dos dados ligados e sua extensão com os dados abertos. Ainda foi exposto um panorama da similaridade semântica e tipos de medidas. Por fim, foram apresentados projetos proeminentes no cenário da Web Semântica. No capítulo 4.6 será discutido os conceitos da proposta do sistema de recomendação implementado por este trabalho.

4

Um Sistema de Recomendação Semântico baseado em conteúdo

Desde de tempos o homem busca construir ferramentas e máquinas que facilitem, ampliem, sustem sua capacidade de trabalho e produção. Com o advento dos computadores e dos programas de máquina, o *software* tornou-se essencial para a contínua demanda de problemas e desafios da crescente população global. Como avaliado por [Sommerville \(2010\)](#), o software não se restringe a propriedades materiais das leis da física ou por processos de manufatura. Por um lado, este fato simplifica a engenharia de software devido falta de restrições físicas, mas o torna complexo e de alto custo na realização de mudanças. Dessa forma, com a crescente quantidade de computadores e a diversidade de dispositivos, é cada vez mais relevante a qualidade de software.

O tema não é novo e já é levantado desde a década de sessenta, como na conferência NATO ([Naur and Randell, 1968](#)) sobre problemas e desafios no desenvolvimento de software. A qualidade de software não somente aborda problemas do ponto de vista da coordenação do desenvolvimento, viabilizando a execução pelas máquinas, mas também estuda a importância da legibilidade a fim facilitar a manutenção e compreensão por humanos. Assim, é de suma importância documentar funcionalidades, decisões técnicas a serem utilizadas no processo do desenvolvimento de software, para que outros possam entender o trabalho que está sendo construído ([Pressman, 2010](#)).

Neste capítulo será apresentado os requisitos funcionais e não funcionais para um sistema de de recomendação semântico baseado em conteúdo. Será discutida as tecnologias, comportamentos, modelos e arquiteturas utilizadas. Por fim, será demonstrado o funcionamento de um protótipo de um site para recomendações de filmes, implementando esse sistema.

4.1 Requisitos

Os requisitos de um sistema são descrições do que deve fazer, suas funcionalidades e serviços que restringem sua operação (Sommerville, 2010). Tais requisitos são uma reflexão das necessidades dos consumidores do sistema e definem um propósito específico, como cadastrar um usuário, encontrar produtos etc. Os requisitos de software, então, tratam-se de descobrir, analisar e documentar tais serviços e restrições para a operação do produto final. A descrição desses requisitos deve ser clara e objetiva, para apenas descrever o objetivo final da funcionalidade a ser desenvolvida.

Os requisitos de software são tradicionalmente classificados entre **funcionais** e **não funcionais**, para diferentes níveis de detalhamento para diferentes leitores.

- **Requisitos funcionais:** Descrevem o funcionamento do sistema, e para isso devem prover como o sistema deve reagir à entrada/saída assim como seus comportamentos em diferentes situações.
- **Requisitos não funcionais:** Devem estabelecer as restrições das funcionalidades e serviços oferecidos pelo sistema. São descritas características gerais do sistema, como a usabilidade que não se referem a termos específicos como os requisitos funcionais. Comumente também são descritas questões que devem ser atendidas para a segurança e confiabilidade do sistema.

Para cada requisito é utilizado um código para identificar a funcionalidade, assim facilitando referenciá-la durante o desenvolvimento. Nesta seção serão apresentados os requisitos funcionais e não funcionais para o desenvolvimento deste projeto. Para a descrição das funcionalidades optou-se por usar códigos com a sintaxe [RF0X] para requisitos funcionais e [RNF0X] para requisitos não funcionais. Junto ao código e a descrição do requisito foi adicionada a sua prioridade. As prioridades são classificadas em três categorias: a) **Essencial** para os que precisam ser implementados indispensavelmente, ou seja, são estritamente necessários para o funcionamento do sistema; b) **Importante** para os que são importantes para o funcionamento, mas não são cruciais; c) **Desejável** para os que não interferem diretamente nas funcionalidades básicas do sistema, embora relevantes, mas que podem ser deixados para ser implementados posteriormente.

4.1.1 Requisitos funcionais

A Tabela 4.1 apresenta os requisitos funcionais do sistema.

Tabela 4.1 Requisitos funcionais do sistema.

Código	Nome	Descrição	Prioridade
RF01	Cadastrar usuário	Realizar o cadastro de usuários para criação do seu perfil	Essencial
RF02	Cadastrar usuário com o Facebook	Permitir o cadastro de usuários utilizando sua conta do Facebook	Importante
RF03	Fazer login/logout	Usuários devem ser identificados permitindo a entrada e saída da aplicação	Essencial
RF04	Fazer login pelo Facebook	Permitir o login do usuário pelo Facebook	Importante
RF05	Cadastrar filmes do usuário	Usuários identificados podem cadastrar novos filmes no seu perfil	Essencial
RF06	Visualizar filmes do usuário	Usuários identificados podem visualizar os filmes cadastrados no seu perfil	Essencial
RF07	Remover filmes do usuário	Usuários identificados podem remover filmes cadastrados no seu perfil	Essencial
RF08	Pesquisar filmes cadastrados	Usuários identificados podem pesquisar filmes cadastrados na base do sistema	Importante
RF09	Cadastrar filmes	O sistema deve permitir a alimentação de filmes para a base dados	Essencial
RF010	Edição de filmes	O sistema deve permitir a edição de filmes alimentados para a base	Essencial
RF011	Visualização de filmes recomendados	O sistema deve ser capaz de criar uma lista de filmes recomendados, baseado nos filmes registrados do perfil do usuário	Essencial
RF012	Coletar filmes do Facebook	Usuários identificados devem poder importar filmes marcados no Facebook	Importante
RF013	Visualizar informações do filme	Usuários identificados devem ser capaz de visualização a informação de um filme específico, seja na lista de recomendações ou registrados no seu perfil	Importante

4.1.2 Requisitos não funcionais

A tabela 4.2 apresenta os requisitos não funcionais informando sua característica. Todos os requisitos apresentados são considerados essenciais.

Tabela 4.2 Requisitos não funcionais do sistema.

Código	Descrição	Característica
RNF01	O sistema deve ser fácil uso e dispensar prévio treinamento	Usabilidade
RNF02	O sistema deve ser simples de ser utilizado provendo informações e feedback de forma clara e objetiva	Usabilidade
RNF03	O sistema deve estar organizado de tal forma a não deixar o usuário confuso e ao mesmo tempo incentivar a sua exploração com interesse visual	Usabilidade
RNF04	O sistema deve ser capaz de atualizar as similaridades calculadas dos filmes em <i>background</i>	Funcionalidade
RNF05	O sistema deve prover uma metalinguagem para momentos em que o usuário deve aguardar o processamento de dados, tais como ícones, informativos etc.	Usabilidade
RNF06	O sistema deve possuir um desempenho adequado para que o cálculo da lista de filmes recomendados tenham o menor impacto de carregamento na navegação do usuário.	Funcionalidade
RNF07	O sistema deve ser capaz de incluir novos filmes em <i>background</i>	Funcionalidade

4.2 Arquitetura

A arquitetura de software trata-se das estruturas e componentes, assim como as interações entre essas partes que irão compor o software do sistema. Para [Perry and Wolf \(1992\)](#) a arquitetura de software manifesta-se principalmente em partes do software do produto em relação a: 1) Requisitos para a determinação da informação, processamento e características que serão necessárias para o usuário e o sistema; 2) Arquitetura quando preocupa-se

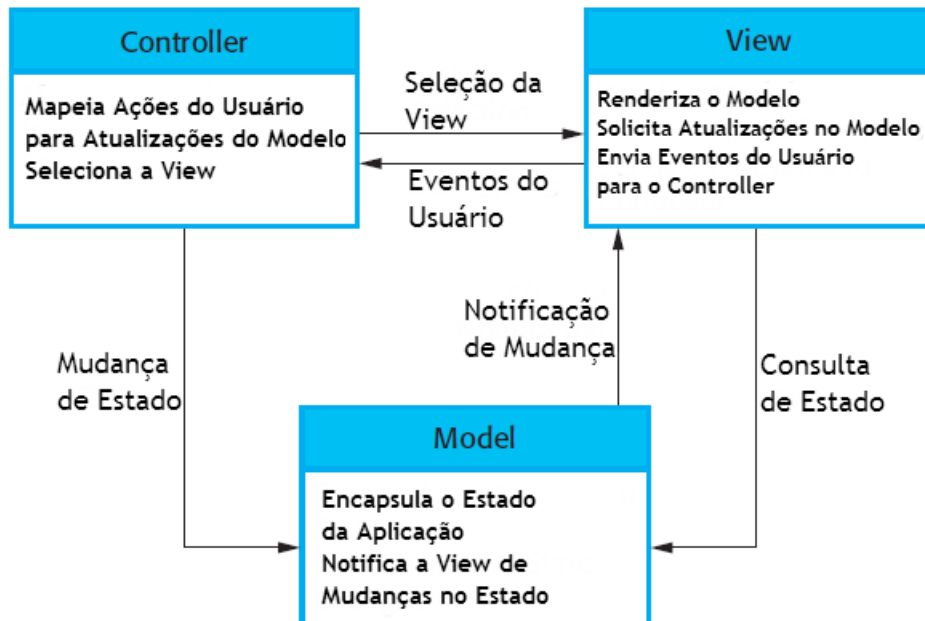


Figura 4.1 Organização das camadas do padrão MVC (Sommerville, 2010)

com a seleção de elementos, suas interações, e restrições necessárias para prover um *framework* que satisfaça os requisitos; 3) Design quando está interessado na modularização e detalhamento do design dos elementos, algoritmos, procedimentos e tipos de dados que suportem a arquitetura e os requisitos; 4) Implementação quando preocupa-se com a representação de algoritmos, tipos de dados que satisfaçam a arquitetura, design e os requisitos.

Para a organização e estrutura deste projeto foi escolhida o padrão de arquitetura Model View Controller (MVC). O objetivo desse padrão é organizar o sistema em camadas em que cada uma seja responsável por funcionalidades específicas no fluxo entre o sistema e o usuário. Assim, o desenvolvimento e alterações podem ser realizadas de forma independente. No MVC o sistema é estruturado em três camadas que interagem entre si:

- **Model:** Camada da representação ou modelo para a manipulação dos dados da aplicação, sendo usado tanto na manipulação de elementos da interface como na persistência de dados. Comumente para a persistência de dados modela-se cada entidade da base dados como entidades que manipularam os registros armazenados. Nessa camada também são estabelecidos elementos que implementam recursos de *proxy* para os dados de fato, como os *repositories*, assim facilitando o reuso da aplicação, devido a abstração dos dados já que a camada do *Controller* apenas se

comunicaria com estes.

- **View:** Camada da apresentação para o usuário, a interface. Envolve toda a parte de visualização e interação com o sistema do ponto de vista do usuário. Nessa camada serão apresentados os dados das *models*.
- **Controller:** Camada que controla o fluxo das informações e/ou comportamentos entre a *view* e a *model*. Pode receber dados da *view* para ser passado à camada das *models*, além de obter dados da *model* para exibição na *view*. Essa camada também pode acessar a de serviços terceiros que manipulem dados vindos da *view* ou da *model*. A camada ainda é responsável pela validação dos dados vindos *view* passados pelos usuários, além do controle do acesso entre outros comportamentos dos requisitos.

Na figura 4.1 é demonstrado o fluxo da comunicação entre as três camadas do MVC. Iniciando pela requisição e interação do usuário através da interface na camada da *view*. Nos *controllers* são implementados os pontos de acesso, endereços para a camada da *view*, assim como todas as regras e controles da entrada de dados, controle de acesso e utilização das *models*. O *controller* também poderá transformar os dados para suportar o formato de dados da *model*. Por último na *model* encontra-se as representações das entidades da base de dados, assim como dados possíveis dados para facilitar a manipulação dos dados da *view*.

4.3 Tecnologias

Para o desenvolvimento do sistema foram escolhidas algumas tecnologias para arquitetura software, como linguagens de programação, *framework* MVC, processamento e banco dados, entre outras. A seguir serão apresentadas as tecnologias utilizadas.

4.3.1 JAVA

JAVA¹ é uma linguagem de programação de propósito genérico, desenvolvida originalmente por James Gosling na Sun Microsystems² em 1995. Atualmente a linguagem foi comprada pela Oracle Corporation³. As características em destaque da linguagem

¹<https://www.java.com>

² <https://www.oracle.com/br/sun/index.html>

³<https://www.oracle.com>

estão no fato de ser baseada em classes e orientada a objetos. A Object Oriented Programming (OOP) é um paradigma de programação que abstrai conceitos em objetos, que podem conter dados, campos e comportamentos nomeados de *methods* (Lewis and Loftus, 2000).

Outra característica importante da linguagem trata-se da filosofia apresentada pelos desenvolvedores de “escreva uma vez, rode em qualquer lugar”. A filosofia trata-se da linguagem ser compilada por uma Virtual Machine (VM) possibilitando escrever um mesmo pedaço de código que possa ser portado para outra plataforma sem necessidade de alterá-lo, uma vez que cada VM implementa as especificidades da nova plataforma abstraindo o acesso ao Sistema Operacional (SO).

A linguagem JAVA é usada em diversos sistemas e plataformas, com inúmeros propósitos, desde aplicações *desktop*, pesquisa científica, desenvolvimento Web entre outros propósitos.

4.3.2 Spring Boot

Spring Boot⁴ é um projeto da Pivotal Software⁵ para facilitar o processo de configuração e publicação de aplicações e serviços providos pelo Spring⁶, com baixo esforço e configuração. O *Spring* é um framework *open source*⁷ que provê um compreensivo conjunto de modelos de configuração para aplicações JAVA. O elemento principal do *Spring* é prover infraestrutura para aplicações oferecendo os seguintes principais recursos:

- **Inversão de Controle:** Inversion of Control (IOC), também conhecido como *dependency injection* é um princípio que as “dependências” devem ser supridas, injetadas por outro objeto. As dependências são objetos que serão usados como “serviços” para acessar suas funcionalidades, dentro dos *containers* de IOC. A injeção é a passagem da dependência para um objeto (o cliente) (LLC, 2006). O termo “inversão de controle” origina-se do fato que a criação de valores de classes externas ao objeto não pode ser realizada pelo próprio objeto mas, sim pelos *containers* de IOC.
- **Acesso a dados:** O framework possui diversas bibliotecas para o acesso a dados, tanto para bancos relacionais como não relacionais. Também é oferecido um

⁴<https://projects.spring.io/spring-boot/>

⁵<https://pivotal.io>

⁶<https://spring.io>

⁷Modelo de desenvolvimento que promove um licenciamento livre para o design ou esquematização de um produto

sistema ORM que trata-se de uma técnica para traduzir o formato de dados de um banco relacional para [OOP](#), facilitando sua manipulação.

- **Arquitetura MVC:** Fornece todo suporte para customizar e criar uma arquitetura [MVC](#).

4.3.3 HTML, CSS, Javascript

O [HTML](#)⁸, [Cascading Style Sheets \(CSS\)](#)⁹ e [JavaScript](#) forma a principal pilha de tecnologias utilizadas na Web. O HTML é uma linguagem de marcação mantida pela [W3C](#) para criação de páginas, originalmente desenvolvida por Tim-Berners-Lee ([Raggett et al., 1998](#)). O objetivo é a fácil construção e publicação de conteúdo no ambiente Web e consequentemente na [WWW](#). No *Spring Boot* as páginas HTML podem ser escritas utilizando algum dos mecanismo de templates, como o *thymeleaf*. Uma das vantagens porá utilização desses mecanismos é a herança de visualizações, assim como facilidade de interligar em manipular os dados passados do *controller*.

O [CSS](#) é uma linguagem para criar regras de estilização das páginas [HTML](#). O CSS cria ou altera um formato de apresentação (tamanho, cores, margens etc) de algum elemento do HTML, como blocos, parágrafos, imagens entre outros. Quanto ao [JavaScript](#) é uma linguagem de programação originalmente criada por Brendan Eich na Netscape Communications¹⁰. A linguagem é utilizada para controlar o comportamento de páginas HTML, oferecendo dinamicidade, podendo alterar elementos da página em tempo real.

4.3.4 MySQL

O [MySQL](#)¹¹ trata-se de um Sistema de Gerenciamento de Banco de Dados ([SGBD](#)) que utiliza a linguagem [SQL](#) para manipulação de dados guardados em um sistema de arquivos ([Ventavoli, 2014](#)). Originalmente desenvolvido por Michael Widenius em 1994, o seu foco é para o desenvolvimento de aplicações Web, embora tenha se popularizado para a maioria das plataformas existentes ([DuBois, 2013](#)). Foi o banco de dados escolhido para a persistência de dados da aplicação e ser de fácil integração com o *Spring Boot*.

⁸<https://www.w3.org/html>

⁹<https://www.w3.org/Style/CSS/>

¹⁰ <http://isp.netscape.com>

¹¹<https://www.mysql.com>

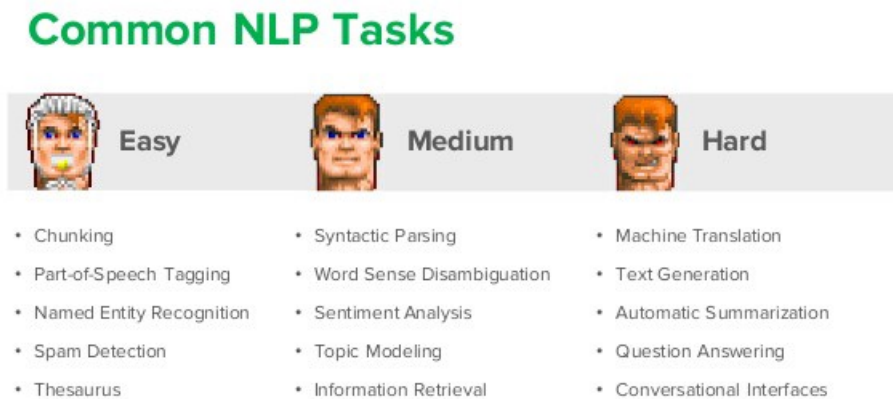


Figura 4.2 Segmentação de tarefas no NLP. (Guts, 2016)

4.3.5 Apache Jena

Apache Jena¹² é um *framework open source* para Web Semântica, escrito na linguagem Java. A biblioteca provê uma [API](#) que facilita a extração e criação de dados nos grafos do [RDF](#), além de oferecer suporte para a linguagem de consulta [SPARQL](#). O objetivo da escolha dessa tecnologia para o projeto, é para facilitar a busca e navegação pelo grafo de entidades (*resources*) no sistema da DBPedia¹³ utilizando [SPARQL](#). Após o [SR](#) extrair entidades das descrições do filme, essas serão buscadas no serviço da Web Semântica estendendo o conhecimento do recurso.

4.3.6 Apache OpenNLP

Apache OpenNLP¹⁴ é um *framework open source* de aprendizado de máquina que é usado para processamento de Natural Language Processing ([NLP](#)). A biblioteca provê uma [API](#) com serviços para geração de *tokens*, sentenças, segmentação, reconhecimento de partes da fala, extração de entidade de nome, geração de *chunks* (pedaços), entre outras tarefas do [NLP](#). A figura 4.2 mostra algumas das tarefas envolvidas no processamento de linguagem natural.

No projeto essa tecnologia será utilizada para o Name Entity Recognition ([NER](#)) e

¹²<https://jena.apache.org>

¹³<http://wiki.dbpedia.org>

¹⁴<https://opennlp.apache.org>

CAPÍTULO 4. UM SISTEMA DE RECOMENDAÇÃO SEMÂNTICO BASEADO EM CONTEÚDO

extração de partes gramaticais presentes na descrição do filme, assim como a geração dos *tokens*. O objetivo é que com essa biblioteca seja possível gerar *tokens* com entidades encontradas, de nomes localizações, como também partes do texto de nomes próprios, substantivos e adjetivos.

4.3.7 Apache Lucene

Apache Lucene¹⁵ é um *framework open source* para sistemas de recuperação de informação e recomendação. O projeto oferece dois principais recursos: indexação e pesquisa de texto. Lucene é muito reconhecido por sua utilidade na implementação em mecanismos de buscas na Internet (McCandless *et al.*, 2010). O projeto também é muito utilizado em sistemas de recomendação com implementação de diversos algoritmos para calcular a similaridade de documentos. No projeto essa tecnologia será utilizada para tirar proveito dos algoritmos de similaridade, como o *cosine similarity* (ver 3.2), possibilitando estender seu funcionamento e permitir integração com a biblioteca, facilitando o seu uso para outras pessoas e outros projetos.

4.4 Funcionamento

As tecnologias apresentadas anteriormente serão utilizadas para construir toda a arquitetura do sistema de recomendação. A proposta é criar uma recomendação baseada em conteúdo, e para este trabalho foi definido o domínio de filmes como exemplo de utilização. Sendo assim, o sistema possui algumas etapas de processamento para viabilizar a recomendação:

- **Coleta dos filmes:** Serão coletados dados dos filmes utilizando o projeto MovieLens¹⁶ (ver 4.4.2).
- **Coleta das preferências do usuário:** Serão coletados dados das preferências dos usuários, ou seja, os filmes de interesse. Nessa etapa poderá ser utilizada o perfil do Facebook¹⁷ para obter tais dados.
- **Pré-processamento dos filmes:** Nessa etapa após a coleta dos filmes, os dados serão previamente processados para a geração de *tokens* com NLP analisando a

¹⁵<https://lucene.apache.org>

¹⁶<https://movielens.org>

¹⁷<https://facebook.com>

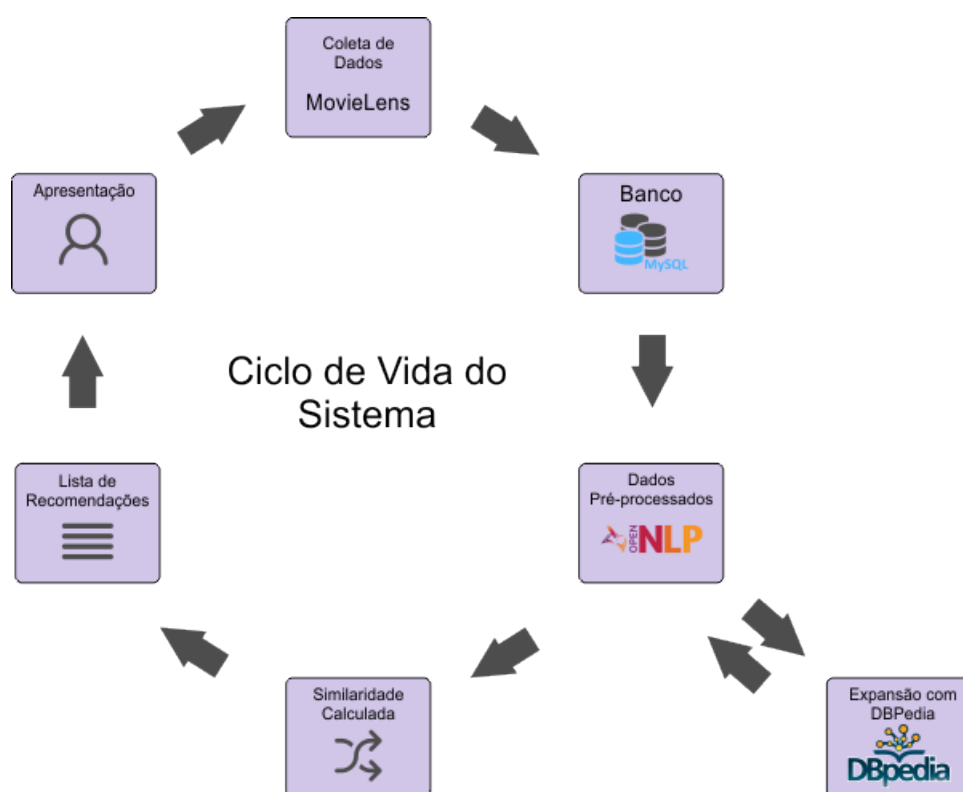


Figura 4.3 Fluxo das camadas do sistema de recomendação

descrição dos itens. Os dados gerados também serão expandidos analisando as entidades extraídas no [NER](#) da sinopse do filme, com o serviço da DBpedia¹⁸. Após todos os processamentos os *tokens* serão persistidos no banco de dados.

- **Cálculo da Similaridade:** Após a etapa de pré-processamento dos filmes, será realizado o cálculo da similaridade para todos os filmes da base dados utilizando uma similaridade do cosseno semanticamente estendida (ver [4.4.3](#)). Os dados gerados serão persistidos no banco para viabilizar o desempenho da geração das recomendações.
- **Geração das recomendações:** Com as similaridades calculadas e persistidas o sistema deverá gerar listas de filmes para recomendação (ver [4.4.3](#)).
- **Apresentação dos resultados:** Apresentação dos resultados: Por fim o sistema apresentará os resultados das recomendações para o usuário.

A [4.3](#) mostra como esse fluxo de funcionalidades é operado por todo o sistema. A seguir será aprofundado mais algumas questões sobre a integração dessas camadas.

¹⁸<http://wiki.dbpedia.org>

4.4.1 Integração entre camadas

4.4.2 Modelo de dados

4.4.3 Algoritmo de similaridade e recomendação

4.5 Sistema em ação

4.6 Sumário

5

Avaliação

Referências Bibliográficas

- (2009). Skoob: Socializando o ato da leitura. Disponível em: <https://archive.is/TwnCc>. [Último acesso em 30 de Novembro de 2017].
- (2009). Web semântica: O futuro das aplicações - java magazine 85. Disponível em: <https://www.devmedia.com.br/web-semantica-o-futuro-das-aplicacoes-java-magazine-85/18493#>. [Último acesso em 02 de Novembro de 2017].
- (2011). *The Filter Bubble: What the Internet is Hiding from You*.
- (2016). Conheça o panorama do e-commerce no brasil. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/artigos/conheca-o-panorama-do-e-commerce-no-brasil,66d975e0dc256510VgnVCM1000004c00210aRCRD>. [Último acesso em 24 de Outubro de 2017].
- (2017). Netflix officially kills star ratings, replacing them with thumbs up and down. Disponível em: <http://variety.com/2017/digital/news/netflix-kills-star-ratings-thumbs-up-thumbs-down-1202023257/>. [Último acesso em 30 de Novembro de 2017].
- (2017). Number of monthly active facebook users worldwide as of 2nd quarter 2017 (in millions). Disponível em: <https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/>. [Último acesso em 24 de Outubro de 2017].
- (2017). Semantic web. Disponível em: https://en.wikipedia.org/wiki/Semantic_Web. [Último acesso em 02 de Novembro de 2017].
- (2017). Skoob: Quem somos. Disponível em: https://www.skoob.com.br/inicio/quem_somos. [Último acesso em 30 de Novembro de 2017].
- Aggarwal, C. C. (2016a). An introduction to recommender systems. In *Recommender Systems*, pages 1–28. Springer International Publishing.
- Aggarwal, C. C. (2016b). *Recommender Systems*, pages 139–166. Springer International Publishing.

REFERÊNCIAS BIBLIOGRÁFICAS

- Andrejs Abele, J. M. (2017). The linking open data cloud diagram. Disponível em: <http://lod-cloud.net/>. [Último acesso em 03 de Novembro de 2017].
- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07*, pages 722–735, Berlin, Heidelberg. Springer-Verlag.
- Baubonienė and Gulevičiūtė (2015). E-commerce factors influencing consumers online shopping decision.
- Bennett, J., Lanning, S., *et al.* (2007). The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA.
- Berners-Lee, T. (2006). Linked data. Disponível em: <https://www.w3.org/DesignIssues/LinkedData.html>. [Último acesso em 03 de Novembro de 2017].
- Berners-Lee, T. (2008). Semantic web architecture. Disponível em: <https://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html>. [Último acesso em 03 de Novembro de 2017].
- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, **284**(5), 34–43.
- Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, **5**(3), 1–22.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, **12**(4), 331–370.
- Crecente (2017). Activision researched using matchmaking tricks to sell in-game items. Disponível em: <http://www.rollingstone.com/glixel/news/how-activision-uses-matchmaking-tricks-to-sell-in-game-items-w509288>. [Último acesso em 24 de Outubro de 2017].
- DBPedia (2014). Dbpedia 2014 data set statistics. Disponível em: <http://wiki.dbpedia.org/services-resources/datasets/dataset-statistics>. [Último acesso em 03 de Novembro de 2017].

- DBPedia (2017). Learn about dbpedia. Disponível em: <http://wiki.dbpedia.org/about>. [Último acesso em 03 de Novembro de 2017].
- del Toro, G., Navarro, B., Cuarón, A., Torresblanco, F., and Augustin, A. (2006). El laberinto del fauno.
- DuBois, P. (2013). *MySQL (Developer's Library)*. Addison-Wesley Professional.
- Ekstrand, M. D., Riedl, J. T., and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, **4**(2), 81–173.
- Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, **35**(12), 61–70.
- Gracia, J. and Mena, E. (2008). *Web-Based Measure of Semantic Relatedness*, pages 136–150. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Guts, Y. (2016). Natural language processing (nlp). Disponível em: <https://www.slideshare.net/YuriyGuts/natural-language-processing-nlp>. [Último acesso em 21 de Fevereiro de 2018].
- Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, **22**(1), 5–53.
- Isinkaye, F., Folajimi, Y., and Ojokoh, B. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, **16**(3), 261 – 273.
- Júnio César de Lima, C. L. d. C. (2005). Ontologias - owl (web ontology language). techreport, Instituto de Informática Universidade Federal de Goiás.
- Keating, G. (2012). *Netflixed: The Epic Battle for America's Eyeballs*. Portfolio/Penguin.
- Konstan, J. A. and Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, **22**(1), 101–123.
- Lewis, J. and Loftus, W. (2000). *Java Software Solutions: Foundations of Program Design*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.
- Lin, D. (1993). Principle-based parsing without overgeneration. In *ACL*.
-

REFERÊNCIAS BIBLIOGRÁFICAS

- LLC, T. I. (2006). Dependency injection demystified. Disponível em: <http://www.jamesshore.com/Blog/Dependency-Injection-Demystified.html>. [Último acesso em 21 de Fevereiro de 2018].
- McCandless, M., Hatcher, E., and Gospodnetic, O. (2010). *Lucene in Action, Second Edition: Covers Apache Lucene 3.0*. Manning Publications Co., Greenwich, CT, USA.
- Naur, P. and Randell, B., editors (1968). *Proceedings, NATO Conference on Software Engineering*, Garmisch, Germany.
- Patil, R. S., Fikes, R., Patel-Schneider, P. F., McKay, D. P., Finin, T. W., Gruber, T. R., and Neches, R. (1992). The darpa knowledge sharing effort: A progress report. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, KR'92*, pages 777–788, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Perry, D. E. and Wolf, A. L. (1992). Foundations for the study of software architecture. *SIGSOFT Softw. Eng. Notes*, **17**(4), 40–52.
- Pressman, R. (2010). *Software Engineering: A Practitioner's Approach*. McGraw-Hill, Inc., New York, NY, USA, 7 edition.
- Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. In *IEEE Transactions on Systems, Man and Cybernetics*, pages 17–30.
- Raggett, D., Lam, J., Alexander, I., and Kmiec, M. (1998). *Raggett on HTML 4 (2Nd Ed.)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Resnick, P. and Varian, H. R. (1997). Recommender systems. *Commun. ACM*, **40**(3), 56–58.
- Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Int. Res.*, **11**(1), 95–130.
- Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA.
- Singhal, A. (2001). Modern information retrieval: A brief overview. **24**, 35–43.

- Singhal, A. (2012). Introducing the knowledge graph: things, not strings. Disponível em: <https://googleblog.blogspot.com.br/2012/05/introducing-knowledge-graph-things-not.html>. [Último acesso em 03 de Novembro de 2017].
- Slimani, T. (2013). Description and evaluation of semantic similarity measures approaches. *International Journal of Computer Applications*, **80**(10), 25–33.
- Snyder, D. and Snyder, Z. (2011). Sucker punch.
- Sommerville, I. (2010). *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edition.
- Tversky, A. (1977). Features of similarity. **84**, 327–352.
- Ventavoli, F. (2014). *Sistema Gerenciador de Banco de Dados MySQL: Guia Prático (Portuguese Edition)*.
- VIANA NETO, J. Q. (2010). Skoob: ambiente virtual de socialização entre leitores e produtores de textos. resreport, Universidade Federal de Pernambuco.
- W3C (2001). W3c semantic web activity. Disponível em: <https://www.w3.org/2001/sw/>. [Último acesso em 02 de Novembro de 2017].
- W3C (2006). Linked data. Disponível em: <https://www.w3.org/standards/semanticweb/data>. [Último acesso em 03 de Novembro de 2017].
- W3C (2009). Owl web ontology language guide. Disponível em: <https://www.w3.org/TR/owl-guide/>. [Último acesso em 02 de Novembro de 2017].
- W3C (2013). Sparql query language for rdf. Disponível em: <https://www.w3.org/TR/rdf-sparql-query/>. [Último acesso em 02 de Novembro de 2017].
- W3C (2014). Rdf. Disponível em: <https://www.w3.org/RDF/>. [Último acesso em 02 de Novembro de 2017].
- Walker, S. J. (2014). Big data: A revolution that will transform how we live, work, and think. *International Journal of Advertising*, **33**(1), 181–183.
- Wellman, D. (2013). What is big data? Disponível em: <https://www.slideshare.net/dwellman/what-is-big-data-24401517>. [Último acesso em 23 de Outubro de 2017].
-

REFERÊNCIAS BIBLIOGRÁFICAS

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.